New Algorithmic Paradigms for Discrete Problems using Dynamical Systems and Polynomials

THÈSE N^{O} 8357 (2018)

PRÉSENTÉE LE 3 MAI 2018

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE THÉORIE DU CALCUL 3
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Damian Mateusz STRASZAK

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

to my wife Marta

Acknowledgments

First and foremost I would like to express my sincere gratitude to my advisor Prof. Nisheeth K. Vishnoi. Not only was he a tremendous mentor and teacher for me but also a great research partner. He has been my closest collaborator throughout my Ph.D. studies and has outstanding contributions to all results presented in this thesis. The overall shape of the thesis and especially the viewpoint on dynamical systems and polynomials are results of our numerous valuable discussions. I am grateful to him for the continuous support, for his motivation and immense knowledge. His advice on both research as well as on my career have been invaluable.

Besides my advisor, I would like to thank the rest of my thesis committee: Profs. Olivier Lévêque, Aleksander Mądry, Rüdiger Urbanke, Piyush Srivastava and Nikhil Srivastava for serving as my committee members and providing many valuable comments despite their busy schedules.

Many thanks to all the people whose guidance I experienced in the very early stages of my research career, especially Aleksander Mądry, Paweł Gawrychowski, Krzysztof Loryś, Marcin Bieńkowski and Jarosław Byrka.

The dissertation would not have come to a successful completion, without my research collaborators: Elisa Celis, Amit Deshpande, Javad Ebrahimi and Tarun Kathuria. I thank them for the stimulating discussions and for the sleepless nights we were working together before deadlines.

My thanks also goes to Jakub Tarnawski who has been my friend and research partner since the beginning of my undergraduate studies, and to other my fellow labmates: Javad Ebrahimi, Ankit Gupta, Ligxiao Huang, Christos Kalaitzis, Sayash Kapoor, Vijay Keswani, Oren Mangoubi, Slobodan Mitrovic, Ashkan Norouzi Fard, Sai Praneeth Reddy and Ozan Yildiz. Taking part in our research meetings, hikes and other outdoor activities allowed me to keep my mind clear during this challenging period of my life. I am grateful to Oren Mangoubi and Robin Scheibler for their help in working out some details of the dissertation.

I would also like to thank my parents and my sister for supporting me mentally throughout writing this thesis.

Finally, I would like to thank my wife, Marta. She was always there cheering me up and stood by me through the good times and bad.

Abstract

Optimization is a fundamental tool in modern science. Numerous important tasks in biology, economy, physics and computer science can be cast as optimization problems. Consider the example of machine learning: recent advances have shown that even the most sophisticated tasks involving decision making, can be reduced to solving certain optimization problems. These advances however, bring several new challenges to the field of algorithm design. The first of them is related to the ever-growing size of instances, these optimization problems need to be solved for. In practice, this forces the algorithms for these problems to run in time linear or nearly linear in their input size. The second challenge is related to the emergence of new, harder and harder problems which need to be dealt with. These problems in most cases are considered computationally intractable because of complexity barriers such as NP completeness, or because of non-convexity. Therefore, efficiently computable relaxations for these problems are typically desired.

The material of this thesis is divided into two parts. In the first part we attempt to address the first challenge. The recent tremendous progress in developing fast algorithm for such fundamental problems as maximum flow or linear programming, demonstrate the power of continuous techniques and tools such as electrical flows, fast Laplacian solvers and interior point methods. In this thesis we study new algorithms of this type based on continuous dynamical systems inspired by the study of a slime mold Physarum polycephalum. We perform a rigorous mathematical analysis of these dynamical systems and extract from them new, fast algorithms for problems such as minimum cost flow, linear programming and basis pursuit.

In the second part of the thesis we develop new tools to approach the second challenge. Towards this, we study a very general form of discrete optimization problems and its extension to sampling and counting, capturing a host of important problems such as counting matchings in graphs, computing permanents of matrices or sampling from constrained determinantal point processes. We present a very general framework, based on polynomials, for dealing with these problems computationally. It is based, roughly, on encoding the problem structure in a multivariate polynomial and then recovering the solution by means of certain continuous relaxations. This leads to several questions on how to reason about such relaxations and how to compute them. We resolve them by relating certain analytic properties of the arising polynomials, such as the location of their roots or convexity, to the combinatorial structure of the underlying problem.

We believe that the ideas and mathematical techniques developed in this thesis are only a beginning and they will inspire more work on the use of dynamical systems and polynomials in the design of fast algorithms.

Keywords: Discrete Problems, Dynamical Systems, Polynomials, Counting, Sampling, Optimization, Continuous Relaxations, Physarum Polycephalum, Linear Programming, Convex Optimization, Real Stable Polynomials, Anti-Concentration, Determinantal Point Processes, Maximum Entropy Distributions

Résumé

L'optimisation est un outil fondamental dans la science moderne. De nombreuses tâches en biologie, économie, physique et informatique peuvent s'exprimer comme des problèmes d'optimisation. Considérons l'exemple de l'apprentissage automatique : les progrès récents ont montré que même les tâches les plus complexes impliquant une prise de décision peuvent se réduire à résoudre des problèmes d'optimisation. Ces progrès, néanmoins, amènent plusieurs nouveaux défis dans le domaine de la conception d'algorithmes. Le premier est lié à la taille croissante des instances pour lesquelles ces problèmes d'optimisation ont besoin d'être résolus. En pratique, cela force les algorithmes pour ces problèmes de s'exécuter en temps linéaire en la taille de l'entrée, ou presque. Le second défi est lié à l'émergence de nouveaux problèmes, de plus en plus difficiles, qui ont besoin d'être traités. Ces problèmes sont, dans la plupart des cas, considérés comme impossibles à résoudre calculatoirement, à cause de barrières de complexité comme la NP-complétude, ou bien à cause de la non-convexité. En conséquence, ils ne peuvent souvent pas avoir de solutions algorithmiques exactes.

Cette thèse est divisée en deux parties. Dans la première partie, on tente de s'attaquer au premier défi. Récemment, les formidables progrès dans le développement d'algorithmes rapides pour des problèmes aussi fondamentaux que le flot maximal ou la programmation linéaire démontrent la puissance des techniques continues et des outils comme les flots électriques, la résolution rapide du laplacien, ou les méthodes de points intérieurs. Dans cette thèse, nous étudions de nouveaux algorithmes de ce type, basés sur les systèmes dynamiques continus, inspirés par l'étude du champignon Physarum polycephalum. Nous faisons une analyse mathématique rigoureuse de ces systèmes dynamiques, et en extrayons de nouveaux algorithmes rapides pour des problèmes comme le flot de coût minimal, la programmation linéaire ou la poursuite de base.

Dans la deuxième partie de cette thèse, nous développons de nouveaux outils pour se rapprocher du deuxième défi. Pour ce faire, nous étudions une forme très générale de problème d'optimisation discrète, et ses extensions à l'échantillonage et au dénombrement, ce qui s'applique à une classe de problèmes importants comme le comptage de couplages dans les graphes, le calcul de permanents de matrices, ou l'échantillonage à partir de processus ponctuels déterminantaux contraints. Nous présentons un cadre très général, basé sur les polynômes, pour traiter ces problèmes calculatoirement. Il est basé en gros sur l'encodage de la structure du problème dans un polynôme à plusieurs variables, et sur le fait que les solutions peuvent alors se retrouver par certaines relaxations continues. Cela amène plusieurs questions sur comment raisonner sur de telles relaxations, et comment les calculer. Nous les résolvons en reliant certaines propriétés analytiques des polynômes qui apparaissent, comme le lieu de leurs racines ou leur convexité, à la structure combinatoire du problème sous-jacent.

Nous croyons que les idées et les techniques mathématiques développées dans cette thèse sont juste un commencement, et inspireront encore des travaux sur l'utilisation des systèmes dynamiques et des polynômes dans la conception d'algorithmes rapides.

Mots-clés: Problèmes discrets, systèmes dynamiques, polynômes, dénombrement, échantillonage, optimisation, relaxations continues, Physarum Polycephalum, programmation linéaire, optimisation convexe, polynômes stables réels, anti-Concentration, processus ponctuels déterminantaux, distributions d'entropie maximale

Contents

Chapter 1

Introduction

Optimization is a fundamental tool in modern science. Numerous tasks in biology, economy, physics and computer science can be cast as optimization problems. Consider the example of machine learning: recent advances have shown that even the most sophisticated tasks involving decision making, can be reduced to solving certain optimization problems. These developments however present new challenges for algorithm design. The first of them concerns the explosion of the size of data used by today machine learning systems. Data is being produced at an incredibly fast rate, forcing the algorithms to process it in time essentially proportional to its size, i.e., nearly linear. Secondly, with the emergence of deep learning and other related techniques, the problems which arise in such settings become much more sophisticated, thus no longer convex, and often even **NP**-hard. Still, algorithms are desired to solve these intractable problems very efficiently and with good precision. Several ways of dealing with these types of intractability are currently studied. Among them, there are attempts to go beyond worst case in the analysis of algorithms by studying special classes of instances, random instances or randomly perturbed instances [ST09]. Another idea is to design efficient relaxations of these problems, i.e., simpler problems which can be solved rapidly, yet approximating the original ones with good accuracy. Several approaches of this kind are presented in this thesis.

Optimization, as a research field, can be divided, roughly, into two ”schools”: combinatorial optimization and continuous optimization. The former is typically concerned with discrete objects such as graphs, matchings, spanning trees or 0-1 sequences and the latter deals with spaces, such as the hypercube $[0,1]^m$ and studies vectors, matrices and functions defined on continuous domains. Numerous important results have been obtained in both these subfields.

In combinatorial optimization, very early on, polynomial time algorithms for several fundamental graph problems have been designed, including the shortest path problem [Dij59], the minimum spanning tree problem [Kru56], the maximum flow problem [Ful56] and the maximum matching problem in the case of bipartite graphs and finally for non-bipartite graphs as well [Edm65]. Another important achievement of this field is the introduction of matroids (see [Oxl06]) which generalize “independence” in combinatorial structures and in particular allow to capture spanning trees or linearly independent sets of vectors. The problem of matroid intersection has been shown to be efficiently solvable [Edm01] which then had several important implications. Numerous approximation algorithms have been also developed for various **NP**-hard problems, such as set-cover problem or max-cut, see also [Vaz13] for more examples.

On the continuous side, among the most important developments are general tools for minimizing convex functions, which include the gradient descent algorithm, its accelerated variant [Nes13], Newton's method, the multiplicative weight update method [AHK12] and many others, see [Bub14] for a survey. One of the most important general optimization primitives – linear programming has been shown to be polynomial time solvable using continuous techniques by [Kha80]. Soon after his work, several other polynomial time algorithms have been developed for this problem using the interior point method [Kar84, Ren88] and very recently the fastest known algorithm for linear programming has been obtained by [LS14].

On the interface of these two worlds, recently, enormous progress has been achieved in the design of fast algorithms for the maximum flow problem. This was due to a breaktrough result of Spielman and Teng [ST04] who proved that linear systems involving matrices called graph Laplacians can be solved very efficiently, in time nearly linear in the number of edges in the graph. This result allowed to finally overcome the running time barrier of, roughly, $O(mn^{2/3})$ (where n is the number of vertices and m is the number of edges in the graph) for the maximum flow problem, which stood essentially for more than 35 years [ET75, GR98]. More specifically, for the approximate variant of undirected maximum flow, through a sequence of improvements [CKM$^+$11, LRS13] finally a nearly linear algorithm has been developed [She13, KLOS14, Pen16]. For the directed variant, an improvement with running time of $\widetilde{O}(m^{10/7})$ was established by [Mad13] and $\widetilde{O}(mn^{1/2})$ by [LS14], both of them relied on the interior point method for linear programming.

For these improvements to be possible it was crucial to abandon the classical discrete approach to this discrete problem and attack it using continuous methods. On the other hand, fast convergence of continuous algorithms applied to this problem was a consequence of the underlying combinatorial structure. Thus really a *synthesis* of these two different sets of techniques resulted in such a significant improvement on this problem.

In contrast to the max flow which is polynomial time solvable, let us now consider several combinatorial **NP**-complete problems for which, historically, applying continuous ideas brought significant progress. This was the case for the max-cut problem on graphs, for which in [GW95] a novel semidefinite programming relaxation has been developed which provides the best known approximation guarantee for this problem.

Another important example, for which a synthesis between continuous and discrete techniques not only improved the running time, but also made the problem tractable at all, is the sparsest-cut problem for undirected graphs. A simple algorithm based on eigenvalue computation and Cheeger's inequality [AM84] gives a non-trivial approximation guarantee, which otherwise would be hard to achieve using just combinatorial techniques. Currently the best known polynomial time approximation algorithm for this problem is based on a continuous relaxation as a semidefinite program and yields $O(\sqrt{\log n})$-approximation [ARV04].

Finally, let us mention the subdeterminant maximization problem: given a PSD matrix $L \in \mathbb{R}^{m\times m}$ we ask for a set $S \subseteq [m]$ of size n which maximizes $\det(L_{S,S})$, where $L_{S,S}$ is a square submatrix of L with rows and columns coming from the set S. It has been shown [Nik15] by means of a convex relaxations that there is an e^n-approximation algorithm for solving this problem. This has been also recently extended to partition matroids in [NS16].

To discuss further examples, we leave for a moment the realm of optimization problems and consider the problem of sampling. In sampling problems, instead of finding the maximum-weight solution (as in optimization) one is interested in sampling a solution with probability proportional to its weight. This can be seen as a robust variant of optimization, which for many important cases is actually harder and more general than the corresponding optimization problem. Consider the example of perfect matchings in a bipartite graph. Finding the maximum weight matching is a polynomial-time solvable problem, whereas sampling a matching or equivalently counting [JVV86] (computing the sum of all weights) is already #**P**-hard [Val79]. For this problem, discrete methods based on Markov Chains have been shown to perform very well [JSV04], yet this approach is quite specialized and does not extend to small modifications of the problem. Another approach for counting perfect matchings, or equivalently for computing permanents of non-negative matrices, based on a continuous relaxation has been proposed by Gurvits [Gur06]. Gurvits proves that his relaxation to this problem yields a bounded approximation ratio because of a certain underlying polynomial having "good" analytic properties, i.e., being *real stable*. Real stable polynomials have recently found numerous other applications in mathematics [BB09, MSS15] and computer science [MSS13, AOG15, NS16]. However, the result [Gur06] remained the only use of these polynomial techniques to counting and sampling. This brings us to the question of how general the method of Gurvits really is, or in other words

Are there more general polynomial-based continuous relaxations to counting problems?

Going back to our discussion on optimization problems, let us now consider an immense source of algorithms for solving them – the nature. Recent results on computational aspects of natural phenomena clearly show how well designed natural algorithms really are, examples include the work of [Cha12] on bird-flocking, results relating sex and a robust algorithm in computer science [CLPV14] and the analysis of efficiency of evolution in [Val09, Vis15], to name a few. In fact, by inspecting the above it seems that nature is indeed addressing the most important challenges of optimization: it provides fast algorithms and overcomes intractability. However, since the algorithms implemented by nature run in continuous time on "analog computers" it is not clear whether they can be claimed efficient also in the classical sense.

In this thesis we study examples of such algorithms inspired by the growth process of a slime mold Physarum polycephalum. In a striking experiment, [NYT00] showed that this single-celled organism, could solve the shortest path problem on a maze. This sparked interest in its computational abilities. Roughly speaking, when the organism is distributed throughout a maze with “food” only at the entry and the exit points of the maze, it quickly reaches an equilibrium where it occupies only the shortest path in the maze connecting the two endpoints. Subsequently, the inner workings of Physarum were mathematically modeled as continuous time dynamical systems and natural discretizations of these dynamics were proposed as algorithms to solve several problems: the shortest path problem, a certain variant of minimum cost flow and linear programming (see [TKN07, BMV12, IJNT11, JZ12]). This brings us to our second set of questions

Are the algorithms based on Physarum dynamics provably correct?

Could they be efficient?

In this thesis, we answer both these questions affirmatively. We prove convergence for all variants of the Physarum dynamics, both in the continuous case as well as for the discretization. Moreover, we provide bounds on the number of steps required to get close to an optimal solution. As a consequence we conclude that the variant of the dynamics for solving flow problems yields a polynomial time algorithm; a similar conclusion holds for linear programming, whenever the constraint matrix is totally unimodular.

Apart from that we prove that the Physarum dynamics for linear programming has a natural interpretation from the viewpoint of optimization. It can be seen as an istance of interior point method based on the so called *entropy barrier*. This draws a connection to other interior point methods studied in the linear programming literature, yet makes the Physarum-based algorithm unique, as such a barrier function does not seem to have been studied before.

In the second part of the thesis we provide a general polynomial-based framework for sampling and counting problems, thus answering the former question asked in this section. To this end we show how one can use multivariate polynomials to encode a given structure in discrete counting problems so that the solution can be then recovered as a sum of its certain coefficients. Then, depending on the setting, such coefficients can be recovered either exactly, in polynomial time – using polynomial interpolation, or using continuous relaxations.

The relaxations we introduce for recovering coefficients generalize the work of [Gur06] and [NS16]. We prove bounds on their approximation guarantees under the condition of real stability of the underlying polynomial and a certain combinatorial structure of the counting problem at hand. To prove computability of these relaxations, we study a general family of entropy-maximization problems and show bounds on the bit complexity of close to optimal solutions. Such bounds then allow us to run the ellipsoid algorithm to solve these relaxations.

The techniques used in both parts of the thesis mainly draw from convex optimization, polyhedral combinatorics, analysis and linear algebra. An important aspect of our results is the synthesis of discrete and continuous techniques, which in particular materializes itself in the study of analytic properties of polynomials encoding certain combinatorial structures and in the design of continuous relaxations for discrete counting problems.

We believe that the results of this thesis, both from the first and the second its part will inspire future work on new approaches to fast algorithms for combinatorial optimization problems and efficient methods for approximately solving intractable counting problems. In fact, it is of no doubt that there are dozens of other, incredibly efficient dynamical systems developed by nature, waiting to be discovered and turned into fast algorithms. Similarly, our polynomial-based framework for solving counting problems is constructed using one particular way of encoding combinatorial structure in a polynomial. Certainly there are reasons to believe that by using other encodings, using more variables, or switching to non-commutative or matrix-variables one can arrive at different – better behaved polynomials and thus obtain stronger relaxations. We leave these as open questions, hopefully to be resolved soon.

Chapter 2

Dynamical Systems for Combinatorial Optimization Problems

In this chapter we give a brief introduction to dynamical systems which are the main objects of study in Chapters 4 and 5 of this thesis. We discuss the main problems and questions related to the analysis of dynamical systems. Subsequently we introduce the Physarum dynamics – a nature-inspired dynamical system studied in this thesis and outline the contributions of Chapters 4 and 5.

2.1 Introduction to Dynamical Systems

The theory of dynamical systems is a large, standalone branch of mathematics, with dozens of books written on this topic (see e.g. [Per01]), hence we will not be able to even touch upon every aspect of it. Instead, in this exposition we focus on delivering as much intuition as possible, while sometimes resorting to certain simplification and being not entirely formal.

Dynamical systems in mathemathics are typically divided into two classes: continuous and discrete. Both of them consist of a domain X and a "rule" according to which a point moves in the domain. The difference is though that in a discrete dynamical system the time at which the point moves is discrete $t = 0, 1, 2, \ldots$, while in the continuous dynamical system it is continuous, i.e., $t \in [0, \infty)$. More formally

Continuous Dynamical System

A continuous dynamical system consists of a domain $\Omega \subseteq \mathbb{R}^n$ and a function $G : \Omega \to \mathbb{R}^n$. For any point $s \in \Omega$ we define a solution (a trajectory) originating at s to be a curve $x : [0, \infty) \to \Omega$ such that

$$x(0) = s \quad \text{and} \quad \frac{d}{dt}x(t) = G(x(t)) \quad \text{for } t \in (0, \infty).$$

For brevity, the differential equation in the definition above is often written as $\dot{x} = G(x)$. The definition essentially says that a solution to a dynamical system is any curve $x : [0, \infty) \to X$ which is tangent to $G(x(t))$ at $x(t)$ for every $t \in (0, \infty)$, thus $G(x)$ gives a direction to be followed at any given point $x \in X$. Continuous dynamical systems are ubiquitous in physics, mathematics and biology. They are used to describe physical laws, such as the Newtonian dynamics, and appear when modeling various processes in nature, such as the atmospheric convection.

Existence of Solutions. In the study of continuous dynamical systems, the first question one normally asks is whether a solution to such a system exists (for a given initial point $s \in \Omega$). This often turns out to be a nontrivial question which depends on the properties of the function G and on the structure of the domain. In fact, existence (and uniqueness) of solutions is one of the main topics of study in the field of differential equations. Intuitively, the reason why a solution may not exist globally is that the solution curve might try to "escape" the domain Ω and thus it might be well defined only on a finite interval $[0, T)$ for some $T > 0$. Thus, proving that a dynamical system has global solutions is then equivalent, roughly, to showing that the solution curves never come "too close" to the boundary in finite time (even though they might attain it at infinity).

Long-term behavior of trajectories. Given that global solutions to a dynamical system of interest exist one often asks about their asymptotic behavior. This question is relevant especially when studying systems modelling certain processes (in engineering, chemistry, drug design, etc.) for which one would like to know if or when do they stabilize, and if so what are the states towards which they converge. In the table below we summarize what may happen.

Possible Long-Term Behaviors of Continuous Dynamical Systems

(1) **Convergence** to a fixed point;

(2) **Cyclic** behavior;

(3) **Chaotic** behavior.

For a formal discussions of these, we refer to [Per01]. Here, instead, we just give some intuitions. Behaviors (1) and (2) are considered tame, where either the trajectories have limits at $t \to \infty$ or they are "trapped" in infinite loops such as $x(t) = (\sin(t), \cos(t))$ (the solution traverses a cycle indefinitely). The chaotic behavior occurs when, roughly, small perturbations of the initial point lead to significantly different solutions (sometimes called the "butterfly effect") and thus the trajectories are "chaotic". While one might think that the chaotic behavior might only happen for obscure, specially crafted systems, this is not the case. In fact, even the 3-dimensional model of atmospheric convection developed by Lorenz in 1963 turned out to admit chaotic behavior, and thus gave evidence that weather prediction is a hard, if not impossible, task.

Types of convergence for optimization. In the field of Theoretical Computer Science, one is often interested in dynamical systems for solving optimization problems. In the standard setting, one considers an objective (cost function) $C : \Omega \to \mathbb{R}$ and the goal is to find its minimum value, we denote it by $C^\star := \min_{x \in \Omega} C(x)$. One would like to construct and analyze dynamical systems, whose trajectories converge to optimal solutions. Since the set of optimal solutions, i.e., $\{x \in \Omega : C(x) = C^\star\}$ might be quite large, there are two qualitatively different ways to define such a convergence.

Possible Types of Convergence for Optimization Problems

Let $C : \Omega \to \mathbb{R}$ be a cost function with $C^\star := \min_{x\in\Omega} C(x)$ and let $x : [0, \infty) \to \Omega$ be a trajectory, then we say that

(1) x converges **in value** (to an opt. solution), if $\lim_{t\to\infty} C(x(t)) = C^\star$,

(2) x converges **pointwise** (to an opt. solution), if $\lim_{t\to\infty} x(t) = x^\star$ and $C(x^\star) = C^\star$.

Note that pointwise convergence is a strictly stronger notion than convergence in value and that convergence in value does not even guarantee convergence of the dynamical system, as all kinds of oscillatory behaviors may happen. For an example we refer to Figure 2.1 where a trajectory, which converges in value, yet oscillates around the set of optimal values, is shown.

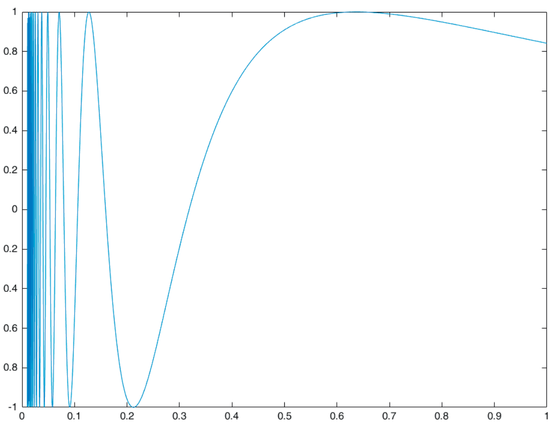

Figure 2.1: A trajectory $x(t) = (\frac{1}{t}, \sin t)$ shows convergence in value for the cost function $C(x_1, x_2) = x_1$, yet the trajectory admits oscillatory behavior and does not converge pointwise.

Discrete dynamical systems. Before we discuss further questions involving dynamical systems let us introduce the class of discrete dynamical systems, which in computer science show up much more frequently than their continous counterpart.

Discrete Dynamical System

A discrete dynamical system consists of a domain Ω and a function $F : \Omega \to \Omega$. For any point $s \in \Omega$ we define a solution (a trajectory) originating at s to be the infinite sequence of points $\{x^{(k)}\}_{k\in\mathbb{N}}$ with $x^{(0)} = s$ and $x^{(k+1)} = F(x^{(k)})$ for every $k \in \mathbb{N}$.

Such dynamical systems can, in particular, model all kinds of computation, for example the state of a Turing machine, the memory of a computer when a particular program is run, or on a higher level, the current iterate in iterative improvement algorithms such as gradient descent or interior point method. All the issues and questions discussed for continuous systems apply to discrete systems as well. The issue of existence of solutions seems obvious here, but in fact the difficulty typically lies in proving that the range of the transition function F is contained in Ω

(which we assume in the definition). Similarly, the same long-term behaviors of solutions (as in the continuous case) show up in the discrete case and the issue of convergence in value versus pointwise convergence is also present.

Discretization of dynamical systems. Given a continuous dynamical system one is often interested in discretizing it, for instance to simulate its behavior on a computer, or in order to turn it into a discrete algorithm. The simplest way to do it is perhaps via the Euler discretization.

Euler Discretization of a Continuous Dynamical System

Consider a dynamical system given by $\dot{x} = G(x)$ on some domain $\Omega \subseteq \mathbb{R}^n$. Then, given a step size $h \in (0,1)$, its **Euler discretization** is a discrete dynamical system (Ω, F), where

$$F(x) = x + h \cdot G(x).$$

This is perfectly compatible with the intuition that a continuous dynamical system at x is simply moving "just a little bit" along $G(x)$. Note that formally the above definition is not quite correct, as $F(x)$ might land outside of Ω – this is a manifestation of the existence issue for discrete systems – sometimes in order to achieve existence one has to take h very small, and in some cases it might not be possible at all. The Euler discretization is only one of many other possible discretization methods, it uses just the first term in the Taylor expansion of $G(x)$ to approximate it. In fact, one can also define higher order discretizations of continous dynamical systems. However, this raises the computational cost of performing one step, requires access to derivates of G and does not always provide an improvement over the first order method.

It is important to note that when applying the Euler discretization there is barely any guarantee that the resulting discrete system will be close to the original one. One can prove that if we consider the kth iterate of the Euler discretization $x^{(k)}$ and the state $x(k \cdot h)$ of the original system, then roughly $\|x^{(k)} - x(k \cdot h)\| \leqslant h \cdot e^{Lhk}$, where L is the Lipschitz constant of the function G. Note that, in particular, this bound quickly deteriorates to infinity as we increase k. Thus analyzing a discretization of a dynamical system is in general a nontrivial task, even if the underlying continuous system is well understood.

Rate of convergence. In computer science, in contrast to the mathematical perspective on dynamical systems, a qualitative convergence result is usually not enough and one is interested in *non-asymptotic* convergence bounds. In other words, suppose we study a dynamical systems aimed to optimize a given cost function $C(x)$ over $x \in \Omega$, then the question of interest becomes: how quickly does the trajectory $x^{(k)}$ reach a close neighborhood of the optimal solution? More formally, for convergence in value one asks for a bound on k for which $C(x^{(k)}) \leqslant C^\star + \varepsilon$ or $C(x^{(k)}) \leqslant C^\star(1+\varepsilon)$ in the case when $C^\star$ is positive. For pointwise convergence, the question becomes: After how many steps k, $\|x^{(k)} - x^\star\| < \varepsilon$ holds? This question is crucial from the algorithmic perspective, as it says how much time does an algorithm take to solve a particular problem. We are interested in a bound on the number of steps k needed to reach an ε-approximate solution as a function of ε. Methods such as gradient descent typically achieve a dependency of the form $O(\varepsilon^{-1})$ or $O(\varepsilon^{-2})$ while generally the best we can hope for is a logarithmic dependency on ε, i.e., $O(\log \varepsilon^{-1})$.

Optimization Viewpoint. Finally we discuss a question which is important especially from the viewpoint of Theoretical Computer Science and Optimization. Suppose we are given a dynamical system (either continuous or discrete) which we know optimizes a cost function $C : \Omega \to \mathbb{R}$. To prove convergence the most common techniques are potential functions (Lyapunov functions) or arguments based on contraction. These, while they often allow one to show existence of a unique limit, do not provide much insight about the limit and convergence process itself. What is much more useful is an interpretation of the dynamics in terms of optimization. In other words, we would like to know what is the optimization primitive which stands behind the convergence of the dynamics. Recall that the basic setting in optimization is that we are given a domain $\Omega \subseteq \mathbb{R}^n$ and we would like to minimize a function $C : \Omega \to \mathbb{R}$ over this domain, i.e., find $\min_{x \in X} C(x)$. Several different general methods have been developed for this problem in the area of optimization. The most basic of them being perhaps the gradient descent method which starts at an arbitrary point $x^{(0)} \in X$ and makes small step along the opposite direction of the gradient at any given time step, i.e., the $(k+1)$th iterate of $x^{(k+1)}$, given the kth one, is defined as follows

$$x^{(k+1)} = x^{(k)} - h \cdot \nabla C(x^{(k)}),$$

where $h > 0$ is a (typically very small) step size. Among other well known methods are the Newton method, mirror descent and several variants of non-Euclidean gradient descent. Being able to put a given natural dynamical system in one of these frameworks leads to its much deeper understanding, and using the well-established tools of optimization yields alternative ways to analyze its convergence and possibly its convergence rate. Further, when none of the well known general optimization schemes captures the studied dynamical system, the situation is even more interesting as it might give rise to new optimization primitives!

Questions regarding dynamical systems. Concluding our discussion, here are the most important question about a dynamical system one can ask from the viewpoint of optimization.

Questions Regarding Dynamical Systems for Optimization

(1) Are there global solutions to this dynamical system?

(2) Do they converge? Pointwise, in value? What is the convergence rate?

(3) Can the dynamical system be explained from the optimization viewpoint?

Physarum inspired dynamical systems. In this thesis we discuss dynamical systems inspired by the study of a slime mold *Physarum polycephalum.* This single celled organism has been the source of much excitement among biologists and computer scientists due to its ability to solve complex optimization problems. This started with an experiment [NYT00] that showed the slime mold could solve the shortest path problem on a maze. Roughly speaking, the slime mold is a collection of tubes which it uses to transport food across its body and it is in this process that it shows the ability to compute. Soon after, the time evolution of Physarum was captured by mathematical biologists [TKN07] giving rise to a dynamical system to model its behavior. This dynamical system was aimed to solve the shortest path problem (see Figure 2.2) on undirected graphs. Notably, [IJNT11] introduced a slightly different variant of the dynamics, which turns out to be applicable to the same problem but for directed graphs. This gave rise to two lines of work on Physarum dynamics: undirected and directed. Both variants of the dynamics were then significantly generalized to other "undirected" and "directed" problems.

In fact, already the basic variant of the dynamics presented by [IJNT11] has been designed for a more general problem, called the transshipment problem (see Figure 2.2 for a definition). In a subsequent paper [JZ12] a significant generalization of the above was proposed, which applies to linear programming.

In a different line of work, [BMV12] considered the undirected transshipment problem (based on the idea of [IJNT11]) and finally in our work [SV17c] we introduced an undirected Physarum dynamics generalizing both the shortest path problem and the transhipment problem; it solves the so called weighted basis pursuit problem (see Figure 2.2).

The box in Figure 2.3 succinctly summarizes all the different variants of the Physarum dynamics considered in these works. Note that the original variant of the dynamics – for the shortest path problem – has a very intuitive explanation in the language of electrical networks. One thinks of x as a vector of edge conductances which evolves in time. The vector q determines an electrical flow on the graph G, where the resistance of each edge $e \in E$ is set to ${}^{c_e}/_{x_e}$ and we pump in one unit of current at s and take it out from t. Equivalently, q is determined as the unit flow from s to t with minimum energy, i.e.,

$$q := \underset{f:s-t \text{ flow}}{\operatorname{argmin}} \sum_{e \in E} \frac{c_e f_e^2}{x_e}.$$

Qualitatively, according to (2.2) the resistance of an edge increases if the magnitude of the current is high, and reduces otherwise. In fact, this interpretation has been very important both conceptually and mathematically in proving properties of this dynamics.

Note that the Physarum dynamics is originally defined in continuous time. However for the purpose of its algorithmic applications, the Euler disretization of the dynamics has been also extensively studied.

The table in Figure 2.4 summarizes the contributions in the study of all variants of the Physarum dynamics. We remark that while the question of **existence of solutions** is highly non-trivial for the directed dynamics, it can be established quite easily for the undirected variants and hence it is not considered in Figure 2.4. The second question – **convergence** – is interesting and non-trivial for all variants of the dynamics. A more detailed discussion on types of convergence and convergence rates is provided in the two subsequent sections. Finally, the third question on **optimization viewpoint** was open for both the directed and undirected variants of the dynamics even in the simplest of cases. In this thesis we provide such interpretations of both these dynamics from the viewpoint of optimization, mores details are provided in Sections 2.2 and 2.3 below.

2.2 Directed Physarum Dynamics

The goal of Chapter 4 of this thesis is to resolve all 3 questions: existence, convergence and optimization viewpoint for the directed Physarum dynamics in its most general form: for linear programming. We start by considering the question of existence and convergence for the continuous case. For clarity, here we state an informal version of our result, see Theorem 4.2 for a formal exposition.

Definitions of Computational Problems

The Undirected (Directed) Shortest Path Problem:
Given an undirected (directed) graph $G = (V, E)$ with positive edge lengths $c \in \mathbb{R}^E_{>0}$ and two vertices $s, t \in V$, find the shortest (directed) path between s and t in G.

The Undirected (Directed) Transshipment Problem:
Given an undirected (directed) graph $G = (V, E)$ with positive edge lengths $c \in \mathbb{R}^E_{>0}$ and a vector of demands $b \in \mathbb{R}^V$ such that $\sum_{v \in V} b_v = 0$, find a flow vector $f \in \mathbb{R}^E$ ($f \in \mathbb{R}^E_{\geqslant 0}$) that routes the demands as specified by b (i.e. $Bf = b$, where B is the signed incidence matrix of G) and minimizes $\sum_{e \in E} c_e |f_e|$ (minimizes $\sum_{e \in E} c_e f_e$).

Linear Programming:
Given a matrix $A \in \mathbb{R}^{n \times m}$, a vector $b \in \mathbb{R}^n$ and a cost vector $c \in \mathbb{R}^m$, find

$$\min\ c^\top x \quad \text{s.t.}\ \ Ax = b, \quad x \geqslant 0. \tag{2.1}$$

Weighted Basis Pursuit:
Given a matrix $A \in \mathbb{R}^{n \times m}$, a vector $b \in \mathbb{R}^n$ and a cost vector $c \in \mathbb{R}^m_{\geqslant 0}$, find

$$\min \sum_{i=1}^{m} c_i |x_i| \quad \text{s.t.}\ \ Ax = b.$$

Figure 2.2: Computational problems solved by the Physarum dynamics.

The Physarum Dynamics

Let $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}^m$. **The Physarum dynamics** is defined as a dynamical system over the domain $\Omega = \mathbb{R}^m_{>0}$ by the system of differential equations

$$\begin{aligned} \dot{x} &= |q| - x \qquad \text{(undirected)} \\ \dot{x} &= q - x \qquad \text{(directed)}, \end{aligned} \tag{2.2}$$

where $q \in \mathbb{R}^m$ is a vector calculated as a function of the current point $x = x(t)$ according to the following rule

$$q = WA^\top (AWA^\top)^{-1} b,$$

and $|q|$ is its entrywise absolute value. In the above $W \in \mathbb{R}^{m \times m}$ is a diagonal matrix with $W_{i,i} := \frac{x_i}{c_i}$.

Special cases:

(1) **The transshipment problem.** The matrix A is equal to $B \in \mathbb{R}^{V \times E}$ – the signed incidence matrix of the graph G, i.e., for every edge $e = vw$ the column corresponding to e in B is $e_v - e_w$.

(2) **The shortest path problem.** As above, and additionally, the vector b is equal to $\chi_{s,t} = e_s - e_t \in \mathbb{R}^V$.

Figure 2.3: The definition of Physarum dynamics for various problems. Above, e_v denotes the vth standard basis vector in the linear space $\mathbb{R}^V$.

	Existence		Convergence			
	Directed Dynamics		Directed Dynamics		Undirected Dynamics	
	Continuous	Discrete	Continuous	Discrete	Continuous	Discrete
Shortest Path	[BBD+13]	[BBD+13]	[IJNT11]	[BBD+13]	[BMV12]	[BBD+13]
Transshipment	[SV16b]	[SV16a]	[IJNT11]	[SV16a]	[BMV12]	[SV16a]
LP	[SV16b]	[SV16b]	[SV16b]	[SV16b]	[SV17c]	[SV17c]

Figure 2.4: Summary of results on existence and convergence for all variants of the Physarum dynamics. LP stands for Linear programming (directed) and Basis Pursuit (undirected). The results marked in blue are contributions of this thesis.

Theorem 2.1 (*Existence and Convergence – Informal*)

Let (A, b, c) be an instance of linear programming, as in (2.1), and $s \in \mathbb{R}^n_{>0}$ be any initial condition.

(1) There exists a unique solution $x : [0, \infty) \to \mathbb{R}^m_{>0}$ to the dynamics originating at s.

(2) The trajectory $x : [0, \infty) \to \mathbb{R}^n_{>0}$ converges to an optimal solution $x^\infty \in \mathbb{R}^n$ of the underlying linear program. Moreover, the convergence rate is exponential, i.e.,

$$\|x(t) - x^\infty\| \leqslant e^{-\alpha t}$$

for some $\alpha > 0$ which depends on the input and s only.

Thus we obtain pointwise convergence with an exponential convergence rate. We also show an analogue of Theorem 2.1 for the case of discrete Physarum dynamics (Theorem 4.3) – we show that it converges in value whenever the step size h is small enough, we also provide specific bounds on the step size and the convergence rate – the number of steps requires to reach an ε-approximation of the optimal value is proportional to ε^{-3}.

For the special case of the transshipment problem, we sharpen this result and provide an algorithm whose convergence rate is proportional to $\log \frac{1}{\varepsilon}$ and hence leads to a polynomial time algorithm for solving this problem (see Theorem 4.4).

The second theorem we would like to state here, provides an optimization viewpoint on the directed Physarum dynamics. We present an informal version of our theorem here, see Theorem 4.5 for a formal treatment.

Theorem 2.2 (*Optimization Viewpoint – Informal*)

The directed Physarum dynamics for linear programming is a continuous time gradient descent for the objective function $c^\top x$ run with respect to a Riemmanian manifold structure on the positive orthant $\mathbb{R}^n_{>0}$ induced by the Hessian of the entropy function $H(x) = \sum_i x_i \ln x_i$.

What the above says, intuitively, is that if we skew the geometry of the positive orthant $\mathbb{R}^n_{>0}$ using the entropy function $H(x)$, then the curve which the gradient descent algorithm would follow to minimize the linear objective $c^\top x$, coincides with the Physarum dynamics. Note that this change of geometry stretches the space close to its boundary so that the trajectories are naturally slowing down and thus never hit the boundary. More formally, let $\Omega = \{x \in \mathbb{R}^m : Ax = b, x > 0\}$ be the strictly feasible region of the considered linear program. We define the *local norm* at a point $x \in \Omega$ to be

$$\|u\|_x = \left(\sum_{i=1}^m \frac{u_i^2}{x_i}\right)^{1/2} = \sqrt{u^\top \nabla^2 H(x) u}.$$

This defines a Hessian (Riemannian) manifold structure on Ω. We then consider the gradient direction, defined as the direction which provides the most significant drop of the objective when followed from point x, i.e.,

$$\min_{u \in T_x\Omega, \|u\|_x = 1} \langle c, u \rangle ,$$

where $T_x\Omega = \{u \in \mathbb{R}^m : Au = 0\}$ is the tangent space at x. This can be reduced to finding a minimizer of a quadratic function (using duality) or solved directly using Lagrange multipliers to yield that the optimal direction is $\frac{u^\star}{\|u^\star\|_x}$, where

$$u^\star = XA^\top(AXA^\top)^{-1}AXc - Xc.$$

A dynamical system defined by such a vector field turns out to coincide (up to a rescaling) with the Physarum dynamics, when restricted to the feasible region, which provides an interpretation of Physarum as a gradient flow on a Riemannian manifold.

We complement this result by yet another interpretation of the Physarum dynamics from the viewpoint of optimization. We show that the trajectories of Physarum are in fact paths of optimizers to a parametrized family of convex programs, whose objective is a linear cost function regularized by an entropy barrier function; see Theorem 4.7 for a formal statement. Thus, roughly, if $x : [0, \infty) \to \mathbb{R}^m_{>0}$ is a trajectory, then $x(t)$ is a solution to a convex program of the following form

$$\begin{aligned} \text{min.} \quad & c^\top x + \frac{1}{t}\sum_{i=1}^{m} x_i \ln x_i \\ \text{s.t.} \quad & Ax = b, \qquad x \geqslant 0. \end{aligned}$$

Note that the contribution of the entropy term $\sum_{i=1}^m x_i \ln x_i$ vanishes as $t \to \infty$. This draws an interesting connection between Physarum and Interior Point Methods for linear programming [Wri97], however the latter typically use self-concordant barrier functions for regularization and the entropy barrier does not fall into this class. It is an interesting problem whether the entropy barrier can be used to design fast algorithms for linear programming, as the standard lower bounds do not apply to it.

2.3 Undirected Physarum Dynamics

In Chapter 5 we study the undirected variant of the Physarum dynamics and more specifically, its most general form: the dynamics for the basis pursuit problem. For, clarity we focus on the unweighted variant of the problem. The existence of solutions for both the discrete and continuous dynamics can be easily established (see Fact 5.1), hence we focus on convergence and an optimization viewpoint. Below we state the convergence result for the undirected Physarum dynamics which answers the convergence question for this model. Note that the below theorem talks about the discrete dynamics, but in fact we can deduce convergence of the continuous-time Physarum dynamics by just taking the step size $h \to 0$ (see Lemma 5.4). We refer to Theorem 5.2 for a formal statement.

Theorem 2.3 (*Convergence – Informal*)

Let (A, b) be any instance to the basis pursuit problem and let $x^\star$ be any optimal solution. Let $x^{(0)}, x^{(1)}, \ldots$ be a sequence of points obtained by running the discrete undirected Physarum dynamics on this input for step size $h = \Theta\left(\frac{\varepsilon}{m\mathcal{D}}\right)$. Then, after $k = O\left(\frac{m\mathcal{D}\cdot\ln(m\|x^\star\|_1)}{\varepsilon^3}\right)$ steps, the following holds:

$$\|x^\star\|_1 \leqslant \|x^{(k)}\|_1 \leqslant (1+\varepsilon)\|x^\star\|_1.$$

In the above statement $\mathcal{D}$ represents the maximum subdeterminant of the matrix A. In the case when A is totally unimodular, $\mathcal{D}$ equals 1, hence, in particular, for flow problems this algorithm runs in polynomial time. This implies convergence in value for all the known variants of the undirected dynamics. It is an open question to decide whether this dynamics (even for the shortest path case) converges pointwise. The proof of Theorem 2.3 is based on tracking a potential function of the form $\Phi(k) := \log\|x^{(k)}\|_1 + E(x^\star, x^{(k)})$, where E is an entropy-like distance function (similar to KL-divergence). We prove that $\Phi(k)$ drops in every step by a certain amount which then allows us to deduce convergence. Interestingly, the form of $\Phi(k)$ bears resemblance to the free energy function studied in physics, suggesting that the general Physarum dynamics might have an intepretation in terms of a physical system.

To provide an optimization viewpoint on the undirected Physarum dynamics, we define the IRLS (Iteratively Reweighted Least Squares) algorithm. It can be seen as a discrete dynamical system aimed to solve the basis pursuit problem by solving a sequence of weighted ℓ_2 minimization problems. Formally, given a starting point $y^{(0)}$ with $Ay^{(0)} = b$, the IRLS algorithm update is as follows

$$y^{(k+1)} = \operatorname{argmin} \left\{ \sum_{i=1}^{m} \frac{x_i^2}{|y_i^{(k)}|} : x \in \mathbb{R}^m, Ax = b \right\}.$$

Even though no complete theoretical analysis is known for the IRLS algorithm (in fact this algorithm might fail for certain specially crafted instances), empirical evidence shows that it converges very quickly to an optimal solution of the basis pursuit problem. Our next result is the following connection between the IRLS algorithm and Physarum dynamics. We refer to Theorem 5.1 for a formal statement.

Theorem 2.4 (*Optimization Viewpoint – Informal*)

The Euler discretization of the undirected Physarum dynamics with step size $h = 1$ is equivalent to the IRLS algorithm.

Thus, essentially, the above states that the undirected Physarum dynamics is a damped version of the IRLS algorithm. Interestingly, while convergence of the IRLS algorithm is still an open problem, our convergence result in Theorem 2.3 implies that a slightly less aggressive variant of it (with smaller steps size) provably converges to an optimal solution.

Chapter 3

Polynomial-Based Algorithms for Discrete Optimization and Sampling Problems

In this chapter we start by giving a brief introduction to discrete optimization and sampling problems. Next, we outline the polynomial-based approach to these problems which is developed in Chapters 6, 7, 8, 9, 10 of this thesis. Finally we discuss contributions of these chapters in more detail.

3.1 Introduction to Polynomial-Based Methods for Counting and Sampling

We study optimization problems over discrete structures. The most common examples of such, which we often refer to in this chapter, are finding a maximum weight spanning tree in an undirected graph or finding a maximum weight perfect matching in a graph. Both these as well as numerous other problems in combinatorial optimization can be stated in the following general framework. Below, the family of all subsets of a set X we denote by 2^X.

Discrete Optimization Problem

Given a universe $[m] = \{1, 2, \dots, m\}$, a function $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ assigning non-negative values to subsets of the universe and a family $\mathcal{B} \subseteq 2^{[m]}$ of subsets of the universe, the **Optimization Problem** is to compute

$$\max_{S \in \mathcal{B}} \mu(S).$$

Such a function μ we often call a *measure* over subsets of $[m]$. For example, if $G = (V, E)$ is a weighted undirected graph with m edges and weights $w \in \mathbb{R}^m$, then by defining a measure

$$\mu^w(S) := \prod_{i \in S} e^{w_i} \qquad \text{for every } S \in 2^{[m]}, \tag{3.1}$$

and choosing $\mathcal{B} \subseteq 2^{[m]}$ to be the set of all spanning trees in G or the set of all perfect matchings in G, we recover the maximum weight spanning tree or the maximum weight perfect matching problem respectively. Note that in the example we defined the measure of a set $S \subseteq [m]$ to be as in (3.1) and not as $\sum_{i \in S} w_i$ to ensure nonnegativity of $\mu(S)$, which will be important later on. Importantly in these examples, even though both $\mathcal{B}$ and μ are exponential-size objects, they are still provided succinctly as input – as a graph and a set of edges. The form in which input is provided is an important issue for this kind of problems, as we aim for algorithms with polynomial running time with respect to m. Later on, we will introduce a much more general way of succinctly representing measures μ and families $\mathcal{B}$.

Sampling. In many practical applications instead of finding one – best solution, we are rather interested in the ability to find multiple good solutions; one way to achieve it, is to sample a solution with probability proportional to its measure. Note that such a sampling procedure can be used as a robust alternative for optimization, especially when the measure μ is noisy or has some built-in bias. Primitives based on sampling are often preferred over optimization in data summarization where one is interested in finding a small yet informative subset of a dataset. Before we discuss some examples, let us first provide a formal definition of sampling. At the same time we also introduce counting – a task inherently related to sampling, as discussed below.

Discrete Sampling and Counting Problems

Given a universe $[m] = \{1, 2, \ldots, m\}$, a function $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ assigning nonnegative values to subsets of the universe and a family $\mathcal{B} \subseteq 2^{[m]}$ of subsets of the universe, we define

(1) **Sampling Problem**: output a set $S \in \mathcal{B}$ according to the distribution $\mu_{\mathcal{B}}$, where

$$\mu_{\mathcal{B}}(S) = \begin{cases} \frac{\mu(S)}{\sum_{T \in \mathcal{B}} \mu(T)} & \text{for } S \in \mathcal{B}, \\ 0 & \text{for } S \notin \mathcal{B} \end{cases},$$

(2) **Counting Problem**: compute the sum

$$\sum_{S \in \mathcal{B}} \mu(S).$$

Thus, the counting problem is simply to compute the normalizing factor for the distribution $\mu_{\mathcal{B}}$. It has been shown that for many instances (satisfying a certain self-reducibility property), being able to solve the counting problem efficiently implies an efficient sampling method, conversely – fast sampling schemes can be often used for approximate counting [JVV86]. For this reason, to derive fast sampling algorithms it is enough to do the same for counting, which is often a simpler (at least conceptually) problem to study.

To illustrate the sampling and counting problems consider again the case where $[m]$ is the set of edges in a graph G and the measure is μ^w as defined in (3.1). Then, if $\mathcal{B}$ is the set of all spanning trees in G the sampling problem is to sample a spanning tree T with probability proportional to $\mu^w(T)$ and the counting problem is to compute $\sum_{T\in\mathcal{B}} e^{w(T)}$, where $w(T) := \sum_{i\in S} w_i$. Note that for $w \equiv 0$ this corresponds to uniform sampling and counting all spanning trees respectively. Note also that by slightly modifying the weight function to $\widetilde{w}(\alpha) = \alpha \cdot w$ for some large positive number $\alpha \in \mathbb{R}_{>0}$, the sampling problem corresponding to $\mu^{\widetilde{w}(\alpha)}$ becomes essentially equivalent to optimization, as the probability distribution $\mu_{\mathcal{B}}^{\widetilde{w}(\alpha)}$ is concentrated on sets with large weights and tends to a uniform distribution over maximum-weight subsets as $\alpha \to \infty$. This observation shows that sampling is actually more general than optimization in many important regimes.

Approaches for sampling and counting. There has been a lot of research on sampling and counting for various classes of discrete problems. One of the well known approaches is the Markov Chain Monte Carlo (MCMC) method. It is based on constructing a Markov chain whose stationary distributions corresponds to $\mu_{\mathcal{B}}$ and then running a long enough random walk on such a chain to sample from this distribution. This approach, although very succesful for several important problems, such as sampling perfect matchings and computing permanents [Sin93, JSV04], requires a lot of structure on the family $\mathcal{B}$ and the measure μ. Other approaches for sampling and counting include analytic combinatorics [FS09], methods based on belief propagation [YFW05] or on the idea of correlation decay [Wei06, BG06, BGK$^+$07, SST14]. All these methods provide strong results when specialized to certain settings, however their applicability is rather limited and no general framework can be derived for these methods.

In this thesis we propose and give several examples of applying a much more versatile methodology based on polynomials. It leads to a very general framework, able to capture a variety of important problems; and reduces efficient counting to proving certain analytic properties of families of polynomials.

Encoding structure in a polynomial. Consider again the example of counting for spanning trees on weighted graphs. We are given a measure $\mu(S) = e^{w(S)}$ for $S \subseteq [m]$ and $\mathcal{B} \subseteq [m]$ is the family of spanning trees in a graph. We can encode the information about the measure μ in a polynomial as follows.

Encoding Measures as Polynomials

Given a measure $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$, its **generating polynomial** $g_\mu \in \mathbb{R}[x_1, x_2, \dots, x_m]$ is defined as

$$g_\mu(x_1, x_2, \dots, x_m) := \sum_{S\subseteq[m]} \mu(S) x^S,$$

where $x^S := \prod_{i\in S} x_i$ for any set $S \subseteq [m]$.

Note in particular that g_μ contains the full information about μ, i.e., this representation is lossless. A question arises: but what do we gain by using such a representation? It turns out that for many important examples this polynomial interpretation is **efficiently computable** and moreover allows to **extract** important statistics of the measure μ, such as marginal probabilities. For instance, for the measure defined above, the polynomial, even though being a sum of exponentially many terms, can be succintly written as

$$g_\mu(x) = \prod_{i=1}^{m}(1 + e^{w_i} x_i).$$

Then, in particular, the value $g_\mu(0, 1, 1, \ldots, 1)$ gives the total weight of all subsets of edges not containing edge labeled 1.

Example – spanning trees. Let us now go back to the example of spanning trees – we are given a polynomial g_μ and would like to recover the sum of all its coefficients corresponding to monomials (sets) $S \in \mathcal{B}$. To the rescue comes Kirchhoff's matrix tree theorem. To state it, for every edge $i \in \{1, 2, \ldots, m\}$ that connects vertices $v, w \in V = \{1, 2, \ldots, n\}$ define a vector $b_i := e_v - e_w$, where $e_v, e_w \in \mathbb{R}^n$ are the standard basis vectors. Then the following formula holds

$$\det\left(\sum_{i\in[m]} x_i e^{w_i} b_i b_i^\top\right) = \sum_{S\in\mathcal{B}} \mu(S) x^S.$$

In other words, the determinant above captures exactly the polynomial being the restriction of g_μ to sets in $\mathcal{B}$! (For brevity, let us denote this polynomial by $p(x)$.) Now, given access to $p(x)$ we can solve the counting problem by simply evaluating $p(1, 1, \ldots, 1)$. This boils down to computing one determinant and thus can be executed efficiently, in $O(n^3)$ time. Moreover, $\frac{p(0,1,\ldots,1)}{p(1,1,\ldots,1)}$ is the probability that the edge labeled 1 does not belong to a random spanning tree (with respect to $\mu_\mathcal{B}$). This observation can be used to design a polynomial time sampling algorithm for $\mu_\mathcal{B}$ using the so called self-reducibility property of spanning trees, see [JVV86]; it says, roughly, that by including or not including the edge 1 in a spanning tree and conditioning on this event we obtain a sampling subproblem of the same form (on a reduced graph).

Example – permanent. As a second example consider the counting problem for perfect matchings in weighted, bipartite graphs. This problem is well known to be equivalent to computing permanents of non-negative matrices, which we focus on now. Given a matrix $A \in \mathbb{R}_{\geqslant 0}^{n\times n}$ we would like to compute

$$\mathrm{Per}(A) := \sum_{\sigma\in S_n} \prod_{i=1}^{n} A_{i,\sigma(i)},$$

where S_n is the set of all permutations of $\{1, 2, \ldots, n\}$. We will use the same strategy as in the spanning tree example – first encode the measure μ using a polynomial. This time the universe for the counting problem is the set of n^2 cells in the matrix, i.e., $U := [n] \times [n]$ and thus the polynomials we consider, have variables $\{x_{i,j}\}_{i,j\in[n]}$. The family $\mathcal{B}$ we consider, is the family of all subsets $S \subseteq U$ of cardinality n such that no two cells in S share the same row or column. The measure of a set S is defined $\mu(S) := \prod_{(i,j)\in S} A_{i,j}$. Thus we have captured the problem of computing permanents as a counting problem in our general framework.

Let us now try to design a polynomial which encodes the measure μ as in the previous example. To this end consider

$$p(x) := \prod_{i=1}^{n}\sum_{j=1}^{n} x_{i,j} A_{i,j}. \tag{3.2}$$

The above does not quite encode the measure μ over all subsets of U, but still, $p(x)$ is of the form

$$p(x) = \sum_{S \in \mathcal{B}_r} \mu(S) x^S,$$

where $\mathcal{B}_r \supseteq \mathcal{B}$ is the set of all subsets of U of size n so that no two cells from S share a common row. Let us define an analogous family $\mathcal{B}_c \subseteq 2^U$ for columns instead of rows. Now we observe that computing the permanent boils down simply to recovering the sum of all coefficients of sets in $\mathcal{B}_c$ in $p(x)$, as $\mathcal{B} = \mathcal{B}_c \cap \mathcal{B}_r$. Thus we end up asking the same question as in the example for spanning trees:

Question

Given a **polynomial** p that is **easy to evaluate** and a certain family of its monomials $\mathcal{B}$, how to **recover** the sum of **coefficients** in p corresponding to monomials in $\mathcal{B}$?

This question turns out to be rather tricky to answer as the permanent problem is already $\#P$-complete and hence unlikely to have polynomial time algorithms [Val79]. Thus the right question to ask is: can we approximate this sum of coefficients efficiently? We show that this task can be very generally captured by a **continuous relaxation**.

For the case of permanents, such a relaxation was provided by Gurvits [Gur06], who studied the following optimization problem:

$$\mathrm{Cap}(q) = \inf_{y>0} \frac{q(y)}{\prod_{i=1}^n y_i}, \qquad \text{where} \quad q(y_1, y_2, \ldots, y_n) := \prod_{i=1}^n \sum_{j=1}^n y_j A_{i,j}. \tag{3.3}$$

Note the similarity between the polynomial $p(x)$ (defined in (3.2)) and $q(y)$ above – yet they are not the same, as q has n variables and p has n^2 of them. Gurvits proved that $\mathrm{Cap}(q)$ satisfies

$$e^{-n}\mathrm{Cap}(q) \leqslant \mathrm{Per}(A) \leqslant \mathrm{Cap}(q), \tag{3.4}$$

and hence $\mathrm{Cap}(q)$ is a deterministic e^n-approximation to the permanent. Moreover, one can show that $\mathrm{Cap}(q)$ can be reformulated as a convex program and computed efficiently. The question which arises is: how general this machinery really is, can it be applied to more general polynomials or families $\mathcal{B}$?

Relaxations for counting problems. The general outline of our framework for solving counting problems is so far as follows: encode part of the problem structure in a polynomial and then reformulate it as recovering a sum of certain coefficients. The latter is dealt with using relaxations, which brings us to the following list of questions

Relaxations for Counting Problems

(1) How to **arrive** at such a relaxation?

(2) How to **reason** about the approximation factor?

(3) How to **efficiently compute** such relaxations?

Question (1) above is very general and thus does not have a universal answer; at a later point we will outline one possible strategy for that. Now, we focus on question (2) from the above list and specifically discuss it for the above relaxation (3.3) for the permanent.

Approximation ratios and real stability. We consider the capacity $\mathrm{Cap}(q)$ of the polynomial $q(y) = \prod_{i=1}^{n} \sum_{j=1}^{n} y_j A_{i,j}$ as defined in (3.3). Note that the permanent $\mathrm{Per}(A)$ appears as a coefficient of the monomial $y^{[n]} := \prod_{j=i}^{n} y_i$ in q. Thus a general question one might ask is: given any polynomial $r \in \mathbb{R}[y_1, y_2, \ldots, y_n]$, how well does $\mathrm{Cap}(r)$ approximate $r_{[n]}$ – the coefficient of the monomial $y^{[n]}$ in r? Clearly, for this question to make sense, we need to assume that the coefficients of r are non-negative, and perhaps, to make it more similar to the permanent, that all the monomials in r are of the same degree. Still, by inspecting simple examples, such as $r(y_1, y_2) = y_1^2 + y_2^2$, one can observe that $\mathrm{Cap}(r)$ is an infinitely bad approximation to this coefficient, as $r_{[2]} = 0$ and $\mathrm{Cap}(r) > 0$.

It turns out that the polynomial q has additional analytic properties which allow to prove such a bound! In fact it can be proved to be real stable – and this turns out to guarantee a good approximation ratio for such a relaxation. Real stability is a geometric condition on the location of zeros of a polynomial, which generalizes real-rootedness.

Real Stable Polynomial

A polynomial $r \in \mathbb{C}[x_1, \ldots, x_n]$ is called **real stable** if all its coefficients are real numbers and the following condition holds

$$\forall_{z \in \mathbb{C}^n} \quad (\Im(z_i) > 0 \text{ for all } i = 1, 2, \ldots, n) \quad \Rightarrow \quad r(z) \neq 0.$$

Real stable polynomials have recently found numerous applications in mathematics [BB09, MSS15] and computer science [Gur06, MSS13, AOG15, NS16] (see also surveys [Wag11, Pem11, Vis13]). From a combinatorial viewpoint, they can be seen as a special class of log-concave functions on matroid-like structures [Brä07, Wag11], yet a precise combinatorial characterization is not known up to date. The strength of real stable polynomials stems in part from their numerous closure properties:

Closure Properties of Real Stable Polynomials

Let $p, r \in \mathbb{R}[x_1, x_2, \ldots, x_n]$ be real stable polynomials, then

(1) $p \cdot r$ is real stable,

(2) $\frac{\partial}{\partial x_1} p(x)$ is real stable,

(3) $p(a, x_2, x_3, \ldots, x_n)$ is real stable for any $a \in \mathbb{R}$.

In particular, using property (1) above one can conclude that the polynomial q is indeed real stable as every factor in the product defining it is easily seen to be real stable. Now, to prove the non-trivial direction of Gurvits inequality (3.4), i.e., that $e^{-n}\mathrm{Cap}(q) \leqslant q_{[n]}$ (the other follows directly from the definition of capacity), we will use properties (2) and (3). To this end we first note that

$$q_{[n]} = \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \cdots \frac{\partial}{\partial x_n} q(x).$$

And thus by using induction and properties (2) and (3), the problem of proving the inequality $e^{-n}\mathrm{Cap}(q) \leqslant q_{[n]}$ reduces to a univariate inequality for real-rooted polynomials! In fact, since in every step of this inductive argument we roughly loose a factor e, the total approximation is e^n. Interestingly, this bound is actually tight for permanents, hence none of the steps of the above proof has incured a significant loss!

Thus, consequently, additional analytic properties of polynomials, such as real stability, can allow us to prove approximation bounds for certain continuous relaxations, while simultaneously not restricting generality by too much.

Entropy interpretation and new relaxations. We go back to the question on how to design continuous relaxations for counting problems. For this, consider again the permanent example and the polynomial $p(x) = \sum_{S\in\mathcal{B}_r} \mu(S)x^S$. Recall that the goal was to recover the sum of coefficients of all monomials in the set $\mathcal{B}_c$. One can prove that the following expression is equal to $\mathrm{Cap}(q)$ and thus approximates $\mathrm{Per}(A)$ up to a factor of e^n

$$\sup_{\theta\in P(\mathcal{B}_c)} \inf_{x>0} \frac{p(x)}{x^\theta}, \tag{3.5}$$

where $x^\theta := \prod_{i,j} x_{i,j}^{\theta_{i,j}}$ and $P(\mathcal{B}_c)$ is the convex hull of the set $\mathcal{B}_c$, i.e., formally $P(\mathcal{B}_c) = \mathrm{conv}\{1_S : S \in \mathcal{B}_c\} \subseteq [0,1]^U$ (here 1_S is the indicator vector of a set $S \subseteq U$ in the n^2-dimensional space R^U).

We present an interpretation of (3.5) which gives a systematic way to arrive at such a relaxation. The inner optimization problem turns out to have a dual form being a variant of an entropy-maximization problem. More formally, recall that the polynomial $p(x)$ represents the restriction of the measure μ to $\mathcal{B}_r$. Consider any $\theta \in P(\mathcal{B}_c)$, the inner optimization problem (or more precisely its logarithm) $\inf_{x>0} \frac{p(x)}{x^\theta}$ has then the following dual form

$$\begin{aligned} \max \quad & \sum_{S\in\mathcal{B}_r} q_S \log \frac{\mu(S)}{q_S}, \\ \text{s.t.} \quad & \sum_{S\in\mathcal{B}_r} q_S \cdot 1_S = \theta, \\ & \sum_{S\in\mathcal{B}_r} q_\alpha = 1, \\ & q \geqslant 0. \end{aligned} \tag{3.6}$$

In the above, we look for a probability distribution over $\mathcal{B}_r$ such that the marginal probabilities are specified by θ and the objective is to maximize the relative entropy with respect to the measure induced by the polynomial p. The outer optimization problem is to maximize the entropy over all possible marginal vectors θ which belong to the convex hull of $\mathcal{B}_c$. The rationale behind maximization over $\theta \in P(\mathcal{B}_c)$ is that we are interested in coefficients of the polynomial for monomials in $\mathcal{B}_c$ only and hence only the restriction of the distribution given by p to the $\mathcal{B}_c$ matters – in the continuous domain we translate it to optimizing over its convex hull. This idea can be significantly extended as we discuss in the next section.

3.2 Estimation Algorithms for Counting and Optimization

In this section we outline the results of Chapters 6 and 9 of the thesis. Suppose we are given a multiaffine polynomial p with non-negative coefficients (representing a certain measure μ), i.e., of the form $p(x) = \sum_{S\subseteq[m]} p_S x^S$ with $p_S \geqslant 0$ for all $S \subseteq [m]$. The key problem mentioned in the previous section was to recover the sum of coefficients over all sets in $\mathcal{B}$ or, in other words, to compute $\sum_{S\in\mathcal{B}} p_S$. Inspired by the entropy interpretation in (3.6) we define the following notion of $\mathcal{B}$-capacity of the polynomial p

$$\mathrm{Cap}_{\mathcal{B}}(p) := \sup_{\theta\in P(\mathcal{B})} \inf_{x>0} \frac{p(x)}{\prod_{i=1}^m x_i^{\theta_i}}. \tag{3.7}$$

Recall that $P(\mathcal{B})$ is defined as the convex hull of indicator vectors of sets in $\mathcal{B}$, i.e, $P(\mathcal{B}) := \mathrm{conv}\{1_S : S \in \mathcal{B}\}$. Even though, in general, (as in the case of Gurvits' capacity (3.3)) the above quantity can be an arbitrarily bad approximation to the value of the counting problem, we prove that it works well when p is real stable and $\mathcal{B}$ comes from a matroid. For background on matroids we refer the reader to Section 6.1.2. Here it is enough to know that matroids are an axiomatic formalization of the notion of independence and generalize spanning trees in graphs and linear independence of vectors. For a formal statement of the following theorem we refer to Theorem 6.1.

Theorem 3.1 (*Integrality Gap – Informal*)

Suppose that $p \in \mathbb{R}[x_1, \ldots, x_m]$ is a real stable polynomial and $\mathcal{B}$ is a family of bases of a matroid. Then $\mathrm{Cap}_{\mathcal{B}}(p)$ approximates the value of $\sum_{S\in\mathcal{B}} p_S$ up to a factor of $e^{O(m)}$.

In Chapter 6, inspired by the notion of $\mathcal{B}$-capacity, we also define an analogous relaxation for the related optimization problem, i.e., computing $\max_{S\in\mathcal{B}} p_S$. It is defined in a similar way as $\mathrm{Cap}_{\mathcal{B}}(p)$:

$$\mathrm{opt}_{\mathcal{B}}(p) := \sup_{z\in P(\mathcal{B})} \sup_{\theta\in P(\mathcal{B})} \inf_{x>0} \frac{p(x_1z_1, x_2z_2, \ldots, x_mz_m)}{\prod_{i=1}^m x_i^{\theta_i}} \tag{3.8}$$

and, as we prove in Theorem 6.1, $\mathrm{opt}_{\mathcal{B}}(p)$ achieves similar approximation guarantees for estimating $\max_{S\in\mathcal{B}} p_S$.

A question which arises naturally is whether $\mathrm{Cap}_{\mathcal{B}}(p)$ and $\mathrm{opt}_{\mathcal{B}}(p)$ can be efficiently computed given appropriate access to p and $\mathcal{B}$. This follows from our results on computability of max-entropy distributions in Chapter 7, whose content is outlined in the subsequent section.

The results obtained in Chapter 9 are of the same flavor as these above yet apply to a slightly different setting. There we study the model of factor graphs and are interested in the corresponding counting problem – of computing partition functions. We prove that a well known heuristic – the Bethe approximation [Bet35, SWW07] for approximating the partition function can be stated as a polynomial relaxation similar to (3.7) and (3.8). Using techniques based on real stable polynomials we are able to arrive at a bound in this setting, see Theorem 9.2.

3.3 Computability of Maximum Entropy Distributions

Consider a finite subset $\mathcal{F} \subseteq \mathbb{Z}^m$ of the integer lattice. Given a vector θ, the following max-entropy convex program solves for a probability distribution over $\mathcal{F}$ that has maximum entropy with expectation θ. In the language of information theory, this is the task of finding the I-projection of the uniform distribution over $\mathcal{F}$ onto the set of all distributions with expectation

θ.

$$\begin{aligned} \max \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \log \frac{1}{q_\alpha}, \\ \text{s.t.} \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \cdot \alpha = \theta, \\ & \sum_{\alpha \in \mathcal{F}} q_\alpha = 1, \\ & q \geqslant 0. \end{aligned} \tag{3.9}$$

Max-entropy distributions arise and have found numerous applications in information theory [Jay57a, Jay82], machine learning [PPL97, Nig99], economics [AFHS97, Vin06], physics [MS06] and statistics [Soo00, SW87]. In Theoretical Computer Science, max-entropy distributions over spanning trees in a graph have been used to derive approximation algorithms for the TSP [AGM$^+$10, OSS11] and max-min fair allocation problem [AS10].

Note that the interesting regime is when the family $\mathcal{F}$ has size exponential in the dimension m, in which case the program (3.9) cannot be even stored in polynomial space. In this chapter we consider the question of when can one solve such programs in polynomial time and obtain a succinct (polynomial space) representation of the maximum entropy distribution $q^\star$. Our main theorem follows. (For a formal statement we refer to Theorem 7.2.)

Theorem 3.2 (*Computability of Maximum Entropy Distributions – Informal*)

Suppose that $\mathcal{F} \subseteq \mathbb{Z}^m$ has polymially bounded diameter and $\mathrm{conv}(\mathcal{F})$ has polynomial unary facet complexity. There is an algorithm that given a counting oracle for $\mathcal{F}$ and and $\varepsilon > 0$ outputs a succinct (of size polynomial in m and $\log \frac{1}{\varepsilon}$) description of a distribution q such that $\|q - q^\star\|_1 < \varepsilon$ in time polynomial in m and $\log \frac{1}{\varepsilon}$.

A few remarks are in order. A counting oracle is a primitive for solving a certain counting problem on $\mathcal{F}$. The unary facet complexity of an integral polytope P is, roughly, the smallest integer M such that P can be described by linear inequalities with integer entries of magnitude at most M. Most interesting families $\mathcal{F}$ such as matroids or perfect matchings in graphs, satisfy the low unary facet complexity assumption.

Interestingly, by intepreting several recent results in the language of maximum entropy distributions we can obtain various corollaries of Theorem 3.2, including the efficient computability of the relaxations $\mathrm{Cap}_{\mathcal{B}}(p)$ and $\mathrm{opt}_{\mathcal{B}}(p)$ from Chapter 6.

Corollary 3.1 (*Computability of Polynomial Relaxations – Informal*)

There is a polynomial time algorithm which given evaluation oracle access to real stable polynomial p and a separation oracle for $P(\mathcal{B})$, outputs a $(1+\varepsilon)$ multiplicative approximation to $\mathrm{Cap}_{\mathcal{B}}(p)$ and $\mathrm{opt}_{\mathcal{B}}(p)$ in time polynomial in m and $\log \frac{1}{\varepsilon}$.

3.4 Approximation Algorithms for Subdeterminant Maximization Problems

In Chapter 8 of the thesis we study an important special case of the discrete optimization problem which arises when the measure μ is determinantal. More precisely, given a PSD matrix

$L \in \mathbb{R}^{m\times m}$, we consider the measure

$$\mu(S) = \det(L_{S,S})$$

over subsets $S \subseteq [m]$, where $L_{S,S}$ is a submatrix of L corresponding to rows and columns in the set S. Such measures are also known as Determinantal Point Processes (DPPs) [Lyo02, HKPV05] and have been successfully applied to a number of problems, such as document summarization, sensor placement and recommendation systems [LB11, KSG08, ZKL$^+$10, ZCL03, YJ08].

The optimization result presented in Chapter 6 when applied to DPPs, implies that given a matrix $L \in \mathbb{R}^{m\times m}$ and a family of subsets $\mathcal{B} \subseteq 2^{[m]}$ being the set of bases of a matroid, one can estimate the value of

$$\max_{S\in\mathcal{B}} \det(L_{S,S})$$

in polynomial time up to a factor of $e^{O(m)}$. However, unfortunately the above yields only a numerical estimate of the value of the optimal solution, but does not output a set which attains this value. Morever, there seem to be significant obstacles when trying to obtain such a set, as standard rounding techniques lead to $\#P$-hard counting problems.

In Chapter 8 we present a completely different approach for tackling these kind of problems, which does not rely on real stability or any geometric properties of the considered polynomials. The main idea is to first reformulate the discrete optimization problem as an equivalent continuous optimization problem of the form

$$\max_{x\in[0,1]^m} |f(x)|$$

with f being a multiaffine polynomial (this optimization problem is then highly non-convex). Afterwards, the maximum value of f over the hypercube is approximated simply by a value at a uniformly random point in the hypercube. Finally, using the fact that f is a multiaffine polynomial, we can efficiently, and without any loss, round the random point to a vertex with a value at least as large. To show that such a procedure yields a good approximation guarantee with high probability, we prove a general anti-concentration inequality for the type of functions we obtain from maximizing subdeterminants. Using this technique we arrive at approximation algorithms for subdeterminant maximization over $\mathcal{B}$ for partition and regular matroids. For simplicity the bound for partition matroids stated below is only for the case when the size of every part is a constant. (We refer to Theorems 8.2 and 8.3 for a formal statement.)

Theorem 3.3 (*Subdeterminant Maximization under Matroid Constraints – Informal*)

There is a polynomial time randomized algorithm such that given a PSD matrix $L \in \mathbb{R}^{m\times m}$ and a family of bases $\mathcal{B} \subseteq 2^{[m]}$ of a partition or regular matroid with high probability, outputs a set $T \in \mathcal{B}$ such that

$$\det(L_{T,T}) \geqslant e^{-O(m)} \cdot \max_{S\in\mathcal{B}} \det(L_{S,S}).$$

The approximation guarantees obtained by the above theorem match the ones obtained in Chapter 6 up to a constant in the exponent.

3.5 Exact Sampling and Counting for DPPs

In Chapter 10 we revisit our first example: of spanning trees, for which the sampling and counting problem could be solved exactly in polynomial time. We prove that this is also the case in a much more general setting. Consider any non-negative measure $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$. Given a cost vector $c \in \mathbb{Z}^m$ and a budget $C \in \mathbb{Z}$ we define the family of subsets which satisfy a budget constraint with respect to c

$$\mathcal{B} := \left(S \subseteq [m] : \sum_{i \in S} c_i \leqslant C \right). \tag{3.10}$$

Constraints of this type are quite common in various machine learning applications. Moreover, by combining several such constraints one can obtain quite non-trivial families of sets. The goal is to sample from the probability distribution $\mu_{\mathcal{B}}$ as defined in Section 3.1. We work under the assumption that an oracle access to the generating polynomial of μ, $g_\mu(x) := \sum_{S \subseteq [m]} \mu(S) x^S$ is given. This setting clearly captures the important example of (budget) constrained DPPs as the generating polynomial

$$p(x) = \sum_{S \subseteq [m]} x^S \det(L_{S,S})$$

is efficiently computable. Indeed, $p(x) = \prod_{i=1}^m x_i \cdot \det(X^{-1} + L)$, where X is a diagonal matrix with $X_{i,i} = x_i$.

We prove that for budget constraints as above, the sampling problem can be solved in polynomial time whenever the cost vector is given in unary as input, more precisely

Theorem 3.4 (*Sampling under Budget Constraints – Informal*)

There is an algorithm which given oracle access to g_μ, a cost vector $c \in \mathbb{Z}^m$ and a budget $C \in \mathbb{Z}$, outputs a sample from the probability distribution $\mu_{\mathcal{B}}$ (with $\mathcal{B}$ as in (3.10)) in time polynomial with respect to m and $\|c\|_1$.

