Optimal Approximabilities beyond CSPs:
Thesis Proposal

Euiwoong Lee
Computer Science Department
Carnegie Mellon University

August 26, 2016

Abstract

This thesis tries to expand the frontiers of approximation algorithms, with respect to the range of optimization problems as well as mathematical tools for algorithms and hardness. We show tight approximabilities for various fundamental problems in combinatorial optimization.

We organize these problems into three categories — new and applied constraint satisfaction problems (CSP), hypergraph coloring under promise, and graph covering/cut problems. Conceptually extending the traditional problems like CSP, coloring, and covering, they include important open problems in approximation algorithms. Some problems are inspired by other areas of theory of computing including coding theory and computational economics.

We also try to present new techniques to prove tight approximabilities. For coloring and cut problems, we provide a unified framework to show strong hardness results, encompassing many previous techniques for related problems and yielding new results. For CSPs, our new techniques often stem from a combination of ideas from multiple traditional techniques that have been developed separately.
# Contents

1 Introduction 1

2 Preliminaries 3

3 Constraint Satisfaction Problems 4
   3.1 New CSPs 5
      3.1.1 Hard / Balanced CSPs 5
      3.1.2 Symmetric CSPs 6
   3.2 Applied CSPs 7
      3.2.1 Unique Coverage 7
      3.2.2 Decoding LDPC Codes 8
      3.2.3 Graph Pricing 9

4 Hypergraph Coloring 10
   4.1 Min Coloring 11
   4.2 Max 2-Coloring 12

5 Covering and Cut 13
   5.1 $H$-Transversal and Graph Partitioning 13
      5.1.1 $H$-Transversal / Packing 14
      5.1.2 Partitioning Graphs into Small Pieces 15
   5.2 Cut and Interdiction 16
      5.2.1 Multicut 16
      5.2.2 Length-bounded Cut / Shortest Path Interdiction 17
      5.2.3 Firefighter 19

6 Future Directions 20
   6.1 Ongoing Projects 20
      6.1.1 Fixed Parameter Tractability of Edge $k$-Cut 20
      6.1.2 Blocking Arborescences 20
   6.2 Open Problems from Previous Works 20
      6.2.1 Symmetric CSPs 20
      6.2.2 Shortest Path Interdiction 21
      6.2.3 $k$-Path Packing and $H$-Edge Transversal/Packing 21
   6.3 Long-term Directions 21
      6.3.1 NP-hardness 21
      6.3.2 Problems with Wider Gaps 22
1 Introduction

Combinatorial optimization is a topic that involves searching for an optimal object from a finite set of objects. Since its establishment as a coherent discipline in the 1950’s [Sch05], it has been actively studied from mathematics, operations research, and computer science.

Its inherent interdisciplinarity is based on three pillars that are indispensable for complete understanding of each problem in the topic — algorithms, hardness, and mathematical programming. Edmonds’ seminar paper for MAXIMUM MATCHING [Edm65] pioneered the relationship between efficient algorithms and properties of associated polyhedra formed by natural linear programming (LP) relaxations. This tight relationship has been proved to hold for so many combinatorial optimization problems, that one of the most comprehensive textbooks in the field is devoted to it [Sch03].

On the other hand, the same paper successfully established polynomial time as the major criterion to measure an algorithm’s efficiency. This work is followed by the Cook-Levin theorem [Coo71, Lev73] and Karp’s 21 NP-complete problems [Kar72] that show NP-completeness of some fundamental combinatorial optimization problems, implying they will not admit an efficient algorithm unless $\text{P} \neq \text{NP}$. Active subsequent research has successfully identified the complexity of each problem, so already in the early 21st century “almost every combinatorial optimization problem has since been either proved to be polynomial-time solvable or NP-complete” [Sch03].

Unfortunately, numerous combinatorial optimization problems are proved to be NP-complete, so polynomial-time algorithms are unlikely to exist for those problems. Approximation algorithms are considered as one of most natural ways to circumvent this difficulty. Formally, for a maximization (resp. minimization) problem with an objective function $f$, an algorithm is called a $c$-approximation algorithm for some $c < 1$ (resp. $c > 1$) if for every instance of the problem, it returns a solution $s$ in polynomial time with the guarantee that $f(s) \geq c \cdot f^*$ (resp. $f(s) \leq c \cdot f^*$), where $f^*$ is the objective function value of an optimal solution. In this context, a problem is considered to be completely understood if we find $c$ such that

- A $c$-approximation algorithm exists.
- For any $c' > c$ (resp. $c' < c$), if $\text{P} \neq \text{NP}$, there is no $c'$-approximation algorithm.

Study of approximation algorithms and hardness of approximation have resulted in beautiful results that tighten the relationship between algorithms, hardness, and mathematical programming. From the algorithms side, the work of Goemans and Williamson [GW95] showed the first application of semidefinite programming (SDP) to approximation algorithms, strictly improving previous LP-based algorithms. The search for more powerful convex relaxations beyond LP and SDP has produced several LP and SDP hierarchies, including Lovász-Schrijver [LS91], Sherali-Admas [SA90], and Sum-of-Squares [Par00, Las01].

From the hardness side, the celebrated PCP theorem [ALM+98, AS98] first proved that there exists a universal constant $c < 1$ such that it is NP-hard to approximate MAX 3-SAT within a factor of $c$. The parallel repetition theorem [Raz98] and the introduction of long codes [BGS98] created a framework to prove strong hardness results for many problems, culminating in Hästad’s optimal inapproximability results for various constraint satisfaction problems [Hä01]. The Unique Games Conjecture [Kho02] (UGC), though its truth seems far from being settled, suggested a even tighter relationship between mathematical programming and hardness of approximation. Raghavendra [Rag08] showed that assuming the UGC, for every problem in the wide class of MAX CSP, no polynomial time algorithm outperforms an algorithm based on natural SDP re-
laxations. Conversely, ideas from computational hardness results often led to limitations of convex relaxations [KV05, Sch08, Tul09, BCK15] for some problems by showing integrality gaps.

This thesis tries to continue this line of work to prove tight approximabilities for combinatorial optimization problems, and strengthen it in the following two aspects.

**Expanding the Range of Problems.** Most of the classic aforementioned algorithms and hardness results, including the Goemans-Williamson algorithm [GW95], the PCP theorem [ALM+98, AS98], the parallel repetition theorem [Raz98], Håstad’s optimal inapproximability results [Hås01], the UGC [Kho02], and the optimality of SDP-based algorithms [Rag08] focus on one class of combinatorial optimization problems called Max CSP. While it is a wide class that contains many natural problems including Max 3-SAT, Max 3-LIN, and Max Cut, there are numerous fundamental problems not captured by this class.

One of the most fundamental classes not captured by Max CSP is covering/packing problems, which include fundamental graph problems including Maximum Matching, Maximum Disjoint Paths, Minimum Cut, and played an important role in the development of exact combinatorial optimization algorithms. Although the most general problems in the class (e.g., Minimum Set Cover and Maximum Independent Set), or some simple problems (e.g., Minimum Vertex Cover) are well understood, approximabilities of many fundamental graph problems in the class are not completely understood yet.

In addition to classical combinatorial optimization problems, the success of formulating numerous computational tasks as optimization problems creates new challenges and opportunities for approximation algorithms. Such examples arise from various academic fields including information/coding theory, machine learning/data mining, and computational economics. The thesis tries to study approximabilities of the following natural problems, classified into three categories.

- **New and applied CSPs (Section 3):** We introduce three variants of Max CSP, called HARD CSP, BALANCE CSP, and SYMMETRIC CSP. Our results indicate current hardness theories for Max CSP can be extended to its generalizations (HARD CSP, BALANCE CSP) to prove much stronger hardness, or can be significantly simplified for a special case (SYMMETRIC CSP).

  We also study problems motivated by error correcting codes and computational economics. These problems a priori do not look like a traditional CSP, but careful formulations enable them to admit technical ideas that have been previously used for CSPs.

- **Coloring (Section 4):** We study complexity of hypergraph coloring problems when instances are promised to have a structure much stronger than admitting a proper 2-coloring. These problems naturally capture a wider class of combinatorial optimization problems (e.g., scheduling), and exhibit a richer connection to discrepancy theory. We prove strong hardness results of coloring hypergraphs under certain promises. Our results are almost optimal in the sense that stronger promises will lead to efficient algorithms to color hypergraphs with few colors.

- **Covering (Section 5):** We study the $H$-TRANSVERSAL, where given a graph $G$ and a fixed “pattern” graph $H$, the goal is to remove the minimum number of vertices from $G$ to make sure it does not include $H$ as a subgraph. We show an almost complete characterization of the approximability of $H$-TRANSVERSAL depending on properties of $H$. Our algorithms use the algorithm for $k$-VERTEX SEPARATOR as a subroutine, where given a graph $G$, the problem
asks to remove the minimum number of vertices so that each connected component has at most \( k \) vertices.

We also study various cut problems on graphs, where the goal is to remove the minimum number of vertices or edges to cut desired paths or cycles. We give improved hardness results for Directed Multicut, Length-Bounded Cut, Shortest Path Interdiction, and RMFC.

Unifying, Reinterpreting, and Creating Tools. To answer the problems stated above, we extend the techniques developed mainly for Max CSP and Minimum Vertex Cover. While the current techniques are generally applicable for Max CSP, many other classes of problems lack such a general framework. While it is impossible to create tools applicable to every combinatorial optimization problem, another goal of this thesis is to provide a toolkit for wide classes of problems. We believe that the tools presented in this thesis will be useful to study the associated classes beyond the problems we study.

This task is based on unifying the techniques of previous work. Even though most of them have implicitly shared a large portion of common ideas and technical work, different characteristics of problems that require specialized ideas make it hard to unify them in a common framework. For each class of problems we study, we present a powerful framework that simultaneously captures those specialized ideas and is applicable for many problems in the class. Unifying the existing tools often led to creating new techniques by reinterpreting them or combining ideas from two independently studied techniques.

