

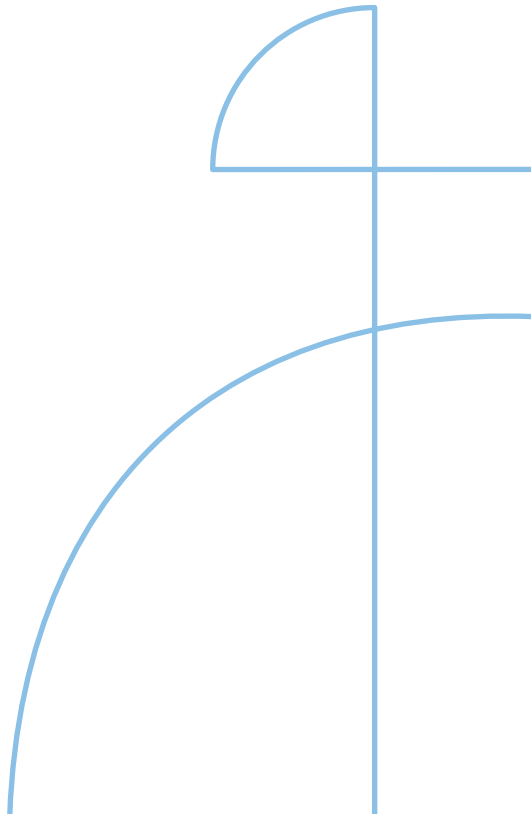


Doctoral Thesis in Computer Science

Fairness and Diversity-Aware Algorithms: Ranking, Streaming, and Graph Analysis

HONGLIAN WANG

KTH ROYAL INSTITUTE OF TECHNOLOGY



Fairness and Diversity-Aware Algorithms: Ranking, Streaming, and Graph Analysis

HONGLIAN WANG

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Tuesday the 5th of May 2026, at 2:00 p.m. in F3, Lindstedtvägen 26, Stockholm.

Doctoral Thesis in Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden 2026

© Honglian Wang

© Aristides Gionis, Sijing Tu, Lutz Oettershagen, Haoyun Zhou, Florian Adriaens

TRITA-EECS-AVL-2026:34

ISBN 978-91-8106-581-7

Printed by: Universitetservice US-AB, Sweden 2026

Abstract

As algorithmic systems increasingly shape human experiences, ensuring fairness and diversity has become a central challenge. This thesis studies fairness and diversity through the lens of algorithm design and optimization theory, providing formal frameworks and efficient algorithms across three domains: ranking-based recommendation, streaming recommendation, and graph analysis.

The first part of the thesis investigates diversity maximization in recommender systems with stochastic user engagement. We first study how to rank items in recommendation systems, where users engage with content sequentially and probabilistically. We introduce two novel diversity measures, sequential sum diversity and sequential coverage diversity, which account for uncertainty in user engagement. Our goal is to find a ranking of items that maximizes these sequential diversity measures. We show that sequential coverage diversity is ordered submodular, enabling a greedy $\frac{1}{2}$ -approximation. For sequential sum diversity, we provide polynomial-time constant-factor approximation algorithms. Separately, we study a streaming setting where items arrive continuously and users may visit the system multiple times at arbitrary moments. For this setting, we aim to design a streaming algorithm that maximizes a stochastic coverage diversity measure. We show that a classic greedy algorithm achieves a tight $\frac{1}{2}$ -competitive ratio but requires memory linear in the stream length. With sublinear memory and an upper bound T' on the number of user visits T , we propose STORM, which achieves a $\frac{1}{4(T'-T+1)}$ -competitive ratio. We further propose STORM++ improving the competitive ratio to $\frac{1}{8\delta}$, where the integer parameter δ controls the tradeoff between solution quality and computational cost.

The second part of the thesis studies diversity as a constraint in densest subgraph discovery and addresses the problem of finding dense communities in networks with heterogeneous relationship types. We model relationship types as edge colors and formulate the At Least h Colored Edges Densest Subgraph problem (ALHCEDGESDSP), which seeks subgraphs that are both dense and contain at least h_i edges of each color i . We prove that even the simplest variant of this problem is NP-hard and develop constant-factor approximation algorithms. Our key technical contribution links the edge-constrained and node-constrained versions of the densest subgraph problem. We first show that algorithms for the At Least k Nodes Densest Subgraph problem (DalkS) can approximate the At Least h edges Densest Subgraph problem (ATLEASTHEDGESDSP), and then extend the algorithm for DalkS to handle colored edge constraints for solving ALHCEDGESDSP.

The third part of the thesis studies graph interventions for fairness in networks. We examine two fairness measures, PageRank fairness and hitting-time fairness, developing methods to balance influence and improve accessibility across groups. For each demographic group, the sum of PageRank scores within it quantifies the influence of that group. PageRank fairness measures how far the current group-wise influence deviates from a given target. We formulate the PageRank fairness problem as an optimization problem that adjusts edge weights such that the resulting graph achieves a group-wise influence distribution as close to the target as possible. The optimization problem involves a nonconvex objective over a convex feasible set under practical constraints, such as not adding new

edges and limiting the magnitude of weight changes. We solve this PageRank fairness maximization problem using efficient projected gradient descent, proving convergence to a stationary point. For hitting-time fairness in bipartite graphs, we formulate two problems, minimizing the average (BMAH) and the maximum hitting time (BMMH) from one group to another via strategic edge additions. We provide a $(2 + \epsilon)$ -approximation for BMAH by combining fast random walk simulation with greedy supermodular minimization. For the more challenging BMMH problem, we develop two approaches, the first leverages its connection to the BMAH problem, and the second employs a method based on the asymmetric k center problem. Both approaches yield provable approximation guarantees for BMMH.

The algorithms and analysis techniques presented in this thesis contribute to the growing body of work on fairness and diversity in algorithmic systems. By formalizing new problem variants that capture realistic constraints in interactive and networked settings, and by providing approximation algorithms with provable guarantees, this work expands the toolkit available for addressing fairness and diversity challenges in computational systems.

Sammanfattning

Eftersom algoritmiska system i allt högre grad formar mänskliga upplevelser har det blivit en central utmaning att säkerställa rättvisa och mångfald. Denna avhandling studerar rättvisa och mångfald genom algoritm design och optimeringsteori, och tillhandahåller formella ramverk och effektiva algoritmer inom tre domäner: rankningsbaserad rekommendation, strömmande rekommendation och grafanalys.

Den första delen av avhandlingen undersöker mångfalldsmaximering i rekommendationssystem med stokastiskt användarengagemang. Vi studerar först hur man rangordnar objekt i rekommendationssystem, där användare engagerar sig med innehåll sekventiellt och probabilistiskt. Vi introducerar två nya mångfalldsmått, sekventiell summa-mångfald och sekventiell täcknings-mångfald, som tar hänsyn till osäkerhet i användarengagemang. Vårt mål är att hitta en rangordning av objekt som maximerar dessa sekventiella mångfalldsmått. Vi visar att sekventiell täcknings-mångfald är ordnad submodulär, vilket möjliggör en girig $\frac{1}{2}$ -approximation. För sekventiell summa-mångfald tillhandahåller vi polynomtids konstant-faktor approximationsalgoritmer. Separat studerar vi en strömmande miljö där objekt anländer kontinuerligt och användare kan besöka systemet flera gånger vid godtyckliga tidpunkter. För denna miljö strävar vi efter att designa en strömmande algoritim som maximerar ett stokastiskt täcknings-mångfalldsmått. Vi visar att en klassisk girig algoritim uppnår ett tight $\frac{1}{2}$ -konkurrensförhållande men kräver minne linjärt i strömlängden. Med sublinjärt minne och en övre gräns T' på antalet användarbesök T , föreslår vi STORM, som uppnår ett $\frac{1}{4(T'-T+1)}$ -konkurrensförhållande. Vi föreslår vidare STORM++ som förbättrar konkurrensförhållandet till $\frac{1}{8\delta}$, där heltalsparametern δ kontrollerar avvägningen mellan lösningskvalitet och beräkningskostnad.

Den andra delen av avhandlingen studerar mångfald som en bivillkor i upptäckt av tätaste delgrafer och behandlar problemet med att hitta täta gemenskaper i nätverk med heterogena relationstyper. Vi modellerar relationstyper som kantfärger och formulerar problemet At Least h Colored Edges Densest Subgraph (ALHCEDGESDSP), som söker delgrafer som är både täta och innehåller åtminstone h_i kanter av varje färg i . Vi bevisar att även den enklaste varianten av detta problem är NP-svår och utvecklar konstant-faktor approximationsalgoritmer. Vårt viktigaste tekniska bidrag kopplar samman kant-begränsade och nod-begränsade versionerna av tätaste delgraf-problemet. Vi visar först att algoritmer för problemet At Least k Nodes Densest Subgraph (Dal k S) kan approximera problemet At Least h edges Densest Subgraph (ATLEASTHEDGESDSP), och utökar sedan algoritmen för Dal k S för att hantera färgade kantbegränsningar för att lösa ALHCEDGESDSP.

Den tredje delen av avhandlingen studerar grafinterventioner för rättvisa i nätverk. Vi undersöker två rättvisemått, PageRank-rättvisa och träffids-rättvisa, och utvecklar metoder för att balansera inflytande och förbättra tillgänglighet mellan grupper. För varje demografisk grupp kvantifierar summan av PageRank-poäng inom den gruppens inflytande. PageRank-rättvisa mäter hur långt den nuvarande gruppvisa influensfördelningen avviker från ett givet mål. Vi formulerar PageRank-rättvisamproblemet som ett optimeringsproblem som justerar kantvikter så att den resulterande grafen uppnår en gruppvis influensfördelning så

nära målet som möjligt. Optimeringsproblemet involverar ett icke-konvext mål över en konvex genomförbar mängd under praktiska begränsningar, såsom att inte lägga till nya kanter och begränsa kantviktningsskalan. Vi löser detta PageRank rättvisemaximerings-problem med hjälp av effektiv projicerad gradientnedstigning, och bevisar konvergens till en stationär punkt. För träffids-rättvisa i bipartita grafer formulerar vi två problem, att minimera medelvärdet (BMAH) och det maximala träffidsvärdet (BMMH) från en grupp till en annan via strategiska kanttillägg. Vi tillhandahåller en $(2 + \epsilon)$ -approximation för BMAH genom att kombinera snabb slumpvandringssimulering med girig supermodulär minimering. För det mer utmanande BMMH-problemet utvecklar vi två tillvägagångssätt, det första utnyttjar dess koppling till BMAH-problemet, och det andra använder en metod baserad på det asymmetriska k center-problemet. Båda tillvägagångssätten ger bevisliga approximationsgarantier för BMMH.

Algoritmerna och analystekniker som presenteras i denna avhandling bidrar till den växande mängden arbete om rättvisa och mångfald i algoritmiska system. Genom att formalisera nya problemvarianter som fångar realistiska begränsningar i interaktiva och nätverksbaserade miljöer, och genom att tillhandahålla approximationsalgoritmer med bevisliga garantier, utökar detta arbete verktygslådan som finns tillgänglig för att hantera utmaningar kring rättvisa och mångfald i beräkningssystem.

List of Papers

Papers Included

- A ***Sequential Diversification with Provable Guarantees***
Honglian Wang, Sijing Tu, and Aristides Gionis
Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (2025)
- B ***Streaming Stochastic Submodular Maximization with On-Demand User Requests***
Honglian Wang, Sijing Tu, Lutz Oettershagen and Aristides Gionis
Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
- C ***Finding Densest Subgraphs with Edge-Color Constraints***
Lutz Oettershagen, **Honglian Wang** and Aristides Gionis
Proceedings of the ACM Web Conference (2024)
- D ***Fairness-aware PageRank via Edge Reweighting***
Honglian Wang, Haoyun Zhou, and Aristides Gionis
Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (2026)
- E ***Minimizing Hitting Time Between Disparate Groups with Shortcut Edges***
Florian Adriaens, **Honglian Wang** and Aristides Gionis
Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2023)

Papers Not Included

- i. ***Scalable Graph Classification via Random Walk Fingerprints***
Peiyan Li, **Honglian Wang** and Christian Böhm
IEEE International Conference on Data Mining (2024)
- ii. ***Influence Without Authority: Maximizing Information Coverage in Hypergraphs***
Peiyan Li, **Honglian Wang**, Kai Li and Christian Bohm
SIAM International Conference on Data Mining (2023)

Claim of Contribution

In Paper A, the problem was proposed by Aris and formulated by Aris and Sijing. The author and Sijing contributed equally to the theoretical analysis and writing. All authors participated in regular discussions to verify proofs and experimental design. The author conducted the experiments.

In Paper B, all authors contributed to conceptualizing the problem and participated in regular discussions. The author and Sijing contributed to the theoretical analysis. The author conducted the experiments. All authors contributed to the writing.

In Paper C, the problem was proposed and formulated by Lutz and Aris. Lutz is the main contributor to the theoretical analysis, experiments, and writing. The author contributed to the theoretical analysis and writing. All authors participated in regular discussions.

In Paper D, the problem was proposed and formulated by Haoyun and Aris. Haoyun and Aris formed the initial draft of the paper. Haoyun completed the initial version of experiments and wrote the abstract, introduction, methods section, and preliminary experimental results before discontinuing work on the project. The author corrected the initial projected gradient descent algorithm and provided convergence analysis for it. The author verified and revised Haoyun's source code, re-ran all experiments, and added new experiments. Aris and the author participated in regular discussions until the paper was completed. The author rewrote the methods and experimental sections. Aris contributed to writing the abstract and introduction, and polished and revised the author's written drafts.

In Paper E, Florian is the main contributor to the problem formulation, theoretical analysis, and writing. The author conducted most of the experiments. The author and Aris contributed to proofreading and polishing the writing. All authors participated in regular discussions.

Acknowledgements

My doctoral journey has been a profound process of pushing beyond my intellectual comfort zone. I entered this field with an ambitious naivety, which was followed by a humbling phase where I felt like a child playing in the mud outside the grand temple of academia. Eventually, I reached a point where I could contribute something uniquely my own. Through persistence, I dissolved my self-doubt, expanded my horizons, and learned to navigate the pressures of research. Looking back, I am most grateful to the version of myself who came very close to quitting but ultimately chose to keep going.

I am deeply grateful to my supervisor, Aris, who is the best advisor I could have hoped for. He treated me with unwavering respect and granted me great freedom. When progress was slow, he never pressured me, but instead trusted that I would find my way. When I doubted my abilities, he reassured me that my struggles were a common part of the PhD experience rather than a personal failure. I am also sincerely thankful for his generous financial support, which allowed me to participate in various conferences and academic activities.

My growth as a researcher was shaped significantly by my collaborators. Florian, my collaborator on Paper E, worked with me when I felt like a headless fly lost in the research process. Together, we experienced how sustained effort leads to breakthroughs. The shared joy we felt when a long-troubling problem finally moved forward is what first showed me the true beauty of theoretical research. That clarity gave me the lasting confidence that time spent struggling is never wasted.

I also wish to thank Sijing, whose professionalism and attention to detail I deeply admire. Collaborating with her taught me not only how to refine my projects but how to work effectively with different personalities. I am grateful to Lutz for the inspiration of his brilliant writing and presentation skills. I also thank Guangyi, my collaborator from my first year, whose advice on time management and research wisdom I have come to appreciate more with every passing year.

The warmth of the research group made this journey a time of connection as well as study. I am grateful to Tianyi for the fun moments we shared in our office and to Ruochun, whose companionship during our local exercises brought a sense of balance to my daily life. I thank Yifei for his warm approachability and Ilie for the energy he brought to our group activities. Stephan gave presentations so brilliant that he became my academic idol, and I am grateful to Sebastian, Suhas, Ece, and Meher for being such friendly colleagues.

Beyond the group, I am grateful to my closest friends in the department, Yuxin,

Andreas, and Amir. To everyone I shared meals and endless conversations with in the kitchen, you helped me improve my English and gave me a deep affection for this department. Likewise, I thank Shuangjie, Xin, and Qiuya for bringing everyday life back into my PhD years and reminding me that I am a person, not just a research machine.

My deepest gratitude belongs to my family. I thank my parents for their quiet and unconditional support. To my husband, Peiyan, we pursued our PhDs side by side while weathering the pressure and celebrating each other's successes. Through love, setbacks, and perseverance, we have emerged as two new doctors and stronger people than when we began.

Finally, I give all my love to my two cats, Alpha and Eta. Your gentle purrs and quiet companionship have been my comfort in every season. Alpha, my best boy in the world, I hope your soul rests in peace.

Contents

Abstract	iii
Acknowledgements	vii
List of Papers	vii
Contents	xi
I Thesis	1
1 Introduction	3
1.1 Diversity Maximization	5
1.2 Streaming Stochastic Coverage Maximization	6
1.3 Diversified Densest Subgraph Discovery	8
1.4 Fairness-Aware PageRank	9
1.5 Reducing Structural Bias via Hitting-Time Minimization	11
1.6 Thesis Organization	12
2 Preliminaries	13
2.1 Basics of Approximation Algorithms	13
2.2 Submodular Optimization and Matroid Constraint	15
2.3 Graphs and Graph Measures	17
2.4 Streaming Algorithms	19
2.5 The k -Center Problem	20
2.6 The Asymmetric k -Center Problem	21
3 Diversity Maximization in Interactive Systems	23
3.1 [Paper A] Sequential Diversity Maximization	23
3.2 [Paper B] Streaming Diversity Maximization	27
4 Diversity-Constrained Densest Subgraph Discovery	33
4.1 [Paper C] Finding Densest Subgraphs with Edge-Color Constraints	33
4.2 Reducing ATLEASTHEDGESDSP to Dal k S	35
4.3 Reducing ALHCEGESDSP to ATLEASTHEDGESDSP	36

4.4	Conclusion	37
5	Graph Interventions for Fairness in Networks	39
5.1	[Paper D] Fairness-Aware PageRank via Edge-Rewiring	39
5.2	[Paper D] Fairness-Aware Structural Bias Reduction	42
6	Conclusion and Future Work	49
6.1	Limitations and Future Work	49
	References	53
	II Included Papers	63
	Paper A	65
	Paper B	87
	Paper C	113
	Paper D	127
	Paper E	147

Part I
Thesis

Chapter 1

Introduction

As algorithmic systems increasingly shape how we access information, form opinions, and interact with one another, concerns about fairness and diversity have become central to both public discourse and scientific inquiry. News recommendation algorithms influence political awareness and polarization [1], social media feeds affect whose voices are amplified or marginalized [2], and online marketplaces and hiring platforms shape access to economic opportunities [3]. Well-documented phenomena such as filter bubbles and echo chambers highlight how a lack of diversity in algorithmic outputs can narrow users' exposure to information and reinforce existing beliefs [4]. At the same time, growing evidence shows that algorithmic decision-making can reproduce or even exacerbate societal biases, leading to unfair outcomes for historically disadvantaged groups in areas such as advertising, credit, policing, and employment [5, 6]. These concerns are no longer abstract or purely philosophical. They arise from systems that are already deployed at scale and affect millions of users. Addressing these concerns therefore requires systematic, principled approaches that translate normative notions of fairness and diversity into concrete algorithmic objectives. This thesis contributes to this effort by studying fairness and diversity from an algorithmic and optimization-oriented perspective, with a focus on interactive recommendation system design and graph analysis.

