A Study on Generalized Solution Concepts in Constraint Satisfaction and Graph Colouring

by

Peng Zhang

B.Eng, East China Normal University, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Interdisciplinary Studies)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)
July 2014
© Peng Zhang, 2014
Abstract

The concept of super solutions plays a crucial role in using the constraint satisfaction framework to model many AI problems under uncertain, dynamic, or interactive environments. The availability of large-scale, dynamic, uncertain, and networked data sources in a variety of application domains provides a challenge and opportunity for the constraint programming community, and we expect that super solutions will continue to attract a great deal of interest. In the first part of this thesis, we study the probabilistic behaviour of super solutions of random instances of Boolean Satisfiability (SAT) and Constraint Satisfaction Problems (CSPs). Our analysis focuses on a special type of super solutions, the (1,0)-super solutions. For random $k$-SAT, we establish the exact threshold of the phase transition of the solution probability for the cases of $k = 2$ and 3, and we upper and lower bound the threshold of the phase transition for the case of $k \geq 4$. For random CSPs, we derive a non-trivial upper bound on the threshold of phase transitions.

Graph colouring is one of the most well-studied problems in graph theory. A solution to a graph colouring problem is a colouring of the vertices such that each colour class is a stable set. A relatively new generalization of graph colouring is cograph colouring, where each colour class is a cograph. Cographs are the minimum family of graphs containing a single vertex and are closed under complementation and disjoint union. We define the cog-chromatic number of a graph $G$ as the minimum number of colours needed by a cograph colouring of $G$. Several problems related to cograph colouring are studied in the second part of this thesis, including properties of graphs that have cog-chromatic number 2; computational hardness of deciding and approximating the cog-chromatic number of graphs; and graphs that are critical in terms of cog-chromatic numbers. Several interesting constructions of graphs with extremal properties with respect to cograph colouring are also presented.
Table of Contents

Abstract ......................................................... ii
Table of Contents .................................................. iii
List of Tables ....................................................... v
List of Figures ....................................................... vi
Acknowledgements .................................................. vii

Chapter 1: Introduction .......................................... 1
  1.1 Threshold phenomena of super solutions in CSPs ........... 1
  1.2 Partitioning graphs into cographs .......................... 3
  1.3 Overview ................................................... 4

Chapter 2: Super solutions for random $k$-SAT and CSPs ... 6
  2.1 $(1,0)$-super solutions for $k$-SAT ........................ 6
  2.2 Random models of $k$-SAT ................................. 7
  2.3 The second moment method ................................. 8
  2.4 The threshold for $(1,0)$-2-SAT ............................ 10
  2.5 The threshold for $(1,0)$-3-SAT ............................ 11
    2.5.1 Lower bound on the threshold for $(1,0)$-3-SAT ....... 11
    2.5.2 Upper bound on the threshold for $(1,0)$-3-SAT ...... 12
  2.6 The threshold for $(1,0)$-$k$-SAT ........................ 15
  2.7 Designing SAT benchmarks by projections ................. 19
  2.8 Super solutions for random binary CSPs .................. 21

Chapter 3: Partitioning of graphs into cographs ............. 25
  3.1 Graphs and graph colouring ............................... 25
  3.2 Cographs and the cog-chromatic number ................. 27
  3.3 Approximating the cog-chromatic number ................. 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1</td>
<td>Greedy colouring</td>
<td>31</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Maximum induced cograph</td>
<td>33</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Approximation algorithms</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Minimum order of $k$-cog-chromatic graphs</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Cog-critical graphs</td>
<td>40</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Properties of cog-critical graphs</td>
<td>40</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Arbitrarily large $k$-cog-critical graphs</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>Chapter 4: Conclusion</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>53</td>
</tr>
</tbody>
</table>
List of Tables

Table 2.1 Upper bound and lower bound for (1,0)-k-SAT . . . . 19
**List of Figures**

| Figure 2.1 | $k = 5, r = 1, 1.2, 1.6, 2, 2.4, 2.8, 3.2$ (top down) | 19 |
| Figure 2.2 | Experimental results on $(1,0)$-super solutions of random 4-CNF formula, where $r = m/n$ is the clause-to-variable ratio and $n = 500.$ | 20 |
| Figure 3.1 | A vertex-minimal non-cograph+$v$ graph | 29 |
| Figure 3.2 | Reduction from 3-SAT to $\chi(G) = 3$ | 30 |
| Figure 3.3 | Worst case example of greedy $P_4$-free colouring | 32 |
| Figure 3.4 | Reduction from STABLE SET to COGRAPH problem | 34 |
| Figure 3.5 | A graph $J$ of $c(J) = 3$ with 10 vertices | 38 |
| Figure 3.6 | Another view of graph $J$ of $c(J) = 3$ with 10 vertices | 40 |
Acknowledgements

Completing this thesis would have been impossible without the help of Yong Gao, Donovan Hare and James Nastos. The greatest mathematical influence was from my supervisors, Yong Gao and Donovan Hare, and has been invaluable. Besides my supervisors, I would like to give special thanks to James Nastos, for his help on the problems with cographs.
Chapter 1

Introduction

The constraint satisfaction problem (CSP) and graph colouring are important problems in theoretical computer science and graph theory. Super solutions and cograph colouring are generalizations to the standard solution concepts for CSPs and graph colouring respectively. In addition to their combinatorial interest, these solution concepts also have potential real-world applications. For example, super solutions can be used to model problems in Artificial Intelligence (AI) arising in dynamic or uncertain environments. Cograph colouring, also called cograph partitioning, may provide an alternative approach to graph clustering and community structures in network analysis. Graph clustering [49] studies the problem of finding sets of “related” vertices in graphs, while the analysis of community structures [44] deals with how to group nodes in a network into application-specific communities, such as densely connected subgraphs.

1.1 Threshold phenomena of super solutions in CSPs

Dynamic CSPs have been used to model many problems arising in uncertain, dynamic, or interactive environments. For a dynamic CSP, it is desirable to find solutions that can be modified at a low cost in response to changes in the environment. This requires that a solution is not only satisfying, but also has a certain degree of robustness or stability. There are two typical approaches to dynamic CSPs, the reactive approach and the proactive approach [53]. In the reactive approach, one aims at finding a solution that can be easily “repaired” if it is no longer satisfying in a changed environment. In the proactive approach, a solution is required to be robust in the sense that there always exist solutions that are close to it.

The super solution framework [26, 34] is a viable approach to formalize the notion of a robust or stable solution. An \((a, b)\)-super solution of a CSP instance is a satisfying solution such that, if the values assigned to any set of \(a\) variables are no longer available, a new solution can be found by reassign-
1.1. Threshold phenomena of super solutions in CSPs

ing values to these $a$ variables and at most $b$ other variables. The availability of large-scale, dynamic, uncertain, and networked data sources in many application domains provides a challenge and opportunity for the constraint programming community [45]. To address the challenge, we expect that concepts of robust and stable solutions, such as fault-tolerant models [48] and super-solutions [34], will continue to attract a great deal of interest [12].

Threshold phenomena is first observed by Erdős and Rényi in their seminal work on random graphs [20]. Let $G(n, p)$ be a random graph on $n$ vertices where each edge exists with probability $p$. A sequence of events $E_n$ occurs with high probability (w.h.p.) if $\lim_{n \to \infty} \mathbb{P}[E_n] = 1$. It has been shown that for many interesting graph properties $P$, the probability for $G(n, p)$ to have $P$ changes drastically at a certain critical value of $p$. This critical value or range of $p$ is called the threshold with respect to the property $P$. For example, when $p = o\left(\frac{\log n}{n}\right)$, a random graph is disconnected w.h.p., while when $p = \omega\left(\frac{\log n}{n}\right)$, a random graph is connected w.h.p. Threshold phenomena are also found in many other random models. For example, threshold behaviour of the solution probability of random SAT and CSPs has been intensively studied theoretically and empirically since 1990s, [2–4, 10, 17, 21, 54].

In general, finding super solutions to SAT and CSPs is NP-complete. In the AI and theoretical computer science literature, one of the fruitful approaches to understand the typical-case complexity of a hard problem is to study the probabilistic behaviour of random instances [4, 27]. By analysing the threshold phenomena of the solution probability and the correlated easy-hard-easy pattern of the instance hardness of the standard solution concept for SAT and CSPs, much insight has been gained on the effectiveness of many heuristics widely used to tackle these problems [15, 22, 27].

The first part of this thesis focuses on the probabilistic behaviour of super solutions for random instances of SAT and CSPs. Our analysis focuses on a special (but highly non-trivial) type of super solutions, the (1,0)-super solutions. A solution is a (1,0)-super solution if it is resistant to changes of any one variable. That is, for a (1,0)-super solution $\sigma$, there is always another solution $\sigma'$ such that $\sigma'$ and $\sigma$ have different values on one variable. We denote the problems of finding (1,0)-super solution for $k$-SAT and CSPs by $(1, 0)$-$k$-SAT and $(1, 0)$-CSP respectively. We find exact thresholds for the phase transition of $(1,0)$-super solutions for random 2-SAT and 3-SAT. As for $k \geq 4$, we establish upper and lower bounds on the threshold of random $(1,0)$-$k$-SAT. We also establish a non-trivial upper bound on the threshold of random $(1,0)$-binary CSPs.
1.2 Partitioning graphs into cographs

Starting from a single vertex, cographs are recursively defined by graph operations of disjoint union and complementation [40]. There are many interesting properties of cographs, such as not containing an induced path on four vertices, the existence of a unique tree representation, and the existence of a linear-time recognition algorithm. Partitioning graphs into cographs is a relatively new generalization of the well-known graph colouring problem. Thus, we call it graph cog-colouring. Colouring graphs into cographs has been studied in [1, 8, 25, 35]. The first section of [25] provides an interesting and motivating axiomatization for many generalizations of graph colouring, including cog-colouring.

Many difficult questions from the study of classic graph colouring have counterparts worthy of study in the context of graph cog-colouring. Recognizing graphs that can be partitioned into \( k \) cographs is proved to be \( \text{NP-complete} \) for any \( k \geq 2 \) in [1]. Define the cog-chromatic number of a graph \( G \) to be the smallest possible \( k \) such that \( G \) can be partitioned into \( k \) cographs and denote it by \( c(G) \). A graph \( G \) is \( k \)-cog-colourable if \( c(G) \leq k \) and is \( k \)-cog-chromatic if \( c(G) = k \), where \( k \geq 1 \). A variety of bounds and computational complexity questions regarding the cog-chromatic number of graphs with different properties and computational complexity questions have been studied [25]. For example, for any triangle-free graph \( G \), \( c(G) \leq \chi(G) \leq 2 \cdot c(G) \), where \( \chi(G) \) is the chromatic number of \( G \). Another example is that, for any planar graph \( G \) with girth at least 11, \( c(G) \leq 2 \). It is also shown that, deciding cog-chromatic numbers is very hard, even on very restricted graphs. For example, the following two decision problems are \( \text{NP-complete} \): (1) deciding \( c(G) \leq 2 \) for a planar graph \( G \) of maximum degree 6; (2) deciding \( c(G) \leq k \), \( k \geq 2 \), for a chordal graph \( G \). However, it is linear time to decide \( \chi(G) \leq 2 \) for any graph \( G \) and it is polynomial time solvable to determine \( \chi(G) \) for any chordal graph \( G \) [23].

The second part of this thesis studies the cog-chromatic number. We first study 2-cog-colourable graphs. \( P_4 \)-sparse graphs and split-perfect graphs are shown to be 2-cog-colourable. We also prove that graph colouring is \( \text{NP-hard} \) on 2-cog-chromatic graphs. Since the \( k \)-cog-colourability is \( \text{NP-hard} \) to determine for any \( k \geq 2 \), we try to approximate the cog-chromatic number for graphs. Though we believe it is computationally hard to approximate \( c(G) \) for a general graph \( G \), we do not have a proof. We consider two greedy strategies of cograph colouring, one is based on lists of vertices and the other repeatedly colours a maximum induced cograph in a new colour. An algorithm of partitioning graphs into at most \( \left\lceil \frac{\Delta+1}{2} \right\rceil \) cographs is found,
where $\Delta$ is the largest degree of vertices in the graph. Finally, we study how to construct $k$-cog-chromatic graphs that satisfy several conditions. For example, we find a way of creating a graph $G$ with $O(2^k)$ vertices and $c(G) = k$. A graph $G$ is cog-critical if any vertex-deletion decreases its cog-chromatic number. A cog-critical graph $G$ is $k$-cog-critical if $c(G) = k$. We find two different constructions to create $k$-cog-critical graphs with arbitrarily many number of vertices, for any fixed $k \geq 3$.