- For hypergraph coloring, we present a recipe to prove strong hardness under the promise that hypergraphs not only admit a coloring with few colors but also have an additional structure. This encompasses numerous previous results on hypergraph coloring in one framework, and yields new results.

- Unique Coverage is an interesting problem in the sense that it is Max CSP by definition, but it exhibits properties closer to Minimum Set Cover. We prove nearly optimal NP-hardness of Unique Coverage by combining ideas from Max CSP and Minimum Set Cover. The techniques to prove these two classical results have been developed separately.

- Our results for various cut problems including Feedback Vertex Set, Directed Multicut, Length-Bounded Cut, Shortest Path Interdiction, and RMFC are based on length-control dictatorship tests. They are inspired by the earlier results by Bansal and Khot [BK10] and Svensson [Sve13]. We give a new interpretation of their ideas as exploiting the existence of good fractional solutions, and combine LP gap instances to show our results.

2 Preliminaries

Unique Games. We introduce the Unique Games Conjecture.

**Definition 2.1.** An instance \( \mathcal{L}(B(V_B \cup W_B, E_B), [R], \{\pi(v, w)\}_{(v, w) \in E_B}) \) of Unique Games consists of a biregular bipartite graph \( B(V_B \cup W_B, E_B) \) and a set \([R]\) of labels. For each edge \((v, w) \in E_B\) there is a constraint specified by a permutation \(\pi(v, w) : [R] \to [R]\). The goal is to find a labeling \( l : V_B \cup W_B \to [R] \) of the vertices such that as many edges as possible are satisfied, where an edge \( e = (v, w) \) is said to be satisfied if \( l(v) = \pi(v, w)(l(w)) \).
**Definition 2.2.** Given a Unique Games instance \( \mathcal{L}(B(V_B \cup W_B, E_B), [R], \{\pi(v, w)\}_{(v, w) \in E_B}) \), let \( \text{Opt}(\mathcal{L}) \) denote the maximum fraction of simultaneously-satisfied edges of \( \mathcal{L} \) by any labeling, i.e.

\[
\text{Opt}(\mathcal{L}) := \frac{1}{|E|} \max_{l : V_B \cup W_B \rightarrow [R]} \left| \{e \in E : l \text{ satisfies } e\} \right|.
\]

**Conjecture 2.3** (The Unique Games Conjecture [Kho02]). For any constants \( \eta > 0 \), there is \( R = R(\eta) \) such that, for a Unique Games instance \( \mathcal{L} \) with label set \([R]\), it is NP-hard to distinguish between

- \( \text{opt}(\mathcal{L}) \geq 1 - \eta \).
- \( \text{opt}(\mathcal{L}) \leq \eta \).

We call a computational task **UG-hard** if it is NP-hard assuming the Unique Games Conjecture.

**LP/SDP and Integrality Gaps.** Linear programming (LP) concerns the problem of maximizing or minimizing a linear function over a polyhedron. It is expressed as

\[
\begin{align*}
\text{maximize (minimize)} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0,
\end{align*}
\]

where \( x \in \mathbb{R}^n \) is a variable, and \( c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \) are fixed constants.

Semidefinite programming (SDP) concerns the problem of maximizing or minimizing a linear function over the intersection of a positive semidefinite cone and an affine space. It is expressed as

\[
\begin{align*}
\text{maximize (minimize)} & \quad c^T x \\
\text{subject to} & \quad \sum_{i=1}^n x_i A_i \preceq B \\
& \quad x \geq 0,
\end{align*}
\]

where \( x \in \mathbb{R}^n \) is a variable, and \( c \in \mathbb{R}^n, A_i \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{m \times m} \) are fixed constants. Note that \( A \preceq B \) if and only if \( y^T Ay \leq y^T By \) for all \( y \in \mathbb{R}^m \).

Given a maximization problem \( P \), we often consider a natural LP or SDP relaxation \( R \). The optimum of such relaxation \( \text{OPT}_R \) is at least as big as the integral optimum \( \text{OPT}_P \). The **integrality gap** of the relaxation \( R \) is defined to be the supremum of \( \frac{\text{OPT}_R}{\text{OPT}_P} \) over every instance of \( P \). For the minimization problem, it is defined to be the supremum of \( \frac{\text{OPT}_P}{\text{OPT}_R} \).

### 3 Constraint Satisfaction Problems

Constraint Satisfaction Problems (CSPs) are among the most fundamental and well-studied class of optimization problems. For a fixed domain \( D \), a CSP is specified by a finite set \( \Pi = \{P_1, \ldots, P_l\} \) of relations, where each relation \( P_i \) is a subset of \( D^{k_i} \) for some \( k_i \in \mathbb{N} \). Given such \( \Pi \), MAX CSP(\( \Pi \)) is defined as follows.

**MAX CSP(\( \Pi \))**
Input: A set of variables $X = \{x_1, ..., x_n\}$ and a collection of constraints $C = \{C_1, ..., C_m\}$. Each constraint $C_i$ is an expression of the form $R(x_{i_1}, ..., x_{i_k})$ where $R$ is a relation of arity $k$ contained in $\Pi$, and $x_{i_j}$'s are variables.

Output: An assignment $\sigma : X \rightarrow D$.

Goal: Maximize the number of satisfied constraints. A constraint $R(x_{i_1}, ..., x_{i_k})$ is satisfied when $(\sigma(x_{i_1}), ..., \sigma(x_{i_k})) \in R$.

### 3.1 New CSPs

#### 3.1.1 Hard / Balanced CSPs

We consider two natural extensions of Max CSP(II).

**BALANCE CSP(II)**

Input: A set of variables $X = \{x_1, ..., x_n\}$ and a collection of constraints $C = \{C_1, ..., C_m\}$.

Output: A balanced assignment $\sigma : X \rightarrow D$. An assignment is called balanced if for each $q \in D$, $|\sigma^{-1}(q)| = \frac{n}{|D|}$.

Goal: Maximize the number of satisfied constraints.

The notion of BALANCE CSP is interesting both practically and theoretically. Partitioning a set of objects into equal-sized subsets with desired properties is a basic scheme used in Divide-and-Conquer algorithms. BALANCE CUT, also known as MAXIMUM BISECTION, is one of the most well-known examples of BALANCE CSP. Theoretically, the balance constraint is one of the simplest non-local constraints where the current algorithmic and hardness results on ordinary CSPs do not work.

**HARD CSP(II)**

Input: A set of variables $X = \{x_1, ..., x_n\}$, a collection soft constraints $S = \{C_1, ..., C_{m_s}\}$, and a collection hard constraints $H = \{H_1, ..., H_{m_h}\}$.

Output: An assignment $\sigma : X \rightarrow D$ that satisfies every hard constraint.

Goal: Maximize the number of satisfied soft constraints.

HARD CSP contains every MAX CSP by definition, and also several additional fundamental combinatorial optimization problems, such as (HYPERGRAPH) INDEPENDENT SET, MULTICUT, GRAPH $k$-COLORING, and many other covering/packing problems. While every assignment is feasible in ordinary MAX CSP, in HARD CSP only certain assignments that satisfy all the hard constraints are considered as feasible, giving a more general framework to study combinatorial optimization problems.

By the seminal work of Schaefer [Sch78], there are only three nontrivial classes of Boolean CSPs for which satisfiability can be checked in polynomial time: 2-SAT, HORN-SAT, and LIN-MOD-2. Among them, only MAX 2-SAT and MAX HORN-SAT admit a robust algorithm, which outputs an assignment satisfying at least $(1 - g(\epsilon))$ fraction of constraints given a $(1 - \epsilon)$-satisfiable instance,
where \( g(\epsilon) \to 0 \) as \( \epsilon \to 0 \), and \( g(0) = 0 \). We study how balance and hard constraints affect approximabilities of these two problems. More specifically, we ask whether each variant admits a robust algorithm or a constant factor approximation algorithm and obtain the following results. Given an assignment \( \sigma : X \to D \), let \( \text{VAL}(\sigma) \) be the fraction of constraints satisfied by \( \sigma \), and let \( \text{OPT} \) indicate the fraction of constraints satisfied by an optimal assignment (satisfied soft constraints for HARD CSP).

**Theorem 3.1.** There exists an absolute constant \( \delta > 0 \) such that given an instance \( I \) of BALANCE HORN 2-SAT (special case of BALANCE 2-SAT and BALANCE HORN-SAT), it is NP-hard to distinguish between the following cases.

- \( \text{OPT} = 1 \)
- \( \text{OPT} \leq 1 - \delta \)

**Theorem 3.2.** For any \( \epsilon > 0 \), there is a randomized algorithm such that given an instance \( I \) of BALANCE SAT, in time \( \text{poly}(\text{size}(I), \frac{1}{\epsilon}) \), outputs \( \sigma \) with \( \text{VAL}(\sigma) \geq (\frac{3}{4} - \epsilon)\text{OPT}(I) \) with constant probability.

**Theorem 3.3.** For any \( \epsilon > 0 \), given an instance \( I \) of HARD 2-SAT, it is UG-hard to distinguish the following cases.

- \( \text{OPT} \geq 1 - \epsilon \)
- \( \text{OPT} \leq \epsilon \)

**Theorem 3.4.** For any \( \epsilon > 0 \), given an instance \( I \) of HARD HORN 3-SAT, it is NP-hard to distinguish between the following cases.

- \( \text{OPT} \geq 1 - \epsilon \)
- \( \text{OPT} \leq \epsilon \)

The above theorems imply that for both MAX 2-SAT and MAX HORN-SAT, balance constraints rule out robust algorithms but still allow constant-factor approximation algorithms, while hard constraints rule out both robust algorithms and constant-factor approximation algorithms.