For a formal statement of the above theorem we refer to Theorem 10.1. The above also extends to DPP sampling under *partition* constraints, where the family $\mathcal{B}$ is of the form $\mathcal{B} = \{S \subseteq [m] : |S \cap P_j| = b_j \text{ for all } j = 1, 2, \ldots p\}$, where $[m] = P_1 \cup P_2 \cup \ldots \cup P_p$ is a partition of $[m]$ and $b_1, b_2, \ldots, b_p \in \mathbb{N}$. More precisely, it follows that sampling under partition constraints is possible in $m^{O(p)}$ time, where p is the number of parts in the partition. Conversely, we also prove that if the number of parts is large, it is unlikely that there is a polynomial time sampling algorithm, as the corresponding counting problem is hard.

Theorem 3.5 (*Hardness*)

The problem of counting for DPPs under partition constraints is $\#P$-hard.

Moreover we show that the above counting problem is in fact equivalent to computing so called mixed discriminants of tuples of PSD matrices, which is conjectured to not have an efficient approximation scheme [Gur05]. This provides a characterization of hardness for exact counting for partition constrained DPPs.

Chapter 4

The Directed Physarum Dynamics

In this chapter we study the directed Physarum dynamics in its most general form – for linear programming as introduced in Chapter 2. We prove that this dynamical systems always has global solutions which converge to an optimal solution of the underlying linear program exponentially fast. Similarly we prove convergence for its discretization. Finally we provide an optimization viewpoint for directed Physarum dynamics as a gradient flow on a Riemannian manifold as well as a viewpoint in which Physarum is a barrier method based on the entropy barrier.

4.1 Preliminaries and Statement of Results

4.1.1 Preliminaries

The Physarum Dynamics for Linear Programming. Consider a linear program in the standard form

$$\min\ c^\top x \quad \text{s.t. } Ax = b, \quad x \geqslant 0, \tag{4.1}$$

where $A \in \mathbb{Z}^{m\times n}$, $c \in \mathbb{Z}^n_{>0}$ and $b \in \mathbb{Z}^m$ and which has a feasible solution. To make the exposition clear we assume that A is full-rank, in other words: $\operatorname{rank}(A) = m$. We now recall the Physarum dynamics for linear programming.

Consider any vector $x \in \mathbb{R}^n$ with $x > 0$ and let W be the diagonal matrix with entries x_i/c_i. Let $L \overset{\text{def}}{=} AWA^\top$ and $p \in \mathbb{R}^n$ be the solution to $Lp = b$. Let $q \overset{\text{def}}{=} WA^\top p$. The Physarum dynamics for the linear program given by (A, b, c) then is

$$\dot{x} = q - x. \tag{4.2}$$

This can be also rewritten as

$$\dot{x} = W(A^\top L^{-1} b - c).$$

The dynamical system has an initial condition of the form $x(0) = s$ for some $s > 0$. *Note that it is not required that s is feasible*, i.e., $As = b$. Often, the exposition and proofs are simpler when s is also feasible. We often refer to the general case as Physarum dynamics with *infeasible start* and this special case as with *feasible start.*

The Physarum Dynamics for the Directed Transshipment Problem. In the directed transshipment problem, we are given a directed graph $G = (V, E)$, a demand vector $b \in \mathbb{Z}^V$ and a cost vector $c \in \mathbb{Z}^E_{>0}$. Assume that $\sum_v b_v = 0$ and let $B \in \mathbb{R}^{V\times E}$ be any signed incidence matrix of G. The goal is to find a flow vector $f \in \mathbb{R}^E$, $f \geqslant 0$ which satisfies the demand ($Bf = b$) and minimizes the cost of the flow: $\sum_{e\in E} c_e f_e$. To obtain the Physarum dynamics for this case one can simply state the problem as

$$\min\ c^\top f \quad \text{s.t.}\ \ Bf = b, \quad f \geqslant 0$$

and use the dynamics (4.2).

4.1.2 Statement of Results

The first problem we have at hand is whether the system (4.2) has a solution such that $x(t) > 0$ for all $t > 0$. Unfortunately there are no general theorems in dynamical systems to establish this and proving existence can be difficult. Indeed, this existence problem turns out to be non-trivial for Physarum dynamics and is our first result.

Theorem 4.1 (Existence of Solution)

For any initial condition $s \in \mathbb{R}^n_{>0}$, the Physarum dynamics $\dot{x} = q - x$ has a unique solution $x : [0, \infty) \to \mathbb{R}^n_{>0}$ with $x(0) = s$.

First we note that existence of a feasible solution is important: One can prove that whenever the problem (4.1) is infeasible, the solutions to (4.2) exist only for finite time intervals. Further, the problem of existence is difficult not because the dynamics is in continuous time; the problem remains non-trivial even if we consider a discretization. Now that we know that the solution exists, our next set of results establish that no matter where one starts the Physarum dynamics from, one converges to an optimal solution of the linear program.

Theorem 4.2 (Convergence to an Optimal Solution)

For any initial condition $s \in \mathbb{R}^n_{>0}$, consider the solution $x : [0, \infty) \to \mathbb{R}^n_{>0}$ to the Physarum dynamics with $x(0) = s$. Denote by $x^\star$ an optimal solution of the considered linear program. Then:

1. *$|c^\top x(t) - c^\top x^\star| = O(e^{-\alpha t})$ for some positive α, which only depends upon A, b, c and s,*
2. *the limit $x^\infty = \lim_{t\to\infty} x(t)$ exists, is a feasible point and $c^\top x^\infty = c^\top x^\star$.*

A few remarks are in order: (1) The theorem also gives an upper bound on the time to convergence and (2) it proves that limits of trajectories of the Physarum exist even when the optimal solution is *not unique*. Given the amount of technical effort required in proving this, one may wonder again if this has something to do with continuous time. The answer is (again) no: Starting with Karmarkar himself [Kar90], it has been observed that the good convergence properties of his algorithm for linear programming [Kar84] arise from the good geometric properties of the set of continuous trajectories which underlie his method [BL89].

Our next results concern the computational abilities of a discretization of the Physarum dynamics. We prove one general result which applies to all linear programs in the considered form and a result specialized to the special case for a variant of the minimum cost flow problem, for which the obtained bound is much tighter. We start by stating the former.

Theorem 4.3 (Convergence Time of Discrete Physarum Dynamics; see Theorem 4.12)

Consider the discretization of the Physarum dynamics, i.e., $x(k+1) = (1-h)x(k) + hq(k)$. Suppose we initialize the Physarum algorithm with $x(0) = s$, i.e., $As = b$ and $M^{-1} \leqslant s_i \leqslant M$ for every $i = 1, \ldots, n$ and some $M \geqslant 1$. Assume additionally that $c^\top s \leqslant M \cdot \mathrm{opt}$. Choose any $\varepsilon > 0$ and[1] *let $h = \frac{1}{6} \cdot \varepsilon \cdot C_s^{-2} \cdot \mathcal{D}^{-2}$. Then after $k \overset{\mathrm{def}}{=} O\left(\frac{\ln M}{\varepsilon^2 h^2}\right)$ steps $x(k)$ is a feasible solution with:* $\mathrm{opt} \leqslant c^\top x(k) \leqslant (1+\varepsilon) \cdot \mathrm{opt}$.

Thus, when the maximum cost is polynomial and the maximum sub-determinant of A is polynomially bounded, the discretization of Physarum dynamics is efficient.

We proceed with another result on discretization which is specialized to the directed transshipment problem. As discussed before, this is clearly a special case of linear programming and thus Theorem 4.3 applies, however we aim here for an algorithm with better running time, in particular where the dependence on ε is logarithmic instead of polynomial as in Theorem 4.3. In fact the result below implies that such an algorithm indeed exists, but one has to *precondition* the instance first to guarantee a provably good behavior of the Physarum dynamics. This preconditioning boils down to adding a number edges to the graph and specifying a convenient initial solution – it is introduced formally in Section 4.3.6. The theorem below talks about convergence of the Physarum dynamics for preconditioned instances only, even though the bounds are expressed with respect to the parameters (number of vertices, number of edges, cost vector, demands) of the preconditioned instance, they still hold for the instance we start with, as the number of vertices is the same in both.

Theorem 4.4 (Convergence Time of Discrete Physarum Dynamics for Min-Cost Flow)

Consider the discretization of Physarum dynamics $x(k+1) = (1-h)x(k) + hq(k)$ for the directed transshipment problem. Let (G, b, c) be any preconditioned instance with n vertices and m edges. If the step size is $h = O\left({}^{1}\!/_{n^5 C^2 b_P}\right)$ then for any $\varepsilon > 0$ after $k = O\left({}^{nCb_P \ln({}^{mb_P}\!/_{\varepsilon})}\!/_{h}\right)$ iterations we have $\|x(k) - f^\star\|_\infty < \varepsilon$ for an optimal solution $f^\star$.

Finally, we come to the conceptual question which has resisted an answer even for the shortest path Physarum dynamics: While each equation in (4.2) gives us a local update rule, can the Physarum dynamics be obtained from an optimization viewpoint? We answer this question affirmatively. For preliminaries on Riemannian manifolds see Section 4.3.1.

Theorem 4.5 (Optimization Viewpoint of Physarum Dynamics; see Theorem 4.6)

In case of feasible start one can interpret Physarum as a gradient flow on a Riemannian manifold. In other words it is of the form $\dot{x} = \nabla f(x)$, where $f(x) = c^\top x$ is the objective, but the gradient is computed with respect to a certain Riemannian metric on the feasible region. In the infeasible case, Physarum dynamics combines two forces; minimizing the objective and aiming for feasibility. Both can be interpreted similarly as descent directions on a manifold.

We remark that the above mentioned Riemannian metric is induced by the Hessian of the generalized entropy function $\sum_i c_i x_i \ln x_i c_i$ and provides a physical meaning to the dynamics. Moreover, using convex programming duality, we can show that the trajectories of Physarum with a feasible start are in fact paths of optimizers to a parametrized family of convex programs, which objective is a linear cost function regularized by an entropy barrier function; see Theorem 4.7 for a formal statement. Finally, note that the entropy barrier, unlike the log-barrier function is not a self-concordant barrier function; yet the trajectories converge as $e^{-\alpha t}$ as Theorem 4.2 indicates.

4.2 Technical Overview

We start by presenting an overview of our result which presents Physarum dynamics from an optimization viewpoint (Theorem 4.5). We assume 4.1 and 4.2 for the moment. Denote

$$P(x) \overset{\text{def}}{=} q - x = W(A^\top (AWA^\top)^{-1} b - c),$$

so that the Physarum dynamics becomes $\dot{x} = P(x)$. Knowing that (4.2) is in fact solving an optimization problem, one would expect that it implements some gradient-descent-type of procedure. This would mean that $P(x) = \nabla \Phi(x)$ for some function $\Phi(x)$. However, by a simple calculation, it turns out that $P(x)$ is not a gradient of any function. Similarly, $P(x)$ does not represent Newton's method nor steepest descent in any standard norm. To understand Physarum, we need to move from the Euclidean world to Riemannian manifolds.

We consider the manifold $\Omega = \mathbb{R}^n_{>0}$, to every point $x \in \Omega$ we assign a positive definite matrix

$$H(x) \overset{\text{def}}{=} \operatorname{Diag}\left({}^{c_1}/_{x_1}, \ldots, {}^{c_n}/_{x_n}\right),$$

which defines an inner product $\langle u, v\rangle_x = u^\top H(x) v$ on the tangent space at x (which is just $\mathbb{R}^n$ in this case), Ω becomes then a Riemannian manifold. We are mainly interested in the submanifold $\Omega^f = \{x \in \Omega : Ax = b\}$, which we assume to be nonempty for our discussion. It inherits the Riemannian structure from Ω. Note that the Nash Embedding Theorem guarantees that such a manifold always has an isometric embedding in an Euclidean space (possibly in much higher dimension). In our case, the embedding can be given by an explicit formula:

$$F(x) = 2(\sqrt{c_1 x_1}, \ldots, \sqrt{c_n x_n})^\top.$$

This gives as a geometric realization of the curvature imposed on Ω^f.

Let us now see that $P(x)$, when restricted to Ω^f, is just the gradient of the objective $f(x) = c^\top x$ when viewed in the local geometry. To show this, we need to argue that it satisfies

$$f(x+h) - f(x) - \langle h, P(x)\rangle_x = o(\|h\|_x)$$

over h from the tangent space at x, i.e., from $T_x(\Omega^f) = \{h : Ah = 0\}$. We calculate:

$$\begin{aligned} f(x+h) - f(x) - \langle h, P(x)\rangle_x &= c^\top h - h^\top H(x) P(x) \\ &= c^\top h - h^\top (A^\top L^{-1} b - c) \\ &= (Ah)^\top L^{-1} b = 0. \end{aligned}$$

If x lies outside of the feasible region a similar interpretation of $P(x)$ is possible. However in this case $P(x)$ decomposes into two parts: $P_f(x)$ which is the feasibility direction and $P_o(x)$ which aims for optimality. Both can be motivated as descent directions with respect to the above defined Riemannian structure on Ω.

The next result gives another interpretation of the Physarum dynamics in the feasible region. Namely, let us define $x(\mu)$ for $\mu > 0$ to be the minimizer of $c^\top x + \mu^{-1} f(x)$ over $x \in$

$\{x : Ax = b, x \geqslant 0\}$, where f is the following entropy-like function

$$f(x) \stackrel{\text{def}}{=} \sum_{i=1}^{n} c_i x_i \ln(c_i x_i).$$

It turns out that using elementary convex programming techniques such as duality and KKT conditions, we can prove that $x(\mu)$ satisfies the Physarum dynamics. This is also related to the fact that $\nabla^2 f(x) = H(x)$, which at the same time demonstrates that $(\Omega, \langle\cdot,\cdot\rangle_x)$ is a Hessian manifold; see Section 4.3.2.

We now give an overview of the proofs of the set of results that establish existence, convergence and complexity bounds on the Physarum dynamics (Theorems 4.1, 4.2, 4.3). We begin with Theorem 4.1. It asserts that global solutions $x : [0,\infty) \to \Omega$ of the Physarum dynamics indeed exist. This theorem is nontrivial because by standard existence-uniqueness theorems we can only obtain solutions defined on some tiny intervals around 0. To extend those solutions further we need to show that they cannot leave the domain Ω in finite time. More precisely, if $x : [0,T) \to \Omega$ is a solution, we need to show that the limit point $x(T) = \lim_{t\to T} x(t)$ exists and belongs to Ω. The main concern is that potentially $x(T)$ may end up on the boundary of Ω, i.e., have a zero at some coordinate. Of course, we expect some entries of $x(t)$ to vanish with $t \to \infty$, what we need to show is that this happens in a controlled manner.

Let us take a look at the dynamics:

$$\dot{x_i} = q_i - x_i = \frac{x_i}{c_i}(a_i^\top p - c_i),$$

where a_i is the i-th column of A. If we could show that $a_i^\top p$ is uniformly bounded then this would imply that $\dot{x_i} \geqslant -\alpha x_i$ for some constant α, and, by the Gronwall lemma (see, e.g., [Per01]) $x_i(t) \geqslant e^{-\alpha t} x_i(0) > 0$. Indeed Lemma 4.1, which shows that $|a_i^\top p|$ is uniformly bounded over all $x > 0$ such that $Ax = b$, rescues us. At first glance Lemma 4.1 may seem a bit technical, but in a sense it captures exactly the property of the Riemannian space, which is responsible for the well behavior of trajectories. This gives a proof of existence in the special case when the initial point $x(0)$ satisfies $Ax(0) = b$. (Assuming $Ax(0) = b$ one can show that $Ax(t) = b$ for every t, since $\dot{x}$ is tangent to this affine subspace.)[2] Unfortunately, the case of infeasible start turns out to be harder. An analogue of Lemma 4.1 does not hold if we let x vary over Ω.

We proceed by analyzing a certain barrier function $\sum_{i=1}^{n} c_i y_i \ln x_i$, with y being any feasible solution to (4.1). By analyzing its derivative, we are able to prove that on every finite interval $[0,T)$ this function is uniformly lower-bounded, which essentially means that on every finite interval there exists some $\delta > 0$ such that $\delta \cdot y \leqslant x_i(t)$. This implies further that $x_i(t)$ is lower-bounded by a positive constant, on every finite interval, but only for $i \in \operatorname{supp}(y)$. Of course we may repeat this argument multiple times with different y's. This, however, is not enough: it is a common case for linear programs that there is an index $i \in \{1,2,\ldots,n\}$ such that for every feasible solution y, $y_i = 0$. Therefore we need to use a different argument for the remaining indices.

To complete the argument we combine Lemma 4.1 with the above mentioned fact that on every finite interval, $x_i(t) \geqslant \delta y$ for some $\delta > 0$ and some feasible solution y. Applying Lemma 4.1 to y, we can essentially extract a uniform upper bound on $|a_i^\top p(t)|$ over a fixed finite interval. This, again by the Gronwall lemma, allows us to conclude positivity. In the process of the proof

[2]For the case of feasible start, one can also give another proof of existence via Theorem 4.7.

we need to argue that $\|q(t)\|$ and $\|x(t)\|$ remain bounded uniformly over all t. For this, Lemma 4.1 is again a starting point.

After establishing existence we study the limiting behavior of the solutions to (4.2). Our results are summarized in Theorem 4.2. The first result claims that

$$\left|c^\top x(t) - c^\top x^\star\right| \leqslant R \cdot e^{-\nu t},$$

where $R, \nu > 0$ are constants depending only on $A, b, c, x(0)$ (we provide explicit bounds in Theorem 4.10). Furthermore, one can show that $z(t) = x(t) - e^{-t}x(0)$ satisfies $Az(t) = b$. In other words, $x(t)$ approaches feasibility with $t \to \infty$.

To prove exponential convergence to the optimal value we start by showing that for every t there is some $y(t)$, feasible for (4.1), which is exponentially close to $x(t)$. Towards this, we cannot directly use $z(t)$, because it may happen that $z_i(t) < 0$ for some i. Instead, we use a well known fact that if a point x violates constraints defining a polytope up to an additive error ε then its distance to this polytope is at most $N \cdot \varepsilon$, where N depends only on the description size of the polytope (though exponentially).

To proceed, we need to exploit the structure of the feasible region $P = \{x : Ax = b, x \geqslant 0\}$. We write P as a Minkowski sum $H + C$, where H is a polytope and C is a polyhedral cone. We extract vertices V from H and spanning vectors R from C. We study the decomposition of $y(t)$ into

$$y(t) = \sum_{v \in V} \lambda_v(t) v + \sum_{r \in R} \mu_r(t) r,$$

with $\lambda(t), \mu(t) \geqslant 0$ and $\sum_{v \in V} \lambda_v(t) = 1$. By studying potential functions of the kind $\sum_{i=1}^n c_i v_i \ln x_i(t)$ for a fixed vertex $v \in V$ it can be shown that if v is non-optimal then $\lambda_v(t) \to 0$ exponentially fast. Similarly, for every $r \in R$ we have $\mu_r(t) \to 0$. This is then used to deduce the result.

Let us now discuss the second result related to the asymptotic behavior of solutions to (4.2). It says that if $x : [0, \infty) \to \Omega$ is such a solution then $x(t)$ has a limit x^∞ with $t \to \infty$. Furthermore x^∞ is an optimal solution to (4.1). The second part of this result follows easily once the first is established. If the set of optimal solutions to (4.1) has only one element $x^\star$, then it is easier to show that $x(t) \to x^\star$, this basically follows from the fact that $c^\top x(t)$ tends to the optimal value and $x(t)$ approaches the feasible region. However, if the optimal solution is not unique, then convergence of $x(t)$ is far from clear; $x(t)$ could oscillate around the optimal set without converging to a single point. A proof that this does not happen, gives us evidence that Physarum dynamics behaves nicely.

To prove convergence to a limit it is enough to show that

$$\int_0^\infty \|\dot{x}(t)\| dt < \infty.$$

For this, it suffices to show $\|\dot{x}(t)\| = O(e^{-\varepsilon t})$ for some $\varepsilon > 0$. We will now focus on this task. Recall that $\dot{x}_i = \frac{x_i}{c_i}\left(a_i^\top p - c_i\right)$. Using similar tools as in the proof of existence, one can show that $|a_i^\top p(t)|$ is uniformly bounded over all $t \in [0, \infty)$. Hence, if for some $i \in \{1, 2, \ldots, n\}$ we have $x_i(t) = O(e^{-\varepsilon t})$, then it follows easily that $|\dot{x}_i(t)| = O(e^{-\varepsilon t})$. Denote the set of all such i's by N and its complement by J. We can then prove that J is the union of supports of all optimal solutions to (4.1). This in turn implies that the subvector $x_J(t)$ behaves in a sense more stably. We show in particular, that for some optimal solution f, such that $\operatorname{supp}(f) = J$ and some $\varepsilon > 0$, we have $\varepsilon f \leqslant x(t)$ uniformly over $t \in [0, \infty)$. Furthermore the key Lemma 4.16 establishes some form of continuity of $a_i^\top p(t)$ for $i \in J$. It implies that $|a_i^\top p(t) - c_i| = O(e^{-\varepsilon t})$, which concludes the proof.

The next theorem we discuss is Theorem 4.3. It shows that a discretization of the Physarum dynamics is possible, provided that the step length is small enough. The crucial parameter which controls the step length is $\|A^\top p\|_\infty$; if one is able to upper bound it over all possible x by some number $P_{\max}$ then the step length of roughly $P_{\max}^{-2}$ guarantees convergence. In the discretization we choose the starting point $x(0)$ to be feasible, by which $x(k)$ will remain feasible for every k. Therefore, by Corollary 4.1 one can take $P_{\max} = \mathcal{D} \cdot C_s$. Let us discuss it briefly. One can show that $c^\top x(k)$ is decreasing with k. However, we cannot guarantee a large drop of this value at every step. Instead we introduce another potential function

$$\mathcal{B}(k) \overset{\text{def}}{=} \sum_{i=1}^{n} c_i x_i^\star \ln x_i(k)$$

(where $x^\star$ is any optimal solution to (4.1)) and show that at every step when $c^\top x(k) - c^\top x(k+1)$ is small, $\mathcal{B}(k)$ grows by a considerable amount. Moreover, one can show that $\mathcal{B}(k)$ is uniformly upper-bounded. By combining $c^\top x(k)$ and $\mathcal{B}(k)$ into a single potential function $\phi(k)$ we can easily track the progress of our algorithm.

We now proceed with the discussion of 4.4. As in the continuous dynamics and in the proof of Theorem 4.3 we have to prove that the dynamics is indeed well defined, i.e., that all the iterates are *strictly positive*. Here, the notion of x-capacitated flows comes to our rescue: a flow is said to be x-capacitated if $Bf = b$ and $0 \leqslant f \leqslant x$. One would perhaps hope that when we choose an $x(0)$ such that there is an $x(0)$-capacitated flow then it follows that there is an $x(k)$-capacitated flow for all k. This property unfortunately does not hold and presents itself as a significant hurdle. By mimicking the proof for the continuous dynamics it is still possible to prove (though the argument is quite involved) that for every initial point $x(0)$, there is some $h > 0$ for which the discretization gives a well defined sequence of positive vectors $x(k)$, which converges to the optimal solution. However, they key problem is that the step length h needs to be very small, in order to get such a result. In other words, the resulting algorithm is not efficient. The reason for this is the chaotic behavior of the process in the initial steps. It is not clear how one could provide an initial point which makes the process "smooth" enough. This is where preconditioning comes in: for the preconditioned instance, one can show that if $x(j)$s are positive for $j = 0, 1, \ldots, k$, then there is an $x(k)$-capacitated flow. Such a flow, in turn, allows us to prove by a variant of the sweep-cut argument that the maximum potential difference corresponding to $q(k)$ remains bounded. This in turn allows us to prove that $x(k+1)$ is positive and we can continue. The potential bounding step requires h to be roughly at most $1/n\tilde{C}$ where $\tilde{C}$ is the largest cost in the preconditioned instance.

What about the number of iterations required to converge? The key is to start by noting that

$$x_e(k) = x_e(0)(1-h)^k + (1-(1-h)^k)\bar{q}_e$$

where $\bar{q}(k)$ is a certain *geometric time average* of the flows $q(j)$s, giving more importance to the newer flows than the older ones. Thus, as k increases, $x(k) \approx \bar{q}(k)$. Thus $x(k)$, for a large enough k, is nearly a flow and, similarly, $\bar{q}(k)$ is a flow but can have a small negative component. The results of [IJNT11, BBD$^+$13] then allow us to round this flow to an optimal flow, provided for every non-optimal flow g among the vertices of the flow polytope, there is an edge e supported in g for which $x_e(k)$ is tiny after a small number of iterations. This, via standard arguments, allows us to complete the proof of the theorem.

To establish this property for any non-optimal vertex flow g, we consider the barrier function $\mathcal{B}_g(k) = \sum_e g_e c_e \ln x_e(k)$. With a bit of effort, it can be shown that $\mathcal{B}_g(k) \to -\infty$. In fact, one can show that $\mathcal{B}_g(k) \lesssim -hk + \tilde{C} b_P$. On the other hand, if $x_e(k) > \varepsilon$ for all edges e with $g_e > 0$, then we get a lower bound of $\tilde{C} b_P \varepsilon$. Thus, for $k \approx \tilde{C} b_P / h$ this cannot happen and there must be an edge e, where $g_e > 0$ but $x_e(k) < \varepsilon$. This completes an overview of the proof; details appear in Section 4.3.6.

4.3 Proofs

4.3.1 Physarum as Steepest Descent on a Manifold

In this section we interpret the Physarum dynamics as steepest descent on a certain manifold. We begin with a section containing basic background on Riemannian manifolds. Afterwards, we equip the positive orthant with a Riemannian structure suitable for our problem. In particular we formalize Theorem 4.5 in Theorem 4.6 and prove it. The entropy barrier interpretation is given in section 4.3.2

Riemannian Manifolds.

All manifolds we deal with can be seen as open subsets of Euclidean spaces, possibly embedded in Euclidean spaces of higher dimension. Some examples of those are $\mathbb{R}^n$, $\mathbb{R}^n_{>0}$, open polyhedra $\{x \in \mathbb{R}^n : Ax = b, x > 0\}$ or a sphere $\{x \in \mathbb{R}^n : \|x\|_2 = 1\}$. If M is such a manifold, then at every point $x \in M$ one can define a tangent space $T_x M$ in a natural way. A tangent space is equipped with a natural linear structure. For example, the tangent space to any point $x \in M = \{x : Ax = b\}$ is $T_x M = \{x : Ax = 0\}$. Given a manifold M, we can endow it with a Riemannian structure by assigning to every point x a scalar product $\langle \cdot, \cdot \rangle_x$ on $T_x M$ in such a way that $\langle \cdot, \cdot \rangle_x$ varies smoothly with x'. When considering a function $F : M \to \mathbb{R}$ on a Riemannian manifold, the gradient of F is defined with respect to the local inner product. Hence, in particular, one can define gradient-descent algorithms based on a chosen Riemannian structure of the space, those are typically very different to algorithms based on the standard Euclidean geometry.

The Riemannian Structure and its Origin

We focus on two domains:

$$\Omega = \mathbb{R}^n_{>0} \quad \text{and} \quad \Omega^f = \{x \in \Omega : Ax = b\}.$$

Note that Ω and Ω^f are manifolds of dimension n and m respectively. We consider a Riemannian structure $\langle \cdot, \cdot \rangle_x$ on Ω (which automatically induces a Riemannian structure on the submanifold $\Omega^f \subseteq \Omega$) given as a family of positive definite matrices $H(x) = CX^{-1}$ for $x \in \Omega$. This gives rise to the following inner product on the tangent space at x:

$$\langle u, v \rangle_x \stackrel{\text{def}}{=} u^\top H(x) v.$$

We present two interpretations of where does this structure come from. The first one is related to the following entropy-like function $h : \Omega \to \mathbb{R}$ given by

$$h(x) = \sum_{i=1}^{n} c_i x_i \ln(c_i x_i) - \sum_{i=1}^{n} c_i x_i.$$

Note that

$$\nabla h(x) = C \ln(Cx) \qquad \text{and} \qquad \nabla^2 h(x) = CX^{-1},$$

hence $H(x) = \nabla^2 h(x)$ is an example of a so called *Hessian Metric*.
The second interpretation is based on the analysis of the following map: $F : \Omega \to \Omega$ given by

$$F(x) = 2\sqrt{Cx}.$$

We will now explain that F can be seen as an isometry between our Riemannian manifold $(\Omega, \langle \cdot, \cdot \rangle_x)$ and Ω with the standard inner product $\langle \cdot, \cdot \rangle$ at every point. Recall that the Jacobian of F[3], $J(x) = (CX^{-1})^{1/2}$ acts as a linear map $J(x) : T_x(\Omega) \to T_x(\Omega)$, i.e.:

$$J(x)(u) = J(x)u = \left(CX^{-1}\right)^{1/2} u.$$

Take any $x \in \Omega$ and u, v from the tangent space at x, i.e. $T_x(\Omega) \cong \mathbb{R}^n$. We have

$$\langle J(x)(u), J(x)(v) \rangle = \left\langle (CX^{-1})^{1/2} u, (CX^{-1})^{1/2} v \right\rangle = u^\top CX^{-1} v = u^\top H(x) v = \langle u, v \rangle_x .$$

Thus the inner product $\langle \cdot, \cdot \rangle_x$ can be seen as a pull-back of the standard inner product via the map F. This also means that F is an isometry: it preserves angles and distances. Let us call the space obtained by the transformation $x \mapsto F(x)$ the y-space[4]. The point x from the x-space is represented by $y = 2\sqrt{Cx}$ in the y-space. The inverse mapping is $x = \frac{1}{4} C^{-1} y^2$. The strictly feasible set $\Omega^f = \{x \in \mathbb{R}^n : Ax = b, x > 0\}$ corresponds to

$$M = \left\{ y \in \mathbb{R}^n : \frac{1}{4} AC^{-1} y^2 = b, y > 0 \right\}$$

in the y-space.

Two Forces

We intend to show that the direction $P(x) = W(A^\top L^{-1} b - c)$ of the Physarum dynamics (4.2) at the point x decomposes naturally into two directions

$$P_f(x) \stackrel{\text{def}}{=} WA^\top \left(AWA^\top\right)^{-1} (b - Ax),$$
$$P_o(x) \stackrel{\text{def}}{=} W \left(A^\top \left(AWA^\top\right)^{-1} Ax - c\right)$$

[3]Jacobian is the multidimensional analogue of a derivative. In the standard coordinate system it can be expressed as a matrix of partial derivatives.

[4]Of course formally $F(\Omega) = \Omega$, hence the x-space and y-space coincide as sets, the difference however lies in the geometry.

with $P_f(x)$ interpreted as the *feasibility direction* and $P_o(x)$ as the *optimization direction*. As one can easily see $P(x) = P_f(x) + P_o(x)$. We will interpret both directions by studying the optimization problem arising after transforming the linear program (4.1) into the y-space via the map F.

Feasibility. Fix any $\bar{x} \in \Omega$ and consider its image $\bar{y}$ in the y-space. Recall that the strictly feasible region of (4.1) in the y-space is

$$M = \left\{ y > 0 : \frac{1}{4} AC^{-1} y^2 = b \right\}.$$

Let us assume $\bar{y} \notin M$. In which direction do we need to move, to hit M as soon as possible? This does not seem to be a simple problem, because it is non-convex. However, let us try to guess a $y^\star$ which satisfies

$$\|\bar{y} - y^\star\| = \min \{\|\bar{y} - y\| : y \in M\}$$

and set $h = y^\star - \bar{y}$. Since the above minimization is done with respect to the Euclidean distance, we expect the vector h to be orthogonal to the "surface" M at $y^\star$, equivalently to the tangent space $T_{y^\star}M$ i.e. to every vector u which is tangent to the space M at the point $y^\star \in M$. It is easy to see that

$$T_{y^\star}M = \left\{ u \in \mathbb{R}^n : \frac{1}{2} AC^{-1} Y^\star u = 0 \right\}.$$

Since h is just the orthogonal projection of $y^\star - \bar{y}$ onto $T_{y^\star}M$, we can obtain it by the well-known formula[5]:

$$\begin{aligned} h &= Y^\star C^{-1} A^\top \left(A (Y^\star C^{-1})^2 A^\top\right)^{-1} AC^{-1} Y^\star (y^\star - \bar{y}) \\ &= Y^\star C^{-1} A^\top \left(A (Y^\star C^{-1})^2 A^\top\right)^{-1} (AC^{-1}(y^\star)^2 - AC^{-1} Y^\star \bar{y}) \\ &= Y^\star C^{-1} A^\top \left(A (Y^\star C^{-1})^2 A^\top\right)^{-1} (4b - AC^{-1} Y^\star \bar{y}). \end{aligned}$$

We do not know what $y^\star$ is, but drawing from the expectation that $\bar{y}$ is close to feasibility, we can approximate $y^\star \approx \bar{y}$, hence obtaining

$$h \approx \bar{Y} C^{-1} A^\top \left(A \left(\bar{Y} C^{-1}\right)^2 A^\top \right)^{-1} \left(4b - AC^{-1} \bar{y}^2\right).$$

Let us pull back h to the x-space[6] by substituting $\bar{y} = 2\sqrt{C\bar{x}}$ and multiplying h by the inverse-Jacobian $J^{-1}(\bar{x}) = \text{Diag}\left(\frac{\bar{x}_1}{c_1}, \dots, \frac{\bar{x}_n}{c_n}\right)^{1/2} = W^{1/2}$. We get

$$2WA^\top \left(AWA^\top\right)^{-1} (b - A\bar{x}) = 2P_f(\bar{x}),$$

which is the feasibility direction up to the a scaling factor 2.

Optimality. As in the previous discussion fix $\bar{x} \in \Omega$. We will interpret the optimization direction at $\bar{x}$. We know that $\bar{x}$ may not satisfy $A\bar{x} = b$, but we still can write $A\bar{x} = b^{\bar{x}}$, where $b^{\bar{x}} = b - (b - A\bar{x})$ which we should imagine to be "close" to b. Our goal is to find the minimum value

[5]The orthogonal projection onto the row space of a matrix B is given by $B^\top (BB^\top)^{-1} B$.

[6]More accurately: from the space $T_{\bar{y}}(\Omega)$ to the space $T_{\bar{x}}(\Omega)$.

of $c^\top x$ over the set $\{x : x \geqslant 0, Ax = b\}$. Let us consider a related linear program:

$$\begin{aligned} \text{min.} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b^{\bar{x}} \\ & x \geqslant 0, \end{aligned} \tag{4.3}$$

which is just minimization of $c^\top x$ over an affine subspace parallel to $Ax = b$ which contains the point $\bar{x}$. Let us rewrite (4.3) in the y-space.

$$\begin{aligned} \text{min.} \quad & \frac{1}{4} 1^\top y^2 \\ \text{s.t.} \quad & \frac{1}{4} A C^{-1} y^2 = b^{\bar{x}} \\ & x \geqslant 0, \end{aligned} \tag{4.4}$$

where 1 denotes the all-one vector. Let $\bar{y} = F(\bar{x})$ and denote

$$M^{\bar{x}} \overset{\text{def}}{=} \left\{ y : y > 0, \frac{1}{4} A C^{-1} y^2 = b^{\bar{x}} \right\}.$$

Note that $\bar{y} \in M^{\bar{x}}$. Let us then compute the steepest descent direction at $\bar{y}$ with respect to the problem (4.4). We just need to compute the negative gradient of the objective at $\bar{y}$ and project it onto the tangent space $T_{\bar{y}}(M^{\bar{x}})$. We have

$$\nabla_y \left(\frac{1}{4} 1^\top y^2 \right) = \frac{1}{2} y.$$

Hence we need to project the vector $-\frac{1}{2}\bar{y}$ onto the space:

$$T_{\bar{y}}(M^{\bar{x}}) = \left\{ u \in \mathbb{R}^n : \frac{1}{2} A C^{-1} \bar{Y} u = 0 \right\}.$$

As a result we obtain

$$\begin{aligned} u &= \left(I - \bar{Y} C^{-1} A^\top \left(A (\bar{Y} C^{-1})^2 A^\top \right)^{-1} A C^{-1} \bar{Y} \right) \left(-\frac{1}{2} \bar{y} \right) \\ &= \frac{1}{2} \left(\bar{Y} C^{-1} A^\top \left(A (\bar{Y} C^{-1})^2 A^\top \right)^{-1} A C^{-1} \bar{y}^2 - \bar{y} \right). \end{aligned}$$

To get a corresponding direction in the x-space, we compute

$$\begin{aligned} J^{-1}(\bar{x})(u) &= \frac{1}{2} W^{1/2} \left(\bar{Y} C^{-1} A^\top \left(A \left(\bar{Y} C^{-1} \right)^2 A^\top \right)^{-1} A C^{-1} \bar{y}^2 - \bar{y} \right) \\ &= \frac{1}{2} W^{1/2} \left(\frac{1}{2} W^{1/2} A^\top \left(A W A^\top \right)^{-1} A C^{-1} 4 C \bar{x} - 2 \sqrt{C \bar{x}} \right) \\ &= W \left(A^\top \left(A W A^\top \right)^{-1} A \bar{x} - c \right) = P_o(\bar{x}). \end{aligned}$$

Which is the optimization direction at $\bar{x}$. Note that in case when x is feasible, the feasibility direction $P_f(x)$ is zero, hence the following theorem holds.

Theorem 4.6

Consider the manifold $\Omega^f = \{x : Ax = b, x > 0\}$ endowed with the Riemannian metric $\langle u, v\rangle_x = u^\top C X^{-1} v$ at every $x \in \Omega^f$. Then, the Physarum direction

$$P(x) = W\left(A^\top \left(AWA^\top\right)^{-1} b - c\right)$$

is the gradient of the objective function $c^\top x$ with respect to the Riemannian manifold $\left(\Omega^f, \langle\cdot,\cdot\rangle_x\right)$.

Proof.
One can use the above derivation of the optimization direction to deduce this result. If x is feasible then

$$P_o(x) = W\left(A^\top \left(AWA^\top\right)^{-1} Ax - c\right) = W\left(A^\top \left(AWA^\top\right)^{-1} b - c\right) = P(x).$$

□

4.3.2 Physarum as an Entropy Barrier Method

In this section we show that every Physarum trajectory starting from a strictly feasible point can be seen as a path of optimizers to a certain family of convex programs. Let us fix any starting point $s > 0$ satisfying $As = b$ and consider for every $\mu \geqslant 0$ the following convex program

$$\begin{aligned} \text{min.} \quad & \mu c^\top x + \sum_{i=1}^n x_i c_i \ln(x_i c_i) - \sum_{i=1}^n x_i c_i (1 + \ln(s_i c_i)) \\ \text{s.t.} \quad & Ax = b \\ & x \geqslant 0. \end{aligned} \tag{4.5}$$

Note that one can rewrite the objective function equivalently as

$$c^\top x + \frac{1}{\mu} f^s(x).$$

Which is just our original linear objective, regularized by an entropy-like strongly convex function

$$f^s(x) \stackrel{\text{def}}{=} \sum_{i=1}^n x_i c_i \ln(x_i c_i) - \sum_{i=1}^n x_i c_i (1 + \ln(s_i c_i)).$$

Observe that $\nabla^2 f^s(x) = CX^{-1}$ is the Riemannian metric we are studying above. It should be intuitively clear that as $\mu \to \infty$, the solution to (4.5) will tend to the optimal solution of the linear program (4.1).

Theorem 4.7

Take any starting point $s > 0$ with $As = b$. Suppose that $x(\mu)$ is the unique optimal solution to (4.5). Then $x(\mu)$ for $\mu \in [0, \infty)$ is the solution to the Physarum dynamical system with the initial condition $x(0) = s$.

The idea of the proof is to write down KKT conditions for the convex program (4.5) and take advantage of strong duality. By implicit differentiation we conclude that $x(\mu)$ follows the same dynamics as Physarum, furthermore $x(0) = s$, hence they have to coincide.

Proof.
To improve readability, we will assume that $c = 1$, i.e. $c_i = 1$ for all $i = 1, \ldots, n$. To obtain the general version one can go back and forth with the substitution $x \mapsto C^{-1}z$.

Fix any $s > 0$ such that $As = b$. Let us denote $f^s(x) = \sum_i x_i \ln x_i - \sum_i x_i(1 + \ln s_i)$. First one needs to show that in fact for every $\mu \geqslant 0$ there exists a minimizer of $\mu 1^\top x + f^s(x)$ in the set $\{x : Ax = b, x > 0\}$, in other words that $x(\mu) > 0$. This can be seen as follows: pick any point $\bar{x}$ on the boundary, i.e. having $\bar{x}_i = 0$ for some i. Consider the following scalar function

$$h(p) = \mu 1^\top(\bar{x} + p(s - \bar{x})) + f^s(\bar{x} + p(s - \bar{x}))$$

for $p \in [0, 1]$; $h(p)$ is continuous on $[0, 1]$ and differentiable in the interval $(0, 1)$. We have

$$\frac{d}{dp}h(p) = \mu 1^\top(s - \bar{x}) + (s - \bar{x})^\top(\ln(\bar{x} + p(s - \bar{x})) - \ln s).$$

This implies that $h'(p) \to -\infty$ as $p \to 0$, by looking at a coordinate i where $\bar{x}_i = 0$. Hence $h(\varepsilon) < h(0)$ for $\varepsilon > 0$ small enough, implying (together with convexity) that the optimal value of $\mu 1^\top x + f^s(x)$ over $\{x : Ax = b, x \geqslant 0\}$ is attained at some point $x(\mu) > 0$.

Fix $\mu \geqslant 0$, introduce dual variables $y \in \mathbb{R}^m$ for the equality constraints and consider the Lagrangian:

$$L(x, y) = \mu 1^\top x + \sum_i x_i \ln x_i - \sum_i x_i(1 + \ln s_i) - y^\top(Ax - b).$$

We always keep in mind that $x > 0$. Fix $y \in \mathbb{R}^m$ and let us compute the derivative of $L(x, y)$:

$$\nabla_x L(x, y) = \mu 1 + (\ln x - \ln s) - A^\top y.$$

the derivative is 0 at a point x given by

$$\begin{aligned} \ln x_i &= a_i^\top y - \mu + \ln s_i \\ x_i &= s_i \cdot \exp(a_i^\top y - \mu) \end{aligned}$$

Then the dual objective $g(y)$ is given by substituting the above x (at which $L(x, y)$ is minimized) in $L(x, y)$. After some cancellations we get

$$g(y) = y^\top b - \sum_i x_i = y^\top b - \sum_i s_i \cdot \exp(a_i^\top y - \mu)$$

Hence the dual program is:

$$\begin{aligned} \min \quad & y^\top b - \sum_i s_i \cdot \exp(a_i^\top y - \mu) \\ \text{s.t.} \quad & y \in \mathbb{R}^m. \end{aligned}$$

Note that strong duality holds since Slater condition is satisfied for the primal (the witness being s). This means that the unique maximizer of g(y) will also yield the optimal solution for the primal. Let

$$y(\mu) \overset{\text{def}}{=} \operatorname{argmax}\{y^\top b - \sum_i s_i \cdot \exp(a_i^\top y - \mu) : y \in \mathbb{R}^m\},$$

$$x_i(\mu) \overset{\text{def}}{=} s_i \cdot \exp(a_i^\top y(\mu) - \mu).$$

Then $(x(\mu), y(\mu))$ is the optimal primal-dual pair.

We will show that $x(\mu)$ is a solution to the dynamical system

$$\frac{d}{d\mu} x(\mu) = P(x(\mu)) = X\left(A^\top \left(AXA^\top\right)^{-1} AX1 - 1\right),$$

which represents Physarum when $c = 1$. Let us start by examining the initial condition. If $\mu = 0$ then one can easily check that the optimal pair is $(x(0), y(0)) = (s, 0)$, as claimed. Let us now compute the derivative of $x(\mu)$ w.r.t. μ, note that the implicit function theorem guarantees that $(x(\mu), y(\mu))$ is a continuously differentiable function of μ. In the following calculations we write shortly x, y for $x(\mu)$ and $y(\mu)$. We have

$$\dot{x}_i = s_i \cdot \exp(a_i^\top y - \mu) \cdot (a_i^\top \dot{y} - 1) = x_i(a_i^\top \dot{y} - 1).$$

In other words $\dot{x} = X(A^\top \dot{y} - 1)$. Let us now take the equality $Ax = b$ and differentiate w.r.t. μ:

$$A\dot{x} = 0$$
$$AX(A^\top \dot{y} - 1) = 0.$$

In consequence, $\dot{y}$ satisfies $(AXA^\top)\dot{y} = AX1$. Since $\operatorname{rank}(A) = m$, one can show that $AXA^\top$ is invertible, hence $\dot{y}$ is uniquely determined by $\dot{y} = (AXA^\top)^{-1}AX1$. Consequently

$$\dot{x} = X\left(A^\top \dot{y} - 1\right) = X\left(A^\top \left(AXA^\top\right)^{-1} AX1 - 1\right) = P(x).$$

□

4.3.3 Existence of Solution

In this section we would like to argue that no matter what the initial condition $x(0) > 0$ is, the Physarum dynamics has a global solution $x : [0, \infty) \to \mathbb{R}^n_{>0}$. Let us start by explaining what is the significance of the assumption $x(t) > 0$. The Physarum dynamics is defined as a dynamical system $\dot{x} = P(x)$, where $P(x) = W(A^\top(AWA^\top)^{-1}b - c)$. For $x > 0$ one can show that $(AWA^\top)^{-1}b$ exists, (since the kernels of $AWA^\top$ and $A^\top$ coincide) hence $P(x)$ is well defined and continuous over $\Omega = \mathbb{R}^n_{>0}$. Can we extend it out of Ω? This leads to troubles, because in general $AWA^\top$ might be singular when w contains some zeros or negative entries. For example if $w = 0$ then clearly $AWA^\top = 0$, hence there is no hope to extend the vector field to $x = 0$. In general, there is no simple way to extend (continuously) this vector field from $\Omega = \mathbb{R}^n_{>0}$ to a larger set.

We start by a section with a general discussion of the existence problem and prove a simple, yet very useful existence theorem which makes apparent what one has to argue to conclude existence.

Dynamical Systems and Global Existence

We work in the n-dimensional Euclidean space $\mathbb{R}^n$, an open subset $\Omega \subseteq \mathbb{R}^n$ is chosen for the domain of the dynamical system. The equations are of the form

$$\dot{x}_i(t) = F_i(x_1(t), x_2(t), \dots, x_n(t)) \qquad \text{for } i = 1, 2, \dots, n,$$

where $x_1, x_2, \dots, x_n$ are functions of single variable t regarded as unknowns and $F_1, F_2, \dots, F_n : \Omega \to \mathbb{R}$ are given. The above system is often denoted shortly as

$$\dot{x} = F(x),$$

where $F : \Omega \to \mathbb{R}^n$ is just $F = (F_1, F_2, \dots, F_n)$. Typically we impose some initial condition of the form $x(0) = s$ with $s \in \Omega$. A solution to such a system is a function $x : I \to \Omega$, where I is some non-degenerate interval containing 0, $x(0) = s$ and $\dot{x}(t) = F(x(t))$ for every t in the interior of I.

We are interested in the question: when does a dynamical system have global solutions? In our case it is not difficult to obtain a solution on some small interval $(-\varepsilon, \varepsilon)$. However, for applications we need existence on $[0, +\infty)$ (we call it global existence), which is harder to establish. Intuitively, it may happen that at some finite time step T, the solution x will "escape" from the domain Ω. More formally, even if the solution x exists in the interval $[0, T)$, the limit $\lim_{t\to T^-} x(t)$ may be outside of Ω or the limit may not even exist. Escaping from Ω is not only a formal problem of defining Ω, it is actually a serious issue, because in our case, the function F is not well defined outside of Ω.

In this section we provide a sufficient condition for a dynamical system to have global solutions. Later on we prove that in fact the Physarum dynamical system satisfies this condition.

We first state an important general theorem which gives sufficient conditions for existence on some interval around 0. A proof can be found in any textbook on dynamical systems, e.g. [Per01].

Theorem 4.8

Let $\Omega \subseteq \mathbb{R}^n$ be an open set, $s \in \Omega$ and $F : \Omega \to \mathbb{R}^n$ be a function of class $\mathcal{C}^1$. Consider the dynamical system $\dot{x} = F(x)$ with initial condition $x(0) = s$. There exists a unique solution $x : (-\varepsilon, \varepsilon) \to \Omega$ to the system, for some $\varepsilon > 0$.

We conclude this subsection by proving a theorem that a certain simple condition implies existence on the whole half-line.

Theorem 4.9

Let $\Omega \subseteq \mathbb{R}^n$ be an open set, $s \in \Omega$ and $F : \Omega \to \mathbb{R}^n$ be a function of class $\mathcal{C}^1$. Consider the dynamical system $\dot{x} = F(x)$ with initial condition $x(0) = s$. Suppose it satisfies the following conditions for every solution $x : [0, T) \to \Omega$ with $T \in (0, \infty)$:

- *the limit $\lim_{t\to T^-} x(t)$ exists,*
- *if $x(T) = \lim_{t\to T^-} x(t)$ then $x(T) \in \Omega$.*

Then the there exists a global solution $x : [0, \infty) \to \Omega$.

Proof.
Let T_0 be the maximal T such that a solution $x : [0, T) \to \Omega$ exists (we allow $T_0 = \infty$). One

should argue why does such a T_0 exists. This is a consequence of uniqueness: whenever there are two solutions $x : [0, T_1) \to \Omega$ and $y : [0, T_2) \to \Omega$ with $T_1 \leqslant T_2$, then they agree on $[0, T_1)$.

From Theorem 4.8 we know that $T_0 > 0$. We want to show that $T_0 = \infty$. For the sake of contradiction assume that T_0 is a finite number. Denote by $x(T_0)$ the limit $\lim_{t \to T_0^-} x(t)$. By assumption $x(T_0) \in \Omega$, thus we can use Theorem 4.8 to extend the solution from $[0, T_0)$ to $[0, T_0 + \varepsilon)$ for some $\varepsilon > 0$. This gives us a contradiction with the definition of T_0. □

An example of a system which does not satisfy the assumption of Theorem 4.9.

Example 4.1

Consider a one-dimensional system $\dot{x} = \frac{1}{1-x}$ with $x(0) = 0$. The natural choice for the domain is $\Omega = \mathbb{R} \setminus \{1\}$ (only on this set the function $x \mapsto \frac{1}{1-x}$ makes sense). Let $T = \frac{1}{2}$, then there is a solution $x : [0, T) \to \Omega$ given by $x(t) = 1 - \sqrt{1 - 2t}$. However

$$\lim_{t \to T^-} x(t) = 1.$$

But $1 \notin \Omega$, hence this system does not satisfy the condition from Theorem 4.9. Note also that actually there is no solution to this system on the whole half-line $[0, \infty)$.

Basic Results

This section introduces some preparatory results which are essential for studying the existence of solution and then asymptotic properties of the Physarum dynamics. Recall that we are always assuming that the feasible region $\{x : Ax = b, x \geqslant 0\}$ is nonempty.

Remark 4.1

In the proofs we will repeatedly use the fact that polyhedra of the kind $\{x : Ax = b, x \geqslant 0\}$ always have at least one vertex. Let $\bar{x}$ be such a vertex. Denote $Z = \{j \in [n] : \bar{x}_j = 0\}$ and $N = [n] \setminus Z$. Then $\bar{x}_N$ (i.e. the subvector of $\bar{x}$ consisting of entries x_j for $j \in N$) can be determined as a unique solution to the linear system $A_N x_N = b$, where A_N is a submatrix of A consisting of columns a_j for $j \in N$, see e.g. [Kar91].

Our quantitative bounds regarding convergence and related aspects of Physarum often involve the following two quantities:

- $C_s = \sum_{i=1}^{n} c_i$,
- $\mathcal{D} = \max\{|\det(A')| : A'$ square submatrix of $A\}$.

Below we present a key lemma, which then allows us to get a solid grasp on the behavior of the vector field $P(x)$ defining Physarum dynamics.

Lemma 4.1

Let $w \in \mathbb{R}^n_{>0}$, $W = \mathrm{Diag}(w)$, $L = AWA^\top$. There exists a constant $\alpha > 0$ depending only on A such that for every $i \in [n]$,

$$\|A^\top L^{-1} a_i\|_\infty \leqslant \frac{\alpha}{w_i}.$$

Quantitatively, one can take $\alpha = \mathcal{D} = \max\{|\det(A')| : A'$ is a square submatrix of $A\}$.

Proof.
Fix i and denote by p the solution to the system $Lp = a_i$. We can assume that $p^\top a_j \geqslant 0$ for every $j \in [m]$ by replacing the row a_j by $-a_j$ if necessary. One can easily see that such a change does not alter the problem, because L remains the same.

Let us first show that $a_i^\top L^{-1} a_i \leqslant \frac{1}{w_i}$. Note that

$$w_i a_i a_i^\top \preceq \sum_{j=1}^m w_j a_j a_j^\top = L,$$

where $\preceq$ is the PSD order. This means that $w_i u^\top a_i a_i^\top u \leqslant u^\top L u$, for every $u \in \mathbb{R}^n$. Let us pick $u = L^{-1} a_i$. We get

$$\begin{aligned} w_i u^\top a_i a_i^\top u &\leqslant u^\top L u \\ w_i a_i^\top L^{-1} a_i a_i^\top L^{-1} a_i &\leqslant a_i^\top L^{-1} L L^{-1} a_i \\ w_i (a_i^\top L^{-1} a_i)^2 &\leqslant a_i^\top L^{-1} a_i \\ a_i^\top L^{-1} a_i &\leqslant \frac{1}{w_i}. \end{aligned}$$

It remains to argue that for some α and for every $k \in [n]$,

$$a_k^\top p \leqslant \alpha a_i^\top p.$$

Fix $k \in [n]$, assume $k \neq i$. If $a_k^\top p = 0$, then we are done, assume $a_k^\top p > 0$. From $Lp = a_i$ we get

$$\sum_{j=1}^m w_j a_j (a_j^\top p) = a_i.$$

Hence the set $S_k \stackrel{\text{def}}{=} \{s \in \mathbb{R}^m_{\geqslant 0} : As = a_i \wedge s_k > 0\}$ is nonempty ($WA^\top p$ belongs to it). Take $s \in S_k$ with s_k maximum possible (it can be seen that s_k is bounded over all $s \in S_k$). Then $\sum_{j=1}^m s_j a_j = a_i$, hence $\sum_{j=1}^m s_j a_j^\top p = a_i^\top p$. Since $s_j a_j^\top p \geqslant 0$ for all j, we can deduce that $s_k a_k^\top p \leqslant a_i^\top p$ and hence $a_k^\top p \leqslant \frac{a_i^\top p}{s_k} = \alpha_k a_i^\top p$. It is enough to choose $\alpha = \max_k \alpha_k$.

For the quantitative bound one needs to note that α is chosen according to the following values: $\varepsilon_k = \max\{s_k : As = a_i, s \geqslant 0\}$. In fact $\alpha = \max_k \frac{1}{\varepsilon_k}$. Because linear programs attain optimal values in vertices, one can argue that $s^\star$ – the optimal solution to $\max\{s_k : As = a_i, s \geqslant 0\}$ (for some fixed k) can be chosen to be a vertex of the polyhedron $\{s : As = a_i, s \geqslant 0\}$. By the Cramer's rule, every positive entry of $s^\star$ is lower-bounded by $\mathcal{D}^{-1}$. □

Let us now see what happens when $w = C^{-1}x$ comes from a feasible vector.

Corollary 4.1

Suppose that $x > 0$ and $Ax = b$, we have

$$\|A^\top p\|_\infty \leqslant \mathcal{D} \cdot C_s.$$

Proof.

Let $w = C^{-1}x$, using Lemma 4.1 we get

$$\begin{aligned}\|A^\top p\|_\infty = \|A^\top L^{-1}b\|_\infty &= \|\sum_{i=1}^n A^\top L^{-1}(a_i x_i)\|_\infty \\ &\leqslant \sum_{i=1}^n x_i \|A^\top L^{-1} a_i\|_\infty \\ &\leqslant \sum_{i=1}^n x_i \frac{\mathcal{D}}{w_i} = \mathcal{D} \cdot C_s.\end{aligned}$$

□

It turns out that if we could prove a result such as above for all $x \in \Omega$, not only for feasible x, it would imply the existence of solution. Unfortunately, this ceases to be true without this restriction. Intuitively, our strategy is to prove that $\|A^\top p\|$ is bounded over every trajectory $\{x(t) : 0 \leqslant t \leqslant T\}$, this then allows us to reason that no trajectory will leave Ω. The next lemma shows a uniform bound but with $\|A \top p\|$ replaced by $\|W A^\top p\|$.

Lemma 4.2

Suppose y is feasible: $Ay = b$ and $y \geqslant 0$. Then for every $x \in \mathbb{R}^n_{>0}$ it holds that $\|q\|_\infty \leqslant \alpha \|y\|_1$. As in Lemma 4.1 we can take $\alpha = \mathcal{D}$.

Proof.
Recall that $q = WA^\top L^{-1}b$, hence $q_i = w_i a_i^\top L^{-1}b$. We express b as $b = Ay = \sum_{j=1}^n y_j a_j$.

$$\begin{aligned}|q_i| = \left| w_i a_i^\top L^{-1}\left(\sum_{j=1}^n y_j a_j\right)\right| &= \left|\sum_{j=1}^n y_j w_i a_i^\top L^{-1} a_j\right| \\ &\leqslant \sum_{j=1}^n w_i y_j \left|a_i^\top L^{-1} a_j\right| = \sum_{j=1}^n w_i y_j \left|a_j^\top L^{-1} a_i\right| \\ &\overset{\text{Lemma 4.1}}{\leqslant} \sum_{j=1}^n w_i y_j \frac{\alpha}{w_i} = \alpha \|y\|_1\end{aligned}$$

Hence $\|q\|_\infty \leqslant \alpha \|y\|_1$. □

The following corollary gives a concrete bound we can deduce from the above lemma.

Corollary 4.2

For every $x \in \mathbb{R}^n_{>0}$ it holds that $\|q\|_\infty \leqslant \mathcal{D}^2 \cdot n \cdot \|b\|_1$.

Proof.
We use Lemma 4.2. Let us pick y as any vertex of the feasible region $\{x : Ax = b, x \geqslant 0\}$. We know that the non-zero portion y_N of y can be determined as a solution to the linear system $A_N y_N = b$. Hence, by Cramer's rule every non-zero entry of y can be obtained as follows:

$$y_i = \sum_{j=1}^m b_j \frac{\alpha_j}{\alpha_0}.$$

for some $\alpha_0, \alpha_1, \ldots, \alpha_m$ being determinants of square submatrices of A. Since A has integer entries, $|\alpha_0| \geqslant 1$ and $|\alpha_1|, \ldots, |\alpha_m| \leqslant \mathcal{D}$, thus $|y_i| \leqslant \|b\|_1 \mathcal{D}$ and $\|y\|_1 \leqslant n\|b\|_1 \mathcal{D}$. $\square$

The Proof of Existence

We are now ready to prove Theorem 4.1. As previously, we will use Ω to denote $\mathbb{R}^n_{>0}$. From now on we assume that the initial condition $x(0) = s \in \Omega$ is fixed. In the light of Theorem 4.9, to prove 4.1 it suffices to show the following two lemmas.

Lemma 4.3

Suppose $x : [0, T) \to \Omega$ is a solution to (4.2) for some $T \in \mathbb{R}_{>0}$ then $\lim_{t \to T^-} x(t)$ exists.

Lemma 4.4

Suppose $x : [0, T) \to \Omega$ is a solution to (4.2) for some $T \in \mathbb{R}_{>0}$. If $x(T) \overset{\text{def}}{=} \lim_{t \to T^-} x(t)$ exists then $x(T) \in \Omega$, i.e. $x(T)$ is strictly positive.

Let us start by proving that every trajectory stays in a bounded region.

Lemma 4.5

Suppose $x : [0, T) \to \Omega$ is a solution to (4.2), then for every $i \in [n]$ and $t \in [0, T)$:

$$x_i(t) \leqslant \max\left(x_i(0), \beta\right),$$

where $\beta = \mathcal{D}^2 \cdot n \cdot \|b\|_1$.

Proof.
Fix $i \in [n]$, by Corollary 4.2 we have

$$\dot{x}_i = q_i - x_i \leqslant \beta - x_i.$$

Applying Gronwall lemma to $y(t) = x_i(t) - \beta$ yields

$$x_i(t) \leqslant (1 - e^{-t})\beta + e^{-t} x_i(0) \leqslant \max\left(x_i(0), \beta\right).$$

$\square$

We are now ready to prove Lemma 4.3.

Proof of Lemma 4.3:
Suppose $x : [0, T) \to \Omega$ is a solution to (4.2). Fix $i \in [n]$, we show that $x_i(t)$ has a limit with $t \to T^-$. We know that x_i is differentiable in the interval $[0, T)$ and $|\dot{x}_i|$ is bounded, since

$$|\dot{x}_i(t)| = |q_i(t) - x_i(t)| \leqslant |q_i(t)| + |x_i(t)| \leqslant \beta + \max(x_i(0), \beta)$$

by Lemma 4.2 and Corollary 4.5. Hence $x_i(t)$ is a Lipschitz function in the interval $[0, T)$. This implies that $\lim_{t \to T^-} x_i(t)$ exists. $\square$

Now we are left with the task of proving that no coordinate of x will approach 0 as $t \to T^-$. This task simplifies significantly (as implied by the next lemma) if we assume that there is a strictly feasible solution to (4.1), i.e. $y \in \mathbb{R}^n$ such that $Ay = b$ and $y > 0$. Nevertheless, we do not want to make any such assumptions to make our proof valid in full generality. We proceed by showing that positivity holds for every $i \in \operatorname{supp}(y)$ for any feasible y, here $\operatorname{supp}(y)$ is the support of the vector y, i.e. the set $\{j \in [n] : y_j > 0\}$.

Lemma 4.6

Suppose $x : [0,T) \to \Omega$ is a solution to (4.2) and y is any feasible solution to (4.1). If $x(T) = \lim_{t\to T^-} x(t)$ then $x_i(T) > 0$ for every $i \in \operatorname{supp}(y)$.

Proof.
Fix any feasible solution y to (4.1). To justify the claim, we use the following "barrier function"

$$\mathcal{B}(t) \overset{\text{def}}{=} \sum_{j=1}^{n} y_j c_j \ln x_j(t).$$

$\mathcal{B}(t)$ is clearly well defined on $[0,T)$. Suppose for the sake of contradiction that $\inf_{t\in[0,T)} x_i(t) = 0$ for some $i \in \operatorname{supp}(y)$. We know by Corollary 4.5 that $x_j(t)$ are uniformly upper-bounded over $t \in [0,T)$. Hence we know that $y_j c_j \ln x_j(t) \leqslant M$ for all $j \in [n]$, $t \in [0,T)$ and some constant $M \in \mathbb{R}$. It follows that $\inf_{t\in[0,T)} \mathcal{B}(t) = -\infty$. This implies that the derivative $\dot{\mathcal{B}}(t)$ is unbounded from below for $t \in [0,T)$.

However, let us compute this derivative:

$$\dot{\mathcal{B}}(t) = \sum_{j=1}^{n} y_j c_j \frac{\dot{x}_j(t)}{x_j(t)} = \sum_{j=1}^{n} y_j c_j \frac{q_j(t)}{x_j(t)} - \sum_{j=1}^{n} y_j c_j = \sum_{j=1}^{n} y_j c_j \frac{q_j(t)}{x_j(t)} - c^\top y.$$

We analyze the term $\sum_{j=1}^{n} y_j c_j \frac{q_j(t)}{x_j(t)}$.

$$\sum_{j=1}^{n} y_j c_j \frac{q_j(t)}{x_j(t)} = \sum_{j=1}^{n} y_j c_j \frac{w_j a_j^\top p(t)}{x_j(t)} = \sum_{j=1}^{n} y_j a_j^\top p(t) = (Ay)^\top p(t) = b^\top p(t).$$

Note that $b^\top p = b^\top L^{-1} b \geqslant 0$, hence the above sum is nonnegative. In consequence,

$$\dot{\mathcal{B}}(t) \geqslant -c^\top y = \text{const},$$

which is a contradiction since this quantity was claimed to be unbounded from below. □

We now have all necessary tools to prove Lemma 4.4.

Proof of Lemma 4.4:
Let $x : [0,T) \to \Omega$ be a solution to (4.2). Fix y to be any feasible solution to (4.1). Lemma 4.6 implies that there exists $\varepsilon > 0$ such that $\varepsilon \cdot y \leqslant x(t)$ for every $t \in [0,T)$. Indeed, for i such that $y_i > 0$ we have $\inf_{t\in[0,T)} x_i(t) > 0$, so $\varepsilon_i \cdot y_i \leqslant x_i(t)$ for some $\varepsilon_i > 0$ and all $t \in [0,T)$, then take $\varepsilon = \min\{\varepsilon_i : i \in \operatorname{supp}(y)\}$.

Fix any $t \in [0,T)$ and consider $w \in \mathbb{R}^n$ defined as $w_i = \frac{x_i(t)}{c_i}$. Let us bound the norm of $A^\top L^{-1} b$:

$$\begin{aligned}
\|A^\top L^{-1}b\|_\infty &= \|A^\top L^{-1}\left(\sum_{j=1}^n y_j a_j\right)\|_\infty \\
&\leqslant \sum_{j=1}^n y_j \|A^\top L^{-1} a_j\|_\infty \overset{\text{Lemma 4.1}}{\leqslant} \sum_{j=1}^n y_j \frac{\alpha}{w_j} \\
&\leqslant \sum_{j=1}^n \frac{x_j(t)}{\varepsilon}\frac{\alpha}{w_j} = \frac{\alpha}{\varepsilon}\sum_{j=1}^n c_j.
\end{aligned}$$

Let us denote $M = \frac{\alpha}{\varepsilon}\sum_{j=1}^n c_j$. Pick now any $i \in [n]$, we have

$$\dot{x}_i = q_i - x_i = \frac{x_i}{c_i}p_i - x_i \geqslant x_i\left(-\frac{M}{c_i} - 1\right).$$

Hence, from the Gronwall Lemma we obtain

$$x_i(t) \geqslant x_i(0)e^{t\left(-\frac{M}{c_i}-1\right)}.$$

In particular, $x_i(t)$ is lower bounded by a positive constant over the whole interval $[0, T)$. The lemma follows. □

We can now conclude Theorem 4.1.

Proof of Theorem 4.1:
We use Theorem 4.9. One needs to observe that the function $P(x) = q - x$ is in fact of class $\mathcal{C}^1$. Even more is true: $P(x)$ is a rational function in the region Ω, this can be seen using the Cramer's rule. The remaining assumptions are satisfied by Lemmas 4.3 and 4.4. □

4.3.4 Asymptotic Behavior of Physarum Trajectories

In the previous section we have established the existence of solutions to the Physarum dynamics. We know that for every starting point $s > 0$, there is a solution $x : [0, \infty) \to \mathbb{R}^n_{>0}$ to the Physarum dynamics with $x(0) = s$. We would like to study the behavior of $x(t)$ when $t \to \infty$. Let us fix the starting point $x(0)$ and pick $M_x > 0$ such that

$$M_x^{-1} \leqslant x_i(0) \leqslant M_x \qquad \text{for every } i \in [n].$$

Furthermore, pick $x^\star$ to be any optimal basic solution (i.e. $x^\star$ is a vertex of the feasible region) to (4.1). Our first result will be that $c^\top x(t)$ approaches $c^\top x^\star$ – the optimal value of the linear program (4.1). Before we proceed with the proof, let us establish some useful properties of the feasible region.

The Feasible Region

Let us consider $P = \{x : Ax = b, x \geqslant 0\}$ – the feasible region of (4.1). Note that P does not contain a line, so it can be expressed as the Minkowski sum $P = H + K$, where H is the convex hull of vertices of P and K is a polyhedral cone

$$K = \{r \in \mathbb{R}^n : Ar = 0, r \geqslant 0\}.$$

Let us denote by V the set of vertices of P, so $H = \mathrm{conv}(V)$. Further define R to be the set of vertices of the polytope

$$\left\{r : Ar = 0, \ \sum_{i=1}^{n} r_i = 1, r \geqslant 0\right\}.$$

Then K is the conic hull of R:

$$K = \left\{\sum_{r \in R} \mu_r r : \mu_r \geqslant 0\right\}.$$

Let us first establish the following bounds

Lemma 4.7

Let V and R be as defined above.

1. *Let $v \in V$ and $i \in [n]$ be such that $v_i \neq 0$, then $\mathcal{D}^{-1} \leqslant |v_i| \leqslant \mathcal{D} \cdot \|b\|_1$.*
2. *Let $r \in R$ and $i \in [n]$ be such that $r_i \neq 0$, then $\mathcal{D}^{-1} \leqslant |r_i| \leqslant \mathcal{D}$.*

Proof.
This follows immediately from the Cramer's rule and the characterization of vertices of polyhedra (see Remark 4.1). ☐

Let $x \in P$, since $P = H + K$, x can be expressed as

$$x = \sum_{v \in V} \lambda_v v + \sum_{r \in R} \mu_r r$$

with $\lambda_v, \mu_r \geqslant 0$ for $v \in V, r \in R$ and $\sum_{v \in V} \lambda_v = 1$. Note that the sets V and R might be very large (of size exponential in n). However, by Caratheodory's theorem we may pick the vectors of coefficients λ, μ so that $|\mathrm{supp}(\lambda)| \leqslant n + 1$ and $|\mathrm{supp}(\mu)| \leqslant n + 1$.

The last preliminary fact we would like to state is the following proposition on sensitivity analysis of linear programs. It is very useful to estimate the distance from a polyhedron to a point which satisfies its constraints with some additive error.

Proposition 4.1 (*Sensitivity analysis*)

Suppose that $B \in \mathbb{Z}^{M \times N}$, $g \in \mathbb{R}^N$ and $b', b'' \in \mathbb{R}^M$ are such that both linear programs: $\min\{g^\top x : Bx \leqslant b'\}$ and $\min\{g^\top x : Bx \leqslant b''\}$ have finite optimal values, then

$$\left|\min\{g^\top x : Bx \leqslant b'\} - \min\{g^\top x : Bx \leqslant b''\}\right| \leqslant N\mathcal{D}_B \|g\|_1 \cdot \|b' - b''\|_\infty.$$

Where $\mathcal{D}_B = \max\{|\det(B')| : B' \text{ a square submatrix of } B\}$.

A proof of the above can be found in any standard textbook on linear programming, see e.g. [Sch86].

Convergence to the Optimal Value

In this section we establish the following theorem.

Theorem 4.10

Suppose that $x : [0, \infty) \to \Omega$ is any solution to the Physarum dynamics. Then, for some $R, \nu > 0$ depending only on $A, b, c, x(0)$, we have

$$\left|c^\top x(t) - c^\top x^\star\right| \leqslant R \cdot e^{-\nu t}.$$

Quantitatively, one can take $\nu = \mathcal{D}^{-3}$ and $R = \exp(8\mathcal{D}^2 \cdot C_s \cdot \|b\|_1) \cdot (n + M_x)^2$.

Let us begin by splitting the set V into two subsets $V_O = \{v \in V : c^\top v = c^\top x^\star\}$ and $V_N = V \backslash V_O$. Then V_O is basically the set of optimal vertices. The following bound will be very useful in the proof of Theorem 4.10.

Lemma 4.8

Let $v \in V_N$, then $c^\top v - c^\top x^\star \geqslant \mathcal{D}^{-2}$.

Proof.
Since v is a vertex of P, by Cramer's rule it can be expressed as $v = (\frac{z_1}{d}, \frac{z_2}{d}, \ldots, \frac{z_n}{d})^\top$. Where $d, z_1, z_2, \ldots, z_n \in \mathbb{Z}$ and $1 \leqslant d \leqslant \mathcal{D}$. Hence $c^\top v$ is a number of the form $\frac{z}{d}$ for some integer z. Similarly $c^\top x^\star = \frac{z'}{d'}$ with $z', d' \in \mathbb{Z}$ and $1 \leqslant d' \leqslant \mathcal{D}$. We get

$$c^\top v - c^\top x^\star = \frac{d'z - z'd}{dd'} \geqslant \frac{1}{\mathcal{D}^2}.$$

□

The next lemma says that $x(t)$ becomes exponentially close to the feasible region P as $t \to \infty$.

Lemma 4.9

For every $t \geqslant 0$ there exists $y(t) \in P$ such that

$$\|x(t) - y(t)\|_\infty < e^{-t} \cdot 3n^2 \cdot \mathcal{D} \cdot M_x.$$

Proof.
Start with the Physarum dynamics $\dot{x} = q - x$ and multiply both sides by e^t. We get $\frac{d}{dt}(x(t)e^t) = e^t q(t)$. Take integrals of both sides to obtain

$$x(t) = x(0)e^{-t} + (1 - e^{-t}) \int_0^t q(s) \frac{e^{s-t}}{1 - e^{-t}} ds.$$

Observe that $z(t) = (1 - e^{-t}) \int_0^t q(s) \frac{e^{s-t}}{1-e^{-t}} ds$ is a convex combination of $q(s)$ (which satisfy $Aq(s) = b$ for every s), hence we conclude that $z(t) = x(t) - x(0)e^{-t}$ satisfies $Az(t) = b$. Note also that $z(t) \geqslant x(0)e^{-t}$, so $z(t)$ approaches P with $t \to \infty$. We would like to estimate the distance from $z(t)$ to the closest point in P. As a tool for that we would like to use Proposition 4.1. We will use variables $y \in \mathbb{R}^n$ and a new variable $d \in \mathbb{R}$. Let us write down two linear programs:

$$\begin{array}{rl} \text{min.} & d \\ \text{s.t.} & Ay = b \\ & \|y - z(t)\|_\infty \leqslant d \\ & y \geqslant 0 \end{array} \qquad\qquad \begin{array}{rl} \text{min.} & d \\ \text{s.t.} & Ay = b \\ & \|y - z(t)\|_\infty \leqslant d \\ & y \geqslant -x(0)e^{-t}. \end{array}$$

The minimum value of the first linear program is equal to $\text{dist}^\infty(z(t), P)$: the minimum ℓ^∞–distance from $z(t)$ to some point $y \in P$. The second one has optimal value 0, which is demonstrated by taking $y = z(t)$. To apply Proposition 4.1 we need too transform the constraints into inequality form, but this is done in a standard manner. We obtain

$$\text{dist}^\infty(z(t), P) \leqslant (n+1) \cdot (2n) \cdot \mathcal{D} \cdot \|x(0)e^{-t}\|_\infty.$$

In the above, $(2n) \cdot \mathcal{D}$ is the bound on $\mathcal{D}_B$, where B is the matrix obtained after transforming the constraints into the inequality form. By taking $y(t)$ which attains this minimum distance, we obtain the desired point in P:

$$\begin{aligned} \|x(t) - y(t)\|_\infty &\leqslant \|x(t) - z(t)\|_\infty + \|z(t) - y(t)\|_\infty \\ &\leqslant \|x(0)e^{-t}\|_\infty + (n+1) \cdot (2n) \cdot \mathcal{D} \cdot \|x(0)e^{-t}\|_\infty \\ &\leqslant 3n^2 \cdot \mathcal{D} \cdot \|x(0)e^{-t}\|_\infty. \end{aligned}$$

□

By the discussion at the beginning of the section we know that for every t, $y(t)$ (from the above lemma) can be decomposed as

$$y(t) = \sum_{v \in V} \lambda_v(t) v + \sum_{r \in R} \mu_r(t) r$$

for some $\lambda_v(t), \mu_r(t) \geqslant 0$ such that $\sum_{v \in V} \lambda_v(t) = 1$. Of course this decomposition is not unique. We choose it arbitrarily, but as noted previously it can be done in such a way that most of the coefficients are zero, i.e., $|\text{supp}(\lambda(t))| \leqslant n+1$ and $|\text{supp}(\mu(t))| \leqslant n+1$. Our strategy is to show that $\mu_r(t) \to 0$ for all $r \in R$ and $\lambda_v(t) \to 0$ for all $v \in V_N$. This will imply that $y(t)$ (hence also $x(t)$) tends to the optimal region. Towards this let us prove:

Lemma 4.10

The following bounds hold with some $Q, \nu > 0$ depending only on $A, b, c, x(0)$:

1. *For every $r \in R$, $\min\{x_i(t) : i \in \text{supp}(r)\} \leqslant Q \cdot \exp(-\nu t)$.*
2. *For every $v \in V_N$, $\min\{x_i(t) : i \in \text{supp}(v)\} \leqslant Q \cdot \exp(-\nu t)$.*

Quantitatively, one can take $\nu = \mathcal{D}^{-3}$ and $Q = \exp\left(4\mathcal{D}^2 C_s \cdot \|b\|_1\right) \cdot (n + M_x)$.