### 1.3 Overview

In Chapter 2, we analyse a special (but highly non-trivial) type of super solutions, the $(1,0)$-super solutions. In Section 2.1, we discuss our observation on the equivalence between a $(1,0)$-$k$-SAT and a standard satisfying solution of a properly-constructed $(k - 1)$-SAT instance, which plays a crucial role in our analysis of the threshold behaviour of $(1,0)$-super solutions. In Section 2.4 and Section 2.5, we prove exact thresholds for the phase transition of $(1,0)$-super solutions for random $2$-SAT and $3$-SAT by making use of the equivalence presented in Section 2.1. In order to bound the probability of satisfiable $2$-SAT, we use a sufficient condition and a necessary condition of satisfiability of $2$-SAT proposed in [11]. However, our analysis is more involved than theirs because we have to handle the dependencies introduced in the translated equivalent SAT instances. For $k > 3$, we do not have any sufficient or necessary condition of satisfiability of $k$-SAT that are strong enough to be used to prove thresholds. Therefore, we have to dig deeper and try to study the distributions of $(1,0)$-super solutions. The technique developed in [3] and [4] enables us to have a peek at the distribution of solutions and design weights on different solutions such that thresholds can be proved. This technique of weighting solutions is so powerful that the long standing gap between upper bound and lower bound of thresholds for satisfiability of $k$-SAT is almost closed [4]. We apply this technique in Section 2.6 and establish upper and lower bounds on the threshold of random $(1,0)$-$k$-SAT for $k \geq 4$. Finally, in Section 2.8, we establish an upper bound on the threshold of random $(1,0)$-binary CSPs. Our analysis of the threshold for $(1,0)$-CSP is very complicated and the results are not very satisfying. We think that more advanced tools may be needed for analysing super solutions for random CSPs.

In Chapter 3, we study the problem of partitioning graphs into cographs. We first study properties of $2$-cog-colourable graphs in Section 3.2. We find that $P_4$-sparse graphs, split-perfect graphs are subclasses of $2$-cog-colourable...
1.3. Overview

graphs. We prove that deciding the chromatic number of $k$-cog-colourable graphs is NP-complete for any fixed $k \geq 2$. In Section 3.3, we try different ways of partitioning 2-cog-colourable graphs into as small number of cographs as possible. We study the bad performance of greedy colouring and how to find maximum induced cographs from $k$-cog-colourable graphs. We move on to study the size of $k$-cog-chromatic graphs with the smallest number of vertices in Section 3.4. We find a construction which is more efficient than the one in [25] and conjecture that the smallest $k$-cog-chromatic graph has $O(2^k)$ vertices. Finally and most interestingly, we discuss cog-critical graphs in Section 3.5. We give two nice ways of constructing $k$-cog-critical graphs $G$ with arbitrary number of vertices, for any fixed $k \geq 3$.
Chapter 2

Super solutions for random $k$-SAT and CSPs

In this chapter, we study the probabilistic behaviour of the $(1,0)$-super solution for random $k$-SAT and random binary CSPs. Let $X = \{x_1, \ldots, x_n\}$ be a set of $n$ boolean variables. A literal is a variable or its negation. A $k$-clause is a disjunction of $k$ different literals and a $k$-CNF formula is a conjunction of some $k$-clauses. An assignment $\sigma$ is a mapping $\sigma : X \to \{1, 0\}^n$ and is said to satisfy a $k$-CNF formula $F$ if each clause of $F$ contains at least one literal that evaluates to true under $\sigma$. A satisfying assignment is also called a solution.

According to the definition of $(a,b)$-super solutions [34], a $(1,0)$-super solution for a $k$-SAT is a solution such that changing the value assigned to exactly one variable will not violate any clause. Equivalently, a $(1,0)$-super solution is an assignment such that every clause contains at least two literals that evaluate to true under the assignment.

2.1 $(1,0)$-super solutions for $k$-SAT

In this subsection, we present a new equivalent condition for a $(1,0)$-super solution, which plays a crucial role in our analysis.

**Definition 2.1 (Projection).** The projection of a clause $C = (l_1 \lor \cdots \lor l_k)$ is defined to be $T(C) = \land_{i=1}^{k} (\lor_{j \neq i} l_j)$—the conjunction of the $(k-1)$-clauses in $C$. We say that $C$ projects onto $T(C)$ and call clauses in $T(C)$ siblings. The projection of a CNF formula $F$ is defined to be $T(F) = \land_{C_i \in F} T(C_i)$.

For example, the projection of $(x_1 \lor \overline{x_2} \lor x_3)$ is $(x_1 \lor \overline{x_2}) \land (x_1 \lor x_3) \land (\overline{x_2} \lor x_3)$. For a $k$-clause $C$, its projection has $k$ clauses of size $(k-1)$ and each such $(k-1)$-clause is a $(k-1)$ combination from those $k$ literals in $C$. Thus, if $\sigma$ satisfies at least two literals of $C$, then $\sigma$ must satisfy at least one literal of each $(k-1)$-clause in the projection of $C$. Therefore, we have the following lemma.
2.2 Random models of $k$-SAT

**Lemma 2.2.** An assignment $(1,0)$-satisfies $F$ if and only if it satisfies $T(F)$.

The following theorem complements existing results on the worst-case complexity of super solutions given in [34].

**Theorem 2.3.** $(1, 0)$-$k$-SAT is in P for $k \leq 3$, and is NP-complete otherwise.

**Proof.** We can check whether an assignment is a $(1,0)$-super solution of a $k$-CNF in $O(k \cdot m)$ time, where $m$ is the number of clauses. Thus, $(1,0)$-$k$-SAT is in NP for any fixed $k$. Any instance of $(1,0)$-3-SAT $F$ can be solved by solving the 2-SAT instance of $T(F)$, which is in P. For $k \geq 4$, we first prove the NP-completeness of $(1,0)$-4-SAT via a reduction from 3-SAT. Note that, $\sigma$ satisfies $(l_1 \lor l_2 \lor l_3)$ if and only if it $(1,0)$-satisfies $(l_1 \lor l_2 \lor l_3 \lor 1)$.

For any 3-SAT $F$, we reduce it into a 4-SAT $F'$ as following in three steps. First, create 4 additional variables, $Y = \{y_1, y_2, y_3, y_4\}$ and a 4-SAT $F_Y$ of all the possible $\binom{4}{2}$ clauses, where each clause has exactly two negations of variables.

$$F_Y = (y_1 \lor y_2 \lor \overline{y}_3 \lor \overline{y}_4) \land (y_1 \lor y_3 \lor \overline{y}_2 \lor \overline{y}_4) \land (y_1 \lor y_4 \lor \overline{y}_2 \lor \overline{y}_3)$$

$$\land (y_2 \lor y_3 \lor \overline{y}_1 \lor \overline{y}_4) \land (y_2 \lor y_4 \lor \overline{y}_1 \lor \overline{y}_3) \land (y_3 \lor y_4 \lor \overline{y}_1 \lor \overline{y}_2)$$

Secondly, for each clause $c_i$ in $F$, add $c'_i = (c_i \lor y_1)$ into $F'$. Finally, let $F'$ be the conjunction of $F'$ and $F_Y$. Note that, any assignment that $(1,0)$-satisfies $F_Y$ must have $\sigma(y_i) = 1, 1 \leq i \leq 4$. Thus, $\sigma$ is a solution of $F$ if and only if it is a $(1,0)$-super solution of $F'$. Therefore, $(1,0)$-4-SAT is NP-complete. Similar method can be used to reduce any $k$-SAT instance to $(1,0)$-$(k + 1)$-SAT instance. \hfill \Box

2.2 Random models of $k$-SAT

We denote by $F_k(n, m)$ the standard random model for $k$-CNF formulas on $n$ variables where the $m$ clauses are selected uniformly at random without replacement from the set of all possible $2^k \binom{n}{k}$ $k$-clauses. As sometimes it is hard to directly analyse $F_k(n, m)$ due to the dependence created by selecting the clauses without replacement, we consider two related models. The first model selects from all $2^k \binom{n}{k}$ proper clauses with replacement. The second model selects each literal uniformly and independently with replacement. Both models may result in improper formula and the second model may have improper clauses. A clause is proper if it does not have repeated literals. A formula is proper if it does not have repeated clauses and each clause is
2.3. The second moment method

proper. As long as \( k \) is fixed, the number of improper clauses and repeated clauses is \( o(n) \) w.h.p. Therefore, with-high-probability properties of \((1,0)\)-satisfiability hold in \( F_k(n, m) \) and two related models simultaneously. For notational convenience, we denote all three models by \( F_k(n, m) \). Also, when there is no ambiguity from the context, we use \( F_k(n, m) \) to denote a random formula in the model \( F_k(n, m) \). When \( k \leq 3 \), we use the first variant of the model. When \( k \geq 4 \), we use the second variant of the model. Throughout this chapter, we assume that \( k \) is fixed but can be arbitrarily large.

Due to Lemma 2.2, for a fixed \( k \)-SAT \( F \) from \( F_k(n, m) \), the probability for \( F \) to be \((1,0)\)-satisfiable equals the probability for its projection \( T(F) \) to be satisfiable. This, however, does not imply that the probability for a random formula in \( F_k(n, m) \) to be \((1,0)\)-satisfiable equals the probability for a random formula in \( F_{k-1}(n, km) \) to be satisfiable. This is because for a fixed \((k-1)\)-SAT formula \( F \) of \( km \) clauses, the probability that \( F \) is selected from \( F_{k-1}(n, km) \) is different from the probability that \( F \) is the projection of some formula in \( F_k(n, m) \).

2.3 The second moment method

The probabilistic method, initiated by Paul Erdős, is a powerful tool in combinatorics. Roughly speaking, in order to prove that an object with certain properties exists, one constructs an appropriate probability space and shows that a randomly chosen object in this space has the desired properties with positive probability. Though the probabilistic method only proves the existence of some object, there are techniques that help design algorithms to find the satisfying objects [6]. If we can prove that the existence probability goes to arbitrarily close to 1 when the size of the problem goes to infinity, we say that the object exists with high probability (w.h.p). The second moment method is a widely used tool to establish this type of results.

In the context of this section, let \( X \) be a nonnegative integer-valued random variable. Denote by \( E[X] \) the expectation of \( X \), and \( Var[X] \) the variance of \( X \), i.e., \( Var[X] = E[(X - E[X])^2] \). For notational simplicity in formulas, \( \mu \) and \( \sigma^2 \) are often used to replace \( E[X] \) and \( Var[X] \). We assume that \( \sigma > 0 \) and call it the standard deviation. The second moment method is based on the following two famous theorems.

**Theorem 2.4** (Markov’s Inequality, [6]). For any \( a > 0 \),

\[
P[X \geq a] \leq \frac{E[X]}{a}
\]
2.3. The second moment method

**Theorem 2.5** (Chebyshev’s Inequality, [6]). For any positive \( \lambda \),

\[
P[|X - \mu| \geq \lambda \sigma] \leq \frac{1}{\lambda^2}
\]

When \( X \) represents the number of desired objects, e.g. solutions of a problem, we have \( P[X > 0] = P[X \geq 1] \leq \mathbb{E}[X] \). Therefore, in order to show that \( X = 0 \) w.h.p., we prove that \( \mathbb{E}[X] \) goes arbitrarily close to 0. On the other hand, in order to prove \( X > 0 \) w.h.p., we prove that \( P[X = 0] \) goes arbitrarily close to 0.

**Theorem 2.6.** \( P[X = 0] \leq \frac{\text{Var}[X]}{\mathbb{E}[X]^2} \)

*Proof.* Substituting \( \frac{\lambda}{\sigma} \) with \( \frac{\mu}{\sigma} \) in the Chebyshev’s Inequality, we have

\[
P[X = 0] \leq P[|X - \mu| \geq \lambda \sigma] \leq \frac{1}{\lambda^2} = \frac{\sigma^2}{\mu^2}
\]

\[\square\]

In order to show \( X > 0 \) w.h.p. by the second moment method, there are two steps. First, prove that \( \mathbb{E}[X] \) goes to infinity. Second, bound \( P[X = 0] \) with some infinitesimal. This can be done by showing that \( \text{Var}[X] = o(\mathbb{E}[X]^2) \), according to Theorem 2.6. The relation \( \text{Var}[X] = o(\mathbb{E}[X]^2) \) intuitively means that \( \mathbb{E}[X]^2 = \Theta(\mathbb{E}[X^2]) \). The following inequality is stronger than the one in Theorem 2.6 and plays an essential role in our analysis.

**Lemma 2.7** (Exercise 3.6. [42]). For any nonnegative integer-valued random variable \( X \), if \( \mathbb{E}[X] > 0 \), then \( P[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]} \).

*Proof.* Let \( A \) be the set of positive values of \( X \), i.e., \( A = \{i| i > 0, P[X = i] > 0\} \). Then

\[
P[X > 0] \cdot \mathbb{E}[X^2] = \left( \sum_{i \in A} P[X = i] \right) \cdot \left( \sum_{i \in A} i^2 \cdot P[X = i] \right)
\]

\[
= \sum_{i \in A} i^2 \cdot P[X = i] + \sum_{i,j \in A, i \neq j} (i^2 + j^2) \cdot P[X = i] \cdot P[X = j]
\]

\[
\geq \sum_{i \in A} i^2 \cdot P[X = i] + \sum_{i,j \in A, i \neq j} (2 \cdot i \cdot j) \cdot P[X = i] \cdot P[X = j]
\]

\[
= \sum_{i \in A} \sum_{j \in A} i \cdot j \cdot P[X = i] \cdot P[X = j]
\]

\[
= \mathbb{E}[X]^2.
\]
2.4. The threshold for (1,0)-2-SAT

Since $\mathbb{E}[X^2] > 0$, we have $\Pr[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$. \hfill \square

Another easy-to-use lemma is as follows.