### 3.1.2 Symmetric CSPs

In this section we assume that the domain is Boolean (i.e., \( D = \{0, 1\} \)). Recent works on approximability of CSPs focus on characterizing every CSP according to its approximation resistance. We define random assignments to be the class of algorithms that assign \( x_i \leftarrow 1 \) with probability \( \alpha \) independently for some \( \alpha \in [0, 1] \). A CSP is called approximation resistant, if for any \( \epsilon > 0 \), it is NP-hard to have a \( (\rho^* + \epsilon) \)-approximation algorithm, where \( \rho^* \) is the approximation ratio achieved by the best random assignment. Even assuming the UGC, the complete characterization of approximation resistance has not been found, and previous works either change the notion of approximation resistance or study a subclass of CSPs to find a characterization, and more general results tend to suggest more complex characterizations.

We study a natural subclass of CSPs where the domain is Boolean and the constraint language \( \Pi \) has one predicate \( Q \) which is symmetric — for any permutation \( \pi : [k] \to [k], (x_1, \ldots, x_k) \in Q \) if and only if \( (x_{\pi(1)}, \ldots, x_{\pi(k)}) \in Q \). Equivalently, for every such \( Q \), there exists \( S \subseteq [k] \cup \{0\} \)
such that \((x_1, \ldots, x_k) \in Q\) if and only if \((x_1 + \cdots + x_k) \in S\). Let **Symmetric CSP** \((S)\) without negation denote such a symmetric CSP. We also study **Symmetric CSP** \((S)\) with negation where each constraint \(C_j\) is specified by a tuple \((x_j, \ldots, x_{j,k})\) as well as \(b_{j,1}, \ldots, b_{j,k}\) and satisfied if \(\left((x_{j,1} \oplus b_{j,1}) + \cdots + (x_{j,k} \oplus b_{j,k})\right) \in S\) where \(\oplus\) denotes the addition in \(\mathbb{F}_2\) and \(+\) denotes the addition in \(\mathbb{Z}\).

While this is a significant restriction, it is a natural one that still captures the following fundamental problems, such as **Max SAT**, **Max Not-All-Equal-SAT**, **Max t-out-of-k-SAT** (with negation), and **Max Cut**, **Max-Set-Splitting**, **Discrepancy Minimization** (without negation).

There is a simple sufficient condition to be approximation resistant due to Austrin and Mossel [AM09] with negation, and due to Austrin and Håstad [AH13] without negation. For \(s \in [k] \cup \{0\}\), let \(P(s) \in \mathbb{R}^2\) be the point defined by \(P(s) := \left(\frac{s}{k}, \frac{s(s-1)}{k(k-1)}\right)\). For any \(s\), \(P(s)\) lies on the curve \(y = x^2 - \frac{x}{k-1}\), which is slightly below the curve \(y = x^2\) for \(x \in [0, 1]\). Given a subset \(S \subseteq [k] \cup \{0\}\), let \(P_S := \{P(s) : s \in S\}\) and conv\(P_S\) be the convex hull of \(P_S\). For symmetric CSPs, the conditions of [AH13] and [AM09] depend on whether this convex hull intersects a certain curve or a point.

For **Symmetric CSP** \((S)\) without negation, the condition becomes whether conv\(P_S\) intersects the curve \(y = x^2\). If we let \(s_{\min}\) and \(s_{\max}\) be the minimum and maximum number in \(S\) respectively, by convexity of \(y = \frac{k}{k-1}x^2 - \frac{x}{k-1}\), it is equivalent to that the line passing through \(P(s_{\min})\) and \(P(s_{\max})\) and \(y = x^2\) intersect, which is again equivalent to

\[
\frac{(s_{\max} + s_{\min} - 1)^2}{k - 1} \geq \frac{4s_{\max}s_{\min}}{k}. \tag{1}
\]

For **Symmetric CSP** \((S)\) with negation, the condition of Austrin and Mossel [AM09] is simplified to that conv\(P_S\) contains the point \((\frac{1}{2}, \frac{1}{4})\). We suggest that these simplified sufficient conditions might also be necessary and thus precisely characterize approximation resistance. We prove it for two natural special cases (which capture all problems mentioned in the last paragraph) for both symmetric CSPs with/without negation, and provide reasons that we believe this is true at least for symmetric CSPs without negation.

**Conjecture 3.5.** For \(S \subseteq [k - 1]\), **Symmetric CSP** \((S)\) without negation is approximation resistant if and only if (1) holds.

**Theorem 3.6.** If \(S \subseteq [k - 1]\) and \(S\) is either an interval or even, **Symmetric CSP** \((S)\) without negation is approximation resistant if and only if (1) holds (the hardness claim, i.e., the “if” part, is under the Unique Games Conjecture).

**Theorem 3.7.** If \(S \subset [k] \cup \{0\}\) and \(S\) is either an interval or even, **Symmetric CSP** \((S)\) with negation is approximation resistant if and only if conv\(P_S\) contains \((\frac{1}{2}, \frac{1}{4}\) \(\{\) hardness claim, i.e., the “if” part, is under the Unique Games Conjecture).

### 3.2 Applied CSPs

#### 3.2.1 Unique Coverage

We study the following natural problem that models numerous practical situations arising from wireless networks, radio broadcast, and envy-free pricing.
Unique Coverage

Input: A universe $V$ of $n$ elements and a collection $E$ of $m$ subsets of $V$.

Output: $S \subseteq V$.

Goal: Maximize the number of $e \in E$ that intersects $S$ in exactly one element.

When each $e \in E$ has size at most $k$, this problem is also known as 1-in-$k$ Hitting Set. While this problem can be captured as a Max CSP, this problem differs from other famous Max CSP in the sense that arities of constraints (sizes of $e \in E$) can be different and grow with $n$ so that the traditional results for Max CSP are not applicable. It admits a simple $\Omega(1/\log^2 k)$-approximation algorithm.

For constant $k$, we prove that 1-in-$k$ Hitting Set is NP-hard to approximate within a factor $O(1/\log k)$. This improves the result of Guruswami and Zhou [GZ12], who proved the same result assuming the Unique Games Conjecture.

Theorem 3.8. Assuming $P \neq NP$, for large enough constant $k$, there is no polynomial time algorithm that approximates 1-in-$k$ HS within a factor better than $O(1/\log^2 k)$.

For Unique Coverage, we prove that it is hard to approximate within a factor $O(1/\log^2 n)$ for any $\epsilon > 0$, unless NP admits quasipolynomial time algorithms. This improves the results of Demaine et al. [DFHS08], including their $\approx 1/\log^{1/3} n$ inapproximability factor which was proven under the Random 3SAT Hypothesis.

Theorem 3.9. Assuming $NP \not\subseteq QP$, for any $\epsilon > 0$, there is no polynomial time algorithm that approximates Unique Coverage within a factor better than $1/\log^{1-\epsilon} n$.

Our simple proof combines ideas from two classical inapproximability results for Minimum Set Cover and Max CSP, made efficient by various derandomization methods based on bounded independence.

3.2.2 Decoding LDPC Codes

Low-density parity-check (LDPC) codes are a class of linear error correcting codes originally introduced by Gallager [Gal62] and that have been extensively studied in the last decades. A $(d_v, d_c)$-LDPC code of block length $n$ is described by a parity-check matrix $H \in \mathbb{F}_2^{m \times n}$ (with $m \leq n$) having $d_v$ ones in each column and $d_c$ ones in each row. In many studies of LDPC codes, random LDPC codes have been considered. For instance, Gallager studied in his thesis the distance and decoding-error probability of an ensemble of random $(d_v, d_c)$-LDPC codes. Random $(d_v, d_c)$-LDPC codes were further studied in several works (e.g., [SS94, Mac99, RU01, MB01, DPT+02, LS02, KRU12]). The reasons why random $(d_v, d_c)$-LDPC codes have been of significant interest are their nice properties, their tendency to simplify the analysis of the decoding algorithms and the potential lack of known explicit constructions for properties satisfied by random codes.

Sipser and Spielman [SS94] gave a linear-time decoding algorithm correcting a constant fraction of errors (for $d_v, d_c = O(1)$). More precisely, the linear-time decoding algorithm of Sipser-Spielman corrects $\Omega(1/d_c)$-errors on a random $(d_v, d_c)$-LDPC code. A few years after, Feldman, Karger and Wainwright [FWK05, Fel03] introduced a decoding algorithm that is based on a simple linear programming (LP) relaxation that corrects $\Omega(1/d_c)$-errors on a random $(d_v, d_c)$-LDPC code.
However, the fraction of errors that is corrected by the Sipser-Spielman algorithm and the LP relaxation of [FWK05] (which is $O(1/d_c)$) can be much smaller than the best possible: in fact, [Gal62] (as well as [MB01]) showed that for a random $(d_v, d_c)$-LDPC code, the exponential-time nearest-neighbor Maximum Likelihood (ML) algorithm corrects close to $H_b^{-1}(d_v/d_c)$ probabilistic errors, which by Shannon’s channel coding theorem is the best possible.

Inspired by the Sherali-Adams hierarchy, Arora, Daskalakis and Steurer [ADS12] improved the best known fraction of correctable probabilistic errors by the LP decoder (which was previously achieved by Daskalakis et al. [DDKW08]) for some range of values of $d_v$ and $d_c$. Both Arora et al. [ADS12] and the original work of Feldman et al. [FWK05, Fel03] asked whether tightening the base LP using linear or semidefinite hierarchies can improve its performance, potentially approaching the information-theoretic limit. More precisely, in all previous work on LP decoding of error-correcting codes, the base LP decoder of Feldman et al. succeeds in the decoding task if and only if the transmitted codeword is the unique optimum of the relaxed polytope with the objective function being the (normalized) $l_1$ distance between the received vector and a point in the polytope. On the other hand, the decoder is considered to fail whenever there is an optimal non-integral vector.