Proof.
Let us start with the second part. Pick any $v \in V_N$ and $x^\star \in V_O$, consider the following potential function

$$f(t) \stackrel{\text{def}}{=} \sum_{i=1}^n c_i v_i \ln x_i(t) - \sum_{i=1}^n c_i x_i^\star \ln x_i(t)$$

and take its derivative

$$\begin{aligned}\frac{d}{dt}f(t) &= \sum_{i=1}^{n} c_i v_i \frac{\dot{x}_i(t)}{x_i(t)} - \sum_{i=1}^{n} c_i x_i^\star \frac{\dot{x}_i(t)}{x_i(t)} \\ &= \sum_{i=1}^{n} c_i v_i \left(\frac{a_i^\top p(t)}{c_i} - 1\right) - \sum_{i=1}^{n} c_i x_i^\star \left(\frac{a_i^\top p(t)}{c_i} - 1\right) \\ &= \sum_{i=1}^{n} (v_i - x_i^\star)\, a_i^\top p(t) + \left(c^\top x^\star - c^\top v\right).\end{aligned}$$

Observe that, since $Av = Ax^\star = b$, we have

$$\sum_{i=1}^{n} (v_i - x_i^\star)\, a_i^\top p(t) = (v - x^\star)A^\top p(t) = (b^\top - b^\top)p(t) = 0.$$

Hence, the derivate takes a particularly simple form

$$\frac{d}{dt}f(t) = c^\top x^\star - c^\top v.$$

By Lemma 4.8 we know that $\frac{d}{dt}f(t) \leqslant -\varepsilon$ for $\varepsilon = \mathcal{D}^{-2}$. This implies the following bound on $f(t)$:

$$f(t) \leqslant f(0) - \varepsilon \cdot t.$$

Hence

$$\sum_{i=1}^{n} c_i v_i \ln x_i(t) \leqslant f(0) + \sum_{i=1}^{n} c_i x_i^\star \ln x_i(t) - \varepsilon \cdot t.$$

Using Corollary 4.5 and Lemma 4.7 we obtain

$$\sum_{i=1}^{n} c_i x_i^\star \ln x_i(t) \leqslant \sum_{i=1}^{n} c_i \mathcal{D} \cdot \|b\|_1 \ln(\max(\mathcal{D}^2 n\|b\|_1, M_x)) \leqslant C_s \mathcal{D} \cdot \|b\|_1 \cdot \ln(\mathcal{D}^2 n \|b\|_1 + M_x).$$

Denote by M the expression on the right-hand side. Similarly we can bound $f(0) \leqslant 2M$ and obtain

$$\sum_{i=1}^{n} c_i v_i \ln x_i(t) \leqslant 3M - \varepsilon t.$$

Suppose without loss of generality that $x_1(t) = \min\{x_i(t) : i \in \operatorname{supp}(v)\}$, then

$$\ln x_1(t) \cdot \sum_{i \in \operatorname{supp}(v)} c_i v_i \leqslant \sum_{i=1}^{n} c_i v_i \ln x_i(t) \leqslant 3M - \varepsilon t.$$

It remains to bound $\sum_{i \in \operatorname{supp}(v)} c_i v_i \geqslant \mathcal{D}^{-1}$ to get

$$\min\{x_i(t) : i \in \operatorname{supp}(v)\} \leqslant \exp(3M\mathcal{D}) \cdot \exp(-\mathcal{D}^{-3} t).$$

Part (1) follows via an analogous reasoning, using the potential function $\sum_{i=1}^{n} c_i r_i \ln x_i(t)$. □

We are now ready to present a complete proof of Theorem 4.10

Proof of Theorem 4.10:
Let us first pick $r \in R$, we show that $\mu_r(t) \to 0$ exponentially fast. We have for every $i \in \operatorname{supp}(r)$,

$$\mu_r(t) r_i \leqslant y_i(t) \leqslant x_i(t) + Me^{-t}$$

with M as in Lemma 4.9. Hence, by Lemma 4.10, there exists $i \in \operatorname{supp}(r)$ such that

$$\mu_r(t) r_i \leqslant Qe^{-\nu t} + Me^{-t}.$$

Using the bound $r_i \geqslant \mathcal{D}^{-1}$ from Lemma 4.7 we obtain

$$\mu_r(t) \leqslant \mathcal{D}Qe^{-\nu t} + \mathcal{D}Me^{-t} \leqslant 2\mathcal{D}Qe^{-\nu t}.$$

By a similar argument, we get for every $v \in V_N$,

$$\lambda_v(t) \leqslant 2\mathcal{D}Qe^{-\nu t}.$$

Having this, let us consider the difference $c^\top y(t) - c^\top x^\star$. We know that

$$c^\top \left(\sum_{v \in V_O} \lambda_v(t) v \right) = \sum_{v \in V_O} \lambda_v(t) c^\top v = (c^\top x^\star) \sum_{v \in V_O} \lambda_v(t).$$

Hence, we arrive at

$$c^\top y(t) - c^\top x^\star \leqslant c^\top \left(\sum_{v \in V_N} \lambda_v(t) v + \sum_{r \in R} \mu_r(t) r \right).$$

Recall that $\lambda(t)$ and $\mu(t)$ were chosen in such a way that all but at most $n+1$ of their entries are zero, therefore

$$\| \sum_{v \in V_N} \lambda_v(t) v \|_\infty \leqslant \sum_{v \in V_N} \lambda_v(t) \|v\|_\infty \leqslant (n+1) \cdot 2\mathcal{D}Qe^{-\nu t} \cdot \mathcal{D} \cdot \|b\|_1.$$

A similar bound holds for $\sum_{r \in R} \mu_r(t) r$. Altogether, we obtain the following bound

$$c^\top y(t) - c^\top x^\star \leqslant 4(n+1) C_s \mathcal{D}^2 \|b\|_1 Qe^{-\nu t}.$$

To conclude, let us relate $c^\top y(t)$ to $c^\top x(t)$. By Lemma 4.9 we have

$$\left| c^\top y(t) - c^\top x(t) \right| \leqslant C_s \cdot Me^{-t}.$$

This finally yields the claimed bound:

$$\left| c^\top x(t) - c^\top x^\star \right| \leqslant 4(n+1) C_s \mathcal{D}^2 \|b\|_1 Qe^{-\nu t} + C_s \cdot Me^{-t} \leqslant Q^2 e^{-\nu t}.$$

$\square$

Convergence to a Limit

From the previous section we know that for every solution $x : [0, \infty) \to \Omega$, $\left|c^\top x(t) - c^\top x^\star\right| \to 0$ exponentially fast with $t \to \infty$. One can use this fact to prove that whenever (4.1) has a unique solution, then $x(t)$ has a limit x^∞ and x^∞ is this unique optimal solution to (4.1). However, if we do not assume uniqueness, the fact that $x(t)$ has a limit is no more obvious. We are aiming to prove it in the current subsection. Let us now formally state the theorem.

Theorem 4.11

Suppose $x : [0, \infty) \to \Omega$ is a solution to the Physarum dynamics (4.2)*. Then there exists a limit $x^\infty = \lim_{t\to\infty} x(t)$. Furthermore x^∞ is an optimal solution to the linear program* (4.1)*.*

In this subsection the norm symbol $\|\cdot\|$ should be understood as the infinity norm $\|\cdot\|_\infty$. Unlike in the previous proofs we will not keep track of constants and hide them under the big O notation. By a constant we mean any quantity which solely depends on A, b, c, n, m and the fixed starting point $x(0)$ under consideration.

Let us start by giving a simple lemma which provides a sufficient condition for existence of a limit.

Lemma 4.11

Suppose $x : [0, \infty) \to \Omega$ is a differentiable function. If the integral

$$\int_0^\infty \|\dot{x}(t)\| dt$$

is finite, then there exists a limit $x^\infty = \lim_{t\to\infty} x(t)$.

Proof.
Observe that $x(t) = x(0) + \int_0^t \dot{x}(s)ds$. By going with $t \to \infty$, we obtain

$$\lim_{t\to\infty} x(t) = x(0) + \int_0^\infty \dot{x}(s)ds.$$

Since the integral on the right-hand side exists (because it is absolutely convergent by the assumption), we conclude existence of the limit. $\square$

In our case $\dot{x} = q - x$. Our proof strategy is to show that $\|q(t) - x(t)\| = O(e^{-\varepsilon t})$ for some constant $\varepsilon > 0$, then the assumption of Lemma 4.11 is satisfied.

The following lemma is a consequence of Theorem 4.10:

Lemma 4.12

For every $t \geqslant 0$ there exists $x^\star(t)$ – an optimal solution to (4.1) *such that $\|x(t) - x^\star(t)\| = O(e^{-\varepsilon t})$ for some $\varepsilon > 0$.*

Note that the above lemma does not imply that $x(t)$ has a limit. It could potentially happen that $x(t)$ oscillates over the optimal region without approaching a single point. To prove it, we need to understand the behavior of Physarum with respect to the structure of the optimal set.

From the general theory of linear programming (see e.g. [Wri97]) we know that $[n]$ can be decomposed into $[n] = J \cup N$ with $J \cap N = \emptyset$ such that every optimal solution to (4.1) is supported on J and every feasible solution supported on J is optimal. We will denote by x_J

the part of a vector x corresponding to indices $j \in J$, similarly x_N. Let us now establish two important lemmas, which can be proved using techniques developed so far.

Lemma 4.13

If N is the above define set of indices then $\|x_N(t)\| = O(e^{-\varepsilon t})$ for some $\varepsilon > 0$.

Proof.
Follows directly from 4.12. For every t, $x^\star_N(t) = 0$, because $x^\star(t)$ is optimal, hence we get the claim. □

Lemma 4.14

There exists a constant $\varepsilon > 0$ such that $x_j(t) > \varepsilon$ for every $j \in J$ and $t \in [0, \infty)$.

Proof.
Recall that the set of optimal vertices is denoted by V_O. Note that for every $j \in J$, there is a vertex $v \in V_O$ such that $v_j = \Omega(1)$. Hence, it is enough to show that for every $v \in V_O$, the function

$$f_v(t) = \sum_{i=1}^{n} c_i v_i \ln x_i(t)$$

is bounded from below by a universal constant (this is true because of the fact that $\|x(t)\|$ is uniformly bounded, by Corollary 4.5). Note that, by a calculation analogous to that performed in the proof of 4.10, we obtain

$$\frac{d}{dt}(f_v(t) - f_u(t)) = 0$$

for every $v, u \in V_O$, which implies that all f_v differ only by a constant. Hence either all of them are lower-bounded by a universal constant or none of them.

By Lemma 4.12 we know that for every t, $x(t)$ is exponentially close to some $x^\star(t) \in \operatorname{conv}(V_O)$. This implies in particular that for big enough t and for some universal constant $\delta > 0$ there always exists a $v(t) \in V_O$ such that $\delta \cdot v(t) \leqslant x(t)$. Hence

$$f_{v(t)}(t) \geqslant \sum_{i=1}^{n} c_i v_i(t) \ln(\delta \cdot v_i(t)) = \Omega(1).$$

Note that the set V_O is finite, hence we get the bound $\max_{v \in V_O} f_v(t) = \Omega(1)$, which finishes the proof. □

Lemma 4.15

Suppose $p(t)$ is the vector $L^{-1}b$, computed with respect to $x(t)$. Then $\|A^\top p(t)\|$ is uniformly bounded over $t \in [0, \infty)$.

Proof.
By a reasoning as in the proof of Lemma 4.14 we get that for some $v \in V$ and a universal constant $\delta > 0$ we have $\delta \cdot v \leqslant x(t)$ for every $t \in [0, \infty)$. Hence we get

$$\|A^\top p(t)\| = \|A^\top L^{-1} b\| = \|A^\top L^{-1} \left(\sum_{i=1}^{n} a_i v_i\right)\| \leqslant \sum_{i=1}^{n} v_i \|A^\top L^{-1} a_i\|.$$

By Lemma 4.1, we have that $\|A^\top L^{-1}a_i\| \leqslant \frac{\alpha c_i}{x_i}$, for some universal constant α. Hence, we get

$$\sum_{i=1}^{n} v_i \|A^\top L^{-1}a_i\| \leqslant \sum_{i=1}^{n} v_i \frac{\alpha c_i}{x_i} \leqslant \sum_{i=1}^{n} (\delta^{-1}x_i)\frac{\alpha c_i}{x_i} = \alpha\delta^{-1}\sum_{i=1}^{n} c_i = O(1).$$

The lemma follows. □

We now present the main technical lemma required to prove existence of a limit. Intuitively, it shows that the subvector of $A^\top p$ corresponding to J behaves continuously assuming x is close to the optimal set.

Lemma 4.16

Suppose $\varepsilon > 0$ is small enough and $M, d > 0$. Pick any $x > 0$ such that $\|x_N\| < \varepsilon$ and $\|x_J - y_J\| < \varepsilon$ for some optimal solution y such that $y_j > d$ for all $j \in J$. Assume $\|x\| \leqslant M$ and $\|A^\top p\| \leqslant M$. There exists a constant M' depending only on A, b, c, d, n such that $\|q - x\| < M' \cdot \varepsilon$.

Proof.
We will denote the columns of A corresponding to J by A_J, similarly for A_N. We have $(AWA^\top)p = b$, which can be rewritten as

$$A_J W_J A_J^\top p + A_N W_N A_N^\top p = b.$$

where $W_J = \text{Diag}\,(w_J)$ and $W_N = \text{Diag}\,(w_N)$. Since $\|A_N^\top p\| = O(1)$ and $\|A_N W_N\| = \|A_N C^{-1} X_N\| = O(\varepsilon)$ we have $\|A_N W_N A_N^\top p\| = O(\varepsilon)$. Moreover, we can write $x_J = y_J + \tau_J$, where $\|\tau_J\| < \varepsilon$. This gives a decomposition of W_J into $\overline{W_J} = \text{Diag}\left(C_J^{-1}y_J\right)$ and $\widetilde{W_J} = \text{Diag}\left(C_J^{-1}\tau_J\right)$ such that $\|A_J\widetilde{W_J}A_J^\top p\| = O(\varepsilon)$. Consequently,

$$\|A_J\overline{W_J}A_J^\top p - b\| \leqslant \|A_J W_J A_J^\top p - b\| + \|\widetilde{W_J}A_J^\top p\| = O(\varepsilon).$$

Pick $\bar{p}$: any optimal solution to the dual of the linear program (4.1). From complementary slackness we have $Y(A^\top\bar{p} - c) = 0$. In particular $A_J^\top\bar{p} = c_J$. We obtain:

$$A_J\overline{W_J}A_J^\top\bar{p} = A_J\overline{W_J}c_J = A_J\text{Diag}\left(C_J^{-1}y_J\right)c_J = A_J y_J = b.$$

Hence

$$\|A_J\overline{W_J}A_J^\top(p - \bar{p})\| = \|A_J\overline{W_J}A_J^\top p - A_J\overline{W_J}A_J^\top\bar{p}\| = \|A_J\overline{W_J}A_J^\top p - b\| = O(\varepsilon).$$

We want to show that $\|A_J^\top(p - \bar{p})\| = O(\varepsilon)$.
Denote $K = A_J\overline{W_J}A_J^\top$, it is a symmetric PSD matrix. Moreover, the kernels of $A_J^\top$ and K coincide, since $y_J > 0$. Pick a vector $v \in \mathbb{R}^m$ so that $A_J^\top(p - \bar{p}) = A_J^\top v$ and v is orthogonal to the kernel of $A_J^\top$. (In other words v is the orthogonal projection of $p - \bar{p}$ onto the orthogonal complement of the kernel of $A_J^\top$.) Using the fact that $\frac{y_i}{c_i} > \frac{d}{c_i} \geqslant d'$ for $i \in J$ and some absolute positive constant d', we obtain

$$d' A_J A_J^\top = \sum_{j\in J} d' a_j a_j^\top \preceq \sum_{j\in J}\frac{y_j}{c_j}a_j a_j^\top = A_J\overline{W_J}A_J^\top = K$$

where $\preceq$ denotes the PSD ordering. This implies in particular that $d'\lambda^+ \leqslant \lambda_K^+$, where λ^+, λ_K^+ are the smallest positive eigenvalues of $A_J A_J^\top$ and K respectively. Since v is orthogonal to the kernel of K and K is PSD we have

$$\|Kv\|_2 \geqslant \lambda_K^+ \|v\|_2 \geqslant d'\lambda^+ \|v\|_2.$$

Observe that λ^+ depends solely on A, b, c, hence

$$\|v\| \leqslant \|v\|_2 \leqslant \frac{1}{d'\lambda^+}\|Kv\|_2 = \frac{1}{d'\lambda^+}\|K(p-\bar{p})\|_2 = O(\varepsilon).$$

In consequence, $\|A_J^\top(p-\bar{p})\| = \|A_J^\top v\| = O(\varepsilon)$ and hence $\|\overline{W_J} A_J^\top(p-\bar{p})\| = O(\varepsilon)$, because $\|y\| = O(1)$ (the set of optimal solutions is bounded). Finally,

$$\|W_J A_J^\top p - \overline{W_J} A_J^\top \bar{p}\| \leqslant \|W_J A_J^\top p - \overline{W_J} A_J^\top p\| + \|\overline{W_J} A_J^\top p - \overline{W_J} A_J^\top \bar{p}\| = O(\varepsilon) + O(\varepsilon) = O(\varepsilon).$$

But $W_J A_J^\top p = q_J$ and $\overline{W_J} A_J^\top \bar{p} = y_J$, which means

$$\|q_J - x_J\| \leqslant \|q_J - y_J\| + \|y_J - x_J\| = O(\varepsilon) + O(\varepsilon) = O(\varepsilon).$$

To conclude, note that

$$\|q_N - x_N\| \leqslant \|x_N\| + \|q_N\| = \|x_N\| + \|W_N A_N^\top p\| = O(\varepsilon)$$

and hence trivially:

$$\|q - x\| = \max(\|q_N - x_N\|, \|q_J - x_J\|) = O(\varepsilon).$$

□

Let us now conclude Theorem 4.11 from Lemma 4.16.

Proof of Theorem 4.11:
By Lemma 4.11, it remains to show that $\|q(t) - x(t)\| = O(e^{-\delta t})$ for some $\delta > 0$. By Corollary 4.5 and Lemma 4.15 we obtain a uniform bound M on both $\|x(t)\|$ and $\|A^\top p(t)\|$ over $t \in [0, \infty)$. From Lemma 4.14 we obtain some $d > 0$ such that $x_j(t) > 2d$ for $j \in J$ and $t \in [0, \infty)$.

Let us now pick a large t. We take $x = x(t)$ and $y = x^\star(t)$ from Lemma 4.12. We know that $x_j > 2d$ for $j \in J$, y is exponentially close to x, hence $y_j > d$ for $j \in J$. Moreover, we can pick $\varepsilon = O(e^{-\delta t})$ such that $\|x_N\| < \varepsilon$ (by Lemma 4.13) and $\|y - x\| < \varepsilon$. Hence the assumptions of Lemma 4.16 are satisfied and we may conclude that $\|q(t) - x(t)\| < M'\varepsilon = O(e^{-\delta t})$. The theorem follows. □

4.3.5 Discretization for Linear Programming

In this section we study the efficiency of the natural discretization of the Physarum dynamics. It is defined with respect to the step length $0 < h < 1$ and a starting point $x(0) > 0$. In this section we will always assume that the starting point was chosen from the feasible region, i.e. $Ax(0) = b$. The discrete dynamics is as follows:

$$x(k+1) = (1-h)x(k) + hq(k) \tag{4.6}$$

with the usual meaning of $q(k)$. For the above dynamics to be well defined we need to make sure that $x(k)$ stays positive for all $k \geqslant 0$. Note that

$$x_i(k+1) = x_i(k)\left(1 - h\left(\frac{a_i^\top p(k)}{c_i} - 1\right)\right),$$

hence we need to make sure that $\left(\frac{a_i^\top p(k)}{c_i} - 1\right) < \frac{1}{h}$. Corollary 4.1 tells us that

$$P_{\max} \overset{\text{def}}{=} \max\left\{\left|\frac{a_i^\top p}{c_i} - 1\right| : Ax = b, x \geqslant 0\right\} \leqslant C_s \mathcal{D} + 1.$$

This means that it is enough to set $h < P_{\max}^{-1}$ (and we do so in the following theorem).

Theorem 4.12

Let $0 < \varepsilon < 1/2$, $h \leqslant P_{\max}^{-2} \cdot \varepsilon/6$. Suppose we initialize the Physarum algorithm (4.6) with $x(0)$, s.t. $Ax(0) = b$ and $M_x^{-1} \leqslant x_i(0) \leqslant M_x$ for every $i \in [n]$ and some $M_x \geqslant 1$. Assume additionally that $c^\top x(0) \leqslant M \cdot \mathrm{opt}$. Then after $k \overset{\text{def}}{=} O\left(\frac{\ln M}{\varepsilon^2 h^2} + \frac{\ln M_x}{\varepsilon h}\right)$ steps $x(k)$ is a feasible solution with $\mathrm{opt} \leqslant c^\top x(k) \leqslant (1+\varepsilon)\mathrm{opt}$.

To prove Theorem 4.12 we analyze the following potential function

$$\phi(k) \overset{\text{def}}{=} 4 \ln \mathcal{V}(k) - \frac{\varepsilon h}{\mathrm{opt}} \mathcal{B}(k).$$

The intuitive meaning of ϕ is as follows: we know that $\mathcal{V}(k)$ is non-increasing with k, so $\ln \mathcal{V}(k)$ is non-increasing as well, however we may not guarantee a big drop of $\mathcal{V}(k)$ in every step, this is why we have the second term; $\mathcal{B}(k)$ gets bigger at such steps. The crucial lemma we would like to show asserts that ϕ drops significantly at every step

Lemma 4.17 (*Potential drop*)

For every k with $\mathcal{V}(k) > (1+\varepsilon)\mathrm{opt}$ we have $\Delta\phi(k) \leqslant -\frac{h^2\varepsilon^2}{6}$.

Before proving the lemma we first conclude Theorem 4.12 from it.

Proof of Theorem 4.12 assuming Lemma 4.17:
We will first estimate the value of $\phi(0)$. Observe that

$$\ln(\mathcal{V}(0)) = \ln(c^\top x(0)) \leqslant \ln(M \cdot \mathrm{opt}).$$

The value $\mathcal{B}(0)$ can be bounded as follows

$$\sum_{i=1}^n x_i^\star c_i \ln x_i(0) \geqslant x_i^\star c_i \ln M_x^{-1} = \mathrm{opt} \cdot \ln M_x^{-1}.$$

Hence, we obtain

$$\phi(0) \leqslant 4\ln(M \cdot \mathrm{opt}) + \varepsilon h \ln M_x.$$

Let us now fix k and provide a lower bound on the value of $\phi(k)$. We know that $\mathcal{V}(k) \geqslant \mathrm{opt}$ for every k, for $\mathcal{B}$ we have

$$\mathcal{B}(k) = \sum_{i=1}^{n} x_i^\star c_i \ln x_i(k) \leqslant \ln(M_x) \sum_{i=1}^{n} x_i^\star c_i = \mathrm{opt} \cdot \ln(M_x),$$

and thus

$$\phi(k) \geqslant 4\ln(\mathrm{opt}) - \varepsilon h \ln(M_x).$$

Putting together the above estimates on $\phi(0), \phi(k)$ and Lemma 4.17, we obtain that

$$k\frac{h^2\varepsilon^2}{6} \leqslant \phi(0) - \phi(k) \leqslant 4\ln M + \varepsilon h(\ln(M_x) + \ln(M_x)).$$

Simplifying,

$$k = O\left(\frac{\ln M}{\varepsilon^2 h^2} + \frac{\ln M_x}{\varepsilon h}\right).$$

□

We split the Lemma 4.17 into two cases and deal with them separately. They are expressed in the following facts. We use below $\mathcal{E}(k)$ to denote $q(k)^\top W^{-1} q(k)$.

Fact 4.1

If $\frac{\mathcal{E}(k)}{\mathcal{V}(k)} < (1 - \varepsilon/3)$ *then* $\Delta\phi(k) \leqslant -\frac{h^2\varepsilon^2}{6}$.

Fact 4.2

If $\mathcal{E}(k) > (1 + \varepsilon/3)\mathrm{opt}$ *then* $\Delta\phi(k) \leqslant -\frac{h^2\varepsilon^2}{6}$.

Before we proceed with the proof of facts, let us state three simple inequalities which we are going to use:

$$\ln(1+\alpha) \leqslant \alpha \qquad \text{for every } \alpha \in \mathbb{R} \tag{4.7}$$

$$\ln(1+\alpha) \geqslant \alpha - \alpha^2 \qquad \text{for every } \alpha \in \left[-\frac{1}{2}, \frac{1}{2}\right] \tag{4.8}$$

$$\ln(1-\alpha) \geqslant -2\alpha \qquad \text{for every } \alpha \in \left[0, \frac{1}{2}\right] \tag{4.9}$$

All can be proved by simple calculus. Another useful fact is the following identity.

Fact 4.3

Let $x \in \mathbb{R}^n_{>0}$ *and* $p = (AWA^\top)^{-1}b$ *be given. Suppose* $y \in \mathbb{R}^n$ *satisfies* $Ay = b$, *then*

$$\sum_{i=1}^{n} y_i a_i^\top p = q^\top W^{-1} q,$$

where $q = WA^\top p$.

Proof of Fact 4.1:
We will show that $\ln\mathcal{V}$ drops and $\mathcal{B}$ may increase only a little. Let us first look at $\ln\mathcal{V}(k+1) -$

$\ln \mathcal{V}(k)$. We have

$$\mathcal{V}(k+1) = \sum_{i=1}^{n} c^\top x(k+1) = (1-h)c^\top x(k) + h\sum_{i=1}^{n} x(k)a_i^\top p(k) = (1-h)\mathcal{V}(k) + h\mathcal{E}(k).$$

The last inequality follows from 4.3. Further,

$$\begin{aligned}\frac{\mathcal{V}(k+1)}{\mathcal{V}(k)} &= \left(1 + h\left(\frac{\mathcal{E}(k)}{\mathcal{V}(k)} - 1\right)\right)\\ &< \left(1 + h\left(\left(1 - \frac{\varepsilon}{3}\right) - 1\right)\right)\\ &= 1 - h\frac{\varepsilon}{3}.\end{aligned}$$

We obtain

$$\ln \mathcal{V}(k+1) - \ln \mathcal{V}(k) = \ln \frac{\mathcal{V}(k+1)}{\mathcal{V}(k)} \overset{(4.7)}{\leqslant} -\frac{h\varepsilon}{3}.$$

Consider now $\mathcal{B}(k+1) - \mathcal{B}(k)$:

$$\begin{aligned}\mathcal{B}(k+1) - \mathcal{B}(k) &= \sum_{i=1}^{n} x_i^\star c_i \ln \frac{x_i(k+1)}{x_i(k)}\\ &= \sum_{i=1}^{n} x_i^\star c_i \ln\left(1 + h\left(\frac{a_i^\top p(k)}{c_i} - 1\right)\right)\\ &\geqslant \sum_{i=1}^{n} x_i^\star c_i \ln\left(1 - hP_{\max}\right)\\ &\overset{(4.9)}{\geqslant} -2h(n^2C+1)\sum_{i=1}^{n} x_e^\star c_e\\ &= -2hP_{\max}\cdot \mathrm{opt}\\ &\geqslant -\mathrm{opt}.\end{aligned}$$

The last inequality follows from the definition of h. Putting these two pieces together yields

$$\phi(k+1) - \phi(k) \leqslant -\frac{4h\varepsilon}{3} + h\varepsilon = -\frac{h\varepsilon}{3} \leqslant -\frac{h^2\varepsilon^2}{6}.$$

□

Proof of Fact 4.2:
We know that $\mathcal{V}(k)$ is non-increasing with k, it remains to show that $\mathcal{B}$ will increase by a considerable amount. As in the proof of Fact 4.1 we obtain

$$\mathcal{B}(k+1) - \mathcal{B}(k) = \sum_{i=1}^{n} x_i^\star c_i \ln\left(1 + h\left(\frac{a_i^\top p(k)}{c_i} - 1\right)\right).$$

By the choice of h,

$$\left|h\left(\frac{q_e(k)-x_e(k)}{x_e(k)}\right)\right| = |h|\left|\frac{a_i^\top p(k)}{c_e}-1\right| \leqslant h\cdot P_{\max} \leqslant \frac{1}{2}.$$

Hence, we can apply the inequality 4.8, by which we get

$$\begin{aligned}
\mathcal{B}(k+1)-\mathcal{B}(k) &\overset{(4.8)}{\geqslant} \sum_{i=1}^n x_i^\star c_i h\left(\frac{a_i^\top p(k)}{c_i}-1\right) - \sum_{i=1}^n x_i^\star c_i h^2\left(\frac{a_i^\top p(k)}{c_i}-1\right)^2 \\
&\geqslant h\sum_{i=1}^n x_i^\star a_i^\top p(k) - h\cdot\mathrm{opt} - \sum_{i=1}^n x_i^\star c_i\left(hP_{\max}\right)^2 \\
&\overset{\text{Fact4.3}}{=} h\mathcal{E}(k) - h\cdot\mathrm{opt} - h^2\mathrm{opt}P_{\max}^2 \\
&\geqslant h\mathcal{E}(k) - h\cdot\mathrm{opt} - \frac{1}{6}h\varepsilon\mathrm{opt} \\
&\geqslant h\cdot\mathrm{opt}\left(1+\frac{\varepsilon}{3}\right) - h\cdot\mathrm{opt} - \frac{1}{6}h\varepsilon\mathrm{opt} \\
&\geqslant -\frac{1}{6}h\cdot\varepsilon\cdot\mathrm{opt}.
\end{aligned}$$

This implies the following drop of ϕ:

$$\phi(k+1)-\phi(k) \leqslant -\frac{h\varepsilon}{\mathrm{opt}}\left(\mathcal{B}(k+1)-\mathcal{B}(k)\right) \leqslant -\frac{h^2\varepsilon^2}{6}.$$

□

Proof of Lemma 4.17:
If ε is small enough and $\mathcal{V}(k) > (1+\varepsilon)\mathrm{opt}$ then obviously either $\frac{\mathcal{E}(k)}{\mathcal{V}(k)} < (1-\varepsilon/3)$ or $\mathcal{E}(k) > (1+\varepsilon/3)\mathrm{opt}$ and we use Fact 4.1 or Fact 4.2 respectively. This concludes our proof. □

4.3.6 Discretization for the Directed Transshipment Problem

This section provide a proof of Theorem 4.4. We start with some background on combinatorial and electrical flows – these objects are crucial to the understanding of the Physarum dynamics for this case and hence also are essential in the analysis. In the subsequent section we explain our preconditioning scheme and proceed with the convergence proof afterwards. Note that in this section the dimension of the Physarum dynamical system is denoted by m (the number of edges) and the number of linear equation is n (the number of vertices) – opposite as in our treatment of the linear programming case. This change of notation is to preserve consistency with the standard use of n and m in graph theory.

Combinatorial Flows, Electrical flows and their Properties

For a directed graph G and a demand vector $b \in \mathbb{Z}^V$ with $\sum_{v\in V} b_v = 0$, we consider flows $f \in \mathbb{R}^E$ satisfying $Bf = b$, where B is the incidence matrix of G. A flow f is called *nonnegative* if $f \geqslant 0$. If f does not contain a directed cycle in its support, then f is called a *noncircular flow. Basic flows* are defined by vertices of the polytope $\{f : Bf = b, f \geqslant 0\}$.

The mincut-maxflow theorem in this setting says that in a graph (V, E) with edge capacities $x \in \mathbb{R}^E_{\geqslant 0}$, there exists a nonnegative flow f respecting the capacities: $0 \leqslant f \leqslant x$ if and only if for every partition of V into S and $\bar{S}$, we have $\sum_{e \in E(\bar{S}, S)} x_e \geqslant b_S$, where $b_S = \sum_{v \in S} b_v$.

In this chapter we are particularly interested in electrical flows. These are flows defined with respect to a conductance vector $w \in \mathbb{R}^E_{>0}$. Denote $W = \mathrm{Diag}\,(w_1, w_2, \ldots, w_m)$. A flow q is called an electrical flow if it is the unique flow minimizing the quadratic form $\mathcal{E}(q) = q^\top W^{-1} q$. Equivalently, let $p \in \mathbb{R}^V$ be the potential vector obtained as a solution to the following Laplacian system: $Lp = b$ (where $L = BWB^\top$ is the Laplacian matrix). Then q is the flow induced by p, i.e., $q_{uv} = w_{uv}(p_u - p_v)$. The following are some folklore facts about electrical flow, basic and noncircular flows, see [IJNT11, Vis12].

Fact 4.4

Suppose q is the electrical flow in the graph $G = (V, E)$ for vertex demands $b \in \mathbb{Z}^V$ and conductances $w \in \mathbb{R}^E_{>0}$. Let $p \in \mathbb{R}^V$ be the corresponding potential vector.

1. *If $b_P \overset{\text{def}}{=} \sum_{v : b_v > 0} b_v$ is the total demand, then $|q_e| \leqslant b_P$ for every edge $e \in E$.*
2. *The energy of the flow $\mathcal{E}(q) \overset{\text{def}}{=} \sum_{e \in E} q_e^2/w_e$ is equal to $b^\top p$.*

Fact 4.5

Suppose g is a basic flow in the directed graph $G = (V, E)$ for vertex demands $b \in \mathbb{Z}^V$. Then $g \in \mathbb{Z}^E_{\geqslant 0}$ and $g_e \leqslant b_P$, where $b_P \overset{\text{def}}{=} \sum_{v : b_v > 0} b_v$ is the total demand.

Fact 4.6

Suppose g is a noncircular flow in the directed graph $G = (V, E)$ for vertex demands $b \in \mathbb{R}^V$. Then g can be expressed as a convex combination of at most $|E|$ basic flows.

Preconditioning the Instance

Recall that an instance of the directed transshipment problem is simply a graph $G_0 = (V, E_0)$ together with a positive cost vector $c_0 \in \mathbb{Z}^E_{>0}$ and a demand vector $b \in \mathbb{Z}^V$. We assume without loss of generality that this instance is feasible, i.e., that there exists a vector $f \in \mathbb{R}^E_{\geqslant 0}$ such that $Bf = b$, where B is the incidence matrix of G. We construct a new instance (G, b, c) and an initial point $x(0) \in \mathbb{Z}^E_{>0}$ in the following way. Let $G = (V, E)$ for $E = E_0 \cup E'$ where E' consists of new edges for every pair of vertices u, v with $b_u < 0$ and $b_v > 0$, i.e.,

$$E' = \{(u, v) : b_u < 0 \wedge b_v > 0\}$$

(note that E may be a multiset). The costs of old edges remain the same, while for a new edge $e' \in E'$ we set $c_{e'} = nC$, where C is the maximum cost over $e \in E_0$. For the initial vector we assign $x_e(0) = 1$ for all $e \in E_0$ and for $e' = (u, v) \in E'$ we set $x_{e'} = 2m \cdot |b_u| \cdot b_v$. Note that by adding E' to G_0 we do not introduce any solution with lower cost than the optimal solution in G_0. As before, let $C = \max_{e \in E_0} c_e$ and b_P be the total demand.

We now study properties of the preconditioned instance. Denote

$$\widetilde{C} := \max_{e \in E} c_e = nC.$$

In what follows, for an edge $e = (u, v)$ we denote the potential drop $p_u - p_v$ by $\Delta_e p$. We start by proving that the sequence $x(k)$ is well defined for all $k \in \mathbb{N}$. Even this basic property is not straightforward to obtain. In the next lemma we prove a key property of the preconditioned instance: existence of x-capacitated flows. It implies a bound on the maximum potential difference and positivity of the conductance vector.

Lemma 4.18 (*Properties of the preconditioned instance*)

Let (G, b, c) be the preconditioned instance and $x(0)$ be the corresponding initial vector. Suppose that $h < {}^{1}/_{n\widetilde{C}+1}$ Then, for every $k \in \mathbb{N}$, $x(k)$ satisfies:

1. *Positivity: $x_e(k) > 0$ for every $e \in E$,*
2. *Bounded potentials: $\max_{u,v\in V} |p_u(k) - p_v(k)| \leqslant n\widetilde{C}$,*
3. *x-capacitated flows: there is a vector $f \in \mathbb{R}^E$ with $0 \leqslant f \leqslant x(k)$ such that $Bf = b$.*

Proof.
We prove this lemma by joint induction on the listed properties. Towards this end, we start by proving some useful implications between the properties. First we prove that whenever both properties (1) and (3) hold for some k, then the property (2) holds as well.

$(1)_k, (3)_k \Rightarrow (2)_k$**:** Let f be such a flow, respecting the capacities x and satisfying demands b. By the mincut-maxflow theorem this means that whenever we have a partition of V into two sets $S, \bar{S}$, such that $b_S = \sum_{v\in S} b_v > 0$ then $\sum_{e\in E(\bar{S},S)} x_e \geqslant b_S$. Sort all of the potentials in nondecreasing order and pick two neighbouring ones p_u, p_v. In other words, take u, v, such that $p_u \leqslant p_v$ and for all $w \in V$ either $p_w \leqslant p_u$ or $p_w \geqslant p_v$. We show that $p_v - p_u \leqslant \widetilde{C}$.

Assume the contrary, $p_v - p_u > \widetilde{C}$. Recall that $\widetilde{C}$ is the maximum cost c_e over all $e \in E$. We define a partition of V into two sets

$$S = \{w \in V : p_w \leqslant p_u\} \quad \textit{and} \qquad \bar{S} = \{w \in V : p_w > p_u\}.$$

Let 1_S be the indicator vector of the set S. We know that $Bq = b$, multiplying both sides by $1_S^\top$ yields $1_S^\top Bq = 1_S^\top b = b_S$. Thus,

$$b_S = \sum_{e\in E(\bar{S},S)} q_e - \sum_{e\in E(S,\bar{S})} q_e.$$

Note that since electrical flow goes always from bigger potential to lower potential, for $e \in E(\bar{S}, S)$ we have $q_e > 0$ and for $e \in E(S, \bar{S})$ we have $q_e < 0$. Moreover, there is at least one edge between $S, \bar{S}$, since the graph is connected, hence $b_S > 0$. We have

$$b_S \geqslant \sum_{e\in E(\bar{S},S)} q_e = \sum_{e\in E(\bar{S},S)} {}^{x_e \Delta_e p}/_{c_e} > \sum_{e\in E(\bar{S},S)} {}^{x_e \widetilde{C}}/_{c_e} \geqslant \sum_{e\in E(\bar{S},S)} x_e \geqslant b_S.$$

The last inequality follows from mincut-maxflow. We have reached a contradiction, hence $p_v - p_u \leqslant \widetilde{C}$. Consequently,

$$\max_{u,v\in V} |p_u - p_v| \leqslant n\widetilde{C}.$$

Note that we have implicitly used $(1)_k$ to reason about potentials (if we do not have $(1)_k$ the electrical flow could not be well defined). The next step is to argue that property (2) for step k implies (1) for step $k+1$.

$(1)_k, (2)_k \Rightarrow (1)_{k+1}$**:** Pick any edge $e \in E$. We have

$$x_e(k+1) = (1-h)x_e(k) + hq_e(k) = (1-h)x_e(k) + h^{x_e \Delta_e p}/_{c_e} \geqslant x_e(k)(1 - h(1 + n\widetilde{C})) > 0.$$

The last implication we would like to argue about is the following: whenever $x(0), \ldots, x(k)$ are well defined (i.e., (1) holds), then (3) is true for step k.

$(1)_0, (1)_1, \ldots, (1)_k \Rightarrow (3)_k$**:** To argue that a certain flow exists, we show lower bounds on cut capacities. Pick any partition of V into $S, \bar{S}$ with $b_S > 0$. We need to show that

$$\sum_{e \in E(\bar{S}, S)} x_e(k) \geqslant b_S.$$

We make use of the quantity

$$x_S(j) \stackrel{\text{def}}{=} \sum_{e \in E(\bar{S}, S)} x_e(j) - \sum_{e \in E(S, \bar{S})} x_e(j).$$

Clearly,

$$\sum_{e \in E(\bar{S}, S)} x_e(k) \geqslant x_S(k),$$

so it is enough for us to show $x_S(k) \geqslant b_S$. It turns out that for $x_S(j)$ we can obtain a nice formula, let us first compute

$$\begin{aligned}
x_S(j+1) &= \sum_{e \in E(\bar{S}, S)} x_e(j+1) - \sum_{e \in E(S, \bar{S})} x_e(j+1) \\
&= (1-h)x_S(j) + h\Big(\sum_{e \in E(\bar{S}, S)} q_e(j) - \sum_{e \in E(S, \bar{S})} q_e(j)\Big) \\
&= (1-h)x_S(j) + h1_S^\top Bq \\
&= (1-h)x_S(j) + hb_S.
\end{aligned}$$

This yields a recursive formula for $x_S(j)$. We can solve it and obtain

$$x_S(j) = (1-h)^j x_S(0) + (1 - (1-h)^j) b_S.$$

Thus $x_S(j)$ is a convex combination of $x_S(0)$ and b_S. Thus, showing that $x_S(j) \geqslant b_S$ is in fact equivalent to showing $x_S(0) \geqslant b_S$. We count separately the contribution of edges from E_0 and E' to $x_S(0)$. The first part is simple:

$$\sum_{e \in E_0(\bar{S}, S)} x_e(0) - \sum_{e \in E_0(S, \bar{S})} x_e(0) \geqslant -m.$$

From now on we focus on the new edges E'. Let

$$S_- = \{v \in S : b_v < 0\} \quad \text{and} \quad S_+ = \{v \in S : b_v > 0\},$$

similarly we define $\bar{S}_-$ and $\bar{S}_+$. Note that $b_{S_+} = -b_{\bar{S}_-}$ and $b_{S_-} = -b_{\bar{S}_+}$, also clearly $b_S = b_{S_+} + b_{S_-}$. From the definition of $x(0)$, one can see that

$$\textstyle\sum_{e \in E'(\bar{S},S)} x_e(0) = 2m \cdot |b_{\bar{S}_-}| \cdot b_{S_+} \quad \text{and} \quad \textstyle\sum_{e \in E'(S,\bar{S})} x_e(0) = 2m \cdot |b_{S_-}| \cdot b_{\bar{S}_+}.$$

Hence, the contribution of edges from E' to $x_S(0)$ is

$$2m(|b_{\bar{S}_-}| \cdot b_{S_+} - |b_{S_-}| \cdot b_{\bar{S}_+}) = 2m(b_{S_+}^2 - b_{S_-}^2) = 2m(b_{S_+} + b_{S_-})(b_{S_+} - b_{S_-}) \geqslant 2mb_S.$$

Hence $x_S(0) \geqslant 2mb_S - m \geqslant b_S$ (we used $b_S \geqslant 1$), which concludes the proof.

After showing the three implications above, the lemma follows by joint induction. In the base case it is enough to observe that $(1)_0$ holds. Then $(3)_0$ follows immediately, which in turn yields $(2)_0$. The three conditions allow us to conclude $(1)_1$ and so on. □

By Lemma 4.18 we know that the sequence $\{x(k)\}_{k \in \mathbb{N}}$ is well defined and we can reason about its properties. From now on we always assume that we work with the preconditioned instance and h is small enough, so that the conclusion of Lemma 4.18 holds.

Proof of Convergence

In this subsection we present a proof of convergence of the discretization for the directed transshipment problem. Recall that we work with the preconditioned instance. Let us first state some preliminary properties.

Lemma 4.19 (*Bounds for conductances*)

For every $k \geqslant 0$ the following hold:

1. $x_e(k) = x_e(0)(1-h)^k + h \sum_{j=0}^{k-1} (1-h)^{k-j} q_e(j)$,
2. $x_e(k) = x_e(0) \prod_{j=0}^{k-1} \left(1 + h\left({}^{\Delta_e p(j)}/_{c_e} - 1\right)\right)$,
3. $x_e(k) \leqslant 2mb_P^2$.

Proof.
Formulas (1) and (2) can be proved by induction. We proceed with (3).
In the proof we will use a property of electrical flows 4.4 (1), which implies that $q_e(k) \leqslant b_P$. We have

$$x_e(k+1) = (1-h)x_e(k) + hq_e(k) \leqslant (1-h)x_e(k) + hb_P.$$

By induction, it follows that

$$x_e(k) \leqslant (1-h)^k x_e(0) + (1-(1-h)^k)b_P \leqslant 2mb_P^2.$$

□

Since it is not convenient to work with $x(k)$ (we would prefer $x(k)$ to be a flow), we prove that for big enough k there is always a nonnegative flow $f(k)$ which is close to $x(k)$. Then we argue that for big k the resulting flow $f(k)$ is close to the optimal solution, hence $x(k)$ is close as well.

Lemma 4.20 (*Rounding almost nonnegative flows, [IJNT11]*)

Let h be an arbitrary flow. Let $F \subseteq E$ be a set of edges, satisfying $w \overset{\text{def}}{=} \sum_{e \in F} |h_e| < 1$ and $h_e \geqslant 0$ for all $e \in E \setminus F$. Then there exists a nonnegative flow f, such that $\operatorname{supp}(f) \subseteq (\operatorname{supp}(h) \setminus F)$ and $\|f - h\|_\infty \leqslant w$.

We omit the proof. Now, we prove that for large k, $x(k)$ is an almost nonnegative flow.

Lemma 4.21 (*Conductance vector is close to a noncircular flow*)

For every ε with $0 < \varepsilon < 1$ and for every $k > {}^{10\ln(nmb_P/\varepsilon)}/_h$ there exists a noncircular flow $f \geqslant 0$ such that $\|x(k) - f\|_\infty < \varepsilon$.

Proof.
Denote

$$\bar{q}(k) = h \sum_{j=0}^{k-1} (1-h)^{k-j} q(j),$$
$$\bar{p}(k) = \sum_{j=0}^{k-1} p(j),$$
$$X_0 = \max_{e \in E} x_e(0).$$

We will find a noncircular, nonnegative flow f which satisfies $\|f - \bar{q}(k)\|_\infty < \varepsilon/2$, our choice of k implies $\|x(k) - \bar{q}(k)\|_\infty < \varepsilon/2$ (as we will show) and the result follows. First note that by Lemma 4.19 (1):

$$\|x(k) - \bar{q}(k)\|_\infty = \max_{e \in E} x_e(0)(1-h)^k \leqslant X_0(1-h)^k.$$

For this to become smaller than $\varepsilon/2$, it suffices to take $k > \frac{3\ln(X_0/\varepsilon)}{h}$. By taking $k > \frac{10\ln(X_0 nm/\varepsilon)}{h}$ we can take it down to at most $\frac{\varepsilon}{4(m^2+n)}$.

Now assume that $k > \frac{10\ln(nmX_0/\varepsilon)}{h}$ and consider the set of edges

$$F = \{e \in E : \bar{q}(e) \leqslant 0\} \cup \{e \in E : \Delta_e \bar{p}(k) \leqslant 0\}.$$

We intend to apply Lemma 4.20 for h equal to $\bar{q}(k)$, for this we need to bound the quantity $\sum_{e \in F} |\bar{q}_e(k)|$. If e satisfies $\bar{q}(e) \leqslant 0$ then by Lemma 4.19 part (1) we have

$$|\bar{q}_e(k)| \leqslant x_e(0)(1-h)^k < \frac{\varepsilon}{4(m+n^2)} < \frac{\varepsilon}{m+n^2}.$$

Similarly, if $\Delta_e \bar{p}(k) \leqslant 0$, then, by Lemma 4.19 (2),

$$\begin{aligned} x_e(k) &= x_e(0) \prod_{j=0}^{k-1} \left(1 + h\left(\frac{\Delta_e p(j)}{c_e} - 1\right)\right) \\ &\leqslant x_e(0) \exp\left(h \sum_{j=0}^{k-1} \left(\frac{\Delta_e p(j)}{c_e} - 1\right)\right) \\ &= x_e(0) \exp\left(-hk + \frac{h}{c_e} \Delta_e \bar{p}(k)\right) \\ &\leqslant x_e(0) \exp(-hk) \\ &< \frac{\varepsilon}{2(m+n^2)}. \end{aligned}$$

We also know that for such big k, $|x_e(k) - \bar{q}_e(k)| < \frac{\varepsilon}{2(m+n^2)}$ holds, so

$$|q_e(k)| < \frac{\varepsilon}{2(m+n^2)} + \frac{\varepsilon}{4(m+n^2)} \leqslant \frac{\varepsilon}{m+n^2}.$$

By the above calculation we obtain

$$\sum_{e \in F} |\bar{q}_e(k)| < |F| \frac{\varepsilon}{2(m+n^2)} \leqslant \frac{\varepsilon}{2}.$$

Thus we can apply Lemma 4.20 and obtain a flow f. Note that the obtained flow satisfies:

- $f \geqslant 0$,
- f is noncircular. Let $\gamma \geqslant 0$ be a directed cycle, i.e, $B\gamma = 0$ then $\sum_{e \in E} \Delta_e \bar{p}(k) \gamma_e = \bar{p}(k)^\top B\gamma = 0$, therefore there must be some edge $e \in E$ with $\gamma_e > 0$ and $\Delta_e \bar{p}(k) \leqslant 0$, hence $e \in F$, which implies $f_e = 0$,
- $\|f - \bar{q}(k)\|_\infty < \frac{\varepsilon}{2}$, as Lemma 4.20 guarantees.

□

From now on for simplicity we assume that there exists a unique optimal solution $f^\star$ to the underlying instance. For every step k, let $f(k)$ be the nonnegative, noncircular flow which is guaranteed to exist by Lemma 4.21. Our goal is to show that for k large enough, $\|f(k) - f^\star\|_\infty$ is small, this in turn would imply that $x(k)$ is close to $f^\star$. We proceed with two technical lemmas.

Lemma 4.22 (*Noncircular flow is close to optimal flow*)

Let $0 < \varepsilon < 1$. Let f be a nonnegative, noncircular flow. Suppose that for every basic flow $g \neq f^\star$ there exists an edge $e \in E$ with $g_e > 0$, such that $f_e < \varepsilon / 2b_P(m+n^2)$. Then $\|f - f^\star\|_\infty < \varepsilon$.

Proof.
Since f is noncircular, by Fact 4.6, it can be written as a convex combination of basic flows

$$f = \sum_{i=1}^{M} \alpha_i f_i,$$

where $M \leqslant |E| \leqslant n^2 + m$. We assume that f_i are pairwise distinct in this decomposition and $f_1 = f^\star$. Take g to be any of f_i for some $i \geqslant 2$. From the hypothesis there is $e \in E$ such that $g_e > 0$ and $f_e < \frac{\varepsilon}{2b_P(m+n^2)}$. Recall that g is a basic flow, hence by Fact 4.5 it satisfies $1 \leqslant g_e \leqslant b_P$. Moreover,

$$\alpha_i \leqslant \alpha_i g_e = \alpha_i f_{ie} \leqslant \sum_{i=1}^{M} \alpha_i f_{ie} = f_e < \frac{\varepsilon}{2b_P(m+n^2)}.$$

This implies that $\sum_{i=2}^{M} \alpha_i < \frac{\varepsilon}{2b_P}$, hence $\alpha_1 > 1 - \frac{\varepsilon}{2b_P}$. Moreover

$$\|\sum_{i=2}^{M} \alpha_i f_i\|_\infty \leqslant \frac{\varepsilon}{2b_P(m+n^2)} \sum_{i=2}^{M} \|f_i\|_\infty \leqslant \frac{Mb_P \cdot \varepsilon}{2b_P(m+n^2)} \leqslant \frac{\varepsilon}{2}.$$

To finish the proof it remains to argue that $\|\alpha_1 f_1 - f^\star\|_\infty < \frac{\varepsilon}{2}$. Indeed,

$$\|\alpha_1 f_1 - f^\star\|_\infty = \|\alpha_1 f^\star - f^\star\|_\infty = |1-\alpha_1| \|f^\star\|_\infty < \frac{\varepsilon}{2b_P} b_P = \frac{\varepsilon}{2}.$$

□

Lemma 4.23 (*Ruling out nonoptimal basic flows*)

Let $\varepsilon \in (0,1)$ and g be any nonoptimal basic flow. Suppose $h < 1/2nb_P(n\widetilde{C}+\widetilde{C})^2$. For every $k > 12\widetilde{C}b_P/h \ln (mb_P/\varepsilon)$ there is an edge $e \in E$ such that $g_e > 0$ and $x_e(k) < \varepsilon$.

Proof.
Fix ε and g as in the statement. We consider the following parametrized barrier function

$$\mathcal{B}_g(k) \overset{\text{def}}{=} \sum_{e \in E} g_e c_e \ln x_e(k).$$

The idea is to show that $\mathcal{B}_g(k)$ gets very small for k large enough. To this end consider

$$\begin{aligned}
\mathcal{B}_g(k+1) - \mathcal{B}_g(k) &= \sum_{e \in E} g_e c_e \ln \frac{x_e(k+1)}{x_e(k)} \\
&= \sum_{e \in E} g_e c_e \ln \left(1 + h\left(\frac{\Delta_e p(k)}{c_e} - 1\right)\right) \\
&\leqslant h \sum_{e \in E} g_e c_e \frac{\Delta_e p(k)}{c_e} - hc^\top g.
\end{aligned}$$

Observe that

$$\sum_{e \in E} g_e c_e \frac{\Delta_e p(k)}{c_e} = \sum_{e \in E} g_e \Delta_e p(k) = g^\top B^\top p(k) = (Bg)^\top p(k) = b^\top p(k).$$

Moreover, by the property 4.4 (2), we know that $b^\top p(k) = \mathcal{E}(k)$. Hence

$$\mathcal{B}_g(k+1) - \mathcal{B}_g(k) \leqslant h\mathcal{E}(k) - hc^\top g.$$

Let us now look how $\mathcal{B}_{f^\star}(k) \overset{\text{def}}{=} \sum_{e\in E} f^\star_e c_e \ln x_e(k)$ changes. We will use the inequality $\ln(1+\alpha) \geqslant \alpha - \alpha^2$ valid for $|\alpha| \leqslant \frac{1}{2}$,

$$\begin{aligned}\mathcal{B}_{f^\star}(k+1) - \mathcal{B}_{f^\star}(k) &= \sum_{e\in E} f^\star_e c_e \ln\left(1 + h\left(\frac{\Delta_e p(k)}{c_e} - 1\right)\right)\\ &\geqslant h\sum_{e\in E} f^\star_e c_e \left(\frac{\Delta_e p(k)}{c_e} - 1\right) - h^2 \sum_{e\in E} f^\star_e c_e \left(\frac{\Delta_e p(k)}{c_e} - 1\right)^2.\end{aligned}$$

We need to bound the second term:

$$\begin{aligned}\sum_{e\in E} f^\star_e c_e \left(\frac{\Delta_e p(k)}{c_e} - 1\right)^2 &= \sum_{e\in E} f^\star_e \left|\Delta_e p(k) - c_e\right| \left|\frac{\Delta_e p(k)}{c_e} - 1\right|\\ &\leqslant (n\widetilde{C} + \widetilde{C})^2 \sum_{e\in E} f^\star_e\\ &\leqslant (n\widetilde{C} + \widetilde{C})^2 n b_P.\end{aligned}$$

Now we can continue our estimate of $\Delta\mathcal{B}_{f^\star}(k)$:

$$\begin{aligned}\mathcal{B}_{f^\star}(k+1) - \mathcal{B}_{f^\star}(k) &\geqslant h\mathcal{E}(k) - hc^\top f^\star - h^2 \sum_{e\in E} f^\star_e c_e \left(n\widetilde{C} + 1\right)^2\\ &\geqslant h\mathcal{E}(k) - hc^\top f^\star - h^2 (n\widetilde{C} + \widetilde{C})^2 n b_P\\ &\geqslant h\mathcal{E}(k) - hc^\top f^\star - \frac{h}{2}.\end{aligned}$$

By combining our bounds on $\Delta\mathcal{B}_g(k)$ and $\Delta\mathcal{B}_{f^\star}(k)$ we obtain

$$\Delta\mathcal{B}_g(k) \leqslant -hc^\top g + hc^\top f^\star + \frac{h}{2} + \Delta\mathcal{B}_{f^\star}(k) \leqslant -h(c^\top g - c^\top f^\star) + \frac{h}{2} + \Delta\mathcal{B}_{f^\star}(k).$$

Since both $f^\star$ and g are integral solutions $c^\top g \geqslant c^\top f^\star + 1$, this implies in particular that:

$$\Delta\mathcal{B}_g(k) \leqslant -\frac{h}{2}(c^\top g - c^\top f^\star) + \Delta\mathcal{B}_{f^\star}(k).$$

By expanding, we obtain

$$\mathcal{B}_g(k) \leqslant -\frac{hk}{2}(c^\top g - c^\top f^\star) + \mathcal{B}_{f^\star}(k) - \mathcal{B}_{f^\star}(0) + \mathcal{B}_g(0) = -\frac{hk}{2}(c^\top g - c^\top f^\star) + \mathcal{B}_{f^\star}(k) + \mathcal{B}_g(0). \quad (4.10)$$

Before we proceed, let us first bound $\mathcal{B}_{f^\star}(k) + \mathcal{B}_g(0)$, Lemma 4.19 (3) tells us that $x_e(j) \leqslant 2mb_P^2$ for every j, hence

$$\mathcal{B}_g(0) = \sum_{e\in E} g_e c_e \ln x_e(0) \leqslant \ln\left(2mb_P^2\right) \sum_{e\in E} g_e c_e = c^\top g \cdot \ln\left(2mb_P^2\right).$$

A similar bound applies to $\mathcal{B}_{f^\star}(k)$:

$$\mathcal{B}_{f^\star}(k) \leqslant c^\top f^\star \cdot \ln\left(2mb_P^2\right).$$

Combining them,

$$\mathcal{B}_{f^\star}(k) + \mathcal{B}_g(0) \leqslant (c^\top g + c^\top f^\star) \ln\left(2mb_P^2\right). \tag{4.11}$$

Suppose now that $x_e(k) \geqslant \varepsilon$ for all e with $g_e > 0$. We will show that k has to be small. We have:

$$\mathcal{B}_g(k) = \sum_{e\in E} g_e c_e \ln x_e(k) \geqslant \sum_{e\in E} g_e c_e \ln \varepsilon \geqslant \widetilde{C} b_P \ln \varepsilon. \tag{4.12}$$

Combining (4.10), (4.11) and (4.12) we have:

$$\widetilde{C} b_P \ln \varepsilon \leqslant -\frac{hk}{2}(c^\top g - c^\top f^\star) + (c^\top g + c^\top f^\star) \ln\left(2mb_P^2\right).$$

Hence,

$$\frac{k}{2} \leqslant \frac{\widetilde{C} b_P \ln 1/\varepsilon}{h(c^\top g - c^\top f^\star)} + \frac{(c^\top g + c^\top f^\star)\ln\left(2mb_P^2\right)}{h(c^\top g - c^\top f^\star)}.$$

Since $c^\top g - c^\top f^\star \geqslant 1$, the first term is bounded by $h^{-1}\widetilde{C} b_P \ln 1/\varepsilon$. The second term is bounded by $h^{-1}(2c^\top f^\star + 1)\ln\left(2mb_P^2\right)$. Note that $c^\top f^\star \leqslant \widetilde{C} b_P$ since in the optimal solution the cost per one unit of flow is at most $\widetilde{C}$. Altogether this gives a bound

$$k \leqslant \frac{12\widetilde{C} b_P}{h} \ln\left(\frac{mb_P}{\varepsilon}\right).$$

□

Proof of Theorem 4.4:
We prove the theorem in the special case when there is only one optimal solution $f^\star$. The general case requires some minor adjustments in the statements of lemmas.

Fix any $\varepsilon \in (0,1)$. Let $\delta = \frac{\varepsilon}{4b_P(m+n^2)}$. We choose h to be

$$h := \frac{1}{4nb_P(n\widetilde{C})^2} < \frac{1}{2nb_P(n\widetilde{C} + \widetilde{C})^2}$$

(to make sure the hypothesis of Lemma 4.23 is satisfied). Fix

$$k > \frac{12\widetilde{C} b_P}{h} \ln\left(\frac{mb_P}{\delta}\right) = \frac{12\widetilde{C} b_P}{h} \ln\left(4b_P^2(m+n^2)m \cdot \varepsilon^{-1}\right)$$

and use Lemma 4.23 (with δ in place of ε). We are guaranteed that for every basic nonoptimal flow g, there is an edge $e \in E$ such that $g_e > 0$ and $x_e(k) < \delta$.

Because k is big enough, we can use Lemma 4.21 (again with δ in place of ε) to obtain a noncircular flow $f \geqslant 0$ with $\|f - x(k)\|_\infty < \delta$. Hence, for every basic nonoptimal flow g, there is an edge $e \in E$ such that $g_e > 0$ and $f_e < 2\delta = \frac{\varepsilon}{2b_P(m+n^2)}$. By Lemma 4.22, $\|f - f^\star\|_\infty < \frac{\varepsilon}{2}$, hence

$$\|x(k) - f^\star\|_\infty \leqslant \|f^\star - f\|_\infty + \|f - x(k)\|_\infty \leqslant \frac{\varepsilon}{2} + \delta < \varepsilon.$$

□

4.4 Conclusion

4.4.1 Future Work

Several questions on the directed Physarum dynamics remain open. The most interesting and perhaps the most challenging at the same time is whether a discretization of the directed Physarum dynamics can be made into a polynomial time algorithm for linear programming. For this one perhaps needs to resort to higher order discretization methods, as the first order (Euler methods) seems to inherently suffer a linear (not logarithmic) dependency on the cost vector.

The entropy barrier following from the optimization viewpoint of the dynamics seems to be an interesting object. It does not satisfy the usual self-concordance axioms, yet seem to perform well and perhaps can inspire a polynomial time linear programming algorithm.

4.4.2 Notes

The content of this chapter is based on results obtained in [SV16b] and [SV16a]. There are rigorous results for existence, convergence and efficiency for the shortest path problem [BBD$^+$13]. The LP Physarum dynamics was introduced by [JZ12]. There is a large body of literature in the mathematical programming community which studies the geometric properties of continuous trajectories of interior point methods such as projective and affine scaling; see [Kar90, BL89, NT02]. A dynamical system, which is perhaps the closest to Physarum dynamics in the literature, has been studied in [Fay91]. It matches the Physarum dynamics up to a certain rescaling, yet is defined only over the feasible region of the linear program, while Physarum is defined everywhere. Physarum dynamics also bears similarity to primal affine scaling methods, see, e.g., [LV90]. However, for affine scaling methods, the convergence rate can be provably bad: $1/t$, see, e.g., [MS89].

Chapter 5

The Undirected Physarum Dynamics

In this chapter we study the undirected Physarum dynamics. We introduce the most general form of this dynamical system: for the basis pursuit problem and prove that its discretization converges to optimal solutions in value, whenever the step size is small enough. Subsequently we develop an understanding of this dynamics from the viewpoint of optimization. We prove that it is essentially a small-step variant of the IRLS algorithm, whose convergence is still an open problem.

5.1 Preliminaries and Statement of Results

5.1.1 Preliminaries

Basis Pursuit. Recall that in the basis pursuit problem one is given a matrix $A \in \mathbb{R}^{n\times m}$ and a vector $b \in \mathbb{R}^n$, typically with n much smaller than m. The goal is to compute a solution $x \in \mathbb{R}^m$ to the linear system $Ax = b$ that has the smallest possible ℓ_1 norm. In other words, we are interested in the following optimization problem

$$\min_{x\in\mathbb{R}^m} \sum_{i=1}^{m} |x_i| \quad \text{s.t. } Ax = b. \tag{5.1}$$

We assume that the matrix A has rank n, i.e., all of its rows are linearly independent. Every linear system can be efficiently brought into this form by removing certain equations. In the remaining part of this chapter we refer to $x^\star$ as an arbitrary (but fixed) solution to (5.1). Each pair (A, b), composed of a matrix A satisfying the above rank assumption and a vector b such that the linear system $Ax = b$ has a solution, is an instance of the basis pursuit problem.

Weighted ℓ_2-minimization An important component of the IRLS algorithm and the Physarum dynamics is the following **weighted ℓ_2-minimization** problem. For a matrix $A \in \mathbb{R}^{n\times m}$, a vector $b \in \mathbb{R}^n$ and weights $w \in \mathbb{R}^m_{>0}$, the weighted ℓ_2-minimization problem is as follows:

$$\min_{x\mathbb{R}^m} \sum_{i=1}^{m} w_i^{-1} x_i^2 \qquad \text{s.t. } Ax = b. \tag{5.2}$$

We denote by $q(w)$ the optimal solution to the above optimization problem. See Fact 5.2 for a proof that $q(w)$ exists and is unique.

Physarum dynamics. We now recall the undirected Physarum dynamics for the basis pursuit problem. For the purpose of clarity in notation we slightly deviate from the notation in Chapter 2, where the solutions to the Physarum dynamics are denoted by x. Here we use σ for solutions, and we reserve x for a use as a free variable. Similarly for the vector q, we use q as a function in this chapter and hence in the below definition it is replaced by φ. The Physarum dynamics for the basis pursuit problem is defined by the following system of differential equations, with $\sigma(t) \in \mathbb{R}^m_{\geqslant 0}$

$$\frac{d}{dt}\sigma(t) = |\varphi(t)| - \sigma(t), \tag{5.3}$$

where $\varphi(t) := \Sigma(t)A^\top(A\Sigma(t)A^\top)^{-1}b$ and $\Sigma(t)$ denotes a diagonal matrix with $\sigma(t)$ on the diagonal. Additionally, an initial condition $\sigma(0) \in \mathbb{R}^m_{>0}$ is specified. In Section 5.2.2 it is proved that for every such initial condition, (5.3) has a unique global solution $\sigma : [0, \infty) \to \mathbb{R}^m_{>0}$.

The IRLS algorithm for solving the basis pursuit problem, is initialized with $z^{(0)}$ – any solution to the linear system $Ax = b$. Subsequently, given $z^{(k)}$, the algorithm computes $z^{(k+1)}$ as follows:

$$z^{(k+1)} := \operatorname*{argmin}_{x \in \mathbb{R}^m} \frac{x_i^2}{\left|z_i^{(k)}\right|} \quad \text{s.t. } Ax = b.$$

In other words, $z^{(k+1)} = q\left(\left|z^{(k)}\right|\right)$. This gives rise to a sequence of vectors $z^{(0)}, z^{(1)}, \ldots$. We denote the output sequence $\left(z^{(k)}\right)_{k\in\mathbb{N}}$ by

$$\mathrm{IRLS}\left[A, b, z^{(0)}\right].$$

Meta-Algorithm. To formalize the connection between the continuous Physarum dynamics and the IRLS algorithm we introduce the following **Meta-Algorithm**. For a given step size $h \in (0, 1]$, it is initialized with a candidate solution $y^{(0)} \in \mathbb{R}^m$ that satisfies $Ay^{(0)} = b$ and a vector of weights $w^{(0)} \in \mathbb{R}^m_{>0}$, and proceeds according to the update rule:

$$(y^{(k+1)}, w^{(k+1)}) := (1 - h)(y^{(k)}, w^{(k)}) + h(q^{(k)}, |q^{(k)}|),$$

where $q^{(k)} = q(w^{(k)})$. We denote the sequence $\left((y^{(k)}, w^{(k)})\right)_{k\in\mathbb{N}}$ by

$$\mathrm{MA}\left[A, b, h, y^{(0)}, w^{(0)}\right].$$

Note that $\left(w^{(k)}\right)_{k\in\mathbb{N}}$ depends only on h and $w^{(0)}$ and not on $y^{(0)}$. Therefore, we also use $\mathrm{MA}\left[A, b, h, w^{(0)}\right]$ to denote the resulting sequence of weights $\left(w^{(k)}\right)_{k\in\mathbb{N}}$.

Remark 5.1

Note that when one sets $h = 1$ in the Meta-Algorithm, it might happen that the weight vector $w^{(k)}$ ends up having a zero coordinate, i.e., $w_i^{(k)} = 0$ for some $i \in \{1, 2, \ldots, m\}$ and $k \in \mathbb{N}$. In such a case, the corresponding weight in the ℓ_2-minimization problem should be treated as $+\infty$ and, hence, forces $q_i^{(k)} = 0$. In other words, in the presence of zeros in $w^{(k)}$, we set every zero-coordinate to 0 in $q^{(k)}$ and solve a weighted ℓ_2-minimization problem over the remaining coordinates. In fact, this is how the algorithm is formally defined.

5.1.2 Statement of Results

Our first result asserts that both the IRLS and the Physarum dynamics can be seen as special cases of the Meta-Algorithm; this establishes an equivalence between them.

Theorem 5.1 (*Equivalence*)

Let (A, b) be any instance of the basis pursuit problem.

1. *Let $y^{(0)}$ be any solution to the linear system $Ax = b$ and let*
$$\left(y^{(k)}, w^{(k)}\right)_{k\in\mathbb{N}} := \mathrm{MA}\left[A, b, 1, y^{(0)}, \left|y^{(0)}\right|\right].$$
Then $\mathrm{IRLS}\left[A, b, y^{(0)}\right] = \left(y^{(k)}\right)_{k\in\mathbb{N}}$.

2. *Let $\sigma : [0, \infty) \to \mathbb{R}^m_{>0}$ be a solution to the Physarum dynamics (5.3) starting at $\sigma(0)$. Then, for any $t > 0$ we have*
$$\lim_{h\to 0^+} w_h^{(k(h))} = \sigma(t),$$
where $k(h) := \lfloor \frac{t}{h} \rfloor$ and $(w_h^{(k)})_{k\in\mathbb{N}} := \mathrm{MA}\,[A, b, h, \sigma(0)]$ for any $h \in (0, 1)$. In words, for any fixed t, if the Meta-Algorithm and the Physarum dynamics are initiated at the same initial point $\sigma(0)$ then, as $h \to 0$, the value of the Physarum dynamics after time t is the same as that of the Meta-Algorithm after a suitable number of steps.

Our second theorem gives a quantitative convergence bound for the Meta-Algorithm. For the sake of clarity of presentation, it is assumed that the input matrix A and the input vector b have integer entries.

Theorem 5.2 (*Convergence and Complexity*)

Let (A, b) be any integral instance to the basis pursuit problem. Suppose that $y^{(0)}$ and $w^{(0)}$ are chosen so as to satisfy $Ay^{(0)} = b$ and $w^{(0)} \geqslant \left|y^{(0)}\right|$. Furthermore, assume $w_i^{(0)} \geqslant 1$ for every $i \in \{1, 2, \ldots, m\}$ and $\|w^{(0)}\|_1 \leqslant M\|x^\star\|_1$ for some $M \in \mathbb{R}$. Let $\varepsilon \in (0, 1/2)$ and $h \leqslant \frac{\varepsilon}{20m\mathcal{D}}$. Then for $k = O\left(\frac{\ln M + \ln\|x^\star\|_1}{h\varepsilon^2}\right)$ we have $\|y^{(k)}\|_1 \leqslant (1+\varepsilon)\|x^\star\|_1$, where $\left((y^{(k)}, w^{(k)})\right)_{k\in\mathbb{N}} :=$ $\mathrm{MA}\left[A, b, h, y^{(0)}, w^{(0)}\right]$.

In the above statement

$$\mathcal{D} := \max\left\{|\det(A')| : A' \text{ is a square submatrix of } A\right\}. \tag{5.4}$$

A few remarks are in order. The assumptions on the starting point $(y^{(0)}, w^{(0)})$ that we make in the statement are not crucial. However, they allow us to state the bounds in a simple form and make the proof much cleaner. Note that by taking *any* feasible solution $y^{(0)}$ (i.e., $Ay^{(0)} = b$) and defining $w_i^{(0)} := \max(1, \left|y^{(0)}\right|)$ for $i = 1, 2, \ldots, m$ we obtain an initial solution which trivially satisfies the condition of the theorem. For instance, one can obtain $y^{(0)}$ by solving the least squares problem over the subspace $Ax = b$, i.e., minimize $\|x\|_2$ instead of $\|x\|_1$ as in basis pursuit. Since the norms $\|\cdot\|_1$ and $\|\cdot\|_2$ differ only by at most a $\sqrt{m}$ multiplicative factor, one can take $M = O(\sqrt{m})$ in the statement of Theorem 5.2.

The choice of the step size h in Theorem 5.2 follows directly from our analysis and is not likely to be optimal. Experiments suggest that the claimed iteration bound should hold even when h is a small constant (that does not depend on the input).

Our last result concerns an important special case of the basis pursuit problem: the undirected transshipment problem. In the undirected transshipment problem, we are given an undirected graph $G = (V, E)$, a demand vector $b \in \mathbb{Z}^V$ and a cost vector $c \in \mathbb{Z}^E_{>0}$. Assume that $\sum_v b_v = 0$ and let $B \in \mathbb{R}^{V\times E}$ be any signed incidence matrix of G. The goal is to find a flow vector $f \in \mathbb{R}^E$ which satisfies the demand ($Bf = b$) and minimizes the cost of the unsigned flow: $\sum_{e\in E} c_e|f_e|$. Note that in the basis pursuit problem the cost vector is always the all-one vector, however one can easily derive the corresponding meta-algorithm (and Physarum dynamics and IRLS) for the non-uniform case. The only difference in the meta-algorithm is that the $q^{(k)}$ vector is now computed as $q^{(k)} := q\left(\frac{w_1^{(k)}}{c_1}, \ldots, \frac{w_m^{(k)}}{c_m}\right) = q(C^{-1}w^{(k)})$.

In the below theorem we show that an even better bound holds on the number of iterations than it would follow from Theorem 5.2. For an instance (G, b, c), where G is an n-node undirected graph, we denote by $b_P = \|b\|_1$ the total demand and $c_{\max} = \max_{e\in E} c_e$ is the maximum cost. Let also $f^\star \in \mathbb{R}^E$ be any optimal solution to this instance.

Theorem 5.3 (*Iteration Bound for the Undirected Transshipment Problem*)

Let (G, b, c) be an instance of the undirected transshipment problem. Choose the initial solution to be $w_e^{(0)} = b_P$ for every $e \in E$. Then for every $\varepsilon > 0$, by picking the step size $h := \varepsilon/10nc_{\max}$ after $k = O\left(nc_{\max}\ln(nCb_P)/\varepsilon^3\right)$ iterations of the meta-algorithm we have $\sum_{e\in E} c_e\left|f_e^\star\right| \leqslant c^\top w^{(k)} \leqslant (1+\varepsilon)\sum_{e\in E} c_e\left|f_e^\star\right|$.

By comparing the above to Theorem 5.2 for the case when $c \equiv 1$, one can see that up to logarithmic factors, the number of iterations is linear in n, as opposed to linear in m, which is a significant speed-up for dense graphs. This improvement is obtained by taking advantage of the fact that the weighted ℓ_2-minimization problem underlying the Physarum dynamics coincides with the notion of electrical flows. Then, using properties of electrical flows we manage to obtain better bounds on the so-called potential differences on edges and as effect arrive at an improved iteration bound. Also importantly the Spielman-Teng solver [ST04] allows us to execute each iteration in $\tilde{O}(m)$ time giving yet another instance of the "Laplacian paradigm" [Spi12, Ten10, Vis12].

5.2 Proofs

5.2.1 Structural Results

In this section some structural results regarding the weighted ℓ_2-minimization and the behavior of the Meta Algorithm are presented. The first fact establishes a certain useful geometric condition that is maintained by the Meta-Algorithm throughout all steps of its execution, whenever it is initialized to satisfy it.