**Lemma 2.8** (Corollary 4.3.5 of [6]). Let $X = \sum_{i=1}^n X_i$, where $X_i$ is the indicator random variable for event $A_i$. Denote by $i \sim j$ if $i \neq j$ and the events $A_i, A_j$ are not independent. If $\lim_{n \to \infty} \mathbb{E}[X] = \infty$ and $\sum_{j \sim i} \Pr[A_j|A_i] = o(\mathbb{E}[X])$, then $X > 0$ holds w.h.p.

### 2.4 The threshold for (1,0)-2-SAT

In order to establish the threshold for a property, we must prove events describing that property occur w.h.p.

**Theorem 2.9.** $F_2(n, m)$ is (1,0)-satisfiable w.h.p. when $m = o(\sqrt{n})$ and is (1,0)-unsatisfiable w.h.p. when $m = \omega(\sqrt{n})$.

**Proof.** We say that two clauses are conflicting if some literal in one clause is the negation of some literal in the other clause. Note that a 2-CNF formula $F$ is (1,0)-satisfiable if and only if no conflicting clauses exists. Let $F = C_1 \land \cdots \land C_m$ and $X_{i,j}$ be the indicator variable for the event that $C_i$ conflicts with $C_j$. Then,

$$
\mathbb{E}[X_{i,j}] = \Pr[X_{i,j} = 1] = \frac{2(2(n-1) - 1) + 1}{2^2 \binom{n}{2}} = \frac{4n - 5}{2n(n-1)}.
$$

Denote by $X = \sum_{(i,j)} X_{i,j}$ the number of conflicting pairs in $F$, then

$$
\mathbb{E}[X] = \frac{m}{2} \mathbb{E}[X_{i,j}] = \frac{m^2}{n} (1 - o(1)).
$$

When $m = o(\sqrt{n})$, using Markov’s Inequality, we have

$$
\lim_{n \to \infty} \Pr[X > 0] \leq \lim_{n \to \infty} \mathbb{E}[X] = 0.
$$

Let $t = \binom{m}{2}$ and $p = \mathbb{E}[X_{i,j}]$, then $\mathbb{E}[X] = tp$. Note that, $X^2$ is composed of $t^2$ items of $X_{i,j}X_{i',j'}$. Group these items according to $h = |\{i, j, i', j'\}|$. We see that $\mathbb{E}[X_{i,j}X_{i',j'}]$ equals $p$ when $h = 2$, and equals $p^2$ otherwise. Thus, $\mathbb{E}[X^2] = tp + (t^2 - t)p^2$. When $m = \omega(\sqrt{n})$, using Lemma 2.7,

$$
\lim_{n \to \infty} \Pr[X > 0] \geq \lim_{n \to \infty} \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]} = \lim_{n \to \infty} \frac{tp}{tp + 1 - p} = 1.
$$

\hfill \square
2.5 The threshold for (1,0)-3-SAT

We use the equivalence (Lemma 2.2) between a (1,0)-super solution and a standard solution to study the threshold for the phase transition of (1,0)-super solutions of random 3-SAT. Specifically, we upper bound (resp. lower bound) the probability for $F$ to be (1,0)-unsatisfiable by the probability of some necessary (resp. sufficient) condition on the unsatisfiability of its projection $T(F)$ (a 2-CNF formula). The conditions we shall use are proposed in [11]. It is important to note that while $T(F)$ is a 2-CNF formula obtained from a random 3-CNF formula $F_3(n,m)$, $T(F)$ itself is not distributed as the random 2-CNF formula $F_2(n,m)$. This is the major obstacle we have to deal with in our analysis.

**Theorem 2.10.** $F_3(n, rn)$ is (1,0)-satisfiable w.h.p. when $r < 1/3$ and is (1,0)-unsatisfiable w.h.p. when $r > 1/3$.

The above result is proved in Lemma 2.12 and Lemma 2.14. In the proofs, we use $F$ to denote a random formula $F_3(n, rn)$, $m = rn$, and write $N = 2^3 \binom{n}{3}$.

### 2.5.1 Lower bound on the threshold for (1,0)-3-SAT

A bicycle of length $s$, $s \geq 2$, is a set of $s + 1$ 2-clauses $C_0, \ldots, C_s$ over a set of $s$ boolean variables $x_1, \ldots, x_s$ such that:

1. $C_0 = (u \lor l_1)$ and $C_s = (\overline{l_s} \lor v)$,
2. $C_i = (\overline{l_i} \lor l_{i+1})$, $0 < i < s$,

where $l_i$ is either $x_i$ or $\overline{x_i}$, and $u$ and $v$ are from $\{x_i, \overline{x_i} \mid 1 \leq i \leq s\}$. The following Lemma is proved in Theorem 3 of [11].

**Lemma 2.11.** If a 2-CNF is unsatisfiable, then it contains a bicycle.

Lemma 2.11 gives us a necessary condition of unsatisfiability of a 2-SAT. Using this necessary condition, we can upper bound the probability for a 3-SAT to be (1,0)-unsatisfiable, and thus establish a lower bound of the probability for a 3-SAT to be (1,0)-satisfiable.

**Lemma 2.12.** $F_3(n, rn)$ is (1,0)-satisfiable w.h.p. when $r < 1/3$.

**Proof.** For any fixed bicycle $B = C_0 \land \cdots \land C_s$, we consider the number of 3-CNF formulas that contain $B$ in their projection. In order to count this number, we must know the relationships between clauses in $B$. Specifically,
we need to know whether two clauses could be siblings, i.e., being projected from the same 3-clause in a 3-CNF formula. Let \( \mathcal{C} = \{C_1, \ldots, C_{s-1}\} \). It is clear that any pair of clauses in \( \mathcal{C} \) must have 4 different literals. Thus, no two clauses in \( \mathcal{C} \) can be siblings. Similarly, no tuple of three clauses from \( B \) can be siblings. However, \((C_0, C_i)\) can be siblings, and so are \((C_s, C_i)\), \(0 \leq i \leq s\). Let \( h \) be the number of pairs of clauses from \( B \) which are siblings. Denote by \( g(s, h) \) the number of 3-CNF formulas with \( m \) clauses that have \( B \) in their projection. Let \( F \) be such a 3-CNF formula. \( F \) needs to select \((s+1-h)\) 3-clauses so that \( T(F) \) has \( B \). Since \( B \) is fixed, \( h \) 3-clauses of these \((s+1-h)\) 3-clauses are fixed and each clause of the remaining \((s+1-2h)\) clauses has two literals fixed. Thus, there are \((2(n-2))^{s+1-2h}\) choices of the \((s+1-h)\) 3-clauses. The remaining \( m - (s+1-h) \) 3-clauses of \( F \) can be selected from \((N - (s+1-h))\) 3-clauses where \( N = 2^3(n/3) \). Therefore,

\[
g(s, h) = \frac{(N - (s+1-h))}{(m - (s+1-h))} \cdot (2(n-2))^{s+1-2h}.
\]

Let \( p(s) \) denote the probability that a fixed bicycle of length \( s \) is part of \( T(F) \). Then,

\[
p(s) \leq \binom{N}{m}^{-1} (g(s, 0) + 2s \cdot g(s, 1) + g(s, 2)) \leq \binom{N}{m}^{-1} 2(s+1) \frac{N - (s-1)}{m - (s-1)} \cdot (2(n-2))^{s-3} \leq \left( \frac{3r}{2(n-1)} \right)^{s-1} \frac{s+1}{2(n-2)^2}.
\]

Let \( N_s \) denote the number of different bicycles of length of \( s \) and \( X \) be the number of bicycles in \( T(F) \). It is clear that \( N_s < n^s 2^s (2s)^2 \). Therefore,

\[
\mathbb{E}[X] = \sum_{s=2}^{n} N_s p(s) \leq \frac{4n}{(n-2)^2} \sum_{s=2}^{n} s^2(s+1)(\frac{3rn}{n-1})^{s-1}.
\]

When \( r < 1/3 \),

\[
\lim_{n \to \infty} \mathbb{P}[X > 0] \leq \lim_{n \to \infty} \mathbb{E}[X] = 0.
\]

Thus, \( X = 0 \) w.h.p. It follows that \( F_3(n, rn) \) is \((1,0)\)-satisfiable w.h.p. \( \square \)

### 2.5.2 Upper bound on the threshold for \((1,0)\)-3-SAT

A *snake* of length \( t, t \geq 1 \), is a conjunction of \( 2t \) 2-clauses

\[
C_0 \land C_1 \land \cdots \land C_{2t-1}
\]
2.5. The threshold for (1,0)-3-SAT

and has the following structure.

1. \( C_i = (\overline{l_i} \lor l_{i+1}), 0 \leq i \leq 2t - 1. \ l_0 = l_{2t} = \overline{l_t} \)

2. For any \( 0 < i, j < 2t - 1, l_i \neq l_j \) and \( l_i \neq \overline{l_j} \).

The following Lemma is proved in Theorem 4 of [11].

**Lemma 2.13.** If a 2-CNF contains a snake, then it is unsatisfiable.

Lemma 2.13 gives us a sufficient condition of unsatisfiability of a 2-SAT. Using this sufficient condition, we can lower bound the probability for a 3-SAT to be (1,0)-unsatisfiable, and thus establish an upper bound on the probability for a 3-SAT to be (1,0)-satisfiable. Specifically, we show that when \( r > 1/3 \), the projection of \( F_3(k, rn) \) contains a snake of length \( \log_3 r n \) w.h.p.

**Lemma 2.14.** \( F_3(n, rn) \) is (1,0)-unsatisfiable w.h.p. when \( r > 1/3 \).

**Proof.** Let \( A \) be a snake of length \( t \), \( X_A \) be the indicator variable for the event that \( A \) occurs in \( F \). Note that only the two pairs, \((C_0, C_{t-1})\) and \((C_t, C_{2t-1})\), can be siblings. Let \( s = 2t - 1 \) and let \( p(t) \) be the number of occurrences of a fixed snake of length \( t \) in \( T(F) \). Then,

\[
 p(s) = \left( \frac{N}{m} \right)^{-1} \left( g(s, 0) + 2g(s, 1) + g(s, 2) \right) \\
 \approx \left( \frac{N}{m} \right)^{-1} 4g(s, 2) \approx \left( \frac{3r}{2n} \right)^{s-1} \frac{1}{n^2}.
\]

Let \( X \) denote the number of snakes of length \( t \) in \( T(F) \). Then,

\[
 E[X] = \binom{n}{s} s! 2^s p(s) \approx (3r)^s / n.
\]

When \( r > 1/3 \) and \( t = \omega(\log_3 r n) \), \( \lim_{n \to \infty} E[X] = \infty \).

In order to use the second moment method on \( X \), we have to consider correlations between snakes. To satisfy a clause \((l_i \lor l_j)\), if \( \overline{l_i} \) is true, then \( l_j \) must also be true. This implication can be represented by two arcs \((\overline{l_i}, l_j), (\overline{l_j}, l_i)\) in a digraph. The digraph of a snake of length \( t \) is a directed cycle \( \overline{l_t}, l_1, l_2, \ldots, l_t, \overline{l_t} \). It is clear that two snakes are not independent if and only if there are some common arcs in their directed cycles. Let \( B \) be another
2.5. The threshold for \((1,0)-3\text{-SAT}\)

snake of length \(t\). Suppose \(B\) shares \(i\) arcs with \(A\) and these arcs contain \(j\) vertices. Then,

\[
P[B|A] = \frac{\binom{N-2t-(2t-i)}{m-2t-(2t-i)}}{(2(n-2))^{2t}} \cdot \frac{(2(n-2))^{2t-i}}{2^t} \\
\leq \left( \frac{m-2t}{N-2t} \cdot 2(n-2) \right)^{2t-i} \leq \left( \frac{3r}{2n} \right)^{2t-i}.
\]

It is clear that those common \(i\) arcs comprise \((j-i)\) directed paths. Fixing \(A\), there are \(L_1\) choices for the shared \(j\) vertices to occur in \(B\), and there are \(L_2\) choices for the remaining \(2t-j\) vertices to occur in \(B\).

\[
L_1 = \left( 2 \cdot \binom{2t}{2(j-i)} \right)^2 \cdot (j-i)! \leq 4 \cdot (2t)^{4(j-i)}
\]

\[
L_2 \leq \binom{n-j+1}{2t-j} (2t-j)! \cdot 2^{2t-j} \leq (2(n-j+1))^{2t-j}
\]

For a given \(A\), let \(A(i,j)\) be the set of snakes sharing \(i\) arcs and \(j\) vertices with \(A\), and write

\[
p(i,j) = \sum_{B \in A(i,j)} P[B|A] = L_1 L_2 P[B|A]
\]

\[
\leq \left( \frac{3r}{2n} \right)^{2t-i} \cdot 4(2t)^{4(j-i)} (2(n-j+1))^{2t-j}.
\]

If \(i \leq t\), then \(i+1 \leq j \leq 2t\). If \(t < i \leq 2t\), then \(i+1 \leq j \leq 2t\). Let \(A \sim B\) denote the fact that \(A\) and \(B\) are dependent.

\[
\sum_{A \sim B} P[B|A] = \sum_{i=1}^{2t} \sum_{j=i+1}^{\min\{2i,2t\}} p(i,j) = \sum_{j=2}^{2t} \sum_{i=j/2}^{j-1} p(i,j)
\]

\[
\leq \sum_{j=2}^{2t} (2(n-j+1))^{2t-j} 4 \sum_{i=j/2}^{j-1} \left( \frac{3r}{2n} \right)^{2t-i} (2t)^{4(j-i)}
\]

\[
\leq \sum_{j=2}^{2t} (2(n-j+1))^{2t-j} 4 \cdot \frac{3r}{2n} \left( \frac{2r}{2n} \right)^{2t-j+1} (2t)^4
\]

\[
\leq \sum_{j=2}^{2t} 2j \left( \frac{3r}{2n} \right) (2t)^4
\]

\[
\leq \Theta(1) \cdot \frac{1}{n} t^6 = o \left( \frac{1}{n} (3r)^2t \right) = o(E[X]).
\]
According to lemma 2.8, \( \lim_{n \to \infty} \mathbb{P}[X > 0] = 1 \). Therefore, \( F \) is (1,0)-unsatisfiable w.h.p.