Although fairness and diversity are conceptually distinct, they are deeply intertwined in the design of modern algorithmic systems. Fairness concerns the equitable treatment of individuals and groups, addressing questions of justice, equality, and the distribution of resources or opportunities. In contrast, diversity emphasizes heterogeneity and variety, focusing on the breadth of content, perspectives, or options presented by a system. Understanding how these two objectives relate (where they reinforce one another and where they come into tension) requires both philosophical grounding and technical precision.

In computational settings, fairness is typically formalized in several complementary ways. One common perspective is distributive fairness [7, 8], which asks whether algorithmic outcomes are equitably allocated across different groups. In recommendation systems, this may concern whether users from different demographics receive

equally relevant or valuable recommendations. In social or information networks, it may involve whether visibility, influence, or access is fairly distributed across communities. A second perspective, procedural fairness [9, 10], shifts attention from outcomes to the algorithmic process itself, asking whether individuals or groups are treated symmetrically by the underlying mechanism. For example, this might involve whether a ranking algorithm affords each item a comparable opportunity to be surfaced. A further tension arises between individual fairness and group fairness. Individual fairness [11, 12] requires that similar individuals receive similar treatment, while group fairness [12, 13] enforces statistical parity across predefined groups. These notions are often mathematically incompatible, forcing designers to make explicit normative choices. In this thesis, fairness is primarily studied at the group level within graph-based systems. Our work on PageRank (Paper D) aims to enforce target distributions of importance across groups of nodes, while our study of hitting times (Paper E) seeks to equalize accessibility between disparate groups in a network.

Diversity, by contrast, focuses on promoting heterogeneity and breadth. In algorithmic systems, diversity can improve user satisfaction by offering varied choices, encouraging exploration and serendipity, and mitigating filter bubbles and echo chambers. Like fairness, diversity admits multiple mathematical formulations, each capturing a different aspect of heterogeneity. Distance-based formulations quantify diversity through pairwise dissimilarity [14, 15, 16], often assuming items lie in a metric space and measuring diversity as the sum or minimum of pairwise distances. Coverage-based formulations [17, 18] instead emphasize representational breadth, defining a set as diverse if it spans many topics, attributes, or viewpoints, and naturally connecting to classical problems such as set cover and facility location. In interactive or sequential environments, sequential diversity becomes essential, as diversity must be balanced over time while accounting for uncertainty in user engagement. This thesis explores diversity across several such settings. In interactive recommendation systems (Papers A and B), we study distance-based and coverage-based diversity in sequential and streaming contexts, whereas in our work on densest subgraphs (Paper C), diversity is an explicit constraint requiring dense communities to exhibit heterogeneous edge types.

While fairness and diversity are distinct objectives, they often operate in concert [19, 20]. Increasing diversity can promote fairness by broadening exposure and preventing dominant groups or viewpoints from monopolizing attention. Conversely, fairness constraints can enhance diversity by ensuring that minority groups are adequately represented. At the same time, these objectives can conflict. Maximizing diversity alone may overemphasize a small number of highly dissimilar items while neglecting equitable treatment across the full population, while strict fairness constraints may reduce overall diversity when groups exhibit homogeneous preferences.

In the following sections, we introduce the five papers included in this thesis. For each paper, we provide the relevant technical background and summarize our contributions.

1.1 Diversity Maximization

1.1.1 Background

Diversity maximization is a classical algorithmic problem, and it has been studied since the 1980s. The canonical version of this problem is essentially a subset selection problem, i.e., given a ground set of items, the goal is to select a subset subject to constraints that maximizes a diversity measure. The most basic constraint is the cardinality constraint, and the diversity measure typically refers to a distance function or a coverage function.

We are given a ground set of items U and a ground set of topics $\mathcal{C} = \{c_1, c_2, \dots, c_d\}$. A non-negative distance function $d : U \times U \rightarrow \mathbb{R}_+$ maps any pair of items to a non-negative real value, capturing the dissimilarity between items. A labeling function $c : U \rightarrow 2^{\mathcal{C}}$ assigns each item to a set of topics it covers. The diversity maximization problem asks to find a subset $S \subseteq U$ of size $|S| = k$ that maximizes diversity. Diversity can be measured in several ways: the sum of pairwise distances among selected items, the minimum pairwise distance, or the coverage of topics. We formally define these three diversities below.

$$\text{MAX-MIN Diversification: } \max_{S \subseteq U, |S| \leq k} \min_{x, y \in S} \{d(x, y)\} \quad (1.1)$$

$$\text{MAX-SUM Diversification: } \max_{S \subseteq U, |S| \leq k} \sum_{x, y \in S} d(x, y) \quad (1.2)$$

$$\text{MAX-COV Diversification: } \max_{S \subseteq U, |S| \leq k} \bigcup_{x \in S} c(x) \quad (1.3)$$

Hardness and Approximation Algorithms To the best of the author's knowledge, the MAX-MIN diversification problem is first studied by Kuby [21] in 1987, while it is later proved to be NP-hard even if the distance function d satisfies the triangle inequality by Erkut [22] in 1990. Later in 1994, Ravi et al. [15] prove that if the distance function d is not required to satisfy the triangle inequality, then there is no polynomial time relative approximation algorithm for the problem unless P=NP. Given the inapproximability results, research on the MAX-MIN diversification problem focuses on situations where the distance d satisfies the triangle inequality. In their paper, Ravi et al. [15] propose a greedy algorithm that first selects two items furthest apart from each other, then greedily adds the items that maximize the minimum distance from items in the chosen subset. This greedy algorithm provides a 2-approximation to the MAX-MIN diversification problem, and the approximation ratio is tight unless P=NP.

The MAX-SUM diversification problem is proven to be NP-hard even in metric spaces by Hansen and Moon [23]. Regarding approximability, unlike the MAX-MIN diversification problem, polynomial-time approximation algorithms exist for the MAX-SUM diversification problem in metric spaces. To name a few, Hansen and Moon [23] introduce a 1/4-approximation greedy algorithm; Borodin et al. [14] and Gollapudi and Sharma [16] both propose 1/2-approximation greedy algorithms under

a cardinality constraint; and Borodin et al. [14] also present a $1/2$ -approximation local search algorithm for general matroid constraints. More recently, Cevallos et al. [24] develop an algorithm for the cardinality constraint that provides a PTAS (Polynomial-Time Approximation Scheme, which we introduce in Section 2.1) when the distance function d is extended to a more general type known as negative type.

The MAX-COV diversification problem is also NP-hard since it is a variant of the maximum coverage problem. The objective function of this problem is submodular, and thus, according to Nemhauser et al. [25], a classic greedy algorithm provides a $(1 - 1/e)$ approximation.

1.1.2 Contributions of Paper A

The aforementioned canonical research problems formulate diversity maximization as a subset-selection task and overlook the importance of item ordering within the selected set. However, in real-world user–system interaction settings such as recommender systems, item order plays a critical role due to position-dependent user engagement.

The work most closely related to ours is the foundational study by Carbonell and Goldstein [26], which introduces Maximal Marginal Relevance (MMR). MMR greedily selects items by balancing query relevance with diversity relative to previously selected items. Although MMR has become ubiquitous in practice, it offers no formal approximation guarantees. Moreover, the resulting ranking is merely a by-product of the greedy selection process rather than an explicit optimization objective.

A further limitation shared by most existing diversification approaches is the assumption that users examine a fixed number of items k . In practice, user interaction is sequential, and users may stop at any position depending on satisfaction or fatigue. Coppolillo et al. [27] are the first to model such sequential user behavior in recommender systems. They formalize sequential diversity through the EXPLORE algorithm, modeling user continuation with Weibull distributions. However, their approach is purely empirical and provides no approximation guarantees.

We mitigate these limitations by introducing a theoretical framework for sequential diversification that models position-dependent user engagement. Users examine items sequentially and may stop at any position, with continuation probabilities governing their behavior. We define diversity as the expected diversity experienced by a user, computed over all possible stopping points. This formulation naturally integrates relevance and diversity through continuation probabilities and the underlying diversity measure. We show that the resulting optimization problems are NP-hard and develop approximation algorithms with constant-factor guarantees for both sum-diversity and coverage-diversity objectives.

1.2 Streaming Stochastic Coverage Maximization

Paper B extends the stochastic diversification model introduced in paper A to streaming settings, with a focus on coverage diversity. Since the objective is submodular, we first

review the fundamentals of submodular maximization and then conclude by outlining the main contributions of this paper.

1.2.1 Background

Submodular Optimization In the offline setting, given a ground set of items U , the submodular maximization problem selects a subset $S \subseteq U$ subject to some constraint, such that a given submodular function f is maximized. For this thesis, we focus on the case where f is monotone, and omit the discussion for the non-monotone cases.

The most basic version of this problem is when S is required to satisfy a cardinality constraint, i.e., $|S| \leq k$ for some integer k . For this simple case, Nemhauser et al. [25] propose a standard greedy algorithm, which selects at each step the element yielding the largest incremental in f , and provides a $1 - 1/e$ approximation. Feige [28] prove that improving the ratio $1 - 1/e$ is NP-hard, while Nemhauser and Wolsey [29] show that any improvement over $1 - 1/e$ requires an exponential number of queries in the value oracle setting.

When S is required to satisfy a matroid constraint, Fisher et al. [30] show that the standard greedy algorithm that selects the best elements while maintaining independence is a $1/2$ -approximation. This ratio is later improved to the optimal $1 - 1/e$ via the *continuous greedy algorithm* by Calinescu et al. [31] and Vondrák [32]. Filmus and Ward [33] propose a randomized non-oblivious local search algorithm that also achieves the optimal $1 - 1/e$ approximation, and converges in polynomial time.

Streaming Submodular Maximization In the streaming setting, the items of the ground set U arrive sequentially, that is, one item becomes available at each time step. Equivalently, the ground set can be viewed as ordered in an arbitrary manner, and any streaming algorithm must process U in this order. At any point during the stream, the algorithm must be able to output a solution S . In the following, we focus on the case where the submodular function is monotone.

For the streaming submodular maximization problem, the solution set is further required to satisfy given constraints, such as a cardinality constraint $|S| \leq k$ or a matroid constraint $S \in \mathcal{M}$ for some matroid $\mathcal{I} = (U, \mathcal{M})$.

The first streaming algorithm for the monotone submodular maximization problem subject to a cardinality constraint with provable guarantees is SIEVE-STREAMING, proposed by Badanidiyuru et al. [34]. SIEVE-STREAMING achieves a one-pass $1/2 - \epsilon$ approximation with time complexity $\mathcal{O}(n \log(k)/\epsilon)$ and space complexity $\mathcal{O}(k \log(k)/\epsilon)$ for an input stream of length n . Kazemi et al. [35] later improve the space complexity to $\mathcal{O}(k)$ and prove that $1/2$ is tight. Norouzi-Fard et al. [36] further show that $1/2$ is optimal when elements in the stream arrive in an arbitrary order. Subsequent works [36, 37] improve the approximation ratio beyond 0.5 under the assumption that elements arrive in a random order. We do not discuss these works further, as this thesis focuses exclusively on the arbitrary-order setting.

Beyond the cardinality constraint, the streaming submodular maximization problem has also been studied under matroid and matchoid constraints. Chakrabarti and Kale [38] propose a $\frac{1}{4p}$ -approximation algorithm when the constraint is the intersection

of p matroids. Later, Chekuri et al. [39] introduce a streaming-greedy algorithm that extends the constraint to the more general p -matchoid setting while maintaining the same approximation ratio. Feldman et al. [40] subsequently propose a randomized version of the streaming-greedy algorithm, which is three times faster while achieving the same $1/4p$ approximation guarantee.

1.2.2 Contributions of Paper B

The aforementioned streaming submodular maximization algorithms are designed to produce solutions only at the end of the stream. In applications that require anytime or on-demand responses, existing methods cannot guarantee approximation bounds at arbitrary query times.

We study a streaming setting in which content arrives online, and users may access the system at any time. Users click on each recommended item with a certain probability, and our goal is to maximize the expected number of topics covered by the items a user clicks on by the end of the interaction.

We introduce algorithms that maintain valid approximation guarantees throughout the stream while accounting for stochastic user behavior. Our algorithms achieve different competitive ratios depending on memory constraints. With memory linear in the stream length, they match offline guarantees. With sublinear memory, we develop two algorithms that make multiple concurrent predictions about the user behavior and output the best prediction greedily, both achieving constant competitive ratios. These results demonstrate that strong theoretical guarantees can be preserved in settings with limited memory and uncertain user behavior.

1.3 Diversified Densest Subgraph Discovery

For paper C, we shift our focus from sets to graphs and move from explicit diversity maximization to diversity imposed as a structural constraint. In many real-world networks, edges carry rich semantic information. Relationships in social networks may represent friendship, family ties, or professional connections, while citations between papers may indicate agreement, disagreement, or extensions of prior work. Ignoring such attributes when searching for dense subgraphs often leads to overly homogeneous communities that fail to capture important forms of heterogeneity. Motivated by these observations, this paper studies the densest subgraph discovery problem under edge color constraints, which can be naturally interpreted as diversity constraints on edge types. We first briefly review the classical densest subgraph problem and then describe our main contributions.

1.3.1 Background

Given an undirected graph $G = (V, E)$ with n vertices and m edges, the unconstrained densest subgraph problem asks for a subset of vertices $S \subseteq V$ that maximizes the ratio $|E[S]|/|S|$, where $E[S]$ denotes the edges induced by S . This problem can be solved in polynomial time using Goldberg's max flow algorithm [41] or by solving

the linear programming formulation introduced by Charikar [42]. Charikar’s greedy peeling algorithm achieves a 2 approximation in linear time $\mathcal{O}(n + m)$ by iteratively removing vertices of minimum degree.

Subsequent work has explored several extensions of this problem. Khuller and Saha [43] generalize exact algorithms to directed graphs and showed that size-constrained variants, such as finding a densest subgraph with exactly k vertices, are NP hard. Andersen and Chellapilla [44] study the densest subgraph with at least k vertices and proposed a linear time algorithm with a $\frac{1}{3}$ approximation guarantee.

When diversity or fairness constraints are introduced, the problem becomes significantly more challenging. Anagnostopoulos et al. [45] prove that the fair densest subgraph problem with binary node colors is NP hard. They further show that approximating this problem in polynomial time is at least as hard as approximating the densest subgraph with at most k vertices, for which no constant factor approximation is known. For the special case of fair graphs, where fairness requires an equal number of vertices from each color class, they obtained a 2-approximation. Miyauchi et al. [46] extend this line of work to multiple node colors through the Densest Diverse Subgraph Problem and achieve an $\Omega(1/\sqrt{n})$ approximation that is independent of the number of colors.

1.3.2 Contributions of Paper C

Prior work on fairness and diversity in densest subgraph discovery has focused exclusively on node-level constraints, typically enforcing proportional representation across vertex groups. Paper C is the first to introduce edge-color constraints into the densest subgraph discovery problem. The edge-color constraint setting captures a wide range of real-world scenarios, including social networks where diversity corresponds to multiple types of relationships, multilayer networks with different interaction modalities, and biological networks with distinct types of connections. The paper establishes that the problem is NP-complete even when there are only two edge colors. It also presents linear time $\mathcal{O}(1)$ -approximation algorithms for the variant that requires at least h distinct edge colors, under the assumption that the input graph is everywhere sparse.

1.4 Fairness-Aware PageRank

1.4.1 Background

The concept of PageRank fairness was recently introduced by Tsioutsoulis et al. [47], who define it as the fair allocation of PageRank weight (formally defined in Section 2.3.2) among nodes. To be specific, let \mathbf{p} be the PageRank vector of a given graph $G = (V, E)$, where the PageRank weight of node $u \in V$ is equal to \mathbf{p}_u . According to Tsioutsoulis et al. [47], a PageRank algorithm is ϕ -fair if the fraction of the total PageRank weight assigned to the members of a target group is exactly ϕ . To achieve ϕ -fairness, the authors proposed several methods. The FSPR algorithm modifies the restart vector and achieves ϕ -fairness without changing the graph structure, while LFPR modifies both the restart vector and the transition matrix

to ensure fair behavior at each vertex, i.e., fair probabilities to visit all groups from each node. Tsioutsoulouklis et al. [48] later proposed an alternative definition, where PageRank fairness requires the total PageRank weight of a target group to meet a threshold ϕ . To achieve this threshold, they propose a link-recommendation algorithm that adds edges to the graph, increasing the PageRank weight of the target group and thereby maximizing the fairness gain.

Beyond these two works that explicitly optimize PageRank fairness, some studies implicitly promote it. For instance, Fairwalk [49] and Crosswalk [50] focus on fairness in graph embeddings by adjusting random walk probabilities. Their methods revise the transition probabilities of a random walker moving to neighboring nodes, ensuring that each node distributes its influence fairly among neighbors belonging to different groups. Although not explicitly framed in terms of PageRank fairness, such approaches indirectly increase PageRank fairness at the individual-node level and, as a consequence, can also enhance fairness across the entire graph.

1.4.2 Contributions of Paper D

The above-mentioned prior works demonstrate multiple pathways to achieving fair PageRank, which can be broadly categorized as (i) modifying the restart probability, (ii) modifying the restart vector, (iii) adding or recommending edges, and (iv) reweighting existing edges. While each approach has its merit, we focus on edge reweighting for several principled reasons.