Fact 5.1 (*Positivity and boundedness*)

Suppose $\left((y^{(k)}, w^{(k)})\right)_{k\in\mathbb{N}} := \mathrm{MA}\left[A, b, h, y^{(0)}, w^{(0)}\right]$ *is a sequence produced by the Meta-Algorithm for some* $h \in (0,1)$*. If* $w^{(0)} > 0$ *and* $\left|y^{(0)}\right| \leqslant w^{(0)}$ *then* $w^{(k)} > 0$ *and* $\left|y^{(k)}\right| \leqslant w^{(k)}$ *for every* $k \in \mathbb{N}$*.*

Proof.
The proof proceeds by induction. When $k = 0$, the claim is valid by the assumption on $y^{(0)}$ and $w^{(0)}$. For $k \geqslant 0$, we have

$$w_i^{(k+1)} = (1-h)w_i^{(k)} + h\left|q_i^{(k)}\right| > 0$$

because $h \in (0,1)$. Similarly

$$\begin{aligned}\left|y_i^{(k+1)}\right| &= \left|(1-h)y_i^{(k)} + hq_i^{(k)}\right| \\ &\leqslant (1-h)\left|y_i^{(k)}\right| + h\left|q_i^{(k)}\right| \\ &\leqslant (1-h)w_i^{(k)} + h\left|q_i^{(k)}\right| \\ &= w_i^{(k+1)}.\end{aligned}$$

□

The next fact summarizes some simple but useful properties of the weighted ℓ_2-minimization problem.

Fact 5.2 (*Unique solution and its norm*)

Let $A \in \mathbb{R}^{n\times m}$ *be a matrix of rank* n*,* $b \in \mathbb{R}^n$ *and* $w \in \mathbb{R}^m_{>0}$*. Then, there exists a unique solution* $q = q(w) \in \mathbb{R}^m$ *to the weighted* ℓ_2*-minimization problem* (5.2) *and it is given by*

$$q = WA^\top(AWA^\top)^{-1}b.$$

Moreover, its weighted ℓ_2*-norm satisfies*

$$\sum_{i=1}^{m} \frac{q_i^2}{w_i} = b^\top L^{-1} b,$$

where $L := AWA^\top \in \mathbb{R}^{n\times n}$ *is invertible.*

Proof.
To prove the first claim, consider the strictly convex function $f : \mathbb{R}^m \to \mathbb{R}$ given by $f(x) := \sum_{i=1}^m \frac{x_i^2}{w_i}$. The optimality conditions for the convex program $\min\{f(x) : Ax = b\}$ are then given by

$$\begin{cases} AWA^\top\lambda = b \\ x = WA^\top\lambda \end{cases},$$

where $\lambda \in \mathbb{R}^n$ are Lagrangian multipliers for the linear constraints $Ax = b$. We claim that $L = AWA^\top$ is a full rank matrix. Indeed, suppose that $u \in \mathbb{R}^n$ is such that $Lu = 0$, then

$$0 = u^\top Lu = u^\top AWA^\top = \|W^{1/2}A^\top u\|^2,$$

hence $u = 0$, since A, and consequently $W^{1/2}A$, have rank n. For this reason L is invertible and the optimizer is explicitly given by the expression $WA^\top(AWA^\top)^{-1}b$.

The proof of the second claim starts with the formula $q = WA^\top L^{-1}b$ established in the first claim. Thus, $\sum_{i=1}^m \frac{q_i^2}{w_i} = q^\top W^{-1}q$ and, hence:

$$\begin{aligned} q^\top W^{-1}q &= b^\top L^{-1}AWW^{-1}WA^\top L^{-1}b \\ &= b^\top L^{-1}(AWA^\top)L^{-1}b \\ &= b^\top L^{-1}b. \end{aligned}$$

□

The following lemma plays an important role in the proof of Theorem 5.2. The fact that the upper bound is a constant (α) ends up being one of the main reasons why one can prove the convergence of the Meta-Algorithm for a *fixed* h (which, in turn, depends on α). Below, the columns of $A \in \mathbb{R}^{n\times m}$ are denoted by $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$.

Lemma 5.1 (*Uniform upper bound*)

Let $A \in \mathbb{R}^{n\times m}$ be a matrix of rank n and $w \in \mathbb{R}^m_{>0}$ be a weight vector. Then

$$\forall i, j \in \{1, 2, \ldots, m\} \quad w_i \cdot |a_i^\top L^{-1} a_j| \leqslant \alpha$$

where $L := AWA^\top$ and $\alpha \in \mathbb{R}$ is a constant that depends only on A.

The above follows directly from Lemma 4.1. The following its corollary is used multiple times in our proofs.

Corollary 5.1

Let $A \in \mathbb{R}^{n\times m}$ be a matrix of rank n, let $w \in \mathbb{R}^m_{>0}$ be a weight vector, and let $q = q(w)$ be the corresponding weighted ℓ_2-minimizer. Let $y \in \mathbb{R}^m$ be any solution to $Ay = b$ such that $\frac{|y_i|}{w_i} \leqslant K$ for every $i = 1, 2, \ldots, m$. Then

$$\forall i \in \{1, 2, \ldots, m\} \qquad \frac{|q_i|}{w_i} \leqslant K \cdot \alpha \cdot m,$$

where $\alpha \in \mathbb{R}$ is a constant that depends only on A.

Proof.
Let $L = AWA^\top$ (note that both L and L^{-1} are symmetric matrices), then $q = WA^\top L^{-1}b$ and hence

$$\frac{|q_i|}{w_i} = \left|a_i^\top L^{-1} b\right|.$$

Since $Ay = b$, we obtain

$$\begin{aligned}
\frac{|q_i|}{w_i} &= \left|a_i^\top L^{-1} A y\right| \\
&\leqslant \sum_{j=1}^{m} \left|y_j \cdot a_i^\top L^{-1} a_j\right| \\
&= \sum_{j=1}^{m} |y_j| \cdot \left|a_j^\top L^{-1} a_i\right| \\
&\overset{\text{Lemma 5.1}}{\leqslant} \sum_{j=1}^{m} |y_j| \cdot \frac{\alpha}{w_i^{(k)}} \\
&\leqslant K \cdot \alpha \cdot m.
\end{aligned}$$

□

Remark 5.2

It can be shown that if A is a matrix with integer entries then α (as in Lemma 5.1 and Corollary 5.1) can be chosen to be

$$\mathcal{D} := \max\left\{|\det(A')| : A' \text{ is a square submatrix of } A\right\};$$

see Lemma 4.1 in Chapter 4. We use this value in some of our bounds.

5.2.2 Equivalence of Physarum Dynamics and IRLS

In this section we present a proof of Theorem 5.1. The proof has two parts that are established in the two subsequent subsections. While the first is rather trivial, the second takes some effort as it requires establishing several technical facts about the Physarum dynamics.

IRLS as the Meta-Algorithm for $h = 1$

Proof of Theorem 5.1, Part 1:
When $h = 1$, the Meta-Algorithm proceeds as follows:

$$\left(y^{(k+1)}, w^{(k+1)}\right) = \left(q^{(k)}, \left|q^{(k)}\right|\right),$$

where $q^{(k)} = q(w^{(k)})$. Therefore, at each step, $w^{(k)} = \left|y^{(k)}\right|$. In particular, the dynamics is guided only by the y variables and is given by $y^{(k+1)} = q(|y^{(k)}|)$. This is exactly the same update rule as IRLS whose iterations correspond to $z^{(k+1)} = q(|z^{(k)}|)$.

□

Physarum Dynamics as the limiting case ($h \to 0$) of the Meta-Algorithm

Fix an instance of the basis pursuit problem: $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$. Note that by Fact 5.1, the vector $\varphi(t)$ defined in (5.3) can be equivalently characterized as $q(\sigma(t))$. Let $G(\sigma) = |q(\sigma)| - \sigma$

so that the Physarum dynamics can then be written compactly as

$$\frac{d}{dt}\sigma(t) = G(\sigma(t)). \tag{5.5}$$

In this section, we use the ℓ_2-norm to measure lengths of vectors and, hence, $\|\sigma\|$ should be understood as $\sqrt{\sum_{i=1}^{m} \sigma_i^2}$. We now state some technical lemmas whose proofs have been deferred to Section 5.2.2.

Lemma 5.2 (*Properties of Physarum trajectories*)

Let $T \in \mathbb{R}_{>0}$ and $\sigma : [0, T) \to \mathbb{R}^m_{>0}$ be any solution to the Physarum dynamics (5.5).

1. *For every $t \in [0, T)$ and for all $i \in \{1, 2, \ldots, m\}$, $\sigma_i(t) \geqslant \sigma_i(0)e^{-t}$.*
2. *The solution stays in a bounded region, i.e., $\sup_{t\in[0,T)} \|\sigma(t)\| < \infty$.*
3. *The limit $\lim_{t\to T^-} \sigma(t)$ exists and is a point in $\mathbb{R}^m_{>0}$.*

The next lemma implies the existence of a global solution of Physarum trajectories for all valid starting points.

Lemma 5.3 (*Existence of global solution*)

For every initial condition $\sigma(0) \in \mathbb{R}^m_{>0}$ there exists a global solution $\sigma : [0, +\infty) \to \mathbb{R}^m_{>0}$ to the Physarum dynamics (5.5).

The final lemma, whose proof relies on the previous lemmas, implies the proof of the second part of Theorem 5.1 trivially.

Lemma 5.4 (*Error analysis*)

Let $\sigma : [0, T] \to \mathbb{R}^m_{>0}$ be a solution to (5.5) *for $T \in \mathbb{R}_{\geqslant 0}$ and let $\left(w^{(k)}\right)_{k\in\mathbb{N}}$ be the sequence of weights produced by the Meta-Algorithm when initialized at $w^{(0)} = \sigma(0)$ with some step size $h \in (0, 1)$. Then, there exists a constant $K > 0$ such that for every small enough $h > 0$ and $k := \lfloor \frac{T}{h} \rfloor$ it holds that*

$$\|w^{(k)} - \sigma(T)\| \leqslant Kh.$$

Proof.
Fix any solution $\sigma : [0, T] \to \mathbb{R}^m_{>0}$ and let

$$\varepsilon := \min\left\{ \frac{\sigma_i(0)e^{-T}}{2} : i = 1, 2, \ldots, m \right\}.$$

Take K to be the closed ε-neighborhood of $\{\sigma(t) : t \in [0, T]\}$, i.e., the set of all points of distance at most ε from any point on the solution curve. Note that by Lemma 5.2, K is a compact subset of $\mathbb{R}^m_{>0}$. Let $L_1, L_2 > 0$ be constants such that

$$\|G(x) - G(y)\| \leqslant L_1\|x - y\| \qquad \text{for all } x, y \in K,$$
$$\|\sigma(t) - \sigma(s)\| \leqslant L_2 \,|t - s| \qquad \text{for all } t, s \in [0, T].$$

Such an L_1 exists because G is locally Lipschitz. To see why L_2 exists one can use the fact that G is bounded on K (as a continuous function on a compact domain) together with the formula

$$\sigma(t) - \sigma(s) = \int_s^t G(\sigma(\tau))d\tau.$$

We claim that for every $h \in (0,1)$ and $k \in \mathbb{N}$ such that $h \cdot k \leqslant T$ and $hL_2e^{L_1T} \leqslant \varepsilon$,

$$\|w^{(k)} - \sigma(hk)\| \leqslant hL_2e^{L_1T}.$$

The above claim is enough to conclude the proof. Towards its proof, first define

$$d_h(\ell) := \|w^{(\ell)} - \sigma(h\ell)\|$$

for any $h \in (0,1)$ and $\ell \in \mathbb{N}$. Recall that

$$w^{(\ell+1)} = w^{(\ell)} + hG(w^{(\ell)}).$$

Assuming that $w^{(\ell+1)} \in K$, we obtain

$$\begin{aligned} d_h(\ell+1) &= \|\sigma((\ell+1)h) - w^{(\ell+1)}\| \\ &= \|\sigma(\ell h) - w^{(\ell)} + \sigma((\ell+1)h) - \sigma(\ell h) - hG(w^{(\ell)})\| \\ &\leqslant d_h(\ell) + \|\sigma((\ell+1)h) - \sigma(\ell h) - hG(w^{(\ell)})\|. \end{aligned}$$

Next, we analyze the error term:

$$\begin{aligned} \|\sigma((\ell+1)h) - \sigma(\ell h) - hG(w^{(\ell)})\| &= \|\int_{\ell h}^{(\ell+1)h} G(\sigma(\tau))d\tau - hG(w^{(\ell)})\| \\ &= \|\int_{\ell h}^{(\ell+1)h} \Big[G(\sigma(\tau)) - G(w^{(\ell)})\Big]\, d\tau\| \\ &\leqslant \int_{\ell h}^{(\ell+1)h} \|G(\sigma(\tau)) - G(w^{(\ell)})\| d\tau \\ &\leqslant \int_{\ell h}^{(\ell+1)h} L_1\|\sigma(\tau) - w^{(\ell)}\| d\tau. \end{aligned}$$

To bound the distance $\|\sigma(\tau) - w^{(\ell)}\|$ for any $\tau \in [\ell h, (\ell+1)h]$, note that

$$\|\sigma(\tau) - w^{(\ell)}\| \leqslant \|\sigma(\tau) - \sigma(\ell h)\| + \|\sigma(\ell h) - w^{(\ell)}\| \leqslant hL_2 + d_h(\ell).$$

Altogether, we obtain the following recursive bound on $d_h(\ell+1)$

$$d_h(\ell+1) \leqslant d_h(\ell) + hL_1(hL_2 + d_h(\ell)) = d_h(\ell)(1 + hL_1) + h^2L_1L_2.$$

By expanding the above expression, one can show that

$$d_h(\ell) \leqslant h^2L_1L_2\sum_{i=0}^{\ell-1}(1+hL_1)^i \leqslant hL_2(1+hL_1)^\ell \leqslant hL_2e^{h\ell L_1}.$$

In particular, whenever $h \cdot \ell \leqslant T$, we obtain that $d_h(\ell) \leqslant hL_2e^{tL_1}$. Note that the above derivation is correct under the assumption that all the points $w^{(0)}, w^{(1)}, \ldots, w^{(\ell)}$ belong to K, however this is implied by the assumption that h is small: $hL_2e^{L_1T} \leqslant \varepsilon$. □

Technical lemmas and their proofs

Lemma 5.5 (*Local Lipschitzness*)

The function $G : \mathbb{R}^m_{>0} \to \mathbb{R}^m$ is locally Lipschitz, i.e., for every compact subset K of $\mathbb{R}^m_{>0}$, the restriction $G\restriction_K$ is Lipschitz.

Proof.
Fix any compact subset $K \subseteq \mathbb{R}^m_{>0}$. Since $G(\sigma) = |q(\sigma)| - \sigma$ and the identity function is Lipschitz, it is enough to prove that $\sigma \mapsto |q(\sigma)|$ is Lipschitz on K. It follows from Fact 5.2 that

$$q(\sigma) = \Sigma A^\top (A \Sigma A^\top)^{-1} b$$

and, hence, Cramer's rule implies that $q_i(\sigma)$ is a rational function of the form $\frac{Q_i(\sigma)}{\det(A\Sigma A^\top)}$ where $Q_i(\sigma)$ is a polynomial. Since $\det(A\Sigma A^\top)$ is positive for $\sigma \in \mathbb{R}^m_{>0}$, $q_i(\sigma)$ is a continuously differentiable function on $\mathbb{R}^m_{>0}$. Further, since K is a compact set, the magnitude of the derivative of q is upper bounded by a finite quantity, i.e.,

$$\sup_{\sigma \in K} \|\nabla q_i(\sigma)\| \leqslant C \qquad \text{for all } i = 1, 2, \ldots, m$$

for some $C \in \mathbb{R}$. Now, for any $x, y \in K$ and any $i \in \{1, 2, \ldots, m\}$:

$$q_i(x) - q_i(y) = \int_0^1 \langle \nabla q_i(y + t(x - y)), x - y \rangle \, dt$$

and, thus, using the Cauchy-Schwarz inequality it follows that

$$|q_i(x) - q_i(y)| \leqslant \|x - y\| \cdot \sup_{\sigma \in [x,y]} \|\nabla q_i(\sigma)\| \leqslant C\|x - y\|.$$

Thus, the Lipschitz constant of q on K is at most $\sqrt{m} \cdot C$. □

Proof of Lemma 5.2:
The first claim follows directly from Gronwall's inequality (see Section 2.3 in [Per01]) since

$$\frac{d}{dt}\sigma_i(t) = |q_i(\sigma(t))| - \sigma_i(t) \geqslant -\sigma_i(t).$$

For the second claim, it is enough to show that there exists a constant $C_1 > 0$ such that

$$\frac{|q_i(\sigma(t))|}{\sigma_i(t)} \leqslant C_1 \tag{5.6}$$

for all $t \in [0, T]$ and $i = 1, 2, \ldots, m$. Indeed, Gronwall's inequality then implies that $\sigma_i(t) \leqslant \sigma_i(0)e^{C_1 t}$, for all $t \in [0, T]$. Towards the proof of (5.6), let $y \in \mathbb{R}^m$ be any fixed solution to

$Ay = b$. From the first claim of Lemma 5.2, for every $t \in [0, T]$ and $i = 1, 2, \ldots, m$ we obtain

$$\frac{|y_i|}{\sigma_i(t)} \leqslant |y_i|\, \sigma_i(0)^{-1} e^t \leqslant |y_i| \sigma_i(0)^{-1} e^T.$$

Hence, by Corollary 5.1, $\frac{|q_i(\sigma(t))|}{\sigma_i(t)}$ is upper bounded by a quantity C_1 that depends on $T, \sigma(0), A, b$ only; proving the second claim.

The last claim follows from the previous two and the continuity G. Indeed, one can deduce that the solution curve $\{\sigma(t) : t \in [0, T]\}$ is contained in a compact set $K \subseteq \mathbb{R}^m_{>0}$. Denote $C_2 := \max \{\|G(\sigma)\| : \sigma \in K\}$. For any $s, t \in [0, T]$

$$\|\sigma(t) - \sigma(s)\| = \|\int_s^t G(\sigma(\tau)) d\tau\| \leqslant C_2 \, |s - t| \, .$$

The above readily implies that $\lim_{t \to T^-} \sigma(t)$ exists. Since K is a compact set, the limit $\sigma(T)$ belongs to K and thus $\sigma(T) \in \mathbb{R}^m_{>0}$. □

Proof of Lemma 5.3:
By Lemma 5.5, the function $G(\sigma)$ is locally Lipschitz and, hence, there is a maximal interval of existence $[0, T)$ of a solution $\sigma(t)$ to (5.5) where $0 < T \leqslant +\infty$ (see Theorem 1, Section 2.4 in [Per01]). Suppose that $x : [0, T) \to \mathbb{R}^m$ is a solution with $T \in \mathbb{R}_{>0}$. We show that it can be extended to a strictly larger interval $[0, T + \varepsilon)$ for some $\varepsilon > 0$. Let $\sigma(T) := \lim_{t \to T^-} \sigma(t)$; the limit exists and $\sigma(T) \in \mathbb{R}^m_{>0}$ by Lemma 5.2. Since $G(\sigma)$ is locally Lipschitz, one can apply the Fundamental Existence-Uniqueness theorem (see Section 2.2 in [Per01]) to obtain a solution $\tau : (T - \varepsilon, T + \varepsilon) \to \mathbb{R}^m_{>0}$ with $\tau(T) = \sigma(T)$ for some $\varepsilon > 0$. Because of uniqueness, τ and σ agree on $(T - \varepsilon, T]$ and, hence, can be combined to yield a solution on a larger interval: $[0, T + \varepsilon)$. This concludes the proof of the lemma. □

5.2.3 Convergence and Complexity of the Meta-Algorithm

In this section we present a proof of Theorem 5.2. In the rest of this section, we assume that the starting vectors $(y^{(0)}, w^{(0)})$ satisfy the condition stated in Theorem 5.2. It then follows from Fact 5.1 that $w^{(k)} > 0$ and $\left|y^{(k)}\right| \leqslant w^{(k)}$ for every $k \in \mathbb{N}$. Our goal is to prove that $y^{(k)}$ approaches an optimal solution to the problem (5.1). Since $Ay^{(k)} = b$ for every k, the proof reduces to showing that $\|y^{(k)}\|_1 \to \|x^\star\|_1$. Towards this goal, we first introduce the potential functions that are used in Section 5.2.3. Subsequently, we show how these potential functions can be used to explain how the vector $y^{(k)}$ moves towards optimality. The analysis has two cases: one when the energy is significantly smaller than the cost (see Section 5.2.3), and one where the energy is higher than the optimal value (see Section 5.2.3).

Potential functions

Cost. We call $\|w^{(k)}\|_1$ the *cost* of the current solution. It follows from Fact 5.1 that $\left|y^{(k)}\right| \leqslant w^{(k)}$. Hence, proving that $\|w^{(k)}\|_1 \to \|x^\star\|_1$ implies the same for $y^{(k)}$. We show, in particular, that the cost decreases with k. To reason about the rate at which it decreases two additional potential functions are required.

Energy. The *energy* of the current solution is defined to be

$$E(k) := \sum_{i=1}^{m} \frac{\left(q_i^{(k)}\right)^2}{w_i^{(k)}}.$$

It corresponds to the optimal value of the ℓ_2-minimization problem that is solved at step k.
Entropy. The *relative entropy* of the current solution with respect to the optimal one, or the generalized Kullback-Leibler divergence between $x^\star$ and $w^{(k)}$, is denoted by

$$I(k) := D_{KL}\left(x^\star, w^{(k)}\right) = \sum_{i=1}^{m} x_i^\star \ln \frac{x_i^\star}{w_i^{(k)}} - \sum_{i=1}^{m} x_i^\star + \sum_{i=1}^{m} w_i^{(k)}.$$

While the proof idea and its intuitive meaning is best understood in terms of $I(k)$, it is more convenient to use a simplified variant of $I(k)$. Dropping the constant terms and the $\|w^{(k)}\|_1$ term, we arrive at the potential

$$\mathcal{B}(k) := \sum_{i=1}^{m} x_i^\star \ln w_i^{(k)}$$

which we call the *barrier*, as it resembles the logarithmic barrier function used in interior point methods; see [Wri97].

Case 1: Cost is far from energy

Lemma 5.6

For every $k \in \mathbb{N}$ it holds that $\|w^{(k+1)}\|_1 \leqslant \|w^{(k)}\|_1$. Furthermore, if for some $\varepsilon \in \left(0, \frac{1}{2}\right)$, $\|w^{(k)}\| > \left(1 + \frac{\varepsilon}{3}\right) E(k)$, then $\|w^{(k+1)}\| \leqslant \left(1 - \frac{h\varepsilon}{8}\right) \|w^{(k)}\|$.

Proof.
Start by noting that

$$\begin{aligned}\|w^{(k)}\|_1 - \|w^{(k+1)}\|_1 &= h \sum_{i=1}^{m} \left(w_i^{(k)} - \left|q_i^{(k)}\right|\right) \\ &= h\left(\|w^{(k)}\|_1 - \|q^{(k)}\|_1\right).\end{aligned}$$

Furthermore,

$$\|q^{(k)}\|_1 = \sum_{i=1}^{m} \left|q_i^{(k)}\right| = \sum_{i=1}^{m} \sqrt{w_i^{(k)}} \frac{|q_i^{(k)}|}{\sqrt{w_i^{(k)}}}.$$

Thus, by applying the Cauchy-Schwarz inequality, we obtain

$$\sum_{i=1}^{m} \sqrt{w_i^{(k)}} \frac{|q_i^{(k)}|}{\sqrt{w_i^{(k)}}} \leqslant \|w^{(k)}\|_1^{1/2} \cdot E(k)^{1/2}.$$

Consequently, the following holds:

$$h\left(\|w^{(k)}\|_1 - \|q^{(k)}\|_1\right) \geqslant h\|w^{(k)}\|_1^{1/2}\left(\|w^{(k)}\|_1^{1/2} - E(k)^{1/2}\right).$$

Since $q^{(k)}$ minimizes the weighted ℓ_2-norm over the subspace $Ax = b$, it follows that:

$$E(k) = \sum_{i=1}^{m} \frac{\left(q_i^{(k)}\right)^2}{w_i^{(k)}} \leqslant \sum_{i=1}^{m} \frac{\left(y_i^{(k)}\right)^2}{w_i^{(k)}} \leqslant \sum_{i=1}^{m} \frac{\left(w_i^{(k)}\right)^2}{w_i^{(k)}} = \|w^{(k)}\|_1.$$

This establishes the first part of the lemma. Assume now that $\|w^{(k)}\| > \left(1 + \frac{\varepsilon}{3}\right) E(k)$. As a consequence,

$$\|w^{(k)}\|_1 - \|w^{(k+1)}\|_1 \geqslant h\|w^{(k)}\|_1^{1/2}\left(\|w^{(k)}\|_1^{1/2} - E(k)^{1/2}\right) \geqslant h\left(1 - \left(1 + \frac{\varepsilon}{3}\right)^{-1/2}\right)\|w^{(k)}\|_1.$$

To complete the proof of the lemma, it remains to note that $1 - \left(1 + \frac{\varepsilon}{3}\right)^{-1/2} \geqslant \frac{\varepsilon}{8}$. □

Case 2: Energy is far from the optimal value

To track the convergence process for steps when the cost is close to energy we use the barrier potential $\mathcal{B}(k)$. The following lemma characterizes its behavior.

Lemma 5.7

Suppose that $h \leqslant \frac{\varepsilon}{20 \cdot m \cdot \mathcal{D}}$, then for every k it holds that

$$\mathcal{B}(k+1) \geqslant \mathcal{B}(k) + h\left(\left(1 - \frac{\varepsilon}{10}\right) E(k) - \left(1 + \frac{\varepsilon}{10}\right)\|x^\star\|_1\right).$$

Here $\mathcal{D}$ is as defined in Equation (5.4).

The proof uses the following simple inequality

$$\forall x \in [-1/2, 1/2] \qquad x - x^2 \leqslant \ln(1+x) \leqslant x. \tag{5.7}$$

Proof.
Consider the change in the barrier potential:

$$\begin{aligned}\mathcal{B}(k+1) - \mathcal{B}(k) &= \sum_{i=1}^{m} x_i^\star \ln \frac{w_i^{(k+1)}}{w_i^{(k)}} \\ &= \sum_{i=1}^{m} x_i^\star \ln\left(1 + h\left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}} - 1\right)\right).\end{aligned}$$

We apply the left hand side of (5.7) to every summand. This is possible by the assumption that

$x^\star \geqslant 0$. For simplicity let $z_i := \left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}} - 1\right)$. Thus, we obtain that

$$\begin{aligned}\mathcal{B}(k+1) - \mathcal{B}(k) &\geqslant \sum_{i=1}^{m} x_i^\star (h z_i - h^2 z_i^2) \\ &= h \sum_{i=1}^{m} x_i^\star z_i - h^2 \sum_{i=1}^{m} x_i^\star z_i^2.\end{aligned} \tag{5.8}$$

The linear and the quadratic terms are analyzed separately. For the linear term, note that

$$h \sum_{i=1}^{m} x_i^\star z_i = h \sum_{i=1}^{m} x_i^\star \left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}}\right) - h\|x^\star\|_1.$$

Henceforth, for brevity, denote the sum $\sum_{i=1}^{m} x_i^\star \left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}}\right)$ by $\widetilde{E}(k)$. Then the above linear term becomes $h \cdot \widetilde{E}(k) - h \cdot \|x^\star\|_1$. To analyze the quadratic term in (5.8), we apply Corollary 5.1 to obtain:

$$\begin{aligned}\sum_{i=1}^{m} h^2 x_i^\star z_i^2 &\leqslant h^2 \cdot |m\mathcal{D} + 1| \cdot \sum_{i=1}^{m} x_i^\star |z_i| \\ &\leqslant h \cdot \frac{\varepsilon}{10} \cdot \sum_{i=1}^{m} x_i^\star \left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}} + 1\right) \\ &= h \cdot \frac{\varepsilon}{10} \widetilde{E}(k) + h \cdot \frac{\varepsilon}{10} \|x^\star\|_1.\end{aligned}$$

Combining the linear and quadratic order bounds, we obtain:

$$\mathcal{B}(k+1) - \mathcal{B}(k) \geqslant h\left(1 - \frac{\varepsilon}{10}\right)\widetilde{E}(k) - h\left(1 + \frac{\varepsilon}{10}\right)\|x^\star\|_1.$$

To complete the proof, it suffices to show that $\widetilde{E}(k) \geqslant E(k)$. Towards this, note that

$$\begin{aligned}\sum_{i=1}^{m} x_i^\star \left(\frac{\left|q_i^{(k)}\right|}{w_i^{(k)}}\right) &\geqslant \sum_{i=1}^{m} x_i^\star \frac{q_i^{(k)}}{w_i^{(k)}} \\ &= (x^\star)^\top \left(W^{(k)}\right)^{-1} q^{(k)} = (x^\star)^\top \left(W^{(k)}\right)^{-1} W^{(k)} A^\top L^{-1} b \\ &= (x^\star)^\top A^\top L^{-1} b = b^\top L^{-1} b\end{aligned}$$

where $L := AW^{(k)}A^\top$. The above, together with Fact 5.2, gives

$$\widetilde{E}(k) \geqslant b^\top L^{-1} b = E(k),$$

which concludes the proof of the lemma. □

Proof of Theorem 5.2

We would like to upper bound the number of steps until the first time when $\|w^{(k)}\|_1 \leqslant (1+\varepsilon)\|x^\star\|_1$. From Lemma 5.6, the ℓ_1-norm of $w^{(k)}$ is non-increasing with k and whenever $\|w^{(k)}\|_1 > \left(1+\frac{\varepsilon}{3}\right)E(k)$ (i.e., Case 1 occurs), $\|w^{(k)}\|_1$ decreases by a multiplicative factor of $(1-\frac{h\varepsilon}{8})$. This means that there can be at most

$$\frac{\ln\left(\frac{M}{1+\varepsilon}\right)}{\ln(1-h\varepsilon)^{-1}} = O\left(\frac{\ln M}{h\varepsilon}\right)$$

such steps. Consider a step for which $\|w^{(k)}\|_1 \leqslant \left(1+\frac{\varepsilon}{3}\right)E(k)$. We obtain:

$$(1+\varepsilon)\|x^\star\|_1 \leqslant \|w^{(k)}\|_1 \leqslant \left(1+\frac{\varepsilon}{3}\right)E(k).$$

This implies in particular that

$$E(k) \geqslant \left(1+\frac{\varepsilon}{2}\right)\|x^\star\|_1,$$

i.e., Case 2 occurs. We apply Lemma 5.7 to conclude that in this case

$$\mathcal{B}(k+1) \geqslant \mathcal{B}(k) + \frac{h\varepsilon}{5}\|x^\star\|_1.$$

We now analyze how $\mathcal{B}(k)$ changes. Start by observing that $\mathcal{B}(0) \geqslant 0$ (since $w_i^{(0)} \geqslant 1$ for every $i \in \{1,2,\ldots,m\}$) and $\mathcal{B}(k)$ is upper bounded by $\|x^\star\|_1 \cdot (\ln M + \ln\|x^\star\|_1)$ (this holds because $\|w^{(k)}\|_1 \leqslant \|w^{(0)}\|_1 \leqslant M\|x^\star\|_1$). At every step k for which $\|w^{(k)}\|_1 > \left(1+\frac{\varepsilon}{3}\right)E(k)$, $\mathcal{B}(k)$ drops by at most

$$h\left(1+\frac{\varepsilon}{10}\right)\|x^\star\|_1 \leqslant 2h\|x^\star\|_1$$

by Lemma 5.7. Note that by the reasoning above there are at most $O\left(\frac{\ln M}{h\varepsilon}\right)$ such steps. On the other hand, if $\|w^{(k)}\|_1 \leqslant \left(1+\frac{\varepsilon}{3}\right)E(k)$ then $\mathcal{B}(k)$ increases by at least $\frac{h\varepsilon}{5}\|x^\star\|_1$. This means that the total decrease of $\mathcal{B}(k)$ is at most

$$O\left(\frac{\ln M}{\varepsilon}\|x^\star\|_1\right).$$

Therefore, the number of steps in which $\|w^{(k)}\|_1 \leqslant \left(1+\frac{\varepsilon}{3}\right)E(k)$ is at most

$$O\left(\frac{\frac{\ln M}{\varepsilon}\|x^\star\|_1 + \|x^\star\|_1 \cdot (\ln M + \ln\|x^\star\|_1)}{\frac{h\varepsilon}{5}\|x^\star\|_1}\right)$$

which is bounded by $O\left(\frac{\ln M + \ln\|x^\star\|_1}{h\varepsilon^2}\right)$. This completes the proof of Theorem 5.2.

5.2.4 Iteration Bound for Undirected Transshipment Problem

The main component of our improved iteration bound for the transshipment problem is the following strenghtening of Corollary 5.1 for the case where the matrix A is an incidence matrix of a graph.

Lemma 5.8

Let $B \in \mathbb{R}^{V \times E}$ be an indcidence matrix of an undirected graph $G = (V, E)$, $b \in \mathbb{Z}^m$ be a demand vector and $c \in \mathbb{Z}^E_{>0}$ a cost vector. Let $w \in \mathbb{R}^E_{>0}$ be any weight vector, and let $q = q(C^{-1}w)$ be the corresponding weighted ℓ_2-minimizer. Let $y \in \mathbb{R}^E$ be any solution to $By = b$ such that $\frac{|y_e|}{w_e} \leqslant K$ for every $e \in E$. Then

$$\forall e \in E \qquad \frac{|q_e|}{w_e} \leqslant K \cdot c_{\max} \cdot n,$$

where $c_{\max} := \max_{e \in E} c_e$.

Proof.
By adapting the electrical network interpretation (see [Vis12]) of the weighted ℓ_2-minimization problem, observe that the vector $p \in \mathbb{R}^V$ defined as

$$p := (BC^{-1}WB^{\top})^{-1}b$$

defines node potentials for the electrical flow on the graph G, where the resistance of an edge e is $\frac{c_e}{w_e}$ for all $e \in E$. This vector is not unique but any two such vector differ by a multiple of the all-one vector, hence in particular the potential differences on edges: $p_u - p_v$ for $uv \in E$ are well defined.

The quantity of our interest is $\frac{|q_e|}{w_e}$ for an edge $e = uv \in E$ which is exactly

$$\frac{|q_e|}{w_e} = |p_u - p_v|\,.$$

Thus it is enough to prove that for any pair of vertices $u, v \in V$ (not necessarily connected by an edge) we have

$$|p_u - p_v| \leqslant c_{\max} \cdot K \cdot n.$$

For this, sort all the potentials in nondecreasing order and pick two neighbouring ones p_u, p_v. In other words, take u, v, such that $p_u \leqslant p_v$ and for all $w \in V$ either $p_w \leqslant p_u$ or $p_w \geqslant p_v$. We show that $p_v - p_u \leqslant K \cdot c_{\max}$.

Assume the contrary: $p_v - p_u > K \cdot c_{\max}$. Recall that $c_{\max}$ is the maximum cost c_e over all $e \in E$. We define a partition of V into two sets $S, \bar{S}$:

$$S = \{w \in V : p_w \leqslant p_u\}, \qquad \bar{S} = \{w \in V : p_w \geqslant p_v\}.$$

Let $E_S \subseteq E$ be the set of edges going between S and $\bar{S}$. Since the graph is connected we know that $E_S \neq \emptyset$. Since the potentials in $\bar{S}$ are higher than those in S, it is clear that q sends some non-zero flow from $\bar{S}$ to S, in other words $b_S = \sum_{v \in S} b_v > 0$. Moreover, no flow is going back from S to $\bar{S}$ (as it would violate the potentials) hence

$$\sum_{e \in E_S} |q_e| = b_S.$$

On the other hand:

$$\sum_{e=u_1v_1\in E_S} |q_e| = \sum_{e\in E_S} \left|\frac{(p_{u_1}-p_{v_1})w_e}{c_e}\right| > K\sum_{e\in E_S}\left|\frac{c_{\max}w_e}{c_e}\right|,$$

since for every $u_1 \in S$ and every $v_1 \in \bar{S}$ we have $|p_{u_1} - p_{v_1}| > K \cdot c_{\max}$. Note now that by our assumption there exists a solution $y \in \mathbb{R}^E$, i.e., $Ay = b$ such that for every $e \in E$ we have $|y_e| \leqslant K w_e$. Thus further we obtain:

$$b_S > K\sum_{e\in E_S}\left|\frac{c_{\max}w_e}{c_e}\right| \geqslant \sum_{e\in E_S}|y_e| \geqslant b_S,$$

since y being a valid flow implies that y sends at least b_S units of flow between S and $\bar{S}$. This contradiction concludes the proof. □

We are now ready to sketch the proof of Theorem 5.3.

Proof of Theorem 5.3:
The argumetn follows closely the argument for Theorem 5.2 presented in Section 5.2.3. In particular the same set of potential functions and the same two cases are considered in the convergence analysis. Below we highlight where do these two arguments differ, and hence where does the improvement come from.

Note first that by initializing the algorithm with $w^{(0)} \equiv b_P$ we make sure that there is a feasible solution $y^{(0)}$ such that $\left|y^{(0)}\right| \leqslant w^{(0)}$ coordinatewise – for example, simply the minimum cost solution. Thus it follows from Fact 5.1 that this is the case for all iterates.

The only significant change in the argument occurs in Case 2: when Energy is far from the optimal value. The quantity of interest there is

$$\sum_{e\in E}\ln\left(1 + h\left(\frac{\left|q_e^{(k)}\right|}{w_e^{(k)}} - 1\right)\right)$$

and in particular a bound on $\frac{\left|q_e^{(k)}\right|}{w_e^{(k)}}$ is required in order to use the approximation $\ln(1+x) \approx x$. Given the bound from Lemma 5.8 one can pick $h \approx \frac{1}{n\cdot c_{\max}}$ for this to hold. Indeed from what our above observation it follows that the constant K in Lemma 5.8 can be taken to be 1. Finally, in a later stage of this argument (for Case 2) one has to argue that

$$h\cdot\left(\frac{\left|q_e^{(k)}\right|}{w_e^{(k)}} - 1\right) \leqslant \frac{\varepsilon}{10}.$$

This follows by choosing $h = \Theta\left(\frac{\varepsilon}{n\cdot c_{\max}}\right)$. □

5.2.5 Example of Non-Convergence of IRLS

In this section, we present an example instance of the basis pursuit problem for which IRLS fails to converge to the optimal solution.

Theorem 5.4

There exists an instance (A,b) of the basis pursuit problem (5.1) *and a strictly positive point $y^{(0)} \in \mathbb{R}^m_{>0}$ (with $Ay^{(0)} = b$) such that if IRLS is initialized at $y^{(0)}$ (and $\{y^{(k)}\}_{k\in\mathbb{N}}$ is the sequence produced by IRLS) then $\|y^{(k)}\|_1$ does not converge to the optimal value.*

The proof is based on the simple observation that if IRLS reaches a point $y^{(k)}$ with $y_i^{(k)} = 0$ for some $k \in \mathbb{N}$, $i \in \{1, 2, \ldots, m\}$ then $y_i^{(l)} = 0$ for all $l > k$.

Consider an undirected graph $G = (V, E)$ with $V = \{u_0, u_1, ..., u_6, u_7\}$, $E = \{e_1, e_2, \ldots, e_9\}$, and also let $s = u_0$, $t = u_7$. G is depicted in Figure 5.1.

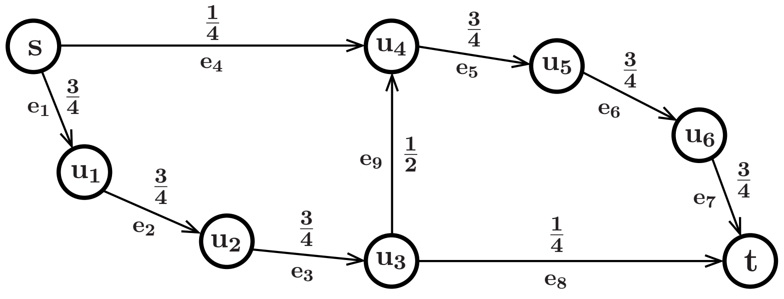

Figure 5.1: The graph G together with a feasible solution $y^{(0)} \in \mathbb{R}^V$.

We define $A \in \mathbb{R}^{8\times 9}$ to be the signed incidence matrix of G with edges directed according to increasing indices and let $b := (-1, 0, 0, 0, 0, 0, 0, 1)^\top$. Then the following problem

$$\min \|x\|_1 \qquad \text{s.t. } Ax = b$$

is equivalent to the shortest $s-t$ path problem in G. The linear system $Ax = b$ is stated explicitly in Figure 5.2. The unique optimal solution is the path $s - u_4 - u_3 - t$, i.e., $y^\star = (1, 0, 0, 0, 0, 0, 0, 1, -1)^\top$ (note that we work with undirected graphs here). In particular, the edge (u_3, u_4) is in the support of the optimal vector (it corresponds to the last coordinate of $y^\star$). Consider an initial solution $y^{(0)}$ given below

$$y^{(0)} = ({}^3\!/\!{}_4, {}^3\!/\!{}_4, {}^3\!/\!{}_4, {}^1\!/\!{}_4, {}^3\!/\!{}_4, {}^3\!/\!{}_4, {}^3\!/\!{}_4, {}^1\!/\!{}_4, {}^1\!/\!{}_2)^\top .$$

Claim 5.1

IRLS initialized at $y^{(0)}$ produces in one step a point $y^{(1)}$ with $y_9^{(1)} = 0$.

The above claim implies that IRLS initialized at $y^{(0)}$ (which has full support) does not converge to the optimal solution, which has 1 in the last coordinate. Thus to prove Theorem 5.4 it suffices to show Claim 5.1.

Proof of Claim 5.1:
IRLS chooses the next point $y^{(1)} \in \mathbb{R}^9$ according to the rule:

$$y^{(1)} = \operatorname*{argmin}_{y\in\mathbb{R}^9} \sum_{i=1}^{9} \frac{x_i^2}{y_i} \qquad \text{s.t. } Ax = b$$

$$\begin{array}{rrrrrrrrrcr}
x_1 + & & & +x_4 & & & & & & = & 1 \\
-x_1 + x_2 & & & & & & & & & = & 0 \\
& -x_2 + & x_3 & & & & & & & = & 0 \\
& & -x_3 + & & & & & x_8 + x_9 & & = & 0 \\
& & & - x_4 + x_5 & & & & - x_9 & & = & 0 \\
& & & & - x_5 + x_6 & & & & & = & 0 \\
& & & & & - x_6 + x_7 & & & & = & 0 \\
& & & & & & - x_7 - x_8 & & & = & -1
\end{array}$$

Figure 5.2: The linear system that encodes the shortest path problem in Figure 5.1

which is the same as the unit electrical $s-t$ flow in G corresponding to edge resistances $\frac{1}{y_e}$. (This is due to the fact that electrical flows minimize energy, see [Vis12].) In such an electrical flow the potentials of u_4 and u_3 are equal (the paths $s - u_4$ and $s - u_1 - u_2 - u_3$ have equal resistances), hence the flow through (u_3, u_4) is zero. $\square$

5.3 Conclusion

5.3.1 Future Work

In this chapter we have established convergence of a damped version of the IRLS algorithm. It is then natural to ask whether the standard version of this algorithm converges and what is its rate of convergence. The example presented in section 5.2.5 shows that it does not converge to optimal solutions for all instances. The following two variants of the convergence question give a way to bypass this negative example:

(1) Does the IRLS algorithm converge from *almost* every starting point? Formally: is the set of "bad" starting points of measure zero?

(2) Does a stochastic variant of IRLS converge? By a stochastic version we mean one which perturbs the point in every iteration by a small amount of Gaussian noise.

Another interesting question is whether the convergence result for the Physarum dynamics can be strenghtened to point-wise convergence. The current result gives only convergence in value, yet the analogous result for the directed dynamics, see Chapter 4, Theorem 4.2 proves pointwise convergence.

5.3.2 Notes

The content of this chapter is based on results obtained in [SV16a, SV17c]. The barrier potential $-\mathcal{B}(k)$ (as in the proof of Theorem 5.2) appeared first in the work of [MO07, MO08] for a special of the shortest path problem in undirected graphs. In the context of the Physarum dynamics, the energy potential was first studied in [BMV12].

Chapter 6

Estimation Algorithms for Counting and Optimization

In this chapter we study counting and optimization problems for measures represented by polynomials, as introduced in Chapter 3. In such a problem, a polynomial $p \in \mathbb{R}[x_1, x_2, \ldots, x_m]$ is given along with a family $\mathcal{B} \subseteq 2^{[m]}$ and the goal is to find $p_{\mathcal{B}} := \sum_{S \in \mathcal{B}} p_S$ (the sum of coefficients of p corresponding to sets in $\mathcal{B}$) or $\max_{S \in \mathcal{B}} p_S$ (the maximum coefficients over $\mathcal{B}$). We introduce $\mathrm{Cap}_{\mathcal{B}}(p)$: the notion of $\mathcal{B}$-capacity of a polynomial p and prove that it provides a decent approximation to the above counting problem whenever, roughly, the polynomial p is real stable and the family $\mathcal{B}$ is a family of bases of a matroid. Further, we extend this idea to the optimization problem and thus obtain estimation algorithms for both the counting and optimization problem in a fairly general setting.

6.1 Preliminaries and Statement of Results

6.1.1 Statement of Results

We start by formally defining the notion of $\mathcal{B}$-capacity of a polynomial.

Definition 6.1 (*$\mathcal{B}-$capacity*)

For a polynomial $g \in \mathbb{R}^+[x_1, \ldots, x_m]$ and any family of sets $\mathcal{B} \subseteq 2^{[m]}$ we define the $\mathcal{B}-$capacity of g to be

$$\mathrm{Cap}_{\mathcal{B}}(g) \stackrel{\text{def}}{=} \sup_{\theta \in P(\mathcal{B})} \inf_{z>0} \frac{g(z)}{\prod_{i=1}^{m} z_i^{\theta_i}}. \tag{6.1}$$

When $\mathcal{B} = \{\{1, 2, \ldots, m\}\}$ (i.e., when $\mathcal{B}$ contains just a single element) one recovers Gurvits' capacity function [Gur06] from (6.1). In Section 6.3.7 we show that, when g is real stable, $\mathrm{Cap}_{\mathcal{B}}(g)$ can be computed using a convex program. The running time of this algorithm depends upon the maximum number of bits required to represent any coefficient of g. When the coefficients of g correspond to a probability distribution on the set of all monomials $\mathrm{Cap}_{\mathcal{B}}(g)$ has a natural interpretation – it can be viewed as the optimal value of a certain entropy maximizing program; see Section 6.3.1. Importantly, an (equivalent) dual characterization of $\mathcal{B}-$capacity allows us to prove that $\mathrm{Cap}_{\mathcal{B}}(g)$ is an upper bound on $g_{\mathcal{B}}$, see Section 6.3.1.

Counting Our counting algorithm outputs $\mathrm{Cap}_{\mathcal{B}}(g)$ as an estimate for $g_{\mathcal{B}} := \sum_{S\in\mathcal{B}} g_S$. The next definition captures how good an approximation $\mathrm{Cap}_{\mathcal{B}}(g)$ is to $g_{\mathcal{B}}$.

Definition 6.2 (*Approximation Ratios*)

For a family $\mathcal{B} \subseteq \binom{[m]}{n}$ we define:

$$M(\mathcal{B}) \overset{\text{def}}{=} \sup\left\{\frac{\mathrm{Cap}_{\mathcal{B}}(p)}{p_{\mathcal{B}}} : p \in H^+(m,n) \text{ and } p_{\mathcal{B}} > 0\right\},$$
$$M_{\text{lin}}(\mathcal{B}) \overset{\text{def}}{=} \sup\left\{\frac{\mathrm{Cap}_{\mathcal{B}}(p)}{p_{\mathcal{B}}} : p \in H_1^+(m,n) \text{ and } p_{\mathcal{B}} > 0\right\},$$

where $H^+(m,n)$ is the set of all m-variate, n-homogeneous real stable polynomials with non-negative coefficients and $H_1^+(m,n)$ is $H^+(m,n)$ restricted to multi-linear polynomials.

It follows that $M_{\text{lin}}(\mathcal{B}) \leqslant M(\mathcal{B})$. The reason for the second definition is because in some applications, the polynomial g may be multi-linear and one can expect better bounds under such an assumption. Thus, along with the computability of $\mathrm{Cap}_{\mathcal{B}}(g)$, the above definitions imply the following result trivially.

Theorem 6.1 (*Counting*)

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of sets and let $g \in \mathbb{R}^+[x_1,\ldots,x_m]$ be any real stable, $n-$homogeneous polynomial. Given access to a separation oracle for $P(\mathcal{B})$ and an evaluation oracle for g, there is a polynomial time algorithm which computes $\mathrm{Cap}_{\mathcal{B}}(g)$ and, further,

$$\frac{1}{M(\mathcal{B})}\mathrm{Cap}_{\mathcal{B}}(g) \leqslant \sum_{S\in\mathcal{B}} g_S \leqslant \mathrm{Cap}_{\mathcal{B}}(g).$$

Additionally, when g is multi-linear, $M(\mathcal{B})$ can be replaced by $M_{\text{lin}}(\mathcal{B})$ in the bound above.

It remains to bound the approximation ratios $M(\mathcal{B})$ and $M_{\text{lin}}(\mathcal{B})$. We first study the finiteness of these quantities. We give a sufficient condition for $M(\mathcal{B})$ to be finite that relies on the interplay between matroids and supports of *strongly Rayleigh* distributions. A distribution μ over subsets of $[m]$ is called strongly Rayleigh if its generating polynomial $p_\mu(z) = \sum_{S\subseteq[m]} \mu(S)z^S$ is real stable.

Theorem 6.2 (*Finiteness of $M(\mathcal{B})$*)

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a family of bases of a matroid and let $\mathcal{B}^\star$ be the family of bases of the dual matroid. If there exists a strongly Rayleigh distribution supported on $\mathcal{B}^\star$, then $M(\mathcal{B}) < \infty$.

Interestingly, when one gives up either real stability of g or the assumption that $\mathcal{B}$ comes from a matroid, then $M(\mathcal{B})$ can be infinite; we provide examples in Section 6.3.4. For precise definitions of matroids, their duals, partition matroids, linear matroids and the *unbalance* $\mathrm{un}(\mathcal{M})$ of a matroid $\mathcal{M}$, we refer the reader to Section 6.1.2.

Theorem 6.3 (***Quantitative Bounds***)

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a family of bases of a matroid $\mathcal{M}$.

1. **Strongly Rayleigh.** *If there is a strongly Rayleigh distribution $\{p_S\}_{S\in\mathcal{B}^\star}$ (with $p_S > 0$ for every $S \in \mathcal{B}^\star$) and $P \overset{\text{def}}{=} \max_{S,T\in\mathcal{B}^\star} \frac{p_S}{p_T}$, then*
   - $M(\mathcal{B}) \leqslant P \cdot \frac{m^m}{m!}$,
   - $M_{\text{lin}}(\mathcal{B}) \leqslant P \cdot 2^m$.
2. **Linear Matroids.** *If $\mathcal{M}$ is $\mathbb{R}-$linear, then*
   - $M(\mathcal{B}) \leqslant \text{un}(\mathcal{M}^\star) \cdot \frac{m^m}{m!}$,
   - $M_{\text{lin}}(\mathcal{B}) \leqslant \text{un}(\mathcal{M}^\star) \cdot 2^m$.
3. **Partition Matroids.** *If $\mathcal{M}$ is a partition matroid induced by a partition $P_1 \cup \cdots \cup P_p = [m]$ and integers $b_1, \dots, b_p$ then*
   - $M(\mathcal{B}) \leqslant \frac{m^m}{m!} \cdot \prod_{j=1}^{p} \frac{(|P_j|-b_j)!}{(|P_j|-b_j)^{|P_j|-b_j}} \leqslant e^{n+p}\left(\frac{m}{p}\right)^{p/2}$,
   - $M_{\text{lin}}(\mathcal{B}) \leqslant \frac{n^n}{n!} \prod_{j=1}^{p} \frac{b_j!}{b_j^{b_j}}$.

Combining Theorem 6.1 and Theorem 6.3 one can obtain new and old algorithms that estimate $g_{\mathcal{B}}$ for a large class of matroids. For instance, regular matroids, being strongly Rayleigh and linear, satisfy both parts (1) and (2) of the theorem above with $P = 1$ and $\text{un}(\mathcal{M}^\star) = 1$ respectively, resulting in an e^m-approximation algorithm.

In Section 6.3.5 we provide an example where $M(\mathcal{B})$ can be as large as $e^{\sqrt{m}}$ for partition matroids.

Optimization Our main result on computing $\max_{S\in\mathcal{B}} g_S$ is the following

Theorem 6.4 (***Optimization***)

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of sets and let $g \in \mathbb{R}^+[x_1, \dots, x_m]$ be any real stable, $n-$homogeneous polynomial. Given access to a separation oracle for $P(\mathcal{B}) = \text{conv}\{1_S : S \in \mathcal{B}\}$ and an evaluation oracle for g, there is an algorithm which estimates the value of

$$\max_{S\in\mathcal{B}} g_S$$

up to a factor of $M(\mathcal{B}) \cdot e^n$. The running time of the algorithm is polynomial in m and the maximum number of bits required to represent any coefficient of g. In the case when $g \in \mathbb{R}_1^+[x_1, \dots, x_m]$, $M(\mathcal{B})$ can be replaced by $M_{\text{lin}}(\mathcal{B})$ in the bound above.

The following corollary concerns maximizing sub-determinants subject to matroid constraints.

Corollary 6.1 (***Sub-determinant Maximization***)

Let $L \in \mathbb{R}^{m\times m}$ be a PSD matrix and $\mathcal{B} \subseteq \binom{[m]}{n}$ be a family of sets. Given access to a separation oracle for $P(\mathcal{B})$ there is a polynomial time algorithm which estimates the value of $\max_{S\in\mathcal{B}} \det(L_{S,S})$ up to a factor of $M_{\text{lin}}(\mathcal{B}) \cdot e^n$.

6.1.2 Preliminaries

In this section we give the necessary background on matroids. We state the definitions and examples of matroids, which are most relevant to our results. For a comprehensive treatment of matroid theory we refer the reader to [Oxl06].

A matroid is a pair $(U, \mathcal{M})$ such that U is a finite set and $\mathcal{M} \subseteq 2^U$ satisfies the following three axioms: (1) $\emptyset \in \mathcal{M}$, (2) if $S \in \mathcal{M}$ and $S' \subseteq S$ then $S' \in \mathcal{M}$, (3) if $A, B \in \mathcal{M}$ and $|A| > |B|$, then there exists an element $a \in A \setminus B$ such that $B \cup \{a\} \in \mathcal{M}$. The collection $\mathcal{B} \subseteq \mathcal{M}$ of all inclusion-wise maximal elements of $\mathcal{M}$ is called the set of bases of the matroid. It is known that all the sets in $\mathcal{B}$ have the same cardinality, which is called the rank of the matroid.

Given a matroid $\mathcal{M} \subseteq 2^U$ with a set of bases $\mathcal{B}$ we define another collection of sets $\mathcal{B}^\star \subseteq 2^U$ to be

$$\mathcal{B}^\star \stackrel{\text{def}}{=} \{U \setminus S : S \in \mathcal{B}\}.$$

Then $\mathcal{B}^\star$ can be shown to be a collection of bases of another matroid $\mathcal{M}^\star$, called the dual of $\mathcal{M}$.

Linear and Strongly Rayleigh Matroids. We say that a matroid $\mathcal{M} \subseteq 2^{[m]}$ is $\mathbb{R}-$linear if there exists a matrix $V \in \mathbb{R}^{m\times n}$ (with rows $v_1, v_2, \ldots, v_m \in \mathbb{R}^n$) such that for every set $S \subseteq [m]$ we have $S \in \mathcal{M}$ if and only if the collection of vectors $\{v_j : j \in S\}$ is linearly independent over $\mathbb{R}$. Such a matrix V we call an $\mathbb{R}-$representation of the matroid $\mathcal{M}$. If $\mathcal{B}$ is a set of bases of $\mathcal{M}$ and $V \in \mathbb{R}^{m\times n}$ is a representation, we define the unbalance of V to be

$$\mathrm{un}(V) \stackrel{\text{def}}{=} \max\left\{\frac{\det(V_S^\top V_S)}{\det(V_T^\top V_T)} : S, T \in \mathcal{B}\right\},$$

where V_S is an $|S| \times n$ sub-matrix of V corresponding to rows from S. For an $\mathbb{R}-$linear matroid $\mathcal{M}$ with set of bases $\mathcal{B}$ we define $\mathrm{un}(\mathcal{M})$ (or equivalently $\mathrm{un}(\mathcal{B})$) to be the minimum $\mathrm{un}(V)$ over all $\mathbb{R}-$representations V of this matroid.

Matroids $\mathcal{M}$ which have a totally unimodular $\mathbb{R}$-representation are called regular, in such a case $\mathrm{un}(\mathcal{M}) = 1$. An example of such is the so called graphic matroid, whose universe U is the set of all edges of an undirected graph G and the family of bases corresponds to all subsets of edges which are spanning trees of G.

A matroid $\mathcal{M}$ with a set of bases $\mathcal{B}$ is called strongly Rayleigh if the polynomial $g(z) = \sum_{S\in\mathcal{B}} z^S$ is real stable. Regular matroids are examples of strongly Rayleigh matroids.

Partition and Uniform matroids. A matroid $\mathcal{M}$ is said to be a partition matroid if there exists a partition $\mathcal{P} = \{U_1, U_2, \ldots, U_t\}$ of the ground set U and a sequence of non-negative integers $b = (b_1, b_2, \ldots, b_t)$ such that every basis S of $\mathcal{M}$ satisfies $|S \cap U_i| = b_i$ for all $i = 1, 2, \ldots, t$. A uniform matroid is a partition matroid in which $t = 1$ and $U_1 = U$.

6.2 Technical Overview

We first describe the key steps of the proof of Theorem 6.2. The first step connects $\mathcal{B}-$capacity to the standard notion of capacity (see Lemma 6.1)

$$\mathrm{Cap}(g \cdot h) \geqslant \mathrm{Cap}_{\mathcal{B}}(g) \cdot \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h), \tag{6.2}$$

where $\underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h)$ denotes, what we call, the *lower $\mathcal{B}^\star-$capacity* of h – it is defined as

$$\underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h) \stackrel{\mathrm{def}}{=} \inf_{\theta \in P(\mathcal{B}^\star)} \inf_{z>0} \frac{h(z)}{\prod_{i=1}^m z_i^{\theta_i}}.$$

Note that in the definition of lower capacity, as compared to (6.1), the supremum is replaced by infimum.

Since inequality (6.2) holds for every polynomial h, one can upper-bound the $\mathcal{B}-$capacity by providing an appropriate h. The choice of h should ideally allow us to relate $\mathrm{Cap}(g \cdot h)$ to $g_{\mathcal{B}}$, as our primary goal is to upper bound the ratio $\frac{\mathrm{Cap}_{\mathcal{B}}(g)}{g_{\mathcal{B}}}$. To this end we present a notion of a $\mathcal{B}^\star-$selection (see Definition 6.4), which essentially describes sufficient conditions on h for this to succeed. Roughly, h being a $\mathcal{B}^\star-$selection means that the coefficient of $\prod_{i=1}^m x_i$ in $g \cdot h$ is equal to $g_{\mathcal{B}}$. Further, if h is a real stable $\mathcal{B}^\star-$selection, one can apply Gurvits' inequality [Gur06] along with inequality (6.2) to obtain

$$g_{\mathcal{B}} \geqslant \frac{m!}{m^m} \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h) \cdot \mathrm{Cap}_{\mathcal{B}}(g). \tag{6.3}$$

Thus, the task of proving a lower-bound on $g_{\mathcal{B}}$ reduces to that of coming up with a "good" $\mathcal{B}^\star-$selection h – an h whose lower capacity $\underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h)$ is as large as possible. Sections 6.3.2, 6.3.2, 6.3.2 deal with this problem and propose several choices depending on $\mathcal{B}$.

A canonical choice for such a $\mathcal{B}^\star-$selection is

$$h(z) \stackrel{\mathrm{def}}{=} \sum_{S \in \mathcal{B}^\star} z^S.$$

In Lemma 6.4 we prove that for such an h, $\underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h) \geqslant 1$, which when combined with (6.3), gives us a lower bound on $g_{\mathcal{B}}$ in terms of $\mathrm{Cap}_{\mathcal{B}}(g)$ whenever h is real stable. Consequently, the fact that such an h is real stable when $\mathcal{B}$ is a regular matroid gives us an e^m-approximation for the case of regular matroids.

To extend the above reasoning when the canonical choice of h is not real stable, we define approximate $\mathcal{B}^\star-$selections and prove that a variant of inequality (6.3) holds whenever the support of h coincides with $\mathcal{B}^\star$ and h is real stable. For instance, if $\mathcal{B}$ is a linear matroid, then there exists a real stable polynomial of the form $g(z) = \det(\sum_{i=1}^m z_i v_i v_i^\top)$ (for some vectors $v_1, v_2, \ldots, v_m \in \mathbb{R}^r$) whose support is exactly $\mathcal{B}^\star$. This leads to a finite bound on $M_{\mathrm{lin}}(\mathcal{B})$ and $M(\mathcal{B})$. In general, by analyzing specific classes of matroids and coming up with $\mathcal{B}^\star-$selections for them we obtain several bounds that are listed in the theorem below.

Towards the proof of Theorem 6.4 the first step is to introduce a convex relaxation for the optimization problem $\max_{S \in \mathcal{B}} g_S$. Perhaps the most natural choice for such a relaxation is

$$\sup_{x \in P(\mathcal{B})} \quad g(x_1, \ldots, x_m).$$

While this works for the uniform matroid case ($\mathcal{B} = \binom{[m]}{n}$), for other families $\mathcal{B}$, it might have an unbounded integrality gap. Instead, consider the polynomial

$$r(z) \stackrel{\mathrm{def}}{=} g(x_1 z_1, \ldots, x_m z_m)$$

parametrized by $x > 0$. It is easy to see that

$$r_{\mathcal{B}} = \sum_{S \in \mathcal{B}} x^S g_S.$$

To avoid the influence of terms outside of $\mathcal{B}$ one can try to maximize $r_{\mathcal{B}}$ over $x \in P(\mathcal{B})$. The issue is that $r_{\mathcal{B}}$ is not necessarily efficiently computable. However, we know that $\mathrm{Cap}_{\mathcal{B}}(r)$ provides a good approximation to $r_{\mathcal{B}}$. Hence, we arrive at the following relaxation for $\max_{S \in \mathcal{B}} g_S$

$$\sup_{x \in P(\mathcal{B})} \quad \mathrm{Cap}_{\mathcal{B}}(g(x_1 z_1, \ldots, x_m z_m)).$$

Finally, we show that computing the above quantity reduces to a concave-convex saddle point problem which can be solved using the entropy maximization framework provided in Chapter 7.

6.3 Proofs

6.3.1 $\mathcal{B}$-Capacity

This section is devoted to the study of $\mathcal{B}-$capacity. We first provide an intuitive entropy interpretation of $\mathrm{Cap}_{\mathcal{B}}(p)$ and subsequently proceed to the proofs of several inequalities, as outlined in the Section 6.2 and the dual characterization of $\mathcal{B}-$capacity. We start by introducing the related notion of lower $\mathcal{B}$-capacity.

Definition 6.3 (***Lower $\mathcal{B}-$ Capacity***)

Consider an m-variate polynomial $g \in \mathbb{R}^+[z_1, \ldots, z_m]$ and let $\mathcal{B} \subseteq 2^{[m]}$ be any family of sets. The lower $\mathcal{B}-$capacity of g is:

$$\underline{\mathrm{Cap}}_{\mathcal{B}}(g) = \inf_{\theta \in P(\mathcal{B})} \inf_{z > 0} \frac{g(z)}{\prod_{i=1}^m z_i^{\theta_i}}.$$

An Entropy Interpretation of $\mathcal{B}-$Capacity

In this section it is convenient to think of $p \in \mathbb{R}^+[z_1, \ldots, z_m]$ as a probability distribution over monomials z^α, more precisely, the probability of a monomial z^α is simply $\frac{p_\alpha}{p(1)}$. (Note that $p(1) = \sum_\alpha p_\alpha$.) We show that computing $\mathcal{B}-$capacity of a polynomial p can be equivalently seen as finding a distribution q minimizing the KL-divergence between p and q subject to marginal constraints (see [SV14]). In this section we use

$$\Lambda_n \stackrel{\text{def}}{=} \{\alpha \in \mathbb{N}^m : |\alpha| = n\}$$

to denote the set of all monomials of degree n.

Proposition 6.1 (*Entropy Interpretation*)

Given a real stable, $n-$homogeneous polynomial $p \in \mathbb{R}^+[z_1, \dots, z_m]$ and $\mathcal{B} \subseteq \binom{[m]}{n}$, the capacity $\mathrm{Cap}_{\mathcal{B}}(p)$ is equal to the exponential of the optimum value of the following convex optimization problem

$$\begin{aligned} \sup_{q,\theta} \quad & -KL(q,p), \\ \text{s.t.} \quad & \sum_{\alpha \in \Lambda_n} q_\alpha = 1, \\ & \sum_{\alpha \in \Lambda_n} q_\alpha \cdot \alpha = \theta, \\ & \theta \in P(\mathcal{B}), \\ & q \geqslant 0. \end{aligned} \tag{6.4}$$

where q is a vector indexed by all possible multi-indices $\alpha \in \mathbb{N}^m$ with $|\alpha| = n$ and $KL(q,p) \stackrel{\text{def}}{=} \sum_{\alpha \in \mathbb{N}^m} q_\alpha \log \frac{q_\alpha}{p_\alpha}$.

Recall that p can seen as a probability distribution over monomials, or in other words over multisubsets of $[m]$ of cardinality n, which we represent by Λ_n. If q is a distribution over Λ_n, then $\theta = \sum_{\alpha \in \Lambda_n} q_\alpha \cdot \alpha$ is the marginal vector of q in which case θ_i (for $i \in [m]$) represents the expected number of copies of i in a sample α drawn according to q.

The optimization problem (6.4) asks to find a distribution q over Λ_n which is the closest (in relative entropy) to the given distribution p under the constraint that its marginal lies in $P(\mathcal{B})$. This is also known as the I-projection in information theory.

Assume now for brevity that p is normalized, i.e., $p(1) = 1$. In case when its marginal vector θ already lies in $P(\mathcal{B})$, we know that (since $KL(q,p) \geqslant 0$) the optimal solution to (6.4) is $q = p$ and hence $\mathrm{Cap}_{\mathcal{B}}(p) = 1$. In view of our results (see Theorem 6.2), this implies a quite surprising fact. If p and $\mathcal{B}$ satisfy the assumptions of Theorem 6.2 then we can lower-bound $p_{\mathcal{B}} = \sum_{S \in \mathcal{B}} p_S$ by an absolute positive number (not depending on p). If p was an arbitrary polynomial (not real stable as in Theorem 6.2) then we could easily make its marginal lie in $P(\mathcal{B})$ (and thus $\mathrm{Cap}_{\mathcal{B}}(p) = 1$) while keeping $p_S = 0$ for all $S \in \mathcal{B}$.

Proof of Proposition 6.1:
The proof relies on convex duality. Fix any probability distribution p on Λ_n and a marginal vector $\theta \in P(\mathcal{B})$. We derive the dual of the following convex program

$$\begin{aligned} \max_{q} \quad & -KL(q,p), \\ \text{s.t.} \quad & \sum_{\alpha \in \Lambda_n} q_\alpha = 1, \\ & \sum_{\alpha \in \Lambda_n} q_\alpha \cdot \alpha = \theta, \\ & q \geqslant 0. \end{aligned} \tag{6.5}$$

We make a simplifying assumption that there exists $q > 0$ such that $\theta = \sum_{\alpha \in \Lambda_n} q_\alpha \alpha$. This implies that the Slater's condition is satisfied for (6.5), which makes the analysis much simpler.

Introduce Lagrangian multipliers $z \in \mathbb{R}$ and $\lambda \in \mathbb{R}^m$ and consider the Lagrangian function:

$$L(q, \lambda, z) = -KL(q, p) - \lambda^\top \left(\sum_{\alpha \in \Lambda_n} q_\alpha \cdot \alpha - \theta \right) - z \cdot \left(\sum_{\alpha \in \Lambda_n} q_\alpha - 1 \right).$$

We are going to derive a formula for $g(\lambda, z) = \max_q L(q, \lambda, z)$. To this end, we derive optimality conditions with respect to q.

$$\frac{\partial}{\partial q_\alpha} L = -\log q_\alpha - 1 + \log p_\alpha - \lambda^\top \alpha - z = 0.$$

Note that the above implies that $q > 0$ and this is why we do not need to introduce dual variables for non-negativity constraints. Using the above conditions we obtain

$$g(\lambda, z) = \sum_{\alpha \in \Lambda_n} p_\alpha e^{-\lambda^\top \alpha - z - 1} + \lambda^\top \theta + z.$$

Hence the dual can be written as

$$\min_{\lambda \in \mathbb{R}^m, z \in \mathbb{R}} \sum_{\alpha \in \Lambda_n} p_\alpha e^{-\lambda^\top \alpha - z - 1} + \lambda^\top \theta + z.$$

We eliminate the z variable from the above by minimizing the objective with respect to z. Thus we obtain

$$\min_{\lambda \in \mathbb{R}^m} \log \left(\sum_{\alpha \in \Lambda_n} p_\alpha e^{-\lambda^\top \alpha} \right) + \lambda^\top \theta. \tag{6.6}$$

Because of our assumption, Slater's condition is satisfied and hence strong duality holds, thus we obtain equality between the optimal value of (6.5) and (6.6). To obtain the desired form, replace $-\lambda_i \in \mathbb{R}$ by $\log z_i$ for $z > 0$. This gives

$$\min_{z > 0} \log \left(\sum_{\alpha \in \Lambda_n} p_\alpha z^\alpha \right) - \sum_{i=1}^m \theta_i \ln z_i.$$

and after taking the exponential we recover the familiar

$$\min_{z > 0} \frac{p(z)}{\prod_{i=1}^m z_i^{\theta_i}}. \tag{6.7}$$

After dropping the assumption that θ can be obtained as $\sum_\alpha q_\alpha \alpha$ with $q > 0$, the equality we established above still holds, but the minimum value might not be attained (only in the limit). We skip the proof in the general case.

It is now enough to observe that taking the maximum over $\theta \in P(\mathcal{B})$ of (6.7) gives $\mathrm{Cap}_{\mathcal{B}}(p)$, hence indeed we obtain equality between $\mathrm{Cap}_{\mathcal{B}}(p)$ and the exponential of the optimal value of (6.5).

$\square$

An Inequality on $\mathcal{B}-$Capacity

For the setting when g is $m-$homogeneous and $\mathcal{B} = \{\{1, 2, \ldots, m\}\}$ one recovers from $\mathrm{Cap}_{\mathcal{B}}(g)$ the capacity defined in [Gur06]. The main goal of this section is to provide an extension of Gurvits' result which asserts that

$$\mathrm{Cap}(g) \leqslant \frac{m^m}{m!} g_{[m]}$$

(where $g_{[m]}$ is the coefficient of $z^{[m]}$ in g) under the assumption that g is $m-$homogeneous, real stable and has non-negative coefficients.

One of the crucial ingredients of our extension of [Gur06] is the following inequality which ties together the classical capacity and the ones we introduced here.

Lemma 6.1

Let $g, h \in \mathbb{R}^+[z_1, \ldots, z_m]$ be $m-$variate polynomials, $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of $n-$subsets of $[m]$, and $\mathcal{B}^\star = \{[m] \setminus S : S \in \mathcal{B}\}$ be its dual. Then

$$\mathrm{Cap}(g \cdot h) \geqslant \mathrm{Cap}_{\mathcal{B}}(g) \cdot \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h).$$

Proof.
We have, for every $\theta \in [0, 1]^m$ that

$$\mathrm{Cap}(g \cdot h) = \inf_{z>0} \frac{g(z) \cdot h(z)}{\prod_{i=1}^m z_i} \geqslant \inf_{z>0} \frac{g(z)}{\prod_{i=1}^m z_i^{\theta_i}} \cdot \inf_{z>0} \frac{h(z)}{\prod_{i=1}^m z_i^{1-\theta_i}}. \tag{6.8}$$

Note now that whenever $\theta \in P(\mathcal{B})$ then $(1 - \theta) \in P(\mathcal{B}^\star)$. To prove it, let $\theta = \sum_{S \in \mathcal{B}} \alpha_S 1_S$ for some $\{\alpha_S\}_{S \in \mathcal{B}}$ with $\alpha \geqslant 0$ and $\sum_S \alpha_S = 1$. Then

$$(1 - \theta) = \sum_{S \in \mathcal{B}} \alpha_S (1 - 1_S) = \sum_{S \in \mathcal{B}} \alpha_S 1_{\bar{S}} = \sum_{S \in \mathcal{B}^\star} \alpha_{\bar{S}} 1_S \in P(\mathcal{B}^\star).$$

By minimizing the second factor in the right hand side of (6.8) over $\theta \in P(\mathcal{B})$ we obtain the following

$$\mathrm{Cap}(g \cdot h) \geqslant \inf_{z>0} \frac{g(z)}{\prod_{i=1}^m z_i^{\theta_i}} \cdot \inf_{\tau \in P(\mathcal{B}^\star)} \inf_{z>0} \frac{h(z)}{\prod_{i=1}^m z_i^{\tau_i}}$$

for every fixed $\theta \in P(\mathcal{B})$. By maximizing the right hand side of the above with respect to θ, we finally arrive at

$$\mathrm{Cap}(g \cdot h) \geqslant \mathrm{Cap}_{\mathcal{B}}(g) \cdot \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h).$$

□

Since we are interested in proving an upper bound on the $\mathcal{B}-$capacity we will apply the Lemma 6.1 in the following way

$$\mathrm{Cap}_{\mathcal{B}}(g) \leqslant \frac{\mathrm{Cap}(g \cdot h)}{\underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h)}.$$

Thus the task of upper bounding the capacity boils down to finding an appropriate polynomial h, which allows us to relate $\mathrm{Cap}(g \cdot h)$ in the right hand side to the sum of coefficients $g_{\mathcal{B}}$. There is some freedom in the choice of h, hence one can set the second goal to make the lower capacity $\underline{\mathrm{Cap}}_{\mathcal{B}^\star}$ as large as possible.

Below we provide a definition which makes precise which properties of h are relevant.

Definition 6.4 (*$\mathcal{B}-$Selection*)

Assume $h \in \mathbb{R}^+[z_1, \ldots, z_m]$ is an $m-$variate, $n-$homogeneous polynomial and let $\mathcal{B} \subseteq \binom{[m]}{n}$. We call h a $\mathcal{B}$-selection if it satisfies the following two conditions

1. *For $S \in \binom{[m]}{n}$, $h_S > 0$ only if $S \in \mathcal{B}$,*
2. *$h_S = 1$ for every $S \in \mathcal{B}$.*

We say that h is a $c-$approximate $\mathcal{B}$-selection if it satisfies (1) and (2) is replaced by $h_S \in [1, c]$ for every $S \in \mathcal{B}$ (here $c \geqslant 1$ is any number).

Note that h is not assumed to be square-free, and hence importantly there is quite a lot of flexibility in the choice of a $\mathcal{B}$-selection. We are now ready to state and prove the main technical result of this section, which relates $\mathrm{Cap}_{\mathcal{B}}(p)$ to $p_{\mathcal{B}}$. The precision of this approximation depends on the quality of a $\mathcal{B}^\star$-selection (its lower capacity) one can provide.

Lemma 6.2

Let $g \in \mathbb{R}^+[z_1, \ldots, z_m]$ be a real stable $n-$homogeneous polynomial and $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of sets. For every real stable $\mathcal{B}^\star$-selection $h \in \mathbb{R}^+[z_1, \ldots, z_m]$ we have

$$\mathrm{Cap}_{\mathcal{B}}(g) \cdot \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h) \cdot \frac{m!}{m^m} \leqslant g_{\mathcal{B}} \leqslant \mathrm{Cap}_{\mathcal{B}}(g).$$

Moreover, if $g, h \in \mathbb{R}_1^+[z_1, \ldots, z_m]$, then the term $\frac{m!}{m^m}$ in the bound above can be replaced by 2^{-m}.