In this paper, we prove the first lower bounds on the performance of the Sherali-Adams and Sum-of-Squares hierarchies when applied to the problem of decoding random $(d_v, d_c)$-LDPC codes.

**Theorem 3.10** (Lower bounds in the Sherali-Adams hierarchy). For any $d_v$ and $d_c \geq 5$, there exists $\eta > 0$ (depending on $d_c$) such that a random $(d_v, d_c)$-LDPC code satisfies the following with high probability: for any received vector, there is a fractional solution to the $\eta n$ rounds of the Sherali-Adams hierarchy of value $1/(d_c - 3)$ (for odd $d_c$) or $1/(d_c - 4)$ (for even $d_c$). Consequently, $\eta n$ rounds cannot decode more than a $\approx 1/d_c$ fraction of errors.

**Theorem 3.11** (Lower bounds in the Sum-of-Squares hierarchy). For any $d_v$ and $d_c = 3 \cdot 2^i + 3$ with $i \geq 1$, there exists $\eta > 0$ (depending on $d_c$) such that a random $(d_v, d_c)$-LDPC code satisfies the following with high probability: for any received vector, there is a fractional solution to the $\eta n$ rounds of the Sum-of-Squares hierarchy of value $3/(d_c - 3)$. Consequently, $\eta n$ rounds cannot decode more than a $\approx 3/d_c$ fraction of errors.

### 3.2.3 Graph Pricing

We consider the following natural problem for a seller with a profit-maximization objective. Let $I$ denote the indicator function.

**Graph Pricing**

**Input:** A graph $G = (V, E)$. For each edge $e$, its budget $b_e \in \mathbb{R}^+$.  

**Output:** Pricing $p : V \to (\mathbb{R}^+ \cup \{0\})$.  

**Goal:** Maximize $\sum_{(u,v) \in E} I(p(u) + p(v) \leq b_{(u,v)})(p(u) + p(v))$.  

This problem was proposed by Guruswami et al. [GHK+05], and has received much attention. The best known approximation algorithm for a general instance, which guarantees $\frac{1}{4}$ of the optimal solution, is given by Balcan and Blum [BB07] and Lee et al. [LBA+07]. The algorithm is simple enough to state here. First, assign 0 to each vertex with probability half independently. For each remaining vertex $v$, assign the price which maximizes the profit between $v$ and its neighbors already assigned 0. This simple algorithm has been neither improved nor proved to be optimal. Graph
Pricing is APX-hard [GHK+05], but the only strong hardness of approximation result rules out an approximation algorithm with a guarantee better than $\frac{1}{2}$ [KKMS09] under the Unique Games Conjecture (via reduction from Maximum Acyclic Subgraph).

The $\frac{1}{4}$-approximation algorithm is surprisingly simple and does not even rely on the power of a linear programming or semidefinite programming relaxation. The efforts to exploit the power of LP relaxations to find a better approximation algorithm have produced positive results for special classes of graphs. Krauthgamer et al. [KMR11] studied the case where all budgets are the same (but the graph might have a self-loop), and proposed a $\frac{1+\sqrt{2}}{3+\sqrt{2}} \approx 1.15$-approximation algorithm based on a LP relaxation. In general case, the standard LP is shown to have an integrality gap close to $\frac{1}{4}$ [KKMS09]. Therefore, it is natural to consider hierarchies of LP relaxations such as the Sherali-Adams hierarchy [SA90] (see [CT12] for a general survey and [GTW13, YZ14] for recent algorithmic results using the Sherali-Adams hierarchy). Especially, Chalermsook et al. [CKLN13] recently showed that there is a FPTAS when the graph has bounded treewidth, based on the Sherali-Adams hierarchy. However, the power of the Sherali-Adams hierarchy and SDP, as well as the inherent hardness of the problem, was not well-understood in general case.

We show that any polynomial time algorithm that guarantees a ratio better than $\frac{1}{4}$ must be powerful enough to refute the Unique Games Conjecture.

**Theorem 3.12.** Under the Unique Games Conjecture, for any $\epsilon > 0$, it is NP-hard to approximate Graph Pricing within a factor of $\frac{1}{4} + \epsilon$.

By the results of Khot and Vishnoi [KV05] and Raghavendra and Steurer [RS09] that convert a hardness under the UGC to a SDP gap instance, our result unconditionally shows that even a SDP-based algorithm will not improve the performance of a simple algorithm. For the Sherali-Adams hierarchy, we prove that even polynomial rounds of the Sherali-Adams hierarchy has an integrality gap close to $\frac{1}{4}$.

**Theorem 3.13.** Fix $\epsilon > 0$. There exists $\delta > 0$ such that the integrality gap of $n^\delta$-rounds of the Sherali-Adams hierarchy for Graph Pricing is at most $\frac{1}{4} + \epsilon$.

### 4 Hypergraph Coloring

Coloring (hyper)graphs is one of the most important and well-studied tasks in discrete mathematics and theoretical computer science. A $k$-uniform hypergraph $G = (V, E)$ is said to be $\chi$-colorable if there exists a coloring $c : V \mapsto \{1, \ldots, \chi\}$ such that no hyperedge is monochromatic, and such a coloring $c$ is referred to as a proper $\chi$-coloring. Coloring has been the focus of active research in both fields, and has served as the benchmark for new research paradigms such as the probabilistic method (Lovász local lemma [EL75]) and semidefinite programming (Lovász theta function [Lov79]).

Given a general $\chi$-colorable $k$-uniform hypergraph, the problem of reconstructing a $\chi$-coloring is known to be a hard task. Even assuming 2-colorability, reconstructing a proper 2-coloring is a classic NP-hard problem for $k \geq 3$. Given the intractability of proper 2-coloring, two notions of approximate coloring of 2-colorable hypergraphs have been studied in the literature of approximation algorithms. The first notion, called MIN COLORING, is to minimize the number of colors while still requiring that every hyperedge be non-monochromatic. The second notion, called MAX 2-COLORING allows only 2 colors, but the objective is to maximize the number of non-monochromatic
hyperedges.¹

Even with these relaxed objectives, the promise that the input hypergraph is 2-colorable seems grossly inadequate for polynomial time algorithms to exploit in a significant way. For MIN COLORING, given a 2-colorable k-uniform hypergraph, the best known algorithm uses \(O(n^{1-\frac{1}{k}})\) colors [CF96, AKMH96], which tends to the trivial upper bound \(n\) as \(k\) increases. On the other hand, [KS14] shows quasi-NP-hardness of \(2^{(\log n)^{\Omega(1)}}\)-coloring a 2-colorable hypergraphs (very recently the exponent was shown to approach \(\frac{1}{4}\) in [Hua15]).

The hardness results for MAX 2-COLORING show an even more pessimistic picture, wherein the naive random assignment (randomly give one of two colors to each vertex independently to leave a \((\frac{1}{2})^{k-1}\) fraction of hyperedges monochromatic in expectation), is shown to have the best guarantee for a polynomial time algorithm when \(k \geq 4\) (see [Hås01]).

Given these strong intractability results, it is natural to consider what further relaxations of the objectives could lead to efficient algorithms. This motivates our main question "how strong a promise on the input hypergraph is required for polynomial time algorithms to perform significantly better than naive algorithms for MIN COLORING and MAX 2-COLORING?"

There is a very strong promise on \(k\)-uniform hypergraphs which makes the task of proper 2-coloring easy. If a hypergraph is \(k\)-partite (i.e., there is a \(k\)-coloring such that each hyperedge has each color exactly once), then one can properly 2-color the hypergraph in polynomial time [Alo14, McD93]. The same algorithm can be generalized to hypergraphs which admit a \(c\)-balanced coloring (i.e., \(c\) divides \(k\) and there is a \(k\)-coloring such that each hyperedge has each color exactly \(\frac{k}{c}\) times).

The promises on structured colorings that we consider in this work are natural relaxations of the above strong promise of a perfectly balanced coloring.

- A hypergraph is said to have discrepancy \(\ell\) when there is a 2-coloring such that in each hyperedge, the difference between the number of vertices of each color is at most \(\ell\).
- A \(\chi\)-coloring \((\chi \leq k)\) is called rainbow if every hyperedge contains each color at least once.
- A \(\chi\)-coloring \((\chi \geq k)\) is called strong if every hyperedge contains \(k\) different colors.

These three notions are interesting in their own right, and have been independently studied. It is easy to see that \(\ell\)-discrepancy \((\ell < k)\), \(\chi\)-rainbow colorability \((2 \leq \chi \leq k)\), and \(\chi\)-strong colorability \((k \leq \chi \leq 2k - 2)\) all imply 2-colorability. For odd \(k\), both \((k+1)\)-strong colorability and \((k-1)\)-rainbow colorability imply discrepancy-1, so strong colorability and rainbow colorability seem stronger than low discrepancy.

## 4.1 Min Coloring

We prove the following strong hardness result.

**Theorem 4.1.** For any \(\epsilon > 0\) and \(Q, k \geq 2\), given a \(Qk\)-uniform hypergraph \(H = (V, E)\), it is NP-hard to distinguish between the following cases.

- **Completeness:** There is a \(k\)-coloring \(c : V \rightarrow [k]\) such that for every hyperedge \(e \in E\) and color \(i \in [k]\), \(c\) has at least \(Q - 1\) vertices of color \(i\).

¹The maximization version is also known as MAX-SET-SPLITTING, or more specifically MAX \(k\)-SET-SPLITTING when considering \(k\)-uniform hypergraphs, in the literature.
• **Soundness:** Every $I \subseteq V$ of measure $\epsilon$ induces at least a fraction $\epsilon^{O_Q,\epsilon(1)}$ of hyperedges. In particular, there is no independent set of measure $\epsilon$, and every $\lfloor \frac{1}{2} \rfloor$-coloring of $H$ induces a monochromatic hyperedge.