First, edge reweighting provides interventions that are intrinsic to the graph structure and directly translatable to actionable operations. In social networks, for instance, edge weights can be adjusted by prioritizing or demoting content from a target source, which are operations that platforms can readily implement. Second, while link recommendation strategies like those in Tsioutsoulouklis et al. [48] are also structure-based, they introduce practical challenges by depending on user acceptance of recommended links, creating uncertainty that complicates reliable deployment. Third, strategies that modify PageRank hyperparameters, such as the restart vector adjustments in FSPR and LFPRs, are extrinsic to the graph structure and often lack clear operational meaning. For example, in personalized PageRank, the restart vector encodes user interests and should not be artificially manipulated solely for fairness objectives.

Our approach, therefore, focuses on edge reweighting as a principled, graph-intrinsic method for fairness intervention. This method preserves the interpretability of PageRank parameters while enabling practical deployment through structural modifications that remain faithful to the original network. Specifically, we modify transition probabilities in the PageRank matrix to bring the resulting group-wise PageRank weight as close as possible to a target. Unlike methods that enforce local fairness at each vertex, we target group-level fairness across the entire graph, enabling smaller modifications to the transition matrix. We formulate the objective as a non-convex function over a convex feasible set, incorporating practical constraints such as preserving the original network topology and bounding the magnitude of weight changes. We solve this optimization problem using efficient projected gradient descent, guaranteeing convergence to a stationary point.

1.5 Reducing Structural Bias via Hitting-Time Minimization

1.5.1 Background

Real-world networks often exhibit structural segregation, a phenomenon where distinct groups form densely connected internal clusters but maintain only sparse connections between each other. This structural bias arises across many domains, including social media and content recommendation platforms, leading to concerning patterns such as polarized political discussions on Twitter, echo chambers in news consumption, and radicalization pathways in video-sharing platforms. In these settings, users predominantly interact with like-minded individuals and infrequently encounter diverse viewpoints. For example, in video-sharing or news recommendation platforms, the content graph may split into clusters of "harmful" and "neutral" content, or into communities reflecting opposing stances on controversial issues.

From a fairness perspective, such segregation raises concerns about equitable access to information across groups. If members of one group can easily navigate to another while the reverse requires significantly more steps, the network exhibits fundamental asymmetry in cross-group accessibility. This asymmetry can restrict user exposure to diverse viewpoints, reinforce polarization, and increase the risk of harmful content consumption. Mitigating structural bias is therefore an important objective for improving the fairness, safety, and informational value of recommendation systems.

Recent research has investigated algorithmic interventions that reduce structural bias by modifying network structure. One line of work considers edge rewiring, where existing recommendations are replaced to reduce exposure to harmful content. Fabbri et al. [51] show that it is NP-hard to find the optimal set of recommendations to rewire to reduce the segregation in a recommendation graph; they further prove that the problem cannot be approximated within any multiplicative factor. Coupette et al. [52] study the related problem of minimizing exposure to harmful content through edge rewiring. They prove the objective is submodular and proposed a greedy algorithm that achieves a $(1 - 1/e)$ -approximation. Another line of work focuses on introducing shortcut edges, i.e., edges not present in the original network but added as new recommendations, such as suggesting additional videos, articles, or users to follow. In this direction, Haddadan et al. [53] formulate the problem of maximizing bubble radius reduction as a submodular maximization task and solve it using a greedy approach.

The body of work discussed above relies on random-walk-based measures to quantify structural bias, such as bubble radius, inter-group hitting time, and segregation score. Other studies, however, define structural bias using different frameworks. For instance, Zhu et al. [54] model polarization through a polarization-disagreement index derived from opinion disagreement in the Friedkin-Johnsen model, and reduce this index by strategically adding edges to the network.

1.5.2 Contribution of Paper E

The existing works [51, 52, 53] have several key limitations. First, they aim to reduce structural bias rather than to minimize it directly. Second, they rely on specific assumptions about the graph or the random walk process. For instance, Fabbri et al. [51] assume the graph is d -regular, and Haddadan et al. [53] restrict the random walk to a bounded length.

Our work addresses these limitations and tackles the problem in a more general and direct way. We directly minimize the hitting times between groups without making assumptions about graph structure or the random walk process. We focus on general undirected graphs with uniform random walks and design algorithms that minimize both the average and the maximum hitting time between groups.

The rewiring and shortcut-edge technique can be applied to mitigate structural bias and polarization in real-world networks, particularly in recommender and content-navigation systems such as social media feeds, video platforms, and news aggregators. By strategically adding a small number of shortcut edges between otherwise weakly connected groups, the technique reduces the expected time it takes for users to transition from harmful content to neutral content. Importantly, this approach does not require altering or removing nodes in the graph, making it well-suited for deployment in real-world systems. From a societal perspective, it helps counter echo chambers and polarization by gently promoting exposure to diverse viewpoints, fostering healthier information ecosystems while preserving platform constraints and user experience.

1.6 Thesis Organization

This thesis is organized as follows. Chapter 2 presents the preliminaries that are required to understand the included papers. Chapter 3 presents our work on diversity maximization in interactive systems, covering both the sequential ranking problem (paper A) and the streaming coverage problem (paper B). Chapter 4 presents our study of diversity-constrained densest subgraph discovery (paper C), establishing connections between edge-constrained and node-constrained formulations. Chapter 5 presents our work on graph interventions for fairness, covering both PageRank fairness through edge reweighting (paper D) and hitting-time fairness through strategic edge additions (paper E). Chapter 6 concludes with a synthesis of our contributions, discusses broader implications for the design of fair and diverse algorithmic systems, and outlines directions for future research.

Together, these chapters demonstrate that fairness and diversity, properly formalized, can be pursued through principled algorithmic approaches with provable guarantees, contributing both to the theoretical foundations of algorithmic fairness and to practical system design.

Chapter 2

Preliminaries

This chapter provides the necessary background knowledge for understanding the technical content of this thesis. We will start with the basics of approximation algorithms for papers A, B, C, and E, followed by the definition of submodularity and matroids to understand paper B, and finally graph-theoretic notions needed for the papers.

2.1 Basics of Approximation Algorithms

Approximation algorithms [55, 56, 57] are a class of algorithms designed to tackle hard optimization problems for which no efficient method exists to compute the exact optimal solution, or for which computing the exact solution is prohibitively expensive, even in cases where the problem can be solved in polynomial time but the exact algorithm remains too costly for practical use at scale. Instead of guaranteeing optimality, approximation algorithms aim to produce solutions that are provably close to optimal while running in polynomial time. A *relative* approximation algorithm guarantees a solution within a *multiplicative* factor of the optimal value for every instance of the problem, whereas an *absolute* approximation algorithm guarantees a solution within an *additive* constant of the optimal value for every instance. In this thesis, unless otherwise specified, the term *approximation algorithm* refers to a *relative* approximation algorithm.

Formally, an α -approximation algorithm for an optimization problem is a polynomial-time algorithm that, for any instance of the problem, returns a solution whose value is within a factor α of the optimal value. Let OPT denote the value of an optimal solution and ALG the value returned by the algorithm. Then, for a maximization problem, it holds

$$\text{OPT} \geq \text{ALG} \geq \alpha \text{OPT},$$

with $\alpha < 1$, and for a minimization problem, it holds

$$\text{OPT} \leq \text{ALG} \leq \alpha \text{OPT},$$

with $\alpha > 1$. We call the quantity α the approximation ratio.

Competitive Ratio Competitive analysis [58, 59] is a framework for evaluating online algorithms, which must make irrevocable decisions as input arrives incrementally, without knowledge of future arrivals. This lack of foresight means that decisions appearing optimal at the moment may prove suboptimal later. In the worst case, an adversary can craft future inputs to exploit every decision the algorithm makes. To assess an online algorithm's quality, we compare it against an offline algorithm with full knowledge of the input sequence in advance. The *competitive ratio* quantifies this worst-case performance gap over all possible input instances.

Formally, for an input instance σ , let $\text{ALG}(\sigma)$ and $\text{OPT}(\sigma)$ denote the objective values achieved by the online and optimal offline algorithms, respectively. For a maximization problem, an algorithm is α -*relatively competitive* if there exists a constant k such that $\text{ALG}(\sigma) \geq \alpha \text{OPT}(\sigma) + k$ for every possible input σ , and we call α the *relative competitive ratio*. An algorithm is k -*additively competitive* if $\text{ALG}(\sigma) \geq \text{OPT}(\sigma) - k$ for every possible input σ . For minimization problems, the inequalities are reversed. In this thesis, unless otherwise specified, *competitive ratio* refers to the *relative* competitive ratio.

Optimality of Approximation and Approximability [56] If we call an approximation ratio α *optimal*, it means there is no polynomial-time relative approximation algorithm that can provide a performance guarantee better than α unless $\text{P} = \text{NP}$. A problem is called *inapproximable* (or *hard to approximate*) if there exists no polynomial-time relative approximation algorithm for it unless $\text{P} = \text{NP}$. More formally, a problem is said to be *inapproximable within factor c* if there is no polynomial-time algorithm that guarantees a solution within a multiplicative factor c of the optimal value, unless $\text{P} = \text{NP}$.

Tightness of Approximation An approximation ratio α is said to be *tight* if the worst-case performance guarantee of an algorithm coincides with its performance on a specific instance. Specifically, the algorithm guarantees a solution within a factor α of the optimal for all problem instances (upper bound), and there exists at least one instance for which the algorithm achieves exactly this ratio α (lower bound). When these bounds coincide, the analysis precisely characterizes the algorithm's worst-case behavior, and the approximation ratio cannot be improved without modifying the algorithm.

Polynomial-Time Approximation Scheme (PTAS) [60] A *polynomial-time approximation scheme* is an algorithmic scheme with the following property: for every fixed $\epsilon > 0$ there exists an algorithm A_ϵ that runs in time polynomial in the input size n (the degree of the polynomial may depend arbitrarily on $1/\epsilon$) and returns a $(1 + \epsilon)$ -approximation for minimization problems, or a $(1 - \epsilon)$ -approximation for maximization problems.

2.2 Submodular Optimization and Matroid Constraint

Given a set U of items, a set function $f : 2^U \rightarrow \mathbb{R}_{\geq 0}$ maps a subset of items to a non-negative value. A typical example is the coverage function which calculates the total number of unique topics spanned by a specific collection of news items. A set function f is monotonically increasing if for any $S \subseteq T \subseteq U$, we have

$$f(S) \leq f(T).$$

Similarly, f is monotonically decreasing if for any $S \subseteq T \subseteq U$, we have

$$f(S) \geq f(T).$$

2.2.1 Submodular Set Function

A function f is submodular [61, 62, 63] if for all $S \subseteq T \subseteq U$ and for all $\{e\} \in U \setminus T$ we have

$$f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T). \quad (2.1)$$

An equivalent definition of submodularity is that for all $S, T \subseteq U$, we have

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T). \quad (2.2)$$

Many practical problems naturally exhibit submodularity. For instance, in data summarization, the goal is to choose a small set of items that well represents a large dataset, while in information maximization, the objective is to select a set of seed nodes that maximizes the spread of information across a network. Intuitively, these problems are submodular because they exhibit a diminishing returns property: as the selected set grows, newly added items increasingly overlap with what has already been covered, contributing less marginal benefit. This structure aligns directly with the definition of submodularity, providing a principled foundation for designing efficient algorithms.

2.2.2 Supermodular Set Function

By reversing the inequality in Equation (2.2), we obtain the definition of supermodularity [64]. A set function f is supermodular if for all $S, T \subseteq U$, we have

$$f(S) + f(T) \leq f(S \cup T) + f(S \cap T). \quad (2.3)$$

In contrast to submodularity, supermodularity exhibits increasing marginal gains: adding an element to a larger set yields greater benefit than adding it to a smaller set.

Weakly- α -Supermodular Set Function A non-negative monotonically decreasing set function f is said to be *weakly- α -supermodular* [65] if there exists $\alpha \geq 1$ such that for any two sets $S, T \subseteq U$

$$f(S) - f(S \cup T) \leq \alpha \cdot |T \setminus S| \cdot \max_{\{e\} \in T \setminus S} [f(S) - f(S \cup \{e\})]. \quad (2.4)$$

The supermodularity states that if adding $T \setminus S$ is beneficial to minimizing f , then there is an element $\{e\} \in T \setminus S$ that contributes at least a fraction of that benefit. Liberty and Sviridenko [65] prove that a non-increasing non-negative supermodular function f is weakly- α -supermodular with parameter $\alpha = 1$.

2.2.3 Oracle Model

In submodular optimization, it is standard to assume access to a value oracle [25, 66], which is a black box that returns $f(S)$ for any $S \subseteq U$. Each oracle query is assumed to require one unit of time, and the time complexity of an algorithm is therefore measured by the number of oracle calls it makes, relative to the ground set size in the offline setting, and to the stream length in the streaming setting. The use of a value oracle is natural because submodular functions arise in diverse domains with vastly different evaluation procedures, and the oracle model captures this generality by enabling a unified analysis that is agnostic to the internal structure of f .

2.2.4 Matroids

In combinatorics, we frequently deal with constrained optimization problems, where the solution is a subset of elements chosen from a ground set that must satisfy some given constraints. A matroid [67, 68] specifies a particular type of constraint by introducing a structure of "independence," meaning that valid solutions are required to satisfy an independence condition. For instance, in the case of a graphic matroid, the selected edges must not form a cycle in the underlying graph.

Formally speaking, a *matroid* is a pair $\mathcal{I} = (U, \mathcal{M})$, where U is called the *ground set*, and \mathcal{M} is called the *independent sets*. \mathcal{M} is a set of the subsets of U , each element in \mathcal{M} is called a independent set. For \mathcal{I} to be a matroid, \mathcal{M} must satisfy three axioms:

- (I_1) (hereditary) $\emptyset \in \mathcal{M}$
- (I_2) (downward closure) For any $Y \in \mathcal{M}$, if $X \subset Y$, then $X \in \mathcal{M}$
- (I_3) (augmentation property) For any $X, Y \in \mathcal{M}$, if $|X| < |Y|$, then there exists $e \in Y$ such that $X \cup \{e\} \in \mathcal{M}$

A matroid generalizes the concept of "independence" from vector spaces to more abstract settings where the ground set is arbitrary rather than limited to vectors. Analogously to the vector space, the notions of rank, basis, and span are also defined. Specifically, the rank of a matroid $\mathcal{I} = (U, \mathcal{M})$ is a function $2^U \rightarrow \mathbb{R}_+$ denoted r or $r_{\mathcal{M}}$ such that

$$r(S) = \max\{|X| : X \subseteq S, X \in \mathcal{M}\} \text{ for any } S \subseteq U$$

Axiom (I_2) implies that all maximal (inclusion-wise) independent sets have the same cardinality. A maximal independent set is called a *base* (or *basis*) of the matroid. Given a matroid $\mathcal{I} = (U, \mathcal{M})$ and any set $S \subseteq U$, the *span* of S , denoted as $span$, is defined as

$$span(S) = \{e \in U : r(S \cup \{e\}) = r(S)\}$$

2.3 Graphs and Graph Measures

In this section, we introduce graphs and some measures and problems that are particularly relevant for analyzing graphs.

2.3.1 Graphs

A graph is defined as $G = (V, E)$, where V is a finite set of *vertices* with $|V| = n$, and $E \subseteq V \times V$ is a set of *edges* connecting pairs of vertices, with $|E| = m$. Graphs provide a natural way to model many real-world systems, such as friendships in social networks or protein interactions in biology.

Depending on the problem being modeled, graphs can be directed or undirected, weighted or unweighted. We use $w(u, v)$ to denote edge weight of edge (u, v) , with $w(u, v) \in \{0, 1\}$ for unweighted graphs and $w(u, v) \in \mathbb{R}_{\geq 0}$ for weighted graphs. We use $d(u)$ to denote the degree of a vertex u , which is defined as the number of edges connected to it in unweighted graphs and the sum of edge weights connected to it in weighted graphs.

Graphs can be represented and stored using different data structures:

- The *adjacency matrix* $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$, where $\mathbf{A}_{uv} = 1$ (or $w(u, v)$) if there is an edge from vertex u to vertex v , and 0 otherwise.
- The *adjacency list*, which stores for each node a list of its neighbors.

Given a subset S of the vertices V , we denote by $G(S) = (S, E[S])$ the node-induced subgraph by S , where $E[S] = \{\{u, v\} \in E \mid u, v \in S\}$.

2.3.2 PageRank

PageRank is a well-known algorithm popularized by Larry Page and Sergey Brin in the late 1990s [69]. It was the foundation of Google's search engine. The algorithm takes a graph $G = (V, E)$ as input, and outputs a PageRank vector \mathbf{p} , with \mathbf{p}_u denoting the importance of node u in the graph. The intuition of the algorithm is to imagine a random surfer moving from one vertex to another, following the edges, and the rank of a vertex represents how likely the surfer is to land there.

Given a directed graph $G = (V, E)$, let \mathbf{P} denote its transition matrix, where \mathbf{P}_{uv} is the probability that a random surfer moves from node u to node v in a single step. The matrix \mathbf{P} is obtained by normalizing the adjacency matrix, with special handling for sink nodes. Formally,

$$\mathbf{P}_{uv} = \begin{cases} \frac{w(u,v)}{d(u)}, & \text{if } d(u) \neq 0, \\ \frac{1}{|V|}, & \text{otherwise.} \end{cases} \quad (2.5)$$

Thus, for non-sink nodes, the random surfer's transition probability is proportional to the normalized edge weight, while for sink nodes, the surfer jumps uniformly to any node in the graph.

PageRank computes a vector \mathbf{p} , which represents the stationary distribution of a random walk on the graph G . The vector \mathbf{p} satisfies the equation

$$\mathbf{p}^\top = (1 - \gamma)\mathbf{p}^\top \mathbf{P} + \gamma \mathbf{v}^\top \quad (2.6)$$

where γ denotes the restart probability, which represents the probability that the random surfer restarts at any step. The vector \mathbf{v} is the restart vector, i.e., a probability distribution over V that determines the restart probabilities at each vertex. By convention, γ is often set to 0.15, and \mathbf{v} is often chosen as the uniform distribution.

By solving Equation (2.6), the PageRank vector \mathbf{p} can be computed as

$$\mathbf{p}^\top = \gamma \mathbf{v}^\top [\mathbf{I} - (1 - \gamma)\mathbf{P}]^{-1}$$

Computationally, the PageRank score vector \mathbf{p} can be computed efficiently using the power iteration method [70], which repeatedly applies Equation (2.6) until convergence.

2.3.3 Hitting Time

The concept of *hitting time* [71] is crucial in the study of random walks on graphs. Given a graph $G = (V, E)$ with transition matrix \mathbf{P} , we are interested in random walks on the graph G according to \mathbf{P} . Intuitively, the hitting time from a vertex u to v is the expected number of steps a random walker starting at node u takes to reach node v for the first time, and it captures how “accessible” node v is from u .