Proof.
Consider the polynomial $g(z) \cdot h(z)$ which is real stable as a product of real stable polynomials. Note that from our assumptions it is homogeneous of degree m. We apply Gurvits' inequality [Gur06] to $g \cdot h$. Let $s \in \mathbb{R}$ be the coefficient of $\prod_{i=1}^m z_i$ in $g \cdot h$, then

$$\mathrm{Cap}(g \cdot h) \leqslant \frac{m^m}{m!} s.$$

Since h is a $\mathcal{B}^\star-$selection, it follows that

$$s = \sum_{S \in \mathcal{B}} g_S = g_{\mathcal{B}},$$

because the only pairs of monomials from g and h, which contribute to s, are of the form x^S and $x^{[m] \setminus S}$. By combining this with Lemma 6.1, we obtain

$$\mathrm{Cap}_{\mathcal{B}}(g) \cdot \underline{\mathrm{Cap}}_{\mathcal{B}^\star}(h) \leqslant \mathrm{Cap}(g \cdot h) \leqslant g_{\mathcal{B}} \frac{m!}{m^m}.$$

To obtain the improved bound under the assumption that $g, h \in \mathbb{R}_1^+[z_1, \ldots, z_m]$ we observe that in the reasoning above, the degree of every variable in the polynomial $g \cdot h$ is at most 2.

Hence we can apply a stronger form of Gurvits' inequality [Gur06], where $\frac{m^m}{m!}$ is replaced by

$$\prod_{i=1}^{m}\left(\frac{d_i}{d_i-1}\right)^{d_i-1} \leqslant 2^m,$$

where d_i is the degree of z_i in $g \cdot h$.

This provides us the left hand side of the inequality. The right hand side follows easily from the dual characterization of $\mathcal{B}-$capacity provided in Lemma 6.3. ☐

Remark 6.1

The above lemma, still holds (with the same proof) when we assume h to be only a c-approximate $\mathcal{B}^\star$-selection. The only required modification is to divide the left hand side of the lower bound inequality by c.

Dual Characterization of $\mathcal{B}-$Capacity

The way $\mathrm{Cap}_{\mathcal{B}}(g)$ is defined makes it well suited for proving lower bounds on $g_{\mathcal{B}}$. In the following lemma we provide an equivalent, dual characterization, which gives a straightforward upper bound and is often preferred from the computational viewpoint.

Lemma 6.3 (*Equivalent definition of $\mathcal{B}-$capacity*)

Let $g \in \mathbb{R}^+[z_1, \ldots, z_m]$ be an $n-$homogeneous polynomial and $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of sets. Then

$$\mathrm{Cap}_{\mathcal{B}}(g) = \inf\{g(z) : z > 0, z^S \geqslant 1 \text{ for all } S \in \mathcal{B}\}.$$

Proof.
We depart from the following convex program

$$\begin{aligned} \inf_{y\in\mathbb{R}^m} \quad & \log g(e^y), \\ \text{s.t.} \quad & \sum_{i\in S} y_i \geqslant 0, \quad S \in \mathcal{B}. \end{aligned} \tag{6.9}$$

By $g(e^y)$ in the above we mean $g(e^{y_1}, \ldots, e^{y_m})$. The objective is simply

$$y \mapsto \log\left(\sum_{\alpha} g_\alpha e^{\langle \alpha, y\rangle}\right).$$

Such a function (for non-negative g_α) is well known to be convex (which follows from Hölder's inequality).

Note that Slater's condition for (6.9) is satisfied, hence strong duality holds. In order to derive the dual of the convex program (6.9) introduce multipliers $\mu_S \geqslant 0$ for every $S \in \mathcal{B}$ and consider the Lagrangian

$$L(y, \mu) \stackrel{\text{def}}{=} \log g(e^y) - \sum_{S\in\mathcal{B}} \mu_S \sum_{i\in S} y_i.$$

By taking the derivative with respect to y_i and equating to zero, we obtain the following optimality condition

$$e^{y_i} \cdot \frac{\frac{\partial}{\partial z_i} g(z)|_{z=e^y}}{g(e^y)} = \sum_{S \ni i} \mu_S.$$

By summing up all these conditions for $i = 1, 2, \ldots, m$ we obtain n on the left hand side (because g is homogeneous) and $n \cdot \sum_{S \in \mathcal{B}} \mu_S$ on the right. Hence, at optimality $\sum_{S \in \mathcal{B}} \mu_S = 1$. From strong duality, we obtain

$$\max_{\substack{\mu \geqslant 0, \\ \sum_{S \in \mathcal{B}} \mu_S = 1}} \min_{y \in \mathbb{R}^n} \left(\log g(e^y) - \sum_{S \in \mathcal{B}} \mu_S \sum_{i \in S} y_i \right) = \min_{\substack{\sum_{i \in S} y_i \geqslant 0 \\ \forall S \in \mathcal{B}}} \log g(e^y).$$

It remains to observe that

$$\sum_{S \in \mathcal{B}} \mu_S \sum_{i \in S} y_i = \sum_{i=1}^{m} y_i \sum_{S \ni i} \mu_S,$$

hence what really matters are the marginals $\theta_i = \sum_{S \ni i} \mu_S$ and not the probability distribution μ itself. For this reason one can rewrite the above equality as

$$\max_{\theta \in P(\mathcal{B})} \min_{y \in \mathbb{R}^n} \left(\log g(e^y) - \sum_{i=1}^{m} y_i \theta_i \right) = \min_{\substack{\sum_{i \in S} y_i \geqslant 0 \\ \forall S \in \mathcal{B}}} \log g(e^y).$$

The lemma follows by replacing e^y by z and taking exponentials on both sides. □

Remark 6.2

Our definition of $\mathcal{B}-$capacity also makes sense when $\mathcal{B} \subseteq \mathbb{N}^m$. It can be stated analogously to the dual formulation for $\mathcal{B} \subseteq 2^{[m]}$ by enforcing $z^\alpha \geqslant 1$ for $\alpha \in \mathcal{B}$.

6.3.2 Counting

This section discusses applications of $\mathcal{B}-$capacity to the counting problem. We start by showing that $\mathcal{B}-$capacity can be efficiently computed and subsequently provide bounds on the approximation ratio it yields for several classes of matroids.

The Choice of $\mathcal{B}$-Selection

As demonstrated in Section 6.3.1 the task of proving that $\mathrm{Cap}_{\mathcal{B}}(g)$ well approximates $g_{\mathcal{B}}$ boils down to coming up with a real stable polynomial h which is a $\mathcal{B}^\star-$selection and its lower capacity with respect to $\mathcal{B}^\star$ is as large as possible. We will provide one generic way of coming up with such polynomials and proving lower bounds on their lower capacity. It captures the case when $\mathcal{B}^\star$ is a strongly Rayleigh matroid. In the next subsection we extend it to capture more general settings.

Recall that for a given family $\mathcal{B} \subseteq \binom{[m]}{n}$ (which should be thought as the dual of what we normally call $\mathcal{B}$) we are interested in an $n-$homogeneous real stable polynomial h, which is a $\mathcal{B}-$selection and has a large lower $\mathcal{B}$-capacity. There exists one natural choice for h, the generating polynomial of $\mathcal{B}$

$$h(z) = \sum_{S \in \mathcal{B}} z^S$$

It satisfies the conditions for a $\mathcal{B}-$selection in an obvious way, thus the remaining questions are:

- Is $h(z)$ real stable?
- What is the lower $\mathcal{B}-$capacity of h?

The first question leads us directly to the notion of Rayleigh and strongly Rayleigh matroids introduced in [CW06]. As shown in [Brä07] the class of matroids $\mathcal{M}$ such that $h(z)$ is real stable (for $\mathcal{B}$ being the set of bases of $\mathcal{M}$) is precisely equal to the class of matroids enjoying the strongly Rayleigh property. In the next subsection we discuss possible ways to weaken this requirement of $\mathcal{B}$ being strongly Rayleigh by manipulating the coefficients of $h(z)$.

Now we address the second question from the above list.

Lemma 6.4

For every non-empty family $\mathcal{B} \subseteq \binom{[m]}{n}$, the polynomial $f(z) = \sum_{S \in \mathcal{B}} \beta_S z^S$, with $\beta_S \geqslant 1$ for every $S \in \mathcal{B}$, satisfies

$$\underline{\mathrm{Cap}}_{\mathcal{B}}(f) \geqslant 1.$$

Proof.
We need to prove that for every choice of $\theta \in P(\mathcal{B})$ we have

$$\inf_{z>0} \frac{f(z)}{\prod_{i=1}^m z_i^{\theta_i}} \geqslant 1.$$

Let us then fix $\theta \in P(\mathcal{B})$ and any $z > 0$. Since $\theta \in P(\mathcal{B})$ we can write it as $\theta = \sum_{S \in \mathcal{B}} \alpha_S 1_S$ for some non-negative $\alpha \in [0,1]^{\mathcal{B}}$ with $\sum_S \alpha_S = 1$. From Jensen's inequality we obtain

$$\begin{aligned} \log\left(\sum_{S \in \mathcal{B}} \alpha_S z^S\right) &\geqslant \sum_{S \in \mathcal{B}} \alpha_S \log\left(z^S\right) \\ &= \sum_{S} \alpha_S \sum_{i \in S} \log z_i \\ &= \sum_{i=1}^m \sum_{S \ni i} \alpha_S \log z_i \\ &= \sum_{i=1}^m \theta_i \log z_i. \end{aligned}$$

By taking exponentials and using the estimate $f(z) \geqslant \sum_{S \in \mathcal{B}} \alpha_S z^S$ we obtain the lemma. □

One can also observe that the above lemma is actually tight and indeed $\underline{\mathrm{Cap}}_{\mathcal{B}}(f)$ is equal to $\min\{\beta_S : S \in \mathcal{B}\}$. For any $S \in \mathcal{B}$, by taking $\theta = 1_S$ and $x \to 1_S$ one obtains that $\underline{\mathrm{Cap}}_{\mathcal{B}}(f) \leqslant \beta_S$.

$\mathcal{B}-$Selections for Linear Matroids

Consider the case when $\mathcal{B} \subseteq \binom{[m]}{n}$ is the set of bases of a linear matroid $\mathcal{M}$ (over $\mathbb{R}$). This means that there is a matrix $V \in \mathbb{R}^{m\times d}$ having rows $v_1, v_2, \ldots, v_m$ such that

$$S \in \mathcal{M} \quad \Leftrightarrow \quad \text{the collection } \{v_i\}_{i\in S} \text{ is linearly independent.}$$

This can be also restated as

$$S \in \mathcal{M} \quad \Leftrightarrow \quad \det(V_S^\top V_S) > 0,$$

where V_S is the matrix V restricted to rows with indices in S. Such a matrix V we call a linear representation of $\mathcal{M}$. Note that the polynomial

$$h(z) = \sum_{S\in\binom{[m]}{n}} \det(V_S^\top V_S) z^S$$

has as its support exactly $\mathcal{B}$. We have the following well known, but important fact.

Fact 6.1

For any matrix $V \in \mathbb{R}^{m\times d}$ the polynomial $h(z) = \sum_{S\in\binom{[m]}{n}} \det(V_S^\top V_S) z^S$ is $n-$homogeneous and real stable.

Proof.
We present the proof in the special case when $d = n$. Consider the polynomial

$$r(z) = \det\left(\sum_{i=1}^m z_i v_i v_i^\top\right) = \det\left(V^\top Z V\right),$$

where $Z = \mathrm{Diag}\,(z_1, \ldots, z_m)$. As a determinantal polynomial, $r(z)$ is real stable (see e.g. [Vis13]). It remains to observe that from the Cauchy Binet formula

$$r(z) = \sum_{S\in\binom{[m]}{n}} z^S \det(V_S^\top V_S).$$

$\square$

This fact allows us to use $h(z)$, after a suitable rescaling, as an approximate $\mathcal{B}-$selection. This rescaling can be controlled by the *unbalance* of a linear matroid which essentially measures the maximum distortion $\frac{\det(V_S^\top V_S)}{\det(V_T^\top V_T)}$ over $S, T \in \mathcal{B}$ (see definition in Section 6.1.2). We obtain

Lemma 6.5

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a set of bases of an $\mathbb{R}-$linear matroid. There exists a real stable $c-$approximate $\mathcal{B}-$selection h with $c \leqslant \mathrm{un}(\mathcal{B})$ (the unbalance of $\mathcal{B}$) and $\underline{\mathrm{Cap}}_{\mathcal{B}}(h) \geqslant 1$.

Proof.
Take V to be the most balanced representation of $\mathcal{B}$, i.e. $\mathrm{un}(V) = \mathrm{un}(\mathcal{B})$. We can scale the vectors

of V in such a way that

$$1 \leqslant \det(V_S V_S^\top) \leqslant \mathrm{un}(V) \quad \text{for all } S \in \mathcal{B}.$$

From the Fact 6.1 the polynomial

$$h(z) = \sum_{S \in \binom{[m]}{n}} \det(V_S V_S^\top) z^S$$

is real stable, and clearly it is an $\mathrm{un}(V)-$approximate $\mathcal{B}-$selection. □

Also, more generally, we can state the following lemma on nonuniform generating polynomials.

Lemma 6.6

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a set of bases of a matroid. Suppose that μ is a strongly Rayleigh distribution with $\mathrm{supp}(\mu) = \mathcal{B}$ and $P = \max_{S,T \in \mathcal{B}} \frac{\mu(S)}{\mu(T)}$. Then there exists a real stable $P-$approximate $\mathcal{B}-$selection h with $\underline{\mathrm{Cap}}_{\mathcal{B}}(h) \geqslant 1$.

Proof.
Take $h(z)$ to be

$$\frac{1}{\min_{S \in \mathcal{B}} \mu(S)} \sum_{T \in \mathcal{B}} z^T \mu(T).$$

We have that $1 \leqslant h_S \leqslant P$ for every $S \in \mathcal{B}$. Since $h \in \mathbb{R}_1^+[z_1, \ldots, z_m]$ and $\mathrm{supp}(h) = \mathcal{B}$, h is a $P-$approximate $\mathcal{B}-$selection. Also, $h(z)$ is real stable, because $\sum_{T \in \mathcal{B}} z^T \mu(T)$ is. □

$\mathcal{B}-$Selections for Partition Matroids

The generic choice of an $\mathcal{B}-$selection $h(z)$ to be $\sum_{S \in \mathcal{B}} z^S$ is indeed natural and intuitively "right", however it seems to be suboptimal. Two important cases, where we can provably surpass this sub-optimality, are uniform matroids and partitions matroids.

Lemma 6.7

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a set of bases of a partition matroid, i.e. $\mathcal{B}$ is of the form

$$\mathcal{B} = \{S : |S \cap P_j| = b_j \text{ for } j = 1, 2, \ldots, p\},$$

where $P_1, P_2, \ldots, P_p$ is a partition of $[m]$ into p disjoint sets and $\sum_{j=1}^{p} b_j = n$. Then there exists a real stable $\mathcal{B}-$selection $h(z)$ with

$$\underline{\mathrm{Cap}}_{\mathcal{B}}(h) \geqslant \prod_{j=1}^{p} \frac{b_j^{b_j}}{b_j!}.$$

Proof.

Consider the following choice of h

$$h(z) = \prod_{j=1}^{p} \frac{\left(\sum_{i \in P_j} z_i\right)^{b_j}}{b_j!}.$$

It is not hard to see that $h(z)$ is a $\mathcal{B}$-selection. Indeed, all the coefficients of monomials z^S for $S \in \mathcal{B}$ are 1 and these are the only square-free monomials which appear with a nonzero coefficient.

It suffices to show a lower bound on $\underline{\mathrm{Cap}}_{\mathcal{B}}$. To this end fix any $\theta \in P(\mathcal{B})$, any $z > 0$ and consider

$$\frac{h(z)}{\prod_{i=1}^{m} z_i^{\theta_i}} = \prod_{j=1}^{p} \frac{\left(\sum_{i \in P_j} z_i\right)^{b_j}}{b_j! \prod_{i \in P_j} z_i^{\theta_i}}. \tag{6.10}$$

At this point one can observe that all the terms in the product can be analyzed separately, this is because $P(\mathcal{B})$ is actually a cartesian product of p sets. More precisely

$$P(\mathcal{B}) = \{\theta \in [0,1]^m : \sum_{i \in P_j} \theta_i = b_j \text{ for all } j = 1, 2, \ldots, p\}.$$

Let us consider one particular term in the product in (6.10). Without loss of generality take $j = 1$, and assume that $P_1 = \{1, 2, \ldots, d\}$ and rename b_1 to simply b. Our goal now reduces to lower-bounding

$$\frac{(\sum_{i=1}^{d} z_i)^b}{b! \prod_{i=1}^{d} z_i^{\theta_i}},$$

where $\theta \in [0,1]^d$ satisfies $\sum_{i=1}^{d} \theta_i = b$. From Jensen's inequality for log we obtain

$$b \cdot \log\left(\sum_{i=1}^{d} z_i\right) = b \cdot \log\left(\sum_{i=1}^{d} \frac{\theta_i}{b} \cdot \frac{b z_i}{\theta_i}\right) \geqslant \sum_{i=1}^{d} \theta_i \log \frac{b z_i}{\theta_i}.$$

Further,

$$\sum_{i=1}^{d} \theta_i \log \frac{b z_i}{\theta_i} = \sum_{i=1}^{d} \theta_i \log \frac{b}{\theta_i} + \sum_{i=1}^{d} \theta_i \log z_i.$$

Finally note that $\left\{\frac{\theta_i}{b}\right\}_{i \in [d]}$ is a probability distribution over d items with probabilities bounded from above by $1/b$, this implies that its negative entropy is at least

$$-\sum_{i=1}^{d} \frac{\theta_i}{b} \log \frac{\theta_i}{b} \geqslant \log b.$$

Concluding, we obtain

$$b \cdot \log\left(\sum_{i=1}^{d} z_i\right) \geqslant \sum_{i=1}^{d} \theta_i \log z_i + b \log b.$$

After taking exponentials and dividing by $b!$ we get

$$\frac{(\sum_{i=1}^d z_i)^b}{b!\prod_{i=1}^d z_i^{\theta_i}} \geqslant \frac{b^b}{b!}.$$

□

Proofs of Theorems 6.2 and 6.3

Proof of Theorem 6.2:
Since there is a strongly Rayleigh distribution supported on $\mathcal{B}^\star$, applying Lemma 6.6 we can conclude existence of a real stable $P-$approximate $\mathcal{B}^\star-$selection for some (possibly large) $P > 0$, such that $\underline{\mathrm{Cap}}_{\mathcal{B}}(h) \geqslant 1$. Now, the approximate version of Lemma 6.2 (see Remark 6.1) concludes the proof. □

Proof of Theorem 6.3:
For points 1. and 2. we reason as in the proof of Theorem 6.2, we construct suitable $\mathcal{B}^\star$ selections using Lemmas 6.6 and 6.5 respectively and then apply the approximate variant of Lemma 6.2. Note that the above mentioned $\mathcal{B}^\star-$selections are multi-linear, hence we can apply the sharper 2^{-m} bound in Lemma 6.2 in the case when g is multi-linear as well.

For point 3. we derive the bound it in Section 6.3.6. To obtain a bound on $M(\mathcal{B})$, note that $\mathcal{B}^\star$ is a partition matroid $\{U(P_j, |P_j| - b_j)\}_{j\in[p]}$. We set $h(z)$ to be the $\mathcal{B}^\star-$selection constructed in Lemma 6.7 and apply Lemma 6.2, this yields a bound

$$M(\mathcal{B}) \leqslant \frac{m^m}{m!} \cdot \prod_{j=1}^p \frac{(|P_j| - b_j)!}{(|P_j| - b_j)^{|P_j|-b_j}}.$$

It remains to argue that

$$\frac{m^m}{m!} \cdot \prod_{j=1}^p \frac{(|P_j| - b_j)!}{(|P_j| - b_j)^{|P_j|-b_j}} \leqslant e^n \left(\frac{2\pi m}{p}\right)^{p/2}.$$

After applying the bound $\frac{k!}{k^k} \leqslant \sqrt{k}e^{-k+1}$ and using $\sum_{j=1}^p (|P_j| - b_j) = m - n$ we obtain

$$\frac{m^m}{m!} \cdot \prod_{j=1}^p \frac{(|P_j| - b_j)!}{(|P_j| - b_j)^{|P_j|-b_j}} \leqslant e^m e^{-m+n+p} \prod_{j=1}^p \sqrt{|P_j| - b_j}.$$

Finally, by the AM-GM inequality

$$\prod_{j=1}^p \sqrt{|P_j| - b_j} \leqslant \left(\frac{m-n}{p}\right)^{p/2} \leqslant \left(\frac{m}{p}\right)^{p/2}.$$

□

6.3.3 Optimization

In this section we discuss the problem of finding

$$\max_{S \in \mathcal{B}} g_S \tag{6.11}$$

for a given $n-$homogeneous polynomial $g \in \mathbb{R}^+[x_1, \ldots, x_m]$ and a set family $\mathcal{B} \subseteq \binom{[m]}{n}$.

One naive way to approach problem (6.11) would be to apply Theorem 6.3 directly. Since $\mathrm{Cap}_{\mathcal{B}}(g)$ approximates $g_{\mathcal{B}}$ up to a factor of $M(\mathcal{B})$ we could just output $\mathrm{Cap}_{\mathcal{B}}(g)$ as an approximation to (6.11) and obtain a guarantee of

$$M(\mathcal{B}) \cdot |\mathcal{B}| \leqslant M(\mathcal{B}) \cdot m^n = M(\mathcal{B}) \cdot e^{n \log m}.$$

We believe that the bounds in Theorem 6.3 can be strengthened, so that $M(\mathcal{B})$ (or $M_{\mathrm{lin}}(\mathcal{B})$) depend on n only (as for the case of uniform matroids), which makes the $e^{n \log m}$ inefficient. In the next subsection we propose a method which achieves an approximation guarantee of at most $M(\mathcal{B}) \cdot e^n$ and can be better, depending on a particular $\mathcal{B}$.

Convex Relaxation

We consider the following relaxation to problem (6.11)

$$\sup_{x \in P(\mathcal{B})} \inf_{\substack{y > 0 \\ \forall S \in \mathcal{B}\ y^S \geqslant 1}} g(x_1 y_1, \ldots, x_m y_m). \tag{6.12}$$

Lemma 6.8

Let $\mathcal{B} \subseteq \binom{[m]}{n}$ be a family of bases of a matroid. For every real stable, $n-$homogeneous polynomial $g \in \mathbb{R}^+[z_1, \ldots, z_m]$. The optimal value OPT of the relaxation (6.12) *satisfies*

$$\frac{OPT}{M(\mathcal{B}) \cdot A(\mathcal{B})} \leqslant \max_{S \in \mathcal{B}} g_S \leqslant OPT,$$

where $A(\mathcal{B}) \overset{\text{def}}{=} \max\left\{\sum_{S \in \mathcal{B}} x^S : x \in P(\mathcal{B})\right\} \leqslant e^n$. Moreover, in the case when $g \in \mathbb{R}_1^+[z_1, \ldots, z_m]$, $M(\mathcal{B})$ can be replaced by $M_{\mathrm{lin}}(\mathcal{B})$ in the bound above.

Proof.
Let $p(x, y)$ denote the polynomial $g(x_1 y_1, \ldots, x_m y_m)$. One can easily see that if $x = 1_S$ for some $S \in \mathcal{B}$ then $p(x, y) \geqslant y^S \cdot g_S$, this implies that

$$\max_{S \in \mathcal{B}} g_S \leqslant OPT.$$

To prove the lower-bound, fix the optimal solution $\bar{x}$ to the relaxation (6.12) and consider the real stable polynomial $g(z) = p(\bar{x}, z)$. It follows that $OPT = \mathrm{Cap}_{\mathcal{B}}(g)$, hence

$$OPT \leqslant M(\mathcal{B}) \cdot g_{\mathcal{B}}.$$

We also have

$$g_{\mathcal{B}} = \sum_{S\in\mathcal{B}} \bar{x}^S g_S \leqslant \left(\sum_{S\in\mathcal{B}} \bar{x}^S\right) \cdot \max_{S\in\mathcal{B}} g_S \leqslant A(\mathcal{B}) \cdot OPT.$$

What remains to prove is that $A(\mathcal{B}) \leqslant e^n$. This turns out to be a quite simple consequence of the constraints $x \geqslant 0$ and $\sum_{i=1}^n x_i = n$. Indeed, the largest possible value of the sum $\sum_{S\in\mathcal{B}} \bar{x}^S$ is attained when $\bar{x}_i = \frac{n}{m}$ for all $i \in [m]$. □

Lemma 6.9

The relaxation (6.12) *is efficiently computable. Given access to an evaluation oracle for g and a separation oracle for $P(\mathcal{B})$ one can obtain a $(1+\varepsilon)-$approximation to the optimal solution of* (6.12) *in time polynomial in m,* $\log \frac{1}{\varepsilon}$ *and L, where L is an upper bound on the description size of coefficients of g.*[1]

The Proof of Lemma 6.9 appears in Section 6.3.7. The results of this section allow us to deduce Theorem 6.4.

Proof of Theorem 6.4.
Given an optimization problem (6.11) we apply the relaxation (6.12) to it. Lemma 6.9 guarantees that it can be solved in polynomial time. Now by applying Lemma 6.8 we obtain the claimed approximation guarantee. □

Application to Maximizing Sub-determinants

In this section we discuss the problem of maximizing sub-determinants under constraints. Let $L \in \mathbb{R}^{m\times m}$ be a symmetric PSD matrix and let $\mathcal{B} \subseteq \binom{[m]}{n}$ be any family of sets. We consider the problem

$$\max_{S\in\mathcal{B}} \det(L_{S,S}), \tag{6.13}$$

where $L_{S,S}$ denotes the sub-matrix of L obtained by restricting it to rows and columns from S. Typically in this setting it is useful to consider the Cholesky decomposition $L = VV^\top$ for some matrix $V \in \mathbb{R}^{m\times d}$. This allows us to write the generating polynomial

$$g(z) = \sum_{S\in\binom{[m]}{n}} z^S \det(L_{S,S}) = \sum_{S\in\binom{[m]}{n}} z^S \det(V_S^\top V_S) = \det(V^\top Z V),$$

where $Z = \operatorname{Diag}(z_1, \ldots, z_m)$. We can then apply the result of Theorem 6.4 to obtain Corollary 6.1.

Proof of Corollary 6.1:
We consider

$$g(z) = \sum_{S\in\binom{[m]}{n}} z^S \det(L_{S,S})$$

as above. As observed in Fact 6.1 $g(z)$ is $n-$homogeneous and real stable. Moreover $g(z)$ is efficiently computable, since it just boils down to computing a determinant of a $d \times d$ matrix. Through the optimization problem (6.11) $g(z)$ encodes exactly (6.13), hence we can apply Theorem 6.4. □

6.3.4 Counterexamples

We provide examples showing that $\mathrm{Cap}_{\mathcal{B}}(g)$ might not give a good approximation to $g_{\mathcal{B}}$ if either $\mathcal{B}$ is not a family of bases of a matroid or g is not real stable.

Example 1. We consider the case when $m = 4$ and $n = 2$. Consider a polynomial $g(z) = (z_1 + z_2)(z_3 + z_4)$ which is real stable as a product of real stable polynomials. We pick a family $\mathcal{B}$ which is not a set of bases of a matroid, namely $\mathcal{B} = \{\{1, 2\}, \{3, 4\}\}$. Using the dual characterization of $\mathrm{Cap}_{\mathcal{B}}$ we have

$$\mathrm{Cap}_{\mathcal{B}}(g) = \inf_{\substack{z>0 \\ z_1 z_2 \geqslant 1, z_3 z_4 \geqslant 1}} g(z).$$

Hence we obtain that $\mathrm{Cap}_{\mathcal{B}}(g) = 4$, while clearly $g_{\mathcal{B}} = 0$.

Example 2. Consider now a similar example with the roles of g and $\mathcal{B}$ "reversed". Let $\mathcal{B} = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}$ be a family of bases of a partition matroid and $g(z) = z_1 z_2 + z_3 z_4$. Again, we have

$$\mathrm{Cap}_{\mathcal{B}}(g) \geqslant 2,$$

while $g_{\mathcal{B}} = 0$. In this example the polynomial $g(z)$ is not real stable. Indeed, if $\omega = e^{\frac{2\pi i}{8}}$ then $(\omega, \omega, \omega^3, \omega^3)$ is a root of g with positive imaginary part.

6.3.5 A Lower bound on $M(\mathcal{B})$

We prove a lower bound on $M(\mathcal{B})$, which can be seen as another interpretation of the fact that the quality of approximating the permanent of a non-negative matrix A by the capacity of its product polynomial p_A can be e^n in the worst case.

Fact 6.2

There exists a family $\mathcal{B} \subseteq \binom{[m]}{n}$ of bases of a partition matroid for which $M(\mathcal{B}) \geqslant e^{\sqrt{m}}$.

Proof.
Consider a universe $U = [n] \times [n]$ of cardinality $m = n^2$. Bases of the considered matroid have exactly one element in every part $\{i\} \times [n]$ for $i = 1, 2, \ldots, n$. Formally

$$\mathcal{B} = \{\{(1, f(1)), (2, f(2)), \ldots, (n, f(n))\} : f \in [n]^{[n]}\}.$$

Of course $|\mathcal{B}| = n^n$. Consider now a polynomial $g(z) = \left(\sum_{i,j} z_{i,j}\right)^n$, clearly $g(z)$ is real stable and it can be proved that $\mathrm{Cap}_{\mathcal{B}}(g) \geqslant n^{2n}$. Indeed, from the AM-GM inequality

$$g(z) = \left(\sum_{i,j} z_{i,j}\right)^n \geqslant (n^2)^n \prod_{i,j} z_{i,j}.$$

Under the condition that $z^S \geqslant 1$ for every $S \in \mathcal{B}$ the above implies that $g(z) \geqslant n^{2n}$.

It is also easy to calculate that $g_{\mathcal{B}} = |\mathcal{B}| \cdot n! = n^n \cdot n!$, hence we obtain $M(\mathcal{B}) \geqslant \frac{\mathrm{Cap}_{\mathcal{B}}(g)}{g_{\mathcal{B}}} = \frac{n^{2n}}{n^n \cdot n!} \approx e^n = e^{\sqrt{m}}$. □

6.3.6 A Bound for Partition Matroids

Lemma 6.10

Let $\mathcal{M}$ be a partition matroid $\{U(P_j, b_j)\}_{j\in[p]}$ then $M_{\text{lin}}(\mathcal{B}) \leqslant \frac{n^n}{n!}\prod_{j=1}^{p}\frac{b_j!}{b_j^{b_j}}$.

Proof.
Let $n = \sum_{j=1}^{p} b_j$ be the rank of the partition matroid. Consider an $n-$homogeneous polynomial $g \in \mathbb{R}_1^+[z_1, \dots, z_m]$. We perform the following *symmetrization* procedure. For every part P_j we introduce b_j new variables $u_{j,1}, u_{j,2}, \dots, u_{j,b_j}$ and define $s_j = \sum_{i=1}^{b_j} u_{j,i}$. For notational convenience, let us define a function $\sigma : [m] \to [p]$ which given an element $e \in [m]$, outputs an index $\sigma(e)$ such that $e \in P_{\sigma(e)}$. We consider

$$f(u) = g(s_{\sigma(1)}, s_{\sigma(2)}, \dots, s_{\sigma(m)}).$$

Note that now $f(u)$ is $n-$variate, $n-$homogeneous and real stable. Moreover, we can relate $g_{\mathcal{B}}$ to the coefficient (call it c) of $\prod_{i,j} u_{i,j}$ in $f(u)$ as follows,

$$c = p_{\mathcal{B}} \cdot \prod_{j=1}^{p} b_j!.$$

By Gurvits' inequality for $n-$variate $n-$homogeneous polynomials we have

$$\text{Cap}(f) \leqslant \frac{n^n}{n!} c.$$

We will use the following simple upper bound on $\text{Cap}_{\mathcal{B}}(g)$,

$$\text{Cap}_{\mathcal{B}}(g) = \inf_{\substack{z>0 \\ \forall S\in\mathcal{B}\ z^S \geqslant 1}} g(z) \leqslant \inf_{\substack{z>0 \\ \forall S\in\mathcal{B}\ z^S=1}} g(z).$$

Note that importantly

$$\begin{aligned}\inf_{\substack{z>0 \\ z^S=1, S\in\mathcal{B}}} g(z) &= \frac{1}{\prod_{j=1}^{p} b_j^{b_j}} \cdot \inf_{\substack{u>0 \\ \prod_{i,j} u_{i,j}=1}} f(u) \\ &= \frac{1}{\prod_{j=1}^{p} b_j^{b_j}} \cdot \text{Cap}(f).\end{aligned}$$

this equality follows because the constraints $z^S = 1$ for every $S \in \mathcal{B}$, imply that the value of z_i is constant inside every partition P_j. Hence there exists a one-to-one mapping between feasible

z and feasible u. As a consequence we obtain

$$\begin{aligned}\mathrm{Cap}_{\mathcal{B}}(g) &\leqslant \frac{1}{\prod_{j=1}^{p} b_j^{b_j}} \cdot \mathrm{Cap}(f)\\ &\leqslant \frac{1}{\prod_{j=1}^{p} b_j^{b_j}} \frac{n^n}{n!} \cdot c\\ &= p_{\mathcal{B}} \cdot \frac{n^n}{n!} \cdot \prod_{j=1}^{p} \frac{b_j!}{b_j^{b_j}}.\end{aligned}$$

□

6.3.7 Efficient Computability

To state our complexity bounds we first introduce the complexity measure of a polynomial $p \in \mathbb{R}^+[z_1, \ldots, z_m]$ given by a separation oracle. Let $d \in \mathbb{N}$ be the degree of p and $\mathrm{supp}(p)$ be the support of p, i.e., the set of all $\alpha \in \mathbb{N}^m$ such that $p_\alpha > 0$. The complexity measure of p we denote by L_p and define as

$$L_p \overset{\text{def}}{=} m + d + \max_{\alpha \in \mathrm{supp}(p)} |\log p_\alpha|.$$

In other words, the complexity is determined by the number of variables, the degree and (roughly) the maximum number of bits required to represent any of the coefficients. Note importantly that the actual amount of space required to store such a polynomial explicitly might be exponentially higher.

Computability of the Relaxation for the Counting Problem

The goal of this subsection is to provide a proof of the following

Theorem 6.5

If $p \in \mathbb{R}^+[z_1, \ldots, z_m]$ is an $m-$variate, real stable polynomial given by an evaluation oracle and $\mathcal{B} \subseteq 2^{[m]}$ is a family of sets given by a separation oracle for $P(\mathcal{B})$, then $\mathrm{Cap}_{\mathcal{B}}(p)$ can be computed up to a multiplicative factor of $(1+\varepsilon)$ in time polynomial in $\log \frac{1}{\varepsilon}$ and L_p.

The above is established using the entropy interpretation of $\mathrm{Cap}_{\mathcal{B}}(p)$ and the result in Chapter 7. For this one has to provide an efficient separation oracle for the polytope $\mathrm{Newt}(p) \overset{\text{def}}{=} \mathrm{conv}(\mathrm{supp}(p))$ – the Newton polytope of p. This is the subject of the following lemma.

Lemma 6.11

Let $p \in \mathbb{R}^+[z_1, z_2, \ldots, z_m]$ be a real stable polynomial given by an evaluation oracle. There exists a separation oracle for $\mathrm{Newt}(p)$ which runs in polynomial time with respect to the input size and L_p.

Proof.
The proof relies on the fact that in the case when p is real stable, $\mathrm{supp}(p)$ has special combinatorial properties. In fact, if p is homogenous and multi-affine then its support is a set of bases of a matroid; in the general case $\mathrm{supp}(p)$ is a *jump system* (see [Brä07]), which can be seen as a generalization of matroids to arbitrary subsets of $\mathbb{N}^m$.

Denote $\mathcal{F} \overset{\text{def}}{=} \operatorname{supp}(p)$, from the connection between jump systems and bisubmodular polyhedra established in [BC95], one obtains the following characterization of $\operatorname{conv}(\mathcal{F}) = \operatorname{Newt}(p)$:

$$\operatorname{conv}(\mathcal{F}) = \{x \in \mathbb{R}^m : \forall s \in \{-1,0,1\}^m \ \exists \alpha \in \mathcal{F} \quad \langle x, s\rangle \leqslant \langle \alpha, s\rangle\}.$$

Thus to construct a polynomial time separation oracle one has to give a polynomial time algorithm to solve the problem: given $x \in \mathbb{R}^m_{\geqslant 0}$ find $\alpha \in \mathcal{F}$ and $s^\star \in \{-1,0,1\}^m$ which attains the minimum value of

$$v^\star = \min_{s\in\{-1,0,1\}^m} \max_{\alpha\in\mathcal{F}} \left(\langle \alpha, s\rangle - \langle x, s\rangle\right).$$

Given $s^\star$, if $v^\star \geqslant 0$ then $x \in \operatorname{conv}(\mathcal{F})$ and otherwise $s^\star$ provides a separating hyperplane. It suffices then to compute $v^\star$. The inner maximization problem can be solved in polynomial time by Fact 6.3 below. For the outer problem one has to find

$$\min_{s\in\{-1,0,1\}^m} \left(f(s) - \langle x, s\rangle\right), \tag{6.14}$$

where $f(s) = \max_{\alpha\in\mathcal{F}} \langle \alpha, s\rangle$. Since $f(s)$ (and hence also $f(s) - \langle x, s\rangle$) is a bisubmodular function (see [BC95]), (6.14) is an instance of *bisubmodular minimization* which can be solved in polynomial time using a polynomial number of calls to an evaluation oracle (see [Qi88, FI05]) for $g(s) = f(s) - \langle x, s\rangle$. As proved in Fact 6.3, one can query g in polynomial time, given only an evaluation oracle for p. □

Fact 6.3

Let $p \in \mathbb{R}^+[z_1, z_2, \ldots, z_m]$ be a polynomial, the problem of computing

$$\max_{\alpha\in\operatorname{supp}(p)} \langle \alpha, s\rangle,$$

given an evaluation oracle access to p and $s \in \{-1,0,1\}^m$, can be solved in polynomial time with respect to L_p.

Proof.
Fix a polynomial p as in the statement and $s \in \{-1,0,1\}^m$. Denote

$$p_{\min} = \min_{\alpha\in\operatorname{supp}(p)} p_\alpha,$$
$$p_{\max} = \max_{\alpha\in\operatorname{supp}(p)} p_\alpha,$$
$$p_{\text{tot}} = \sum_{\alpha\in\operatorname{supp}(p)} p_\alpha.$$

Note that $p_{\min} \geqslant e^{-L_p}$ and $p_{\max} \leqslant e^{L_p}$. Consider now a point $x(t) \in \mathbb{R}^m$ (with $t \in \mathbb{R}$) such that

$$x_i(t) = e^{s_i t} \quad \text{for all } i = 1, 2, \ldots, m.$$

Let Γ be the set of $\alpha \in \operatorname{supp}(p)$ which attain the maximum value $v^\star$ of $\langle \alpha, s\rangle$. We have

$$\sum_{\alpha\in\Gamma} p_\alpha e^{tv^\star} \leqslant \sum_{\alpha\in\Gamma} p_\alpha e^{t\langle s,\alpha\rangle} \leqslant p(x(t)) \leqslant \sum_{\alpha\in\Gamma} p_\alpha e^{tv^\star} + \sum_{\alpha\in\operatorname{supp}(p)\setminus\Gamma} p_\alpha e^{t(v^\star-1)}.$$

Consequently
$$\sum_{\alpha\in\Gamma} p_\alpha e^{tv^\star} \leqslant p(x(t)) \leqslant \sum_{\alpha\in\Gamma} p_\alpha e^{tv^\star}\left(1+\frac{p_{\text{tot}}}{e^t p_{\min}}\right).$$
By taking logs on both sides
$$tv^\star + \log\left(\sum_{\alpha\in\Gamma} p_\alpha\right) \leqslant \log p(x(t)) \leqslant tv^\star + \log\left(\sum_{\alpha\in\Gamma} p_\alpha\right) + \log\left(1+\frac{p_{\text{tot}}}{e^t p_{\min}}\right).$$

Our algorithm is then to take a large enough integer $\hat{t}\in\mathbb{N}$ and compute
$$v^\star \approx \log p(x(\hat{t}+1)) - \log p(x(\hat{t})),$$
more precisely, we round $\log p(x(\hat{t}+1)) - \log p(x(\hat{t}))$ to the nearest integer. It remains to argue how large $\hat{t}$ do we need to take in order to make $\log\left(1+\frac{p_{\text{tot}}}{e^{\hat{t}} p_{\min}}\right)$ negligible. For this, observe that
$$\frac{p_{\text{tot}}}{p_{\min}} \leqslant (d+1)^m \frac{p_{\max}}{p_{\min}} \leqslant (d+1)^m e^{2L_p}.$$
Hence, it is enough to take $\hat{t} = \text{poly}(L_p)$ to make $\frac{p_{\text{tot}}}{e^{\hat{t}} p_{\min}} < \frac{1}{2}$ and hence $\log\left(1+\frac{p_{\text{tot}}}{e^{\hat{t}} p_{\min}}\right) < \frac{1}{2}$. Note importantly that the point $x(\hat{t})$ we are querying p at has polynomial description size, hence also the output of the oracle will be so. □

Proof of Theorem 6.5:
To apply Theorem 7.6 we need an evaluation oracle for p, a separation oracle for the Newton polytope of p and a separation oracle for $P(\mathcal{B})$. Out of these three, only the separation oracle for $\text{Newt}(p)$ does not directly follow from assumptions. However, we can construct it using Lemma 6.11.

□

Computability of the Relaxation for the Optimization Problem

Proof of Lemma 6.9:
The proof is analogous to that of computability of $\mathcal{B}-$capacity of a polynomial and follows from Theorem 7.6. Here we highlight the main differences and additional details which do not show up when proving Theorem 6.5.

The starting point is the following form of the relaxation (6.12)
$$\text{Cap}_{\mathcal{B}}(g) = \sup_{x\in P(\mathcal{B})}\ \sup_{\theta\in P(\mathcal{B})}\ \inf_{z>0} \frac{g(x_1z_1,\ldots,x_mz_m)}{\prod_{i=1}^m z_i^{\theta_i}}.$$
Let us consider the logarithm of the above and plug in e^{y_i} for z_i. We obtain
$$\log \text{Cap}_{\mathcal{B}}(g) = \sup_{x\in P(\mathcal{B})}\ \sup_{\theta\in P(\mathcal{B})}\ \inf_{y\in\mathbb{R}^m}\ \log g(x_1e^{y_1},\ldots,x_me^{y_m}) - \langle\theta,y\rangle.$$
Denote the objective of the above by
$$h_g(x,\theta,y) \overset{\text{def}}{=} \log g(x_1e^{y_1},\ldots,x_me^{y_m}) - \langle\theta,y\rangle.$$

This allows us to take advantage of the entropy interpretation and the results of Chapter 7. More specifically, as long as the above function is jointly concave in x and θ, the conclusion of Theorem 7.6 applies and gives polynomial running time for this relaxation.

First note that joint concavity follows from just concavity in x and θ as these sets of variables do not interfere with each other in h_g. Concavity in θ is rather easy to see, since h_g is an affine function of θ. Concavity in x is rather surprising as it does not hold for an arbitrary polynomial g, it follows from real stability of g, see Fact 7.3.

$\square$

6.4 Conclusion

6.4.1 Future Work

An immediate open question which remains, is whether the approximation guarantees for our relaxations for counting and optimization are tight. This seems rather unlikely, as there is some evidence (inspired by inspecting special cases) that the approximation ratios should rather depend exponentially on n (the rank of the matroid) and not on m.

One can also ask whether the assumption of real stability in our theorems is necessary. Here, one can give several nontrivial examples in which the polynomial is not real stable and still the $\mathcal{B}$-capacity gives a good approximation. Hence, real stability is certainly not essential here, yet formulating a more general condition under which our bounds hold is a challanging problem.

6.4.2 Notes

Strongly Rayleigh distributions have been extensively studied (see, e.g., [COSW04, Brä07, BBL09]) because of their negative dependence properties. It is known that most interesting matroids satisfy the condition stated in Theorem 6.2 for the integrality gap of $\mathcal{B}$-capacity to be finite. A non-example has been discovered by [BBL09]: the 7-element Fano matroid.

A relaxation similar to $\mathcal{B}$-capacity was used in [NS16] in the context of sub-determinant maximization and in [AOSS17] for the Nash Social Welfare problem. In fact, the relaxation in [NS16] is an upper bound for ours. Both of them behave similarly in the case when $\mathcal{B}$ is a partition family. However, for other matroids, such as spanning tree matroids, the relaxation of [NS16] has an unbounded integrality gap, whereas $\mathrm{Cap}_{\mathcal{B}}$ is finite. Lemma 6.7 for the case of uniform matroids follows implicitly from the results of [AOSS17].

Chapter 7

On the Bit Complexity of Entropy Maximization

In this chapter we consider the entropy-maximization problem, as introduced in Chapter 3. We prove that such distributions, even though having exponential-size supports, often can be computed efficiently. The connection between max-entropy distributions and the $\mathcal{B}$-capacity studied in Chapter 6 allows us, in particular, to deduce polynomial time algorithms for computing $\mathrm{Cap}_{\mathcal{B}}(p)$ under mild conditions on p. By intepreting other problems in the language of entropy maximizations more corollaries follow.

7.1 Preliminaries and Statement of Results

7.1.1 Preliminaries

In this section we introduce the problems we study and state our main results. For the purpose of clarity, we provide simplified and slightly informal variants of our theorems here and refer the reader to Sections 7.3.1 and 7.3.2 for a detailed and formal exposition.

The Max-Entropy Program. Consider a finite subset $\mathcal{F} \subseteq \mathbb{Z}^m$ of the integer lattice[1]. Given a vector θ, the following max-entropy convex program solves for a probability distribution over $\mathcal{F}$ that has maximum entropy with expectation θ:

$$\begin{aligned} \max \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \log \frac{1}{q_\alpha}, \\ \text{s.t.} \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \cdot \alpha = \theta, \\ & \sum_{\alpha \in \mathcal{F}} q_\alpha = 1, \\ & q \geqslant 0. \end{aligned} \tag{7.1}$$

While an explicit solution $q^\star$ to this problem would require at least $\Omega(|\mathcal{F}|)$ (possibly exponential) space, by analyzing the dual program to (7.1) one can prove that there exists a vector

[1]For brevity, in this section we assume that $\mathcal{F}$ has a diameter polynomial in m, see Section 7.3.1 for details.

$\gamma \in \mathbb{R}^m_{>0}$ that satisfies $q^\star_\alpha \propto \prod_{i=1}^m {\gamma_i}^{\alpha_i}$ (see [SV14, WJ08]). However, there is no guarantee that the number of bits required to store γ is small (poly(m)) and in fact can be infinite.

From the work of [SV14], it follows that the efficient computability of such representations essentially relies on the two following conditions:

1. a polynomial bound on the bit complexity of γ and
2. the existence of an efficient *counting oracle* for $\mathcal{F}$.

To explain the latter we consider the function $g_\mathcal{F}(x) := \sum_{\alpha \in \mathcal{F}} x^\alpha$ where $x^\alpha := \prod_{i=1}^m x_i^{\alpha_i}$. A counting oracle for $\mathcal{F}$ is then an algorithm which, given a positive $x > 0$, outputs $g_\mathcal{F}(x)$. Note in particular, that when we plug in the all-ones vector we obtain $g_\mathcal{F}(\mathbb{1}) = |\mathcal{F}|$. In fact, [SV14] show that existence of efficient counting oracles is also necessary for max-entropy computations, as being able to find the optimal value of the max-entropy program efficiently for a given family $\mathcal{F}$ implies (roughly) an efficient method for evaluating $g_\mathcal{F}(x)$. Thus, when computing max-entropy distributions one can assume without loss of generality that an oracle for answering queries $x \mapsto g_\mathcal{F}(x)$ is available.

Consequently, the problem of computing efficient representations of max-entropy distributions boils down to proving upper bounds on the bit complexity of optimal dual solutions, and is the main focus of this chapter. The main result of [SV14] states that whenever the point θ lies in $P := \text{conv}(\mathcal{F})$ and also a ball of radius $\eta > 0$ centered at θ is contained in P, then the bit complexity of γ is bounded by $O(^{\log |\mathcal{F}|}/_\eta)$. In particular, this bound deteriorates as $\eta \to 0$ and it can be shown that the bit complexity of γ goes to infinity as $\eta \to 0$.

However, in applications an arbitrarily good approximate solution to the max-entropy program suffices and, in the light of the discussion above, one may ask if we can compute a distribution that is ε-close to the max-entropy distribution in time that is proportional to $\log {}^1/_\varepsilon$.

7.1.2 Statement of Results

Computability of Max-Entropy Distributions Our main result answers the above question affirmatively under mild conditions on the polytope corresponding to $\mathcal{F}$. In particular, we assume that this polytope has a low unary facet complexity, as formalized below

Definition 7.1 (*Informal; see Definition 7.2*)

Let $P \subseteq \mathbb{R}^m$ be a convex polytope with integer vertices. The **unary facet complexity** *of P, denoted by* fc(P)*, is the smallest number $M \in \mathbb{N}$, such that P can be described by linear inequalities with coefficients in the set $\{-M, \dots, 0, \dots, M\}$.*

This class is very general and includes most polytopes of interest. Consequently, as our first result, we close the problem of computability of max-entropy distributions for polytopes with polynomial unary facet complexity for all points $\theta \in \text{conv}(\mathcal{F})$, which was left open in the work of [SV14].

Theorem 7.1 (*Simplified; see Theorem 7.5*)

Let $\mathcal{F}$ be any finite subset of $\mathbb{Z}^m$ and assume that the unary facet complexity of $P := \operatorname{conv}(\mathcal{F})$ is polynomial in m. Then, there exists an algorithm such that given an evaluation oracle for $g_{\mathcal{F}}$, a $\theta \in P$ and an $\varepsilon > 0$, computes a vector $y \in \mathbb{R}^m$ with $\|y\| \leqslant \operatorname{poly}(m, \log 1/\varepsilon)$ such that

$$\|q^y - q^\star\|_1 < \varepsilon,$$

where $q^\star$ is the optimal solution to (7.1) and q^y is a distribution over $\mathcal{F}$ defined as $q^y_\alpha := \frac{e^{\langle \alpha, y\rangle}}{\sum_{\beta\in\mathcal{F}} e^{\langle \beta, y\rangle}}$ for some $\alpha \in \mathcal{F}$. The running time of this algorithm is polynomial in m and $\log 1/\varepsilon$.

Note that above, a vector γ given by $\gamma_i = e^{y_i}$ for all $i \in [m]$, approximately satisfies $q^\star_\alpha \propto \gamma^\alpha$. Thus, such a γ has a polynomial bit complexity as the norm of y is polynomially bounded. This result allows us to sample from a distribution that is ε-close in total variation distance to the optimal distribution, hence, adding at most an ε-error to any further use of such samples.

Bit Complexity of Entropy Maximization The proof of Theorem 7.1 is based on a structural result on the bit complexity of close-to-optimal dual solutions and comprises the technical core of the chapter. Before we state it, let us introduce the dual problem to (7.1). For $\theta \in \operatorname{conv}(\mathcal{F})$ and any $y \in \mathbb{R}^m$ define $h(\theta, y) := \log\left(\sum_{\alpha\in\mathcal{F}} e^{\langle \alpha-\theta, y\rangle}\right)$. The dual problem is parametrized by θ and is to minimize $h(\theta, y)$ over all $y \in \mathbb{R}^m$, i.e.,

$$g(\theta) := \inf_{y\in\mathbb{R}^m} h(\theta, y) := \inf_{y\in\mathbb{R}^m} \log\left(\sum_{\alpha\in\mathcal{F}} e^{\langle \alpha-\theta, y\rangle}\right). \tag{7.2}$$

Theorem 7.2 (*Main Structural Result - Simplified; see Theorem 7.4*)

There exists a bound $R = R\left(m, \log\frac{1}{\varepsilon}\right) \in \mathbb{R}_{>0}$, polynomial in m and $\log\frac{1}{\varepsilon}$ such that for every set $\mathcal{F} \subseteq \mathbb{Z}^m$ with unary facet complexity of $\operatorname{conv}(\mathcal{F})$ polynomially bounded[2]*, for every $\varepsilon > 0$ we have*

$$\forall_{\theta\in\operatorname{conv}(\mathcal{F})}\ \exists_{y\in B(0,R)}\quad h(\theta, y) \leqslant g(\theta) + \varepsilon,$$

where h and g are defined as in (7.2).

Here $B(0, R)$ denotes the unit ℓ_2-ball of radius R around the origin. Note that the bound we provide is *uniform* over the whole polytope P, meaning that the radius R does not depend on θ – this is crucial for the subsequent applications.

Below we present a result complementing Theorem 7.2 that asserts that the conclusion of Theorem 7.2 does not always hold and that some assumptions on $\mathcal{F}$ are necessary. More precisely, for certain sets $\mathcal{F}$, the length of y which yields a solution $\varepsilon-$close to the optimal one cannot depend logarithmically on $\log\frac{1}{\varepsilon}$ as in Theorem 7.2.

Theorem 7.3 (*Lower Bound*)

For every bound $R = R\left(m, \log\frac{1}{\varepsilon}\right) \in \mathbb{R}_{>0}$, polynomial in m and $\log\frac{1}{\varepsilon}$ there exists an $m \in \mathbb{N}$, an $\varepsilon > 0$ and a set $\mathcal{F} \subseteq \mathbb{Z}^m$ with $\|\alpha\| \leqslant O(m^{3/2})$ for every $\alpha \in \mathcal{F}$ such that

$$\exists_{\theta\in\operatorname{conv}(\mathcal{F})}\ \forall_{y\in B(0,R)}\quad h(\theta, y) > g(\theta) + \varepsilon.$$

The above result relies on existence of "flat" $0-1$ polytopes by Alon and Vu [AV97], i.e., full-dimensional polytopes whose vertices are in $\{0,1\}^m$ and which fit between two $e^{-\theta(m \log m)}$-close hyperplanes. It is an intriguing question whether a similar lower bound as in Theorem 7.3 holds also for families $\mathcal{F} \subseteq \{0,1\}^m$.

Applications and Connections Max-entropy distributions appear in various guises in many distinct fields of mathematics, physics and computer science. As a consequence, we are able to present a host of disparate looking results from a unified point of view and sometimes obtain new corollaries. Below we only provide a list and defer their discussion to Section 7.3.4.

- **Bounds for the matrix scaling problem.** For the doubly-stochastic (and more general) matrix scaling problem we prove polynomial bit complexity bounds on the scaling factors using Theorem 7.2.
- **Computability of recent continuous relaxations for counting and optimization problems.** The convex relaxations for counting and optimization problems involving polynomials studied in Chapter 6 can be solved in polynomial time whenever the supports of the underlying polynomials have small unary facet complexity.
- **Computability of worst-case Brascamp-Lieb constants.** We consider the question of computability of Brascamp-Lieb constants, and show that in the rank-1 regime one can compute worst-case constants (over the Brascamp-Lieb polytope) in polynomial time.

7.2 Technical Overview

Since Theorem 7.2 is the technical core of the chapter we start by discussing its proof and then show how Theorem 7.1 can be derived from it.
Theorem 7.2 – previous work and possible approaches. In the context of Theorem 7.2, the result of [SV14] can be restated as follows: whenever a ball of radius $\eta > 0$ centered at θ is contained in $P := \text{conv}(\mathcal{F})$ the optimal solution $y^\star$ of (7.2) exists and its length $\|y^\star\|$ is bounded by $O\left(\frac{m \log m}{\eta}\right)$. The proof is simple: the term $O(m \log m)$ above is really a bound on $\log|\mathcal{F}|$, and the $\frac{1}{\eta}$ term comes from the fact that after scaling by $\log|\mathcal{F}|$, the optimal solution $y^\star$ of the dual program belongs to the polar of the radius-η ball around θ. (The polar of radius-η ball is itself a ball of radius $1/\eta$.)

While it is not clear whether the dependency on η is optimal, one can see that $\|y^\star\| \to \infty$ when θ tends to the boundary of P and $y^\star$ might not even exist when θ is on the boundary. For this reason, Theorem 7.2 cannot be derived from the results of [SV14]. Moreover, any approach to bound the norm of the *optimal* solution would be unsuccessful; this is why we focus on approximately optimal solutions.

One could consider the following two approaches to prove Theorem 7.2 that try to take advantage of the above mentioned bound: *centering* and *projection*. Both are based on slightly moving θ to a new point θ' so that the result by [SV14] is applicable to θ' and then reasoning that the small shift does not affect the optimal dual solution. Centering is based on moving the point more towards the interior, for example, by taking the centroid of the polytope P (which is far from the boundary) and taking a small step from θ towards θ'. One can then prove that θ' is well in the interior of the polytope, and hence a suitable bound for it follows. The second idea is based on projections: start with any point θ, if θ is already (inverse-polynomially) far from the boundary of P, then the result of [SV14] implies a suitable bound. Otherwise, project the point

θ onto the closest facet of P and continue recursively. By doing this, either the resulting new point θ' will end up in a vertex of P, or on a lower-dimensional face of P, where it is far from the boundary.

In both approaches, proving the result for the new point θ' is easy, given [SV14]; what remains is to bound the error introduced when moving from θ to θ'. Proving such a bound on the error turns out to be a non-trivial task. In the case of centering one has to pick an appropriate direction along which the point θ is moved. Similarly, the first challenge in the projection approach is to even define a suitable "projection operator" on a facet which would behave as expected and do not cause the point θ to land outside of the polytope P (as Euclidean projections might do). An issue which concerns both approaches is to deal with the behavior of the function g close to the boundary, where it can be shown to be non-Lipschitz[3] and hence very susceptible to local perturbations. More precisely: moving from θ to θ' guarantees that the gradient $\nabla_y h(\theta, y)$ at $y = y'^{\star}$ – the dual optimal solution at θ' – is small, however it is a challenge to derive a guarantee on the gap $g(\theta) - h(\theta, y'^{\star})$ as the function $y \mapsto h(\theta, y)$ is not strongly convex[4]. In fact, even a weaker claim that $g(\theta) \approx g(\theta')$ does not follow from [SV14], since the convexity argument (as in Lemma 7.5 in the full version of [SV14]) only allows one to prove that $g(\theta') \geqslant g(\theta) - O(\varepsilon)$, where ε denotes the distance between θ and its "centering" θ', but not in the opposite direction, as required.

Our proof of Theorem 7.2 is based on a purely geometric reasoning and bypasses the above obstacles by working entirely in the dual space (working in which we believe is necessary). This allows us to appropriately capture the geometry of sub-level sets of $h(\theta, y)$ and as a consequence, understand effects of seemingly large perturbations in y which lead to only small changes in the function value. When tracked in the primal domain, the proof resembles the "centering" idea, however the implicit direction to move θ along is not easy to come up with (or analyze) just from the primal perspective.

Theorem 7.2 – proof overview. At a high level, in the proof we consider the optimal dual solution $y^{\star}$ and first identify a vertex $\alpha^{\star}$ such that $y^{\star}$ belongs to $C_{\alpha^{\star}}$ – the normal cone at $\alpha^{\star}$. Subsequently, a projection operation of $y^{\star}$ with respect to the cone $C_{\alpha^{\star}}$ is used to find a vector y° – a witness for a short and close to optimal solution of the dual problem. The proof can be decomposed naturally into the following three steps, which we subsequently explain in more detail.

A Identify a "good" basis for the dual space with respect to θ.

B Truncate the optimal dual solution with respect to the basis in (a).

C Establish a bound on the length of the truncated solution.

In what follows, we assume for simplicity that the polytope $\operatorname{conv}(\mathcal{F})$ is full-dimensional. Given any $\theta \in P$ and a point $y^{\star} \in \mathbb{R}^m$ which satisfies $h(\theta, y^{\star}) \leqslant g(\theta) + \varepsilon/2$ (i.e., is close to optimal[5]) we aim to find a point y° whose length is polynomial in m and $\log 1/\varepsilon$ such that $h(\theta, y^{\circ}) \leqslant h(\theta, y^{\star}) + \varepsilon/2$.

Step (a). We first identify a subset $I_0 \subseteq I$ such that $\{a_i : i \in I_0\}$ is a basis of $\mathbb{R}^m$ and $y^{\star}$ is expressed as a nonnegative linear combination in this basis, i.e., $y^{\star} = \sum_{i \in I_0} \beta_i a_i$ with $\beta \geqslant 0$. The

[3]In the simple case when $\mathcal{F} := \{0,1\}$, the function $g : [0,1] \to \mathbb{R}$ is of the form $g(\theta) = -\theta \log \theta - (1-\theta)\log(1-\theta)$. One can see that on every interval $(\varepsilon, 1-\varepsilon)$ for $\varepsilon > 0$ the function g is Lipschitz, but it is not Lipschitz on $(0,1)$.

[4]One can again consider the case of $\mathcal{F} := \{0,1\}$ – the function $h(0,y)$ is then of the form $h(0,y) = \log(1+e^y)$. Note that $h(0,y)$ is a convex function of y, but $\frac{d^2}{dy^2} h(0,y) \to 0$ whenever $y \to \pm\infty$, hence the function is essentially "flat" at infinity, and not strongly convex.

[5]Note that there might not exist a point $y^{\star}$ such that $g(\theta) = h(\theta, y^{\star})$ when θ is on the boundary of P; that is why we allow a slack of $\varepsilon/2$.

basis I_0 is chosen as a basis of tight constraints at a point $\alpha^\star \in \mathcal{F}$, i.e., $\alpha^\star$ satisfies $\langle a_i, \alpha^\star\rangle = b_i$ for $i \in I_0$. This follows by selecting an α that maximizes $\langle \alpha, y^\star\rangle$ over all $\alpha \in \mathcal{F}$ and invoking Farkas' lemma and Caratheodory's theorem. This part of the reasoning does not make any assumptions on the polytope and only relies on the convexity of P.

Step (b). Subsequently we prove that the point $y^\circ = \sum_{i \in I_0} \min(\Delta, \beta_i) a_i$ satisfies the claim stated above, for a suitable choice of Δ, polynomial in the considered parameters. The bound $h(\theta, y^\circ) \leqslant h(\theta, y^\star) + \varepsilon/2$ is proved by replacing β_i by $\min(\Delta, \beta_i)$ one by one and showing that the value $h(\theta, \cdot)$ does not increase by more than $\varepsilon/2m$. This relies on a careful analysis of the effect such a perturbation has on the function and crucially uses the fact that the coefficients β_i for $i \in I_0$ are nonnegative. Most importantly, we rely on the fact that the coefficients of the inequalities defining P are integral; and hence for any point $\alpha \in \mathcal{F}$ which does not lie on a facet $\langle a_i, x\rangle = b_i$ for some $i \in I_0$ we have $\langle \alpha^\star - \alpha, a_i\rangle \geqslant 1$.

Step (c). To bound the length of y° note first that all the vectors $\{a_i\}_{i \in I_0}$ are short, i.e., $\|a_i\|_2 \leqslant \|a_i\|_\infty \cdot m \leqslant \mathrm{fc}(P) \cdot m = \mathrm{poly}(m)$, because we assume that the unary facet complexity of P is polynomially bounded. Further, from the triangle inequality, we obtain $\|y^\circ\| \leqslant m \cdot \Delta \cdot \mathrm{poly}(m) = \mathrm{poly}(m, \log \frac{1}{\varepsilon})$. Thus, we conclude the proof of Theorem 7.2.

Theorem 7.1 – proof overview. Given $\theta \in P$ and $y \in \mathbb{R}^m$ we first relate the closeness of q^y to the max-entropy distribution q with the suboptimality gap $h(\theta, y) - g(\theta)$. For this, even though we would like to bound the ℓ_1 distance between q^y and $q^\star$, the right distance function to consider turns out to be the KL-divergence $KL(p, q) := -\sum_{\alpha \in \mathcal{F}} p_\alpha \log \frac{q_\alpha}{p_\alpha}$. This is the case as one can show that $h(\theta, y) - g(\theta)$ is equal to $KL(q^\star, q^y)$. Further, using Pinsker's inequality ($\|q - p\|_1 \leqslant \sqrt{2 \cdot KL(p, q)}$) we can recover a bound on the ℓ_1 distance.

Consequently, to find a $y \in \mathbb{R}^m$ whose induced distribution q^y is close to the max-entropy distribution $q^\star$ it is enough to ensure that the dual suboptimality gap $h(\theta, y) - g(\theta)$ is small. Hence, the problem boils down to finding an approximate solution y to the dual problem at θ. This can be accomplished using the ellipsoid algorithm; the crucial part of the implementation is to provide a bounding box, i.e., a bound on the norm of the solution we are looking for – such a bound follows from Theorem 7.2.

Theorem 7.3 – proof overview. The proof of Theorem 7.3 is based on existence of so called *flat* 0-1 polytopes. These are polytopes of the form $\mathrm{conv}\{\alpha_0, \ldots, \alpha_m\}$ with $\alpha_0, \ldots, \alpha_m \in \{0,1\}^m$ such that the distance from $\alpha_0 = 0$ to the affine subspace H generated by $\alpha_1, \ldots, \alpha_m$ is exponentially small. Existence of such configurations was proved in [AV97]. Given such a polytope we consider the lattice generated on H by the points $\alpha_1, \ldots, \alpha_m$ and construct a new polytope by taking a certain finite subset of such lattice points and the point 0. The vertices of the new polytope are still integral and have relatively small entries (polynomial in m). Moreover, the projection of 0 onto H lies within the opposite facet. The family $\mathcal{F}$ is defined to be all the vertices of the newly constructed polytope, θ is chosen to be 0 and we pick $\varepsilon \approx e^{-m}$.

To prove that for every short $y \in \mathbb{R}^m$ the gap between $h(\theta, y) - g(\theta)$ is significant we consider the gradient $\nabla_y h(\theta, y)$. Intuitively, if the gradient is large (in magnitude) at a point y, then y cannot be an approximately optimal solution, hence it is enough to show that every short vector y admits a suitably long gradient. For this one can show that the gradient at y is given by $\theta - \theta^y$ where θ^y is the expectation of a distribution defined by y.

In order to make θ^y ε-close to $\theta = 0$ one has to ensure that $\langle 0, y\rangle - \langle \alpha, y\rangle \gtrsim \Omega(1)$ for all $\alpha \in \mathcal{F} \setminus \{0\}$. However, by introducing an auxiliary optimization problem (see Fact 7.1) we show that this can happen only when $\|y\|$ is roughly, inverse-proportional to the distance from 0 to $\mathrm{conv}(\mathcal{F} \setminus \{0\})$. Since the distance is exponentially small, we arrive at a lower-bound on $\|y\|$.

7.3 Proofs

7.3.1 Proof of the Bit Complexity Upper Bound

We start by reintroducing several notions and restating the structural result, as in Section 7.1.2 only simplified or informal definitions and theorem statements were given. Consider a finite subset $\mathcal{F} \subseteq \mathbb{Z}^m$ of the integer lattice and a positive function $p : \mathcal{F} \to \mathbb{R}_{>0}$. We will use p_α to denote the value of the function at a point $\alpha \in \mathcal{F}$, thus treating p as an $|\mathcal{F}|-$dimensional vector with coordinates indexed by $\mathcal{F}$. For any $\theta \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ we define the following generalized max-entropy program

$$\begin{aligned} \max \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \log \frac{p_\alpha}{q_\alpha}, \\ \text{s.t.} \quad & \sum_{\alpha \in \mathcal{F}} q_\alpha \cdot \alpha = \theta, \\ & \sum_{\alpha \in \mathcal{F}} q_\alpha = 1, \\ & q \geqslant 0. \end{aligned} \tag{7.3}$$

In the case when p is normalized so that $\sum_{\alpha \in \mathcal{F}} p_\alpha = 1$, (7.3) describes the distribution closest to p (in KL-distance), whose expectation is equal to θ. In the case when $p \equiv 1$ (corresponds to the uniform distribution) the above program asks simply for a max-entropy distribution with expectation θ, as in Section 7.1.2. Let us also extend the definition of the dual program, as introduced in Section 7.1.2 to capture general functions p not only $p \equiv 1$.

$$g(\theta) := \inf_{y \in \mathbb{R}^m} h(\theta, y) := \inf_{y \in \mathbb{R}^m} \log \left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha - \theta, y \rangle} \right). \tag{7.4}$$

The following notion of facet complexity of a polytope plays an important role in our main result.

Definition 7.2

Let $P \subseteq \mathbb{R}^m$ be a convex polytope with integer vertices. Let $M \in \mathbb{N}$ be the smallest integer such that P has a description of the form

$$P = \{x \in \mathbb{R}^m : \langle a_i, x \rangle \leqslant b_i, \text{ for } i \in I\} \cap H \tag{7.5}$$

where I is a finite index set, $a_i \in \mathbb{Z}^m$, $\|a_i\|_\infty \leqslant M$ and $b_i \in \mathbb{R}$ for $i \in I$, and H is a linear subspace of $\mathbb{R}^m$. Then we call M the unary facet complexity *of P and denote* $\text{fc}(P) = M$.

Observe that in a description of a polytope P as in the definition above we could have also included H in the first term of (7.5), by adding $\langle c, x \rangle \leqslant d$ and $\langle c, x \rangle \geqslant d$, for every equation $\langle c, x \rangle = d$ defining H. However, the unary facet complexity defined as above might be significantly lower than one measured with respect to all (including equality) constraints and, as it turns out, this is the right measure to study the bit complexity of close to optimal solutions of (7.3).

To state our main result also a notion of *bit complexity* of a function $p : \mathcal{F} \to \mathbb{R}_{>0}$ is required. We denote it by L_p and define as

$$L_p := \max_{\alpha \in \mathcal{F}} |\log p_\alpha|.$$

Note that L_p is finite, because we assume that $p_\alpha > 0$ for every $\alpha \in \mathcal{F}$. It represents, roughly, the maximum[6] number of bits required to store the binary representation of any p_α (for $\alpha \in \mathcal{F}$).

The last definition we need to state our structural result is the *diameter* of the set $\mathcal{F}$ (or equivalently P), it is simply $\operatorname{diam}(\mathcal{F}) := \max\{\|\alpha_1 - \alpha_2\| : \alpha_1, \alpha_2 \in \mathcal{F}\}$. [7]

Theorem 7.4 (*Main Structural Result*)

Let $\mathcal{F}$ be any finite subset of $\mathbb{Z}^m$ and let $d \in \mathbb{R}_{\geqslant 0}$ be its diameter, let M be the unary facet complexity of $\operatorname{conv}(\mathcal{F})$*. Then, for every function $p : \mathcal{F} \to \mathbb{R}_{>0}$ and for every $\varepsilon > 0$ there exists a number $R > 0$ which is polynomial in $m, \log d, M, L_p$ and $\log \frac{1}{\varepsilon}$ such that*

$$\forall_{\theta \in P}\ \exists_{y \in B(0,R)}\ \ h(\theta, y) \leqslant g(\theta) + \varepsilon,$$

where h and g are defined as in (7.2).

We start by a preliminary lemma which is then used in the proof of Theorem 7.4.

Lemma 7.1

Let $P \subseteq \mathbb{R}^m$ be a polytope $P := \{x \in \mathbb{R}^m : \langle a_i, x\rangle \leqslant b_i \text{ for } i \in I\} \cap H$, where I is a finite index set, $H \subseteq \mathbb{R}^m$ is a linear subspace of $\mathbb{R}^m$ and $a_i \in H$, $b_i \in \mathbb{R}$ for $i \in I$. Let $y \in H$ be any vector, then there exists a vertex $v \in P$ and a subset of the constraints $I_0 \subseteq I$ of size $|I_0| \leqslant \dim(H)$, such that $\langle a_i, v\rangle = b_i$ for $i \in I_0$ and there exist non-negative numbers $\{\beta_i\}_{i \in I_0}$ satisfying

$$\sum_{i \in I_0} \beta_i a_i = v.$$

Proof.

We start by picking $v \in P$ which maximizes $\langle x, y\rangle$ over $x \in P$. Note that since P is a polytope (and is compact) such a maximum exists and, moreover without loss of generality we might assume that v is a vertex of P. Given such a $v \in P$ determine all inequalities which are tight at v, i.e.,

$$I^\star = \{i \in I : \langle a_i, v\rangle = b_i\}.$$

Note that $|I^\star|$ might be arbitrarily large, not even polynomially bounded. We claim that there exist $\{\beta_i\}_{i \in I^\star}$ with $\beta_i \geqslant 0$ for all $i \in I^\star$, such that

$$y = \sum_{i \in I^\star} \beta_i a_i.$$

Suppose it is not the case. Then from the Farkas lemma, there exists a vector $z \in \mathbb{R}^m$ such that:

$$\forall_{i \in I^\star}\ \ \langle z, a_i\rangle \leqslant 0 \qquad \text{and} \qquad \langle z, y\rangle > 0.$$

Note also that we may assume that $z \in H$, by projecting z orthogonally onto H if necessary. Further, the above is true also for $\delta \cdot z$ (in place of z) for an arbitrarily small $\delta > 0$. In other words we can take z of arbitrarily small norm. Hence we obtain that the cone

$$C = \{u \in H :\ \forall_{i \in I^\star}\ \ \langle u, a_i\rangle \leqslant 0\}$$

[6]Importantly, the complexity measure L_p is the *maximum* – not the *total* – number of bits required to store any of the coefficients. The latter is always at least $|\mathcal{F}|$, hence typically exponential in m.

[7]In this chapter we use $\|x\|$ to denote the Euclidean ℓ_2 norm of x. This choice of a norm is by any means essential, as we tend to ignore factors polynomial in the dimension m.

contains arbitrarily short vectors z with $\langle z, y\rangle > 0$. Note that since $I^\star$ is the collection of all inequalities tight at v, it follows that every point in H, which is sufficiently close to v and satisfies the inequalities in $I^\star$ belongs itself to P. In other words there exists a $\delta > 0$ (might be exponentially small) such that

$$(v + C) \cap B(v, \delta) \subseteq P,$$

where $v + C = \{v + u : u \in C\}$ and $B(v, \delta) = \{x \in \mathbb{R}^m : \|x - v\| \leqslant \delta\}$. Combining this with our previous observation regarding the cone C it follows that there exists $z \in H$ such that $\mu := v + z \in P$ and $\langle z, y\rangle > 0$ and hence

$$\langle \mu, y\rangle > \langle v, y\rangle .$$

This contradicts our choice of $v \in P$.

$$\langle \alpha, y\rangle > \langle v, y\rangle .$$

Knowing that y belongs to the cone generated by $\{a_i\}_{i\in I^\star}$ we can apply Caratheodory's theorem to reduce the number of nonzero coefficients in the resulting conic combination. Indeed there exist a set $I_0 \subseteq I^\star$, such that $|I_0| \leqslant \dim(H)$ and non-negative $\{\beta_i\}_{i\in I_0}$ (possibly different β_is than obtained above) such that

$$y = \sum_{i\in I_0} \beta_i a'_i.$$

□

Proof of Theorem 7.4:
Before we proceed with the argument let us first observe that one can assume without loss of generality that $0 \in \mathcal{F}$. This follows from the "shift invariance" of our problem. Indeed if we consider $\mathcal{F}$ and $\mathcal{F}' = \mathcal{F} + \gamma = \{\alpha + \gamma : \alpha \in \mathcal{F}\}$, then the corresponding functions h ad h' satisfy

$$h(\theta, y) = h'(\theta + \gamma, y)$$

for every $y \in \mathbb{R}^m$. Hence, by shifting $\mathcal{F}$ by $\gamma = -\alpha$ for some $\alpha \in \mathcal{F}$ we obtain an equivalent instance of our problem with $0 \in \mathcal{F}$. It follows in particular, that the affine subspace H on which P is full-dimensional is now a *linear subspace* of $\mathbb{R}^m$.