Fixing $Q = 2$ gives a hardness of rainbow coloring with $K$ optimized to be $2k$.

**Corollary 4.2.** For all integers $c,k \geq 2$, given a $2k$-uniform hypergraph $H$, it is NP-hard to distinguish whether $H$ is rainbow $k$-colorable or is not even $c$-colorable.

On the other hand, fixing $k = 2$ gives a strong hardness result of discrepancy minimization (with 2 colors).

**Corollary 4.3.** For any $c,Q \geq 2$, given a $2Q$-uniform hypergraph $H = (V,E)$, it is NP-hard to distinguish whether $H$ is 2-colorable with discrepancy 2 or is not even $c$-colorable.

The above result strengthens the result of Austrin et al [AGH14] that shows hardness of 2-coloring in the soundness case. However, their result also holds in $(2Q + 1)$-uniform hypergraphs with discrepancy 1, which is not covered by the results in this work. For algorithms, we prove that all three promises lead to an $\tilde{O}(n^{\frac{1}{k}})$-coloring that is decreasing in $k$.

**Theorem 4.4.** Consider any $k$-uniform hypergraph $H = (V,E)$ with $n$ vertices and $m$ edges. For any $0 < \ell < O(\sqrt{k})$, if $H$ has discrepancy-$\ell$, $(k - \ell)$-rainbow colorable, or $(k + \ell)$-strong colorable, one can color $H$ with $\tilde{O}(\frac{m}{n^{\frac{1}{k} - 2}}) = \tilde{O}(n^{\frac{1}{k}})$ colors.

These results significantly improve the current best algorithm that assumes only 2-colorability and uses $\tilde{O}(n^{1 - \frac{1}{k}})$ colors.

### 4.2 Max 2-Coloring

For **Max 2-Coloring**, we prove that our three promises, unlike mere 2-colorability, give enough structure for polynomial time algorithms to perform significantly better than naive algorithms. We also study these promises from a hardness perspective to understand the asymptotic threshold at which beating naive algorithms goes from easy to UG/NP-Hard. In particular assuming the UGC, for Max-2-Coloring under discrepancy $(k - \ell)$-colorability, this threshold is $\ell = \Theta(\sqrt{k})$.

**Theorem 4.5.** There is a randomized polynomial time algorithm that produces a 2-coloring of a $k$-uniform hypergraph $H$ with the following guarantee. For any $0 < \epsilon < \frac{1}{2}$ (let $\ell = k^\epsilon$), there exists a constant $\eta > 0$ such that if $H$ is $(k - \ell)$-rainbow colorable or $(k + \ell)$-strong colorable, the fraction of monochromatic edges in the produced 2-coloring is $O((\frac{1}{k})^nk)$ in expectation.

For the discrepancy case, we observe that when $\ell < \sqrt{k}$, our work for **Symmetric CSP** yields an approximation algorithm that marginally (by an additive factor much less than $2^{-k}$) outperforms the random assignment.

The following hardness results suggest that this gap between low-discrepancy and rainbow/strong colorability might be intrinsic.

**Theorem 4.6.** For sufficiently large odd $k$, given a $k$-uniform hypergraph which admits a 2-coloring with at most a $(\frac{1}{2})^{6k}$ fraction of edges of discrepancy larger than 1, it is UG-hard to find a 2-coloring with a $(\frac{1}{2})^{5k}$ fraction of monochromatic edges.
Theorem 4.7. For even \( k \geq 4 \), given a \( k \)-uniform hypergraph which admits a 2-coloring with no edge of discrepancy larger than 2, it is NP-hard to find a 2-coloring with a \( k^{-O(k)} \) fraction of monochromatic edges.

Theorem 4.8. For \( k \) sufficiently large, given a \( k \)-uniform hypergraph which admits a 2-coloring with no edge of discrepancy larger than \( O(\log k) \), it is NP-hard to find a 2-coloring with a \( 2^{-O(k)} \) fraction of monochromatic edges.

Theorem 4.9. For \( k \) such that \( \chi := k - \sqrt{k} \) is an integer greater than 1, and any \( \epsilon > 0 \), given a \( k \)-uniform hypergraph which admits a \( \chi \)-coloring with at most \( \epsilon \) fraction of non-rainbow edges, it is UG-hard to find a 2-coloring with a \( (\frac{1}{2})^{k-1} \) fraction of monochromatic edges.

Table 1 summarizes our results.

<table>
<thead>
<tr>
<th>Promises</th>
<th>( \ell )-Discrepancy</th>
<th>( (k - \ell) )-Rainbow</th>
<th>( (k + \ell) )-Strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max-2-Coloring Algorithm</td>
<td>( 1 - (1/2)^{k-1} + \delta, \quad \ell &lt; \sqrt{k} )</td>
<td>( 1 - (1/k)^{O(k)}, \quad \ell \ll \sqrt{k} )</td>
<td>( 1 - (1/k)^{O(k)}, \quad \ell \ll \sqrt{k} )</td>
</tr>
<tr>
<td>Max-2-Coloring Hardness</td>
<td>UG: ( 1 - (1/2)^k, \quad \ell = 1 )</td>
<td>UG: ( 1 - (1/2)^{k-1}, \quad \ell = \Omega(\sqrt{k}) )</td>
<td>UG: ( 1 - (1/2)^{k-1}, \quad \ell = \Omega(\sqrt{k}) )</td>
</tr>
<tr>
<td></td>
<td>NP: ( 1 - (1/k)^{O(k)}, \quad \ell = 2 )</td>
<td>NP: ( 1 - (1/2)^{O(k)}, \quad \ell = \Omega(\log k) )</td>
<td>NP: ( 1 - (1/2)^{k-1}, \quad \ell \geq \sqrt{k} )</td>
</tr>
<tr>
<td>Min-Coloring Algorithm</td>
<td>( n^{\ell/\log k}, \quad \ell = O(\sqrt{k}) )</td>
<td>( n^{\ell/\log k}, \quad \ell = O(\sqrt{k}) )</td>
<td>( n^{\ell/\log k}, \quad \ell = O(\sqrt{k}) )</td>
</tr>
<tr>
<td></td>
<td>NP: ( \omega(1), \quad \ell \geq 2 )</td>
<td>NP: ( \omega(1), \quad \ell \leq k/2 )</td>
<td>( n^{\ell/\log k}, \quad \ell = O(\sqrt{k}) )</td>
</tr>
</tbody>
</table>

Table 1: Summary of our algorithmic and hardness results with valid ranges of \( \ell \). The numbers of the first row indicate lower bounds on the fraction of non-monochromatic edges in a 2-coloring produced by our algorithms. \( \delta := \delta(k, \ell) > 0 \) is a small constant. The second row shows upper bounds on the fraction of non-monochromatic edges achieved by polynomial time algorithms. For the UG-hardness results, note that the input hypergraph does not have all edges satisfying the promises but almost edges satisfying them. The third row shows the upper bound upto log factors, on the number of colors one can use to properly 2-color the graph.

5 Covering and Cut

5.1 \( H \)-Transversal and Graph Partitioning

Consider the following two classic problems.

**Minimum Set Transversal**

Input: A universe \( U \) and a collection of subsets \( S_1, \ldots, S_m \).

Output: \( F \subseteq U \) such that \( F \) intersects every \( S_i \).

Goal: Minimize \( |F| \).

This problem is equivalent to Minimum Set Cover by taking the dual set system.
MAXIMUM SET PACKING

Input: A universe \( U \) and a collection of subsets \( S_1, \ldots, S_m \).

Output: A subcollection \( S_{i_1}, \ldots, S_{i_m}' \) which are pairwise disjoint.

Goal: Maximize \( m' \).

Given the same input, it is clear that the optimum of the former is always at least that of the latter (i.e., weak duality holds). Studying the (approximate) reverse direction of the inequality (i.e., strong duality) as well as the complexity of both problems for many interesting classes of set systems is arguably the most studied paradigm in combinatorial optimization.

We focus on set systems where the size of each set is bounded by a constant \( k \). With this restriction, MINIMUM SET TRANSVERSAL and MAXIMUM SET PACKING are known as \( k \)-HYPERGRAPH VERTEX COVER (\( k \)-HVC) and \( k \)-SET PACKING (\( k \)-SP), respectively. This assumption significantly simplifies the problem since there are at most \( n^k \) sets. While there is a simple factor \( k \)-approximation algorithm for both problems, it is NP-hard to approximate \( k \)-HVC and \( k \)-SP within a factor less than \( k - 1 \) [DGKR05] and \( O\left(\frac{k}{\log k}\right) \) [HSS06] respectively.

5.1.1 \( H \)-Transversal / Packing

We study the following special cases of \( k \)-HYPERGRAPH VERTEX COVER and \( k \)-SET PACKING. Let \( H \) be a fixed graph with \( k \) vertices.

\( H \)-TRANSVERSAL

Input: A graph \( G = (V, E) \)

Output: \( F \subseteq V \) such that the induced subgraph \( G|_{V \setminus F} \) does not have \( H \) as a subgraph.

Goal: Minimize |\( F \)|.

\( H \)-PACKING

Input: A graph \( G = (V, E) \)

Output: Disjoint subsets \( S_1, \ldots, S_m \subseteq V \) where for each \( i \), |\( S_i \)| = \( k \) and \( G|_{S_i} \) has \( H \) as a subgraph.

Goal: Maximize \( m \).