We denote the hitting time from vertex u to vertex v as $H(u, v)$. Let $\tau_v(u)$ denote the random variable representing the first time step at which a random walk, starting from vertex u , visits vertex v . The hitting time is defined as $H(u, v) = \mathbb{E}[\tau_v(u)]$, where the expectation is taken over all possible random walk trajectories starting from vertex u .

There are two related measures to hitting time, the commute time $c(u, v)$ and cover time C , defined as follows

- *Commute time* $c(u, v)$: the expected time required for a random walk to start at vertex u , hit v for the first time, then hit u . This can be represented as $c(u, v) = H(u, v) + H(v, u)$.
- *Cover time* C : let $C_u(G)$ denote the expected time required for a random walk to start at vertex u , and visit all nodes in the graph G for at least one time. The cover time $C = \max_{u \in V} C_u(G)$. The cover time of any graph with n nodes is *upper bounded* by $2n(n - 1)^2$ [72].

2.3.4 Densest Subgraph Discovery

Given a graph $G = (V, E)$, the densest subgraph discovery problem [41] aims to find a subset $S \subseteq V$ of vertices such that the induced subgraph $G[S]$ maximizes a density measure. Densest subgraphs contain tightly interconnected substructures, making

them valuable for applications including community detection, fraud detection, and web-graph analysis.

One of the most commonly used density measures is *edge density*, defined as $d(S) = \frac{|E[S]|}{|S|}$, where $E[S] = \{(u, v) \in E \mid u, v \in S\}$ represents the edges with both endpoints in S .

Computational Complexity and Algorithms The unconstrained densest subgraph problem (finding $S^* = \operatorname{argmax}_{S \subseteq V} d(S)$) is solvable in polynomial time via max-flow computations [41, 73], even though there are exponentially many subgraphs to consider. For efficient approximate solutions, Asahiro et al. [74], Charikar [42] show that a linear-time greedy *peeling* algorithm that iteratively removes the node of smallest degree and returns the best solution encountered achieves a 2-approximation ratio. More recently, Chekuri et al. [75] provide an almost linear-time flow-based $(1 + \epsilon)$ -approximation algorithm and analyze an iterative peeling algorithm by Boob et al. [76], proving its convergence to optimality.

Size-Constrained Variants Research has also investigated densest subgraph discovery under cardinality constraints, seeking subgraphs with at most k nodes (Dam k S), at least k nodes (Dal k S), or exactly k nodes (D k S). The D k S problem is NP-hard, Khot [77] show that there does not exist any PTAS for the D k S problem under a reasonable complexity assumption. The best-known approximation ratio is $O(n^{1/4})$ [78]. The Dam k S problem is also NP-hard, and Andersen and Chellapilla [44] show that it is nearly as hard to approximate as the D k S problem. Khuller and Saha [43] show that an α -approximation for Dam k S yields an $\alpha/4$ -approximation for D k S.

The Dal k S problem seeks the densest subgraph containing at least k vertices and is also NP-hard [43], but easier to approximate than D k S. Andersen and Chellapilla [44] design a linear-time $1/3$ -approximation algorithm based on greedy peeling, while Khuller and Saha [43] provide two $1/2$ -approximation algorithms using flow computations and linear programming, respectively. However, it was left open as to whether or not this problem is NP-complete.

2.4 Streaming Algorithms

Data is generated at high velocity, and in many cases, accessing the entire dataset is infeasible. This may occur because the ground set is too large to fit into memory or because the data arrives as a continuous stream at a rate that makes complete storage impractical (e.g., sensor measurements, real-time traffic data). Even when the full dataset is available, repeated querying can be computationally expensive. Streaming algorithms are well-suited to such settings, as they require limited memory, typically much smaller than the ground set, and process each incoming item efficiently in real time.

Given memory constraints, a streaming algorithm can store and process only a small fraction of the data at any given time [79]. The available memory must accommodate both the current solution and auxiliary information necessary for producing

a high-quality approximation. For each incoming item, the algorithm must decide whether to retain it, discard it, or use it to replace an existing item in memory.

The performance of a streaming algorithm is typically evaluated according to four criteria: the number of passes over the data stream, memory usage, running time, and approximation ratio. In Paper B, running time is measured by the number of oracle calls to evaluate a submodular function. In other applications, the appropriate measure is context-dependent. For example, finding the maximum integer in a data stream requires a single pass, memory for one element, time proportional to stream length, and achieves an exact solution.

2.5 The k -Center Problem

The k -center problem is a classical clustering problem that is well studied in combinatorial optimization. We are given a large dataset containing both similar and dissimilar data points. Our goal is to organize these data into groups by selecting representative data points as cluster centers. Each data point in the dataset is then assigned to its closest cluster center, which naturally defines the clustering structure.

Metric Distance [80] The standard k -center problem is defined in a metric space. Let U be a set of n points in a metric space. We are given a distance function $d : U \times U \rightarrow \mathbb{R}_{\geq 0}$ that computes the distance between any two points $x, y \in U$. We assume d is a metric, meaning that it satisfies the following properties for any $x, y, z \in U$:

(I_1) (Positive semidefiniteness) $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$.

(I_2) (Symmetric) $d(x, y) = d(y, x)$.

(I_3) (Triangle inequality) $d(x, y) + d(y, z) \geq d(x, z)$.

k -Center Problem [81] Given a set of points U and a distance $d : U \times U \rightarrow \mathbb{R}_{\geq 0}$, find a set $S \subseteq U$ of k points in order to minimize the maximum distance of any point of U to its closest center in S . That is,

$$\min_{S \subseteq U, |S|=k} \max_{v \in U} d(v, S),$$

where $d(x, S) = \min_{s \in S} d(x, s)$.

The k -center problem is NP-hard [82], but it admits efficient approximation algorithms. Hsu and Nemhauser [83] prove that in metric space, no polynomial-time algorithm can achieve an approximation factor better than 2 unless $P=NP$. While several algorithms achieve the optimal approximation ratio of 2, most notably those by Hsu and Nemhauser [83], Plesník [84], Dyer and Frieze [85], and Hochbaum and Shmoys [86], the approach by Gonzalez [81] stands out for its elegant simplicity. Known as the furthest-first traversal, Gonzalez's algorithm iteratively selects the point at the maximum distance from the current set of centers until k points are established.

This greedy strategy is not only computationally efficient with a running time of $O(nk)$, but it also serves as a fundamental benchmark in clustering theory.

2.6 The Asymmetric k -Center Problem

When the distance function in Section 2.5 does not satisfy the symmetry property, but non-negativity and the triangle inequality still hold, we call this space a *quasi-metric* space [87]. The asymmetric k -center problem is the k -center problem defined on a quasi-metric space.

Asymmetric k -Center Problem [88] Given a set of points U in a quasi-metric space and a distance function $d : U \times U \rightarrow \mathbb{R}_{\geq 0}$ that satisfies the triangle inequality, find a set $S \subseteq U$ of k points that minimizes the maximum distance from any point in U to its closest center in S . That is,

$$\min_{S \subseteq U, |S|=k} \max_{v \in U} d(v, S),$$

where $d(v, S) = \min_{s \in S} d(v, s)$.

The asymmetric k -center problem is no easier than the standard k -center problem and is thus also NP-hard. Chuzhoy et al. [89] show that there exists a constant $\alpha > 0$ such that the asymmetric k -center problem cannot be approximated within a factor of $\log^* n - \alpha$ unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$. To approximate this problem, Panigrahy and Vishwanathan [88] proposed an $\mathcal{O}(\log^* n)$ -approximation algorithm, which was later improved by Archer [90] to $\mathcal{O}(\log^* k)$.

Chapter 3

Diversity Maximization in Interactive Systems

In this chapter, we examine two distinct interactive systems and explore various notions of diversity maximization for each. We use a news recommendation system as a representative model for both settings. In this common framework, both systems are probabilistic: when a user is presented with a news item, they examine it with a given probability or otherwise ignore it. Our goal is to maximize the expected diversity a user obtains by the time they cease interaction with the system. The two systems differ in their delivery mechanisms: the first is an offline system that, given a ground set of news items, presents the user with a shuffled list to be examined sequentially until quitting; the second is an online system where news items arrive in a streaming fashion, and the system must select a subset of items to present whenever a user submits a request.

This chapter summarizes papers A and B: in Section 3.1, we study the sequential diversity maximization problem for the first system as studied in paper A; in Section 3.2, we study the streaming diversity maximization problem for the second system as investigated in paper B. For each paper, we introduce the problem setting, provide the main theoretical results, and provide some intuition for the proofs.

3.1 [Paper A] Sequential Diversity Maximization

We consider a personalized user–item interaction system as defined below.

NEWS ITEMS: Given a ground set of n news items $U = \{1, \dots, n\}$ and a ground set of attributes \mathcal{X} , each news item i is associated with a probability p_i and a set of attributes $X(i) \subseteq \mathcal{X}$. The distance between two news items i and j is represented as $d(i, j)$, where $d : U \times U \rightarrow \mathbb{R}$ is a metric distance.

RECOMMENDER SYSTEM: The news recommendation system shuffles the ground set U via a permutation function $\pi : U \rightarrow U$, assigning item $\pi(i)$ to position i , and presents the shuffled list $O_{(\pi)} = (\pi(1), \pi(2), \dots, \pi(n))$ to the

user. We omit the subscript (π) and use O to denote a ranking of U when the context is clear.

USER–ITEM INTERACTION: The user examines the list $O_{(\pi)}$ from top to bottom. Upon examining item $\pi(i)$, the user has two options: they either accept it with probability $p_{\pi(i)}$ and proceed to the next item, or reject it and leave the system permanently. Since the probability p_i determines whether a user continues or quits, we call p_i the continuation probability.

3.1.1 Diversity Measures

Based on this interaction system, we study Sequential Sum Diversity (\mathcal{S}_+) and Sequential Coverage Diversity (\mathcal{S}_c). These measures are considered "sequential" as they quantify the expected diversity accumulated over the items viewed before a user stops. Our objective is to solve two corresponding problems: maximizing these measures by finding the best ordering of the ground set.

Definition 1 (Sequential sum diversity (\mathcal{S}_+)). *The sequential sum diversity of a sequence O , denoted by $\mathcal{S}_+(O)$, is defined as the expected sum of pairwise distances that the user accepts before quitting, where the expectation is taken with respect to the probability distribution over prefixes. Formally,*

$$\mathcal{S}_+(O) = \sum_{t=0}^n \mathbb{P}[\text{accept } O_t \text{ and quit}] \sum_{i,j \in O_t} d(i, j), \quad (3.1)$$

where $O_t = (\pi(1), \dots, \pi(t))$ represent the length t prefix of O , $O_0 = \emptyset$, and $\mathbb{P}[\text{accept } O_t \text{ and quit}] = \prod_{j=1}^t p_{\pi(j)}(1 - p_{\pi(t+1)})$ represents the probability that a user examines and accepts O_t , then proceeds to examine and subsequently reject $\pi(t+1)$.

We show that $\mathcal{S}_+(O)$ can be reformulated as

$$\mathcal{S}_+(O) = \sum_{i=1}^{n-1} p_{O_{i+1}} d(\pi(i+1), O_i), \quad (3.2)$$

where $p_{O_i} = \prod_{t=1}^i p_{\pi(t)}$ and $d(\pi(i+1), O_i) = \sum_{t=1}^i d(\pi(i+1), \pi(t))$.

Definition 2 (Sequential coverage diversity (\mathcal{S}_c)). *The sequential coverage diversity of a sequence O , denoted by $\mathcal{S}_c(O)$, is the expected number of attributes covered by a user before quitting the system, where the expectation is taken over the probability distribution of all possible prefixes of O , i.e.,*

$$\mathcal{S}_c(O) = \sum_{t=0}^n \mathbb{P}[\text{accept } O_t \text{ and quit}] \bigcup_{i \in O_t} X(i). \quad (3.3)$$

3.1.2 Problem Definition

Given the definition of \mathcal{S}_+ and \mathcal{S}_c , we define the Maximizing Sequential Sum Diversity problem (MAXSSD) and Maximizing Sequential Coverage Diversity problem (MAXSCD) as finding the best item ordering that maximizes \mathcal{S}_+ and \mathcal{S}_c , respectively. Formally, given a finite set $U = \{1, \dots, n\}$ of n distinct items and associated probabilities p_1, \dots, p_n , associated attributes $X(1), \dots, X(n)$, find an ordering O^* of the items in U that maximizes \mathcal{S}_+ and \mathcal{S}_c , i.e.,

$$\text{MAXSSD} : O^* = \arg \max_{O=\pi(U)} \mathcal{S}_+(O), \quad (3.4)$$

$$\text{MAXSCD} : O^* = \arg \max_{O=\pi(U)} \mathcal{S}_c(O). \quad (3.5)$$

The MAXSCD problem is an instance of the ordered submodular maximization problem as introduced by Kleinberg et al. [91], and thus, a simple greedy algorithm proposed by the authors can be applied to obtain a $\frac{1}{2}$ -approximation. In the rest of this section, we focus on the main theoretical results for the MAXSSD problem.

3.1.3 Analysis and Intermediate Problem

The sequential sum diversity $\mathcal{S}_+(O)$, defined in Equation (3.2), can be expressed as a sum of $n - 1$ marginal contributions. The t -th term is the contribution of placing item $\pi(t + 1)$ after the prefix O_t , and is given by the product of (i) the probability p_{O_t} that the user reaches position $t + 1$, and (ii) the marginal increase in pairwise distance, $d(\pi(t + 1), O_t)$.

Since the reach probability p_{O_t} decreases with t and converges to zero, the value of $\mathcal{S}_+(O)$ is dominated by the early positions in the ordering. This structure motivates a greedy construction: at each step, select an item that simultaneously has a high continuation probability and yields a large marginal distance with respect to the current prefix. The main technical difficulty in analyzing such an approach is that the marginal distance term depends on the entire prefix O_t . Specifically, an item that yields a large marginal distance at the current step may contribute very little to the marginal distances of items added in subsequent steps.

To simplify the analysis, we introduce an intermediate problem in which the marginal distance contribution of an item depends only on the item placed immediately before it, rather than on the entire prefix. We refer to this problem as Maximizing Ordered Hamiltonian Path (MAXOHP). The objective value for a given ordering O is denoted as $\mathcal{H}(O)$, and it is defined as

$$\mathcal{H}(O) = \sum_{i=1}^{n-1} W_{O_i} d(\pi(i), \pi(i + 1)),$$

where $W_{O_i} = \sum_{j=i+1}^n \prod_{t=1}^j p_{\pi(t)}$.

Formally, given a finite set of n distinct items $U = \{1, \dots, n\}$ with associated probabilities p_1, \dots, p_n , the goal of MAXOHP is to compute an ordering O^* of the items in U that maximizes the \mathcal{H} objective, that is,

$$O^* = \arg \max_{O=\pi(U)} \mathcal{H}(O). \quad (3.6)$$

Table 3.1: Approximation ratios achieved by different algorithms for the MAXOHP problem under various continuation probability settings

	<i>p</i> is uniform			<i>p</i> is not uniform
	<i>p</i> is a constant	$p = 1 - \frac{1}{o(n)}$	$p = c - \frac{1}{\Theta(n)}$	$a \leq p_i \leq b$
Approx. ratio	$1 - p^{\tau-1} - p^{n-\tau} + p^n$	$\frac{3(\epsilon-1)}{16\epsilon^2} - \Theta\left(\frac{1}{t_n}\right)$	$c - \frac{1}{\Theta(n)}$	$\frac{a^2(1-b)(1-b^{k-1})}{a^2 + (k-1)b^{k+1}}$
Algorithm	Best- τ items	Greedy matching	Arbitrary ranking	Best τ items

We show that $\mathcal{H}(O)$ can be equivalently formulated as

$$\mathcal{H}(O) = \sum_{i=1}^{n-1} p_{O_{i+1}} d_L(O_{i+1}), \quad (3.7)$$

where $d_L(O_k) := \sum_{t=1}^{k-1} d(\pi(t), \pi(t+1))$.

3.1.4 Main Results

In Theorem 1, we show that any good solution to the MAXOHP problem yields a good solution to the MAXSSD problem. Consequently, it suffices to focus on solving MAXOHP. We begin with the simpler setting of uniform probabilities, where all items have the same continuation probability, and then extend our approach to the more general non-uniform case. In both settings, we design constant-factor approximation algorithms.

Theorem 1. *Let $p_i \in [a, b]$, with $0 < a < b < 1$. An α -approximation solution for MAXOHP is a $\frac{\alpha(1-b)}{2b(1-a)}$ -approximation for MAXSSD.*

The proof of Theorem 1 relies on two key bounds: (1) by the triangle inequality, $2\mathcal{S}_+(O) \geq \mathcal{H}(O)$ for any ordering O , and (2) for an optimal sequence $O^* = \operatorname{argmax}_O \mathcal{S}_+(O)$ to the MAXSSD problem, and assuming that $p_i \in [a, b]$, for all $i \in U$, it holds that $\mathcal{S}_+(O^*) \leq \frac{b(1-a)}{a(1-b)} \mathcal{H}(O^*)$. Combining these bounds yields Theorem 1. When all probabilities are equal, $\frac{\alpha(1-b)}{2b(1-a)} = \frac{1}{2}$, so solving MAXOHP incurs only a factor-2 loss for MAXSSD.

Approximation Guarantees As discussed in Section 3.1.3, the early positions in an ordering dominate the value of the sequential diversity objective. A key question, therefore, is how many items constitute this “first few.” Let this number be denoted by τ ; an appropriate choice of τ depends on the continuation probabilities p_i .

We begin with the uniform case, where $p_i = p$ for all i . In this setting, the coefficient associated with position i is

$$W_{O_i} = p^i + p^{i+1} + \dots + p^n \approx \frac{p^i}{1-p},$$

which decreases geometrically with i . This behavior gives rise to three qualitatively different regimes:

- **Constant Continuation Probability.** If $p < 1$ is a fixed constant, then it follows that $\lim_{i \rightarrow \infty} W_{O_i} = 0$, and only a constant number of early positions contribute significantly to \mathcal{H} . In this case, we use a greedy algorithm to compute a high-quality τ -permutation of the n items and order the remaining items arbitrarily. We refer to this approach as the *best- τ items algorithm*.
- **Near-certain Continuation.** If $p = 1 - \frac{1}{o(n)}$, then $W_{O_i} \approx 1/e$ for $i = o(n)$ and decreases to zero only as i approaches n . Thus, the first $o(n)$ terms all contribute substantially to the objective. Since computing the optimal $o(n)$ -permutation is computationally intractable, we instead design a greedy matching algorithm that ensures the first $o(n)$ consecutive distances are sufficiently large, yielding a constant-factor approximation.
- **Full-ordering Regime.** If $p = c - \frac{1}{\Theta(n)}$ for a constant $c < 1$, then $W_{O_i} \approx c/e$ even for $i = n$, implying that all positions contribute non-negligibly. In this regime, any ordering achieves a constant-factor approximation. When $p = 1$, all orderings are equivalent.