## 2.6 The threshold for (1,0)-\( k \)-SAT

The projection in Definition 2.1 provides a way to translate between an (1,0)-3-SAT instance and a 2-SAT instance. Because there are easy-to-use sufficient condition and necessary condition for 2-SAT, we can establish the thresholds for (1,0)-2-SAT without looking into the distributions of its (1,0)-super solutions. However, for \( k > 3 \), we do not have such sufficient conditions and necessary conditions we had before. Therefore, we have to study the distributions of (1,0)-super solutions in order to get the thresholds. We start with upper bounds on the thresholds for (1,0)-\( k \)-SAT.

**Theorem 2.15.** For all \( k \geq 3 \), \( F_k(n, rn) \) is (1,0)-unsatisfiable w.h.p. when \( r > \frac{2^k}{k+1} \ln 2 \).

**Proof.** For a fixed assignment \( \sigma \), a random \( k \)-clause is satisfied with probability \( 1 - \frac{1+k}{2^k} \). Denote by \( X = X(F) \) the number of (1,0)-super solutions of \( F_k(n, m) \). Then

\[
\mathbb{E}[X] = 2^n \left(1 - \frac{1+k}{2^k}\right)^m = \left(2\left(1 - \frac{1+k}{2^k}\right)^r\right)^n.
\]

When \( r > \frac{-\ln 2}{\ln(1 - \frac{1+k}{2^k})} \), by Markov’s Inequality, we have

\[
\lim_{n \to \infty} \mathbb{P}[X \geq 1] \leq \lim_{n \to \infty} \mathbb{E}[X] = 0.
\]

Therefore, \( F \) is (1,0)-unsatisfiable w.h.p. Since \( \frac{2^k}{k+1} \ln 2 > \frac{-\ln 2}{\ln(1 - \frac{1+k}{2^k})} \), the theorem follows.

In the rest of this section, we establish a lower bound on the threshold for \( k > 3 \) and show that the ratio of the lower bound over the upper bound goes to 1 as \( k \) goes to infinity. Our analysis uses the techniques introduced in [4] for proving lower bounds on the threshold for the phase transition of standard satisfying solutions of random SAT, but the calculation we have to deal with is even more complicated. The idea is to use a weighting scheme on satisfying assignments when using the second moment method to prove lower bounds on the threshold.
2.6. The threshold for (1,0)-k-SAT

For a clause \( c \), denote by \( S(c) \) the set of (1,0)-super solutions of \( c \), \( S^0(c) \) (resp. \( S^1(c) \)) the set of assignments that satisfies exactly 0 (resp. 1) literal of \( c \). Define \( H(\sigma, c) \) to be the number of satisfied literals minus the number of unsatisfied literals under an assignment \( \sigma \). For an event \( A \), let \( 1_A \) be its indicator variable. The weight of \( \sigma \) w.r.t. \( c \) is defined as 
\[
w(\sigma, F) = \gamma^{H(\sigma,c)} 1_{\sigma \in S(F)} = \prod_{c_i} w(\sigma, c_i).
\]

Let \( X = \sum_{\sigma} w(\sigma, F) \). It is clear that \( X > 0 \) if and only if \( F \) is (1,0)-satisfiable. We note that by viewing an instance of (1,0)-k-SAT as a generalized Boolean satisfiability problem (Boolean CSP) and applying the conditions established in [14], random (1,0)-k-SAT has a sharp threshold. Therefore, to show \( X > 0 \) w.h.p., it is sufficient to prove that \( \Pr[X > 0] \) is larger than some constant.

For a fixed \( \sigma \) and a random \( k \)-clause \( c \),
\[
\mathbb{E}[w(\sigma, c)] = \mathbb{E}\left[\gamma^{H(\sigma,c)} \left(1 - 1_{\sigma \notin S^0(c)} - 1_{\sigma \in S^1(c)}\right)\right] = \left(\frac{\gamma + \gamma^{-1}}{2}\right)^k - 2^{-k-1} - k 2^{-k} \gamma^{-k} = \phi(\gamma).
\]

Thus, \( \mathbb{E}[X] = \sum_{\sigma} \prod_{c_i} \mathbb{E}[w(\sigma, c)] = (2\phi(\gamma)r)^n. \)

We now consider \( \mathbb{E}[X^2] \). Fix a pair of assignments \( \sigma, \tau \) such that they overlap each other on exactly \( z = \alpha n \) variables. Consider a random \( k \)-clause \( c \) and write
\[
f(\alpha) = \mathbb{E}[w(\sigma, c)w(\tau, c)] = \mathbb{E}\left[\gamma^{H(\sigma,c)+H(\tau,c)} 1_{\sigma,\tau \in S(c)}\right].
\]

We have the following equations for relevant events
\[
1_{\sigma,\tau \in S(c)} = 1 - 1_{\sigma \notin S(c)} - 1_{\tau \notin S(c)} + 1_{\sigma,\tau \notin S(c)};
1_{\sigma \notin S(c)} = 1_{\sigma \in S^0(c)} + 1_{\sigma \in S^1(c)};
1_{\sigma,\tau \notin S(c)} = 1_{\sigma \in S^0(c), \tau \in S^0(c)} + 1_{\sigma \in S^0(c), \tau \in S^1(c)} + 1_{\sigma \in S^1(c), \tau \in S^0(c)} + 1_{\sigma \in S^1(c), \tau \in S^1(c)}.
\]

For mathematical expectations, we have
\[
\mathbb{E}\left[\gamma^{H(\sigma,c)+H(\tau,c)}\right] = \left(\alpha \left(\frac{\gamma^2 + \gamma^{-2}}{2}\right) + 1 - \alpha\right)^k.
\]
2.6. The threshold for (1,0)-\(k\)-SAT

\[
E \left[ \gamma^{H(\sigma,c)+H(\tau,c)}1_{\sigma \notin S(c)} \right] = 2^{-k} \left( (\alpha \gamma^{-2} + 1 - \alpha)^k + k(\alpha \gamma^{-2} + 1 - \alpha)^{k-1}(\alpha \gamma^2 + 1 - \alpha) \right),
\]

\[
E \left[ \gamma^{H(\sigma,c)+H(\tau,c)}1_{\sigma,\tau \notin S(c)} \right] = 2^{-k} \left( \alpha^k \gamma^{-2k} + 2k\gamma^{-2k+2}\alpha^{k-1}(1 - \alpha) + \gamma^{-2k+4}(k\alpha^k + k(k - 1)\alpha^{k-2}(1 - \alpha)^2) \right).
\]

Therefore, the expectation of \(X^2\) can be written as

\[
E[X^2] = \sum_{\sigma,\tau} E[w(\sigma, F)w(\tau, F)] = \sum_{\sigma,\tau} \prod_{c_i} E[w(\sigma, c_i)w(\tau, c_i)] = 2^n \sum_{z=0}^{n} \binom{n}{z} f(z/n)^{rn}.
\]

The following lemma from [3] enables us to consider the dominant part of \(E[X^2]\).

**Lemma 2.16.** Let \(h\) be a real analytic positive function on \([0, 1]\) and define \(g(\alpha) = h(\alpha)/(\alpha^\alpha(1 - \alpha)^{1-\alpha})\), where \(0^0 \equiv 1\). If \(g\) has exactly one maximum at \(g(\beta), \beta \in (0, 1)\), and \(g'(\beta) < 0\), then there exists some constant \(C > 0\) such that for all sufficiently large \(n\), \(\sum_{z=0}^{n} \binom{n}{z} h(z/n)^n \leq C \times g(\beta)^n\).

Let \(g_r(\alpha) = f(\alpha)^r/(\alpha^\alpha(1 - \alpha)^{1-\alpha})\). We say that \(g_r(\alpha)\) satisfies the dominant condition if \(g_r''(1/2) < 0\) and \(g_r(1/2)\) is the unique global maximum. According to lemma 2.16 and \(\phi(\gamma)^2 = f(1/2)\), if \(g_r(\alpha)\) satisfies the dominant condition, then

\[
\Pr[X > 0] > \frac{E[X^2]}{E[X^2]} = \frac{4^n f(1/2)^{rn}}{E[X^2]} > \frac{(2g_r(1/2))^n}{C \cdot (2g_r(1/2))^n} = \frac{1}{C},
\]

where \(C\) is a constant when \(k\) is fixed.

If we can find suitable \(\gamma\) and \(r\) so that \(g_r(\alpha)\) satisfies the dominant condition, then we can prove \(X > 0\) w.h.p. Note that the dominant condition implies \(f'(1/2) = 0\). The weighting function \(w(\sigma, c)\) defined on an assignment \(\sigma\) and a \(k\)-clause \(c = (l_1 \wedge \cdots \wedge l_k)\) can be viewed as defined on a vector \(v\) in \((-1, 1)^k\), where \(v(i) = -1\) if \(l_i\) evaluates to False under \(\sigma\) and \(v(i) = 1\).
2.6. The threshold for (1,0)-k-SAT

otherwise. According to [4], a necessary condition of \( f'(1/2) = 0 \) is that the sum of vectors scaled by their corresponding weights is 0, i.e.,

\[
\sum_{v \in \{-1,1\}^k} w(v)v = 0.
\]

Following the idea of [4], the \( \gamma \) which gives the best lower bound \( r \) should also make weights of (1,0)-satisfying assignments as equal as possible. For our problem of (1,0)-k-SAT, \( \gamma \) should satisfy the following equation

\[
\sum_{i=2}^{k} \binom{k}{i} \gamma^{2i-k}(2i - k) = 0. \tag{2.1}
\]

When \( k = 4 \), this equation requires \( \gamma = 0 \), which contradicts our prerequisite that \( \gamma > 0 \). Thus, the weighting scheme is not meaningful when \( k = 4 \). Therefore, we consider \( k > 4 \) first and then solve the \( k = 4 \) case in a different way.

It is too complicated to directly prove that \( g_r(\alpha) \) satisfies the dominant condition, at least for small \( k \). Therefore, we plot figures to show how \( g_r(\alpha) \) changes when \( k \) is fixed. Figure 2.1 shows the case when \( k = 5 \). Figures showing \( g_r(\alpha) \) of other fixed \( k \)s share the same changing pattern with the case \( k = 5 \). For each \( k \), when \( r \) is smaller than some \( r^*_k \), \( g_r(\alpha) \) satisfies the dominant condition and \( F_r(n,rn) \) is (1,0)-satisfiable w.h.p. Thus \( r^*_k \) is a lower bound for \( F_k(n,rn) \) to be (1,0)-satisfiable. We do this analysis for \( k \) up to 11 and show the values in Table 2.1. It is clear to observe that the ratio of the lower bound over the upper bound of thresholds of \( F_k(n,rn) \) goes to 1 as \( k \) becomes large.

We still have to solve the case \( k = 4 \) separately, where the weighting scheme, \( w(\sigma,c) = \gamma^{H(\sigma,c)}1_{\sigma \in S(c)} \), does not work for any \( \gamma > 0 \). Since \( 2i - k \) is either 0 or positive when \( k = 4 \) and \( i \geq 2 \), equation 2.1 cannot hold. Thus, a compromise is to consider only those assignments which satisfy \( 2i - k = 0 \). Specifically, for each clause of \( F \), exactly two literals are satisfied and exactly two literals are unsatisfied. And every satisfying assignment has the same weight, 1. By doing this, the likelihood for an assignment not to be in \( X \) is doubled. Therefore, the upper bound for such solutions becomes \( \frac{2^{k-1}}{1+k} \ln 2 \), half of the upper bound for (1,0)-4-SAT. The remaining analysis of finding \( r^*_k \) that satisfying the dominant condition is similar to the analysis of \( k > 4 \). The \( r^*_4 \) we found is 0.602.
2.7 Designing SAT benchmarks by projections

We conducted preliminary experiments by using the uniform $k$-SAT generator by Adrian Balint [5] and the SAT solver, MiniSAT [18], to solve 3-CNF formulas projected from random instances of $(1,0)$-4-SAT on $n = 500$ variables. The experiments were conducted on a 2.5GHz Intel Core i5 processor with 8GB memory. The time limit set for the solver are 600 seconds. Instances that cannot be solved within the time limit are treated as unsatisfiable and a running time of 600 seconds is used in calculating the average time. The results of the solution probability and solution time (measured in CPU seconds) are plotted in Figure 2.2, where each data point is the average of 100 randomly generated instances. As depicted in Figure 2.2, the phase transition of the $(1,0)$-super solution is clear and the hardness peak at the phase transition seems to be as dramatic as (if not more dramatic than) those of standard random 3-SAT instances.