Fix $\theta \in P$ and let $y^\star$ be such that

$$h(\theta, y^\star) \leqslant g(\theta) + \frac{\varepsilon}{2}.$$

Note that we may assume that $y^\star \in H$, by projecting it orthogonally onto H (which does not alter the value). Further, note that by denoting by $a'_i \in H$ (for $i \in I$) the orthogonal projection of a_i onto H, the polyotope P can be equivalently written as

$$P = \{x \in H : \langle a'_i, x\rangle \leqslant b_i\}.$$

By applying Lemma 7.1 for $y^\star$ and P we obtain a vertex $\alpha^\star \in \mathcal{F}$ and a subset of constraints I_0 of size at most $\dim(H) \leqslant m$, tight at $\alpha^\star$ such that

$$y^\star = \sum_{i\in I_0} \beta_i a'_i,$$

for some non-negative scalars $\{\beta_i\}_{i\in I_0}$. We prove that by modifying the coefficients in the above conic combination we can obtain a point

$$y' = \sum_{i\in I_0} \beta_i' a_i'$$

(with $\beta_i' \geqslant 0$ for $i \in I_0$) such that the norm of y' is small (polynomial in $L_p, m, \log d, M$ and $\log \frac{1}{\varepsilon}$) and

$$h(\theta, y') \leqslant h(\theta, y^\star) + \frac{\varepsilon}{2} \leqslant g(\theta) + \varepsilon.$$

Showing that will complete the argument. Let $\Delta > 0$ be a certain number (to be specified later), polynomial in L_p and $\log \frac{1}{\varepsilon}$. We define $\beta_i' := \min(\Delta, \beta_i)$ and prove that the point $y' = \sum_{i\in I_0} \beta_i' a_i'$ satisfies the above claim.

To this end, we prove that by changing one coordinate i_0, from $\beta_{i_0} > \Delta$ to Δ we cause only a slight increase in the value of $h(\theta, y)$. In other words, by taking $y^\star$ as before and $y' = y^\star - (\beta_{i_0} - \Delta)a_i$ we want to show that

$$\log\left(\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle}\right) \leqslant \log\left(\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle\alpha,y^\star\rangle - \langle\theta,y^\star\rangle}\right) + \frac{\varepsilon}{2m}$$

Towards this, define

$$\mathcal{F}_0 = \{\alpha \in \mathcal{F} : \langle\alpha, a_{i_0}\rangle = b_{i_0}\}.$$

Below we analyze separately the effect of changing y to y' on the terms $p_\alpha e^{\alpha,y}$ for $\alpha \in \mathcal{F}_0$ and for $\alpha \in \mathcal{F} \setminus \mathcal{F}_0$.

Case 1: $\mathcal{F} \setminus \mathcal{F}_0$. Consider any $\alpha \in \mathcal{F} \setminus \mathcal{F}_0$. We have

$$\begin{aligned}\langle\alpha, y'\rangle - \langle\alpha^\star, y'\rangle &= \sum_{i\in I_0} \beta_i' \langle\alpha - \alpha^\star, a_i'\rangle \\ &= \sum_{i\in I_0} \beta_i' \langle\alpha - \alpha^\star, a_i\rangle \\ &\leqslant \Delta \langle\alpha - \alpha^\star, a_{i_0}\rangle \\ &\leqslant -\Delta.\end{aligned}$$

In the above we used the fact that a_i' is the projection of a_i onto H, $\langle\alpha - \alpha^\star, a_i\rangle \leqslant 0$ for every $i \in I_0$ and that $\langle\alpha - \alpha^\star, a_{i_0}\rangle$ is a negative integer. This implies in particular that

$$\frac{p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle}}{p_{\alpha^\star} e^{\langle\alpha^\star,y'\rangle - \langle\theta,y'\rangle}} \leqslant \frac{p_\alpha}{p_{\alpha^\star}} e^{-\Delta}$$

and hence:

$$\begin{aligned}\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle} &= \sum_{\alpha\in\mathcal{F}_0} p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle} + \sum_{\alpha\in\mathcal{F}\setminus\mathcal{F}_0} p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle} \\ &\leqslant \left(\sum_{\alpha\in\mathcal{F}_0} p_\alpha e^{\langle\alpha,y'\rangle - \langle\theta,y'\rangle}\right)\left(1 + |\mathcal{F} \setminus \mathcal{F}_0| \frac{\max_{\alpha\in\mathcal{F}\setminus\mathcal{F}_0} p_\alpha}{p_{\alpha^\star}} e^{-\Delta}\right)\end{aligned}$$

Note that for any $\alpha \in \mathcal{F}$ we have $e^{-L_p} \leqslant p_\alpha \leqslant e^{L_p}$, further $|\mathcal{F} \setminus \mathcal{F}_0| \leqslant |\mathcal{F}| \leqslant \exp(\mathrm{poly}(\log d, m))$, hence we can pick $\Delta := \mathrm{poly}(L_p, m, \log d, \log \frac{1}{\varepsilon})$ to guarantee

$$|\mathcal{F} \setminus \mathcal{F}_0| \frac{\max_{\alpha \in \mathcal{F} \setminus \mathcal{F}_0} p_\alpha}{p_{\alpha^\star}} e^{-\Delta} \leqslant \frac{\varepsilon}{2m}.$$

For such a choice of Δ we have:

$$h(\theta, y') \leqslant \log \left(\sum_{\alpha \in \mathcal{F}_0} p_\alpha e^{\langle \alpha, y' \rangle - \langle \theta, y' \rangle} \right) + \log \left(1 + \frac{\varepsilon}{2m}\right) \leqslant \log \left(\sum_{\alpha \in \mathcal{F}_0} p_\alpha e^{\langle \alpha, y' \rangle - \langle \theta, y' \rangle} \right) + \frac{\varepsilon}{2m}.$$

Case 2: $\mathcal{F}_0$. Consider now $\alpha \in \mathcal{F}_0$, we have

$$\langle \alpha, y' \rangle - \langle \theta, y' \rangle = \langle \alpha, y^\star \rangle - \langle \theta, y^\star \rangle - (\beta_i - \Delta) \langle \alpha - \theta, a_i \rangle \leqslant \langle \alpha, y^\star \rangle - \langle \theta, y^\star \rangle$$

as $\langle \theta, a_i \rangle \leqslant \langle \alpha, a_i \rangle = b_i$ (because $\theta \in P$). Consequently, we obtain

$$\begin{aligned} h(\theta, y') &\leqslant \log \left(\sum_{\alpha \in \mathcal{F}_0} p_\alpha e^{\langle \alpha, y' \rangle - \langle \theta, y' \rangle} \right) + \frac{\varepsilon}{2m} \\ &\leqslant \left(\sum_{\alpha \in \mathcal{F}_0} p_\alpha e^{\langle \alpha, y^\star \rangle - \langle \theta, y^\star \rangle} \right) + \frac{\varepsilon}{2m} \\ &\leqslant h(\theta, y^\star) + \frac{\varepsilon}{2m}. \end{aligned}$$

It remains to argue that after performing the above procedure, the norm of y' is small. For this observe

$$\|y'\| = \|\sum_{i \in I_0}^{m} \beta_i' a_i'\| \leqslant \sum_{i \in I_0} \beta_i' \|a_i'\| \leqslant m \cdot \Delta \cdot (\sqrt{m} \cdot M) = \mathrm{poly}\left(m, \log \frac{1}{\varepsilon}\right).$$

In the above we used the fact that since a_i' is a projection of a_i onto H (for any $i \in I_0$) we have

$$\|a_i'\| \leqslant \|a_i\| \leqslant \sqrt{m} \cdot M.$$

□

Below we present a useful corollary of Theorem 7.4 which is often convenient in applications.

Corollary 7.1

Under the assumptions of Theorem 7.4, for every function $p : \mathcal{F} \to \mathbb{R}_{>0}$, for every $\varepsilon > 0$ there exists a number $R > 0$ which is polynomial in $m, \log d, M, L_p$ and $\log \frac{1}{\varepsilon}$ such that

$$\forall_{\theta \in P} \ \exists_{y \in B(0,R)} \ \ \| \frac{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle} \cdot \alpha}{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle}} - \theta \| < \varepsilon.$$

Proof of Corollary 7.1:
It is enough to establish it for $\theta \in \mathrm{int}(P)$ (relative interior is meant here, i.e. interior of P when

restricted to H). To this end consider $y^\star$ to be

$$y^\star = \operatorname*{argmin}_{y \in \mathbb{R}^m} h(\theta, y) = g(\theta).$$

It is not hard to prove that such a $y^\star$ exists, i.e., the minimum is attained, for θ in the relative interior of P. Consider now the gradient of h with θ fixed:

$$\nabla_y h(\theta, y) = \frac{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle} \cdot \alpha}{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle}} - \theta. \tag{7.6}$$

To conclude the Corollary from Theorem 7.4 one has to prove that if the value at a point is close to optimal, then the gradient is short. Towards this we show that $y \mapsto h(\theta, y)$ is $L-$smooth (for some polynomially bounded L), i.e.

$$\|\nabla_y h(\theta, y_1) - \nabla_y h(\theta, y_2)\| \leqslant L\|y_1 - y_2\|.$$

This can be deduced from the fact that the Hessian matrix of h, $\nabla_y^2 h(\theta, y)$ has polynomially bounded entries (and the bound does not depend neither on y nor on θ). Indeed, under the notation that $\alpha' = \alpha - \theta$ for $\alpha \in \mathcal{F}$ we obtain

$$\left(\nabla_y^2 h(\theta, y)\right)_{i,j} = \frac{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle} \alpha_i' \alpha_j'}{\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle}} - \frac{\left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle} \alpha_i'\right)\left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle} \alpha_j'\right)}{\left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha, y \rangle}\right)^2},$$

hence it follows

$$\left|\left(\nabla_y^2 h(\theta, y)\right)_{i,j}\right| \leqslant 2 \max_{\alpha \in \mathcal{F}} \|\alpha - \theta\|^2 \leqslant 2d^2.$$

Hence the function $y \mapsto h(\theta, y)$ is $L := 2d^2-$smooth. Now, it is well known that for a convex, $L-$smooth function, we have (see e.g. [Nes14]):

$$h\left(\theta, y + \frac{1}{2L} v\right) \leqslant h(\theta, y) - \frac{1}{4L}\|v\|^2,$$

where $v = \nabla_y h(\theta, y)$. Hence, if y is as in Theorem 7.4 then we obtain $\|\nabla_y h(\theta, y)\|^2 \leqslant 4L\varepsilon$ and consequently

$$\|\nabla_y h(\theta, y)\| \leqslant 4d\varepsilon^{1/2}.$$

The corollary follows by combining the above obtained bound with (7.6). $\square$

7.3.2 Proof of the Computability of Max-Entropy Distributions

Below we restate Theorem 7.1 in its fully general form. For definitions of the unary facet complexity, the complexity measure L_p of p and the diameter of $\mathcal{F}$ we refer to Section 7.3.1

Theorem 7.5

Let $\mathcal{F}$ be any finite subset of $\mathbb{Z}^m$ and let $d \in \mathbb{R}_{\geqslant 0}$ be its diameter, let M be the unary facet complexity of $\operatorname{conv}(\mathcal{F})$. Then, there exists an algorithm such that given a probability distribution p on $\mathcal{F}$ (via an evaluation oracle for g_p), $\theta \in P$ and an $\varepsilon > 0$, computes a vector $y \in \mathbb{R}^m$ with $\|y\| \leqslant \operatorname{poly}\left(m, M, \log d, L_p, \log \frac{1}{\varepsilon}\right)$ such that

$$\|q^y - q^\star\|_1 < \varepsilon,$$

where $q^\star$ is the optimal solution to (7.3) and q^y is a distribution over $\mathcal{F}$ defined as $q^y_\alpha = \frac{p_\alpha e^{\langle \alpha, y\rangle}}{\sum_{\mathcal{B}\in\mathcal{F}} p_\beta e^{\langle \beta, y\rangle}}$, for $\alpha \in \mathcal{F}$. The running time of the algorithm is polynomial in $m, M, d, L_p, \log \frac{1}{\varepsilon}$.

In the above g_p is a generalized counting function – the function $g_{\mathcal{F}}$ introduced in Section 7.1.2 is a special case when p is the uniform distribution over $\mathcal{F}$. An oracle for g_p is then simply defined as a procedure that given an $x > 0$ outputs

$$g_p(x) := \sum_{\alpha\in\mathcal{F}} p_\alpha x^\alpha, \qquad \text{where} \quad x^\alpha := \prod_{i=1}^m x_i^{\alpha_i}.$$

.

Proof.
For convenience without loss of generality we might assume that $\mathcal{F} \subseteq \mathbb{N}^m$ and in fact even $\mathcal{F} \subseteq [0, 2d]^m \cap \mathbb{N}^m$ where d is the diameter of $\mathcal{F}$. This is because one can always shift the set $\mathcal{F}$ (together with θ) and this operation does not affect the problem nor its parameters. To obtain a vector y, as required, we solve the dual program up to a desired precision (below we explain why is this enough). Recall that the dual program to (7.3) is given by

$$\begin{aligned} \inf \quad & h(\theta, y) = \log\left(\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle \alpha-\theta, y\rangle}\right), \\ \text{s.t.} \quad & y \in \mathbb{R}^m. \end{aligned}$$

By a direct calculation one can show that for every feasible solution q of the primal problem (7.3)

$$KL(q, q^y) = h(\theta, y) - \sum_{\alpha\in\mathcal{F}} q_\alpha \log \frac{p_\alpha}{q_\alpha},$$

where $KL(\cdot,\cdot)$ denotes the KL-divergence. In particular for $q := q^\star$

$$h(\theta, y) - g(\theta) = KL(q^\star, q^y).$$

This means that in order to obtain a distribution q^y being $\varepsilon-$close in the KL-distance to the max-entropy distribution $q^\star$ it is enough to find an $\varepsilon-$optimal solution to the dual program. Moreover, from Pinsker's inequality we have

$$\|q - q^\star\|_1 \leqslant \sqrt{2 \cdot KL(q^\star, q)}$$

hence it suffices to find a solution y to the dual program which is $\delta := \Theta(\varepsilon^2)$-optimal, to guarantee that $\|q^\star - q^y\|_1 < \varepsilon$.

To find a δ-optimal solution to the dual problem we apply the ellipsoid method. First of all we note that $h(\theta, y)$ is a convex function of y (this follows from Hölder's inequality), which is the first requirement for the ellipsoid method to be applicable. It follows from Theorem 7.5 that in order to find a δ-optimal solution to $\inf_{y\in\mathbb{R}^m} h(\theta, y)$ it is sufficient to solve

$$\min_{y\in B(0,R)} h(\theta, y),$$

where R is a certain bound, polynomial in $m, M, \log d, L_p$ and $\log \frac{1}{\delta}$. Now, following the treatment of the ellipsoid method in [BTN12] (Theorem 8.2.1) it remains to address the following issues.

1. Construct a first order oracle for $y \mapsto h(\theta, y)$, i.e. an efficient way to evaluate values $h(\theta, y)$ and gradients $\nabla_y h(\theta, y)$ of this function.
2. A bound on the gap between the maximum and minimum value of $h(\theta, \cdot)$ in $B(0, R)$. More precisely, defining $D := \max_{y\in B(0,R)} h(\theta, y) - \min_{y\in B(0,R)} h(\theta, y)$ we would like $\log D$ to be polynomially bounded.
3. Provide an outer ball – containing the domain of the considered optimization problem and an inner ball – contained in the domain. The radii of them should be of polynomial bit complexity.
4. Provide a separation oracle for the domain $B(0, R)$.

Point (1) We first note that h can be equivalently written as

$$h(\theta, y) = \log\left(\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle \alpha, y\rangle}\right) - \langle \theta, y\rangle .$$

Thus $h(\theta, y) = \log g_p(e^y) - \langle \theta, y\rangle$, where $e^y = (e^{y_1}, e^{y_2}, \ldots, e^{y_m})$ and consequently $h(\theta, y)$ can be evaluated using just the evaluation oracle for g_p. For the case of gradients observe first that

$$\nabla_y h(\theta, y) = \frac{1}{g_p(e^y)} \sum_{\alpha\in\mathcal{F}} \alpha p_\alpha e^{\langle \alpha, y\rangle} - \theta.$$

Since computing $g_p(e^y)$ is easy, it remains to deal with $\sum_{\alpha\in\mathcal{F}} \alpha p_\alpha e^{\langle \alpha, y\rangle}$. For this note that the ith coordinate of the above is

$$\sum_{\alpha\in\mathcal{F}} \alpha_i p_\alpha e^{\langle \alpha, y\rangle} = \frac{d}{dt} g_p(e^{y_1}, \ldots, e^{y_{i-1}}, e^{y_i} + t, e^{y_{i+1}}, \ldots, e^{y_m}).$$

The right hand side above is a univariate polynomial $h(t)$ of degree at most $2d$ (since $\mathcal{F} \subseteq [0, 2d]$). To compute its derivative it is enough to learn all its coefficients (in fact it is enough to learn the coefficient of t in $h(t)$). Towards this, note that the evaluation oracle for g_p implies an evaluation oracle for $t \mapsto h(t)$, and hence we can simply evaluate h at $2d + 1$ different points and recover its coefficients using polynomial interpolation. The running time of such a procedure is polynomial in d – as required.

Note also that, importantly, to implement the first order oracle for $h(\theta, \cdot)$ the oracle g_p is queried only on inputs of polynomial bit complexity, as for every $y \in B(0, R)$ the vector e^y has polynomial bit complexity (since $R = \text{poly}\left(m, \log \frac{1}{\delta}\right)$). Thus the running time of these procedures is polynomial in $m, \log \frac{1}{\delta}$ and d.

Point (2) Note that from the Cauchy-Schwarz inequality, for every $y \in B(0, R)$

$$\langle y, \alpha - \theta \rangle \leqslant \|y\| \|\alpha - \theta\| \leqslant R \cdot d,$$

hence, we have

$$\max_{y \in B(0,R)} h(\theta, y) \leqslant \log \left(e^{L_p} |\mathcal{F}| \cdot e^{R \cdot d} \right) \leqslant L_p \cdot R \cdot d \cdot \log |\mathcal{F}|.$$

Similarly, for the minimum

$$\min_{y \in B(0,R)} h(\theta, y) \geqslant -L_p \cdot R \cdot d \cdot \log |\mathcal{F}|.$$

Clearly the logarithm of the gap:

$$\log \left(\max_{y \in B(0,R)} h(\theta, y) - \min_{y \in B(0,R)} h(\theta, y) \right) \leqslant \log(2L_p \cdot R \cdot d \cdot \log |\mathcal{F}|)$$

is polynomially bounded (even without taking the logarithm it is still true), as $|\mathcal{F}| \leqslant (2d)^m$.
Point (3) The outer ball is the domain itself: $B(0, R)$; R is polynomially bounded (note that in fact we only need that $\log R$ is polynomially bounded). For the inner ellipsoid we can just take $B(0, 1)$.
Point (4) This is clear - given a vector $y_0 \in \mathbb{R}^m$, if $\|y\| \leqslant R$ we just report y_0 to be in the domain and if $\|y_0\| = R' > R$ then $\{y \in \mathbb{R}^m : \langle y, y_0 \rangle = \frac{R+R'}{2}\}$ is the required separating hyperplane.

□

7.3.3 Proof of the Bit Complexity Lower Bound

We start by proving a technical fact which will be useful in establishing the large bit complexity example.

Fact 7.1

Let $v_1, v_2, \ldots, v_N \subseteq \mathbb{R}^m$ be a set of vectors and denote $\delta := \operatorname{dist}(0, \operatorname{conv}(v_1, v_2, \ldots, v_N))$. Assume that $\delta > 0$ and consider the optimal value of the following optimization problem:

$$\tau = \min \left\{ \|y\| : y \in \mathbb{R}^m, \langle y, v_i \rangle \leqslant -1 \text{ for all } i = 1, 2, \ldots, N \right\},$$

then $\tau \geqslant \frac{1}{\delta}$.

Proof.
We formulate the problem as a convex, quadratic program with linear constraints:

$$\begin{aligned} \min \quad & \|y\|^2, \\ \text{s.t.} \quad & \langle y, \alpha \rangle \leqslant -1 \quad \text{for all } \alpha \in \mathcal{F}'. \end{aligned} \tag{7.7}$$

To derive a lower bound on the optimal value of (7.7) we consider the dual program:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \lambda_i - \frac{1}{4} \|\sum_{i=1}^{N} \lambda_i v_i\|^2, \\ \text{s.t.} \quad & \lambda_i \geqslant 0 \quad \text{for all } i = 1, 2, \ldots, N. \end{aligned} \tag{7.8}$$

From weak duality we know that the optimal value of (7.8) is a lower bound to (7.7) thus we just need to provide a feasible solution to the dual program. To this end let v be the shortest vector in the convex hull of $v_1, v_2, \ldots, v_N$, i.e., $v \in \operatorname{conv}(v_1, v_2, \ldots, v_N)$ and $\|v\| = \delta$. v can be written as

$$v = \sum_{i=1}^{N} \mu_i v_i$$

for some $\mu \geqslant 0$ with $\sum_{i=1}^{N} \mu_i = 1$. Consider now $\lambda := \frac{2}{\delta^2}\mu \in \mathbb{R}^N_{\geqslant 0}$. The dual objective value for λ is

$$\sum_{i=1}^{N} \lambda_i - \frac{1}{4}\|\sum_{i=1}^{N} \lambda_i v_i\|^2 = \frac{2}{\delta^2} - \frac{1}{4} \cdot \frac{4}{\delta^4}\|\sum_{i=1}^{N} \mu_i v_i\|^2 = \frac{1}{\delta^2}.$$

This provides us with a lower bound of $\frac{1}{\delta^2}$ on the optimal value of (7.7) and thus a lower bound of $\frac{1}{\delta}$ on the optimization problem as in the statement of the Fact. □

Remark 7.1

It is not hard to prove that in Fact 7.1 the value $\frac{1}{\tau}$ is not only a lower bound but is actually equal to the optimal value of the considered optimization problem. This can be established by plugging in an appriopiate scaling of the shortest vector in $\operatorname{conv}(v_1, v_2, \ldots, v_N)$ for y.

Proof Theorem 7.3:
Our construction of $\mathcal{F}$ is based on existence of "flat" 0-1 polytopes, as established by [AV97]. There exist $m+1$ affinely independent points $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m \in \{0,1\}^m$ such that if we let $H = \alpha_1 + \operatorname{span}\{\alpha_2 - \alpha_1, \ldots, \alpha_m - \alpha_1\}$ (the $(m-1)-$dimensional affine subspace containing all points $\alpha_1, \ldots, \alpha_m$) then

$$\operatorname{dist}(\alpha_0, H) = e^{-\Omega(m \log m)}.$$

Without loss of generality we assume that $\alpha_0 = 0$. Let $y \in H$ be the projection of $\alpha_0 = 0$ onto H, i.e., a point such that

$$\langle y, \alpha_i - y \rangle = 0 \qquad \text{for every } i = 1, 2, \ldots, m.$$

Consider the lattice

$$L = \alpha_1 + \sum_{i=2}^{m} (\alpha_i - \alpha_1) \cdot \mathbb{Z},$$

with the origin at α_1, with basis $\{(\alpha_i - \alpha_1)\}_{2 \leqslant i \leqslant m}$. Since the hyperplane H is covered by disjoint copies of the fundamental parallelepiped

$$F := \left\{ \sum_{i=2}^{m} \beta_i(\alpha_i - \alpha_1) : \beta_2, \ldots, \beta_m \in [0,1) \right\},$$

there is an integer translation of F which contains the point y. More formally, there exists an integer vector $\gamma \in \mathbb{Z}^m$ such that

$$y \in \gamma + F \subseteq H.$$

Note now that by denoting $F' = \gamma + F$ we obtain

$$\operatorname{diam}(F') = \operatorname{diam}(F) \leqslant m^{3/2},$$

since every vertex of F has integer coordinates in the range $[0, m]$. Let $\alpha \in \mathbb{Z}^m$ be now any of the 2^m vertices of F'. Since y belongs to F' we have

$$\|\alpha\| \leqslant \|y\| + \operatorname{diam}(F') \leqslant O(1) + m^{3/2}.$$

Let $\mathcal{F}'$ be a subset of $\mathbb{Z}^m$ consisting of all the vertices of F' and let $\mathcal{F} := \mathcal{F}' \cup \{0\}$. Further, define $\theta := 0$. We prove that the conclusion of the Lemma holds under such a choice of $\mathcal{F}$ and θ.

Towards this we first note that affine hull of $\mathcal{F}'$ is equal to H, as $F' \subseteq H$ and its vertices clearly span H. Moreover the point y, which is the projection of 0 onto H belongs to the convex hull of $\mathcal{F}'$. Let $\delta > 0$ be the distance between 0 and H and let $a \in \mathbb{R}^m$ (with $\|a\| = 1$) be the normal vector of H, in other words

$$H = \{x \in \mathbb{R}^m : a^\top x = \delta\}.$$

Note that the gradient of $h(\theta, y)$ with respect to y is given by:

$$\nabla_y h(\theta, y) = \frac{\sum_{\alpha \in \mathcal{F}} \alpha \cdot e^{\langle y, \alpha \rangle}}{\sum_{\alpha \in \mathcal{F}} e^{\langle y, \alpha \rangle}}.$$

Since h is $L-$smooth for some $L = \operatorname{poly}(m)$ (see the proof of Corollary 7.1), we know that points with a large-magnitude gradient cannot be close to optimal. Quantitatevely, we have

$$|g(\theta) - h(\theta, y)| \geqslant \frac{\|\nabla_y h(\theta, y)\|^2}{L}.$$

Thus to prove that $|g(\theta) - h(\theta, y)| \geqslant \varepsilon$ (for some $\varepsilon > 0$) it is enough to prove that $\|\nabla_y h(\theta, y)\| \geqslant \sqrt{\varepsilon L}$. We pick ε to be $\frac{\delta^2}{e^4 \cdot L}$, then $\varepsilon = e^{-O(m \log m)}$. Moreover, the condition $|g(\theta) - h(\theta, y)| < \varepsilon$ implies that $\|\nabla_y h(\theta, y)\| < \frac{\delta}{e^2}$. We prove that the latter is possible only when $\|y\| \geqslant e^{\Omega(m \log m)}$.

Indeed, assume that $\|\nabla_y h(\theta, y)\| < \frac{\delta}{e^2}$. By the Cauchy-Schwarz inequality we have

$$\langle a, \nabla_y h(\theta, y) \rangle \leqslant \|\nabla_y h(\theta, y)\| \cdot \|a\| = \|\nabla_y h(\theta, y)\|$$

and moreover

$$\langle a, \nabla_y h(\theta, y) \rangle = \frac{0 + \sum_{\alpha \in \mathcal{F}'} \langle a, \alpha \rangle \, e^{\langle y, \alpha \rangle}}{\sum_{\alpha \in \mathcal{F}} e^{\langle y, \alpha \rangle}} = \delta \cdot \frac{\sum_{\alpha \in \mathcal{F}'} e^{\langle y, \alpha \rangle}}{1 + \sum_{\alpha \in \mathcal{F}'} e^{\langle y, \alpha \rangle}}.$$

It follows that

$$\frac{\sum_{\alpha \in \mathcal{F}'} e^{\langle y, \alpha \rangle}}{1 + \sum_{\alpha \in \mathcal{F}'} e^{\langle y, \alpha \rangle}} < \frac{1}{e^2},$$

and consequently $\sum_{\alpha \in \mathcal{F}'} e^{\langle y, \alpha \rangle} < \frac{1}{e}$, which implies in particular that

$$\forall_{\alpha \in \mathcal{F}'} \quad \langle y, \alpha \rangle \leqslant -1. \tag{7.9}$$

The question we would like to answer is: what is the shortest $y \in \mathbb{R}^m$ which satisfies condition (7.9)? This will give us a lower bound on $\|y\|$ satisfying $|g(\theta) - h(\theta, y)| < \varepsilon$. To answer this question, we apply Fact 7.1 and conclude that every such y has length at least $\frac{1}{\delta}$. As $\delta = e^{-\Omega(m \log m)}$ we conclude that the optimal solution $y^\star$ to (7.7) satisfies $\|y^\star\| = e^{\Omega(m \log m)}$ and the Lemma follows by contraposition. □

7.3.4 Proofs of Applications

Preliminaries on Real Stability

In this section state two important facts on real stable polynomials which appear in some applications of our results. For a survey on real stable polynomials we refer the reader to [Wag11].

Fact 7.2 (*[BC95, Brä07]*)

Let $p \in \mathbb{R}[x_1, \dots, x_m]$ be a real stable polynomial with non-negative coefficients. Then, there exists a "rank function" $r : \{-1, 0, 1\}^m \to \mathbb{Z}$ is, such that the convex hull of $\mathcal{F} \subseteq \mathbb{N}^m$ – the support of p can be described as:

$$\mathrm{conv}(\mathcal{F}) = \{x \in \mathbb{R}^m : \ \forall_{c \in \{-1,0,1\}^m} \ \langle x, c \rangle \leqslant r(c)\}.$$

Proof.
The proof is a simple consequence of two results. It was proved in [Brä07] that the support of a real stable polynomial is a *jump system*. Such sets were studied previously in [BC95], where a polyhedral characterization, as in the conclusion, was shown. □

Fact 7.3 (*[Gül97]*)

Let $p \in \mathbb{R}[x_1, \dots, x_m]$ be a real stable polynomial with non-negative coefficients. Then the function $x \mapsto \log p(x)$ is concave over $x \in \mathbb{R}^m_{>0}$.

Bounds for the Matrix Scaling Problem

Consider the $(r, c)-$matrix scaling problem, where one is given a nonnegative square matrix $A \in \mathbb{R}^{n \times n}$ and two vectors $r, c \in \mathbb{N}^n$ (with $\|r\|_1 = \|c\|_1$) and the goal is to find a scaling: two positive vectors $x, y \in \mathbb{R}^n_{>0}$ such that for B defined as $B := XAY$ (with $X = \mathrm{Diag}\,(x)$ and $Y = \mathrm{Diag}\,(y)$) it holds that $B\mathbb{1} = r$ and $B^\top \mathbb{1} = c$ where $\mathbb{1} \in \mathbb{R}^n$ is the all-one vector. In other words, we want the row-sums of the matrix B to be equal to r and column sums to be equal to c.

One can prove that if such a scaling exists even *asymptotically* (i.e., a sequence of scalings exists such that they satisfy the scaling condition in the limit), then it can be recovered from the optimal solution to the following convex program

$$\inf_{z \in \mathbb{R}^n} \sum_{i=1}^n r_i \log \left(\sum_{j=1}^n A_{i,j} e^{z_j} \right) - \langle c, z \rangle \, ; \tag{7.10}$$

see [Gur06, KLRS08]. Indeed, the scaling is recovered as $x_i := r_i \cdot \left(\sum_{l=1}^n A_{i,l} e^{z_l^\star} \right)^{-1}$ for $i = 1, 2, \dots n$ and $y_j := e^{z_j^\star}$ for $i = 1, 2, \dots, n$, where $z^\star$ stands for the optimal (or approximately optimal) solution to (7.10). The question which arises naturally is: does the optimal (or approximately optimal) scaling (x, y) have polynomial bit complexity? Or in other words: can we prove that the vector $z^\star$ has polynomially bounded entries: $\|z\|_\infty = \mathrm{poly}(n, L_A)$? Here L_A denotes the bit complexity of the matrix A, i.e., $L_A := \max_{i,j \in [n]} |\log A_{i,j}|$. We derive the following Corollary of Theorem 7.4.

Corollary 7.2

Let $A \in \mathbb{R}^{n\times n}$ be a nonnegative matrix which is asymptotically $(r,c)-$scalable with $\|r\|_1 = \|c\|_1 = h$. Then for every $\varepsilon > 0$ there exists a scaling (x,y) such that:

$$\left|\sum_{k=1}^{n} x_i y_k A_{i,k} - r_i\right| < \varepsilon \quad \textit{for all } i \in [n],$$

$$\left|\sum_{l=1}^{n} x_l y_j A_{l,j} - c_j\right| < \varepsilon \quad \textit{for all } j \in [n],$$

and the bit complexities of all entries of x and y, i.e., $\max_{i\in[n]} |\log x_i|$ *and* $\max_{i\in[n]} |\log y_i|$ *are bounded by* $\operatorname{poly}\left(n, L_A, h, \log\frac{1}{\varepsilon}\right)$.

Proof.
Our strategy is to derive the result from Corollary 7.1. We first rewrite the objective (7.10) so that it matches the form as in Theorem 7.4.

$$\begin{aligned}\sum_{i=1}^{n} r_i \log\left(\sum_{j=1}^{n} A_{i,j}e^{z_j}\right) - \langle c, z\rangle &= \log\left(\prod_{i=1}^{n}\left(\sum_{j=1}^{n} A_{i,j}e^{z_j}\right)^{r_i}\right) - \langle c,z\rangle \\ &= \log\left(\sum_{\alpha\in\mathcal{F}} p_\alpha e^{\langle\alpha - c, z\rangle}\right) \\ &= h(c,z)\end{aligned}$$

For some set $\mathcal{F} \subseteq \mathbb{N}^m$ and some family of positive numbers $\{p_\alpha\}_{\alpha\in\mathcal{F}}$.

The next step is to obtain bounds on the various quantities m, d, M, L_p which appear in Corollary (7.1). Clearly we choose $m := n$. Next observe that $\|\alpha\|_\infty \leqslant h$ for $\alpha \in \mathcal{F}$, hence $d \leqslant h$ and that $|\log p_\alpha| \leqslant \operatorname{poly}(h, L_A)$ for all $\alpha \in \mathcal{F}$, hence $L_p = \operatorname{poly}(h, L_A)$.

Let us now prove that the polytope $\operatorname{conv}(\mathcal{F})$ can be described by inequalities with small integer coefficients. To this end note that p can be naturally treated as a polynomial (p_α is the coefficient of $x^\alpha := \prod_{i\in[n]} x_i^{\alpha_i}$).

$$p(x) = \sum_{\alpha\in\mathcal{F}} p_\alpha x^\alpha = \prod_{i=1}^{n}\left(\sum_{j=1}^{n} A_{i,j}x_j\right)^{r_i},$$

and the support of p is equal to $\mathcal{F}$. Since p is a product of linear polynomials with nonnegative coefficients, it is a *real stable polynomial* (see preliminaries at the beginning of this section for some background). As shown in the preliminaries (Fact 7.2), the Newton polytope of a real stable polynomial can be described by inequalities of the form

$$\langle a, x\rangle \leqslant b$$

where $a \in \{-1, 0, 1\}^n$. Hence we can take $M = O(1)$ in the statement of Corollary 7.1.

We now apply Corollary 7.1 to obtain a point $z^\star$ and use it to define a scaling by the following formulas

$$x_i = r_i \cdot \left(\sum_{l=1}^{n} A_{i,l} e^{z_l^\star}\right)^{-1} \qquad \text{and} \qquad y_j = e^{z_j^\star}, \qquad \text{for } i,j = 1,2,\ldots,n.$$

Note that such a pair (x, y) has bit complexity which is polynomial in $n, L_A, h, \log \frac{1}{\varepsilon}$, hence it remains to reason about the precision of the resulting scaling.

By a direct calculation one obtains that

$$\left|\sum_{k=1}^{n} x_i y_k A_{i,k} - r_i\right| = 0 \qquad \text{for all } i \in [n],$$

hence it remains to prove a bound on the precision of the scaling with respect to columns. For this, note that

$$(\nabla_z h(c, z))_j = \sum_{l=1}^{n} x_l y_j A_{l,j} - c_j.$$

Since Corollary 7.1 implies that

$$\|\nabla_z h(c, z^\star)\|_2 < \varepsilon$$

the bound on the scaling precision follows. □

Computability of Recent Continuous Relaxations for Counting and Optimization Problems

We are interested in solving the following optimization problem studied in Chapter 6.

$$\mathrm{Cap}_{\mathcal{B}}(p) := \sup_{\theta \in P(\mathcal{B})} \inf_{x>0} \frac{p(x)}{\prod_{i=1}^{m} x_i^{\theta_i}},$$

where $p \in \mathbb{R}[x_1, x_2, \ldots, x_m]$ is a polynomial with non-nagative coefficients and $\mathcal{B} \subseteq \{0,1\}^m$ is a certain family of sets (points in the binary cube). We prove the following

Theorem 7.6

Let $p \in \mathbb{R}[x_1, x_2, \ldots, x_m]$ be a polynomial with nonnegative coefficients with support $\mathcal{F} \subseteq \mathbb{N}^m$ and let $\mathcal{B} \subseteq \mathbb{N}^m$. Asumme that the unary facet complexity of $\mathrm{conv}(\mathcal{F})$ is M and denote $d = \max(\mathrm{diam}(\mathcal{F}), \mathrm{diam}(\mathcal{B}))$. There is an algorithm which given an evaluation oracle for p, a separation oracle for $\mathrm{conv}(\mathcal{F})$, a separation oracle for $\mathrm{conv}(\mathcal{B})$ and an $\varepsilon > 0$ computes a number X such that $1 - \varepsilon < \frac{X}{\mathrm{Cap}_{\mathcal{B}}(p)} < 1 + \varepsilon$ in time $\mathrm{poly}(m, L_p, \log d, M, \log \frac{1}{\varepsilon})$.

Note that in the above theorem we require separation oracle access to the corresponding Newton polytope: this is actually not necessary in various cases, in particular when p is real stable, in which case such a separation oracle can be constructed given access to an evaluation oracle to p only (see Chapter 6).

Proof.
Denote by $P \subseteq \mathbb{R}^m$ the convex hull of $\mathcal{F}$ – the support of p. We observe that $\mathrm{Cap}_{\mathcal{B}}(p)$ can we

rewritten equivalently as

$$\log \mathrm{Cap}_{\mathcal{B}}(p) = \sup_{\theta \in P(\mathcal{B})} \inf_{y \in \mathbb{R}^m} \log \left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha - \theta, y \rangle} \right) = \sup_{\theta \in P(\mathcal{B})} \inf_{y \in \mathbb{R}^m} h(\theta, y),$$

where $\mathcal{F} \subseteq \mathbb{N}^m$ denotes the support of p. Thus we obtain a form of $\log \mathrm{Cap}_{\mathcal{B}}(p)$ in terms of a function h as in (7.4). Let us denote

$$g(\theta) = \inf_{y \in \mathbb{R}^m} \log \left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha - \theta, y \rangle} \right).$$

Hence the goal is to solve $\sup_{\theta \in P(\mathcal{B})} g(\theta)$. In fact, since $g(\theta) = -\infty$ whenever $\theta \notin P$, we can rewrite it equivalently as

$$\sup_{\theta \in P(\mathcal{B}) \cap P} g(\theta).$$

Importantly, since we are only interested in an additive ε-approximation of the above quantity we can apply Theorem 7.4 to replace g by the following Lipscthiz proxy:

$$\widetilde{g}(\theta) := \inf_{y \in B(0,R)} h(\theta, y)$$

for an appropriate number R – polynomial in the input size and $\log \frac{1}{\varepsilon}$ (as in Theorem 7.4).

We now apply the ellipsoid method to find an additive ε-approximation to

$$\sup_{\theta \in P(\mathcal{B}) \cap P} \widetilde{g}(\theta).$$

Firstly, observe that the function $\widetilde{g}$ is concave – as a pointwise infimum of affine functions. Now, following the treatment of the ellipsoid method in [BTN12] (Theorem 8.2.1) we have to address the following requirements to obtain polynomial running time

(1) Construct a first order oracle for $\theta \mapsto \widetilde{g}(\theta)$, i.e. an efficient way to evaluate values $\widetilde{g}(\theta)$ and (sub)gradients of this function for $\theta \in P(\mathcal{B}) \cap P$.

(2) A bound on the gap between the maximum and minimum value of $\widetilde{g}(\theta)$ in $P(\mathcal{B}) \cap P$. More precisely, defining $D := \max_{\theta \in P(\mathcal{B}) \cap P} \widetilde{g}(\theta) - \min_{\theta \in P(\mathcal{B}) \cap P} \widetilde{g}(\theta)$ we would like $\log D$ to be polynomially bounded.

(3) Provide an outer ball – containing the domain of the considered optimization problem and an inner ball – contained in the domain. The radii of them should be of polynomial bit complexity.

(4) Provide a separation oracle for the domain $P(\mathcal{B}) \cap P$.

Point (1). To obtain an evaluation oracle for $\widetilde{g}$ note that Theorem 7.5 gives an algorithm to compute $\widetilde{g}$ up to δ additive error in time polynomial in $\log \frac{1}{\delta}$. This provides an approximate (weak) evaluation oracle, which is still enough to run the ellipsoid method (using shallow cuts, see [GLS88]).

Let us now discuss the oracle for the gradient of $\widetilde{g}$. It is a standard fact in convex programming that for any $\theta \in P$ it holds that the point

$$y^\star := \operatorname*{argmin}_{y \in B(0,R)} h(\theta, y)$$

is a subgradient of $\widetilde{g}$ at θ. Using Theorem 7.5 we can efficiently compute a vector y which is a δ-approximation to $y^\star$ in the sense that $h(\theta, y) - h(\theta, y^\star) \leqslant \delta$. We use this algorithm to implement an approximate subgradient oracle for $\widetilde{g}$, i.e., for a given θ we output the vector y as above, using the algorithm in Theorem 7.5.

It remains to justify why is such an approximate gradient enough to run the ellipsoid method. Note that if $\theta \in P$ and $y \in B(0,R)$ is such that $h(\theta, y) \leqslant \widetilde{g}(\theta) + \delta$ then for any $\theta' \in P$ we have

$$g(\theta') \leqslant h(\theta', y) = h(\theta, y) + \langle \theta' - \theta, y \rangle \leqslant g(\theta) + \delta + \langle \theta' - \theta, y \rangle,$$

where the equality above holds because h is an affine function of θ. Hence, by applying such a y as a gradient oracle, we obtain a separation hyperplane which never cuts out a point $\theta' \in P$ of value $g(\theta') > g(\theta) + \delta$. This, along with the fact that y can be found in time polynomial in $\log \frac{1}{\delta}$, allows us to apply the shallow cut ellipsoid method. We also refer the reader to the full version of [SV14] and the proof of Theorem 2.11 therein where a detailed discussion of an ellipsoid algorithm for a related problem is provided.

Point (2). As in the proof of Theorem 7.5 we observe that for every $y \in B(0,R)$

$$-L_p \cdot R \cdot m \cdot d \log d \leqslant \log\left(\sum_{\alpha \in \mathcal{F}} p_\alpha e^{\langle \alpha - \theta, y \rangle}\right) \leqslant L_p \cdot R \cdot m \cdot d \log d,$$

and hence the gap (and hence clearly its logarithm as well) is polynomially bounded.

Point (3). An outer ball is easy to obtain since $P(\mathcal{B}) \subseteq [0,1]^m$. For the inner ball, suppose first that $P(\mathcal{B}) \cap P$ is full-dimensional. Then, since the vertices of both these polytopes have small integer entries, one can show using standard techniques that a ball of radius $\Omega(e^{-\mathrm{poly}(m)})$ can be fit inside this polytope. In the non-full-dimensional case one has to work in a linear subspace $H \subseteq \mathbb{R}^m$ on which $P(\mathcal{B}) \cap P$ is full-dimensional; H can be found given separation oracles of $P(\mathcal{B})$ and P using standard subspace identification techniques.

Point (4). This is clear as we are given separation oracles for both $P(\mathcal{B})$ and P.

$\square$

Computability of worst-case Brascamp-Lieb constants

Given a sequence of matrices $B_j \in \mathbb{R}^{n_j \times n}$ for $j \in [m]$ and a vector $p \in \mathbb{R}^m_{\geqslant 0}$ consider the following Brascamp Lieb inequality:

$$\int_{\mathbb{R}^n} \prod_{j=1}^m f_j(B_j x)^{p_j} dx \leqslant C \prod_{j=1}^m \left(\int_{\mathbb{R}^{n_j}} f_j(x_j) dx_j\right)^{p_j},$$

for any sequence of integrable functions $f_j : \mathbb{R}^{n_j} \to \mathbb{R}_{\geqslant 0}$. These capture in particular the Hölder inequality and Loomis-Whitney inequalities as special cases [BL02, Lie90]. The best constant C for which the above holds universally is called the Brascamp-Lieb constant and can be computed [Lie90] as $\mathrm{BL}(B,p)^{-2}$ (B stands here for the collection of all matrices $B_1, B_2, \ldots, B_m$) where

$$\mathrm{BL}(B,p) = \inf \left\{ \frac{\det\left(\sum_{j=1}^m p_j B_j^\top X_j B_j\right)}{\prod_{j=1}^m \det(X_j)^{p_j}} : X_j \in \mathbb{R}^{n_j \times n_j}, X_j \succeq 0, j = 1,2,\ldots,m \right\}.$$

The constant $\mathrm{BL}(B,p)$ is non-zero whenever p belongs to the so called Brascamp-Lieb polytope, which can be described as follows

$$P_B = \left\{ p \in \mathbb{R}^m_{\geqslant 0} : \sum_{j=1}^m p_j \dim(B_j U) \geqslant \dim(U), \text{ for every lin. subspace } U \subseteq \mathbb{R}^n \right\}.$$

Ee ask a question of computability of the worst possible Brascamp-Lieb constant over the whole Brascamp-Lieb polytope, i.e. what $\sup_{p\in P_B} \mathrm{BL}(B,p)$ is. This quantity can be used in particular as a universal constant (for any $p \in P_B$) for the so called reverse Brascamp-Lieb inequality [Bar98]. We prove that it can be computed efficiently when all the matrices $B_1, B_2, \ldots, B_m$ are of rank 1.

Theorem 7.7

Consider a sequence of matrices $B_1, B_2, \ldots, B_m \in \mathbb{R}^{1\times n}$. The worst-case Brascamp-Lieb constant $\sup_{p\in P_B} \mathrm{BL}(B,p)$ can be computed up to precision $\varepsilon > 0$ in time polynomial in the description size of $B_1, B_2, \ldots, B_m$, m and $\log \frac{1}{\varepsilon}$.

Proof.
Let us denote by $v_j \in \mathbb{R}^n$ the only row of the matrix B_j and by $V \in \mathbb{R}^{m\times n}$ the matrix collecting all the v_j's as rows. Then the the Brascamp-Lieb constant (for the rank-1 case) can be computed as

$$\mathrm{BL}(B,p) = \inf_{x>0} \frac{\det\left(\sum_{j=1}^m p_j x_j v_j v_j^\top\right)}{\prod_{j=1}^m x_j^{p_j}}.$$

Note that the numerator is simply a polynomial $r(x) = \sum_{S\subseteq[m],|S|=n} p^S x^S \det(V_S V_S^\top)$, where V_S is the submatrix of V corresponding to the index set $S \subseteq [m]$ and $x^S := \prod_{i\in S} x_i$.

Then, the problem of computing the (logarithm of the) worst-case constant can be reformulated as

$$\sup_{p\in P_B} \inf_{y\in\mathbb{R}^m} \log\left(\sum_{\alpha\in\mathcal{F}} r_\alpha(p) e^{\langle \alpha - p, y\rangle}\right),$$

where $\mathcal{F} \subseteq \{0,1\}^m$ is the support of r and $r_\alpha(p)$ is the coefficient of α in r, as a function of $p \in P_B$. Thus, the inner optimization problem is the dual of a certain max-entropy program. Moreover, the function

$$p \mapsto \log\left(\sum_{\alpha\in\mathcal{F}} r_\alpha(p) e^{\langle \alpha - p, y\rangle}\right) = \log\left(\sum_{S\subseteq[m],|S|=n} p^S x^S \det(V_S V_S^\top)\right)$$

is concave over $\mathbb{R}^m_{>0}$, as the polynomial $p \mapsto \sum_{S\subseteq[m],|S|=n} p^S x^S \det(V_S V_S^\top)$ (treating $x > 0$ as a vector of positive constants) is real stable, see Fact 7.3. From this point on, the argument follows along the same lines as the proof of Theorem 7.6.

□

7.4 Conclusion

7.4.1 Future Work

As shown in Theorem 7.3, some assumptions on the family $\mathcal{F}$ are necessary for polynomial bit-complexity bounds of near-optimal solutions. However, it is very likely that Theorem 7.4 still holds under weaker assumptions. Notably, our lower-bound technique does not yield examples of $\mathcal{F}$ in the hypercube $\{0,1\}^m$, but requires larger entries. One can then ask a question whether the conclusion of Theorem 7.4 holds for all sets $\mathcal{F} \subseteq \{0,1\}$.

7.4.2 Notes

The content of this chapter is based on the work [SV17a]. The max-entropy principle has been introduced in [Jay57a, Jay57b]. Max-entropy distributions arise and have found numerous applications in information theory [Jay57a, Jay82], machine learning [PPL97, Nig99], economics [AFHS97, Vin06], physics [MS06] and statistics [Soo00, SW87]. In Theoretical Computer Science, max-entropy distributions over spanning trees in a graph have been used to derive approximation algorithms for the TSP [AGM$^+$10, OSS11] and max-min fair allocation problem [AS10].

A variant of Theorem 7.2 which applies to families $\mathcal{F} \subseteq \mathbb{N}^m$ that are spanning trees of undirected graphs was obtained in [AGM$^+$10] and, for matroids (and more generally – jump systems) in [AO17]. The notion of unary facet complexity, we use in this chapter, is similar to that *binary facet complexity* defined in [GLS88].

Fast algorithms for the matrix scaling problem have been recently derived in [ALOW17, CMTV17]. For applications of matrix scaling we refer to [ALOW17, CMTV17], where recently fast algorithms for matrix scaling were recently derived.

Recently [GGOW17] gave a method for calculating the Brascamp-Lieb constant in polynomial time when the vector p is given in unary.

In optimizaton, the dual of the max-entropy program can be viewed as an instance of geometric programming. Geometric programs have been widely studied and appear in a number of applications; see the survey by Boyd *et al.* [BKVH07]. The crucial difference is that in the standard description of a Geometric Program the *posynomial* (the objective) is given explicitly while in our setting, the number of monomials is exponential.

Chapter 8

Subdeterminant Maximization via Nonconvex Relaxations and Anti-Concentration

In this chapter we study the problem of maximizing subdeterminants under constraints, i.e., $\max_{S\in\mathcal{B}} \det(L_{S,S})$ for a PSD matrix $L \in \mathbb{R}^{m\times m}$ and a family of subsets $\mathcal{B} \subseteq 2^{[m]}$. We provide approximation algorithms for the cases of $\mathcal{B}$ being a family of bases of either a partition matroid or a regular matroid. To this end we introduce certain non-convex relaxations for these problems and solve them by simple random sampling. We recover a bound on the approximation guarantee by proving an anti-concentration inequality.

8.1 Preliminaries and Statement of Results

Anti-concentration inequality. We start by describing the common component to both our algorithmic results – an anti-concentration inequality. We consider multi-variate functions in which each variable is uniformly and independently distributed over a probability simplex. Roughly, our anti-concentration inequality says that if the restriction of such a function along each variable has a certain *anti-concentration* property then the function is anti-concentrated over the entire domain. Formally, the anti-concentration result applies whenever the multi-variate function satisfies the following property.

Definition 8.1 (*Anti-concentrated functions*)

For $\gamma \geqslant 1$, a nonnegative measurable[1] *function $f : \Delta_d \to \mathbb{R}$ is called* **γ-anti-concentrated** *if for every $c \in (0,1)$*

$$\mathbf{Pr}\left[f(x) \geqslant c\cdot \mathrm{OPT}\right] \geqslant 1-\gamma d c,$$

where x is drawn from the uniform distribution over Δ_d and $\mathrm{OPT} := \max_{z\in\Delta_d} f(z)$ is the maximum value f takes on Δ_d.[2]

Similarly, for any $r \geqslant 1$ and any $p_1, p_2, \ldots, p_r \geqslant 0$, a nonnegative function $f : \prod_{i=1}^{r} \Delta_{p_i} \to \mathbb{R}$ is said to be γ-anti-concentrated if for every coordinate $i \in \{1, 2, \ldots, r\}$, and for every choice of $a_j \in \Delta_{p_j}$ for $j \neq i$, the function $x \mapsto f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_r)$ is γ-anti-concentrated.

Perhaps one of the simplest examples of an anti-concentrated function is the univariate map $t \mapsto |at+b|$ over the domain $[0,1]$. It is not hard to see that it satisfies the condition of Definition 8.1 for $\gamma = 2$ (see Lemma 8.2). It also follows that for every multi-affine polynomial $p \in \mathbb{R}[x_1, x_2, \ldots, x_r]$ the function $x \mapsto |p(x)|$ is 2-anti-concentrated. Another class of functions that satisfy such an anti-concentration property arise by considering norms and volumes in Euclidean spaces; for instance, functions of the form $t \mapsto \|ut + (1-t)v\|_2$ for vectors u, v.

Theorem 8.1 (*Anti-concentration inequality*)

Let $\gamma \geqslant 1$ be a constant. Let $r \geqslant \gamma$ and $p_1, \ldots, p_r$ be positive integers. For every γ-anti-concentrated function $f : \prod_{i=1}^r \Delta_{p_i} \to \mathbb{R}$, if x is sampled from the uniform distribution on $\prod_{i=1}^r \Delta_{p_i}$, then

$$\mathbf{Pr}\left[f(x) \geqslant (\gamma e^2)^{-r} \cdot \prod_{i=1}^r \frac{1}{p_i} \cdot \mathrm{OPT}\right] \geqslant \frac{1}{2},$$

where $\mathrm{OPT} := \max\{f(z) : z \in \prod_{i=1}^r \Delta_{p_i}\}$ *is the maximum value f takes on its domain.*

Consequently, the value of a γ-anti-concentrated function at a random point in its domain gives an estimate of its maximum value. In the simplest non-trivial case, it applies to multi-affine functions over the hypercube $[0,1]^r$ and says that the value of the function at a random point is at least c^{-r} times its optimal value, with significant probability (where $c > 1$ is an absolute constant). It is also easy to see that the bound in Theorem 8.1 is tight: For $p(x) = \prod_{i=1}^r x_i$, one can show that the probability that $|p(x)| \geqslant (3/4)^r$ over a random choice of $x \in [0,1]^r$ is exponentially small.

As an important special case of Theorem 8.1, consider the setting in which $p_i = 2$ for $i = 1, 2, \ldots, r$ (i.e., the domain is the hypercube $[0,1]^r$) and $f(x) := |p(x)|$ where $p \in \mathbb{R}[x_1, \ldots, x_r]$ is a multi-affine polynomial. Using the fact noted earlier that such an f is 2-anti-concentrated, we conclude from Theorem 8.1 that for some absolute constant $c > 1$ and a uniformly random choice of $x \in [0,1]^r$,

$$\mathbf{Pr}\left[|p(x)| \geqslant c^{-r} \cdot \max_{z \in [0,1]^r} |p(z)|\right] \geqslant 1/2.$$

This gives us a way to *estimate* the maximum of $|p(x)|$ over $[0,1]^r$ by just evaluating it on a certain number of random points and outputting the largest one. However, this observation does not directly give us much insight about the problem we typically would like to solve; that of maximizing $|p(b)|$ over binary vectors $b \in \{0,1\}^r$. Towards this, note that for a multi-affine polynomial p,

$$\max_{z \in \{0,1\}^r} |p(z)| = \max_{z \in [0,1]^r} |p(z)|.$$

Moreover, the above has a simple algorithmic proof that follows from the convexity of $x \mapsto |p(x)|$ restricted to coordinate-aligned lines. This allows us to use the above algorithm to find a point $b \in \{0,1\}^r$ whose value is at most c^r times worse than optimal given only an evaluation oracle for p. In particular, no assumptions are made on the analytic properties of p, such as concavity or real stability. In fact, in most interesting cases, such functions are highly nonconvex, hence standard convex optimization tools do not apply.

Partition matroids. As a first application of Theorem 8.1, we provide an approximation algorithm for the problem of subdeterminant maximization under partition constraints. Let $\mathcal{P} := \{M_1, M_2, \ldots, M_t\}$ be a partition of $[m] := \{1, 2, \ldots, m\}$ into non-empty, pairwise disjoint subsets and let $b = (b_1, b_2, \ldots, b_t)$ be a sequence of positive integers. Then the set $\mathcal{B} := \{S \subseteq [m] : |S \cap M_i| = b_i \text{ for all } i = 1, 2, \ldots, t\}$ is called a partition family induced by $\mathcal{P}$ and b. We first show that the problem of finding the determinant-maximizing set under partition constraints can be reformulated as

$$\max_{x \in \Delta} \det \left(W(x)^\top W(x)\right)^{1/2}$$

where Δ is a certain product of simplices, and $W(x)$ is a matrix whose i-th column is a convex combination of certain vectors derived from $L = V^\top V$ and the variables in x. Subsequently, we show that such functions are 2-anti-concentrated, which allows us to apply Theorem 8.1 to obtain the following result.

Theorem 8.2 (***Subdeterminant maximization under partition constraints***)

There exists a polynomial time randomized algorithm such that given a PSD matrix $L \in \mathbb{R}^{m \times m}$*, a partition* $\mathcal{P} = \{M_1, M_2, \ldots, M_t\}$ *of* $[m]$ *and a sequence of numbers* $b = (b_1, b_2, \ldots, b_t) \in \mathbb{N}^t$ *with* $\sum_{i=1}^t b_i = r$*, outputs a set* S *in the induced partition family* $\mathcal{B}$ *such that with high probability*

$$\det(L_{S,S}) \geqslant \mathrm{OPT} \cdot (2e)^{-2r} \cdot \prod_{i=1}^{t} \left(\frac{1}{p_i}\right)^{b_i},$$

where $\mathrm{OPT} := \max_{S \in \mathcal{B}} \det(L_{S,S})$ *and* $p_i := |M_i|$ *for* $i = 1, 2, \ldots, t$.

Regular matroids. Our second result for the constrained subdeterminant maximization problem is for the case of regular matroids (i.e., when the constraint family $\mathcal{B}$ arises as a set of bases of a regular matroid; see Section 8.3). To apply Theorem 8.1 we consider the polynomial

$$h(x) = \det(VXB^\top),$$

where X is a diagonal matrix with $X_{i,i} := x_i$, $B \in \mathbb{R}^{d \times m}$ is the linear representation of $\mathcal{B}$ and $V \in \mathbb{R}^{d \times m}$ is such that $V^\top V = L$. We observe that $|h(x)|$ is 2-anti-concentrated and has a number of desirable properties, which allows us to prove

Theorem 8.3 (***Subdeterminant maximization under regular matroid constraints***)

There exists a polynomial time randomized algorithm such that given a PSD matrix $L \in \mathbb{R}^{m \times m}$ *of rank* d*, and a totally unimodular matrix* B *that is a representation of a rank-*d *regular matroid with bases* $\mathcal{B} \subseteq 2^{[m]}$*, outputs a set* $S \in \mathcal{B}$ *such that with high probability*

$$\det(L_{S,S}) \geqslant \max(2^{-O(m)}, 2^{-O(d \log m)}) \cdot \mathrm{OPT},$$

where $\mathrm{OPT} := \max_{S \in \mathcal{B}} \det(L_{S,S})$.

8.2 Technical overview

We start by describing the approach of Nikolov and Singh [NS16] for the case of partition matroids. Consider the following simple variant of the constrained subdeterminant maximization problem for partition matroids: Given vectors $v_1, \ldots, v_r, u_1, \ldots, u_r \in \mathbb{R}^r$ the goal is to pick a vector $w_i \in \{v_i, u_i\}$ for each i so as to maximize $|\det(W)|$, where $W \in \mathbb{R}^{r \times r}$ is a matrix that has the w_is as its columns. Denote by OPT the maximum value of the determinant in the above problem.

They start by reformulating the problem as polynomial maximization problem as follows. First, define matrices $A_i(x_i) := x_i v_i v_i^\top + (1-x_i)u_i u_i^\top$ for $i = 1, 2, \ldots, r$. Then, consider the polynomial $p(x, y) := \det\left(\sum_{i=1}^r y_i A_i(x_i)\right)$ and let $g(x)$ be the polynomial that appears as the coefficient of $\prod_{i=1}^r y_i$ in $p(x, y)$.[3] Multi-linearity of g can be used to reduce the task of finding OPT to that of finding $\max_{x\in[0,1]^r} g(x)$. Then, the difficulty that arises is that $g(x)$ is hard to evaluate. To bypass this, a general idea by Gurvits [Gur06] allows them to approximate $g(x)$ by $\inf_{y>0} \frac{p(x,y)}{\prod_{i=1}^r y_i}$, giving rise to the following optimization problem involving two sets of variables

$$\max_{x\in[0,1]^r} \inf_{y>0} \frac{p(x, y)}{\prod_{i=1}^r y_i}. \tag{8.1}$$

Real stability of $p(x, y)$ for any fixed x implies that this program can be efficiently solved using convex programming. Their main result is that the value of this program is within a factor of e^r of OPT. The key component in the proof of this bound is the above-mentioned result by Gurvits that, in this context where $p(x, y)$ is real-stable with respect to y, implies that, for all $x \in [0, 1]^r$

$$g(x) \leqslant \inf_{y>0} \frac{p(x, y)}{\prod_{i=1}^r y_i} \leqslant e^r \cdot g(x). \tag{8.2}$$

While this immediately implies that one can obtain a number that is within an e^r factor of OPT, when trying to obtain an integral solution $x \in \{0, 1\}^r$ from the fractional optimal solution $x^\star \in [0, 1]^r$ to (8.1), the intractability of $g(x)$ becomes a bottleneck.[4] Nikolov and Singh present a rounding algorithm which, unfortunately, can require an exponential number of trials to find an e^r-approximate solution.

Overview of the proof of Theorem 8.2. Our approach is based on a different formulation of the problem as polynomial maximization, which has the advantage over $g(x)$ that *it is easy to evaluate and does not rely on real-stability.* For every $i = 1, 2, \ldots, r$ and $t \in [0, 1]$ define a vector $w_i(t) := (1-t)v_i + tu_i$. Furthermore, for $x \in [0, 1]^r$, let $W(x) \in \mathbb{R}^r$ be a matrix with columns $w_1(x_1), w_2(x_2), \ldots, w_r(x_r)$. The polynomial that we consider is

$$\det(W(x))$$

which is easy to evaluate for any x. As before, the multi-linearity of $\det(W(x))$ implies the following:

$$\max_{x\in[0,1]^r} |\det(W(x))| = \max_{x\in\{0,1\}^r} |\det(W(x))| = \text{OPT}. \tag{8.3}$$

Indeed, if we let $f(x) := |\det(W(x)|$, then the multi-linearity of $\det(W(x))$ implies that whenever we fix all but one of the arguments of f, i.e., $s(t) := f(t, y_2, y_3, \ldots, y_r)$ for some $y_2, y_3, \ldots, y_r \in [0, 1]$, then s attains its maximum at either 0 or 1. This means, in particular, that given any point $x \in [0, 1]^r$, one can efficiently find a point $\tilde{x} \in \{0, 1\}^r$ such that $f(\tilde{x}) \geqslant f(x)$.

However, the nonconvexity of this formulation is a serious obstacle to solving the optimization problem in Equation (8.3). This is where a key insight comes in: f shows a remarkable anti-concentration property which, in turn, allows us to get an estimate of OPT by evaluating f at a

[3] $g(x)$ is also called the mixed-discriminant of the matrices $A_i(x_i)$.

[4] One can use Equation (8.2) r times to give an approximation algorithm with factor e^{r^2}; we omit the details.

random point. Formally, the anti-concentration inequality (Theorem 8.1) applies to f and allows us to deduce that

$$\mathbf{Pr}\left[f(x) \geqslant c^{-r} \cdot \mathrm{OPT}\right] \geqslant \tfrac{1}{2}$$

for some constant $c > 1$. This also results in a simple approximation algorithm to maximize f: Sample a point $x \in [0,1]^r$ uniformly at random, round x to a vertex $\tilde{x} \in \{0,1\}^r$ such that $f(\tilde{x}) \geqslant f(x)$ as above, and output $\tilde{x}$ as a solution.

We should mention that at this point we could also attempt to invoke the following anti-concentration result (here translated to our setting) proved by Carbery and Wright.

Theorem 8.4 (*Theorem 2 in [CW01]*)

Let $p \in \mathbb{R}[x_1, x_2, \ldots, x_r]$ be a polynomial of degree r. If a point x is sampled uniformly at random from the hypercube $[0,1]^r$, then for every $\beta \in (0,1)$

$$\mathbf{Pr}\left[|p(x)| \leqslant \beta^r \cdot \mathrm{OPT}\right] \leqslant C \cdot \beta \cdot r,$$

where $C > 0$ is an absolute constant.

When applied to our setting, observe that $\det(W(x))$ is indeed a degree-r polynomial in r variables. We have to pick β so as to make $C \cdot \beta \cdot r < 1$, i.e., for $\beta = O(1/Cr)$, we obtain

$$\mathbf{Pr}\left[f(x) \geqslant r^{-O(r)} \cdot \mathrm{OPT}\right] \geqslant \tfrac{1}{2}.$$

This implies that the algorithm described above achieves an approximation ratio of (roughly) r^r. Our Theorem 8.1 is a certain strengthening of Theorem 8.4 which asserts that under the same assumptions

$$\mathbf{Pr}\left[|p(x)| \geqslant c^{-r} \cdot \mathrm{OPT}\right] \geqslant \frac{1}{2},$$

for some absolute constant $c > 1$. In fact, Theorem 8.1 is a generalization of the above for a larger class of functions (not only polynomials) and for more general domains – this is useful in the case of general partition matroids.

We now show how to extend our algorithm to a general instance of the constrained subdeterminant maximization problem under partition constraints and sketch a proof of Theorem 8.2. Recall that in this problem we are given a PSD matrix $L \in \mathbb{R}^{m \times m}$ of rank d and a partition family $\mathcal{B}$ induced by a partition of $[m]$ into disjoint sets $M_1, M_2, \ldots, M_t$ and numbers $b_1, b_2, \ldots, b_t \in \mathbb{N}$ with $\sum_{i=1}^t b_i = r$. The goal is to find a subset $S \in \mathcal{B}(\mathcal{M})$ such that $\det(L_{S,S})$ is maximized. If we consider a decomposition of L into $L = V^\top V$ for $V \in \mathbb{R}^{d \times m}$ then the objective can be rewritten as $\det(L_{S,S}) = \det(V_S^\top V_S)$. For simplicity, we assume that $b_1 = b_2 = \cdots = b_t = 1$, which can be achieved by a simple reduction. To define the relaxation for the general case, for every part M_i for $i = 1, 2, \ldots, t$, introduce a vector $x^i \in \Delta_{p_i}$ where $p_i := |M_i|$ and define a vector $w^i(x^i)$ to be

$$w^i(x^i) := \sum_{j=1}^{p_i} x^i_j v^i_j$$

where $v^i_1, v^i_2, \ldots, v^i_{p_i}$ are the columns of V corresponding to indices in M_i. We denote by x the vector $(x^1, x^2, \ldots, x^r)$ and by $W(x) \in \mathbb{R}^{d \times r}$ the matrix with columns $w^1(x^1), w^2(x^2), \ldots, w^r(x^r)$. Finally we let

$$f(x^1, x^2, \ldots, x^r) := \det(W(x)^\top W(x))^{1/2}.$$

Note that $f(x)$ is no longer a multi-linear polynomial, but as we show in Lemma 8.2 it is 2-anti-concentrated. Having established this property, Theorem 8.2 follows. Indeed, as in the illustrative example in the beginning, we can prove that given any fractional point x, we can efficiently find its integral rounding (i.e., round every component x^i to a vertex of the corresponding simplex Δ_{p_i}, for $i = 1, 2, \dots, t$) which then provides us with a suitable approximate solution.

Overview of the proof of Theorem 8.3. In the setting of Theorem 8.3 we are given a PSD matrix $L \in \mathbb{R}^{m\times m}$ of rank d and a family of bases $\mathcal{B} \subseteq 2^{[m]}$ of a regular matroid of rank d. The goal is to find a set that attains $\mathrm{OPT} := \max_{S\in\mathcal{B}} \det(L_{S,S})$. The approach of [SV17d] to obtain an estimate on OPT was inspired by that of [NS16] for the partition matroid case and is as follows:[5] Given the matrix $L = V^\top V$, first, define the following polynomial

$$g(x) := \textstyle\sum_{S\in\mathcal{B}} x^S \det(V_S^\top V_S).$$

This polynomial again turns out to be hard to evaluate. As before, an optimization problem involving two sets of variables, x and y is set up. The purpose of y variables is to give estimates of values of $g(x)$ and the x variables are constrained to be in the matroid base polytope corresponding to $\mathcal{B}$. On the one hand, real stability along with the fact that $\mathcal{B}$ is a matroid allows them to compute the optimal solution to this bivariate problem, on the other hand, with some additional effort, they are able to push Gurvits' result to obtain roughly an e^m estimate of OPT. However, the main bottleneck is that an iterative rounding approach for finding an approximate integral point does not seem possible as the matroid polytope corresponding to $\mathcal{B}$ may not have a product structure as in the partition matroid case.

We present a new formulation to capture OPT that does not suffer from the intractability of the objective function and allows for rounding via a relaxation that maximizes a certain function h over the hypercube $[0,1]^m$. Start by noting that the objective becomes $\det(L_{S,S}) = \det(V_S^\top V_S) = \det(V_S)^2$, which we can simply think of as maximizing $|\det(V_S)|$ over $S \in \mathcal{B}$. Let $B \in \mathbb{Z}^{m\times d}$ be the linear representation of the matroid $\mathcal{B}$; i.e., for every set $S \subseteq [m]$ of size d, if $S \in \mathcal{B}$ then $|\det(B_S)| = 1$, and $\det(B_S) = 0$ otherwise. Next, consider $h : [0,1]^m \to \mathbb{R}$ given by

$$h(x) := \det(VXB^\top),$$

where $X \in \mathbb{R}^{m\times m}$ is a diagonal matrix with $X_{i,i} := x_i$ for all $i = 1, 2, \dots, m$. It is not hard to see that $h(x)$ is a polynomial in x and (using the Cauchy-Binet formula) can be written as

$$h(x) = \sum_{S\subseteq[m],|S|=d} x^S \det(V_S)\det(B_S),$$

where x^S denotes $\prod_{i\in S} x_i$. Such a function was studied before in the context of matroid intersection problems [Lov89, Har09, GT17]. Importantly, the restriction of $h(x)$ to indicator vectors of sets of size d is particularly easy to understand. Indeed, let 1_S be the indicator vector of some set $S \subseteq [m]$ with $|S| = d$. We have

$$h(1_S) = \det(V_S)\det(B_S) = \begin{cases} \pm\det(V_S) & \text{if } S \in \mathcal{B}, \\ 0 & \text{if } S \notin \mathcal{B}. \end{cases}$$

[5]The approach of [AO17] is also similar.

Hence, we are interested in the largest magnitude coefficient of a multi-linear polynomial $h(x)$. The maximum of $|h(x)|$ over $[0,1]^m$ is an upper bound for this quantity. The algorithm then simply selects a point $x \in [0,1]^m$ at random, which by Theorem 8.1 can be related to the maximum value of $|h(x)|$, and then performs a rounding.

First, given $x \in [0,1]^m$ it constructs a binary vector $\tilde{x} \in \{0,1\}^m$ such that $|h(\tilde{x})| \geqslant |h(x)|$; this is possible because the function $|h(x)|$ is convex along any coordinate direction. The vector $\tilde{x}$ is then treated as a set $S_0 \subseteq [m]$, but its cardinality is typically larger than d. We then run another procedure which repeatedly removes elements from S_0 while not loosing too much in terms of the objective. It is based on using $h(1_{S_0})$ as a certain proxy for the sum $\sum_{S\subseteq S_0} |\det(V_S)\det(B_S)|$. This allows us to finally arrive at a set $S \subseteq S_0$ of cardinality d, such that $|h(1_S)| \geqslant \binom{m}{d}^{-1} |h(1_{S_0})|$. The set S is then the final output.

By applying Theorem 8.1 one can conclude that $h(1_{S_0})$ is within a factor of c^m of the maximal value of $|h(x)|$, which results in a $2^{O(m)}$-approximation guarantee for the algorithm. Alternatively, by utilizing the fact that h is a polynomial of degree d, one can apply the result by Carbery-Wright (see Theorem 8.4) to obtain a bound of roughly $m^{O(d)}$, which is better whenever m is large compared to d.

Overview of the proof of Theorem 8.1. For the sake of clarity, we present only the hypercube case of the anti-concentration inequality, which corresponds to taking $p_1 = p_2 = \cdots = p_r = 2$ in the statement of Theorem 8.1. Recall the setting: We are given a function $f : [0,1]^r \to \mathbb{R}_{\geqslant 0}$ that satisfies a one-dimensional anti-concentration inequality. I.e., for every function of the form $g(t) := f(x_1, x_2, \ldots, x_{i-1}, t, x_{i+1}, \ldots, x_r)$ where $x_j \in [0,1]$ for $j \neq i$ are fixed and $t \in [0,1]$, it holds that

$$\mathbf{Pr}\left[g(t) < c \cdot \max_{t\in[0,1]} g(t)\right] \leqslant 2c, \tag{8.4}$$

where the probability is over a random choice of $t \in [0,1]$. The goal is to prove a similar statement for $f(x)$, i.e., $\mathbf{Pr}[f(x) < \alpha \cdot \mathrm{OPT}]$ is small, where OPT is the maximum value f takes on the hypercube and α is a parameter which we want to be as large as possible.

As an initial approach, one can define (for some $c > 0$) events of the form

$$A_i := \left\{x \in [0,1]^r : f(x_1, x_2, \ldots, x_i, x^\star_{i+1}, \ldots, x^\star_r) \geqslant c \cdot f(x_1, x_2, \ldots, x_{i-1}, x^\star_i, \ldots, x^\star_r)\right\},$$

where $x^\star := \arg\max_x f(x)$. Note crucially that the events $A_1, A_2, \ldots, A_r$ are not independent. However, we can still write

$$\begin{aligned}\mathbf{Pr}\left[f(x) \geqslant c^r \cdot \mathrm{OPT}\right] &\geqslant \mathbf{Pr}\left[A_1 \cap A_2 \cap A_3 \cdots \cap A_r\right]\\ &= \mathbf{Pr}[A_1] \cdot \mathbf{Pr}[A_2|A_1] \cdot \mathbf{Pr}[A_3|A_1, A_2] \cdots \mathbf{Pr}[A_r|A_1, A_2, \ldots, A_{r-1}].\end{aligned}$$

From assumption (8.4) we know that

$$\mathbf{Pr}[A_i|A_1, A_2, \ldots, A_{i-1}] \geqslant 1 - 2c$$

for all $i = 1, 2, \ldots, r$ and hence

$$\mathbf{Pr}\left[f(x) \geqslant c^r \cdot \mathrm{OPT}\right] \geqslant (1-2c)^r.$$

To get a probability that is not exponentially small, one has to take c of size roughly $O(1/r)$, in which case we recover the result by Carbery and Wright [CW01] in our setting. To go beyond this, a tighter analysis is required. For this we consider the random variables

$$Z_i := \frac{f(x_1, x_2, \ldots, x_i, x_{i+1}^\star, \ldots, x_r^\star)}{f(x_1, x_2, \ldots, x_{i-1}, x_i^\star, \ldots, x_r^\star)}$$

for $i = 1, 2, \ldots, r$. Note that $\prod_{i=1}^r Z_i = \frac{f(x)}{\mathrm{OPT}}$ hence the goal reduces to proving that

$$\mathbf{Pr}\left[\prod_{i=1}^r Z_i \geqslant c^{-r}\right] \geqslant \frac{1}{2}.$$

with a decent probability. To obtain such a bound, we first translate the product to a sum and define $X_i := -\log Z_i$ which then reduces our task to proving

$$\mathbf{Pr}\left[\sum_{i=1}^r X_i \leqslant O(r)\right] \geqslant \frac{1}{2}. \tag{8.5}$$

the anti-concentration assumption on f translates to the following convenient bound on the CDF of X_i

$$\mathbf{Pr}[X_i \geqslant t | X_1, X_2, \ldots, X_{i-1}] \leqslant \min(1, 2e^{-t}) \qquad \forall_{t \in \mathbb{R}}.$$

Hewever, as previously, the fact that X_i's are not independet presents itself as a significant issue. To overcame it, we prove (in Lemma 8.1) that for the purpose fo proving the bound (8.5), the variables X_i can be replaced by independent copies Y_i of a random variable with CDF $t \mapsto \min(1, 2e^{-t})$. The analysis of the tail bound for independent variables follows then from standard tools, such as Chebyshev's inequality.

8.3 Proofs

Preliminaries

Simplices and Measures The d-dimensional Lebesgue measure (volume) on $\mathbb{R}^d$ is denoted by λ_d. When the dimension is clear from the context, we use λ to denote the volume. Throughout this chapter, the probability distributions we consider, are typically uniform over an appropriate domain.