Finally, we consider the non-uniform case in which the continuation probabilities satisfy $a \leq p_i \leq b$ for constants a and b . The best- τ items algorithm and its analysis extend naturally to this setting, preserving a constant-factor approximation guarantee. We summarize the approximation ratios achieved by our proposed algorithms in Table 3.1.

3.2 [Paper B] Streaming Diversity Maximization

In this section, we investigate another user-item interaction system under a streaming and stochastic setting. We introduce a novel diversity maximization problem called Streaming Stochastic Submodular Maximization (*S3MOR*).



Figure 3.1: Illustration of the streaming stochastic submodular maximization problem.

3.2.1 Problem Setting

Assume items arrive in a news recommendation system in a streaming manner. A user can visit the system at any time and as many times as they want. Whenever the user visits, the system must present a set of at most k items to the user. Each item is associated with a click probability, which represents the likelihood of the user clicking on it when presented. Each item is also associated with a set of topics; if a user clicks an item, we say the user covers all topics of that item. The goal is to maximize the expected number of topics the user covers by the end of the stream.

Figure 3.1 provides a toy example of the system, where each news item is represented as a box with colored dots indicating its topics. The user visits the system three times: after the arrival of items 3, 6, and 9, respectively. At each visit, the system outputs a recommendation set of size 2. After the third visit, the user can cover the dark blue topic with probability $1 - (1 - 0.5) \times (1 - 0.7) = 0.85$, the pink topic with probability $1 - (1 - 0.5) \times (1 - 0.5) = 0.75$, the yellow topic with probability 0.4, and the teal topic with probability $1 - (1 - 0.7) \times (1 - 0.6) = 0.88$. The expected number of topics covered by the user after the third visit is thus $0.85 + 0.75 + 0.4 + 0.88 = 2.88$.

3.2.2 Problem Formulation

We start by introducing the notation needed to define our problem. We are given an ordered set of N items $\mathcal{V} = \{V_1, V_2, \dots, V_N\}$, where item V_i arrives at time step i . If the same item appears multiple times, we treat each copy of it as a distinct item. Each news item $V_i \in \mathcal{V}$ covers a subset of topics from a predefined set $\mathcal{C} = \{c_1, c_2, \dots, c_d\}$, where d denotes the total number of topics. Formally, $V_i \subseteq \mathcal{C}$ for each $i \in \{1, \dots, N\}$. Additionally, each news item V_i is associated with a probability $p_i \in [0, 1]$ that represents the likelihood that the user will click on item V_i when it is presented. We also define a binary indicator variable $\tau_i \in \{0, 1\}$ at time step i , with $\tau_i = 1$ denoting an access after the arrival of V_i , and $\tau_i = 0$ otherwise. Whenever $\tau_i = 1$, the system must output a subset of at most k items from the stream $\{V_1, \dots, V_i\}$.

Assume the user visits the system T times at time steps $\{r_1, \dots, r_T\}$. At each time step $j \in [T]$, the system outputs a subset $\mathcal{S}^j \subseteq \{V_1, \dots, V_{r_j}\}$. Let $\mathcal{S} = \{\mathcal{S}^1, \dots, \mathcal{S}^T\}$, we use $f(\mathcal{S})$ to denote the expected number of covered topics, and $f(\mathcal{S})$ is defined as follows:

$$f(\mathcal{S}) = \sum_{j=1}^d \left(1 - \prod_{t=1}^T \prod_{\substack{V_i \in \mathcal{S}^t \\ V_i \ni c_j}} (1 - p_i) \right). \quad (3.8)$$

We note that the exact visit times are not required in practice, and they are included in the objective function purely for notational clarity. In Equation (3.8), the term $1 - \prod_{t=1}^T \prod_{V_i \in \mathcal{S}^t, V_i \ni c_j} (1 - p_i)$ represents the probability that topic c_j is covered, i.e., at least one item containing c_j is clicked across all presented sets. Since each topic's coverage is a Bernoulli random variable with this success probability, the expected number of topics covered equals the sum of individual coverage probabilities across all d topics.

We prove that $f(\mathcal{S})$ is a submodular function. Combined with the stochastic nature of the problem setting, this leads us to name our problem Streaming Stochastic Submodular Maximization (*S3MOR*), which we formally define below.

Problem 1 (*S3MOR*). We define a news item stream \mathbf{S} as a sequence of triples: $\mathbf{S} = ((V_1, p_1, \tau_1), \dots, (V_N, p_N, \tau_N))$ where N is the (unknown) length of the stream. At each time step i , the system receives a triple (V_i, p_i, τ_i) , where $V_i \in \mathcal{V}$, $p_i \in [0, 1]$, and $\tau_i \in \{0, 1\}$. Whenever $\tau_i = 1$, the system selects up to k news items from the set of items $\{V_1, \dots, V_i\}$ received so far to present to the user. Let $\mathcal{S} = \{\mathcal{S}^1, \dots, \mathcal{S}^T\}$ denote the sets of news items presented to the user over T accesses. The objective is to maximize the expected number of distinct topics the user is exposed to by the end of the stream, measured by the function $f(\mathcal{S})$, that is,

$$\begin{aligned} & \max_{\mathcal{S} = \mathcal{S}^1 \cup \dots \cup \mathcal{S}^T} f(\mathcal{S}), \\ & \text{subject to } \mathcal{S}^t \subseteq \{V_1, \dots, V_{\tau_t}\}, \quad \forall t \in [T], \\ & |\mathcal{S}^t| \leq k, \quad \forall t \in [T]. \end{aligned} \quad (3.9)$$

3.2.3 Challenges

The main challenge in solving the *S3MOR* problem stems from three key characteristics. First, user access is *on-demand*: we do not know in advance when or how many times users will access the system. The system must therefore always be prepared to provide fast responses for each user access. Second, recommendation results are *irrevocable*: once items are shown to a user, they cannot be modified. Third, operating in a streaming setting, the system has only sublinear memory with respect to the stream length.

3.2.4 Reduction to Partition Matroid Constraints

Returning to the toy example in Figure 3.1, recall that at the first user visit, items $\{V_1, V_2, V_3\}$ have arrived. At the second visit, items $\{V_1, \dots, V_6\}$ have arrived, and at the third visit, items $\{V_1, \dots, V_9\}$ have arrived. We must select \mathcal{S}^1 from the first set, \mathcal{S}^2 from the second, and \mathcal{S}^3 from the third. To reformulate this problem, we construct an expanded stream by concatenating the three arrival sets and treating each occurrence of an item as distinct. Specifically, we mark each occurrence with a superscript i to indicate it belongs to the i -th arrival set, yielding:

$$\{V_1^1, V_1^2, V_1^3, V_2^1, V_2^2, V_2^3, V_3^1, V_3^2, V_3^3, V_4^2, V_4^3, V_5^2, V_5^3, V_6^2, V_6^3, V_7^3, V_8^3, V_9^3\}.$$

This expanded stream naturally partitions into three disjoint candidate sets: $\mathcal{C}_1 = \{V_1^1, V_2^1, V_3^1\}$, $\mathcal{C}_2 = \{V_1^2, \dots, V_6^2\}$, and $\mathcal{C}_3 = \{V_1^3, \dots, V_9^3\}$. The feasibility constraints of the *S3MOR* problem, namely $\mathcal{S}^t \subseteq \mathcal{C}_t$ and $|\mathcal{S}^t| \leq k$ for $t \in \{1, 2, 3\}$, then define a partition matroid over the expanded stream.

We can generalize the toy example to arbitrary inputs. Assume we know the exact number of visits T and that the user visits the system at (unknown) time steps

$\{r_1, \dots, r_T\}$. For each item arriving after the $(t - 1)$ -th visit and before the t -th visit, we create $T - t + 1$ copies of it, marking each copy with superscripts $t, t + 1, \dots, T$ respectively. By concatenating all copies of all items, we obtain an expanded stream where all items with superscript t belong to the t -th partition \mathcal{C}_t for each $t \in \{1, \dots, T\}$. This generalizes the partition matroid structure from the toy example to arbitrary T .

When T is known, our expanded-stream construction and partitioning reduce the *S3MOR* problem to the streaming submodular maximization problem under a partition matroid constraint. Thus, we can use existing algorithms to solve the *S3MOR* problem. However, not all existing algorithms are suitable. Since we present $\mathcal{S}^t = \mathcal{S} \cap \mathcal{C}_t$ upon the t -th visit, we require that \mathcal{S}^t remains fixed once visit t occurs. Algorithms satisfying this property include those by Chakrabarti and Kale [38], Chekuri et al. [39], and Feldman et al. [40], all achieving $\frac{1}{4}$ - approximation. In this thesis, we use the Streaming-Greedy algorithm from Chekuri et al. [39].

3.2.5 The STORM Algorithm

In the previous section, we assumed that the exact number of user visits T is known. This assumption enables a reduction from the *S3MOR* problem to a streaming submodular maximization problem subject to a partition matroid constraint. However, assuming T is known a priori is unrealistic in practice.

We therefore consider a more practical setting where T is unknown, but an upper bound $T' \geq T$ is provided as input. In practice, T' can be estimated by eliciting from users the maximum number of visits they anticipate, or by analyzing historical user behavior patterns to determine a reasonable upper bound.

A Naive Approach and Its Failure. Given T' , a natural extension of the approach from the previous section would proceed as follows: (1) create $T' - t + 1$ copies of each item arriving before the t -th user access, marking each copy with superscript $t, t + 1, \dots, T'$; (2) define a partition matroid where items with superscript j belong to the j -th partition; (3) run the Streaming-Greedy algorithm on the expanded stream and output results from the t -th partition upon the t -th user visit.

Unfortunately, this straightforward adaptation fails to provide meaningful guarantees. We demonstrate this via the following adversarial example. Let $T = 1$, with $V_1 = \{c_1\}$ and $p_1 = 1 - \epsilon$, followed by $V_2 = \{c_2\}$ with $p_2 = \epsilon$, where $\epsilon < 1$ satisfies $1 - (1 - \epsilon)^{T'} < 1 - (1 - \epsilon)^{T'-1} + \epsilon$. Suppose a user accesses the system after V_2 arrives, and $k = 1$. Initially, the Streaming-Greedy algorithm maintains empty results in each partition. After V_1 arrives, all partition results become $\{V_1\}$. However, after V_2 arrives, the expected coverage of $\{V_2^1, V_1^2, V_1^3, \dots, V_1^{T'}\}$ exceeds that of $\{V_1^1, V_1^2, \dots, V_1^{T'}\}$. Consequently, the first partition's result is replaced by $\{V_2\}$. Upon the user's visit, the system outputs the first partition, achieving a competitive ratio of $\frac{\epsilon}{1 - \epsilon}$, which can be arbitrarily poor.

The Active Partition Strategy. To address the shortcoming of the previous naive approach, we propose a simple yet effective modification: construct an extended

stream and partition it into T' partitions as described in the naive approach, upon each user visit, greedily select and output results from the best *active* partition. We call this algorithm STORM, and it yields a competitive ratio of $\frac{1}{4(T'-T+1)}$.

The key insight is as follows. Initially, we mark all partitions as active and update each partition with every arriving item. Upon a user visit, the results from all active partitions are valid, as each represents a subset of size k selected from the items that have arrived up to that point. We select the partition with the best objective value, output its solution to the user, mark that partition as inactive, and continue updating all remaining active partitions. When $T' = T$, the competitive ratio is equal to $1/4$, which matches the approximation ratio of the underlying Streaming-Greedy algorithm.

3.2.6 The STORM++ Algorithm

When T' is significantly larger than T , the competitive ratio of $\frac{1}{4(T'-T+1)}$ degrades substantially. To improve the competitive ratio, the key insight is to reduce the gap between T' and T .

Discretization and Parallel Execution. We can reduce $T' - T$ by discretizing the range $[1, T']$ into a geometric sequence of candidate values $\{\delta, 2\delta, \dots, \lceil \frac{T'}{\delta} \rceil \delta\}$. By construction, at least one of these candidates differs from the true value T by at most δ . If we could execute STORM with this closest guess, we would obtain a competitive ratio of at least $\frac{1}{4\delta}$.

However, since we do not know a priori which guess is closest to T , we run multiple parallel instances of STORM, each configured with a different candidate value from our discretized set. This parallel execution introduces a new challenge: upon each user visit, each STORM instance produces its own output, which one should be presented to the user?

Greedy Selection Across Instances. We answer the above question by applying a greedy selection strategy: upon the t -th user visit, we collect the t -th output from each STORM instance and present to the user the solution with the highest expected coverage. This greedy selection guarantees an expected coverage at least half that of the solution produced by the instance running with the best guess. The best guess produces a solution with a competitive ratio of at least $\frac{1}{4\delta}$, consequently, STORM++ achieves a competitive ratio of $\frac{1}{8\delta}$, where the parameter $\delta \in \mathbb{N}$ controls the trade-off between solution quality and computational efficiency. Smaller values of δ yield better competitive ratios but require running more parallel instances.

3.2.7 Complexity

To evaluate our proposed algorithms, we report three complexity metrics: *space complexity*, *time complexity*, and *response time complexity*. Assuming a common space stores all original items, *space complexity* measures the additional memory each user requires for personalized recommendations, including document identifiers and user-item association probabilities. We assume an *oracle* can evaluate $f(S)$ for any

feasible set S . *Response time complexity* measures the worst-case oracle calls needed to produce a size- k recommendation after a user request, capturing user-experienced latency, while *time complexity* counts oracle calls over the entire stream.

3.2.8 Main Results

Previously in the challenges section, we mentioned the goal of using sublinear memory. However, if we relax this constraint and allow $\mathcal{O}(N)$ memory to store item probabilities for all items in the stream, the problem degrades to an offline setting and reduces to submodular maximization subject to a partition matroid constraint, which is studied by Fisher et al. [30]. In this case, we store all incoming items' probabilities and, upon each user request, greedily select the best k items from all available items. This greedy algorithm achieves a $\frac{1}{2}$ -approximation ratio [30]. We name this algorithm LMGREEDY, and report its performance below.

Theorem 2. *The LMGREEDY algorithm for Problem 1 has a competitive ratio of $\frac{1}{2}$, space complexity $\Theta(N + kT)$, time complexity $\mathcal{O}(NTk)$, and response time complexity $\mathcal{O}(Nk)$. Moreover, the competitive ratio of $\frac{1}{2}$ is tight, i.e., for Problem 1, no streaming algorithm can achieve a competitive ratio better than $\frac{1}{2}$ without violating the irrevocability constraint.*

We also report the performance of the STORM and STORM++ algorithm below. The proofs of Theorem 3 and Theorem 4 are built on the submodularity of the objective function and the greedy selection criterion.

Theorem 3. *Let T be the number of user accesses and T' a given upper bound. STORM has competitive ratio $\frac{1}{4(T'-T+1)}$, space complexity $\mathcal{O}(T'k)$, time complexity $\mathcal{O}(NkT')$, and response time complexity $\mathcal{O}(T')$.*

Theorem 4. *STORM++ has a competitive ratio of $\frac{1}{8\delta}$, space complexity $\mathcal{O}(T'^2k/\delta)$, time complexity $\mathcal{O}(NkT'^2/\delta)$, and response time complexity $\mathcal{O}(T')$.*

Chapter 4

Diversity-Constrained Densest Subgraph Discovery

From this chapter onwards, we shift our attention from interactive recommender systems to problems formulated in graph settings. In this chapter, we study the densest subgraph discovery problem (DSP) with edge-color constraints. In this problem, diversity is not directly maximized but implicitly enforced as an optimization constraint.

4.1 [Paper C] Finding Densest Subgraphs with Edge-Color Constraints

In large social networks, users are connected by many kinds of relationships, such as friendship, family, acquaintance, and work ties, which also vary in strength. Simply finding dense subgraphs ignores this heterogeneity and may miss communities that matter for downstream tasks. For instance, in fraud detection on financial transaction graphs where nodes represent accounts and edges represent transactions with attributes such as amount, merchant category, and geographic location, fraudulent account clusters may be characterized by specific edge-type combinations such as transactions in unusual merchant categories for those accounts, or activity from locations that mismatch the accounts' typical patterns. Such patterns would be missed if all edge types were treated equally. Densest subgraph discovery with edge type constraints allows us to focus on groups that are not only well connected, but also have specific mixes of relationship types and tie strengths. Different applications require different edge-type combinations. For example, contact recommendations seek communities rich in weak professional ties, while identifying functional protein complexes requires groups with strong interaction weights across multiple binding types.

To formalize these diverse relationship types, we adopt the classical graph-theoretic convention of representing edge types as colors. Formally, let $G = (V, E)$ denote an undirected, simple graph where V is a finite vertex set and E is a finite edge set satisfying $E \subseteq \{\{u, v\} \subseteq V \mid u \neq v\}$. The cardinalities of these sets are denoted by $n = |V|$ and $m = |E|$.

An edge-colored graph $G = (V, E, c)$ extends this basic structure by assigning colors to edges through a coloring function $c : E \rightarrow 2^{\mathbb{N}}$ that maps each edge to a set of natural numbers representing colors. While this definition permits edges to have multiple colors, we focus on the single-color case throughout this chapter, where each edge has exactly one color. For convenience, if edge $e \in E$ has a single color i , we abbreviate $c(e) = \{i\}$ by writing $c(e) = i$. The extension to multi-colored edges is discussed in our paper C.

For any subset $S \subseteq V$, we use $E[S]$ to denote the edge set $\{\{u, v\} \in E \mid u, v \in S\}$ containing all edges with both endpoints in S . The induced subgraph on S is then written as $G[S] = (S, E[S])$.

We call a graph $G = (V, E)$ sparse if $|E| = O(|V|)$. A graph $G = (V, E)$ is *everywhere sparse* if for any subset $V' \subseteq V$, the induced subgraph $G(V')$ is sparse. Throughout this chapter, we restrict attention to everywhere sparse graphs.