These problems capture MINIMUM VERTEX COVER and MAXIMUM MATCHING as special cases when \( H \) is a single edge. Other special cases where \( H \) is a clique or a cycle have been also actively studied. In this thesis, we study approximabilities of \( H \)-TRANSVERSAL and \( H \)-PACKING for every fixed graph \( H \). They admit a simple \( k \)-approximation algorithm as special cases of \( k \)-HYPERGRAPH VERTEX COVER and \( k \)-SET PACKING. We study whether significantly better approximation algorithms (i.e., \( k^\delta \)-approximation for some \( \delta < 1 \)) exist. Our main hardness result is the following.

**Theorem 5.1.** If \( H \) is a 2-vertex connected with \( k \) vertices, unless \( \text{NP} \subseteq \text{BPP} \), no polynomial time algorithm approximates \( H \)-TRANSVERSAL within a factor better than \( k - 1 \), and \( H \)-PACKING within a factor better than \( \Omega\left(\frac{k}{\log k}\right) \).
This result leaves us to study 1-connected graphs. In particular, we focus on $k$-Star and $k$-Path, where $k$-Star denote $K_{1,k-1}$, the complete bipartite graph with 1 and $k-1$ vertices on each side, and $k$-Path is a simple path with $k$ vertices. It is easy to see that $k$-STAR PACKING is as hard to approximate as MAXIMUM INDEPENDENT SET on $(k-1)$-regular graphs, which is NP-hard to approximate within a factor $\Omega\left(\frac{k}{\log^2 k}\right)$ [Cha13]. We show that both $k$-STAR TRANSVERSAL and $k$-PATH TRANSVERSAL admit a good approximation algorithm.

**Theorem 5.2.** $k$-STAR TRANSVERSAL can be approximated within a factor of $O(\log k)$ in polynomial time.

**Theorem 5.3.** There is an $O(\log k)$-approximation algorithm for $k$-PATH TRANSVERSAL that runs in time $2^{O(k^3 \log k)} n^{O(1)}$.

Note that the exponential dependence of the running time on $k$ is necessary since finding a $k$-Path for general $k$ is NP-hard.

### 5.1.2 Partitioning Graphs into Small Pieces

We study the following natural graph partitioning problems.

**$k$-Vertex Separator**
- **Input:** An undirected graph $G = (V,E)$ and $k \in \mathbb{N}$.
- **Output:** A subset $S \subseteq V$ such that in the subgraph induced by $V \setminus S$ (denoted by $G|_{V \setminus S}$), each connected component has at most $k$ vertices.
- **Goal:** Minimize $|S|$.

$k$-Vertex Separator is a special case of $(k+1)$-HVC and similar to $H$-TRANSVERSAL in the sense that the goal is to remove the minimum number of vertices such that $G$ has no connected graph with $k+1$ vertices as a subgraph ($H$ is replaced by a family of connected graphs with $k+1$ vertices). The edge version can be defined similarly.

**$k$-Edge Separator**
- **Input:** An undirected graph $G = (V,E)$ and $k \in \mathbb{N}$.
- **Output:** A subset $S \subseteq E$ such that in the subgraph $(V,E \setminus S)$, each connected component has at most $k$ vertices.
- **Goal:** Minimize $|S|$.

Our primary focus is on the case where $k$ is either a constant or a slowly growing function of $n$ (e.g. $O(\log n)$ or $n^{\alpha(1)}$). Our problems can be interpreted as a special case of three general classes of problems that have been studied separately (balanced graph partitioning, $k$-HYPERGRAPH VERTEX COVER, and fixed parameter tractability (FPT)).

Our main result is the following algorithm for $k$-VERTEX SEPARATOR. For fixed constants $b, c > 1$, an algorithm for $k$-VERTEX SEPARATOR is called an $(b,c)$-bicriteria approximation algorithm if given an instance $G = (V,E)$ and $k \in \mathbb{N}$, it outputs $S \subseteq V$ such that (1) each connected component of $G|_{S \setminus V}$ has at most $bk$ vertices and (2) $|S|$ is at most $c$ times the optimum of $k$-VERTEX SEPARATOR.
Theorem 5.4. For any $\epsilon \in (0, 1/2)$, there is a polynomial time $(\frac{1}{1-2\epsilon}, O(\log k))-bicriteria approximation algorithm for $k$-Vertex Separator.

Setting $\epsilon = \frac{1}{4}$ and running the algorithm yields $S \subseteq V$ with $|S| \leq O(\log k) \cdot \text{OPT}$ such that each component in $G[V \setminus S]$ has at most 2$k$ vertices. Performing an exhaustive search in each connected component yields the following true approximation algorithm whose running time depends exponentially only on $k$.

**Corollary 5.5.** There is an $O(\log k)$-approximation algorithm for $k$-Vertex Separator that runs in time $n^{O(1)} + 2^{O(k)}n$.

This gives a FPT approximation algorithm when parameterized by only $k$, and its approximation ratio $O(\log k)$ improves the simple $(k + 1)$-approximation from $k$-Hypergraph Vertex Cover. When $\text{OPT} \gg k$, it runs even faster than the time lower bound $k^{\Omega(\text{OPT})}n^{\Omega(1)}$ for the exact algorithm assuming the Exponential Time Hypothesis [DDvH14].

The natural question is whether superpolynomial dependence on $k$ is necessary to achieve true $O(\log k)$-approximation. The following theorem proves hardness of $k$-Vertex Separator based on Densest $k$-Subgraph. In particular, a polynomial time $O(\log k)$-approximation algorithm for $k$-Vertex Separator will imply $O(\log^2 n)$-approximation algorithm for Densest $k$-Subgraph. Given that the best approximation algorithm achieves $\approx O(n^{1/4})$-approximation [BCC+10] and $n^{\Omega(1)}$-rounds of the Sum-of-Squares hierarchy have a gap at least $n^{\Omega(1)}$ [BCV+12], such a result seems unlikely or will be considered as a breakthrough.

**Theorem 5.6.** If there is a polynomial time $f$-approximation algorithm for $k$-Vertex Separator, then there is a polynomial time $2f^2$-approximation algorithm for Densest $k$-Subgraph.

For $k$-Edge Separator, we prove that the true $O(\log k)$-approximation can be achieved in polynomial time. This shows a stark difference between the vertex version and the edge version.

**Theorem 5.7.** There is an $O(\log k)$-approximation algorithm for $k$-Edge Separator that runs in time $n^{O(1)}$.

When $k = n^{o(1)}$, our algorithm outperforms the previous best approximation algorithm [KNS09, ENRS99].

5.2 Cut and Interdiction

We study variants of the classical $s$-$t$ Min Cut problem in both directed and undirected graphs that have been actively studied and prove the optimal hardness or the first super-constant hardness for them. All our results are based on the general framework of converting an integrality gap instance to a length-control dictatorship test. The structure of our length-control dictatorship tests allows us to naturally convert an integrality gap instance for the basic LP for various cut problems to hardness based on the UGC. While these problems have slightly different characteristics that make it hard to present the single result for a wide class of problems like CSPs, we hope that our techniques may be useful to prove hardness of other cut problems.

5.2.1 Multicut

Given a directed graph and two vertices $s$ and $t$, one of the most natural variants of $s$-$t$ Min Cut is to remove the fewest edges to ensure that there is no directed path from $s$ to $t$ and no directed path
from $t$ to $s$. This problem is known as \textit{s-t Bicut} and admits the trivial 2-approximation algorithm by computing the minimum s-t cut and t-s cut.

**Directed Multiway Cut** is a generalization of \textit{s-t Bicut} that has been actively studied. Given a directed graph with $k$ terminals $s_1, \ldots, s_k$, the goal is to remove the fewest number of edges such that there is no path from $s_i$ to $s_j$ for any $i \neq j$. \textit{Directed Multiway Cut} also admits 2-approximation [NZ01, CM16]. If $k$ is allowed to increase polynomially with $n$, there is a simple reduction from Minimum Vertex Cover that shows $(2 - \epsilon)$-approximation is hard under the UGC [GVY94, KR08].

\textit{Directed Multiway Cut} can be further generalized to \textit{Directed Multicut}. Given a directed graph with $k$ source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$, the goal is to remove the fewest number of edges such that there is no path from $s_i$ to $t_i$ for any $i$. Computing the minimum $s_i$-$t_i$ cut for all $i$ separately gives the trivial $k$-approximation algorithm. Chuzhoy and Kharanna [CK09] showed \textit{Directed Multicut} is hard to approximate within a factor $2^{\Omega(\log^{1-\epsilon} n)} = 2^{\Omega(\log^{1-\epsilon} k)}$ when $k$ is polynomially growing with $n$. Agarwal et al. [AAC07] showed $\tilde{O}(n^{\frac{11}{23}})$-approximation algorithm, which improves the trivial $k$-approximation when $k$ is large.

Very recently, Chekuri and Madan [CM16] showed a simple approximation-preserving reductions from \textit{Directed Multicut} with $k = 2$ to \textit{s-t Bicut} (the other direction is trivially true), and (Undirected) \textit{Node-weighted Multiway Cut} with $k = 4$ to \textit{s-t Bicut}. Since Node-weighted \textit{Multiway Cut} with $k = 4$ is hard to approximate within a factor $1.5 - \epsilon$ under the UGC [EVW13] (matching the algorithm of Garg et al. [GVY94]), the same hardness holds for \textit{s-t Bicut}, \textit{Directed Multiway Cut}, and \textit{Directed Multicut} for constant $k$. To the best of our knowledge, $1.5 - \epsilon$ is the best hardness factor for constant $k$ even assuming the UGC. In the same paper, Chekuri and Madan [CM16] asked whether a factor $2 - \epsilon$ hardness holds for \textit{s-t Bicut} under the UGC.

We prove that for any constant $k \geq 2$, the trivial $k$-approximation for \textit{Directed Multicut} might be optimal. Our result for $k = 2$ gives the optimal hardness result for \textit{s-t Bicut}, answering the question of Chekuri and Madan.