The standard $(d-1)$-simplex, denoted by Δ_d is defined as the convex hull of $e_1, e_2, \ldots, e_d \in \mathbb{R}^d$. Notice that Δ_d is a $(d-1)$-dimensional polytope which is embedded in $\mathbb{R}^d$, and it inherits a $(d-1)$-dimensional Lebesgue measure from the hyperplane it lies on. We use μ_d to denote the induced measure λ_d on the simplex Δ_d, normalized so that $\mu_d(\Delta_d) = 1$. We often deal with Cartesian products of simplices, which we denote by $\Delta = \prod_{i=1}^r \Delta_{p_i}$, for some sequence $p_1, p_2, \ldots, p_r \in \mathbb{N}$. For a point $x \in \Delta$, by x^i we denote i-th component of x belonging to Δ_{p_i} and x_j^i for $j \in [p_i]$ are the components of x^i within Δ_{p_i}. By $V(\Delta)$, we denote the set of points of Δ with integer coordinates. We call $V(\Delta)$ the set of vertices of Δ.

Multi-linear functions. A function $f : \mathbb{R}^m \to \mathbb{R}$ is called multi-linear if f is a polynomial function where the degree of each variable is at most 1. Suppose that $x_1, \ldots, x_m$ are m variables. We denote the monomial $\prod_{i \in S} x_i$ by x^S for every $S \subseteq [m]$. Every multi-linear function can be written in the form $f(x) = \sum_{S \subseteq [m]} f_S x^S$ where f_S's are real numbers, called the coefficients of

f. A function $f : \mathbb{R}^m \to \mathbb{R}$ is called affine when f is a polynomial whose total degree is at most one. A function $f : \mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_r} \to \mathbb{R}$ is called block-multi-linear if for every index $i \in [r]$ and for every choice of $y^j \in \mathbb{R}^{p_j}, j \in [r] \setminus \{i\}$ the function $f(y^1, \ldots, x^i, \ldots, y^r)$ is an affine function over $\mathbb{R}^{p_i}$.

8.3.1 Anti-Concentration Inequality

Lemma 8.1

Let $Y_1, Y_2, \ldots, Y_r$ be real random variables with CDFs $f_1, f_2, \ldots, f_r : \mathbb{R} \to [0, 1]$ respectively, i.e., $f_i(x) := \mathbf{Pr}[Y_i \leqslant x]$ for $i \in [r]$ and $x \in \mathbb{R}$. Suppose $X_1, X_2, \ldots, X_r$ are real random variables such that

$$\mathbf{Pr}[X_i \leqslant x | X_1, X_2, \ldots, X_{i-1}] \geqslant f_i(x) \qquad \text{for every } i = 1, 2, \ldots, r \text{ and } x \in \mathbb{R}$$

then for every function $G : \mathbb{R}^r \to \mathbb{R}_{\geqslant 0}$ which is monotone with respect to every coordinate it holds

$$\mathbb{E}\left[G(X_1, X_2, \ldots, X_r)\right] \leqslant \mathbb{E}\left[G(Y_1, Y_2, \ldots, Y_r)\right].$$

Proof.
We will prove the claim by induction on r. Consider the case of $r = 1$ first. Let g_1 be the CDF of X_1, we have

$$\mathbb{E}\left[G(X_1)\right] = \int G(x_1) dg_1(x_1).$$

where the above is a Riemann-Stieltjes integral with respect to g_1. Since G is monotone and $g_1 \geqslant f_1$, it is an elementary fact on R-S integrals that

$$\int G(x_1) dg_1(x_1) \leqslant G(x_1) df_1(x_1) = \mathbb{E}\left[G(Y_1)\right]$$

and hence the claim for $r = 1$.

Suppose now that the claim holds for $(r-1) \in \mathbb{N}$, we will prove it for r. Denote

$$H(X_1, X_2, \ldots, X_{r-1}) := \mathbb{E}\left[G(X_1, \ldots, X_r) | X_1, \ldots, X_{r-1}\right]$$
$$K(Y_1, Y_2, \ldots, Y_{r-1}) := \mathbb{E}\left[G(Y_1, \ldots, Y_r) | Y_1, \ldots, Y_{r-1}\right]$$

(the conditional expectations). From the assumption and the $r = 1$ case we have that for every tuple $(x_1, \ldots, x_{r-1}) \in \mathbb{R}^{r-1}$ we have

$$H(x_1, \ldots, x_{r-1}) \leqslant K(x_1, \ldots, x_{r-1}).$$

Further:

$$\begin{aligned}\mathbb{E}\left[G(X_1,\ldots,X_r)\right] &= \mathbb{E}\left[H(X_1,X_2,\ldots,X_{r-1})\right]\\ &\leqslant \mathbb{E}\left[H(Y_1,Y_2,\ldots,Y_{r-1})\right]\\ &\leqslant \mathbb{E}\left[K(Y_1,Y_2,\ldots,Y_{r-1})\right]\\ &= \mathbb{E}\left[G(Y_1,\ldots,Y_r)\right]\end{aligned}$$

where the transition from the first to the second line follows from the induction hypothesis. □

Proof of Theorem 8.1:
Let us fix any optimal point $x^\star := (x_1^\star,\ldots,x_r^\star)$ and consider random variables

$$Z_i := p_i \frac{f(x_1,x_2,\ldots,x_i,x_{i+1}^\star,\ldots,x_r^\star)}{f(x_1,x_2,\ldots,x_{i-1},x_i^\star,\ldots,x_r^\star)}$$

under a uniformly random choice of $x \in \prod_{i=1}^r \Delta_{p_i}$. It is not hard to see that

$$\frac{f(x_1,\ldots,x_r)}{\mathrm{OPT}} \cdot \prod_{i=1}^r p_i = \prod_{i=1}^r Z_i.$$

Thus our goal is to prove that

$$\mathbf{Pr}\left[\prod_{i=1}^r Z_i \leqslant (\gamma e^2)^{-r}\right] \leqslant \frac{1}{r}.$$

From the definition of anti-concentration, for every $i \in [r]$ and every $c \in (0,1)$ we have

$$\mathbf{Pr}[Z_i \leqslant c | Z_1,\ldots,Z_{i-1}] \leqslant \gamma c. \tag{8.6}$$

As usually it is more convenient to analyze sums rather than products, hence let us define

$$X_i := -\log Z_i.$$

We would like to prove a tail bound on the probability $\mathbf{Pr}\left[\sum_{i=1}^r X_i \geqslant \Omega(r)\right]$. From (8.6) we obtain that for every $x \in \mathbb{R}_{\geqslant 0}$ we have

$$\mathbf{Pr}[X_i \geqslant x] \leqslant \gamma e^{-x}.$$

Let us now define $Y_1,\ldots,Y_r \in \mathbb{R}_{\geqslant 0}$ to be independent random variables such that for every $i \in [r]$ and $x \in \mathbb{R}_{\geqslant 0}$

$$\mathbf{Pr}[Y_i \geqslant x] = \min\left(1, \gamma e^{-x}\right). \tag{8.7}$$

We claim that Lemma 8.1 implies that for every $x \in \mathbb{R}_{\geqslant 0}$ it holds that

$$\mathbf{Pr}\left[\sum_{i=1}^r X_i \geqslant x\right] \leqslant \mathbf{Pr}\left[\sum_{i=1}^r Y_i \geqslant x\right].$$

Indeed, to arrive at such a conclusion one can consider the function

$$G(t_1, \ldots, t_r) := \left[\sum_{i=1}^{r} t_i \geqslant x\right]$$

where $[\phi]$ is the Iverson bracket, i.e., it is 1 when ϕ holds and 0 otherwise. The function G is clearly monotone and

$$\mathbb{E}\left[G(X_1, X_2, \ldots, X_r)\right] = \mathbf{Pr}\left[\sum_{i=1}^{r} X_i \geqslant x\right].$$

It is now enough to derive a tail bound on $\mathbf{Pr}\left[\sum_{i=1}^{r} Y_i \geqslant x\right]$ for independent variables $Y_1, \ldots, Y_r$ distributed as in (8.7). To this end we simply apply the Chebyshev's inequality. Let us now compute the expectation and variance of a variable Y with distribution as the Y_i's.

Let us denote $g(x) := \gamma e^{-x}$. We have

$$\mathbb{E}\left[Y\right] = \int_{\log \gamma}^{\infty} y(-g'(y))dy = 1 + \log(\gamma).$$

Similarly

$$\mathbb{E}\left[(\mathbb{E}\left[Y\right] - Y)^2\right] = \int_{\log \gamma}^{\infty} (y - 1 - \log(\gamma))^2(-g'(y))dy = 1.$$

Now from Chebyshev's inequality we obtain that for any $M > 0$

$$\mathbf{Pr}\left[\sum_{i=1} Y_i \geqslant \mathbb{E}\left[\sum_{i=1} Y_i\right] + M\right] \leqslant \frac{\mathrm{Var}\left[\sum_{i=1} Y_i\right]}{M^2},$$

and hence

$$\mathbf{Pr}\left[\sum_{i=1} Y_i \geqslant r(1 + \log\gamma) + M\right] \leqslant \frac{r}{M^2}.$$

Thus by taking $M := r$ we obtain

$$\mathbf{Pr}\left[\sum_{i=1} Y_i \geqslant r(2 + \log\gamma)\right] \leqslant \frac{1}{r}.$$

Finally, translating this bound to X_i's and then to Z_i's we conclude

$$\mathbf{Pr}\left[\prod_{i=1}^{r} Z_i \leqslant \left(\gamma e^2\right)^{-r}\right] \leqslant \frac{1}{r},$$

which concludes the proof. □

8.3.2 Partition Matroids

Lemma on Anti-concentration

Lemma 8.2

Let $w_1, w_2, \ldots, w_p \in \mathbb{R}^d$ be any vectors. Then the function $f : \Delta_p \to \mathbb{R}$ defined as $f(x) = \|\sum_{j=1}^p x_j w_j\|$ is 2-anticoncentrated.

Proof.
We begin by establishing the fact for $p = 2$. To this end define $g(x) := |x_1\|w_1\| - x_2\|w_2\||$, we claim that

$$\forall_{x\in\Delta_2} \quad g(x) \leqslant f(x).$$

The above claim follows simply from triangle inequality. Indeed

$$\|x_1w_1\| = \|x_1w_1 + x_2w_2 - x_2w_2\| \leqslant \|x_1w_1 + x_2w_2\| + \|x_2w_2\|$$

and hence

$$\|x_1w_1\| - \|x_2w_2\| \leqslant \|x_1w_1 + x_2w_2\|.$$

By symmetry $\|x_2w_2\| - \|x_1w_1\| \leqslant \|x_1w_1 + x_2w_2\|$ follows as well. Given the claim and observing that $\max_{x\in\Delta_2} f(x) = \max_{x\in\Delta_2} g(x)$, it is enough to prove 2-anti-concentration of g, since then an anologous result for f follows. This is in fact the subject of Fact 8.1, hence the $p = 2$ case follows.

The case of $p \geqslant 3$ is proved differently, by taking advantage of the $p = 2$ case. The challange to prove it comes from the fact that generating a random point from a high-dimensional simplex Δ_p is not equivalent to simply generating its coordinates independently and uniformly at random and then normalizing the obtained point so that it sums up to one. There are several known methods for sampling a random point from Δ_p, however no "practical" method seems to be well suited for this proof and below we simply use the basic definition to deal with it.

Consider any isometric embedding P of Δ_p in $\mathbb{R}^{p-1}$ where it is a full-dimensional polytope. Then consider the uniform distribution over any box contaning P. Conditioned on the sample landing in P, the corresponding distribution is – by definition – uniform on P and thus (via the embedding) uniform on Δ_p.

Denote the vertices of Δ_p in the embedding to be $v_1, v_2, \ldots, v_p \in \mathbb{R}^{p-1}$. Let also $\widetilde{g} : P \to \mathbb{R}$ be the corresponding function g on P, i.e.,

$$\forall_{x\in\Delta_p} \quad \widetilde{g}\left(\sum_{j=1}^p x_j v_j\right) = g(x).$$

Assume without loss of generality that $\|w_1\|$ is the largest among $\|w_1\|, \|w_2\|, \ldots, \|w_p\|$ and that $v_1 = 0$. Now, consider any point $v \in P$ on the facet opposite to v_1, i.e. $v = \sum_{j=2}^p y_j v_j$ where $(y_2, y_3, \ldots, y_p) \in \Delta_{p-1}$. For $z \in [0, 1]$ consider

$$h(z) = \widetilde{g}(zv_1 + (1 - z)v) = \|zw_1 + (1 - z)\sum_{j=2}^p y_j w_j\|.$$

From the $p = 2$ case h is 1-anti-concentrated and moreover $\max_{z\in[0,1]} h(z) = \max_{x\in\Delta_p} f(z) = \|w_1\|$. Thus for every ray $[v_1, v]$ ($v \in \mathrm{conv}\{v_2, \ldots, v_p\}$) we have an anti-concentrated function on it, whose maximum coincides with the maximum of f, and the simplex P is a disjoint union of such rays. Seemingly, this already implies anti-concentration of f, however note that the distribution on the ray $[v_1, v]$ induced from the uniform distribution over P is not uniform and hence the result does not follow yet.

More formally, let us denote the distribution on $[0,1]$ which is induced from the uniform distribution on P when restricted to $[v_1, v] \equiv [0,1]$ by μ_v we would like to prove:

$$\mathbf{Pr}_{z\sim\mu_v}[h(z) < c \cdot \|w_1\|] \leqslant 2pc$$

but what we know is only that when z is uniformly distributed over $[0,1]$:

$$\mathbf{Pr}_{z\sim[0,1]}[h(z) < c \cdot \|w_1\|] \leqslant 2c.$$

Thus it remains to understand μ_v. The density of μ_v can be derived from the hyperspherical coordinate system and its Jacobian. In fact it follows that for any fixed ray $[0, v]$ the density μ_v on $[0,1]$ at a point z is proportional to z^{p-2}. Thus the task of proving anti-concentration finally reduces to the following inequality. Given a set $A \subseteq [0,1]$ of Lebesgue measure at most $2c$, show that

$$\int_A \frac{z^{p-2}}{p-1} dz \leqslant 2pc.$$

Because of monotonicity this is equivalent to proving (note that we may assume that $2c < 1$ here)

$$\int_{1-2c}^{1} \frac{z^{p-2}}{p-1} dz \leqslant 2pc,$$

which further reduces to

$$1 - (1-2c)^{p-1} \leqslant 2pc$$

the above holds by Bernoulli's inequality. $\square$

Fact 8.1

Let $a_1, a_2 \in \mathbb{R}$ be any numbers. Consider the function $f : \Delta_2 \to \mathbb{R}$ given by $f(x) = |a_1x_1 + a_2x_2|$. Then f is 1-anti-concentrated.

Proof.
Let us translate the question to a 1−dimensional problem first. Let $a = \max(|a_1|, |a_2|) = \max_{x\in\Delta_2} f(x)$ and define $g : [0,1] \to \mathbb{R}$ by $g(t) = |(1-t)a_1 + ta_2|$. We would like to prove that when t is sampled uniformly at random from $[0,1]$ then for every $c \in (0,1)$ we have

$$\mathbf{Pr}[g(t) < c \cdot a] \leqslant 2c.$$

Assume without loss of generality that $g(0) = a \geqslant g(1)$ and that g is not a constant function. There are two cases: either g has a single root in $[0,1]$ or it has no roots. We analyze the former, as the latter the also follows.

Let $t_0 \in (0,1]$ be the root of $g(t)$. It is not hard to see that $t_0 \geqslant \frac{1}{2}$, as $g(t - t_0)$ is a symmetric function. Now, the function g on $[0, t_0]$ is linear and hence

$$\mathbf{Pr}_{t\in[0,t_0]}[g(t) < c \cdot a] \leqslant c,$$

and consequently

$$\mathbf{Pr}\left[(g(t) < c \cdot a) \wedge t \in [0, t_0]\right] \leqslant c \cdot t_0.$$

By symmetry and the fact that $g(1) \leqslant g(0)$ we have

$$\mathbf{Pr}\left[(g(t) < c \cdot a) \wedge t \in [t_0, 1]\right] \leqslant \mathbf{Pr}\left[(g(t) < c \cdot a) \wedge t \in [0, t_0]\right],$$

and hence

$$\mathbf{Pr}\left[g(t) < c \cdot a\right] \leqslant 2 \cdot c \cdot t_0 \leqslant 2 \cdot c.$$

□

Proof of Theorem 8.2

Proof of Theorem 8.2:
We start by observing that it suffices to prove the Theorem for the case when $b_1 = b_2 = \cdots = b_t = 1$. Indeed, when b_i's are not all equal to 1, we can perform a simple reduction to the all-ones case. Namely, we construct a new instance of the problem, where every part M_i is repeated b_i times. After doing so, we obtain a new instance with r parts $M_1', M_2', \ldots, M_r'$ and $b_1' = b_2' = \ldots = b_r' = 1$.

Every feasible solution to the original instance corresponds to a feasible solution to the new instance (with the same value). Conversely, every feasible solution *with non-zero value* corresponds to a feasible solution in the original instance.

Finally, the bound on the approximation ratio follows easily by translating the bound in the simple case $b_1 = b_2 = \ldots = b_r = 1$ to the instance after reduction.

From now on we assume that $b_1 = b_2 = \cdots = b_t = 1$; in this case $t = r$. Let

$$L = V^\top V$$

be the Cholesky decomposition of the PSD matrix L with $V \in \mathbb{R}^{d \times m}$. One can easily see that

$$L_{S,S} = V_S^\top V_S, \quad \text{for all } S \subseteq [m].$$

For every part M_i $(i = 1, 2, \ldots, t)$ consider the p_i-simplex Δ_{M_i} indexed by the elements in M_i, i.e.

$$\Delta_{M_i} = \left\{ y \in [0,1]^{M_i} : \sum_{j \in M_i} y_j = 1 \right\}.$$

Further, consider $\Delta := \prod_{i=1}^t \Delta_{M_i}$ and a function $f : \Delta \to \mathbb{R}$ defined as follows

$$f(x) := \det\left[V(x)^\top V(x)\right]^{1/2}$$

where $V(x) \in \mathbb{R}^{d \times t}$ matrix, whose ith column is $V_i(x) := \sum_{j \in M_i} x_j v_j$. Note that when $x \in \Delta$ is a $0-1$ vector, i.e., $x = 1_S$ for some set $S \in \mathcal{B}$ then $f(x)^2 = \det(V_S^\top V_S)$. Thus, there exists a natural bijection between the elements of $\mathcal{B}$ (bases of the partition matroid) and the vertices of $\Delta = \prod_{i=1}^t \Delta_{p_i}$. Therefore, the optimization problem can be stated as the problem of maximizing f over the vertices of Δ. That is

$$\max\left\{f(x) : x \in \Delta \cap \{0,1\}^m\right\}.$$

We prove that maximizing f over integer points in Δ is the same as maximizing it over the whole polytope Δ. This, composed with an algorithm to round a fractional point to a vertex and an anti-concentration result on f will allow us to conclude Theorem 8.2. We start with the former. Let us fix all but the first block-coordinates of $x \in \Delta$, i.e. $x = (y, x')$, where $y \in \Delta_{M_1}$ and $x' \in \prod_{i=2}^{t} \Delta_{M_i}$ is fixed. Further, denote by $V'(x)$ the submatrix of $V(x)$ composed of columns $V_2(x), \dots, V_t(x)$. By the formula on the determinant of a block matrix, we have

$$\det\left[V(x)^\top V(x)\right] = \det\left[V'(x)^\top V'(x)\right] \cdot \left(V_1(x)^\top \cdot \Pi \cdot V_1(x)\right)$$

where $\Pi \in \mathbb{R}^{(t-1)\times(t-1)}$ is a certain projection matrix. Thus in particular, there exist vectors $\{w_j\}_{s\in M_1}$ such that

$$f(x) = \|\sum_{j\in M_1} y_j w_j\| \cdot \det\left[V'(x)^\top V'(x)\right]^{1/2}. \tag{8.8}$$

Note that the above, as a function of $y \in \Delta_{M_1}$ is maximized at some vertex $y \in \Delta_{M_1} \cap \{0,1\}^{M_1}$. And thus (by induction), the whole function $f(x)$ is maximized at an integer vector. This observation also implies a simple rounding algorithm: given any fractional point $x \in \Delta$, go coordinate by coordinate $i = 1, 2, \dots, t$ and round it to a vertex which provides the largest value of f, this requires to evaluate f at p_i points only.

Thus so far we have proved that an (approximation) algorithm for finding a fractional maximizer of f over Δ can be turned into an algorithm maximizing $\det(V_S^\top V_S)$ over $S \in \mathcal{B}$ with the same guarantee and polynomial overhead in the running time.

We prove that f is 2-anticoncentrated which implies that a value of f at a random point gives, with high probability, a decent estimate of the optimal value. In fact, 2-anticoncentration, together with Theorem 8.1 and the observation above implies Theorem 8.2 immediately.

To prove anticoncentration, we need to analyze how does f behave when all but one of its coordinates are fixed. Without loss of generality fix all but the first coordinate. Note that by (8.8) our goal becomes to prove that the function $\Delta_{M_1} \ni y \mapsto \|\sum_{j\in M_1} y_j w_j\|$ is 2-anticoncentrated. However this exactly what we prove in Lemma 8.2. $\square$

8.3.3 Regular Matroids

We start by reducing the subdeterminant maximization problem under a regular matroid constraint to a polynomial optimization problem as follows. Let $B_1, B_2, \dots, B_m \in \mathbb{R}^d$ be the columns of B. Since B is a representation of the matroid $\mathcal{M}$, a set $S \subseteq M$ is a basis of $\mathcal{M}$ if and only the set of the vectors $\{B_i : i \in S\}$ is linearly independent. Let $L = V^\top V$ be a Cholesky decomposition of the PSD matrix L, for $V \in \mathbb{R}^{d\times m}$.

Let us now consider any set $S \in \binom{[m]}{d}$ and define $I_S := \text{Diag}\left(()\, 1_S\right)$. For any $S \in \binom{[m]}{d}$ we have

$$\det\left(V I_S B^\top\right) = \det\left(\sum_{i\in S} V_i B_i^\top\right) = \det\left(V_S\right)\det\left(B_S^\top\right).$$

Since B is a totally unimodular matrix, $|\det(B_S)| = 1$ if $S \in \mathcal{B}(\mathcal{M})$ and 0 otherwise. Thus for all $S \in \binom{[m]}{d}$

$$\left|\det\left(V I_S B^\top\right)\right| = \begin{cases} |\det(V_S)| & \text{if } S \in \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases}$$

Since for all $S \in \binom{[m]}{d}$, $\det(L_{S,S}) = \det(V_S^\top V_S) = \det(V_S)^2$, maximizing $\det(L_{S,S})$ over $S \in \mathcal{B}$ is equivalent to maximizing $|f(x)|$ for $f(x) := \det(VXB^\top)$ over all the 0-1 vectors $x \in \{0,1\}^m$ subject to $\sum_{i=1}^m x_i = d$. We give an approximation algorithm for this problem which proceeds in two phases.

Phase 1: Finding a Fractional Solution.

In the first phase, we drop the $\sum_{i=1}^m x_i = d$ condition and relax the $0-1$ condition to $x \in [0,1]^m$. Our optimization problem then becomes

$$\begin{aligned} \max_x \quad & |f(x)|, \\ \text{s.t.} \quad & x \in [0,1]^m. \end{aligned} \tag{8.9}$$

Our algorithm to find an approximate solution to (8.9) is as follows. We sample a polynomial number of points x from $[0,1]^m$ uniformly and independently at random. Then, we output the point with the largest value of $|f(x)|$. We analyze the performance of this algorithm in two different regimes.

Large d. It follows from the Cauchy-Binet formula that

$$f(x) = \sum_{S \in \mathcal{B}} x^S \det(V_S) \det(B_S). \tag{8.10}$$

Moreover, $f(x)$ is multi-affine and easy to compute (because it is just a determinant of an $m \times m$ matrix). We show that $|f|$ is 2-anti-concentrated. To this end, we show that for every $i \in [m]$ and every choice of $y_j \in [0,1], j \in [m] \setminus \{i\}$, the univariate function

$$\tau \mapsto |f\left(y_1, \ldots, y_{i-1}, \tau, y_{i+1}, \ldots, y_m\right)|$$

is 2-anti-concentrated. Such a function is of the form $\tau \mapsto |a\tau + b|$ for some $a, b \in \mathbb{R}$. 2-anti-concentration of such functions follows easily from Lemma 8.2. Indeed, by setting $d = 1$ and $p = 2$ in Lemma 8.2 we obtain the 2-anti-concentration of $(\tau_1, \tau_2) \mapsto |\tau_1 a_1 + \tau_2 a_2|$, which implies our claim.

Theorem 8.1 implies now that if we sample a uniform point x from $[0,1]^m$ then

$$\mathbf{Pr}\left[|f(x)| > 2^{-m}(2e^2)^{-m} \cdot \mathrm{OPT}\right] \geqslant 1/2.$$

Where $\mathrm{OPT} := \max_{x \in [0,1]^m} |f(x)|$ is clearly an upper bound on $\max_{S \in \mathcal{B}} |\det(V_S)|$. We can amplify the probability of success by repeating the experiment several times and hence, with high probability obtain a point $\hat{x}$ such that

$$|f(\hat{x})| > (2e)^{-2m} \cdot \mathrm{OPT}.$$

Small d. From (8.10) it is clear that the function f is a polynomial of degree d in m variables. According to Theorem 2 in [CW01], if we sample x uniformly from the unit hypercube $[0,1]^m$, then

$$\mathbf{Pr}\left[|f(x)| \leqslant \beta^d \cdot \mathrm{OPT}\right] \leqslant C \cdot \beta \cdot m,$$

for any $\beta > 0$ and some absolute constant $C > 0$. By picking $\beta = \frac{1}{2C \cdot m}$, we conclude that with constant probability we obtain a vector $\hat{x}$ such that

$$|f(\hat{x})| > \left(\frac{1}{2mC}\right)^d \cdot \mathrm{OPT}.$$

Phase 2: Rounding the Fractional Solution.

We first round $\hat{x}$ obtained in the previous phase to a $0-1$ vector, and then finally to a set $\hat{S} \in \binom{[m]}{d}$. Since f is multi-affine, the restriction of f to the first coordinate is a 1-dimensional affine function. Therefore, either

$$|f(0, \hat{x}_2, \ldots, \hat{x}_d)| \geqslant |f(\hat{x})| \quad \text{or} \quad |f(1, \hat{x}_2, \ldots, \hat{x}_d)| \geqslant |f(\hat{x})|.$$

Hence, we can round the first coordinate without decreasing the value of $|f(\hat{x})|$, using one call to the evaluation oracle. We proceed to the next coordinates and round them one at a time. Let $y \in \{0,1\}^m$ be the outcome of the above rounding algorithm.

Let $S_0 \subseteq [m]$ such that $1_{S_0} = y$. It is likely that $|S_0| > d$, hence we will need to remove several elements from S_0 to obtain a set of cardinality d. Define a function $g : 2^{[m]} \to \mathbb{R}$ to be

$$g(S) := f(1_S) = \det(V_S B_S^\top).$$

Note in particular that g can be computed efficiently. Furthermore, by the Cauchy-Binet formula, we have

$$g(S) = \sum_{T \in \binom{[m]}{d}} g(T) = \sum_{T \in \binom{[m]}{d}} \det(V_T)\det(B_T) \tag{8.11}$$

for every subset $S \in 2^{[m]}$. We have $|f(y)| = |f(1_{S_0})| = |g(S_0)|$. Further, (8.11) implies that

$$\sum_{i \in S_0} g(S_0 \setminus \{i\}) = (|S_0| - d) \sum_{T \in \binom{S_0}{d}} g(T) = (|S_0| - d) g(S_0).$$

Consequently, there exists an $i \in S_0$ such that:

$$|g(S_0 \setminus \{i\})| \geqslant \frac{|S_0| - d}{|S_0|} |g(S_0)|.$$

In our algorithm we find such an i and consider $S_1 := S_0 \setminus \{i\}$. This step of removing one element is repeated until we arrive at a set $\hat{S} \subseteq [m]$ of cardinality d. In this process we can guarantee that

$$|g(\hat{S})| \geqslant |g(S_0)| \cdot \prod_{j=1}^{|S_0|-d} \frac{j}{j+d} \geqslant \frac{|g(S_0)|}{\binom{m}{d}}.$$

Finally, since $|g(\hat{S})| = |\det(V_{\hat{S}})|$, we conclude:

$$|\det(V_{\hat{S}})| \geqslant \frac{|f(y)|}{\binom{m}{d}} > \frac{1}{\binom{m}{d}} \max\left((2e)^{-2m}, (2dC)^{-d}\right) \cdot \mathrm{OPT}$$

hence $|\det(V_{\hat{S}})| > \max\left(2^{-O(m)}, 2^{-O(d \log m)}\right) \cdot \mathrm{OPT}$, and Theorem 8.3 follows.

8.4 Conclusion

8.4.1 Future Work

The anti-concentration inequality in Theorem 8.1 applies to simplices and cartesian products of simplices. It is an interesting question to extend it beyond this setting – it might lead to better approximation guarantee for the case of regular matroids.

Another open question we would like to state here is the one about derandomization of the algorithms in Theorems 8.3 and 8.2. In a slightly more abstract form this task can be stated as a search for an $e^{O(n)}$-approximation algorithm for the following problem. Given a multiaffine polynomial $f \in \mathbb{R}[x_1, x_2, \ldots, x_n]$ find $\max_{x \in [0,1]^n} |f(x)|$. Our anti-concentration theorem implies that a random point gives a suitable approximation with high probability, can we find such a point deterministically?

8.4.2 Notes

The content of this chapter is based on results obtained in [ESV17]. Approximation algorithms for the constrained subdeterminant maximization problem are rather rare; Khachiyan [Kha95] proposed the first polynomial time approximation algorithm for the problem when $\mathcal{B} = \binom{[m]}{r}$ which achieved an approximation factor of $r^{O(r)}$. This result was improved by Nikolov [Nik15] who presented an approximation algorithm which achieved a factor of e^r. On the other hand, it was shown [DSEFM15, ÇM09] that there exists a constant $c > 1$ such that approximating the $\mathcal{B} = \binom{[m]}{r}$ case with approximation ratio better than c^r remains NP-hard.

Among estimation algorithms, recently, Nikolov and Singh [NS16] generalized Nikolov's result to the setting when the family $\mathcal{B}$ corresponds to the bases of a *partition matroid.* They presented an elegant convex program that allowed them to efficiently estimate the value of the maximum determinant set from $\mathcal{B}$ to within a factor of e^r where r is the size of the largest set in the partition matroid $\mathcal{B}$. One of the main ingredients in their proof is an inequality due to Gurvits [Gur06] concerning real stable polynomials.

A very general anti-concentration result for polynomial functions over convex domains was obtained by Carbery and Wright [CW01]. The result by Carbery and Wright and, more generally, the anti-concentration phenomena has found several applications in theoretical computer science, especially for Gaussian measures; see for instance [O'D14, DDS16, CTV06, RV13]. Finally, our use of rounding using multi-linearity resembles a similar phenomena in algorithms to optimize concave or sub-modular functions; see for instance a survey by Vondrák [Von10].

Chapter 9

Bethe Approximation via the Lens of Polynomials

In this chapter we consider counting problems specified by factor graphs. We propose a new approach for establishing bounds on the counting problem (partition functions) for graphical models based on polynomial techniques. To this end we express the Bethe approximation as a polynomial optimization problem and subsequently establish that when the underlying polynomials are real stable (as it is the case for several interesting graphical models), then the Bethe approximation gives a lower bound to the partition function.

9.1 Preliminaries and Statement of Results

9.1.1 Preliminaries

Normal Factor Graphs. We work with probability distributions represented by Normal Factor Graphs (NFGs). In an NFG $G = (F, E, \{g_a\}_{a\in F})$, there is a set of factors (or nodes) F and a set of variables (or edges) E. Every edge $e \in E$ connects exactly two factors. The set of edges incident to a factor $a \in F$ is denoted by $\partial a \subseteq E$. The last component of G is a collection of *local functions* $\{g_a\}_{a\in F}$. Every such function g_a takes as input a binary string of length $|\partial a|$ and outputs a non-negative number, in other words $g_a : \{0,1\}^{\partial a} \to \mathbb{R}_{\geqslant 0}$. For a given vector $\sigma \in \{0,1\}^E$ and any set of edges $S \subseteq E$ we denote by σ_S the sub-vector of σ of length $|S|$ indexed by edges in S. Edges are to be thought of as variables that can take one of two possible values: 0 or 1. Then the set of all possible configurations of G is $\{0,1\}^E$. Consider the probability distribution p on $\{0,1\}^E$ by setting

$$\begin{aligned} p_\sigma &:= \frac{\prod_{a\in F} g_a(\sigma_{\partial a})}{Z(G)} \qquad \text{for } \sigma \in \{0,1\}^E, \\ Z(G) &:= \sum_{\sigma\in\{0,1\}^E} \prod_{a\in F} g_a(\sigma_{\partial a}). \end{aligned} \tag{9.1}$$

It is always assumed that $Z(G) \neq 0$, in which case p is a well defined probability distribution over configurations. The focus here is on the problem of estimating $Z(G)$ for a given normal factor graph G.

Note that in a related model of *factor graphs*, variables are represented by variable nodes, whereas in the model considered here they are represented by edges. However, a simple reduction shows that these two models are equivalent [FJV11]. We choose to work with normal factor graphs to allow a cleaner statement of results.

Bethe Approximation is a popular heuristic called for computing $Z(G)$. It is based on computing a quantity $Z_B(G)$ – called the Bethe partition function of G – as a solution to a continuous optimization problem defined with respect to G. To derive the Bethe approximation, one begins with the following convex program

$$\begin{aligned} \sup_{q} \quad & \sum_{\sigma} q_\sigma \log \frac{g(\sigma)}{q_\sigma} \\ \text{s.t.} \quad & \sum_{\sigma \in \{0,1\}^E} q_\sigma = 1, \\ & q \geqslant 0. \end{aligned} \tag{9.2}$$

where $g(\sigma) = \prod_{a \in F} g_a(\sigma_{\partial a})$. It is not hard to prove that the above program has an optimal solution $q^\star = p$ (with p as in (9.1)), and the optimal value is $\log Z(G)$. Thus the problem of computing the partition function is reduced to solving the program (9.2). This reduction, however, does not seem to make the problem any easier, as the number of variables in (9.2) is exponential. Thus, various heuristics have been proposed on how to reduce the number of variables in (9.2) so as to make this approach of estimating $\log Z(G)$ feasible.

The Bethe approximation has variables $\beta_e \in [0,1]$ for $e \in E$, which are the marginals of the distribution $\{q_\sigma\}_{\sigma \in \{0,1\}^E}$, more formally we think of β_e as $\mathbf{Pr}[X_e = 1]$ where $X \in \{0,1\}^E$ is distributed according to q. Similarly one introduces variables representing marginals over factors, i.e. for $a \in F$ we have a vector α_a which is a probability distribution over local configurations $\{0,1\}^{\partial a}$, and its interpretation is that $\alpha_a(c) = \mathbf{Pr}[X_a = c]$. To simplify the program (9.2) the following assumption is made about the form of the distribution $\{q_\sigma\}_{\sigma \in \{0,1\}^E}$

$$\forall_{\sigma \in \{0,1\}^E} \qquad q_\sigma = \frac{\prod_{a \in F} \alpha_a(\sigma_{\partial a})}{\prod_{e \in E} \beta_e^{\sigma_e} (1-\beta_e)^{1-\sigma_e}}. \tag{9.3}$$

The intuition behind such a form of q_σ is that one might (for simplicity) assume independence between factors and calculate the probability of a global configuration as a product of probabilities over local configurations of factors. The term in the denominator can be thought of as a correction term, as every edge is "taken twice into account" in the numerator. Another way of motivating (9.3) is to observe that when the graph G is a tree, then the probability function can be written in this form and, wishfully, one may expect that for other graphs it might serve as a good estimate. Assuming such a special form of q, the program (9.2) reduces to

$$\begin{aligned} \sup_{\alpha,\beta} \quad & \sum_{a \in F} \sum_{c \in \{0,1\}^{\partial a}} \alpha_a(c) \log \frac{g_a(c)}{\alpha_a(c)} - \sum_{e \in E} H(\beta_e) \\ \text{s.t.} \quad & (\alpha, \beta) \in \Gamma(G) \end{aligned} \tag{9.4}$$

where H is the binary entropy function (i.e., $H(x) = -x \log x - (1-x)\log(1-x)$ for $x \in [0,1]$) and $\Gamma(G)$ is the set of all marginal vectors which satisfy *local agreement constraints* (it is thus called the *pseudo-marginal polytope*). This means that β_e and α_a are as above and they satisfy:

$$\sum_{c\in\{0,1\}^{\partial a}} \alpha_a(c)\cdot c = \beta_a \qquad \text{for every } a\in F.$$

The optimal value of (9.4) is called the Bethe partition function and its exponential is denoted by $Z_B(G)$. One expects that $Z_B(G)$ is a decent approximation to $Z(G)$, which has been confirmed empirically for various examples of factor graphs.

However, in general, $Z_B(G)$ can be an arbitrarily bad approximation to $Z(G)$, as for instance it might be positive for some cases where $Z(G)=0$. From a theoretical viewpoint, not much is known about the behavior of Bethe approximation. The main source of difficulty in understanding this relaxation is its non-convexity, which in particular manifests itself in multiple local optima. In this chapter we derive some sufficient conditions under which the Bethe partition function lower-bounds the true partition function.

9.1.2 Statement of Results

Polynomial Form of Bethe Approximation. The main conceptual result of this chapter is a new approach to prove inequalities between the Bethe partition function and the true partition function. We start by presenting an alternative view on the Bethe approximation – through the lens of polynomials. Towards this, let us first define the polynomial representation of local functions. For any $a\in F$ we define a multivariate polynomial h_a over a set of $|\partial a|$ variables $x_a := \{x_{a,e}\}_{e\in\partial a}$ as follows

$$h_a(x_a) := \sum_{\sigma\in\{0,1\}^{\partial a}} h_{a,\sigma} x_a^{\sigma},$$

where x_a^σ is a monomial defined as $x_a^\sigma := \prod_{e\in\partial a} x_{a,e}^{\sigma_e}$ and the coefficient $h_{a,\sigma}$ is given by $h_{a,\sigma} := g_a(\sigma)$. We prove the following, alternative characterization of the Bethe partition function as a polynomial optimization problem. In the statement below we use the convenient notation that for two vectors $x,\sigma\in\mathbb{R}^k$, $x^\sigma := \prod_{i=1}^k x_i^{\sigma_i}$.

Theorem 9.1 (*Bethe Approximation via Polynomials*)

Let G be a normal factor graph with a set of factors F and a set of variables E. For every factor $a\in F$ let h_a be the corresponding $|\partial a|$-variate polynomial. Then the Bethe partition function can be written as

$$Z_B(G) = \max_{\beta\in[0,1]^E}\left[\prod_{e\in E}\beta_e^{\beta_e}(1-\beta_e)^{1-\beta_e}\inf_{x>0}\prod_{a\in F}\frac{h_a(x_a)}{x_a^{\beta_a}}\right]$$

In the above statements x stands for a vector which collects all variables $x_{a,e}$ for $a\in F$ and $e\in\partial a$. The proof of Theorem 9.1 appears in Section 9.2.1. It is established by adapting a dual view on the max-entropy program which defines the Bethe partition function.

Lower Bound on the Partition Function. We prove that assuming a certain geometric condition on the factor graph G, the Bethe approximation provides a lower bound on the true partition function. This condition captures permanents as a special case. Below we state a simplified variant of the main technical result in terms of *local polynomials* h_a. For a more general statement, which is expressed in the language of probability, as well as a proof of the below theorem, we refer to Section 9.2.2.

Theorem 9.2 (*Lower Bound via Real Stability*)

Let G be a bipartite normal factor graph with a set of factors F and a set of variables E. Assume that all the polynomials h_a corresponding to local functions g_a (for $a \in F$) are real stable. Then it holds that $Z_B(G) \leqslant Z(G)$.

A few comments are in order. In the statement above we assume that the NFG G is bipartite. This might seem to be restrictive, but as it turns out, every NFG can be converted into an equivalent bipartite form, with at most a double growth in size, hence no real restriction is put on G with this assumption. The key condition we require is *real stability* of the underlying polynomials.

We remark that coefficients of multi-affine real stable polynomials are known to be given by log-submodular set functions (see [Wag11]), which corresponds to the following assumption on local functions g_a for $a \in F$

$$\forall_{\sigma,\tau \in \{0,1\}^{\partial a}} \qquad g_a(\sigma) \cdot g_a(\tau) \geqslant g_a(\sigma \vee \tau) \cdot g_a(\sigma \wedge \tau).$$

This demonstrates that Theorem 9.2 addresses the opposite case when compared to the result of [Ruo12], where an analogous result for log-supermodular functions is proved. These two assumptions turn out to imply significantly different properties of the underlying factor graphs.

One interesting aspect that is worth mentioning here is that, under log-supermodularity, *feasible fractional configurations* are easy to round to *integral configurations*. More precisely, given a point $(\alpha, \beta) \in \Gamma(G)$ whose objective value in the Bethe approximation is finite (larger than $-\infty$), one can obtain (by just rounding up all entries of β) a configuration $\sigma \in \{0,1\}^E$ such that $g(\sigma) > 0$. Such a procedure might fail in finding a feasible configuration when G is log-submodular (i.e., the resulting σ has $g(\sigma) = 0$). In fact, finding a feasible configuration in such models (even assuming real stability of local polynomials) might be a nontrivial task, even **NP**-complete if no assumptions on the local functions are made. It turns out in particular, that for the case of permanents, the Bethe approximation is implicitly solving a nontrivial combinatorial optimization problem of detecting if a bipartite graph has a perfect matching.

Remark 9.1 (*Upper Bound*)

Using the characterization from Theorem 9.1 one can prove that $Z(G) \leqslant 2^m \cdot Z_B(G)$. Indeed, by plugging in $\beta := \sigma \in \{0,1\}^E$ the term $\prod_{e \in E} \beta_e^{\beta_e}(1-\beta_e)^{1-\beta_e}$ is equal to 1 and we obtain

$$Z_B(G) \geqslant \sum_{\tau \leqslant \sigma} g(\tau) \geqslant g(\sigma)$$

(here by $\tau \leqslant \sigma$ we mean an entry-wise inequality). Hence, altogether, under the assumptions of Theorem 9.2 the Bethe partition function provides a 2^m-approximation to the true partition function.

9.2 Proofs

9.2.1 Bethe Approximation via Polynomials

In this section we derive an equivalent form of the Bethe partition function – stated in terms of a polynomial optimization problem.

Local Functions as Polynomials.

Consider a NFG $G = (F, E, \{g_a\}_{a\in F})$. In this chapter we view the local functions g_a (for $a \in F$) as polynomials. More formally, given a function $g_a : \{0,1\}^{\partial a} \to \mathbb{R}_{\geqslant 0}$, we define the corresponding polynomial representation of g_a as an $|\partial a|$-variate polynomial $h_a(x_a)$ over variables $\{x_{a,e}\}_{e\in\partial a}$ given by the formula

$$h_a(x_a) = \sum_{\sigma\in\{0,1\}^{\partial a}} h_{a,\sigma} x_a^{\sigma},$$

where x_a^σ denotes $\prod_{e\in S} x_{a,e}^{\sigma_e}$ and $h_{a,\sigma} = g_a(\sigma)$ is the value of the function g_a at σ. Note that even if two factors $a, b \in F$ share an edge $e \in E$, the variables of h_a and h_b are still pairwise different.

Bethe Approximation as Polynomial Optimization.

Let $G = (F, E, \{g_a\}_{a\in F})$ be a NFG. Denote by $H(\beta)$ the negative entropy of $\beta \in [0,1]^E$, i.e.,

$$H(\beta) := -\left(\sum_{e\in E} \beta_e \log \beta_e + (1-\beta_e)\log(1-\beta_e)\right).$$

We use $\mathrm{KL}(p,q)$ to denote the KL-divergence between two nonnegative vectors $p, q \in \mathbb{R}^k_{\geqslant 0}$ (typically probability distributions),

$$\mathrm{KL}(p,q) := \sum_{i=1}^{k} p_i \log \frac{p_i}{q_i}.$$

The Bethe approximation problem can be then rewritten as

$$\log Z_B(G) = \max_{(\alpha,\beta)\in\Gamma(G)} -\sum_{a\in F} \mathrm{KL}(\alpha_a, g_a) - H(\beta),$$

where $\Gamma(G)$ is the pseudo-marginal polytope, as introduced in Section 9.1.1. We define the following entropy maximization problem.

Definition 9.1

Let $f : \{0,1\}^k \to \mathbb{R}_{\geqslant 0}$ be any function with $C(f) = \{\sigma \in \{0,1\}^k : f(\sigma) > 0\}$ and $\beta \in [0,1]^k$ be any vector. We define $E_{\max}(f, \beta)$ to be the optimal value of the following optimization problem over vectors $\alpha \in \mathbb{R}^{C(f)}$

$$\begin{aligned} \max_{\alpha} \quad & -\mathrm{KL}(\alpha, f) \\ \text{s.t.} \quad & \sum_{c \in C(f)} \alpha(c) \cdot c = \beta, \\ & \sum_{\sigma \in C(f)} \alpha_\sigma = 1 \\ & \alpha \geqslant 0. \end{aligned} \tag{9.5}$$

In case when no α satisfies the above constraints, we set $E_{\max}(f, \beta) = -\infty$.

Lemma 9.1

For every normal factor graph G, the Bethe approximation can be stated equivalently as

$$\log Z_B(G) = \max_{\beta \in [0,1]^E} \sum_{a \in F} E_{\max}(g_a, \beta_a) - H(\beta).$$

Proof.
The objective of the Bethe approximation $-\sum_{a \in F} \mathrm{KL}(\alpha_a, g_a) - H(\beta)$ has separated α and β variables, however they are implicitly coupled because of the $(\alpha, \beta) \in \Gamma$ constraint. For a fixed β and a factor $a \in F$ the constraint on α_a following from $(\alpha, \beta) \in \Gamma$ is

$$\sum_{c \in C_a, c_e = 1} \alpha_a(c) = \beta_e \qquad \text{for every } e \in E.$$

This can be equivalently written in the vector form as

$$\sum_{c \in C_a} \alpha_a(c) \cdot c = \beta_a.$$

Note that maximizing $-\mathrm{KL}(\alpha_a, g_a)$ under this constraint gives us exactly $E_{\max}(g_a, \beta_a)$. □

The lemma below explains how does the entropy maximization problem underlying $E_{\max}$ relate to polynomial optimization.

Lemma 9.2

Let $f : \{0,1\}^k \to \mathbb{R}_{\geqslant 0}$ and $\beta \in [0,1]^k$ be any vector. Define a k-variate, multi-linear polynomial $h \in \mathbb{R}[x_1, \ldots, x_k]$ to be $h(x) = \sum_{\sigma \in \{0,1\}^k} h_\sigma x^\sigma$ with $h_\sigma := f(\sigma)$. We have

$$E_{\max}(f, \beta) = \inf_{x \in \mathbb{R}^k, x > 0} \quad \log h(x) - \sum_{i=1}^{k} \beta_i \log x_i.$$

Proof.
A proof follows by applying strong duality to the max-entropy program (9.5). □

Theorem 9.1 is now a simple consequence of the above established results.

Proof of Theorem 9.1:
From Lemma 9.1 we have

$$\log Z_B(G) = \max_{\beta \in [0,1]^E} \sum_{a \in F} E_{\max}(g_a, \beta_a) - H(\beta).$$

Next, by Lemma 9.2 this can be rewritten as

$$\log Z_B(G) = \max_{\beta \in [0,1]^E} \sum_{a \in F} \inf_{x_a > 0} \left(\log h_a(x_a) - \sum_{e \in \partial a} \beta_e \log x_{a,e} \right) - H(\beta).$$

By taking exponentials on both sides

$$Z_B(G) = \max_{\beta \in [0,1]^E} \prod_{e \in E} \beta_e^{\beta_e} (1-\beta_e)^{1-\beta_e} \prod_{a \in F} \inf_{x_a > 0} \frac{h_a(x_a)}{\prod_{e \in \partial a} x_a^{\beta_a}}.$$

□

9.2.2 Proof of the Lower Bound

To prove Theorem 9.2 we first formulate a more general condition which we call IPC, and prove that under IPC, the inequality $Z_B(G) \leqslant Z(G)$ holds. Afterwards we conclude the proof by showing that the assumption of Theorem 9.2 implies that IPC is satisfied.

The IPC.

To state IPC we need to introduce some notation related to the bipartite structure of the factor graph $G = (F, E)$. Let the set of factors F be partitioned into two sets L and R such that no edges go between factors within L or within R, only between these two sets. Next, for any $\sigma \in \{0,1\}^E$ we define

$$l_\sigma = \prod_{a \in L} g_a(\sigma_{\partial a}) \qquad \text{and} \qquad r_\sigma = \prod_{a \in R} g_a(\sigma_{\partial a}).$$

Furthermore we define the normalized variants of l and r to be $p^L_\sigma = \frac{l_\sigma}{\sum_{\sigma'} l_{\sigma'}}$, $p^R_\sigma = \frac{r_\sigma}{\sum_{\sigma'} r_{\sigma'}}$. We refer to p^L, p^R as to the distributions induced by L (the "left" side of the bipartition) and induced by R (the "right" side of the bipartition) respectively.

We are now ready to state a condition on the pair of distributions (p^L, p^R) which will turn out sufficient for the inequality $Z_B(G) \leqslant Z(G)$ to hold.

Definition 9.2 (***Iterated Positive Correlation***)

Let q, r be probability distributions over $\{0,1\}^m$ and let $X, Y \in \{0,1\}^m$ be distributed according to q and r respectively. Define the event EQ_k to be $X_j = Y_j$ for all $j = 1, 2, \ldots, k$. For any two sequences of positive reals $s \in \mathbb{R}^m_{>0}$ and $t \in \mathbb{R}^m_{>0}$ and for any pair $A, B \in \{0,1\}$ define

$$E_k(A,B) = \mathbb{E}\left[\prod_{j=k+1}^{m} s_j^{X_j} t_j^{Y_j} \cdot \mathbb{1}_{X_k=A} \cdot \mathbb{1}_{Y_k=B} \middle| EQ_{k-1}\right]$$

Where the expectation is over X and Y, assuming X, Y are independent. We say that the pair of distributions (q, r) satisfies the Iterated Positive Correlation (IPC) property if

$$E_k(0,1) \cdot E_k(1,0) \leqslant E_k(0,0) \cdot E_k(1,1)$$

for every $k \in [m]$ and for every $s, t \in \mathbb{R}^m_{>0}$.

Note that in the definition above we implicitly assume that $\mathbf{Pr}[EQ_m] \neq 0$, as otherwise some conditional expectations above might not be well defined. For the setting which we have in mind, this corresponds to the assumption that $Z(G) \neq 0$.

To gain some intuition about the IPC property it is instructive to examine the special case when $s_1 = \ldots = s_m = t_1 = \ldots = t_m = 1$. Under the notation $p_k(A,B) := \mathbf{Pr}[X_k = A \wedge Y_k = B | EQ_{k-1}]$ we obtain

$$p_k(0,1) \cdot p_k(1,0) \leqslant p_k(0,0) \cdot p_k(1,1),$$

which can be seen as a form of iterated (as $k = 1, 2, \ldots, m$) positive correlation between subsequent X_k's and Y_k's. In other words, it quantifies, in a certain sense the fact that conditioned on $X_i = Y_i$ for $i = 1, 2, \ldots, k-1$, it is more likely to see $X_k = Y_k$ rather than $X_k \neq Y_k$. We are now ready to state the main technical lemma of this chapter, which asserts that if a NFG G satisfies IPC then $Z_B(G) \leqslant Z(G)$.

Lemma 9.3

Let G be a bipartite normal factor graph with a set of factors F, a set of variables E and bipartition $F = L \cup R$. Let p^L and p^R be the distributions over $\{0,1\}^E$ induced by the left side and the right side of the bipartition of G respectively. If the pair (p_L, p_R) satisfies the IPC property then

$$Z_B(G) \leqslant Z(G).$$

A proof of Lemma 9.3 appears in Section 9.2.2. To conclude Theorem 9.2 from the above it suffices to argue that the real stability assumption on local polynomials implies IPC. This is the subject of the next lemma

Lemma 9.4 (***Real Stability implies IPC***)

Let G be a bipartite normal factor graph with a set of factors F and a set of variables E. Assume that all the polynomials h_a corresponding to local functions g_a (for $a \in F$) are real stable. Let p^L and p^R be the distributions over $\{0,1\}^E$ induced by the left side and the right side of the bipartition of G respectively. Then the pair (p^L, p^R) satisfies the IPC property.

The proof of Lemma 9.4 appears in Section 9.2.2. We are now ready to deduce Theorem 9.2.

Proof of Theorem 9.2:

Lemma 9.3 asserts that the inequality $Z_B(G) \leqslant Z(G)$ holds under IPC. Further, by Lemma 9.4 the assumption of Theorem 9.2 (saying that local polynomials are real stable) implies that IPC holds. Thus the Theorem 9.2 follows. □

Remark 9.2

We note that the IPC condition is significantly more general than the real stability assumption in 9.2 and there are examples of factor graphs which do not satisfy real stability, but IPC holds for them. The downside of IPC might be however that there does not seem to be a simple way to verify it, especially since it is a global condition on the factor graph. On the other hand, the real stability assumption is only local and can be checked easily whenever the degrees of all factors are reasonably small.

Proof of the Lower Bound under IPC.

In this section, the following linear operator on the set of polynomials is used.

Definition 9.3

Let $h(z_0, z, y_0, y)$ be a real polynomial with z_0, y_0 being single variables and y, z being tuples of variables. Define

$$\Phi_{z_0,y_0}(h) := (1 + \partial_{z_0}\partial_{y_0})h\ |_{z_0=y_0=0}\ .$$

In other words, Φ_{z_0,y_0} first applies the differential operator $(1 + \partial_{z_0}\partial_{y_0})$ to h and then sets $z_0 = y_0 = 0$; the result is a polynomial in the variables (y, z).

The lemma below explains how the IPC property is related to polynomials.

Lemma 9.5

Let q, r be distributions over $\{0,1\}^m$. Define the polynomials $q(z) := \sum_{\sigma\in\{0,1\}^m} q_\sigma z^\sigma$ and $r(y) := \sum_{\tau\in\{0,1\}^m} r_\tau y^\tau$. Further, for every $k = 0, 1, \ldots, m$ let

$$f_k(z_{k+1}, \ldots, z_m, y_{k+1}, \ldots, y_m) := \Phi_{z_k,y_k} \cdots \Phi_{z_2,y_2}\Phi_{z_1,y_1}\, [q(z) \cdot r(y)]$$

For any number $k = 1, 2, \ldots, m+1$ and for any two sequences $a, b \in \mathbb{R}^{m-k}_{\geqslant 0}$ of non-negative numbers, the polynomial $f_{k-1}(z_k, a_{k+1}, \ldots, a_m, y_k, b_{k+1}, \ldots, b_m)$ is of the form

$$h(z_k, y_k) = h_{00} + h_{10}z_k + h_{01}y_k + h_{11}z_k y_k,$$

where (up to scaling) $h_{cd} = E_k(c, d)$ for every $c, d \in \{0, 1\}$ (as in Definition 9.2 with $a = s$ and $b = t$).

Proof.

We start by providing explicit formulas for the coefficients of f_{k-1}. Note first that all the operators Φ_{z_i,y_i} are linear. Hence it is enough to consider only one monomial $\prod_{i=1}^m z_i^{\sigma_i} \prod_{i=1}^m y_i^{\tau_i}$, for $\sigma, \tau \in \{0,1\}^m$.

$$\Phi_{z_{k-1},y_{k-1}} \cdots \Phi_{z_2,y_2}\Phi_{z_1,y_1}\left(\prod_{i=1}^m z_i^{\sigma_i} \prod_{i=1}^m y_i^{\tau_i}\right) = \begin{cases} \prod_{i=k+1}^m z_i^{\sigma_i} \prod_{i=k+1}^m y_i^{\tau_i} & \text{if } \sigma_1 = \tau_1, \ldots \sigma_k = \tau_k, \\ 0 & \text{otherwise.} \end{cases}$$

For this reason, the coefficient of $\prod_{i=k}^m z_i^{\sigma_i} \prod_{i=k}^m y_i^{\tau_i}$ in f_{k-1} is equal to

$$\sum_{u \in \{0,1\}^{k-1}} q_{u\widetilde{\sigma}} \cdot r_{u\widetilde{\tau}}.$$

where $\widetilde{\sigma} = (\sigma_k, \sigma_{k+1}, \dots, \sigma_m)$ and $\widetilde{\tau} = (\tau_k, \tau_{k+1}, \dots, \tau_m)$. In the language of probability this coefficient is equal to the probability that

$$\begin{aligned} X_i &= Y_i, && \text{for } i = 1, 2, \dots, k-1, \\ X_i &= \sigma_i, && \text{for } i = k, k+1, \dots, m, \\ Y_i &= \tau_i, && \text{for } i = k, k+1, \dots, m. \end{aligned}$$

when X and Y are distributed according to q and r respectively. Thus, when we consider $h_k(z_k, y_k) = f_{k-1}(z_k, a, y_k, b)$ for some $a, b \in \mathbb{R}_{\geqslant 0}^{m-k}$, the corresponding coefficients h_{cd} are given by sums of the form

$$\sum_{u \in \{0,1\}^{k-1}} \sum_{\widetilde{\sigma} \in \{0,1\}^{m-k}} \sum_{\widetilde{\tau} \in \{0,1\}^{m-k}} q_{uc\widetilde{\sigma}} \cdot r_{ud\widetilde{w}} \cdot a^{\widetilde{\sigma}} \cdot b^{\widetilde{\tau}}.$$

Again, probabilistically this corresponds to

$$\mathbb{E}\left[\prod_{j=k+1}^m a_j^{X_j} b_j^{Y_j} \cdot \mathbb{1}_{X_k = c} \cdot \mathbb{1}_{Y_k = d} \cdot \mathbb{1}_{EQ_{k-1}}\right],$$

and the lemma follows. □

Lemma 9.6 (*[AO17]*)

Suppose $h(x, y) = h_{00} + h_{10}x + h_{01}y + h_{11}xy$ is a bivariate multi-linear polynomial such that $h_{ij} \geqslant 0$ for all $i, j \in \{0, 1\}$ and $h_{10} \cdot h_{01} \leqslant h_{00} \cdot h_{11}$, then for every $\beta \in \mathbb{R}_{\geqslant 0}$

$$\inf_{x,y>0} \frac{h(x, y)}{x^\alpha y^\alpha} \alpha^\alpha (1-\alpha)^{1-\alpha} \leqslant h_{00} + h_{11}.$$

Proof.
Fix any $\alpha \geqslant 0$. It is not hard to prove that for $\alpha > 1$, the left hand side of the inequality is actually 0, hence we can focus on $\alpha \in [0, 1]$. Note also that we can assume that $h_{10} \cdot h_{10} = h_{00} \cdot h_{11}$ since if $h_{10} \cdot h_{10} < h_{00} \cdot h_{11}$, we can keep increasing h_{10} until the inequality becomes an equality, this way we might only increase the value of

$$\inf_{x,y>0} \frac{h(x, y)}{x^\alpha y^\alpha}$$

but $h_{00} + h_{11}$ stays the same. From $h_{10} \cdot h_{10} = h_{00} \cdot h_{11}$ it then follows that that

$$h(x) = (a_0 + a_1 x)(b_0 + b_1 x)$$

for some $a_0, a_1, b_0, b_1 \geqslant 0$. Now using Lemma 9.2 we obtain

$$\inf_{x>0} \frac{a_0 + a_1 x}{x^\alpha} = \exp(\mathrm{KL}(a, \widetilde{\alpha})),$$
$$\inf_{y>0} \frac{b_0 + b_1 y}{y^\alpha} = \exp(\mathrm{KL}(b, \widetilde{\alpha})).$$

Where $a = (a_0, a_1)$, $b = (b_0, b_1)$ and $\widetilde{\alpha} = (\alpha, 1-\alpha)$. Therefore

$$\inf_{x,y>0} \frac{h(x,y)}{x^\alpha y^\alpha} \alpha^\alpha (1-\alpha)^{1-\alpha} = \exp(\mathrm{KL}(ab, \widetilde{\alpha})).$$

Where $ab = (a_0 b_0, a_1 b_1)$. What then remains to prove is that

$$\mathrm{KL}(ab, \widetilde{\alpha}) \leqslant \log(a_0 b_0 + a_1 b_1).$$

However, this follows from the fact that the KL-divergence between two probability distributions $p, q \in \Delta_2$ is nonnegative, when applied to: $(p_1, p_2) = (\alpha, 1-\alpha)$ and $(q_1, q_2) = \left(\frac{a_0 b_0}{a_0 b_0 + a_1 b_1}, \frac{a_1 b_1}{a_0 b_0 + a_1 b_1}\right)$. □

Lemma 9.7

Let q, r be distributions over $\{0,1\}^m$ satisfying the IPC property. Then

$$\sup_{\beta \in [0,1]^m} \left[\beta^\beta (1-\beta)^{1-\beta} \inf_{y,z>0} \frac{q(z)}{z^\beta} \cdot \frac{r(y)}{y^\beta} \right] \leqslant \sum_\sigma q_\sigma r_\sigma.$$

Proof.
We proceed by induction. Observe first that $\sum_{\sigma \in \{0,1\}^m} q_\sigma r_\sigma$ can be obtained from $q(z) \cdot r(y)$ by applying a sequence of differential operators. More precisely, let us define

$$g_0(z_1, \dots, z_m, y_1, \dots, y_m) := q(z_1, \dots, z_m) r(y_1, \dots, y_m),$$
$$g_k(z_{k+1}, \dots, z_m, y_{k+1}, \dots, y_m) := \Phi_{z_k, y_k} \left(g_{k-1}(z_k, \dots, z_m, y_k, \dots, y_m) \right).$$

Note that g_m is a constant polynomial given by

$$g_m = \sum_{\sigma \in \{0,1\}^m} q_\sigma r_\sigma. \tag{9.6}$$

Let us fix $\beta \in [0,1]^m$, we prove that for every $k = 0, 1, \dots, m$

$$\prod_{j=k+1}^m \beta_j^{\beta_j} (1-\beta_j)^{1-\beta_j} \inf_{\widetilde{z}, \widetilde{y} > 0} \frac{g_k(\widetilde{z}, \widetilde{y})}{\prod_{j=k+1}^m z_j^{\beta_j} y_j^{\beta_j}} \leqslant \sum_{\sigma \in \{0,1\}^m} q_\sigma r_\sigma \tag{9.7}$$

Where $\widetilde{y} = (y_{k+1}, \dots, y_m)$ and $\widetilde{z} = (z_{k+1}, \dots, z_m)$ Note that for $k = 0$ we obtain the lemma. We proceed by induction starting from the base case $k = m$ and go backwards with $k = m-1, \dots, 1, 0$. The base case follows directly (with equality) from (9.6). Suppose now that $k \in \{1, \dots, m\}$ and (9.7) has been proved for all k' with $k' \geqslant k$, we prove it for $k - 1$. □

Let us fix $\varepsilon > 0$ and values $a_{k+1}, \dots, a_m, b_{k+1}, b_m > 0$ such that

$$\prod_{j=k+1}^{m} \beta_j^{\beta_j}(1-\beta_j)^{1-\beta_j} \frac{g_k(a,b)}{\prod_{j=k+1}^{m} a_j^{\beta_j} b_j^{\beta_j}} \leqslant \varepsilon + \sum_{\sigma\in\{0,1\}^m} q_\sigma r_\sigma.$$

It remains to show that

$$\inf_{z_k,y_k>0} \frac{g_{k-1}(z_k,a,y_k,b)}{z_k^{\beta_k} y_k^{\beta_k}} \leqslant g_k(a,b). \tag{9.8}$$

Towards this, write $g_{k-1}(z_k,a,y_k,b)$ as a polynomial in z_k, y_k

$$g_{k-1}(z_k,a,y_k,b) = g_{k-1}^{00}(a,b) + g_{k-1}^{10}(a,b)z_k + g_{k-1}^{10}(a,b)y_k + g_{k-1}^{11}(a,b)z_k y_k,$$

and note that (9.8) follows from Lemma 9.6 if only we can justify its assumption, that is

$$g_{k-1}^{10}(a,b) \cdot g_{k-1}^{01}(a,b) \leqslant g_{k-1}^{00}(a,b) \cdot g_{k-1}^{11}(a,b).$$

The above follows from the IPC property because from Lemma 9.5 we have

$$E_{k-1}(a,b;c,d) \propto g_{k-1}^{cd}(a,b)$$

for every $c, d \in \{0,1\}$.

Proof of Lemma 9.3:
From Theorem 9.1 the Bethe approximation can be stated in the form of a polynomial optimization problem

$$Z_B(G) = \max_{\beta} \left[\prod_{e\in E} \beta_e^{\beta_e}(1-\beta_e)^{1-\beta_e} \inf_{u>0} \prod_{a\in F} \frac{h_a(x_a)}{x_a^{\beta_a}} \right].$$

Let $L \cup R = F$ be the bipartition of the set of factor nodes, i.e., there is no edge $e \in E$ within L or R. In other words, every edge e has one endpoint $a_e^L \in L$ and one endpoint $a_e^R \in R$, or in other words $e = \{a_e^L, a_e^R\}$.

Let us split the product $\prod_{a\in F} \frac{h_a(x_a)}{x_a^{\beta_a}}$ into two parts

$$\prod_{a\in F} \frac{h_a(x_a)}{x_a^{\beta_a}} = \prod_{a\in L} \frac{h_a(x_a)}{x_a^{\beta_a}} \cdot \prod_{a\in R} \frac{h_a(x_a)}{x_a^{\beta_a}}, \tag{9.9}$$

corresponding to the bipartition. Let us now rename the variables in the above. For an edge $e \in E$ and $a = a_e^L$ we rename the variable $x_{a,e}$ to z_e. Similarly, if $a = a_e^R$ we rename $x_{a,e}$ to y_e. Because the factor graph G is bipartite, the product (9.9) can be then rewritten as

$$\prod_{a\in L} \frac{h_a(z_a)}{z_a^{\beta_a}} \cdot \prod_{a\in R} \frac{h_a(y_a)}{y_a^{\beta_a}}.$$

In the above $z_a = \{z_e\}_{e \in \partial a}$, similarly for y_a. Let us now define two polynomials q, r as follows

$$q(z) = \prod_{a \in L} q_a(z_a),$$
$$r(y) = \prod_{a \in R} q_a(y_a).$$

The expression (9.9) can be then further simplified to

$$\frac{q(z)}{z^\beta} \cdot \frac{r(y)}{y^\beta}.$$

Consequently, we arrive at the following form of the Bethe partition function

$$Z_B(G) = \max_{\beta \in [0,1]^m} \left[\beta^\beta (1-\beta)^{1-\beta} \inf_{y,z>0} \frac{q(z)}{z^\beta} \cdot \frac{r(y)}{y^\beta} \right].$$

Since by the assumption, the corresponding distributions (q, r) satisfy the IPC property, Lemma 9.7 implies that

$$\max_\beta \left[\beta^\beta (1-\beta)^{1-\beta} \inf_{y,z>0} \frac{q(z)}{z^\beta} \cdot \frac{r(y)}{y^\beta} \right] \leqslant \sum_{\sigma \in \{0,1\}^E} q_\sigma r_\sigma.$$

It remains to observe that $\sum_{\sigma \in \{0,1\}^E} q_\sigma r_\sigma = Z(G)$. To prove it, let us first interpret what the coefficients q_σ, r_σ mean in terms of the underlying factor graph. It is not hard to see that

$$q_\sigma = \prod_{a \in L} g_a(\sigma_{\partial a}), \qquad r_\sigma = \prod_{a \in R} g_a(\sigma_{\partial a}).$$

Hence

$$\sum_{\sigma \in \{0,1\}^E} q_\sigma r_\sigma = Z(G),$$

which concludes the proof. □

Proof of Lemma 9.4.

Proof.
Consider the polynomials $q(z)$ and $p(y)$ as constructed in the proof of Lemma 9.3. When written in the form

$$q(z) = \sum_{\sigma \in \{0,1\}^m} q_\sigma z^\sigma, \qquad r(y) = \sum_{\sigma \in \{0,1\}^m} r_\sigma y^\sigma,$$

the coefficients satisfy, for every $\sigma \in \{0,1\}^m$

$$q_\sigma = \prod_{a \in L} g_a(\sigma), \quad \text{and} \quad r_\sigma = \prod_{a \in R} g_a(\sigma).$$

In other words $q_\sigma \propto p^L_\sigma$ and $r_\sigma \propto p^R_\sigma$.

Note that $f(z,y) := q(z) \cdot p(-y)$ is a real stable polynomial, since $q(z)$ and $p(y)$ are real stable as products of real stable polynomials (see [Vis13]).

As observed by [AO17], if for a multi-affine, real polynomial $h(z,y)$ (for $z = (z_1, \ldots, z_m)$

and $y = (y_1, \ldots, y_m)$), $h(z, -y)$ is real stable then $\widetilde{h}(\tilde{z}, -\widetilde{y})$ is real stable as well, where $\widetilde{y} = (y_2, \ldots, y_m)$, $\widetilde{z} = (z_2, \ldots, z_m)$ and $\widetilde{h}(\widetilde{z}, \widetilde{y}) = \Phi_{z_1,y_1}(h)$.

Define a sequence of polynomials $f_0, f_1, \ldots, f_m$ by setting: $f_0 = f(z, y)$ and for $k = 1, 2, \ldots, m$

$$f_k(z_{k+1}, \ldots, z_m, y_{k+1}, \ldots, y_m) := \Phi_{z_k,y_k}(f_{k-1}),$$

By the above stated observation, $f_k(z, -y)$ is real stable, for every $k = 1, 2, \ldots, m$.

We will deduce the IPC property from real stability of $f_0, f_1, \ldots, f_m$. Indeed, by Lemma 9.5 we know that for every $k \in \{1, 2, \ldots, m\}$, $a, b \in \mathbb{R}^{m-k}_{\geqslant 0}$ and $c, d \in \{0, 1\}$, $E_k(a, b; c, d)$ can be expressed as the appropriate coefficient of the polynomial

$$h_{k-1}(z_k, y_k) = f_{k-1}(z_k, a, y_k, b) = h_{00} + h_{10}z_k + h_{01}y_k + h_{11}z_k y_k.$$

Since $f_k(z, -y)$ is real stable, it follows, that $h_{k-1}(z_k, -y_k)$ is real stable. Indeed, this is a consequence of the fact that plugging in real constants into a real stable polynomial preserves real stability (see [Vis13]). Using a characterization of multilinear real stable polynomials by [Brä07], the real stability of h_{k-1} is equivalent to:

$$h_{10} \cdot h_{01} \leqslant h_{00} \cdot h_{11},$$

hence the IPC property holds. □

9.3 Conclusion

9.3.1 Future Work

The real stability condition which we assume to prove the bound in Theorem 9.2 also improves computational properties of the Bethe approximation. Indeed, the fact that the function $x \mapsto \log p(x)$ is concave, for a real stable polynomial $p \in \mathbb{R}_{\geqslant 0}[x_1, \ldots, x_m]$, can be used to show efficient computability of certain relaxations, similar to the Bethe partition function in the polynomial form (see [AO17]). This might eventually lead to designing relaxations which match or even outperform Bethe approximation, while having provably correct and efficient algorithms.

We remark that even though our result seems to require real stability (with respect to the upper-half complex plane) of the underlying polynomials to deduce the desired bound, we believe that other forms of stability, such as stability with respect to a disc, or other analytic assumptions on the polynomials might yield other nontrivial bounds.

9.3.2 Notes

The content of this chapter is based on [SV17b]. The notion of free energy that appears as the objective in the Bethe partition function was formulated in [Bet35] in the physics literature. See also [Mor13] and references therein for more historical notes on Bethe approximation. The correspondence between Bethe approximation and the belief propagation algorithm was explicitly derived in [YFW05]. This combined with the work [Pea89] on the belief propagation method implies that Bethe approximation gives exact values of the partition function on tree factor graphs. It is also known that Bethe partition function gives precise estimates in the asymptotic sense on locally tree-like graphs [DM10].

In the work [CC06], the loop series expansion of the Bethe partition function was introduced, which is a tool to study the relation between the Bethe partition function and the true partition function. In [CC06] the loop expansion was used to prove that Bethe approximation gives a good estimate on the number of independent sets on graphs with small maximum degree and large girth.

The problem of computing permanents of nonnegative matrices has been also intensively studied in the context of Bethe approximation [WC10, Von13b, Gur11, GS14]. Recall that the permanent of a matrix $A \in \mathbb{R}^{n\times n}$ is defined to be

$$\mathrm{Per}(A) := \sum_{\sigma\in S_n} \prod_{i=1}^{n} A_{i,\sigma(i)}$$

and the problem of computing it is a canonical example of a $\#\mathbf{P}$-hard problem [Val79], hence no polynomial time exact algorithm is expected to exist. This problem can be formulated in a natural way as evaluating a certain partition function $Z(G)$ [WC10, Von13b] and hence one can investigate the question on how well the Bethe partition function does approximate permanents.

It has been observed [Von13b] that unlike in the general case, for permanents the program (9.4) is convex. This allows one to analyze the optimality via KKT conditions and to conclude that $Z_B(G) \leqslant Z(G)$ using a permanental inequality due to [Sch98]. The success of this approach crucially relies on the existence of a convex form of the Bethe approximation, this seems to be an exception rather than a rule among various factor graphs.

The Bethe approximation was also studied in the context of the *Ising model* [SWW07], and shown to lower-bound the true partition function for the *ferromagnetic* case under certain technical assumption. This result was extended by [Ruo12] to the class of all *log-supermodular* (also called *attractive*) factor graphs. A factor graph is called log-supermodular if every local function is log-supermodular, i.e., for every $a \in F$ we have

$$\forall_{\sigma,\tau\in\{0,1\}^{\partial a}} \qquad g_a(\sigma)\cdot g_a(\tau) \leqslant g_a(\sigma\vee\tau)\cdot g_a(\sigma\wedge\tau),$$

where $\vee$ and $\wedge$ denote entry-wise OR and entry-wise AND respectively. The proof is based on the following combinatorial characterization of the Bethe approximation, due to [Von13b, Von13a]. It says that

$$Z_B(G) = \limsup_{k\to\infty} \sqrt[k]{\mathbb{E}_{H\in G^{(k)}} Z(H)},$$

where $G^{(k)}$ is the set of k-covers of the factor graph G, and the expectation is over a uniformly random choice of H in $G^{(k)}$ (for details we refer to [Von13a]). It follows that in order to prove that $Z_B(G) \leqslant Z(G)$ for a given factor graph G, it is enough to prove that for every $k \in \mathbb{N}$

$$Z(H) \leqslant Z(G)^k, \quad \text{for every } k\text{-cover } H \text{ of } G.$$

In the context of attractive models, several conjectures regarding similar lower bounds were stated in [Wat11], out of which only one (for independent sets on bipartite graphs) has been so far resolved (by the above result of [Ruo12]).

Chapter 10

Exact Sampling and Counting under Budget Constraints

In this section we consider budget constrained sampling and counting problems over determinantal measures, and more generally – measures represented by efficiently computable polynomials. We prove that for budget constraints specified in unary, or more generally for "succinct" constraints these problems can be solved exactly, in polynomial time. Conversely, when the constraints are not succinct, we prove that this problem can capture a $\#P$-hard problem of computing mixed discriminants.