With this notation in place, we can now formalize the edge color constrained densest subgraph problem. Let K denote the maximum number of colors present in the graph. We represent edge-color constraints using a vector $\mathbf{h} \in \mathbb{N}^K$, leading to the following definition:

Definition 3 (ALHCEGESDSP). *Given an edge-colored graph $G = (V, E, c)$, a number of colors $K \in \mathbb{N}$, and a vector $\mathbf{h} \in \mathbb{N}^K$, find $S \subseteq V$ such that*

- $E[S]$ contains at least h_i edges e with $i \in c(e)$ for all $i \in [K]$, and
- the density $\rho(S) = \frac{|E[S]|}{|S|}$ is maximized.

We name the problem the **At Least \mathbf{h} Colored Edges Densest Subgraph Problem**, and denote it as ALHCEGESDSP.

The simplest case for this problem is when all edges have the same color. Assuming the vector \mathbf{h} sums to an integer h , the ALHCEGESDSP problem reduces to the **At Least h Edges Problem**, denoted as ATLEASTHEGESDSP, which we formally define below.

Definition 4 (ATLEASTHEGESDSP). *Given a graph $G = (V, E)$ and $h \in \mathbb{N}$, find a subset $S \subseteq V$ such that $|E[S]| \geq h$, and the density $\rho(S) = \frac{|E[S]|}{|S|}$ is maximized.*

We prove that the decision version of the ATLEASTHEGESDSP problem is NP-hard. Thus, the decision version of the ALHCEGESDSP problem is also NP-hard. To solve these problems, we utilize existing algorithms for the **Densest At Least k Nodes Subgraph Problem**, denoted as DalkS, which we define below.

Definition 5 (DalkS). *Given a graph $G = (V, E)$ and $k \in \mathbb{N}$, find a subset $S \subseteq V$ such that $|S| \geq k$, and the density $\rho(S) = \frac{|E[S]|}{|S|}$ is maximized.*

The main technical approach of our work is as follows: we first prove that we can approximate the ATLEASTHEGESDSP problem by solving the DalkS problem, then we approximate the ALHCEGESDSP problem by approximating the ATLEASTHEGESDSP problem. Both approximation steps achieve constant approximation ratios.

4.2 Reducing ATLEASTHEDGESDSP to Dal k S

Below, we state the key lemma for our approach. The aim is to approximate the densest at least h edges DSP via the densest at least k node DSP.

Lemma 1. *Given a graph G and $h \in \mathbb{N}$. Let k be the minimum number of nodes over all graphs that are densest subgraphs of G with at least h edges. Furthermore, let S be an optimal solution for the at least k -nodes DSP problem in G . Then S is also an optimal solution for the densest subgraph with at least h edges.*

Lemma 1 suggests an approach to solve the at least h edge DSP. Indeed, it states that there exists an integer $k < N$ such that the optimal density for the at least h edge DSP is equal to that of the at least k nodes DSP. The proof follows by contradiction and can be found in our paper C.

Consequently, provided an α -approximation solution S of the at least k nodes DSP, $\rho(S)$ is also an α approximation of at least h edge DSP. If S is also feasible, S is the desired solution. Otherwise, if $G[S]$ contains fewer than h edges, we can obtain a feasible solution S' by adding at most h edges to S . It remains to bound the density of S' , since adding edges may decrease $\rho(S)$. To facilitate the analysis, we introduce the following definition.

Definition 6 (Constant-Yield Approximation). *Let G be a graph, and let S^* denote an optimal solution to the at least k nodes DSP. Let \mathcal{A} be an α -approximation algorithm for this problem, and let S be the subgraph returned by \mathcal{A} on input G . We say that \mathcal{A} is a **Constant-Yield** algorithm if for any input graph, there exists a constant $c_1 > 0$ such that*

$$c_1 \cdot |E(S)| \geq |E(S^*)|. \quad (4.1)$$

The $\frac{1}{2}$ -approximation algorithm of Khuller and Saha [43] and the $\frac{1}{3}$ -approximation algorithm of Andersen and Chellapilla [44] are both Constant-Yield approximation algorithms.

Analysis. Let ρ^* denote the optimal density for the at least h edges DSP. Let S be the output of a Constant-Yield approximation algorithm such that $\rho(S) \geq \frac{1}{\alpha}\rho^*$. To lower bound the density $\rho(S') = \frac{|E[S']|}{|S'|}$, we upper bound $|S'|$ and lower bound $|E[S']|$ as follows:

- By the definition of Constant-Yield approximation algorithms, there exists a constant c_1 such that $c_1|E[S]| \geq |E[S^*]| \geq h$.
- Since G is everywhere sparse, we have $|E[S]| \leq c_2|S|$ for some constant c_2 . Adding h edges to S introduces at most $2h$ nodes, so

$$|S'| \leq |S| + 2h \leq |S| + 2c_1|E[S]| \leq |S| + 2c_1c_2|S| \leq c|S| \quad (4.2)$$

holds for some sufficiently large constant c .

Combining Equation (4.2) with $|E[S']| \geq |E[S]|$, we obtain

$$d(S') = \frac{|E[S']|}{|S'|} \geq \frac{|E[S]|}{c|S|} = \frac{1}{c}d(S) \geq \frac{1}{c\alpha}\rho^*. \quad (4.3)$$

To summarize, given an everywhere sparse graph G and $h \in \mathbb{N}$, let k denote the minimum number of nodes over all densest subgraphs of G with at least h edges. Let S be the output of a Constant-Yield approximation algorithm [43, 44]. Then S (or any subgraph S' obtained by adding at most h edges to S) is a constant approximation for the at least h edges DSP. For ease of reference, we refer to this procedure as the EDGEAUG algorithm.

To complete the analysis, one essential question remains: how do we determine k ? Determining k exactly is computationally challenging, as it requires either solving the densest subgraph problem with exactly k nodes for all $k \in \{1, \dots, n\}$, which is NP-hard, or enumerating all densest subgraphs, which can be exponential in number.¹

Instead, we compute a lower bound $\ell(h)^2$ depending only on h , and run EDGEAUG for all $k \in \{\ell(h), \dots, n\}$, returning the solution with the highest density. In particular, the peeling algorithm of Andersen and Chellapilla [44] covers all such values of k in a single pass in $O(m + n)$ time, making this approach practical.

The above analysis yields the following theorem.

Theorem 5. *For everywhere sparse graphs, there exists a constant-factor approximation for the at least h edges DSP.*

Proof. To illustrate this point, let $\{G_{\ell(h)}, \dots, G_n\}$ denote the subgraphs produced by a Constant-Yield approximation algorithm for the at least k nodes DSP for $k \in \{\ell(h), \dots, n\}$, respectively. Let \tilde{G}_i denote the result of augmenting G_i with additional edges from the input graph G to satisfy the edge constraint h . Then the subgraph

$$G^* = \arg \max_{i \in \{\ell(h), \dots, n\}} \rho(\tilde{G}_i)$$

provides a constant-factor approximation for the at least h edges DSP. \square

4.3 Reducing ALHCEDGESDSP to ATLEASTHEDGESDSP

In this section, we show that given a constant approximation for the at least h edges DSP, we can obtain a constant approximation for the at least \mathbf{h} colored edges DSP.

Let $l = \sum_{i=1}^K \mathbf{h}_i$. The previous section provides a solution S' that is a $\frac{1}{\alpha c}$ -approximation for the at least l edges DSP. If S' is also valid for the at least \mathbf{h} colored edges DSP, S' is the desired solution. Otherwise, we construct a feasible solution $S'' = (V'', E'')$ by adding edges to S' . Specifically, let f_i be the number of edges

¹Consider a graph consisting of r disjoint densest subgraphs. The union of any non-empty subset of these components is also a densest subgraph, yielding $2^r - 1$ distinct solutions. In the worst case, $r = \Theta(n)$, resulting in $2^{\Theta(n)}$ densest subgraphs in total.

²The details on how to calculate ℓ can be found in our paper C.

of color $i \in [K]$ in S' , we need to add $\max\{0, h_i - f_i\}$ edges of color i to make S' feasible.

Let S^* be the optimal solution for the at least l edges DSP, and S_c^* be the optimal solution for the at least \mathbf{h} colored edges DSP. Since the at least l edges DSP relaxes the at least \mathbf{h} colored edges DSP, we have $\rho(S^*) \geq \rho(S_c^*)$. This inequality enables us to lower bound $\rho(S'')$. Since adding one edge introduces at most two nodes, we have $|E''| \leq |E'| + l$ and $|V''| \leq |V'| + 2l \leq |V'| + 2|E'| \leq c|V'|$ for some constant $c \geq 3$. Thus,

$$d(V'') = \frac{|E''|}{|V''|} \geq \frac{|E'|}{c|V'|} = \frac{1}{c}d(V') \geq \frac{1}{c^2\alpha}d^*.$$

To summarize, we get the following theorem.

Theorem 6. *For everywhere sparse graphs, there exists a constant approximation algorithm for the at least \mathbf{h} colored edges DSP. Specifically, let S' be a $\frac{1}{\alpha}$ -approximation for the at least $\sum_{i=1}^K \mathbf{h}_i$ edges DSP. Then S' (or any subgraph S'' obtained by adding at most h_i edges of color i for each $i \in [K]$ to S') is a $\frac{1}{\alpha c}$ -approximation for the at least \mathbf{h} colored edges DSP for some sufficiently large constant c .*

4.4 Conclusion

In paper C, we introduce new variants of the densest subgraph problem in networks with single or multiple edge attributes. To the best of our knowledge, this is the first work that studies the densest subgraph problem subject to edge color constraints. We define the at least \mathbf{h} colored edges DSP and solve it on everywhere sparse graphs. Our approach first reduces the at least h edges DSP to the at least k nodes DSP, then shows that augmenting the solution with colored edges yields a feasible solution to the at least \mathbf{h} colored edges DSP with only a constant factor loss in approximation. We provide two $\mathcal{O}(1)$ -approximation algorithms for the at least h edges DSP and extend them to obtain constant-approximation algorithms for the at least \mathbf{h} colored edges DSP. While our algorithms are designed for graphs where each edge has a single color, we show in the paper C that they can be easily extended to the general case where edges may have multiple colors, maintaining $\mathcal{O}(1)$ -approximation guarantees.

Our experimental results provided in the paper C validate the practical efficacy and efficiency of the proposed algorithms on a wide range of real-world graphs. As a case study, we evaluate our methods on the DBLP co-authorship network to identify dense research communities with diverse conference representation. Our approach outperforms existing multilayer-based densest subgraph methods by achieving higher density while ensuring balanced participation across all ten conferences.

Chapter 5

Graph Interventions for Fairness in Networks

In this chapter, we continue our exploration of graphs with a focus on fairness, studying two complementary fairness measures: PageRank fairness (Section 5.1) and hitting-time fairness (Section 5.2). PageRank captures the steady-state importance or centrality of nodes as determined by the graph’s link structure, while hitting time measures the expected number of steps a random walk takes to reach a target node, reflecting notions of accessibility and proximity.

For both problem settings, we assume that vertices are partitioned into disjoint groups based on a sensitive attribute. In social or human connection networks, this attribute may represent demographic characteristics such as gender or race, while in video recommendation networks, it may encode content properties such as neutral versus harmful video classifications.

We enhance fairness in both settings through targeted graph interventions. For PageRank fairness, we rebalance influence between vertex groups through edge reweighting, whereas for hitting-time fairness, we introduce a small number of strategic shortcuts to improve accessibility for selected groups. These intervention strategies are motivated by real-world applications where structural modifications can mitigate bias and promote equitable outcomes. In content recommendation systems, adjusting edge weights can rebalance visibility across different types of content, while in polarized social networks, strategic edge additions can reduce segregation and improve connectivity between disparate groups. Together, these two problem settings demonstrate that structural interventions in graphs can serve as powerful tools to address different dimensions of fairness.

5.1 [Paper D] Fairness-Aware PageRank via Edge-Rewiring

The problem of fairness-aware PageRank is to ensure that a graph’s influence is distributed equitably among groups. To achieve such fairness, various intervention strategies can be employed, including adjusting algorithmic parameters such as mod-

ifying restart probabilities and restart vectors, and adjusting graph structures such as adding edges and reweighting existing edges. In our work, we focus on edge reweighting.

Consider a video recommendation graph as an example. Here, each node represents a video, and an edge (u, v) indicates that while a user is watching video u , video v is recommended as the next video to watch. Typically, there is a list of recommended next-to-watch videos, and the ranking of each video can be influenced by the edge weights. For instance, a recommendation list $[v_1, v_2, \dots, v_k]$ corresponds to edge weights $w(u, v_1) \geq w(u, v_2), \dots, \geq w(u, v_k)$. In this scenario, to enhance or reduce the visibility of videos from a target group, a moderator can adjust the rankings within the recommendation lists, which corresponds to modifying the edge weights in the graph.

5.1.1 The PageRank Fairness Loss

Given a directed graph $G = (V, E)$ with n vertices and m edges, suppose the vertices are partitioned into K disjoint groups $\{V_1, \dots, V_K\}$. For each group V_k , let $\mathbf{1}_k \in \{0, 1\}^n$ denote its indicator vector, where $\mathbf{1}_{kj} = 1$ if vertex $j \in V_k$ and $\mathbf{1}_{kj} = 0$ otherwise.

Let \mathbf{P} be the row-stochastic transition matrix associated with G , and let \mathbf{p} denote the PageRank vector with restart probability γ and restart distribution \mathbf{v} . The PageRank vector satisfies

$$\mathbf{p}^\top = \gamma \mathbf{v}^\top (\mathbf{I} - (1 - \gamma)\mathbf{P})^{-1}. \quad (5.1)$$

Given a target distribution of group-wise PageRank scores $\phi = (\phi_1, \dots, \phi_K)$, we define two fairness loss functions.

(1) L_2 PageRank Fairness Loss.

$$L(\mathbf{P}, \gamma, \mathbf{v}, \phi) = \frac{1}{K} \sum_{k=1}^K (\mathbf{1}_k^\top \mathbf{p} - \phi_k)^2, \quad (5.2)$$

where \mathbf{p} is the PageRank vector with respect to $(\mathbf{P}, \gamma, \mathbf{v})$ as defined in Equation (5.1).

(2) Group-Adapted L_2 PageRank Fairness Loss.

$$L_g(\mathbf{P}, \gamma, \phi) = \frac{1}{K^2} \sum_{\ell=1}^K \sum_{k=1}^K (\mathbf{1}_k^\top \mathbf{p}_\ell - \phi_k)^2, \quad (5.3)$$

where the restart vector $\mathbf{v}_\ell = \frac{1}{|V_\ell|} \mathbf{1}_\ell$ corresponds to a random walk that restarts uniformly within group ℓ , and

$$\mathbf{p}_\ell^\top = \gamma \mathbf{v}_\ell^\top (\mathbf{I} - (1 - \gamma)\mathbf{P})^{-1}.$$

Both loss functions measure the deviation between the target and the actual group-wise PageRank scores. The first allows an arbitrary restart vector \mathbf{v} , whereas the second restricts restarts to occur uniformly within individual groups.

5.1.2 Problem Formulation

Our objective is to find a new transition matrix $\hat{\mathbf{P}}$ that minimizes one of the above loss functions subject to structural constraints. To this end, we define two feasible sets:

$$\mathcal{C}(\mathbf{P}) = \left\{ \hat{\mathbf{P}} \in \mathbb{R}_{\geq 0}^{n \times n} \mid \hat{\mathbf{P}}\mathbf{1} = \mathbf{1}, \hat{\mathbf{P}}_{ij} = 0 \text{ if } \mathbf{P}_{ij} = 0 \right\}, \quad (5.4)$$

$$\mathcal{C}_R(\mathbf{P}) = \left\{ \hat{\mathbf{P}} \in \mathbb{R}_{\geq 0}^{n \times n} \mid (1 - \delta)\mathbf{P}_{ij} - \epsilon \leq \hat{\mathbf{P}}_{ij} \leq (1 + \delta)\mathbf{P}_{ij} + \epsilon \right\}. \quad (5.5)$$

The constraint set $\mathcal{C}(\mathbf{P})$ preserves the original graph structure by forbidding the introduction of new edges, while $\mathcal{C}_R(\mathbf{P})$ further restricts the magnitude of edge reweighting by imposing both relative (δ) and absolute (ϵ) bounds.

Given a transition matrix \mathbf{P} , restart probability $0 < \gamma < 1$, groups V_1, \dots, V_K , and a target group-wise PageRank score distribution $\phi = (\phi_1, \dots, \phi_K)$ satisfying $\|\phi\|_1 = 1$, we consider the following general optimization problem:

$$\mathbf{P}^* = \arg \min_{\hat{\mathbf{P}} \in \mathcal{C}} \mathcal{L}(\hat{\mathbf{P}}, \phi), \quad (5.6)$$

where $\mathcal{L} \in \{L(\cdot, \gamma, \mathbf{v}, \phi), L_g(\cdot, \gamma, \cdot, \phi)\}$ and $\mathcal{C} \in \{\mathcal{C}(\mathbf{P}), \mathcal{C}(\mathbf{P}) \cap \mathcal{C}_R(\mathbf{P})\}$. Combining the two objectives with the two types of feasibility constraints yields four concrete optimization problems as follows

- **ϕ -PageRank:** Minimize $L(\hat{\mathbf{P}}, \gamma, \mathbf{v}, \phi)$ subject to $\mathcal{C}(\mathbf{P})$.
- **Restricted ϕ -PageRank:** Minimize $L(\hat{\mathbf{P}}, \gamma, \mathbf{v}, \phi)$ subject to $\mathcal{C}(\mathbf{P}) \cap \mathcal{C}_R(\mathbf{P})$.
- **Group-adapted ϕ -PageRank:** Minimize $L_g(\hat{\mathbf{P}}, \gamma, \phi)$ subject to $\mathcal{C}(\mathbf{P})$.
- **Restricted group-adapted ϕ -PageRank:** Minimize $L_g(\hat{\mathbf{P}}, \gamma, \phi)$ subject to $\mathcal{C}(\mathbf{P}) \cap \mathcal{C}_R(\mathbf{P})$.