Our analysis of $(1,0)$-super solutions in Sections 2.5 and 2.6 makes use of the observation that a random $(1,0)$-k-SAT instance can be projected to an
2.7. Designing SAT benchmarks by projections

Figure 2.2: Experimental results on (1,0)-super solutions of random 4-CNF formula, where $r = m/n$ is the clause-to-variable ratio and $n = 500$.

equivalent standard $(k - 1)$-SAT instance. While the projected $(k - 1)$-SAT instance is “random”, its distribution is different from the standard random $(k - 1)$-SAT instances, due to the correlations among the projected $(k - 1)$ clauses. As our experiments shows, the projected $(k - 1)$-SAT “random” instances not only have a stylish easy-hard-easy hardness pattern, but also exhibit a seemingly more dramatic peak of harness at the phase transition. We note that the idea of projections and some of our analysis extend naturally to the case of $(a,0)$-super solutions for $a > 1$. Given a random instance $F$ of $(a,0)$-k-SAT with $1 < a < k$, we can obtain a $(k - a)$-CNF formula $H$ by projecting the clauses of $F$ recursively. It can be proved that $F$ has an $(a,0)$-super solution if and only if $H$ has a solution. Again, we note that while $F$ is a random $k$-CNF formula, the distribution of $H$ differs from that of the standard random $(k - a)$-CNF formula; we expect to see that clauses in $H$ will have a unique clustering structure that does not exist in a standard random CNF formula. Therefore, projecting from random $(a,0)$-k-SAT instances provides a promising approach to constructing new classes of random but structured SAT distributions. Our theoretical analysis and preliminary experiments on the hardness of such instances suggest that this class of instances may serve as a suite of SAT benchmarks that interpolate
2.8 Super solutions for random binary CSPs

A well-studied random model of Constraint Satisfaction Problems (CSP) is the standard Model B in [24]. Phase transitions for the random binary CSP under Model B is studied in [50]. In this section, we explore phase transitions for the (1,0)-super solution of random binary CSP under Model B. Random binary CSPs are defined on a domain $D$ of size $|D| = d$. A binary CSP $C$ consists of a set of variables $X = \{x_1, \ldots, x_n\}$ and a set of binary constraints $(C_1, \ldots, C_m)$. Each constraint $C_i$ is specified by its constraint scope, an unordered pair of two variables in $X$, and a constraint relation $R_{C_i}$ that defines a set of incompatible value tuples in the binary relation $D \times D$ for the scope variables. An incompatible value tuple is also called a restriction. The constraint graph of a binary CSP is a graph whose vertices correspond to the set of variables and edges correspond to the set of constraint scopes. We use the random CSP model $B_{n,m}^{d,q}$ defined as follows.

1. Its constraint graph is a random graph $G(n, m)$ where the $m$ edges are selected uniformly at random from all the possible $\binom{n}{2}$ edges.

2. For each edge, its constraint relation is determined by choosing each value tuple in $D \times D$ as a restriction independently with probability $q$.

Denote by $H(\sigma_1, \sigma_2)$ the set of variables being assigned different values by two assignments $\sigma_1$ and $\sigma_2$, i.e., $H(\sigma_1, \sigma_2) = \{x_i | \sigma_1(x_i) \neq \sigma_2(x_i), 1 \leq i \leq n\}$. Let $\sigma$ be a fixed assignment and $I$ be a random $B_{n,m}^{d,q}$ instance. Define the following three events:

1. $S(\sigma)$ : $\sigma$ is a solution for $I$.

2. $S_i(\sigma)$ : there exists another solution $\sigma'$ for $I$ such that $H(\sigma, \sigma') = \{x_i\}$.

3. $T(\sigma)$ : $\sigma$ is a (1,0)-super solution for $I$.

According to the relationship between (1,0)-super solutions and standard solutions,

$$\mathbb{P}[T(\sigma)] = \mathbb{P}[S(\sigma)] \mathbb{P}[\cap_{1 \leq i \leq n} S_i(\sigma) | S(\sigma)].$$

Estimating the probability of a (1,0)-super solution for a random CSP instance is, however, more complicated than estimating the probability of a
satisfying assignment, largely due to the fact that the events \( S_i(\sigma), 1 \leq i \leq n \), are not independent. This is the major hurdle we need to overcome.

Note that in a random CSP instance, the selection of constraints and the selection of restrictions for each constraint are independent. Let \( C \subset (X \times X)^m \) be the collection of all possible sets of \( m \) unordered pairs of variables. For a given set \( e \in C \) of \( m \) unordered pairs, denote by \( E(e) \) the event that \( e \) is selected as the set of constraints of the random instance \( I \). Let \( m_i \) be the number of constraints \( x_i \) is involved with. Considering an assignment \( \sigma' \), \( H(\sigma, \sigma') = \{x_i\} \), it is clear that

\[
\mathbb{P}[S_i(\sigma') \mid S(\sigma) \cap E(e)] = 1 - (1 - q)^{m_i}.
\]

Let \( D' = D \setminus \{\sigma(x_i)\}, \sigma'(x_i) = y, p = 1 - q, \) then

\[
\mathbb{P}[S_i(\sigma) \mid S(\sigma) \cap E(e)] = \mathbb{P}[\bigcup_{y \in D'} S(\sigma') \mid S(\sigma) \cap E(e)]
\]

\[
= \mathbb{P}[\bigcap_{y \in D'} S(\sigma') \mid S(\sigma) \cap E(e)]
\]

\[
= 1 - \mathbb{P}[\bigcap_{y \in D'} S(\sigma') \mid S(\sigma) \cap E(e)]
\]

\[
= 1 - (1 - (1 - q)^{m_i})^{d-1}.
\]

This shows that, conditioned on \( S(\sigma) \) and a fixed set of constraints \( e, S_i(\sigma) \) and \( S_j(\sigma) \) are independent for any \( i \neq j \).

\[
\mathbb{P}[T(\sigma)] = \mathbb{P}[\bigcup_{e \in C} (E(e) \cap S(\sigma) \cap (\cap_{1 \leq i \leq n} S_i(\sigma)))]
\]

\[
= \sum_{e \in C} \mathbb{P}[E(e)] \mathbb{P}[S(\sigma) \mid E(e)] \mathbb{P}[\cap_{1 \leq i \leq n} S_i(\sigma) \mid S(\sigma) \cap E(e)]
\]

\[
= \left( \binom{n}{2} \right)^{-1} p^m \sum_{e \in C} \prod_{i=1}^{n} \left( 1 - (1 - p^{m_i})^{d-1} \right). \quad (2.2)
\]

Let \( Y_\sigma \) be an indicator variable of \( T(\sigma) \) and \( Y = \sum_\sigma Y_\sigma \) be the number of \((1,0)\)-super solutions. We have

\[
\mathbb{E}[Y] = d^n \cdot \mathbb{E}[Y_\sigma] = d^n \cdot \mathbb{P}[T(\sigma)]. \quad (2.3)
\]

**Theorem 2.17.** Consider the random CSP \( \mathcal{B}_{n,m}^{d,q} \) with \( d = \sqrt{n} \) and \( m = c \cdot n \ln n \) where \( c \) is a positive constant. Let \( p = 1 - q \). If \( c > \frac{1}{3 \ln 2} \), then

\[
\lim_{n \to \infty} \mathbb{E}[Y] = 0 \text{ and thus, } \mathcal{B}_{n,m}^{d,q} \text{ is } (1,0)\text{-unsatisfiable w.h.p.}
\]
2.8. Super solutions for random binary CSPs

Proof. Subject to \( \sum_{i=1}^{n} m_i = 2m \), the product term on the right hand side of Equation (2.2), achieves the global maximum when \( m_i = \frac{2m}{n} \), \( 1 \leq i \leq n \). This can be proved by the method of Lagrange multipliers. Let \( c = c' - \frac{1}{ln p} \).

According to Equation (2.2) and (2.3), we have

\[
E[Y] \leq (d \cdot p^{c \ln n} \cdot (1 - (1 - p^{c \ln n})^{d-1}))^n
\]

\[
= (d \cdot n^{c \ln p} \cdot (1 - (1 - n^{2c \ln p})^{d-1}))^n
\]

\[
\approx (n^{1/2 - c'} \cdot (1 - (1 - n^{-2c'})^{n^{1/2}}))^n.
\]

For any \( a, b \) satisfying \( 0 \leq a \leq 1 \) and \( ab < 1 \), we have \( (1 - (1 - x)^d) \geq 1 - ab \). If \( c' > 1/3 \), then

\[
E[Y] \leq (n^{1/2 - c'} \cdot n^{-2c'} n^{1/2})^n = (n^{-3c'})^n.
\]

Therefore, \( \lim_{n \to \infty} E[Y] = 0 \).

The case of \( c \leq -\frac{1}{3 \ln p} \) is even more difficult to analyse. In the following, we establish conditions for the expected number of \((1, 0)\)-solutions, \( E[Y] \), to go to infinity. Note that \( 1 - (1 - p^x)^d \) is increasing in terms of \( x \). Let \( \lambda = \frac{2m}{n} = 2c \ln n \) and \( t = r \ln n \). If \( m_i \leq \lambda + t \) for each \( 1 \leq i \leq n \), then

\[
E[Y] > d^m p^m \left( 1 - (1 - p^{\lambda + t})^{d-1} \right)^n \approx n^{(1 + (3c + r) \ln p)n}.
\]

Consequently, when \( r < \frac{1}{3 \ln p} - 3c \), \( \lim_{n \to \infty} E[Y] = \infty \). Therefore, for a fixed \( c < \frac{1}{3 \ln p} \), if we could find an appropriate \( r \) such that \( m_i \leq \lambda + t \) w.h.p., then we can prove \( \lim_{n \to \infty} E[Y] = \infty \). Since dependencies among \( m_i \)'s make it hard to analyse, we approximate them by the corresponding \( m'_i \)'s in the random model where edges in the constraint graph are selected with replacement, in which case each \( m'_i \) is a Binomial random variable with the distribution \( Bin(m, n/2) \). By the Chernoff Bound for Binomial distributions, we have for any \( u \geq 0 \),

\[
\Pr[m_i \geq \lambda + t] \leq \Pr[m'_i \geq \lambda + t] \leq e^{-u(\lambda + t)}(1 + p(e^u - 1))^m.
\]

Therefore,

\[
\Pr[\cap_{1 \leq i \leq n} m_i \leq \lambda + t] \geq 1 - \sum_{1 \leq i \leq n} \Pr[m_i > \lambda + t]
\]

\[
\geq 1 - n \Pr[m_1 > \lambda + t]
\]

\[
\approx 1 - e^{-u(2c + r) + (e^u - 1)2c + 1} \ln n.
\]
Consequently, if

\[
\frac{1}{u} + \frac{e^u - 1}{u} 2c - 2c < r < -\frac{1}{\ln p} - 3c, \quad r > 0, \quad u \geq 0,
\]

then w.h.p. \( m_i \leq \lambda + t, 1 \leq i \leq n \), and thus \( \mathbb{E}[Y] \) goes to infinity. Since \( c = c' \frac{1}{\ln p} \) and \( c' < \frac{1}{3} \), the parameter \( r \) has to satisfy

\[
-\frac{1}{\ln p} (1 - c'(1 + \frac{2(e^u - 1)}{u})) > \frac{1}{u}.
\]

By setting \( u = 1 \), we see that if \( c < -\frac{1}{10} \frac{1}{\ln p} \) and \( q = 1 - p < 0.43 \), then the expected number of \((1,0)\)-super solutions \( \mathbb{E}[Y] \) goes to infinity.

Note that what we get in the above passage is just a coarse lower bound on the thresholds for the \((1,0)\)-satisfiability of a random instance of \( B_{d,q}^{n,m} \). This is because \( \lim_{n \to \infty} \mathbb{E}[X] \) is only a necessary condition for \( \lim_{n \to \infty} \mathbb{P}[X > 0] = 1 \). In order to prove that there is a \((1,0)\)-super solution, we must use the second moment method to prove Lemma 2.8 or some other weaker version of it. Also note the significant gap between the lower bound and the upper bound we have derived in Theorem 2.17 and the above analysis. This indicates that analysing the threshold phenomena of \((1,0)\)-super solutions for random CSPs is a much more challenging task and requires new analytical ideas.
Chapter 3

Partitioning of graphs into cographs

3.1 Graphs and graph colouring

We first introduce some definitions and notations for graphs [7]. A graph $G$ is an ordered pair, $(V(G), E(G))$, often written as $G(V, E)$ where $V(G)$ is a set of vertices and $E(G)$ is a set of unordered pair of vertices. The order of a graph is the number of its vertices. An unordered pair $\{u, v\} \in E(G)$ is called an edge of $G$ and is often written as $uv$. For an edge $e = uv$, $u$ and $v$ are called ends of $e$ and $u$ is said to be adjacent to $v$. Two vertices $u$ and $v$ in $G$ are called non-adjacent if $uv \notin E(G)$. Ends of an edge are said to be incident with the edge, and vice versa. Two edges are adjacent if they share an end and are non-adjacent otherwise. A set of pairwise non-adjacent edges in a graph is called a matching. The neighbourhood of a vertex $v$ in a graph $G$, denoted by $N_G(v)$, is the set of vertices that are adjacent to $v$ in $G$. When $G$ is clear from the context, we simplify the notation as $N(v)$.