10.1 Preliminaries and Statement of Results

10.1.1 Preliminaries

We consider the problem of sampling from constrained DPPs (Determinantal Point Processes) in which one is given a kernel matrix $L \in \mathbb{R}^{m \times m}$ and a family of subsets $\mathcal{C} \subseteq 2^{[m]}$ and the goal is to sample a set S, from a probability distribution defined as $\mathbb{P}(S) \propto \det(L_{S,S})$ for $S \in \mathcal{C}$, and $\mathbb{P}(S) = 0$ otherwise, in other words

$$\mathbb{P}(S) = \frac{\det(L_{S,S})}{\sum_{T \in \mathcal{C}} \det(L_{T,T})}.$$

The starting point of our work is the observation that if we let μ be the measure on subsets of $[m]$ corresponding to the kernel matrix L (i.e., $\mu(S) \stackrel{\text{def}}{=} \det(L_{S,S})$), then given L, there is an efficient algorithm to evaluate the polynomial

$$g_\mu(x) \stackrel{\text{def}}{=} \sum_{S \subseteq [m]} \mu(S) x^S$$

where x^S denotes $\prod_{i \in S} x_i$ for any setting of its variables. Indeed, consider the Cholesky decomposition of the kernel $L = VV^\top$. Then, the polynomial $x \mapsto \det(V^\top X V + I)$ (where X denotes the diagonal matrix with x on the diagonal) is equal to $g_\mu(x)$ (see Fact 10.1) and hence can be efficiently evaluated using Gaussian elimination for any input x. We say that such a μ has an *efficient evaluation oracle* and, as it turns out, this is the *only* property we need from DPPs and our results generalize to any measure μ for which we have such an evaluation oracle. Before we explain our results, we formally introduce the sampling problem in this general framework.

Definition 10.1 (*Sampling*)

Let $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ be a function assigning non-negative real values to subsets of $[m]$ and let $\mathcal{C} \subseteq 2^{[m]}$ be any family of subsets of $[m]$. We denote the (sampling) problem of selecting a set $S \in \mathcal{C}$ with probability $p_S = \frac{\mu(S)}{\sum_{T\in\mathcal{C}} \mu(T)}$ by $\textsc{Sample}[\mu, \mathcal{C}]$.

Building up on the equivalence between sampling and counting [JVV86], we show that if one is given oracle access to the generating polynomial g_μ and if μ is a nonnegative measure, the problem $\textsc{Sample}[\mu, \mathcal{C}]$ is essentially equivalent to the following counting problem; see Theorem 10.5 in Section 10.2.6.

Definition 10.2 (*Counting*)

Let $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ be a function assigning non-negative real values to subsets of $[m]$ and let $\mathcal{C} \subseteq 2^{[m]}$ be any family of subsets of $[m]$. We denote the (counting) problem of computing the sum $\sum_{S\in\mathcal{C}} \mu(S)$ by $\textsc{Count}[\mu, \mathcal{C}]$.

In particular, a polynomial time algorithm for $\textsc{Count}[\mu, \mathcal{C}]$ can be translated into a polynomial time algorithm for $\textsc{Sample}[\mu, \mathcal{C}]$. Interestingly, this relation holds no matter what $\mathcal{C}$ is; in particular, no specific assumptions on how the access to $\mathcal{C}$ is provided are required.

Towards developing counting algorithms in our framework, we focus on a class of families $\mathcal{C} \subseteq 2^{[m]}$, which we call *Budget Constrained Families*, where a cost vector $c \in \mathbb{Z}^m$ and a budget value $C \in \mathbb{Z}$ are given, and the family consists of all sets $S \subseteq [m]$ of total cost $c(S) \overset{\text{def}}{=} \sum_{i\in S} c_i$ at most C. We call the counting and sampling problems for this special case $\text{B}\textsc{Count}[\mu, c, C]$ and $\text{B}\textsc{Sample}[\mu, c, C]$ respectively.

10.1.2 Statement of Results

Our key result is that the BCount problem (and hence also BSample) is efficiently solvable whenever the costs are not too large in magnitude.

Theorem 10.1 (*Counting under Budget Constraints*)

There is an algorithm, which given a function $\mu : 2^{[m]} \to \mathbb{R}$ (via oracle access to g_μ), a cost vector $c \in \mathbb{Z}^m$ and a cost value $C \in \mathbb{Z}$ solves the $\text{B}\textsc{Count}[\mu, c, C]$ *problem in polynomial time with respect to m and $\|c\|_1$.*

The proof of Theorem 10.1 (see Section 10.2.1) benefits from an interplay between probability measures and polynomials. It reduces the counting problem to computing the coefficients of a certain univariate polynomial which, in turn, can be evaluated efficiently given access to the generating polynomial for μ. We can then employ interpolation in order to recover the required coefficients.

It is not hard to see that Theorem 10.1 also implies the same result for families with a single *equality* constraint ($c(S) = C$) or for any constraint of the form $c(S) \in K$, where $K \subseteq \mathbb{Z}$ is given as input together with $c \in \mathbb{Z}^m$ and $C \in \mathbb{Z}$. Furthermore, our framework can be easily extended to the case of multiple (constant number of) such constraints.

As mentioned earlier, what makes DPPs attractive is that their generating polynomial, arising from a determinant, is efficiently computable. Using this fact, Theorem 10.1 and the equivalence between sampling and counting, we can deduce the following result.

Corollary 10.1

There is an algorithm, which given a PSD matrix $L \in \mathbb{R}^{m \times m}$, a cost vector $c \in \mathbb{Z}^m$ and a cost value $C \in \mathbb{Z}$ samples a set S of cost $c(S) \leqslant C$ with probability proportional to $\det(L_{S,S})$. The running time of the algorithm is polynomial with respect to m and $\|c\|_1$.

From the above one can derive efficient sampling algorithms for several classes of constraint families $\mathcal{C}$ which have *succinct descriptions*. Indeed, we establish counting and sampling algorithms for a general class of *linear families* of the form

$$\mathcal{C} = \{S \subseteq [m] : c_1(S) \in K_1, c_2(S) \in K_2, \dots, c_p(S) \in K_p\} \tag{10.1}$$

where $c_1, c_2, \dots, c_p \in \mathbb{Z}^m$ and $K_1, \dots, K_p \subseteq \mathbb{Z}$. We prove the following

Corollary 10.2

There is an algorithm, which given a PSD matrix $L \in \mathbb{R}^{m \times m}$ and a description of a linear family $\mathcal{C}$ as in (10.1), samples a set $S \in \mathcal{C}$ with probability proportional to $\det(L_{S,S})$. The running time of the algorithm is polynomial in m and $\prod_{j=1}^{p} (\|c_j\|_1 + 1)$.

One particular class of families for which the above yields polynomial time sampling algorithms are partition families (families of bases of partition matroids) over constantly many parts (see Corollary 10.4). An important open problem that remains is to come up with even faster algorithms.

Another application of Theorem 10.1, which we present in Section 10.2.4, is to combinatorial sampling and counting problems. More precisely, we note that the indicator measure of bases of regular matroids has an efficiently computable generating polynomial; hence, we can solve their corresponding budgeted versions of counting and sampling problems.

One may ask if the dependence on $\|c\|_1$ in Theorem 10.1 can be improved. We prove that the answer to this question is no in a very strong sense. To state our hardness result, we introduce ECount – a natural variant of the BCount problem – in which the sum is over subsets of cost equal to a given value C instead of at most C (such a problem is no harder than BCount). We provide an approximation preserving reduction showing that ECount$[\mu, c, C]$ is at least as hard as computing *mixed discriminants* of tuples of positive semidefinite (PSD) matrices when c and C are given in binary, and can be exponentially large in magnitude. Recall that for a tuple of $m \times m$ PSD matrices $A_1, \dots, A_m$, their mixed discriminant is the coefficient of the monomial $\prod_{i=1}^{m} x_i$ in the polynomial $\det(\sum_{i=1}^{m} x_i A_i)$.

Theorem 10.2 (*Hardness of Counting under Budget Constraints*)

BCount$[\mu, c, C]$ *is* $\#\mathbf{P}$*−hard. Moreover, when* μ *is a determinantal function,* ECount$[\mu, c, C]$ *is at least as hard to approximate as mixed discriminants of tuples of PSD matrices.*

To prove this result we show an *equivalence* between the counting problem corresponding to partition-constrained DPPs (with a large, super-constant number of parts) and computing mixed discriminants. Unlike permanents [JSV04], no efficient approximation scheme is known for estimating mixed discriminants and there is some evidence [Gur05] that there may be none. To further understand to what extent g_μ is the cause of computational hardness, in Section 10.2.5

(see Theorem 10.4) we provide another hardness result; it considers a μ that is a 0/1 indicator function for spanning trees in a graph (with efficiently computable g_μ). We prove that $\textsc{ECount}[\mu, c, C]$ is at least as hard to approximate as the number of perfect matchings in general (non-bipartite) graphs, which is another problem for which existence of an FPRAS is open.

10.2 Proofs

10.2.1 Counting with Budget Constraints

Proof of Theorem 10.1:
Let us first consider the case in which the cost vector c is nonnegative, i.e., $c \in \mathbb{N}^m$. We introduce a new variable z and consider the polynomial

$$h(z) \overset{\text{def}}{=} g_\mu(z^{c_1}, z^{c_2}, \ldots, z^{c_m}).$$

Since $g_\mu(x_1, \ldots, x_m) = \sum_{S\subseteq[m]} \mu(S) \prod_{i\in S} x_i$, we have

$$h(z) = \sum_{S\subseteq[m]} \mu(S) \prod_{i\in S} z^{c_i} = \sum_{S\subseteq[m]} \mu(S) z^{c(S)} = \sum_{0\leqslant d\leqslant \|c\|_1} z^d \sum_{S:\ c(S)=d} \mu(S).$$

Hence, the coefficient of z^d in $h(z)$ is equal to the sum of $\mu(S)$ over all sets S such that $c(S) = d$. In particular, the output is the sum of coefficients over $d \leqslant C$.

It remains to show how to compute the coefficients of h. Note that we do not have direct access to g_μ. However, we can evaluate $g_\mu(x)$ at any input $x \in \mathbb{R}^m$, which in turn allows us to compute $h(z)$ for any input $z \in \mathbb{R}$. Since $h(z)$ is a polynomial of degree at most $\|c\|_1$, in order to recover the coefficients of h, it suffices to evaluate it at $\|c\|_1 + 1$ inputs and perform interpolation. When using FFT, the total running time becomes:

$$(\|c\|_1 + 1) \cdot T_\mu + \widetilde{O}(\|c\|_1),$$

where T_μ is the running time of the evaluation oracle for g_μ.

In order to deal with the case in which c has negative entries, consider a modified version of h:

$$h(z) \overset{\text{def}}{=} z^{\|c\|_1} g_\mu(z^{c_1}, z^{c_2}, \ldots, z^{c_m}).$$

Clearly, $h(z)$ is a polynomial of degree at most $2 \cdot \|c\|_1$ whose coefficients encode the desired output. □

Remark 10.1

Note that the bit complexity of the output of the proposed algorithm is polynomial in the input size since it is a result of solving a linear system with all the coefficients being polynomially bounded.

We also state a simple consequence of the above proof that is often convenient to work with.

Corollary 10.3

There is an algorithm that, given a vector $c \in \mathbb{Z}^m$, a value $C \in \mathbb{Z}$ and oracle access to g_μ computes the sum $\sum_{S:\ c(S)=C} \mu(S)$ in time polynomial with respect to m and $\|c\|_1$.

In the above, note the equality $c(S) = C$ instead of $c(S) \leqslant C$ as in BCount.

10.2.2 Counting for Constrained DPPs

In this section we show how our general result in Theorem 10.1 implies algorithms for sampling from Determinantal Point Processes.

A Determinantal Point Process (DPP) is a probability distribution μ over subsets of $[m]$ defined with respect to a symmetric positive semidefinite matrix $L \in \mathbb{R}^{m\times m}$ by $\mu(S) \propto \det(L_{S,S})$; i.e.,

$$\mu(S) \overset{\text{def}}{=} \frac{\det(L_{S,S})}{\sum_{T\subseteq[m]} \det(L_{T,T})}.$$

We will often use a different matrix to represent the measure μ; let $V \in \mathbb{R}^{m\times n}$ be a matrix, such that $L = VV^\top$ (the Cholesky decomposition of L). Then, $\det(L_{S,S}) = \det(V_S V_S^\top)$.

An important open problem related to DPPs is the sampling problem under additional combinatorial constraints imposed on the ground set $[m]$. We prove that these problems are polynomial time solvable for succinct budget constraints, as in Theorem 10.1. We start by establishing the fact that generating polynomials for determinantal distributions are efficiently computable.

Fact 10.1

Let $L \in \mathbb{R}^{m\times m}$ be a PSD matrix with $L = VV^\top$ for some $V \in \mathbb{R}^{m\times n}$. If $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ is defined as $\mu(S) \overset{\text{def}}{=} \det(L_{S,S})$ then $\det(V^\top XV + I) = \sum_{S\subseteq[m]} x^S \mu(S)$, where X is the diagonal matrix of indeterminates $X = \operatorname{Diag}(x_1, \ldots, x_m)$ and I is the $n \times n$ identity matrix.

Proof.
We start by applying the Sylvester's determinant identity

$$\det(V^\top XV + I) = \det\left(\left(\sqrt{X}V\right)\left(\sqrt{X}V\right)^\top + I\right).$$

It is well known that for a symmetric matrix $A \in \mathbb{R}^{m\times n}$ the coefficient of t^k in the polynomial $\det(A + tI)$ is equal to $\sum_{|S|=n-k} \det(A_{S,S})$. Applying this result to $A = \left(\sqrt{X}V\right)\left(\sqrt{X}V\right)^\top$, we get

$$\det(A_{S,S}) = x^S \det(V_S V_S^\top) = x^S \det(L_{S,S}),$$

which concludes the proof by simply taking $t = 1$. □

Now we are ready to deduce Corollary 10.1.

Proof of Corollary 10.1:
A polynomial time counting algorithm follows directly from Theorem 10.1 and Fact 10.1. To deduce sampling we apply the result on equivalence between sampling and counting Theorem 10.5. In fact when applied to an exact counting algorithm we obtain an exact sampling procedure. □

We move to the general result on sampling for linear families – Corollary 10.2. One can deduce it directly from Theorem 10.1, but this leads to a significantly suboptimal algorithm. Instead we take a different path and reprove Theorem 10.1 in a slightly higher generality.

Proof of Corollary 10.2:
We will show how to solve the counting problem – sampling will then follow from Theorem 10.5. Also, for simplicity we assume that all the entries in the cost vectors are nonnegative, this can be extended to the general setting as in the proof of Theorem 10.1.

Let g be the generating polynomial of the determinantal function $\mu(S) = \det(L_{S,S})$, which is efficiently computable by Fact 10.1. For notational clarity we will use superscripts to index constraints. For every constraint "$c^{(j)}(S) \in K_i$" ($j = 1, 2, \ldots, p$) introduce a new formal variable y_j. For every index $i \in [m]$ define the monomial:

$$s_i = \textstyle\prod_{j=1}^{p} y_j^{c_i^{(j)}}.$$

The above encodes the cost of element i with respect to all cost vectors $c^{(j)}$ for $j = 1, 2, \ldots, p$. Consider the polynomial $h(y_1, \ldots, y_p) = g(s_1, s_2, \ldots, s_m)$. It is not hard to see that the coefficient of a given monomial $\prod_{j=1}^{p} y_j^{d_j}$ in h is simply the sum of $\mu(S)$ over all sets S satisfying $c^{(1)}(S) = d_1, c^{(2)}(S) = d_2, \ldots, c^{(p)}(S) = d_p$. Hence the solution to our counting problem is simply the sum of certain coefficients of h. It remains to show how to recover all the coefficients efficiently.

Note that we can efficiently evaluate the polynomial h at every input $(y_1, \ldots, y_p) \in \mathbb{R}^p$. One can then apply interpolation to recover all coefficients of h. The running time is polynomial in the total number of monomials in h (this is the number of variables of a linear system which can be used to find the coefficients), which can be bounded from above by $\prod_{j=1}^{p} \left(\|c^{(j)}\|_1 + 1\right)$. $\square$

We derive now one interesting application of Corollary 10.2 – sampling from partition constrained DPPs. Let us first define *partition families* formally.

Definition 10.3

Let $[m] = P_1 \cup P_2 \cup \cdots \cup P_p$ be a partition of $[m]$ into disjoint, nonempty sets and let $b_1, b_2, \ldots, b_p$ be integers such that $0 \leqslant b_i \leqslant |P_i|$. A family of sets of the form

$$\mathcal{C} = \{S \subseteq [m] : |S \cap P_j| = b_j, \text{ for every } j = 1, 2, \ldots, p\}$$

is called a partition family.

We prove the following consequence of Corollary 10.2, which asserts that polynomial time counting and sampling is possible for DPPs under partition constraints for constant p.

Corollary 10.4

Given a DPP defined by $L \in \mathbb{R}^{m \times m}$ and a partition family $\mathcal{C}$ with a constant number of parts, there exists a polynomial time sampling algorithm for the distribution

$$\mu_{\mathcal{C}}(S) \stackrel{\text{def}}{=} \frac{\det(L_{S,S})}{\sum_{T \in \mathcal{C}} \det(L_{T,T})} \qquad \text{for } S \in \mathcal{C}.$$

Proof.
In light of Corollary 10.2 it suffices to show that every partition family has a succinct representation as a linear family. We show that it is indeed the case. Consider a partition family $\mathcal{C}$ induced by the partition $P_1 \cup P_2 \cup \ldots \cup P_p = [m]$ and numbers $b_1, b_2, \ldots, b_p$. Define the following cost vectors: $c_j = 1_{P_j}$, for $j = 1, 2, \ldots, p$, i.e., the indicator vectors of the sets $P_1, P_2, \ldots, P_p$.

Moreover define K_j to be $\{b_j\}$ for every $j = 1, 2, \dots, p$. It is then easy to see that "$c_j(S) \in K_j$" is implementing the constraint $|P_j \cap S| = b_j$. In other words the family $\mathcal{C}$ is equal to the linear family defined by cost vectors $c_1, c_2, \dots, c_p$ and sets $K_1, K_2, \dots, K_p$. It remains to observe that $\|c_j\|_1 = |P_j| \leqslant m$ and hence $\prod_{j=1}^{p}(\|c_j\| + 1) = O(m^p)$. Since $p = O(1)$ the algorithm from Corollary 10.2 runs in polynomial time. □

10.2.3 Hardness Result

In this section we study hardness of BCount$[\mu, c, C]$. Theorem 10.1 implies that BCount is polynomial time solvable whenever we measure the complexity with respect to the unary encoding length of the cost vector c. Here we prove that if c is given in binary, the problem becomes $\#\mathbf{P}-$hard. Moreover, existence of an efficient approximation scheme for a closely related problem (instead of counting all objects of cost *at most* C, count objects of cost *exactly* C) would imply existence of such schemes for counting perfect matchings in non-bipartite graphs (see Section 10.2.5) and for computing mixed discriminants. In both cases, these are notorious open questions and the latter is believed to be unlikely.

Mixed Discriminants

We relate the BCount problem to the well studied problem of computing mixed discriminants of PSD matrices and prove Theorem 10.2. Recall the definition:

Definition 10.4

Let $A_1, A_2, \dots, A_m \in \mathbb{R}^{d \times d}$ be symmetric matrices of dimension d. The mixed discriminant of a tuple $(A_1, A_2, \dots, A_d)$ is defined as

$$D(A_1, A_2, \dots, A_d) \overset{\text{def}}{=} \frac{\partial^d}{\partial z_1 \dots \partial z_d} \det(z_1 A_1 + z_2 A_2 + \cdots + z_d A_d).$$

Computing mixed discriminants of PSD matrices is known to be $\#\mathbf{P}$-hard, since they can encode the permanent. However, as opposed to the permanent, there is no FPRAS known for computing mixed discriminants, and the best polynomial time approximation algorithms by [Bar97, GS02] have an exponentially large approximation ratio.

The main technical component in our proof of Theorem 10.2 is the following lemma.

Lemma 10.1

There is a polynomial time reduction, which given a tuple $(A_1, \dots, A_n)$ of PSD $n \times n$ matrices outputs a PSD matrix $L \in \mathbb{R}^{m \times m}$, a cost vector $c \in \mathbb{Z}^m$ and a cost value $C \in \mathbb{Z}$ such that

$$n! \cdot D(A_1, A_2, \dots, A_n) = \sum_{S \subseteq [m],\ c(S) = C} \mu(S),$$

where $\mu(S) = \det(L_{S,S})$, for $S \subseteq [m]$. Moreover, $\|c\|_1 \leqslant 2^{O(n \log n)}$.

Before proving Lemma 10.1 let us first state several important properties of mixed discriminants, which we will rely on; for proofs of these facts we refer the reader to [Bap89].

Fact 10.2 (*Properties of Mixed Discriminants*)

Let $A, B, A_1, A_2, \ldots, A_n$ be symmetric $n \times n$ matrices.

1. *D is symmetric, i.e.,*

$$D(A_1, A_2, \ldots, A_n) = D(A_{\sigma(1)}, A_{\sigma(2)}, \ldots, A_{\sigma(n)}), \text{ for any permutation } \sigma \in S_n.$$

2. *D is linear with respect to every coordinate, i.e.,*

$$D(\alpha A + \beta B, A_2, \ldots, A_n) = \alpha D(A, A_2, \ldots, A_n) + \beta D(B, A_2, \ldots, A_n).$$

3. *If $A = \sum_{i=1}^n v_i v_i^\top \in \mathbb{R}^{n\times n}$ then we have: $\det(A) = n!\ D(v_1 v_1^\top, \ldots, v_n v_n^\top)$.*

Proof of Lemma 10.1:
Consider a tuple $(A_1, A_2, \ldots, A_n)$ of PSD matrices. The first step is to decompose them into rank-one summands:

$$A_i = \sum_{j=1}^{r} v_{i,j} v_{i,j}^\top,$$

where $v_{i,j} \in \mathbb{R}^n$ for $1 \leqslant i, j \leqslant n$ (some $v_{i,j}$'s can be zero if $\operatorname{rank}(A_i) < n$). This step can be performed using the Cholesky decomposition.

Let $M = \{(i,j) : 1 \leqslant i, j \leqslant n\}$ and for every $i = 1, 2, \ldots, n$ define $P_i = \{i\} \times [n]$. We take $m = |M| = n^2$ and define a family $\mathcal{C}$ of $n-$subsets of M to be

$$\mathcal{C} = \{S \subseteq [m] : |S \cap P_i| = 1 \text{ for every } i = 1, 2, \ldots, n\}.$$

Let V denote an $m \times n$ matrix with rows indexed by M, for which the eth row is v_e as above ($e \in M$, i.e., $e = (i,j)$ for some $i, j \in [n]$). We also set $L = VV^\top$, hence L is an $m\times m$ symmetric, PSD matrix. Finally, let $\mu(S) = \det(L_{S,S})$. Note that for sets S of cardinality n we have

$$\mu(S) = \det(L_{S,S}) = \det(V_S V_S^\top) = \det(V_S^\top V_S) = \det\left(\sum_{e\in S} v_e v_e^\top\right).$$

In the calculation below we rely on properties of mixed discriminants listed in Fact 10.2 and on the fact that $|S| = n$ for $S \in \mathcal{C}$.

$$\begin{aligned}
D(A_1, A_2, \ldots, A_n) &= D\left(\sum_{j=1}^n v_{1,j} v_{1,j}^\top, \sum_{j=1}^n v_{2,j} v_{2,j}^\top, \ldots, \sum_{j=1}^n v_{n,j} v_{n,j}^\top\right) \\
&= \sum_{1\leqslant j_1, j_2, \ldots, j_n \leqslant n} D(v_{1,j_1} v_{1,j_1}^\top, v_{2,j_2} v_{2,j_2}^\top, \ldots, v_{n,j_n} v_{n,j_n}^\top) \\
&= \sum_{e_1\in P_1, e_2 \in P_2, \ldots, e_n \in P_n} D(v_{e_1} v_{e_1}^\top, v_{e_2} v_{e_2}^\top, \ldots, v_{e_n} v_{e_n}^\top) \\
&= \sum_{\{e_1, e_2, \ldots, e_n\}\in\mathcal{C}} \frac{1}{n!} \det(v_{e_1} v_{e_1}^\top + v_{e_2} v_{e_2}^\top + \ldots + v_{e_n} v_{e_n}^\top) = \frac{1}{n!} \sum_{S\in\mathcal{C}} \mu(S).
\end{aligned}$$

It remains to show that the partition family $\mathcal{C}$ can be represented as $\mathcal{C} = \{S \subseteq M : c(S) = C\}$ for some cost vector $c \in \mathbb{Z}^M$ and $C \in \mathbb{Z}$, such that $\|c\|_1 = 2^{O(n \log n)}$. Indeed, by a reasoning as in Corollary 10.4 we can represent $\mathcal{C}$ as a linear family with n constraints of the form $c^{(i)}(S) = 1$ for $i = 1, 2, \ldots, n$ and $c^{(i)} \in \{0,1\}^{n \times n}$. It is not hard to see that these can be combined into one constraint $c(S) = C$ with $\|c\|_1 = (n^2)^{n+O(1)} = 2^{O(n \log n)}$. Now, it remains to observe that all the steps of the reduction are efficient (since the cost vector is represented in binary here). $\square$

Proof of Theorem 10.2:
In light of Lemma 10.1, the problem of computing $\sum_{S \subseteq [m], c(S)=C} \mu(S)$ for determinantal functions μ is at least as hard as computing mixed discriminants. The BCount problem is very similar, with the only difference that it is computing the sum over all sets of cost $c(S)$ at most C. However, clearly by solving the BCount problem for C and $C-1$ one can compute $\sum_{S \subseteq [m], c(S)=C} \mu(S)$ by just subtracting the obtained results. $\square$

10.2.4 Budget-Constrained Sampling and Counting for Regular Matroids

Consider the following problem: given an undirected graph G with weights $c \in \mathbb{R}^m$ on its edges, sample a uniformly random spanning tree of cost at most C in G. This generalizes the problem of sampling uniformly random spanning trees [Pem04] and sampling a random spanning tree of minimum cost [Epp95]. Below we study the generalized version of this problem by considering regular matroids, indeed spanning trees arise as bases of the graphic matroid, which is known to be regular. We prove that the counting and sampling problem in this setting can be solved efficiently whenever c is polynomially bounded.

Theorem 10.3 (*Counting and Sampling Bases of Matroids*)

Let $\mathcal{M}$ be a regular matroid on a ground set $[m]$ with a set of bases $\mathcal{B}$. There exists a counting algorithm which, given a cost vector $c \in \mathbb{Z}^m$ and a value $C \in \mathbb{Z}$, outputs the cardinality of the set $\{S \in \mathcal{B} : c(S) \leqslant C\}$ and a sampling algorithm which, given a cost vector $c \in \mathbb{Z}^m$ and a value $C \in \mathbb{Z}$, outputs a random element in the set $\{S \in \mathcal{B} : c(S) \leqslant C\}$. The running time of both algorithms is polynomial in m and $\|c\|_1$.

Proof of Theorem 10.3:
Let $\mathcal{M} \subseteq 2^{[m]}$ be a regular matroid and $\mathcal{B} \subseteq 2^{[m]}$ be its set of bases. We prove that the generating polynomial $\sum_{S \in \mathcal{B}} x^S$ is efficiently computable. We use the characterization of regular matroids as those which can be linearly represented by a totally unimodular matrix. In other words, there exists a totally unimodular matrix $A \in \mathbb{Z}^{m \times d}$ such that if we denote by $A_e \in \mathbb{Z}^d$ the e^{th} row of A it holds that:

$$S \in \mathcal{M} \quad \Leftrightarrow \quad \{A_e : e \in S\} \text{ is linearly independent.} \tag{10.2}$$

Let $r \leqslant d$ be the rank of the matroid $\mathcal{M}$, i.e., the cardinality of any set in $\mathcal{B}$. We claim that without loss of generality one can assume that $d = r$. Indeed, we prove that there is a submatrix $A' \in \mathbb{Z}^{m \times r}$ of A, such that (10.2) still holds with A replaced by A'. To this end suppose that $d > r$. It is easy to see that the rank of A is r, otherwise, by (10.2) there would be a set S of cardinality at least $r+1$ with $S \in \mathcal{M}$. Hence there is a column in A which is a linear combination of the remaining columns, we can freely remove this column from A, while (10.2) will be still true. By doing so, we finally obtain a matrix A' with exactly r rows, which satisfies (10.2).

By the fact that A has r columns we have:

$$S \in \mathcal{B} \quad \Leftrightarrow \quad A_S \text{ is nonsingular},$$

where by A_S we mean the $|S| \times r$ submatrix of A corresponding to rows from S. In particular, for a set $S \subseteq [m]$ of cardinality r we have:

$$S \in \mathcal{B} \quad \Leftrightarrow \quad \det(A_S) \neq 0 \quad \Leftrightarrow \quad \det(A_S^\top A_S) = 1,$$

where the last equivalence follows from A being totally unimodular. Let us now consider the polynomial

$$g(x_1, x_2, \ldots, x_m) = \det\left(\sum_{e=1}^{m} x_e A_e A_e^\top\right).$$

By the Cauchy-Binet theorem we obtain:

$$g(x_1, x_2, \ldots, x_m) = \sum_{|S|=r} \det\left(\sum_{e \in S} x_e A_e A_e^\top\right) = x^S \det(A_S^\top A_S).$$

In other words, g is equal to g_μ – the generating polynomial of the function $\mu : 2^{[m]} \to \mathbb{R}$ given by

$$\mu(S) = \begin{cases} 1 & \text{if } S \in \mathcal{B} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, since g_μ is efficiently computable, by Theorem 10.1 the BCount$[\mu, c, C]$ is efficiently solvable. This fact, together with Theorem 10.5 imply that sampling also can be made efficient. □

10.2.5 Hardness for Spanning Trees

We show that BCount is at least as hard as counting perfect matchings in a non-bipartite graph. The proof relies on a combinatorial reduction from counting perfect matchings in a graph to counting budget constrained spanning trees.

Theorem 10.4

There is a polynomial time reduction which given a graph $G = (V, E)$ with n vertices and m edges outputs a graph G' with n vertices and $O(m + n^2)$ edges, a cost vector $c \in \mathbb{N}^m$ with $\|c\|_1 \leqslant 2^{O(m \log m)}$ and a value $C \in \mathbb{N}$, such that:

$$PM(G) = \alpha \cdot ST_C(G')$$

where $PM(G)$ denotes the number of perfect matchings in G, $ST_C(G')$ denotes the number of spanning trees of total cost C in G' and $\alpha = \frac{n^2}{2}(2n)^{-n/2}$.

Proof.
Let $G = (V, E)$ be an undirected graph, let $n = |V|$ and $m = |E|$. We construct a new graph G' and a cost vector c, such that counting perfect matchings in G is equivalent to counting spanning trees of specified cost C in G' .

The graph $G' = (V, E')$ is obtained by adding a complete graph to G, i.e., $\binom{n}{2}$ edges, one between every pair of vertices. We call the set of new edges F, hence $E' = E \cup F$. Note that E' is a multiset. To all edges $e \in F$ we assign cost $c_e = 0$, while for the original edges the costs are positive and defined below.

Let $b = m' + 1$, where $m' = |E'|$ is the number of edges in G'. We define the cost of an edge $e = ij \in E$ to be:

$$c_e = b^i + b^j.$$

Note that from the choice of b and c it follows that given a cost $c(S)$ of some set $S \subseteq E$, we can exactly compute how many times a given vertex appears as an endpoint of an edge in S. Indeed, if we have:

$$c(S) = \sum_{i=1}^{n} \delta_i b^i$$

such that $0 \leqslant \delta_i \leqslant b - 1$ (the $b-$ary representation of $c(S)$), then the degree of vertex i in S is δ_i. This follows from the fact that b is chosen to avoid carry overs when computing $c(S)$ in the $b-$ary numerical system. Therefore, it is now a natural choice to define $C \overset{\text{def}}{=} \sum_{i=1}^{n} b^i$. We claim that every perfect matching in G corresponds to exactly $\alpha = \frac{n^2}{2}(2n)^{-n/2}$ different spanning trees of cost C in G'.

To prove this claim, fix any spanning tree S of cost $c(S) = C$. Note first that we have $c(S \cap E) = c(S)$ because all of the edges $e \notin E$ have cost 0. Moreover, the set $M \overset{\text{def}}{=} S \cap E$ is a perfect matching in G, because $c(M) = C$ implies that the degree of every vertex in M is one. It remains to show that every perfect matching M in G corresponds to exactly α spanning trees of cost C in G.

Fix any perfect matching M_0 in G. We need to calculate how many ways are there to add $\frac{n}{2} - 1$ edges from E' to obtain a spanning tree of G'. By contracting the matching M_0 to $\frac{n}{2}$ vertices and considering edges in E' only, we obtain a complete graph on $\frac{n}{2}$ vertices with 4 parallel edges going between every pair of vertices. The answer is the number of spanning trees of the obtained graph. Cayley's formula easily implies that this number is $4^{\frac{n}{2}-1}\left(\frac{n}{2}\right)^{\frac{n}{2}-2}$ which equals α^{-1}. $\square$

10.2.6 Equivalence Between Counting and Sampling

In this section we state and prove a theorem that implies that the COUNT$[\mu, \mathcal{C}]$ and SAMPLE$[\mu, \mathcal{C}]$ problems are essentially equivalent. We prove that, for a given type of constraints $\mathcal{C}$, a polynomial time algorithm for counting can be transformed into a polynomial time algorithm for sampling and vice versa. This section follows the convention that $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ is any function that assigns nonnegative values to subsets of $[m]$ and $\mathcal{C} \subseteq 2^{[m]}$ is any family of subsets of $[m]$.

Theorem 10.5 (*Equivalence Between Approx. Counting and Approx. Sampling*)

Consider any function $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ and a family $\mathcal{C}$ of subsets of $[m]$. Let $\mu_{\mathcal{C}} : \mathcal{C} \to [0,1]$ be a distribution over $S \in \mathcal{C}$ such that $\mu_{\mathcal{C}}(S) \propto \mu(S)$. We assume evaluation oracle access to the generating polynomial g_μ of μ, and define the following two problems:

- **Approximate $\mathcal{C}$-sampling:** *given a precision parameter $\varepsilon > 0$, provide a sample S from a distribution $\rho : \mathcal{C} \to [0,1]$ such that $\|\mu_{\mathcal{C}} - \rho\|_1 < \varepsilon$.*

- **Approximate $\mathcal{C}$-counting:** *given a precision parameter $\varepsilon > 0$, output a number $X \in \mathbb{R}$ such that $X(1+\varepsilon)^{-1} \leqslant \sum_{S\in\mathcal{C}} \mu(S) \leqslant X(1+\varepsilon)$.*

The time complexities of the above problems differ by at most a multiplicative factor of $\mathrm{poly}(m, \varepsilon^{-1})$.

Remark 10.2

The above theorem establishes equivalence between approximate *variants of* Count$[\mu, \mathcal{C}]$ *and* Sample$[\mu, \mathcal{C}]$. *This is convenient for applications, because the exact counting variants of these problems are often $\#\mathbf{P}-$hard. Still, for some of them, efficient approximation schemes are likely to exist. Further, we mention that the implication from exact counting to exact sampling holds, hence the sampling algorithms that we obtain in this chapter are exact.*

Theorem 10.5 follows from a self-reducibility property [JVV86] of the counting problem. Before we present the proof of Theorem 10.5, we introduce some terminology and state assumptions for the remaining part of this section. The function $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$ is given as an evaluation oracle for $g_\mu(x) = \sum_{S\subseteq[m]} \mu(S)x^S$. In particular, we measure complexity with respect to the number of calls to such an oracle. An algorithm which, for a fixed family $\mathcal{C} \subseteq 2^{[m]}$ and every function μ, given access to g_μ computes $\sum_{S\in\mathcal{C}} \mu(S)$ is called a $\mathcal{C}$-counting oracle. Similarly, we define a $\mathcal{C}$-sampling oracle to be an algorithm which, given access to g_μ, provides samples from the distribution

$$\mu_{\mathcal{C}}(S) \stackrel{\text{def}}{=} \frac{\mu(S)}{\sum_{T\in\mathcal{C}} \mu(T)} \qquad \text{for } S \in \mathcal{C}.$$

Counting Implies Sampling

We now show how counting implies sampling. It proceeds by inductively conditioning on certain elements not being in the sample. For this idea to work one has to implement conditioning using the $\mathcal{C}-$sampling oracle and access to the generating polynomial only. Below we state the implication from counting to sampling in the exact variant. The approximate variant also holds, with an analogous proof.

Lemma 10.2 (*Counting Implies Sampling*)

Let $\mathcal{C}$ denote a family of subsets of $[m]$. Suppose access to a $\mathcal{C}$-counting oracle is given. Then, there exists a $\mathcal{C}$-sampling oracle which, for any function $\mu : 2^{[m]} \to \mathbb{R}_{\geqslant 0}$, makes $\mathrm{poly}(m)$ *calls to the counting oracle and to g_μ and outputs a sample from the distribution $\mu_{\mathcal{C}}$.*

Proof.
Let $\mathbf{S}$ be the random variable corresponding to the sample our algorithm outputs; our goal is to have $\mathbf{S} \sim \mu_{\mathcal{C}}$. The sampling algorithm proceeds as follows: It sequentially considers each element

$e \in [m]$ and tries to decide (at random) whether to include $e \in \mathbf{S}$ or not. To do so, it first computes the probability $\mathbb{P}(e \in \mathbf{S})$ *conditioned on all decisions thus far*. It then flips a biased coin with this probability, and includes e in $\mathbf{S}$ according to its outcome. More formally, the sampling algorithm can be described as follows:

1. Input: $V \in \mathbb{R}^{m\times r}$, a number $k \leqslant r$.
2. Initialize: $Y = \emptyset$, $N = \emptyset$.
3. For $e = 1, 2, \ldots, m$:
   (a) Compute the probability $p = \mathbb{P}(e \in \mathbf{S} : Y \subseteq \mathbf{S}, N \cap \mathbf{S} = \emptyset)$ under the distribution $\mathbf{S} \sim \mu_{\mathcal{C}}$.
   (b) Toss a biased coin with success probability p. In case of success add e to the set Y, otherwise add e to N.
4. Output: $\mathbf{S} = Y$.

It is clear that the above algorithm correctly samples from $\mu_{\mathcal{C}}$. It remains to show that $\mathbb{P}(e \in S : Y \subseteq S, N \cap S = \emptyset)$ can be computed efficiently. This follows from Lemma 10.3 below. ☐

Lemma 10.3

Let Y and N be disjoint subsets of $[m]$ and consider any $e \in [m]$. Suppose $\mathbf{S}$ is distributed according to $\mu_{\mathcal{C}}$. If we are given access to a $\mathcal{C}$-counting oracle and to g_μ, then $\mathbb{P}(e \in \mathbf{S} : Y \subseteq \mathbf{S}, N \cap \mathbf{S} = \emptyset)$ can be computed in poly(m) *time.*

Proof.
Assume $e \in [m] \setminus (Y \cup N)$; otherwise the probability is clearly 0 or 1. Let $Y' = Y \cup \{e\}$, then

$$\mathbb{P}(e \in \mathbf{S} : Y \subseteq \mathbf{S}, N \cap \mathbf{S} = \emptyset) = \frac{\sum_{S\in\mathcal{C}, Y'\subseteq S, N\cap S=\emptyset} \mu(S)}{\sum_{S\in\mathcal{C}, Y\subseteq S, N\cap S=\emptyset} \mu(S)}.$$

We now show how to compute such sums: Introduce a new variable y, and for every $e \in [m]$ define:

$$w_e \overset{\text{def}}{=} \begin{cases} yx_e & \text{for } e \in Y, \\ 0 & \text{for } e \in N, \\ x_e & \text{otherwise.} \end{cases}$$

We interpret the expression $g_\mu(w_1, w_2, \ldots, w_m)$ as a generating polynomial for a certain function $\mu'(y) : 2^{[m]} \to \mathbb{R}$; i.e.,

$$g_{\mu'}(x) \overset{\text{def}}{=} g_\mu(w_1, w_2, \ldots, w_m) = \sum_{S\cap N=\emptyset} y^{|S\cap Y|} x^S \mu(S).$$

Define a polynomial

$$h(y) \overset{\text{def}}{=} \sum_{S\in\mathcal{C}, S\cap N=\emptyset} y^{|S\cap Y|} \mu(S).$$

It follows that $h(y)$ is a polynomial of degree at most $|Y|$. In fact, the sum we are interested in is simply the coefficient of $y^{|Y|}$ in $h(y)$. The last thing to note is that we can compute $h(y)$ exactly by evaluating it for $|Y|+1$ different values of y and then performing interpolation. Hence, we just need to query the $\mathcal{C}$-counting oracle $(|Y|+1)$ times giving it μ' as input (for various choices of y).[1] $\square$

Sampling Implies Counting

We show the implication from sampling to counting in Theorem 10.5. Similarly as for the opposite direction we assume for simplicity that the sampling algorithm is exact, i.e., we prove the following lemma. The approximate variant holds with an analogous proof.

Lemma 10.4 (*Sampling Implies Counting*)

Let $\mathcal{C}$ denote a family of subsets of $[m]$. Suppose we have access to a $\mathcal{C}$-sampling oracle. Then, there exists a $\mathcal{C}$-counting oracle which for any input function $\mu : 2^{[m]} \to \mathbb{R}$ (given as an evaluation oracle for g_μ) and for any precision parameter $\varepsilon > 0$ makes $\mathrm{poly}(m, 1/\varepsilon)$ calls to the sampling oracle, and approximates the sum:

$$\sum_{S \in \mathcal{C}} \mu(S)$$

within a multiplicative factor of $(1+\varepsilon)$. The algorithm has failure probability exponentially small in m.

Let us first state the algorithm which we use to solve the counting problem. Later in a sequence of lemmas we explain how to implement it in polynomial time and reason about its correctness. In the description, $\mathbf{S}$ denotes a random variable distributed according to $\mu_{\mathcal{C}}$.

1. Initialize $U \stackrel{\text{def}}{=} [m]$, $X \stackrel{\text{def}}{=} 1$.
2. Repeat
   (a) Estimate the probability $\mathbb{P}(\mathbf{S} = U : \mathbf{S} \subseteq U)$, if it is larger than $(1 - \frac{1}{m})$, terminate the loop.
   (b) Find an element $e \in U$ so that $\mathbb{P}(e \notin \mathbf{S} : \mathbf{S} \subseteq U) \geqslant \frac{1}{m^2}$.
   (c) Approximate $p_e \stackrel{\text{def}}{=} \mathbb{P}(e \notin \mathbf{S} : \mathbf{S} \subseteq U)$ up to a multiplicative factor $\frac{\varepsilon}{m}$.
   (d) Update $X \stackrel{\text{def}}{=} X \cdot \rho_e$, where ρ_e is the estimate for p_e.
   (e) Remove e from U, i.e., set $U \stackrel{\text{def}}{=} U \setminus \{e\}$.
3. Return $X \cdot \mu(U)$.

Lemma 10.5

Given $U \subseteq [m]$ and $e \in U$, assuming access to a $\mathcal{C}$-sampling oracle, we can approximate the quantity

$$p_e = \mathbb{P}(e \notin \mathbf{S} : \mathbf{S} \subseteq U)$$

where $\mathbf{S}$ is distributed according to $\mu_{\mathcal{C}}$, up to an additive error $\delta > 0$ in time $\frac{\mathrm{poly}(m)}{\delta^2}$. The probability of failure can be made $\frac{1}{m^c}$ for any $c > 0$.

[1]The provided argument does not generalize directly to the case when the counting oracle is only approximate (because of the interpolation step). However, as we need to compute the top coefficient of a polynomial $h(y)$ only, we can alternatively do it by evaluating $h(y)$ and dividing by y^d (for $d = \deg(h)$) at a very large input $y \in \mathbb{R}$.

Proof.
We sample a set $S \in \mathcal{C}$ from the distribution $\mathbb{P}(S) \propto \mu(S)$ conditioned on $S \subseteq U$. This can be done using the sampling oracle, however instead of sampling with respect to μ one has to sample with respect to a modified function μ' which is defined as $\mu'(S) = \mu(S)$ for $S \subseteq U$ and $\mu'(S) = 0$ otherwise. Note that the generating polynomial for μ' can be easily obtained from g_μ by just plugging in zeros at positions outside of U. Given a sample S from μ' we define

$$X = \begin{cases} 1 & \text{if } e \notin S, \\ 0 & \text{otherwise.} \end{cases}$$

Repeat the above independently N times, to obtain $X_1, X_2, \ldots, X_N$ and finally compute the estimator:

$$Z = \frac{X_1 + X_2 + \cdots + X_N}{N}.$$

By Chebyshev's inequality, we have:

$$\mathbb{P}(|Z - p_e| \geqslant \delta) \leqslant \frac{1}{N\delta^2}$$

Thus, by taking $N = \frac{\mathrm{poly}(m)}{\delta^2}$ samples, with probability $\geqslant 1 - \frac{1}{\mathrm{poly}(m)}$ we can obtain an additive error of at most δ. □

Lemma 10.6

If $U \subseteq [m]$ is such that $\mathbb{P}(\mathbf{S} = U : \mathbf{S} \subseteq U) \leqslant (1 - \frac{1}{m})$ then there exists an element $e \in U$ such that $\mathbb{P}(e \notin \mathbf{S} : \mathbf{S} \subseteq U) \geqslant \frac{1}{m^2}$, where $\mathbf{S}$ is distributed according to $\mu_{\mathcal{C}}$.

Proof.
Let $\mathbf{T}$ be the random variable $\mathbf{S}$ conditioned on $\mathbf{S} \subseteq U$. Denote $q_e = \mathbb{P}(e \in \mathbf{S} : \mathbf{S} \subseteq U)$, we obtain

$$\sum_{e \in U} q_e = \mathbb{E}(|\mathbf{T}|) \leqslant \left(1 - \frac{1}{m}\right)|U| + \frac{1}{m}(|U| - 1) = |U| - \frac{1}{m}.$$

The inequality in the above expression follows from the fact that the worst case upper bound would be achieved when the probability of $|\mathbf{T}| = |U|$ is *exactly* $1 - \frac{1}{m}$ and with the remaining probability, $|\mathbf{T}| = |U| - 1$. Hence $\sum_{e \in U}(1 - q_e) \geqslant \frac{1}{m}$, which implies that $(1 - q_e) \geqslant \frac{1}{m^2}$ for some $e \in U$. □

We are now ready to prove Lemma 10.4.

Proof.
(of Lemma 10.4) We have to show that the algorithm given above can be implemented in polynomial time and it gives a correct answer.

Step 2(a) can be easily implemented by taking poly(m) samples conditioned on $\mathbf{S} \subseteq U$ (as in the proof of Lemma 10.5). This gives us an approximation of $q_U = \mathbb{P}(\mathbf{S} = U : \mathbf{S} \subseteq U)$ up to an additive error of at most m^{-2} with high probability. If the estimate is less than $(1 - \frac{1}{2m})$ then with high probability $q_U \leqslant (1 - \frac{1}{m})$ otherwise, with high probability we have

$$\mu(U) \leqslant \sum_{S \in \mathcal{C}, S \subseteq U} \mu(S) \leqslant \left(1 + \frac{4}{m}\right)\mu(U) \tag{10.3}$$

and the algorithm terminates.

When performing step 2(b) we have a high probability guarantee for the assumption of Lemma 10.6 to be satisfied. Hence, we can assume that (by using Lemma 10.6 and Lemma 10.5) we can find an element $e \in U$ with $p_e = \mathbb{P}(e \notin \mathbf{S} : \mathbf{S} \subseteq U) \geqslant \frac{1}{2m^2}$. Again using Lemma 10.5 we can perform step 2(c) and obtain a multiplicative $(1 + \frac{\varepsilon}{m})$-approximation ρ_e to p_e.

Denote the set U at which the algorithm terminated by U' and the elements chosen at various stages of the algorithm by $e_1, e_2, ..., e_l$ with $l = m - |U'|$. The output of the algorithm is:

$$X \overset{\text{def}}{=} \rho_{e_1} \rho_{e_2} \cdot \dots \cdot p_{e_l} \mu(U').$$

While the exact value of the sum is

$$Z \overset{\text{def}}{=} p_{e_1} p_{e_2} \cdot \dots \cdot p_{e_l} \cdot \sum_{S \in \mathcal{C}, S \subseteq U'} \mu(S).$$

Recall that for every $i = 1, 2, \dots, l$ with high probability it holds that:

$$\left(1 + \frac{\varepsilon}{m}\right)^{-1} \leqslant \frac{p_{e_i}}{\rho_{e_i}} \leqslant \left(1 + \frac{\varepsilon}{m}\right).$$

This, together with (10.3) implies that with high probability:

$$\left(1 + \frac{\varepsilon}{m}\right)^{-l} \leqslant \frac{X}{Z} \leqslant \left(1 + \frac{\varepsilon}{m}\right)^{l} \cdot \left(1 + \frac{4}{m}\right),$$

which finally gives $(1 + 2\varepsilon)^{-1} \leqslant \frac{X}{Z} \leqslant (1 + 2\varepsilon)$ with high probability, as claimed. Note that the algorithm requires $\text{poly}(m, \frac{1}{\varepsilon})$ samples from the oracle in total. □

10.3 Conclusion

10.3.1 Future Work

The hardness result obtained in this chapter implies that the problem of computing mixed discriminants has nontrivial practical applications. However, unlike permanents [JSV04], no efficient approximation scheme is known for estimating mixed discriminants. It is an interesting open question whether the result of [JSV04] can be extended to this case. There are certain obstacles to achieve that, as standard ways of constructing Markov Chains for this counting problem yield non-ergodic chains. Therefore some new ideas, perhaps based on quantum techniques, are required to make progress on this problem.

10.3.2 Notes

The content of this chapter is based on the work [CDK$^+$17]. Sampling from DPPs has been successfully applied to a number of problems, such as document summarization, sensor placement and recommendation systems [LB11, KSG08, ZKL$^+$10, ZCL03, YJ08]. An open question of efficient sampling and counting algorithms for DPPs with additional combinatorial constraints on the support of the distribution was asked by Kulesza and Taskar in their survey [KT12]. For the problem of sampling from k-DPPs (i.e., when the only allowed sets are of cardinality k)

there are exact polynomial time algorithms known (see [HKPV05, DR10, KT12]). There is also recent work on faster approximate MCMC algorithms for sampling from various unconstrained discrete point processes (see [RK15, AOGR16] and the references therein), and algorithms that are efficient for constrained DPPs under certain restrictions on the kernel and constraints (see [LJS16] and the references therein). To the best of our knowledge, the result of this chapter is the first efficient sampling algorithm that works for all kernels and for any constraint set with small description complexity. On the practical side, diverse subset selection and DPPs arise in a variety of contexts such as structured prediction [PJB14], recommender systems [GPK16] and active learning [WIB15], where the study of DPPs with additional constraints is of importance.

Bibliography

[AFHS97] Marco Avellaneda, Craig Friedman, Richard Holmes, and Dominick Samperi. Calibrating volatility surfaces via relative-entropy minimization. *Applied Mathematical Finance*, 4(1):37–64, 1997.

[AGM$^+$10] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An O(Log N/ Log Log N)-approximation Algorithm for the Asymmetric Traveling Salesman Problem. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 379–389, 2010.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[ALOW17] Zeyuan Allen Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Much faster algorithms for matrix scaling. In *FOCS'17: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, 2017.

[AM84] Noga Alon and V. D. Milman. Eigenvalues, expanders and superconcentrators (extended abstract). In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 320–322, 1984.

[AO17] Nima Anari and Shayan Oveis Gharan. A Generalization of Permanent Inequalities and Applications in Counting and Optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 384–396, 2017.

[AOG15] Nima Anari and Shayan Oveis Gharan. Effective-Resistance-Reducing Flows, Spectrally Thin Trees, and Asymmetric TSP. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 20–39, 2015.

[AOGR16] Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 103–115, 2016.

[AOSS17] Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash Social Welfare, Matrix Permanent, and Stable Polynomials. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science*, 2017.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 222–231, 2004.

[AS10] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.*, 39(7):2970–2989, 2010.

[AV97] Noga Alon and Van H. Vu. Anti-Hadamard Matrices, Coin Weighing, Threshold Gates, and Indecomposable Hypergraphs. *J. Comb. Theory, Ser. A*, 79(1):133–160, 1997.

[Bap89] Ravindra B. Bapat. Mixed discriminants of positive semidefinite matrices. *Linear Algebra and its Applications*, 126:107 – 124, 1989.

[Bar97] Alexander Barvinok. Computing Mixed Discriminants, Mixed Volumes, and Permanents . *Discrete & Computational Geometry*, 18:205–237, 1997.

[Bar98] Franck Barthe. On a reverse form of the brascamp-lieb inequality. *Inventiones mathematicae*, 134(2):335–361, Oct 1998.

[BB09] Julius Borcea and Petter Brändén. The Lee-Yang and Pólya-Schur programs. I. linear operators preserving stability. *Inventiones mathematicae*, 177(3):541–569, 2009.

[BBD$^+$13] Luca Becchetti, Vincenzo Bonifaci, Michael Dirnberger, Andreas Karrenbauer, and Kurt Mehlhorn. Physarum Can Compute Shortest Paths: Convergence Proofs and Complexity Bounds. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, pages 472–483, 2013.

[BBL09] Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *J. of the American Mathematical Society*, 22(2), 2009.

[BC95] André Bouchet and William H. Cunningham. Delta-Matroids, Jump Systems, and Bisubmodular Polyhedra. *SIAM J. Discrete Math.*, 8(1):17–32, 1995.

[Bet35] Hans A. Bethe. Statistical Theory of Superlattices. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 150(871):552–575, 1935.

[BG06] Antar Bandyopadhyay and David Gamarnik. Counting without sampling: New algorithms for enumeration problems using statistical physics. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, pages 890–899, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics.

[BGK$^+$07] Mohsen Bayati, David Gamarnik, Dimitriy A. Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 122–127, 2007.

[BKVH07] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and engineering*, 8(1):67, 2007.

[BL89] David A. Bayer and Jeffrey C. Lagarias. The nonlinear geometry of linear programming. I. Affine and projective scaling trajectories. *Transactions of the American Mathematical Society*, pages 499–526, 1989.

[BL02] Herm Jan Brascamp and Elliott H. Lieb. *Best Constants in Young's Inequality, Its Converse, and Its Generalization to More than Three Functions*, pages 417–439. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[BMV12] Vincenzo Bonifaci, Kurt Mehlhorn, and Girish Varma. Physarum can compute shortest paths. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 233–240, 2012.

[Brä07] Petter Brändén. Polynomials with the half-plane property and matroid theory. *Advances in Mathematics*, 216(1):302–320, 2007.

[BTN12] Aharon Ben-Tal and Arkadi Nemirovski. Optimization III: Convex analysis, nonlinear programming theory, nonlinear programming algorithms. Lecture Notes, 2012.

[Bub14] Sebastien Bubeck. Convex Optimization: Algorithms and Complexity. *ArXiv e-prints*, May 2014.

[CC06] Michael Chertkov and Vladimir Y. Chernyak. Loop calculus in statistical physics and information science. *Physical Review E*, 73(6):065102, 2006.

[CDK$^+$17] L. Elisa Celis, Amit Deshpande, Tarun Kathuria, Damian Straszak, and Nisheeth K. Vishnoi. On the Complexity of Constrained Determinantal Point Processes. 2017.

[Cha12] Bernard Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–110, December 2012.

[CKM$^+$11] Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 273–282, 2011.

[CLPV14] Erick Chastain, Adi Livnat, Christos Papadimitriou, and Umesh Vazirani. Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences*, 111(29):10620–10623, 2014.

[ÇM09] Ali Çivril and Mailk Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theor. Comput. Sci.*, 410(47-49):4801–4811, 2009.

[CMTV17] Michael B. Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained newton's method and interior point methods. In *FOCS'17: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, 2017.

[COSW04] Young-Bin Choe, James G. Oxley, Alan D. Sokal, and David G. Wagner. Homogeneous multivariate polynomials with the half-plane property. *Advances in Applied Mathematics*, 32(1):88 – 187, 2004.

[CTV06] Kevin P. Costello, Terence Tao, and Van H. Vu. Random symmetric matrices are almost surely nonsingular. *Duke Math. J.*, 135(2):395–413, 11 2006.

[CW01] Anthony Carbery and James Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical research letters*, 8(3):233–248, 2001.

[CW06] Young-Bin Choe and David G. Wagner. Rayleigh matroids. *Combinatorics, Probability and Computing*, 15(05):765–781, 2006.

[DDS16] Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. A Robust Khintchine Inequality, and Algorithms for Computing Optimal Constants in Fourier Analysis and High-Dimensional Geometry. *SIAM Journal on Discrete Mathematics*, 30(2):1058–1094, 2016.

[Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[DM10] Amir Dembo and Andrea Montanari. Ising Models on Locally Tree-like Graphs. *The Annals of Applied Probability*, 20(2):565–592, 2010.

[DR10] Amit Deshpande and Luis Rademacher. Efficient Volume Sampling for Row/Column Subset Selection. In *FOCS*, Oct 2010.

[DSEFM15] Marco Di Summa, Friedrich Eisenbrand, Yuri Faenza, and Carsten Moldenhauer. On largest volume simplices and sub-determinants. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 315–323. SIAM, 2015.

[Edm65] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

[Edm01] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization - Eureka, You Shrink!, Papers Dedicated to Jack Edmonds, 5th International Workshop, Aussois, France, March 5-9, 2001, Revised Papers*, pages 11–26, 2001.

[Epp95] David Eppstein. *Representing all minimum spanning trees with applications to counting and generation*. UC Irvine, 1995.

[ESV17] Javad B. Ebrahimi, Damian Straszak, and Nisheeth K. Vishnoi. Subdeterminant maximization via nonconvex relaxations and anti-concentration. In *FOCS'17: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, 2017.

[ET75] Shimon Even and Robert Endre Tarjan. Network flow and testing graph connectivity. *SIAM J. Comput.*, 4(4):507–518, 1975.

[Fay91] Leonid Faybusovich. Hamiltonian structure of dynamical systems which solve linear programming problems. *Physica D: Nonlinear Phenomena*, 53(2-4):217–232, 1991.

[FI05] Satoru Fujishige and Satoru Iwata. Bisubmodular Function Minimization. *SIAM J. Discrete Math.*, 19(4):1065–1073, 2005.

[FJV11] G. David Forney Jr. and Pascal O. Vontobel. Partition Functions of Normal Factor Graphs. In *Information Theory and Applications Workshop, UC San Diego, La Jolla, CA, USA, Feb. 6-11, 2011*, 2011.

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University press, 2009.

[Ful56] Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[GGOW17] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of brascamp-lieb inequalities, via operator scaling. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 397–409, 2017.

[GLS88] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[GPK16] Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Bayesian Low-Rank Determinantal Point Processes. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 349–356, 2016.

[GR98] Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, 1998.

[GS02] Leonid Gurvits and Alex Samorodnitsky. A Deterministic Algorithm for Approximating the Mixed Discriminant and Mixed Volume, and a Combinatorial Corollary. *Discrete & Computational Geometry*, 27:531–550, 2002.

[GS14] Leonid Gurvits and Alex Samorodnitsky. Bounds on the Permanent and Some Applications. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 90–99, 2014.

[GT17] Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-NC. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 821–830, 2017.

[Gül97] Osman Güler. Hyperbolic Polynomials and Interior Point Methods for Convex Programming. *Math. Oper. Res.*, 22(2):350–377, 1997.

[Gur05] Leonid Gurvits. *On the Complexity of Mixed Discriminants and Related Problems*, pages 447–458. Springer Berlin Heidelberg, 2005.

[Gur06] Leonid Gurvits. Hyperbolic polynomials approach to Van der Waerden/Schrijver-Valiant like conjectures: sharper bounds, simpler proofs and algorithmic applications. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 417–426. ACM, 2006.

[Gur11] Leonid Gurvits. Unleashing the power of Schrijver's permanental inequality with the help of the Bethe Approximation. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:169, 2011.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[Har09] Nicholas J. A. Harvey. Algebraic algorithms for matching and matroid problems. *SIAM Journal on Computing*, 39(2):679–702, 2009.

[HKPV05] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. Determinantal Processes and Independence. *ArXiv Mathematics e-prints*, March 2005.

[IJNT11] Kentaro Ito, Anders Johansson, Toshiyuki Nakagaki, and Atsushi Tero. Convergence Properties for the Physarum Solver. *ArXiv e-prints*, January 2011.

[Jay57a] Edwin T. Jaynes. Information Theory and Statistical Mechanics. *Physical Review*, 106:620–630, May 1957.

[Jay57b] Edwin T. Jaynes. Information Theory and Statistical Mechanics. II. *Physical Review*, 108:171–190, October 1957.

[Jay82] Edwin T. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9):939–952, 1982.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A Polynomial-time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *J. ACM*, 51(4):671–697, July 2004.

[JVV86] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

[JZ12] Anders Johannson and James Zou. A Slime Mold Solver for Linear Programming Problems. In *How the World Computes*, volume 7318 of *Lecture Notes in Computer Science*, pages 344–354. Springer Berlin Heidelberg, 2012.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[Kar90] Narendra Karmarkar. Riemannian Geometry Underlying Interior Point Methods for Linear programming. *AMS series on Contemporary Mathematics*, 114:51–75, 1990.

[Kar91] Howard Karloff. *Linear Programming*. Birkhauser Boston Inc., 1991.

[Kha80] Leonid Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.

[Kha95] Leonid Khachiyan. On the complexity of approximating extremal determinants in matrices. *Journal of Complexity*, 11(1):138–153, 1995.

[KLOS14] Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 217–226, 2014.

[KLRS08] Bahman Kalantari, Isabella Lari, Federica Ricca, and Bruno Simeone. On the complexity of general matrix scaling and entropy minimization via the RAS algorithm. *Math. Program.*, 112(2):371–401, 2008.

[Kru56] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[KSG08] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008.

[KT12] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *ArXiv*, July 2012.

[LB11] Hui Lin and Jeff Bilmes. A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, 2011.

[Lie90] Elliott H. Lieb. Gaussian kernels have only gaussian maximizers. *Inventiones mathematicae*, 102(1):179–208, Dec 1990.

[LJS16] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Markov Chain Sampling in Discrete Probabilistic Models with Constraints. In *NIPS*, 2016.

[Lov89] László Lovász. Singular spaces of matrices and their application in combinatorics. *Boletim da Sociedade Brasileira de Matemática-Bulletin/Brazilian Mathematical Society*, 20(1):87–99, 1989.

[LRS13] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 755–764, 2013.

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in õ(vrank) iterations and faster algorithms for maximum flow. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 424–433, 2014.

[LV90] Jeffrey C. Lagarias and Robert J. Vanderbei. II Dikin's convergence result for the affine-scaling algorithm. In *Mathematical Developments Arising from Linear Programming: Proceedings of a Joint Summer Research Conference Held at Bowdoin College, June 25-July 1, 1988*, volume 114, page 109. American Mathematical Soc., 1990.

[Lyo02] Russell Lyons. Determinantal probability measures. *ArXiv Mathematics e-prints*, April 2002.

[Mad13] Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 253–262, 2013.

[MO07] Tomoyuki Miyaji and Isamu Ohnishi. Mathematical analysis to an adaptive network of the Plasmodium system. *Hokkaido Math. J.*, 36(2):445–465, 2007.

[MO08] Tomoyuki Miyaji and Isamu Ohnishi. Physarum can solve the shortest path problem on Riemannian surface mathematically rigourously. *International Journal of Pure and Applied Mathematics*, 47(3):353–369, 2008.

[Mor13] Ryuhei Mori. *New Understanding of the Bethe Approximation and the Replica Method*. PhD thesis, Kyoto University, 2013.

[MS89] Nimrod Megiddo and Michael Shub. Boundary Behavior of Interior Point Algorithms in Linear Programming. *Math. Oper. Res.*, 14(1):97–146, March 1989.

[MS06] Leonid M. Martyushev and Vladimir D. Seleznev. Maximum entropy production principle in physics, chemistry and biology. *Physics reports*, 426(1):1–45, 2006.

[MSS13] Adam Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 529–537. IEEE, 2013.

[MSS15] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families II: Mixed characteristic polynomials and the Kadison–Singer problem. *Annals of Mathematics*, 182(1):327–350, 2015.

[Nes13] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Nes14] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Springer Publishing Company, Incorporated, 1 edition, 2014.

[Nig99] Kamal Nigam. Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[Nik15] Aleksandar Nikolov. Randomized rounding for the largest simplex problem. In *STOC*, 2015.

[NS16] Aleksandar Nikolov and Mohit Singh. Maximizing Determinants Under Partition Constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 192–201, 2016.

[NT02] Yurii E. Nesterov and Michael J. Todd. On the Riemannian Geometry Defined by Self-Concordant Barriers and Interior-Point Methods. *Foundations of Computational Mathematics*, pages 333–361, 2002.

[NYT00] Toshiyuki Nakagaki, Hiroyasu Yamada, and Agota Toth. Maze-solving by an amoeboid organism. *Nature*, 407(6803):470, September 2000.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014.

[OSS11] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *FOCS'11: Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 267–276, 2011.

[Oxl06] James G. Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.

[Pea89] Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.

[Pem04] Robin Pemantle. Uniform random spanning trees. *arXiv preprint math/0404099*, 2004.

[Pem11] Robin Pemantle. Hyperbolicity and stable polynomials in combinatorics and probability. *Current developments in mathematics*, pages 57–123, 2011.

[Pen16] Richard Peng. Approximate undirected maximum flows in $O(m\text{polylog}(n))$ time. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1862–1867, 2016.

[Per01] Lawrence Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2001.

[PJB14] Adarsh Prasad, Stefanie Jegelka, and Dhruv Batra. Submodular meets Structured: Finding Diverse Subsets in Exponentially-Large Structured Item Sets. In *Advances in Neural Information Processing Systems, December 8-13 2014, Montreal, Quebec, Canada*, pages 2645–2653, 2014.

[PPL97] Stephen Della Pietra, Vincent J. Della Pietra, and John D. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, 1997.

[Qi88] Liqun Qi. Directed submodularity, ditroids and directed submodular flows. *Math. Program.*, 42(1-3):579–599, 1988.

[Ren88] James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Math. Program.*, 40(1-3):59–93, 1988.

[RK15] Patrick Rebeschini and Amin Karbasi. Fast Mixing for Discrete Point Processes. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 1480–1500, 2015.

[Ruo12] Nicholas Ruozzi. The Bethe Partition Function of Log-supermodular Graphical Models. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 117–125, 2012.

[RV13] Alexander Razborov and Emanuele Viola. Real advantage. *ACM Transactions on Computation Theory (TOCT)*, 5(4):17, 2013.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Inc., 1986.

[Sch98] Alexander Schrijver. Counting 1-Factors in Regular Bipartite Graphs. *J. Comb. Theory, Ser. B*, 72(1):122–135, 1998.

[She13] Jonah Sherman. Nearly maximum flows in nearly linear time. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 263–269, 2013.

[Sin93] Alistair Sinclair. *Algorithms for random generation and counting - a Markov chain approach.* Progress in theoretical computer science. Birkhäuser, 1993.

[Soo00] Ehsan S Soofi. Principal information theoretic approaches. *Journal of the American Statistical Association*, 95(452):1349–1353, 2000.

[Spi12] Daniel A. Spielman. Algorithms, Graph Theory, and the Solution of Laplacian Linear Equations. In *ICALP (2)*, pages 24–26, 2012.

[SST14] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *Journal of Statistical Physics*, 155(4):666–686, May 2014.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC'04: Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, pages 81–90, 2004.

[ST09] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009.

[SV14] Mohit Singh and Nisheeth K. Vishnoi. Entropy, optimization and counting. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 50–59. ACM, 2014.

[SV16a] Damian Straszak and Nisheeth K. Vishnoi. Natural algorithms for flow problems. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1868–1883, 2016.

[SV16b] Damian Straszak and Nisheeth K. Vishnoi. On a natural dynamics for linear programming. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, page 291, 2016.

[SV17a] D. Straszak and N. K. Vishnoi. Computing Maximum Entropy Distributions Everywhere. *ArXiv e-prints*, November 2017.

[SV17b] Damian Straszak and Nisheeth K. Vishnoi. Belief propagation, bethe approximation and polynomials. *CoRR*, abs/1708.02581, 2017.

[SV17c] Damian Straszak and Nisheeth K. Vishnoi. Irls and slime mold: Equivalence and convergence. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science, 2017*, 2017.

[SV17d] Damian Straszak and Nisheeth K. Vishnoi. Real Stable Polynomials and Matroids: Optimization and Counting. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 370–383, 2017.

[SW87] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[SWW07] Erik B. Sudderth, Martin J. Wainwright, and Alan S. Willsky. Loop Series and Bethe Variational Bounds in Attractive Graphical Models. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1425–1432, 2007.

[Ten10] Shang-Hua Teng. The Laplacian Paradigm: Emerging Algorithms for Massive Graphs. In *TAMC*, pages 2–14, 2010.

[TKN07] Atsushi Tero, Ryo Kobayashi, and Toshiyuki Nakagaki. A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology*, 244(4):553, 2007.

[Val79] Leslie G. Valiant. The Complexity of Computing the Permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[Val09] Leslie G. Valiant. Evolvability. *J. ACM*, 56(1):3:1–3:21, February 2009.

[Vaz13] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[Vin06] Hrishikesh D Vinod. Maximum entropy ensembles for time series inference in economics. *Journal of Asian Economics*, 17(6):955–978, 2006.

[Vis12] Nisheeth K. Vishnoi. $Lx = b$. *Foundations and Trends in Theoretical Computer Science*, 8(1-2):1–141, 2012.

[Vis13] Nisheeth K. Vishnoi. Zeros of polynomials and their applications to theory: a primer. In *FOCS 2013 Workshop on Zeros of Polynomials and their Applications to Theory*, 2013.

[Vis15] Nisheeth K. Vishnoi. The speed of evolution. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1590–1601, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics.

[Von10] Jan Vondrák. Submodularity and curvature: the optimal algorithm. *RIMS Kokyuroku Bessatsu*, B 23:253–266, 2010.

[Von13a] Pascal O. Vontobel. Counting in Graph Covers: A Combinatorial Characterization of the Bethe Entropy Function. *IEEE Trans. Information Theory*, 59(9):6018–6048, 2013.

[Von13b] Pascal O. Vontobel. The Bethe Permanent of a Nonnegative Matrix. *IEEE Trans. Information Theory*, 59(3):1866–1901, 2013.

[Wag11] David Wagner. Multivariate stable polynomials: theory and applications. *Bulletin of the American Mathematical Society*, 48(1):53–84, 2011.

[Wat11] Yusuke Watanabe. A conjecture on independent sets and graph covers. *ArXiv e-prints*, September 2011.

[WC10] Yusuke Watanabe and Michael Chertkov. Belief propagation and loop calculus for the permanent of a non-negative matrix. *Journal of Physics A: Mathematical and Theoretical*, 43(24), 2010.

[Wei06] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 140–149, 2006.

[WIB15] Kai Wei, Rishabh K. Iyer, and Jeff A. Bilmes. Submodularity in Data Subset Selection and Active Learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1954–1963, 2015.

[WJ08] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.

[Wri97] Steve Wright. *Primal-Dual Interior-Point Methods.* Society for Industrial and Applied Mathematics, 1997.

[YFW05] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Information Theory*, 51(7):2282–2312, 2005.

[YJ08] Yisong Yue and Thorsten Joachims. Predicting Diverse Subsets Using Structural SVMs. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1224–1231, 2008.

[ZCL03] Cheng X. Zhai, William W. Cohen, and John 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*, SIGIR '03, pages 10–17, 2003.

[ZKL$^+$10] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

Damian Straszak

Education

2014-2017	**Ph.D.**, *EPFL*, Switzerland, Advisor: Nisheeth K. Vishnoi.
2011-2014	**M.Sc.**, *University of Wrocław*, Poland, Double major studies: Mathematics and Computer Science.
2008-2011	**B.Sc.**, *University of Wrocław*, Poland, Double major studies: Mathematics and Computer Science.

Research Interests

- Convex Optimization and Linear Programming,
- Spectral Methods and Real Stable Polynomials,
- Dynamical Systems.

Publications

2017 **Subdeterminant Maximization via Nonconvex Relaxations and Anti-concentration**, *Javad B. Ebrahimi, Damian Straszak and Nisheeth K. Vishnoi*, Symposium on Foundations of Computer Science (**FOCS**).

2017 **On the Complexity of Constrained Determinantal Point Processes**, *L. Elisa Celis, Amit Deshpande, Tarun Kathuria, Damian Straszak, Nisheeth K. Vishnoi*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (**APPROX/RANDOM**).

2017 **Real Stable Polynomials and Matroids: Optimization and Counting**, *Damian Straszak and Nisheeth K. Vishnoi*, in Symposium on the Theory of Computing (**STOC**).

2017 **IRLS and Slime Mold: Equivalence and Convergence**, *Damian Straszak and Nisheeth K. Vishnoi*, in Innovations in Theoretical Computer Science (**ITCS**), Invited Paper.

2016 **On a Natural Dynamics for Linear Programming**, *Damian Straszak and Nisheeth K. Vishnoi*, in Innovations in Theoretical Computer Science (**ITCS**).

2016 **Natural Algorithms for Flow Problems**, *Damian Straszak and Nisheeth K. Vishnoi*, in ACM-SIAM Symposium on Discrete Algorithms (**SODA**).

2015 **Strong Inapproximability of the Shortest Reset Word**, *Paweł Gawrychowski and Damian Straszak*, in International Symposium on Mathematical Foundations of Computer Science (**MFCS**), ***Best Paper Award***.

2015 **Fast Generation of Random Spanning Trees and the Effective Resistance Metric**, *Aleksander Mądry, Damian Straszak and Jakub Tarnawski*, in ACM-SIAM Symposium on Discrete Algorithms (**SODA**).

2013 **Beating O(nm) in Approximate LZW-Compressed Pattern Matching**, *Paweł Gawrychowski and Damian Straszak*, in International Symposium on Algorithms and Computation (**ISAAC**).

Achievements and Awards

May 2015	Bronze medal in the **Bayan Programming Contest** 2015, Tehran.
Jun 2014	Team Coach at the **ACM ICPC World Finals** 2014, Yekaterinburg.
Jan 2014	Second prize in the competition for the best Master Thesis in Computer Science organized by the Polish Computer Science Society.
Jun 2013	Participant of the **ACM ICPC World Finals** 2013, Saint Petersburg.
2012, 2013	The scholarship of the Polish Minister of Science and Higher Education.

Conference and Invited Talks

Oct 2017	**Subdeterminant Maximization via Nonconvex Relaxations and Anti-concentration**, *Symposium on Foundations of Computer Science (**STOC**)*, Berkeley, US.
June 2017	**Real Stable Polynomials and Matroids: Optimization and Counting**, *Symposium on the Theory of Computing (**FOCS**)*, Montreal, Canada.
Jan 2017	**Real Stable Polynomials and Computing Partition Functions**, *Complexity and Cryptography Workshop 2017*, IISc Bangalore, India, **Invited talk**.
Jun 2016	**Slime Molds and Sparse Recovery**, *International Conference on Continuous Optimization (**ICCOPT**) 2016*, Tokyo, Japan, **Invited talk**.
Jan 2016	**On a Natural Dynamics for Linear Programming**, *Innovations in Theoretical Computer Science (**ITCS**) 2016*, MIT Cambridge, US.
Jan 2016	**On a Natural Dynamics for Linear Programming**, *ACM-SIAM Symposium on Discrete Algorithms (**SODA**)*, Arlington, US.
Jan 2016	**On a Natural Dynamics for Linear Programming**, *Evolution and Computing*, Dagstuhl, Germany.
Dec 2013	**Beating O(nm) in Approximate LZW-Compressed Pattern Matching**, *International Symposium on Algorithms and Computation (**ISAAC**) 2013*, Hong Kong.

Teaching Experience

Fall 2016	Teaching Assistant, **Analytic Algorithms**, EPFL.
Nov 2016	Instructor, Preparing students to the **Regional ACM ICPC**, EPFL.
Spring 2016	Teaching Assistant, **Theory of Computation**, EPFL.
Fall 2015	Teaching Assistant, **Analytic Methods in Algorithms and Complexity**, EPFL.
Nov 2015	Instructor, Preparing students to the **Regional ACM ICPC**, EPFL.
Spring 2015	Teaching Assistant, **Theory of Computation**, EPFL.

Spring 2014	Teaching Assistant, **Formal Languages and Complexity Theory**, University of Wrocław.
Fall 2013	Instructor, **Seminar on Competitive Programming**, University of Wrocław.