The motivation for studying ϕ -PageRank and Restricted ϕ -PageRank is straightforward: our goal is to redistribute group-wise PageRank scores by reweighting existing edges. In contrast, the motivation for Group-adapted ϕ -PageRank and Restricted group-adapted ϕ -PageRank is more subtle. Real-world networks often exhibit strong homophily, whereby vertices within the same group are more densely connected than vertices across groups. As a result, the total PageRank score of a group V_k is largely determined by random walks that restart within that group [92]. The Group-adapted ϕ -PageRank and Restricted group-adapted ϕ -PageRank problems consider this homophilic structure in its extreme form, asking whether desired group-wise PageRank distributions can still be achieved solely through edge reweighting when random walks are restricted to restart within individual groups.

5.1.3 Proposed Solution

We prove that the loss functions defined in Equation (5.2) and Equation (5.3) are non-convex, but the feasibility constraints $\mathcal{C}(\mathbf{P})$ and $\mathcal{C}(\mathbf{P}) \cap \mathcal{C}_R(\mathbf{P})$ are convex, so we

propose efficient projected gradient-descent methods for computing locally-optimal edge weights. We prove that the proposed projected gradient update method with constant step size converges to a stationary point, which achieves a local minimum for all four problems we study.

The gradient of the fairness-loss function $L(\mathbf{P}, \gamma, \mathbf{v})$ with respect to the transition matrix \mathbf{P} is given by

$$\frac{\partial}{\partial \mathbf{P}} L(\mathbf{P}, \gamma, \mathbf{v}) = \frac{2(1-\gamma)}{K} \sum_{k=1}^K (\mathbf{1}_k^\top \mathbf{p} - \phi_k) \mathbf{p} \mathbf{y}_k^\top, \quad (5.7)$$

where \mathbf{p} is the PageRank vector with respect to $(\mathbf{P}, \gamma, \mathbf{v})$, and $\mathbf{y}_k = \mathbf{U}^{-T} \mathbf{1}_k = (\mathbf{I} - (1-\gamma)\mathbf{P})^{-1} \mathbf{1}_k$, with $\mathbf{U} = \mathbf{I} - (1-\gamma)\mathbf{P}^\top$. The gradient of the fairness-loss function $L_g(\mathbf{P}, \gamma)$ with respect to the transition matrix \mathbf{P} is given by

$$\frac{\partial}{\partial \mathbf{P}} L_g(\mathbf{P}, \gamma) = \frac{2(1-\gamma)}{K^2} \sum_{k=1}^K \sum_{\ell=1}^K (\mathbf{1}_k^\top \mathbf{p}_\ell - \phi_k) (\mathbf{p}_\ell \mathbf{1}_k^\top \mathbf{U}^{-1}), \quad (5.8)$$

where \mathbf{p}_ℓ is the PageRank vector with respect to $(\mathbf{P}, \gamma, \mathbf{v}_\ell)$.

After each gradient descent step, $\hat{\mathbf{P}}$ needs to be projected back into the feasible area subject to constraints $\mathcal{C}(\mathbf{P})$ or $\mathcal{C}(\mathbf{P}) \cap \mathcal{C}_R(\mathbf{P})$. Both projections are studied in the literature, with the former constraint type studied by Michelot [93], Pardalos and Kooroor [94], Duchi et al. [95], Condat [96], and the latter by Adam and Mácha [97].

5.2 [Paper D] Fairness-Aware Structural Bias Reduction

As discussed in the previous section, in homophilic networks, a group's PageRank is largely driven by random walks that restart within the group. Beyond this effect on PageRank, homophily also induces structural bias, creating segregation in which disparate groups remain only loosely connected. Such segregation perpetuates inequalities by limiting cross-group information flow and reducing minority group visibility. This phenomenon manifests in polarized social network communities and antagonistic content in video-sharing or news-feed platforms. Increasing connectivity between disparate groups is therefore crucial for minimizing social friction and exposing individuals to diverse viewpoints.

A natural measure for quantifying such structural bias is the hitting time between groups, which captures how quickly a random walk originating in one group is expected to reach another. High inter-group hitting times indicate poor connectivity and limited information flow, both of which perpetuate segregation and deepen inequality. In this paper, we study two optimization problems aimed at reducing structural bias through edge additions, focusing on minimizing the average and the maximum hitting times between disparate groups. The average hitting time captures the overall efficiency of inter-group navigation, while the maximum hitting time addresses fairness by preventing worst-case disparities where certain node pairs remain excessively isolated. Both objectives present significant algorithmic challenges, as average hitting time requires handling supermodular optimization under cardinality constraints

and maximum hitting time lacks supermodularity entirely. Nevertheless, we develop approximation algorithms for both objectives with provable guarantees.

5.2.1 Problem Formulation

Graphs. Let $G = (V, E)$ denote an undirected, connected graph where V is a finite vertex set with $|V| = n$ and E is a finite edge set. The vertex set V is partitioned into two disjoint, non-empty groups R (red nodes) and B (blue nodes). For a subset $X \subseteq V$, let $E[X]$ denote the set of edges with both endpoints in X , and let $G[X] = (X, E[X])$ denote the induced subgraph on X . For a set of non-edges F of G , we write $G + F = (V, E \cup F)$. The degree of a node v in G , denoted $d_G(v)$, is its number of neighbors.

Random walks. We consider uniform simple random walks on G , where at each step a neighbor of the current node is selected uniformly at random. For a subset $A \subseteq V$, the random variable $\tau_u(A)$ denotes the first time a walk starting from node u visits a node in A . If $u \in A$, then $\tau_u(A) = 0$. The expectation $H_G(u, A) = \mathbb{E}_G[\tau_u(A)]$ is the *hitting time* from u to A . We note the following properties: $H_G(u, A) = \min_{a \in A} H_G(u, a)$, $H_G(u, u) = 0$, $H_G(u, A) = 0$ if and only if $u \in A$, and $H_G(u, A) \leq H_G(u, B)$ whenever $B \subseteq A$.

Definition 7 (Maximum and Average Hitting Time). *For inter-group non-edges $F \subseteq (R \times B) \setminus E$, the maximum hitting time from R to B on $G + F$ is*

$$f(F) \triangleq \max_{r \in R} H_{G+F}(r, B),$$

and the average hitting time from R to B on $G + F$ is

$$g(F) \triangleq \frac{1}{|R|} \sum_{r \in R} H_{G+F}(r, B).$$

Problem Definitions. Given an undirected connected graph $G = (V, E)$ with $|V| = n$ nodes, valid bipartition $V = \{R, B\}$ and budget $k \in \mathbb{N}$, we seek to find inter-group non-edges $F \subseteq (R \times B) \setminus E$ with $|F| \leq k$ that minimize hitting time from R to B . We consider two variants:

- **Budgeted Minimum Maximum Hitting Time (BMMH):** Minimize the maximum hitting time $f(F)$ from Definition 7.
- **Budgeted Minimum Average Hitting Time (BMAH):** Minimize the average hitting time $g(F)$ from Definition 7.

5.2.2 Key Properties and Observations

We establish several key properties for solving BMMH and BMAH.

1. **Reduction to Selecting Red Endpoints.** Changing the blue endpoints of edges in a feasible solution F does not affect $f(F)$ or $g(F)$, since random walks

halt upon reaching any blue node. Consequently, both problems reduce to selecting a multiset of red endpoints, where multiple edges may share the same red endpoint.

2. **Monotonicity.** Both f and g are monotonically decreasing: adding inter-group edges can only reduce hitting times from red nodes to blue nodes.
3. **Supermodularity of Individual and Average Hitting Times.** The individual hitting-time function $H_{G+e}(r, B)$ is supermodular for all $r \in R$, which implies that the average hitting-time function g is also supermodular. This supermodularity captures a diminishing returns property for hitting-time reductions when adding shortcut edges. Intuitively, the marginal benefit of adding edge e decomposes into two factors: the expected time saved conditional on e being traversed (which is equal to $H(u, B) - 1$, where u is the red endpoint of edge e), and the probability that the walk traverses e before reaching B . Adding another edge e' before e decreases both factors: e' provides an alternative path that reduces the probability of traversing e , and it also shortens the remaining expected hitting time to B , thereby reducing the potential savings from using edge e . The joint decrease of these two factors ensures diminishing marginal returns from adding shortcut edges.
4. **Non-supermodularity of Maximum Hitting Time.** The maximum hitting-time function f is not supermodular due to the max operator, and it is not weakly- α -supermodular [65] for any $\alpha \geq 1$. A counterexample on a simple path graph establishing this claim is provided in paper E.

5.2.3 Algorithm for BMAH

A Classic Greedy Algorithm

Liberty and Sviridenko [65] show that for a non-increasing weakly- α -supermodular function f , if there exists an η -approximation algorithm for minimizing f , then the greedy algorithm that runs for $\lceil \alpha k \ln(\eta/\epsilon) \rceil$ iterations generates a $(1+\epsilon)$ -approximation solution. We can verify that BMAH meets the requirements of this result: (1) As stated in Section 2.2.2, since BMAH is supermodular, it is weakly- α -supermodular with $\alpha = 1$. (2) The empty solution $g(\emptyset)$ provides an n^3 -approximation of the BMAH problem, because hitting times in connected graphs are bounded by n^3 [71], and the optimal hitting time is at least one. Thus $\eta = n^3$. Taking $\alpha = 1$ and $\eta = n^3$ yields the following lemma:

Lemma 2. *The Greedy strategy that iteratively selects the edge yielding the largest marginal reduction in g returns a $(1 + \epsilon)$ -approximation to an optimal solution of BMAH (that adds at most k edges) while adding at most $\lceil k \ln(n^3/\epsilon) \rceil$ edges.*

The proof leverages a classical result on minimizing supermodular functions under cardinality constraints. Exploiting the supermodularity of g , the greedy algorithm reduces the gap to the optimal solution by a multiplicative factor of approximately

$(1 - 1/k)$ at each iteration. After $\tau = \lceil k \ln(n^3/\epsilon) \rceil$ iterations, this yields a $(1 + \epsilon)$ -approximation, where the n^3 term accounts for the initial approximation ratio between the empty solution and any optimal solution ($g(F^*) \geq 1$).

Accelerating Hitting Time's Computation

The main challenge in applying the greedy algorithm is the computational cost of evaluating g , which requires either solving a linear system with $O(n)$ variables or computing the fundamental matrix $(\mathbf{I} - \mathbf{Q})^{-1}$ of an absorbing Markov chain [98], where \mathbf{Q} is the transition matrix between red nodes [98]. Since evaluating g at any iteration is equally expensive, we focus on $g(\emptyset)$ as a representative case and show below intuitively how to approximate it within a factor of $(1 \pm \epsilon)$ efficiently. The approach decomposes $g(\emptyset)$ as an infinite matrix power series and estimates it through truncation and simulation.

The average hitting time can be expressed as $g(\emptyset) = \frac{1}{r} \mathbf{e}_r^T (\mathbf{I}_r - \mathbf{Q})^{-1} \mathbf{e}_r = \frac{1}{r} \mathbf{e}_r^T \sum_{i=0}^{+\infty} \mathbf{Q}^i \mathbf{e}_r$ [99], where $r = |R|$ and \mathbf{Q} is the transition matrix between red nodes obtained by deleting the rows and columns of corresponding blue nodes in the random walk transition matrix $\mathbf{P} = \mathbf{D}^{-1} \mathbf{A}$. This infinite series naturally splits into two parts: $p_1 = \frac{1}{r} \mathbf{e}_r^T \sum_{i=0}^{\ell} \mathbf{Q}^i \mathbf{e}_r$ representing the first ℓ terms [99, 100], and $p_2 = \frac{1}{r} \mathbf{e}_r^T \sum_{i=\ell+1}^{+\infty} \mathbf{Q}^i \mathbf{e}_r$ representing the tail.

The truncated sum p_1 admits a probabilistic interpretation: each entry of $\sum_{i=0}^{\ell} \mathbf{Q}^i \mathbf{e}_r$ represents the expected number of times a bounded random walk of length ℓ visits each red node before absorption by a blue node. We estimate p_1 by simulating multiple bounded walks from each red node and computing empirical averages. This estimation approach is analyzed by Haddadan et al. [53, Lemma 4.3], whose proof uses Hoeffding's bound [101] and the union bound to show that with sufficient trials, the empirical estimate concentrates around the true value with high probability.

The tail p_2 captures contributions from walks exceeding ℓ steps. Since \mathbf{Q} is a substochastic matrix for an absorbing Markov chain, its spectral radius satisfies $\lambda < 1$, implying that $\|\mathbf{Q}^i\|$ decays exponentially as λ^i . By choosing ℓ sufficiently large based on the spectral radius and desired accuracy, we can bound $p_2 \leq \epsilon g(\emptyset)$, rendering the truncation error negligible.

Combining the estimation of p_1 and the bound on p_2 , we show that $\hat{g} = \frac{1}{r} \mathbf{e}_r^T \hat{h}$ satisfies $(1 - \epsilon)g(\emptyset) \leq \hat{g} \leq (1 + \epsilon)g(\emptyset)$ with high probability, where \hat{h} is the empirical average from simulated walks of sufficiently large length ℓ . We omit the technical details and provide the formal statement in Lemma 3.

Lemma 3. *Given an undirected connected graph $G = (V, E)$ with a valid bipartition $V = \{R, B\}$, we can find an estimate $\hat{g}(\emptyset) \in (1 \pm \epsilon)g(\emptyset)$ with probability at least $1 - 1/n^c$ (for some $c \geq 1$) in time*

$$O\left(\frac{cn \ln(n) \log^3(d_R/\epsilon(1 - \lambda))}{\epsilon^2 \log^3(1/\lambda)}\right),$$

where d_R is the average degree of red nodes, and λ is the spectral radius of $\mathbf{Q} = \mathbf{D}_R^{-1} \mathbf{A}_R$, with \mathbf{A}_R being the adjacency matrix of the induced subgraph $G[R]$ and \mathbf{D}_R the corresponding degree diagonal matrix.

Accelerating the Greedy Algorithm

Lemma 3 states that we can find good estimates $\hat{g}(\cdot) \in (1 \pm \epsilon)g(\cdot)$ relatively quickly and with high probability. The following theorem shows that using this approximation \hat{g} of g in a greedy algorithm still yields a constant-factor approximation for the BMAH problem.

Theorem 7. *The classic greedy algorithm that iteratively selects at most $\tau = \lceil 2k \ln(\frac{n^3}{\epsilon}) \rceil$ edges maximizing the marginal reduction of the function \hat{g} returns a set of shortcut edges F_τ satisfying*

$$g(F_\tau) \leq (2 + \epsilon)OPT, \quad (3)$$

where $OPT = g(F^*)$ and F^* is an optimal solution to BMAH with at most k edges.

The proof follows the classic greedy analysis for minimizing monotone supermodular functions [25]. When using the exact function g in the greedy algorithm, each greedy selection reduces at least $1/k$ of the gap to the optimal value OPT :

$$g(F_{i+1}) \leq \left(1 - \frac{1}{k}\right) g(F_i) + \frac{1}{k} OPT.$$

When using the estimate $\hat{g} \in (1 \pm \epsilon)g$ for greedy selection, we introduce a constant factor of $\frac{1+\epsilon}{1-\epsilon}$ in reducing this gap:

$$g(F_{i+1}) \leq \frac{1+\epsilon}{1-\epsilon} \left[\left(1 - \frac{1}{k}\right) g(F_i) + \frac{1}{k} OPT \right].$$

This is a first-order affine recurrence of the form $g(F_{i+1}) \leq a g(F_i) + b$, which has the closed-form bound $g(F_\tau) \leq a^\tau g(\emptyset) + \frac{1-a^\tau}{1-a} b$ [102]. By carefully choosing τ , we ensure the transient term $a^\tau g(\emptyset)$ is bounded by $\epsilon \cdot OPT$. Regardless of the choice of τ , the steady-state term $\frac{1-a^\tau}{1-a} b$ is bounded by $2OPT$. The sum of these two terms yields the desired $(2 + \epsilon)$ -approximation guarantee.

5.2.4 Algorithm for BMMH

The BMMH problem is not supermodular, so classic greedy algorithms do not yield constant-factor approximations. Instead, we propose two algorithms that approximate BMMH indirectly by reducing it to related problems.

Relating the Average and the Maximum Hitting Times

We start with the following theorem that bounds the maximum hitting time f via average hitting time g :

Theorem 8. *Given an undirected connected graph G with a valid bipartition $V = \{R, B\}$. Let $F \subseteq (R \times B) \setminus E$ be a set of non-edges. Then,*

$$g(F) \leq f(F) \leq 2|R|^{3/4}g(F), \quad (6)$$

and thus, an α -approximation for BMAH is a $(2|R|^{3/4}\alpha)$ -approximation for BMMH.

Intuition Behind the Bound. The key observation is that vertices with very large hitting times must be rare. Let S consist of vertices whose hitting time to the blue set is small (at most $\alpha g(\emptyset)$), and let S^c denote the exceptional vertices with large hitting times. Since the average hitting time from red to blue equals $g(\emptyset)$, the exceptional set satisfies $|S^c| \leq |R|/\alpha$.

To analyze the exceptional set's contribution, we contract all vertices in S into a single supernode. This contraction can only increase the time to reach S from S^c , providing a worst-case estimate. The contracted graph has only $|S^c| + 1$ vertices, allowing us to apply general hitting-time bounds of order $|S^c|^3$ [73, 103].

From any red vertex in S^c , the random walk has two stages. First, it reaches S from within S^c . Second, it proceeds from S to blue. By the strong Markov property, the expected total duration is at most the sum of these two stages. The first stage is bounded by $|S^c|^3$, while the second is at most $\alpha g(\emptyset)$ by definition of S .

This bound, $|S^c|^3 + \alpha g(\emptyset)$, reflects a tradeoff in choosing α . Smaller α reduces the second-stage cost but enlarges S^c , while larger α has the opposite effect. Optimizing this bound yields $\alpha = |R|^{3/4}$, showing that the maximum hitting time exceeds the average by at most a factor of $|R|^{3/4}$.

An Asymmetric k -Center Approach

Hitting times are inherently asymmetric. In general, $H_G(u, v) \neq H_G(v, u)$ [71]. This asymmetry suggests a connection to the asymmetric k -center problem [104], where choosing centers corresponds to selecting which red nodes to connect to blue nodes with shortcuts. We first provide the result, then explain the intuition behind the analysis.