A loop is an edge with identical ends (e.g. for some vertex $u$, $\{u, u\}$ is an edge). Two or more edges with the same ends are called parallel edges. A graph is simple if it has no loops or parallel edges. In the context of this thesis, we consider simple graphs only. A vertex $v$ is isolated in a simple graph $G$ if $N_G(v) = \emptyset$. A path is a simple graph whose vertices can be arranged into a linear sequence such that ends of each edge are consecutive in the sequence. Paths with $k$ vertices are written as $P_k$. Two vertices in a graph $G$ are connected if there is a path between them in $G$. A graph is connected if every two vertices are connected and disconnected otherwise. A cycle with $k$ vertices, denoted by $C_k$, is a simple graph whose vertices can be arranged in a cyclic sequence such that ends of each edge are consecutive in the sequence. Cycles in simple graphs require at least three vertices. The length of a path or a cycle is the number of its edges. A path or cycle is odd (resp. even) if its length is odd (resp. even). An empty graph is graph in which no two vertices are adjacent and a complete graph is a simple graph
3.1. Graphs and graph colouring

in which every pair of vertices is adjacent. A complete graph with \( n \) vertices is denoted by \( K_n \).

The complement of a graph \( G \), denoted \( \overline{G} \), is a graph whose vertex set is \( V(G) \) and whose edges are the pairs of non-adjacent vertices of \( G \). Two graphs are disjoint if they have no vertex in common. The union of simple graphs \( G \) and \( H \) is the graph \( G \cup H \) with vertex set \( V(G) \cup V(H) \) and edge set \( E(G) \cup E(H) \). If \( G \) and \( H \) are disjoint, we refer to their union as a disjoint union and denote it by \( G + H \). A graph \( G \) is disconnected if and only if \( G \) is a disjoint union of some connected subgraphs and each such subgraph is called a connected component of \( G \). Given two disjoint graphs \( G \) and \( H \), the join of \( G \) and \( H \), denoted by \( G \oplus H \), is a graph with the vertex set \( V(G + H) \) and the edge set \( E(G + H) \cup \{ uv | u \in V(G), v \in V(H) \} \).

A graph \( F \) is called a subgraph of a graph \( G \) if \( V(F) \subseteq V(G) \) and \( E(F) \subseteq E(G) \). If \( F \) is a subgraph of \( G \), then \( G \) contains \( F \). A subgraph \( F \) of a graph \( G \) is called a proper subgraph of \( G \) if \( V(F) \subseteq V(G) \) or \( E(F) \subset E(G) \). For an edge \( e \in E(G) \), the edge-deletion subgraph, \( G - e \), is the graph with the same vertex set as \( G \) but whose edge set is \( E(G) \setminus \{e\} \). For a vertex \( v \in V(G) \), the vertex-deletion subgraph, \( G - v \), is the graph with the vertex set \( V(G) \setminus \{v\} \) and the edge set \( E(G) \setminus \{e | e \text{ is incident with } v\} \). A subgraph obtained by vertex deletions only is called an induced subgraph. If \( X \) is a set of vertex deletions, the resulting subgraph is denoted by \( G - X \) or \( G[Y] \) where \( Y = V \setminus X \). \( G[Y] \) is also known as the subgraph of \( G \) induced by \( Y \).

For a graph \( G \) and a nonempty set of vertices \( U \subseteq V(G) \), \( U \) is a stable set of \( G \) if \( G[U] \) is empty and \( U \) is a clique of \( G \) if \( G[U] \) is complete. A partition of a set \( V \) is a set of nonempty subsets of \( V \) such that each element of \( V \) is in exactly one of the subsets. The order of a partition is the number of subsets in the partition. A bipartition is a partition of order two. A partition of a graph \( G \) is the set of subgraphs induced by all subsets of some partition of \( V(G) \).

Two graphs \( G \) and \( H \) are isomorphic, written as \( G \cong H \), if there exists a bijection \( \theta : V(G) \rightarrow V(H) \) such that for every two vertices \( u, v \in V(G) \), \( uv \in E(G) \) if and only if \( \theta(u)\theta(v) \in E(H) \). A graph \( G \) is called \( H \)-free if there is no induced subgraph of \( G \) which is isomorphic to \( H \).

A colouring of a graph \( G \) is a function \( f : V(G) \rightarrow S \) where \( S \) is a finite set. The elements of \( S \) are called labels or colours. The vertices that map to the same colour form a colour class. A colouring is called a \( k \)-colouring if \( S \) has size \( k \). For a graph \( G \) and two subsets \( U \) and \( U' \) of \( V(G) \), \( U \subseteq U' \), to extend a colouring \( \sigma \) of \( G[U] \) to \( G[U'] \) is to exhibit a colouring \( \sigma' \) of \( G[U'] \) such that \( \sigma(v) = \sigma'(v) \) for all \( v \in U \).

A colouring is \( H \)-free if the subgraph induced by each colour class is \( H \)-
free. An $H$-free colouring of a graph is equivalent to a partition of the graph into $H$-free graphs. A graph is $H$-free $k$-colourable if it admits an $H$-free $k$-colouring. The minimum $k$ for which a graph is $H$-free $k$-colourable is called its $H$-free chromatic number. Following the commonly used terminology, the chromatic number of a graph $G$, written as $\chi(G)$, is the $K_2$-free chromatic number of $G$ and a graph is $k$-colourable means it is $K_2$-free $k$-colourable.

The following theorem is conjectured in [8] and proved in [1].

**Theorem 3.1.** Deciding whether a graph admits a $G$-free $k$-colouring is NP-complete for any fixed $G$ with at least 3 vertices and $k \geq 2$.

### 3.2 Cographs and the cog-chromatic number

The set of all cographs (complement reducible graph) is defined recursively using the following rules:

1. A single vertex ($K_1$) is a cograph.
2. The disjoint union of two cographs is a cograph.
3. The complement of a cograph is a cograph.

Cographs form the minimal family of graphs containing $K_1$ and are closed under complementation and disjoint union. Cographs have arisen in many disparate areas and rediscovered under different names by different researchers [40]. A well-known characterization of cographs is that cographs are exactly the $P_4$-free graphs [51]. The process of deciding whether a graph $G$ is in a set of graphs $\mathcal{G}$ is called recognizing $\mathcal{G}$. There are many algorithms [13, 29] which recognize cographs in linear time.

Here we will study partitioning graphs into cographs, i.e., $P_4$-free colouring of graphs. We call a $P_4$-free colouring a cog-colouring and the $P_4$-free chromatic number as the cog-chromatic number, denoted by $c(G)$. For example, cographs have cog-chromatic number 1 and $P_4$ has cog-chromatic number 2. The cog-chromatic number is studied in [25, 46]\(^\dagger\). A graph $G$ is $k$-cog-colourable if $c(G) \leq k$ and is $k$-cog-chromatic if $c(G) = k$.

In order to study this relatively new graph parameter, $c(G)$, we start with a study of graphs $G$ with $c(G) \leq 2$. These graphs have been called $P_4$-bipartite in the literature [35]. Following the notation of graph editing [9], we define cograph+$kv$ to be the set of graphs that have an induced cograph subgraph obtained by deleting at most $k$ vertices. When $k = 1$, we omit the

\(^\dagger\)The cog-chromatic number is called the $c$-chromatic number in [25]
3.2. Cographs and the cog-chromatic number

$k$ and denote the set as $\text{cograph} + v$. For notational convenience, we call a graph $G$ cograph+$kv$ if $G$ is contained in the set cograph+$kv$.

Weakly chordal graphs (or weakly triangulated graphs) are defined for $k \geq 5$ as $C_k$-free graphs in [32]. Since cographs are $P_4$-free, all graphs in cograph+$v$ must be $C_5$-free. Thus, we have the following remark.

Remark 3.2. cograph+$v$ graphs are weakly chordal.

Remark 3.3. cograph+$v$ graphs can be recognized in linear time.

Proof. Run a linear recognition algorithm for cographs on a graph $G$. If $G$ is a cograph, we recognize $G$ as cograph+$v$ in linear time. If $G$ is not a cograph, the algorithm will output an induced $P_4$, say $v_1v_2v_3v_4$. Then, for each $v_i$ in this $P_4$, we run the algorithm on $G - v_i$. If some run recognizes $G - v_i$ as a cograph, then we recognize $G$ as cograph+$v$ in linear time. Otherwise, none of these four runs recognizes $G - v_i$ as a cograph. In this case, if $u$ is not a vertex from $\{v_1, v_2, v_3, v_4\}$, then $G - u$ will contain $v_1v_2v_3v_4$, a $P_4$. Therefore, $G$ is recognized as not being cograph+$v$ in linear time.

Since cograph+$v$ graphs can be recognized in linear time and cographs are the $P_4$-free graphs, cograph+$v$ might be characterized by a finite set of forbidden subgraphs. That is, there may be a finite collection of graphs, $\mathcal{C}$, such that a graph is cograph+$v$ if and only if it is $G$-free for every graph $G \in \mathcal{C}$. We call a graph non-cograph+$v$ if it is not cograph+$v$. A graph $G$ is vertex-minimal with respect to a graph class $\mathcal{G}$ if $G \in \mathcal{G}$ and $G - v \notin \mathcal{G}$ for every $v \in V(G)$. According to the definition of forbidden subgraph characterizations, $\mathcal{C}$ must include all vertex-minimal non-cograph+$v$ graphs. Observe that the complement of a cograph+$v$ graph is still a cograph+$v$ graph and the complement of a non-cograph+$v$ graph is still a non-cograph+$v$ graph. Thus, a graph is a vertex-minimal non-cograph+$v$ graph if and only if its complement is a vertex-minimal non-cograph+$v$ graph. Therefore, $\mathcal{C}$ is self-complementary, i.e., if $G \in \mathcal{C}$, then $\overline{G} \in \mathcal{C}$. Some graphs in the $\mathcal{C}$ are $C_5, C_6, C_7, C_8, P_8$ and their complements. However, this list is not complete and we have been unable to characterize cograph+$v$ in a nice or uniform way. For example, the graph in Figure 3.1 is a vertex-minimal non-cograph+$v$ graph but it is hard to come up with a unifying description of the characterization.

Remark 3.2 and Remark 3.3 imply that cograph+$v$ graphs are easy to recognize and many problems can be solved efficiently on them. However, problems related with cograph+$kv$ graphs seem hard to attack as made precise in the following remark.
3.2. Cographs and the cog-chromatic number

Remark 3.4 ([55]). Deciding whether a graph is cograph+kv, where k is an input parameter, is NP-complete.

$P_4$-bipartite graphs have intersections with many other graph classes. Intuitively, for a graph $G$, the more complex the way of vertices of $G$ form induced $P_4$s is, the harder it is to compute $c(G)$. The $P_4$-structure of a graph $G$ is the collection of subsets of size 4 of $V(G)$ such that the subgraph induced by each subset is a $P_4$ in $G$. Two graphs $G$ and $H$ are $P_4$-isomorphic if there exists a bijection $\theta : V(G) \rightarrow V(H)$ such that for any four vertices $\{u_1, u_2, u_3, u_4\} \subseteq V(G)$, $u_1u_2u_3u_4$ is an induced $P_4$ of $G$ if and only if $\theta(u_1)\theta(u_2)\theta(u_3)\theta(u_4)$ is an induced $P_4$ of $H$. Two graphs which are not isomorphic may be $P_4$-isomorphic (e.g. a triangle and a $P_3$). $P_4$-isomorphic graphs have the same cog-chromatic number. An interesting result in [33] shows that, given a collection of subsets of four vertices, it is polynomially solvable to decide whether it is equivalent to the $P_4$-structure of some graph $G$.

We now explore relationships between $P_4$-bipartite graphs and other commonly known graph classes. A graph is $P_4$-sparse if every subgraph induced by five vertices contains at most one $P_4$. $P_4$-sparse graphs are shown to be $P_4$-bipartite in [35]. A graph is split if it can be partitioned into a stable set and a clique. A graph is split-perfect if and only if it is $P_4$-isomorphic to a split graph. Since stable sets and cliques are cographs, split graphs and split-perfect graphs are $P_4$-bipartite.

Remark 3.5. $P_4$-sparse graphs and split-perfect graphs are subclasses of $P_4$-bipartite graphs.

As for a superclass of 2-cog-colourable graphs, we have not found a well-known graph class other than the trivial class of $k$-cog-colourable graphs, $k \geq 3$.

Unlike cographs which are easy to recognize and for which many problems admit efficient algorithms, Theorem 3.1 shows that it is NP-hard to determine whether a graph is $P_4$-bipartite. Also, many problems on $P_4$-bipartite graphs are shown to be NP-complete, such as MAXIMUM CLIQUE [47] and HAMILTONIAN CYCLE [43]. In the following, we prove that $k$-colourability is NP-complete for $P_4$-bipartite graphs.
3.2. Cographs and the cog-chromatic number

Theorem 3.6. Let $G$ be an arbitrary $P_4$-bipartite graph. Deciding whether $\chi(G) = 3$ is NP-complete.