**Theorem 5.8.** Assuming the Unique Games Conjecture, for every constant $k \geq 2$ and $\epsilon > 0$, \textit{Directed Multicut} with $k$ source-sink pairs is NP-hard to approximate within a factor $k - \epsilon$.

**Corollary 5.9.** Assuming the Unique Games Conjecture, for any $\epsilon > 0$, \textit{s-t Bicut} is hard to approximate within a factor $2 - \epsilon$.

### 5.2.2 Length-bounded Cut / Shortest Path Interdiction

Another natural variant of \textit{s-t Min Cut} is the \textit{Length-Bounded Cut} problem, where given an integer $l$, we only want to cut \textit{s-t} paths of length strictly less than $l$.\footnote{It is more conventional to cut \textit{s-t} paths of length \textit{at most} $l$. We use this slightly nonconventional way to be more consistent with \textit{Shortest Path Interdiction}.} Its practical motivation is based on the fact that in most communication/transportation networks, short paths are preferred to be used to long paths [MM10].

Lovász et al. [LNP78] gave an exact algorithm for \textit{Length-Bounded Vertex Cut} ($l \leq 5$) in undirected graphs. Mahjoub and McCormick [MM10] proved that \textit{Length-Bounded Vertex Cut} admits an exact polynomial time algorithm for $l \leq 4$ in undirected graphs. Baier et al. [BEH10] showed that both \textit{Length-Bounded Vertex Cut} ($l > 5$) and \textit{Length-Bounded Edge Cut}($l > 4$) are NP-hard to approximate within a factor $1.1377$. They presented $O(\min(l, \frac{n}{k})) =$
$O(\sqrt{n})$-approximation algorithm for LENGTH-BOUNDED VERTEX CUT and $O(\min(l, n^{2/3}, \sqrt{m})) = O(n^{2/3})$-approximation algorithm for LENGTH-BOUNDED EDGE CUT, with matching LP gaps. LENGTH-BOUNDED CUT problems have been also actively studied in terms of their fixed parameter tractability [GT11, DK15, BNN15, FHHN15].

If we exchange the roles of the objective $k$ and the length bound $l$, the problem becomes SHORTEST PATH INTERDICATION, where we want to maximize the length of the shortest $s$-$t$ path after removing at most $k$ vertices or edges. It is also one of the central problems in a broader class of interdiction problems. The study of SHORTEST PATH INTERDICATION started in 1980’s when the problem was called as the $k$-most-vital-arcs problem [CD82, MMG89, BGV89] and proved to be NP-hard [BGV89]. Khachiyan et al. [KBB+07] proved that it is NP-hard to approximate within a factor less than 2. While many heuristic algorithms were proposed [IW02, BB08, Mor11] and hardness in planar graphs [PS13] was shown, whether the general version admits a constant factor approximation was still unknown.

If we exchange the roles of the objective $k$ and the length bound $l$, the problem becomes SHORTEST PATH INTERDICATION, where we want to maximize the length of the shortest $s$-$t$ path after removing at most $k$ vertices or edges. It is also one of the central problems in a broader class of interdiction problems. The study of SHORTEST PATH INTERDICATION started in 1980’s when the problem was called as the $k$-most-vital-arcs problem [CD82, MMG89, BGV89] and proved to be NP-hard [BGV89]. Khachiyan et al. [KBB+07] proved that it is NP-hard to approximate within a factor less than 2. While many heuristic algorithms were proposed [IW02, BB08, Mor11] and hardness in planar graphs [PS13] was shown, whether the general version admits a constant factor approximation was still unknown.

Given a graph $G = (V, E)$ and $s, t \in V$, let $\text{dist}(G)$ be the length of the shortest $s$-$t$ path. For $V' \subseteq V$, let $G[V']$ be the subgraph induced by $V'$. For $E' \subseteq E$, we use the same notation $G \setminus E'$ to denote the subgraph $(V, E \setminus E')$. We primarily study undirected graphs. We first present our results for the vertex version of both problems (collectively called as SHORT PATH VERTEX CUT onwards).

**Theorem 5.10.** Assume the Unique Games Conjecture. For infinitely many values of the constant $\ell \in \mathbb{N}$, given an undirected graph $G = (V, E)$ and $s, t \in V$ where there exists $C^* \subseteq V \setminus \{s, t\}$ such that $\text{dist}(G \setminus C^*) \geq \ell$, it is NP-hard to perform any of the following tasks.

1. Find $C \subseteq V \setminus \{s, t\}$ such that $|C| \leq \Omega(\ell) \cdot |C^*|$ and $\text{dist}(G \setminus C) \geq \ell$.
2. Find $C \subseteq V \setminus \{s, t\}$ such that $|C| \leq |C^*|$ and $\text{dist}(G \setminus C) \geq O(\sqrt{\ell})$.
3. Find $C \subseteq V \setminus \{s, t\}$ such that $|C| \leq \Omega(\ell^{2\epsilon}) \cdot |C^*|$ and $\text{dist}(G \setminus C) \geq O(\ell^{1+2\epsilon})$ for some $0 < \epsilon < 1$.

The first result shows that LENGTH-BOUNDED VERTEX CUT is hard to approximate within a factor $\Omega(\ell)$. This matches the best $\frac{1}{2}$-approximation up to a constant. [BEH+10]. The second result shows that SHORTEST PATH VERTEX INTERDICATION is hard to approximate with in a factor $\Omega(\sqrt{\text{OPT}})$, and the third result rules out bicriteria approximation — for any constant $c$, it is hard to approximate both $\ell$ and $|C^*|$ within a factor of $c$.

The above results hold for directed graphs by definition. Our hard instances will have a natural layered structure, so it can be easily checked that the same results (up to a constant) hold for directed acyclic graphs. Since one vertex can be split as one directed edge, the same results hold for the edge version in directed acyclic graphs.

For LENGTH-BOUNDED EDGE CUT and SHORTEST PATH EDGE INTERDICATION in undirected graphs (collectively called SHORT PATH EDGE CUT onwards), we prove the following theorems.

**Theorem 5.11.** Assume the Unique Games Conjecture. For infinitely many values of the constant $\ell \in \mathbb{N}$, given an undirected graph $G = (V, E)$ and $s, t \in V$ where there exists $C^* \subseteq E$ such that $\text{dist}(V \setminus C^*) \geq \ell$, it is NP-hard to perform any of the following tasks.

1. Find $C \subseteq E$ such that $|C| \leq \Omega(\sqrt{\ell}) \cdot |C^*|$ and $\text{dist}(G \setminus C) \geq \ell$.
2. Find $C \subseteq E$ such that $|C| \leq |C^*|$ and $\text{dist}(G \setminus C) \geq \ell^{1/2}$. 

18
3. Find $C \subseteq E$ such that $|C| \leq \Omega(l^{\frac{3}{4}}) \cdot |C^*|$ and $\text{dist}(G \setminus C) \geq O(l^{\frac{2+2\epsilon}{2}})$ for some $0 < \epsilon < \frac{1}{2}$.

Our hardness factors for the edge versions, $\Omega(\sqrt{l})$ for Length-Bounded Edge Cut and $\Omega(\sqrt[3]{\text{OPT}})$ for Shortest Path Edge Interdiction, are slightly weaker than those for their vertex counterparts, but we are not aware of any approximation algorithm specialized for the edge versions. It is an interesting open problem whether there exist better approximation algorithms for the edge versions.

5.2.3 Firefighter

Resource Minimization for Fire Containment (RMFC) is a problem closely related to Length-Bounded Cut with the additional notion of time. Given a graph $G$, a vertex $s$, and a subset $T$ of vertices, consider the situation where fire starts at $s$ on Day 0. For each Day $i$ ($i \geq 1$), we can save at most $k$ vertices, and the fire spreads from currently burning vertices to its unsaved neighbors. Once a vertex is burning or saved, it remains so from then onwards. The process is terminated when the fire cannot spread anymore. RMFC asks to find a strategy to save $k$ vertices each day with the minimum $k$ so that no vertex in $T$ is burnt. These problems model the spread of epidemics or ideas through a social network, and have been actively studied recently [CC10, ACHS12, ABZ16, CV16].

RMFC, along with other variants, is first introduced by Hartnell [Har95]. Another well-studied variant is called the Firefighter problem, where we are only given $s \in V$ and want to maximize the number of vertices that are not burnt at the end. It is known to be NP-hard to approximate within a factor $n^{1-\epsilon}$ for any $\epsilon > 0$ [ACHS12]. King and MacGillivray [KM10] proved that RMFC is hard to approximate within a factor less than 2. Anshelevich et al. [ACHS12] presented an $O(\sqrt{n})$-approximation algorithm for general graphs, and Chalermsook and Chuzhoy [CC10] showed that RMFC admits $O(\log^* n)$-approximation in trees. Very recently, the approximation ratio in trees has been improved to $O(1)$ [ABZ16]. Both Anshelevich et al. [ACHS12] and Chalermsook and Chuzhoy [CC10] independently studied directed layer graphs with $b$ layers, showing $O(\log b)$-approximation.

Our final result on RMFC assumes a variant of the Unique Games Conjecture which is not known to be equivalent to the original UGC. Given a bipartite graph as an instance of Unique Games, it states that in the completeness case, all constraints incident on $(1-\epsilon)$ fraction of vertices in one side are satisfied, and in the soundness case, in addition to having a low value, every $\frac{1}{n}$ fraction of vertices on one side have at least a $\frac{9}{10}$ fraction of vertices on the other side as neighbors. Our conjecture is implied by the conjecture of Bansal and Khot [BK09] that is used to prove the hardness of Minimizing Weighted Completion Time with Precedence Constraints and requires a more strict expansion condition. See Section 7 of [Lee16] for the exact statement.

**Theorem 5.12.** Assuming Conjecture 7.5 of [Lee16], it is NP-hard to approximate RMFC in undirected graphs within any constant factor.