Theorem 9. *Given a β -approximation algorithm for the asymmetric k -center problem, there is an $O(\beta d_m)$ -approximation for BMMH, where d_m is the maximum degree of the chosen vertices.*

This approach constructs an asymmetric k -center problem instance with $|R| + 1$ nodes, where all blue nodes are collapsed into a single representative point, and distances between nodes correspond to hitting times in the BMMH instance and form a quasi-metric space. The asymmetric k -center objective of minimizing the maximum distance from any point to its nearest center directly mirrors the BMMH objective of

minimizing the maximum hitting time. The key technical result establishes that the optimal value of k -center on this space lower bounds the optimal value of BMMH. This enables approximation analysis as follows. Let C^* denote the optimal k -center value. A β -approximation algorithm for k -center guarantees that any red node reaches a chosen center within expected time βC^* . From each center, there exists a shortcut edge to the blue nodes. When the random walk reaches a center of degree d , it traverses the shortcut with probability at least $\frac{1}{d+1}$, reaching the blue nodes in one additional step. Analyzing the expected hitting time using this two-phase process yields an $\mathcal{O}(\beta d_m)$ -approximation for BMMH, where d_m is the maximum degree among the chosen centers.

Chapter 6

Conclusion and Future Work

This thesis investigates fairness and diversity across two complementary domains: recommender systems and social network analysis. We develop algorithms for improving result diversity through pairwise distance maximization and topic-coverage maximization (papers A and B), for enhancing content exposure fairness via ranking adjustments and graph modifications (papers D and E), and for identifying densely connected communities with diverse relationship types (paper C). All problems are formulated as mathematical optimization problems, and the proposed algorithms come with provable performance guarantees or convergence properties.

From an application perspective, our methods correspond to (1) recommendation algorithms that carefully select, rank, or augment recommendation results on real-world platforms such as news aggregators (e.g., Google News), video platforms (e.g., YouTube and TikTok) and (2) network analysis algorithms capable of detecting fraudulent behaviors or identifying meaningful substructures, such as protein-protein interaction modules, which are valuable for downstream tasks including drug discovery and design. Together, these contributions provide a unified algorithmic framework for improving equity and diversity in large-scale networked systems.

6.1 Limitations and Future Work

We discuss limitations and future directions at both the conceptual level, spanning the thesis as a whole, and the technical level, specific to individual papers.

6.1.1 Broad Limitations and Directions

Definitions of Diversity and Fairness. Our diversity measures assume predefined topic taxonomies (paper B) or edge-color categories (paper C), and our fairness measures are group-based (papers D and E). These are reasonable assumptions, but alternatives exist. Individual fairness, intersectional fairness, and procedural fairness each capture aspects that our definitions do not. Studying alternative definitions of

fairness and diversity within the context of our problems is an interesting direction for future work.

Beyond exploring more definitions of fairness and diversity, there is also concern with whether the definitions we employ fully capture the intended concepts. In our recommender system studies, we quantified diversity through topic coverage and sequential sum diversity. While these metrics are mathematically tractable, they may not fully capture a user's subjective experience of diversity, which can be influenced by cognitive biases or aesthetic variety. Similarly, our fairness measures (PageRank and hitting-time fairness) assume that equity is achieved by rebalancing influence or accessibility across disjoint groups based on sensitive attributes. However, these technical metrics may oversimplify the social reality of fairness by focusing on mathematical parity without necessarily capturing broader qualitative impacts or accommodating the diverse ethical standards that real-world applications require.

Trade-offs with Other Objectives. Promoting diversity or fairness often creates a trade-off where gains in long-term satisfaction or content exposure come at the expense of short-term engagement metrics like click-through rate or subgraph density [19, 105]. While we currently address these trade-offs through constrained optimization, developing explicit multi-objective frameworks such as Pareto-optimal formulations would provide practitioners with more robust tools to navigate this engagement-diversity trade-off.

Data Privacy. Our models often assume access to granular data, such as user-item interaction probabilities (p_i) or sensitive demographic attributes. In real-world applications, this data is subject to strict privacy regulations (e.g., GDPR [106]). A critical future direction is the integration of Differential Privacy into our algorithms. Can we maximize diversity or enforce fairness constraints while ensuring that individual user preferences or sensitive attributes remain protected? Furthermore, the use of automated "graph interventions" to rewire networks raises ethical questions regarding platform transparency and user autonomy that merit deeper exploration.

6.1.2 Technical Open Problems

1. **Advanced Modeling of Continuation Probabilities (Papers A and B).** We assume item click probabilities p_i are given as input. Integrating learned prediction models, or studying the online learning variant where p_i must be estimated concurrently with diversity optimization, would connect our work to the bandit and explore-exploit literature [107, 108, 109, 110].
2. **Diverse Optimization Objectives (Papers A and B)** Topic coverage is the sole optimization objective. Can we integrate additional metrics (e.g., completion rate, click-through rate) within a unified formulation, potentially via constrained submodular optimization?

3. **Explore Different Diversity Measure (Paper A)** Apart from the sequential sum and sequential coverage diversity. Can we design approximation algorithms for other definitions of diversity, such as sequential Max-Min diversity?
4. **Dense Graph Generalization (Paper C)** Our algorithms work only for everywhere sparse graphs. Improving approximation results for dense graphs or establishing stronger lower bounds remains an open challenge.
5. **Dense Graph Generalization (Paper C)** We study the "at least h " colored-edge DSP. Can we develop algorithms for the exact and "at most h " variants? Can we still leverage the reduction to at least k node DSP to solve these variants?
6. **Provable Guarantees for Non-Convex Problems (Paper C)** The non-convex formulation currently requires machine-learning-based solutions. Can we reformulate the problem to enable approximation algorithms with provable guarantees, e.g., via convex relaxation or combinatorial techniques?
7. **Improved Hitting-Time algorithms (Paper E)** For fairness interventions based on hitting time, future research should aim to obtain improved approximation algorithms that avoid bicriteria approaches and tighten the gap between algorithmic guarantees and known lower bounds.

References

- [1] Jessica T Feezell, John K Wagner, and Meredith Conroy. Exploring the effects of algorithm-driven news sources on political behavior and polarization. *Computers in human behavior*, 116:106626, 2021.
- [2] Sarah J Jackson, Meredith D Clark, Deen Freelon, and Lori Lopez. How black twitter and other social media communities interact with mainstream news. *Knight Foundation*, 2018.
- [3] Stefan Kirchner and Elke Schüßler. The organization of digital marketplaces: Unmasking the role of internet platforms in the sharing economy. *Organization outside organization*, pages 131–154, 2019.
- [4] Amy Ross Arguedas, Craig Robertson, Richard Fletcher, and Rasmus Nielsen. Echo chambers, filter bubbles, and polarisation: A literature review. Technical report, Reuters Institute for the Study of Journalism, University of Oxford, 2022. URL <https://ora.ox.ac.uk/objects/uuid:6e357e97-7b16-450a-a827-a92c93729a08/files/rzw12z639q>. Accessed: 8 February 2026.
- [5] Frederic Gerdon, Ruben L Bach, Christoph Kern, and Frauke Kreuter. Social impacts of algorithmic decision-making: A research agenda for the social sciences. *Big Data & Society*, 9(1):20539517221089305, 2022.
- [6] Manish Raghavan. *The societal impacts of algorithmic decision-making*. Cornell University, 2021.
- [7] Michal Wiktor Krawczyk. A model of procedural and distributive fairness. *Theory and decision*, 70(1):111–128, 2011.
- [8] Arild Underdal and Taoyuan Wei. Distributive fairness: A mutual recognition approach. *Environmental science & policy*, 51:35–44, 2015.
- [9] Kees Van den Bos, Henk AM Wilke, and E Allan Lind. When do we need procedural fairness? The role of trust in authority. *Journal of Personality and social Psychology*, 75(6):1449, 1998.

-
- [10] Marie Christin Decker, Laila Wegner, and Carmen Leicht-Scholten. Procedural fairness in algorithmic decision-making: the role of public engagement. *Ethics and Information Technology*, 27(1):1, 2025.
- [11] Will Fleisher. What’s fair about individual fairness? In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 480–490, 2021.
- [12] Reuben Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 514–524, 2020.
- [13] Tim Rüz. Group fairness: Independence revisited. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 129–137, 2021.
- [14] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 155–166, 2012.
- [15] Sekharipuram S Ravi, Daniel J Rosenkrantz, and Giri Kumar Tayi. Heuristic and special case algorithms for dispersion problems. *Operations research*, 42(2):299–310, 1994.
- [16] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390, 2009.
- [17] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 15–22, 2016.
- [18] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *IJCAI*, volume 15, pages 1742–1748, 2015.
- [19] Yuying Zhao, Yu Wang, Yunchao Liu, Xueqi Cheng, Charu C Aggarwal, and Tyler Derr. Fairness and diversity in recommender systems: a survey. *ACM Transactions on Intelligent Systems and Technology*, 16(1):1–28, 2025.
- [20] Theodor Cimpanu, Alessandro Di Stefano, Cedric Perret, and The Anh Han. Social diversity reduces the complexity and cost of fostering fairness. *Chaos, Solitons & Fractals*, 167:113051, 2023.
- [21] Michael J Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987.

- [22] Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.
- [23] P. Hansen and I. D. Moon. Dispersing facilities on a network. In *TIMS/ORSA Joint National Meeting*, Washington, DC, 1988. Presentation.
- [24] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Local search for max-sum diversification. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 130–142. SIAM, 2017.
- [25] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming*, 14(1):265–294, 1978.
- [26] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.
- [27] Erica Coppolillo, Giuseppe Manco, and Aristides Gionis. Relevance meets diversity: A user-centric framework for knowledge exploration through recommendations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24*, page 490–501, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671949.
- [28] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
- [29] George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.
- [30] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. An analysis of approximations for maximizing submodular set functions—II. In *Polyhedral Combinatorics: Dedicated to the memory of DR Fulkerson*, pages 73–87. Springer, 2009.
- [31] Gruiua Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
- [32] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74, 2008.
- [33] Yuval Filmus and Justin Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 659–668. IEEE, 2012.

- [34] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680, 2014.
- [35] Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In *International Conference on Machine Learning*, pages 3311–3320. PMLR, 2019.
- [36] Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavifar, and Ola Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In *International Conference on Machine Learning*, pages 3829–3838. PMLR, 2018.
- [37] Shipra Agrawal, Mohammad Shadravan, and Cliff Stein. Submodular secretary problem with shortlists. *arXiv preprint arXiv:1809.05082*, 2018.
- [38] Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Mathematical Programming*, 154(1):225–247, 2015.
- [39] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *International Colloquium on Automata, Languages, and Programming*, pages 318–330. Springer, 2015.
- [40] Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, get more: Streaming submodular maximization with subsampling. *Advances in Neural Information Processing Systems*, 31, 2018.
- [41] Andrew V Goldberg. *Finding a maximum density subgraph*. University of California Berkeley, 1984.
- [42] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *International workshop on approximation algorithms for combinatorial optimization*, pages 84–95. Springer, 2000.
- [43] Samir Khuller and Barna Saha. On finding dense subgraphs. In *International colloquium on automata, languages, and programming*, pages 597–608. Springer, 2009.
- [44] Reid Andersen and Kumar Chellapilla. Finding dense subgraphs with size bounds. In *International workshop on algorithms and models for the web-graph*, pages 25–37. Springer, 2009.
- [45] Aris Anagnostopoulos, Luca Becchetti, Adriano Fazzzone, Cristina Menghini, and Chris Schwegelshohn. Spectral relaxations and fair densest subgraphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20*, pages 35–44, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3340531.3412036.

- [46] Atsushi Miyachi, Tianyi Chen, Konstantinos Sotiropoulos, and Charalampos E. Tsourakakis. Densest diverse subgraphs: How to plan a successful cocktail party with diversity. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23*, pages 1710–1721, New York, NY, USA, August 2023. Association for Computing Machinery. doi: 10.1145/3580305.3599483.
- [47] Sotiris Tsioutsoulouklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-aware pagerank. In *Proceedings of the Web Conference 2021*, pages 3815–3826, 2021.
- [48] Sotiris Tsioutsoulouklis, Evaggelia Pitoura, Konstantinos Semertzidis, and Panayiotis Tsaparas. Link recommendations for pagerank fairness. In *Proceedings of the ACM Web Conference 2022*, pages 3541–3551, 2022.
- [49] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3289–3295. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/456.
- [50] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P Gummadi, Adrian Weller, and Baharan Mirzasoleiman. Crosswalk: Fairness-enhanced node representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11963–11970, 2022.
- [51] Francesco Fabbri, Yanhao Wang, Francesco Bonchi, Carlos Castillo, and Michael Mathioudakis. Rewiring what-to-watch-next recommendations to reduce radicalization pathways. In *Proceedings of the ACM Web Conference 2022*, pages 2719–2728, 2022.
- [52] Corinna Coupette, Stefan Neumann, and Aristides Gionis. Reducing exposure to harmful content via graph rewiring. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–334, 2023.
- [53] Shahrzad Haddadan, Cristina Menghini, Matteo Riondato, and Eli Upfal. Republik: Reducing polarized bubble radius with link insertions. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 139–147, 2021.
- [54] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. Minimizing polarization and disagreement in social networks via link recommendation. *Advances in Neural Information Processing Systems*, 34:2072–2084, 2021.
- [55] Vijay V Vazirani. *Approximation algorithms*, volume 1. Springer, 2001.
- [56] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

-
- [57] Dorit S Hochba. Approximation algorithms for NP-hard problems. *ACM Sigact News*, 28(2):40–52, 1997.
- [58] Daniel D Sleator and Robert E Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.
- [59] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.
- [60] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.
- [61] Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.
- [62] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art: Bonn 1982*, pages 235–257. Springer, 1983.
- [63] Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.
- [64] Donald M Topkis. *Supermodularity and complementarity*. Princeton university press, 1998.
- [65] Edo Liberty and Maxim Sviridenko. Greedy minimization of weakly supermodular set functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, pages 19:1–19:11. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.
- [66] Krause Andreas and Golovin Daniel. Submodular function maximization. *Tractability*, pages 71–104, 2014. doi: 10.1017/cbo9781139177801.004.
- [67] James G Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.
- [68] Dominic JA Welsh. *Matroid theory*. Courier Corporation, 2010.
- [69] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.
- [70] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford infolab, 1999.
- [71] László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

- [72] Thomas Sauerwald and He Sun. Spectral graph theory and applications, lecture 7: Hitting time and cover time of random walks. Lecture notes, 2011. URL <https://resources.mpi-inf.mpg.de/departments/dl/teaching/ws11/SGT/Lecture7.pdf>. WS 2011/2012, Max Planck Institute for Informatics.
- [73] Eugene L Lawler. *Combinatorial optimization: networks and matroids*. Courier Corporation, 2001.
- [74] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.
- [75] Chandra Chekuri, Kent Quanrud, and Manuel R Torres. Densest subgraph: Supermodularity, iterative peeling, and flow. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1531–1555. SIAM, 2022.
- [76] Digvijay Boob, Yu Gao, Richard Peng, Saurabh Sawlani, Charalampos Tsourakakis, Di Wang, and Junxing Wang. Flowless: Extracting densest subgraphs without flow computations. In *Proceedings of The Web Conference 2020*, pages 573–583, 2020.
- [77] Subhash Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.
- [78] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 201–210, 2010.
- [79] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.
- [80] Dmitri Burago, Yuri Burago, Sergei Ivanov, et al. *A course in metric geometry*, volume 33. American Mathematical Society Providence, 2001.
- [81] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.
- [82] Michael R Garey, David S Johnson, et al. A guide to the theory of NP-Completeness. *Computers and intractability*, pages 37–79, 1990.
- [83] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.
- [84] Ján Plesník. On the computational complexity of centers locating in a graph. *Aplikace matematiky*, 25(6):445–452, 1980.
- [85] Martin E Dyer and Alan M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985.

-
- [86] Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.
- [87] Wallace Alvin Wilson. On quasi-metric spaces. *American Journal of Mathematics*, 53(3):675–684, 1931.
- [88] Rina Panigrahy and Sundar Vishwanathan. An $O(\log^*n)$ approximation algorithm for the asymmetric p-center problem. *Journal of Algorithms*, 27(2): 259–268, 1998.
- [89] Julia Chuzhoy, Sudipto Guha, Eran Halperin, Sanjeev Khanna, Guy Kortsarz, Robert Krauthgamer, and Joseph Naor. Asymmetric k-center is \log^*n -hard to approximate. *Journal of the ACM (JACM)*, 52(4):538–551, 2005.
- [90] Aaron Archer. Two $O(\log^*k)$ -approximation algorithms for the asymmetric k-center problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 1–14. Springer, 2001.
- [91] Jon Kleinberg, Emily Ryu, and Éva Tardos. Ordered submodularity and its applications to diversifying recommendations. *arXiv preprint arXiv:2203.00233*, 2022.
- [92] Sotiris Tsioutsoulouklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-Aware PageRank. In *Proceedings of the Web Conference*, pages 3815–3826, 2021.
- [93] Christian Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of ∞ n. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.
- [94] Panos M Pardalos and Naina Kooor. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, 46(1):321–328, 1990.
- [95] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.
- [96] Laurent Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, 158(1):575–585, 2016.
- [97] Lukás Adam and Václav Mácha. Projections onto the canonical simplex with additional linear inequalities. *Optimization Methods and Software*, 37(2):451–479, 2022.
- [98] JG Kemery and JL Snell. Finite markov chains: With a new appendix "generalization of a fundamental matrix", 1960.

- [99] Pan Peng, Daniel Lopatta, Yuichi Yoshida, and Gramoz Goranci. Local algorithms for estimating effective resistance. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1329–1338, 2021.
- [100] Alexandr Andoni, Robert Krauthgamer, and Yosef Pogrow. On solving linear systems in sublinear time. *arXiv preprint arXiv:1809.02995*, 2018.
- [101] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.
- [102] Ronald L Graham. *Concrete mathematics: a foundation for computer science*. Pearson Education India, 1994.
- [103] Graham Brightwell and Peter Winkler. Maximum hitting time for random walks on graphs. *Random Structures & Algorithms*, 1(3):263–276, 1990.
- [104] Oded Kariv and S Louis Hakimi. An algorithmic approach to network location problems. I: The p-centers*. *SIAM journal on applied mathematics*, 37(3): 513–538, 1979.
- [105] Elvin Isufi, Matteo Pocchiari, and Alan Hanjalic. Accuracy-diversity trade-off in recommender systems via graph convolutions. *Information Processing & Management*, 58(2):102459, 2021.
- [106] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A practical guide, 1st ed.*, Cham: Springer International Publishing, 10(3152676):10–5555, 2017.
- [107] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.
- [108] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.
- [109] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM conference on recommender systems*, pages 31–39, 2018.
- [110] Marko Balabanović. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction*, 8(1): 71–102, 1998.

Part II
Included Papers