Proof. For an arbitrary graph $G$, deciding $\chi(G) = 3$ is proved to be NP-complete in Theorem 8.22 of [38]. The proof there reduces solvability of 3-SAT to 3-colourability of graphs. We show that their reduction from 3-SAT is always $P_4$-bipartite.

We describe the reduction here briefly and describe a bipartition of the graph into cographs. Given an instance of 3-SAT with $n$ variables $\{x_1, \cdots, x_n\}$ and $m$ clauses $\{C_1, \cdots, C_m\}$, a graph with $2n + 6m + 3$ vertices is created. There are $2n$ vertices that correspond to the $2n$ literals. For each clause, six vertices are added. There are three special nodes, “T”, “B”, “F” that indicate three colours. Figure 3.2 shows the two types of gadgets. The true-false gadget on the left side forces each vertex that corresponds to some literal to choose colour either “T” or “F” in any 3-colouring. For each clause $C_i$, add six vertices $C_i^1, \cdots, C_i^6$ and connect them to vertices in the way of the clause-gadget. In the clause-gadget, any 3-colouring must assign colour “T” to at least one of the three vertices correspond to the three literals. Thus, clause $C_i$ is satisfied if and only if the clause-gadget is 3-colourable.

Now we prove that $G$ can be partitioned into 2 cographs. Let $A_1 = \{T, B, F\} \cup \{x_i, \overline{x_i} | 1 \leq i \leq n\} \cup \{C_i^1 | 1 \leq i \leq m\}$ and $A_2 = \{C_i^j | 1 \leq i \leq m, 2 \leq j \leq 6\}$. For any four vertices $U$ in $A_1$, $G[U]$ is either disconnected or $G[U]$ has a triangle. Thus, $G[A_1]$ is a cograph. $G[A_2]$ is a set of disjoint edges and hence is also a cograph. Therefore, $\{A_1, A_2\}$ is a bipartition of $G$ into two cographs.

Figure 3.2: Reduction from 3-SAT to $\chi(G) = 3$
Let $G$ be an arbitrary $P_4$-bipartite graph and $u$ be a vertex not in $V(G)$. Since the join of two cographs is a cograph, we have $c(G \oplus \{u\}) = c(G)$. In any $K_2$-free colouring $\sigma$ of $G \oplus \{u\}$, $\sigma(u) \neq \sigma(v)$ for any $v \in V(G)$. Thus $\chi(G \oplus \{u\}) = \chi(G) + 1$. Therefore, we can reduce the problem of deciding $\chi(G) = k$ for $P_4$-bipartite graphs to problem of deciding $\chi(G) = k + 1$ for $P_4$-bipartite graphs, in polynomial time.

**Corollary 3.7.** Let $G$ be an arbitrary $P_4$-bipartite graph. Deciding whether $\chi(G) = k$ is NP-complete for any fixed $k \geq 3$.

### 3.3 Approximating the cog-chromatic number

As it is NP-hard to find a best cog-colouring for graphs, we turn now to finding approximation algorithms that output near-optimum cog-colourings. The study of approximating the chromatic number, $\chi(G)$, plays a very important role in both graph theory and computational complexity theory. The study gives rise to the famous theorem of “Probabilistically Checkable Proof (PCP)” (see Chapter 17 of [28]). The PCP theorem is the cornerstone of the theory of computational hardness of approximation and gives a finer classification of problems in NP. In this section, we approximate $c(G)$ for $P_4$-bipartite graphs $G$.

#### 3.3.1 Greedy colouring

In the study of the classic graph colouring, greedy colouring using an order of the vertices is studied. Suppose colours are represented by consecutive positive integers. The algorithm colours vertices according to the given order and greedily gives the next vertex $u$ the smallest possible colour that has not been used by vertices which are adjacent to $u$ and precede $u$ in the order. It is reasonable to consider how well this greedy strategy works for approximating the cog-chromatic number of a graph. We give the next vertex in the order the smallest colour such that subgraph induced by each colour class (defined up to this iteration of the algorithm) is a cograph. Since any simple graph $G$ with less than four vertices is a cograph, $c(G) \leq \lceil \frac{n}{3} \rceil$, where $n = |V(G)|$. We construct a $P_4$-bipartite graph below and describe an order that makes the greedy colouring algorithm use $n/3$ colours. This implies that greedy cog-colouring strategy is not a good idea for approximating the cog-chromatic number.

For $k \geq 2$, and $0 \leq i < k$, let $v_{3i}v_{3i+1}v_{3i+2}$ be a $P_3$. Join each vertex of $\{v_{3i}, v_{3i+1}, v_{3i+2}\}$ to each vertex of $\{v_{3j+2} | j < i\}$. Let the resulting graph be
3.3. Approximating the cog-chromatic number

Figure 3.3 shows a construction where we label the vertices with their indices. First, observe that $v_0v_1v_2v_3$ is a $P_4$ in each $G_k$ and hence $c(G_k) \geq 2$. Moreover, note that the subgraph induced by $\{v_{3i+2}|i < k\}$ is complete and hence a cograph, and the subgraph induced by $\{v_{3i}, v_{3i+1}|i < k\}$ is a set of disjoint edges and hence is also a cograph. Therefore, $c(G_k) = 2$.

Remark 3.8. Given $G_k$ and an order of $V(G_k)$ that arranges vertices in an increasing order of their indices. Let $C_k = \{c_0, c_1, \ldots, c_{k-1}\}$ be a set of $k$ colours. For any $k \geq 1$ and $0 \leq i < k$, the greedy cog-colouring colours vertices of $\{v_{3i}, v_{3i+1}, v_{3i+2}\}$ with $c_i$.

Proof. We prove by induction on $k$.

Base Step ($k = 1$): Since the graph induced by $\{v_0, v_1, v_2\}$ is a cograph, they are coloured $c_0$.

Inductive Step: Suppose for some $k \geq 1$, the statement of the remark is true for all $G_j$, $j \leq k$. Consider $G_{k+1}$. Let $u$ be any vertex from $\{v_{3k}, v_{3k+1}, v_{3k+2}\}$. For any $i$, $0 \leq i < k$, since the subgraph induced by $\{v_{3i}, v_{3i+1}, v_{3i+2}, u\}$ is a $P_4$ and each vertex of $\{v_{3i}, v_{3i+1}, v_{3i+2}\}$ is, by induction, coloured $c_i$, $u$ cannot be coloured $c_i$. Thus, the vertices of $\{v_{3k}, v_{3k+1}, v_{3k+2}\}$ are coloured $c_k$ and the statement of the remark is true for $G_{k+1}$.

Remark 3.8 implies that the greedy cog-colouring outputs a $k$-cog-colouring of $G_k$ if the given order is an increasing order of the indices of vertices. Since $G_k$ has $3k$ vertices and is $P_4$-bipartite, the greedy cog-colouring strategy is not a good algorithm for approximating the cog-chromatic number.
3.3. Approximating the cog-chromatic number

3.3.2 Maximum induced cograph

In any bipartition of a $P_4$-bipartite graph with $n$ vertices, the larger part must have at least $n/2$ vertices. Thus, if we could find a maximum induced cograph from a $P_4$-bipartite graph $G$, then we can partition $G$ into at most $\log_2 n$ cographs where $n = |V(G)|$. This is because the number of remaining vertices decreases by at least half, after giving a maximum cograph a new colour. In the following, we study the complexity of finding a maximum induced cograph from $P_4$-bipartite graphs.

A graph $G$ is bipartite if there is a bipartition $\{X,Y\}$ of $G$, written as $G[X,Y]$, such that both $G[X]$ and $G[Y]$ are stable sets. A bipartite graph $G[X,Y]$ is complete bipartite if every vertex in $X$ is connected to every vertex in $Y$. A complete bipartite graph is also known as a biclique. A graph is bipartite if and only if it does not contain any odd cycle [7].

Lemma 3.9. A connected cograph $G$ is $K_3$-free if and only if $G$ is a biclique.

Proof. “Only If”: If $G$ is not bipartite, then the smallest odd cycle of $G$ has length larger than or equal to 5, contradicting the assumption that $G$ is a cograph. Let $[X,Y]$ be any bipartition of $G$ such that $X$ and $Y$ are stable sets. Suppose $G$ is not a biclique. Then there exists a pair of vertices $u$ and $v$, $u \in X$, $v \in Y$, but $uv \notin E(G)$. Let $P$ be the shortest path from $u$ to $v$ and $e = uv$. Since $G$ is a connected cograph, $P$ has length of 2, contradicting the assumption that $G[X,Y]$ is bipartite. Therefore, if a connected cograph $G$ is $K_3$-free, then $G$ is a biclique.

“If”: For any biclique $G$, the length of the shortest path between any two vertices of $G$ is either 1 or 2. Therefore, $G$ does not contain an induced $P_4$ in $G$ and $G$ is a connected cograph. \(\square\)

Finding a biclique with the most number of vertices in a bipartite graph is polynomially solvable as shown in [31]. According to Lemma 3.9, a bipartite graph is a cograph if and only if it is a bicluster, i.e., a disjoint union of bicliques. Therefore, a cograph with the most number of vertices in a bipartite graph $G$ is a set of maximum bicliques from the connected components of $G$.

Corollary 3.10. Finding a maximum induced cograph is polynomially solvable for bipartite graphs.

A stable set $U$ of $G$ is maximum if there is no stable set $U'$ of $G$ such that $|U'| > |U|$. Given a graph $G$ and a positive integer $k$, the Stable Set problem asks whether $G$ has a stable set of size $k$. The Stable Set problem
3.3. Approximating the cog-chromatic number

is one of Karp’s 21 NP-complete problems [36]. The following hardness result is implied by a theorem in [55]. The proof given in [55] is for a much more general result and is very involved. We prove it for our specific case in a simpler way.

**Lemma 3.11.** Finding a maximum induced cograph is NP-hard for general graphs.

**Proof.** We prove that deciding whether a graph has a cograph of size $k$ (the **Cograph** problem) is NP-complete, via a reduction from the **Stable Set** problem. It takes linear time to check whether a subgraph induced by $k$ vertices is a cograph [29]. Thus, the **Cograph** problem is in NP. Given a graph $G$ with $n$ vertices, we create a new graph $G'$ with $5n$ vertices as following. Let $G'$ be a copy of $G$. For each $v \in V(G)$, denote by $v'$ the corresponding vertex of $v$ in $G'$. For each vertex of $v \in V(G)$, create a join between $v'$ and a disjoint copy of $P_4$. Figure 3.4 shows an example of this construction on a triangle. We claim that $G$ has a stable set of size $k$ if and only if $G'$ has a cograph of size $3n + k$. Denote by $P_4(v')$ the $P_4$ joined to $v'$.

“Only If”: Suppose $U \subseteq V(G)$ is a stable set of $G$ with $k$ vertices, let $A = \{v'|v \in U\} \cup \{\text{three vertices which form a } P_3 \text{ in } P_4(v')|v \in V(G)\}$. Each component of $G'[A]$ is a join between $K_1$ and $P_3$. Thus, $G'[A]$ is a cograph with $3n + k$ vertices.

“If”: Suppose the subgraph induced by $U \subseteq V(G')$ is a cograph of size $3n + k$. Let $U = U_1 \cup U_2$, where $U_1$ is a subset of vertices which correspond to vertices of $G$ and $U_2$ is the subset of vertices from added $P_4$s. We perform a case analysis according to $|U_1|$.

Case 1: $|U_1| = k$.

Since $|U_2| = 3n$ and $G'[U_2]$ is a cograph, $U_2$ has exactly 3 vertices from each added $P_4$. If $U_1$ is not a stable set, then there is a $P_4$, $v_1v_2v_3v_4$ in

![Figure 3.4: Reduction from Stable Set to Cograph problem](image-url)
3.3. Approximating the cog-chromatic number

$G'$, where $v_2v_3$ is an edge in $G'[U_1]$, $v_1$ and $v_4$ are vertices in $P_4(v_2)$ and $P_4(v_3)$ respectively, contradicting the assumption that $G'[U]$ is a cograph. Therefore, $\{v | v' \in U_1\}$ is a stable set of $G$ with size $k$.

Case 2: $|U_1| > k$.

Since $|U_2| < 3n$, there is a vertex $v'$ that $|U_2 \cap P_4(v')| = l$, $l \leq 2$. The following vertex exchange operation of adding vertices to $U_2$ and removing vertices from $U_1$ reduces the size of $U_1$ by one while preserves the properties that $|U| = 3n + k$ and $G'[U]$ is a cograph. When $v' \in U_1$, the exchange adds $3 - l$ vertices from $P_4(v')$ to $U_2$ and removes $v'$ and other arbitrary $2 - l$ vertices from $U_1$. When $v' \notin U_1$, the exchange adds $3 - l$ vertices from $P_4(v')$ to $U_2$ and removes other arbitrary $3 - l$ vertices from $U_1$. After $|U_1| - k$ vertex change operations, $|U_1| = k$ and we are in case 1.

Lemma 3.11 shows the complexity of the maximum induced cograph problem for general graphs. Note that there are many graphs which are not $P_4$-bipartite. We have the following conjecture.