Again, our reduction has a natural layered structure and the result holds for directed layered graphs. With $b$ layers, we prove that it is hard to approximate with in a factor $\Omega(\log b)$, matching the best approximation algorithms [CC10, ACHS12].
6 Future Directions

6.1 Ongoing Projects

6.1.1 Fixed Parameter Tractability of Edge $k$-Cut

Consider the following classic problem.

**EDGE $k$-CUT**

Input: A graph $G = (V,E)$.

Output: $F \subseteq E$ such that the subgraph $(V,E \setminus F)$ has at least $k$ connected components.

Goal: Minimize $|F|$.

When $k$ is a constant, this problem admits $n^{O(k)}$-time algorithm that computes an optimal $F$ [GH94, KS96]. It also admits a 2-approximation algorithm [GBH00] whose running time is $n^{O(1)}$ even for large values of $k$. It is an interesting open question whether computing an optimal solution or significantly better approximation is fixed parameter tractable — achieved by an algorithm that runs in time $f(k) \cdot n^{O(1)}$ where $f$ is some computable function depending only on $k$.

In an ongoing project with Venkata Guruswami and Alfred Go, we proved that it is $\text{W}[1]$-hard to compute an optimal solution, implying that it is unlikely to have an FPT algorithm. We are currently investigating whether significantly better approximation can be achieved in FPT time.

6.1.2 Blocking Arborescences

We consider the following problem. Given a directed graph $D = (V,A)$ with $|V| = n$, recall that $F \subseteq A$ is called an arborescence if $|F| = n - 1$ and there exists a vertex $r \in V$ such that every $v \in V$ is reachable from $r$ using arcs in $F$.

**BLOCKING ARBORESCENCES**

Input: A directed graph $D = (V,A)$.

Output: $S \subseteq V$ such that the induced subgraph $D|_{V \setminus S}$ has no arborescence.

Goal: Minimize $|S|$.

In an ongoing project with Karthik Chandrasekaran, Tamasz Kiraly, Kristof Berczi and Chao Xu, we proved that this problem is NP-hard to approximate within a factor $1.5 - \epsilon$ under the UGC. This is proved via previously introduced *length-control dictatorship tests*. This problem also has a simple 2-approximation algorithm. We are currently working to close the gap between 1.5 and 2.

6.2 Open Problems from Previous Works

6.2.1 Symmetric CSPs

Conjecture 3.5 is the only conjecture in the thesis. Proving the conjecture will give a simple and complete characterization of approximation resistance of symmetric CSPs without negation. Even though we did not conjecture the analogous conjecture for symmetric CSPs with negation, it would be still interesting to study approximation resistance of symmetric CSPs with negation.
6.2.2 Shortest Path Interdiction

Our hardness result Theorem 5.10 is tight for LENGTH-BOUNDED VERTEX CUT up to a constant, but we are not aware of any approximation algorithm that matches the result of Theorem 5.10 for SHORTEST PATH VERTEX INTERDICATION. It would be interesting to see whether SHORTEST PATH VERTEX INTERDICATION admits an \(O(\sqrt{OPT})\)-approximation algorithm.

Our results for LENGTH-BOUNDED EDGE CUT and SHORTEST PATH EDGE INTERDICATION in undirected graphs are asymptotically weaker than their vertex counterparts. It is open whether the edges versions admit significantly better approximation, or Theorem 5.11 can be improved.

6.2.3 \(k\)-Path Packing and \(H\)-Edge Transversal/Packing

Theorem 5.1 shows that \(H\)-TRANSVERSAL and \(H\)-PACKING do not admit a good approximation algorithm when \(H\) is 2-connected. This thesis focuses on \(k\)-Star and \(k\)-Path as representatives of 1-connected graphs. We proved that \(k\)-STAR TRANSVERSAL and \(k\)-Path TRANSVERSAL admit an \(O(\log k)\)-approximation algorithm, while \(k\)-STAR PACKING is not likely to admit \(k^{\delta}\)-approximation for any \(\delta < 1\). The only remaining problem is \(k\)-PATH PACKING.

It would be also interesting to study the edge deletion versions of \(H\)-TRANSVERSAL and \(H\)-PACKING, where the underlying universe is the set of edges instead of the set of vertices. The same techniques as Theorem 5.1 show that they are unlikely to have \(k^{\delta}\)-approximation for any \(\delta < 1\) when \(H\) is 2-connected, but the simple greedy algorithm only guarantees \(O(k^2)\)-approximation. For example, \(K_r\)-EDGE TRANSVERSAL, where we want to remove the number of edges to make a graph \(K_r\)-free, is a natural problem related to the classic Turán’s theorem. Its approximation ratio is between \(\Omega(r)\) and \(\binom{r}{2}\).

6.3 Long-term Directions

6.3.1 NP-hardness

The following is the list of long-standing problems that we want to address. Here the goal is to prove NP-hardness because either the corresponding UG-hardness is already known or the UGC is not applicable due to the technical nature of problems.

- **Max \(k\)-CSP with perfect completeness:** By the result of Chan [Cha13], it is known that Max \(k\)-CSP is NP-hard to approximate within a factor better than \(\Theta(\frac{1}{\sqrt{k}})\), matching the current best algorithm of Charikar et al. [CMM07], and the previous UG-hardness result of Samorodnitsky and Trevisan [ST09, AM09]. However, when the instance is promised to admit an assignment that satisfies every constraint (also known as perfect completeness), the best algorithm still achieves \(\Omega(\frac{k}{\sqrt{k}})\)-approximation while the best hardness remains at \(\tilde{O}(\sqrt[4]{\frac{2^k}{e^2}})\) [Hua13].

Huang’s hardness result crucially depends on the direct sum technique of Chan [Cha13], which can be viewed as taking a polymorphism of the predicate. While the existence of a nontrivial polymorphism makes it easy to compute a satisfying assignment for satisfiable instances, Huang bypassed it by allowing one error bit per each sum. We hope that tighter combination of a polymorphism and an error bit may lead to tighter inapproximability.

- **Max Horn-SAT:** Guruswami and Zhou [GZ12] proved that given an \((1 - \epsilon)\)-satisfiable instance of Max Horn-3-SAT, it is UG-hard to find an assignment satisfying more than
\[ \left(1 - \frac{1}{O(\log(1/\epsilon))}\right) \] of the constraints. Proving the same hardness without relying on the UGC will be interesting.

Guruswami and Zhou’s result is achieved by constructing a SDP gap instance and applying Raghavendra’s result [Rag08] to convert it to UG-hardness. We hope that a more direct and combinatorial reduction from LABEL COVER may bypass the dependence on the UGC. This approach was successful on UNIQUE COVERAGE, which was another result of Guruswami and Zhou that we converted to NP-hardness.

- **Feedback Vertex Set**: Under the UGC, there are two different proofs showing that Feedback Vertex Set does not admit a constant factor approximation algorithm. The first one is given by Guruswami et al. [GMR08] based on the tools for MAX CSP, and the other is given by Svensson [Sve13]. This thesis contains the further simplification of Svensson’s proof that inspires length-control dictatorship tests for other problems. We believe that proof of the same statement without the UGC will reveal many applications beyond Feedback Vertex Set.

One possible approach is to try a direct reduction from \( k \)-Hypergraph Vertex Cover. Our inapproximability result on \( H \)-Transversal, when \( H \) is a cycle of length \( k \), indeed shows that hitting cycles of length \( o(\log n) \) is hard to approximate. While cycles of length around \( \Theta(\log n) \) remains as a bottleneck, we hope that another idea may lead to strong NP-hardness of Feedback Vertex Set.

### 6.3.2 Problems with Wider Gaps

The following problems are outstanding open problems where the best approximation ratio and the hardness ratio are far apart. For these problems, tight hardness even assuming the UGC is not known.

- **Min-Max and Max-Min Allocation**: MINMAX ALLOCATION and MAXMIN ALLOCATION are also fundamental optimization problems that have resisted attempts to understand their approximability. MAXMIN ALLOCATION admit an \( O(n^\epsilon) \)-approximation algorithm while the best hardness remains at 2 [BCG09, CCK09]. MINMAX ALLOCATION is also known as SCHEDULING UNRELATED PARALLEL MACHINES, and the optimal approximation ratio is between 1.5 and 2 [LST90]. The best hardness results are achieved via a simple reduction from 3-SAT. It would be interesting to see the modern theory of hardness of approximation is applicable to these problems.

- **\( k \)-Set Packing and disjoint path problems**: Our results for cut and interdiction problems are based on the solid understanding of \( k \)-HYPERGRAPH VERTEX COVER, which is UG-hard to approximate within a factor \( k - \epsilon \) and NP-hard to approximate within a factor \( k - 1 - \epsilon \). \( k \)-SET PACKING has a larger gap between algorithms and hardness, where the best algorithm achieves \( k^{1+\epsilon} \) [Cyg13] and the best hardness remains at \( \Omega\left(\frac{k}{\log^k}\right) \) [HSS06]. MAXIMUM INDEPENDENT SET ON \( k \)-REGULAR GRAPHS is an important special case that is also hard to approximate within a factor \( \Omega\left(\frac{k}{\log^k}\right) \) [AKS09, Cha13]. These hardness results use different techniques.

We believe that more unified understanding of \( k \)-SET PACKING and its special cases will lead to tighter results for \( k \)-SET PACKING itself as well as its important variants.

One special case is the following natural disjoint paths problems: Given an undirected graph \( G \) and two pairs of terminals \( (s_1, t_1), (s_2, t_2) \), how can we find one path from \( s_1 \) to \( t_1 \) and as
many paths from $s_2$ to $t_2$, where all paths must be internally vertex disjoint? What if we want to find the same number of $s_1$-$t_1$ paths and $s_2$-$t_2$ paths?

References


24