Conjecture 3.12. Finding a maximum induced cograph is NP-hard for $P_4$-bipartite graphs.

3.3.3 Approximation algorithms

The degree of a vertex $v$ in a simple graph $G$, denote by $d_G(v)$, is the size of its neighbourhood $N_G(v)$. Denote by $\Delta(G)$ the maximum degree of vertices of $G$. When $G$ is clear from the context, we simplify notations of $d_G(v)$ and $\Delta(G)$ as $d(v)$ and $\Delta$ respectively. A $P_3$-free colouring is trivially a $P_4$-free colouring. Denote by $p_3(G)$ the $P_3$-free chromatic number of a graph $G$. The proof technique in the following remark has been used many times [19, 25].

Remark 3.13. For any graph $G$, $p_3(G) \leq \left\lceil \frac{\Delta + 1}{2} \right\rceil$.

Proof. Consider a colouring $\sigma$ of $G$ using $k = \left\lceil \frac{\Delta + 1}{2} \right\rceil$ colours that minimizes the number of monochromatic edges. We claim that $\sigma$ is a $P_3$-free colouring. Suppose there is a monochromatic $P_3$, $v_1uv_2$, that is coloured all red. Since $d_G(u) < 2k$, there is at least one other colour, say blue, which is used at most once by vertices which are adjacent to $u$. Switch $u$’s colour from red to blue and keep colouring of other vertices unchanged. After switching $u$’s colour, we decrease the number of monochromatic edges by at least one, contradicting the assumption that $\sigma$ minimizes the number of monochromatic edges.

Proof. Consider a colouring $\sigma$ of $G$ using $k = \left\lceil \frac{\Delta + 1}{2} \right\rceil$ colours that minimizes the number of monochromatic edges. We claim that $\sigma$ is a $P_3$-free colouring. Suppose there is a monochromatic $P_3$, $v_1uv_2$, that is coloured all red. Since $d_G(u) < 2k$, there is at least one other colour, say blue, which is used at most once by vertices which are adjacent to $u$. Switch $u$’s colour from red to blue and keep colouring of other vertices unchanged. After switching $u$’s colour, we decrease the number of monochromatic edges by at least one, contradicting the assumption that $\sigma$ minimizes the number of monochromatic edges.
3.4 Minimum order of \( k \)-cog-chromatic graphs

Algorithm 1 is an implementation of the proof of Remark 3.13. Let \( n = |V(G)| \) and \( m = |E(G)| \). Lines from 3 to 4 are executed at most \( m \) times since each iteration decreases the number of monochromatic edges. Line 3 checks at most \( \Delta \) vertices for each colour and line 2 needs \( O(n^3) \) time to enumerate all \( P_3 \)s if \( G \) is represented by an adjacency matrix. Thus, the algorithm finds a \( P_3 \)-free colouring of a graph in \( O(mn^3\Delta^2) \) time.

**Algorithm 1** Partitioning \( G \) into \( P_3 \)-free graphs

1. Arbitrarily colour \( G \) using \( k = \left\lceil \frac{\Delta+1}{2} \right\rceil \) colours
2. while There is a monochromatic \( P_3 \), say \( v_1uv_2 \) do
3.    Find a colour \( x \) used by at most one vertex in \( u \)'s neighborhood
4.    Let \( x \) be the new colour of \( u \)

**Theorem 3.14.** A \( P_4 \)-bipartite graph with maximum degree \( \Delta \) can be partitioned into \( \left\lceil \frac{\Delta+1}{2} \right\rceil \) cographs in polynomial time.

A hypergraph is an ordered pair \((V, \mathcal{F})\), where \( V \) is a set of vertices and \( \mathcal{F} \) is a family of subsets of \( V \). Each subset in \( \mathcal{F} \) is called a hyperedge or an edge of the hypergraph. A hypergraph is \( k \)-uniform if each of its hyperedges has \( k \) vertices. A proper colouring of a hypergraph is a colouring of \( V \) such that there is no monochromatic hyperedge. A hypergraph is \( k \)-colourable if it admits a proper colouring which uses \( k \) colours. Thus, a cog-colouring of a graph \( G(V, E) \) into cographs is a proper colouring of a 4-uniform hypergraph \( H(V, \mathcal{F}) \), where the subgraph induced by each hyperedge of \( \mathcal{F} \) is a \( P_4 \) in \( G \). \( G \) is \( P_4 \)-bipartite if and only if \( H \) is 2-colourable. A polynomial algorithm that colours a 2-colourable hypergraph using \( O(n^{3/4}\log^{3/4} n) \) colours is described in [37], where \( n \) is the number of vertices. Thus, a \( P_4 \)-bipartite graph with \( n \) vertices can be partitioned into \( O(n^{3/4}\log^{3/4} n) \) cographs in polynomial time.

### 3.4 Minimum order of \( k \)-cog-chromatic graphs

In this section, we study the minimum number of vertices a \( k \)-cog-chromatic graphs must have. The following way of constructing a \((k+1)\)-cog-chromatic graph from a \( k \)-cog-chromatic graph is given in [25].

**Definition 3.15** (3+1 construction). Given a graph \( G \), let \( G_1, G_2, G_3 \) be three disjoint copies of \( G \). The 3+1 construction \( T(G) \) from \( G \) is the union of \( G_1 \oplus G_2, G_2 \oplus G_3 \) and \( G_3 \oplus K_1 \).
Lemma 3.16 (Remark 9 in [25]). \( c(T(G)) = c(G) + 1 \).

Proof. Let \( T(G) = (G_1 \oplus G_2) \cup (G_2 \oplus G_3) \cup (G_3 \oplus \{x\}) \), where \( x \) corresponds to the vertex of \( K_1 \) in the definition of the 3+1 construction. Let \( k = c(G) \) and \( C_i = \{c_1, \ldots, c_l\} \) be a set of \( l \) colours for any \( l \geq 1 \). We categorize the induced \( P_4 \)'s contained in \( T(G) \) into two types.

1. Type-1 \( P_4 \): The four vertices are contained in \( G_i \), for some \( 1 \leq i \leq 3 \).

2. Type-2 \( P_4 \): The \( P_4 \) is \( u_1 u_2 u_3 x \), where \( u_i \in V(G_i) \) for \( 1 \leq i \leq 3 \).

Note that there are no other types of \( P_4 \)'s in \( T(G) \) since the subgraph induced by any set of four vertices of \( T(G) \) that includes at least two vertices from one \( G_i \) and a vertex not in \( G_i \) either contains a triangle or is disconnected. We first prove that \( c(T(G)) \leq k + 1 \). Let \( \sigma \) be a \( k \)-colouring which cog-colours \( G_1, G_2 \) and \( G_3 \) using colours from \( C_k \). Extend \( \sigma \) to colour \( x \) with \( c_{k+1} \). Neither a Type-1 \( P_4 \) nor a Type-2 \( P_4 \) is monochromatic under \( \sigma \). Thus, \( \sigma \) is a \((k + 1)\)-cog-colouring of \( T(G) \).

We now prove that \( c(T(G)) > k \). Suppose there is a \( k \)-cog-colouring \( \sigma \) of \( T(G) \) using colours from \( C_k \). For each \( 1 \leq i \leq 3 \), since \( c(G_i) = k \), the colour \( \sigma(x) \) must be used by some vertex \( u_i \) of \( G_i \). Thus, \( u_1 u_2 u_3 x \) is a monochromatic induced \( P_4 \), contradicting the assumption that \( \sigma \) is a cog-colouring of \( T(G) \).

Therefore, \( c(T(G)) = k + 1 \). \( \square \)

Let \( \mathcal{P} \) be a graph property. A graph \( G \) is minimal with respect to \( \mathcal{P} \) if \( G \) has property \( \mathcal{P} \) and if any proper subgraph of \( G \) does not have property \( \mathcal{P} \). \( G \) is minimum with respect to \( \mathcal{P} \) if \( G \) has property \( \mathcal{P} \) and if any graph \( G' \) with less than \( |V(G)| \) vertices does not have property \( \mathcal{P} \).

Let \( f_c(k) \) be the order of the minimum \( k \)-cog-chromatic graphs. It is clear that \( f_c(1) = 1 \) and \( f_c(2) = 4 \), because any graph with at most 3 vertices is a cograph and \( P_4 \) is 2-cog-chromatic. Figure 3.5 shows a graph \( J \) with 10 vertices that is 3-cog-chromatic (this graph comes from [25]). The authors conjecture that \( J \) is a minimum 3-cog-chromatic graph. That is, \( f_c(3) \leq 10 \).

In order to check whether \( f_c(3) = 10 \), we use “\textit{geng}”, a computer tool, described in [41], to generate all possible graphs of 9 vertices. Indeed, every graph with 9 vertices can be partitioned into 2 cographs. Thus, we confirm that \( f_c(3) = 10 \). Starting with graph \( J \) in Figure 3.5, for any \( k > 3 \), one obtains a \( k \)-cog-chromatic graph after \( k - 3 \) iterations of applying the 3+1 construction to the resulting graph of the previous iteration. The resulting graph has \((10 \cdot 3^{k-3} + (3^{k-3} - 1)/2)\) vertices. Thus, \( f_c(k) = O(3^k) \).
3.4. Minimum order of \( k \)-cog-chromatic graphs

![Figure 3.5: A graph \( J \) of \( c(J) = 3 \) with 10 vertices](image)

The construction in the following lemma uses fewer vertices than the one of the 3+1 construction in Lemma 3.16.

**Lemma 3.17.** Let \( G_1, G_2, G_3, G_4 \) be four disjoint graphs, \( H \) be the union of \( G_1 \oplus G_2 \), \( G_2 \oplus G_3 \) and \( G_3 \oplus G_4 \). For \( k \geq 1 \), if \( c(G_1) = c(G_4) = k \) and \( c(G_2) + c(G_3) = k + 1 \), then \( c(H) = k + 1 \).

**Proof.** Let \( C_l = \{c_1, \ldots, c_l\} \) be a set of \( l \) colours for any \( l \geq 1 \).

First, we prove that \( c(H) \leq k + 1 \). Let \( \sigma \) be a cog-colouring of \( G_1 \) and \( G_4 \) using colours from \( C_k \). We extend \( \sigma \) to cog-colour \( G_2 \) using colours \( \{c_1, \ldots, c_x\} \) where \( x = c(G_2) \) and to cog-colour \( G_3 \) using colours \( \{c_{x+1}, \ldots, c_{k+1}\} \). The only possible induced \( P_4 \) whose vertices might be given the same colour must have one vertex from \( V(G_1) \), \( V(G_2) \), \( V(G_3) \) and \( V(G_4) \) respectively. However, such a path cannot be monochromatic because vertices in \( G_2 \) and \( G_3 \) do not have a common colour. Therefore, \( \sigma \) is a cog-colouring of \( H \) and \( c(H) = k + 1 \).

Second, we prove that \( c(H) \geq k \). Suppose there is a \( k \)-cog-colouring \( \sigma \) of \( H \) using colours from \( C_k \). Since \( c(G_2) + c(G_3) > k \), the Pigeon-Hole Principle guarantees some colour \( c_i \) must be used by some vertex from \( V(G_2) \) and some vertex from \( V(G_3) \). Since \( c(G_1) = c(G_4) = k \), \( c_i \) must also be used by some vertex from \( V(G_1) \) and some vertex from \( V(G_4) \). Thus, we have a monochromatic \( P_4 \) of colour \( c_i \), contradicting the assumption that \( \sigma \) is a cog-colouring.

Therefore, \( c(H) = k + 1 \). \( \Box \)

The following lemma is used to characterize how the number of vertices
of graphs increases when we use the construction introduced in Lemma 3.17.

**Lemma 3.18.** Let \( f : \mathbb{N}^+ \to \mathbb{N}^+ \), be such that \( f(1) < f(2) \). If \( f(n) \) satisfies the following recurrence equation

\[
f(n) = 2 \cdot f(n - 1) + 2 \cdot f(\lceil n/2 \rceil + 1), \quad n \geq 3,
\]

then \( f(n) = O(2^n) \).

**Proof.** Since \( f(n) > 0 \) for all \( n \), \( f(n) > f(n - 1) \) and hence \( f \) is strictly increasing. Let \( m = \lceil n/2 \rceil + 1 \) and let \( g(n) = \frac{f(n)}{2^n} \), then for \( n \geq 4 \),

\[
g(n) = g(n - 1) + \left(\frac{1}{2}\right)^{n-m-1} \cdot g(m) < \left(1 + \left(\frac{1}{2}\right)^{n/2-2}\right) \cdot g(n - 1).
\]

Taking logarithms on both sides, we have,

\[
\ln g(n) < \ln g(n - 1) + \ln \left(1 + \left(\frac{1}{2}\right)^{n/2-2}\right) < \ln g(n - 1) + \left(\frac{1}{2}\right)^{n/2-2}
\]

\[
< \ln g(3) + 2 \cdot \sum_{i=0}^{n} \left(\frac{1}{2}\right)^i = O(1).
\]

Thus, \( g(n) = O(1) \) and \( f(n) = O(2^n) \).

**Theorem 3.19.** For all \( k \geq 1 \), the order of minimum \( k \)-cog-chromatic graphs is \( O(2^k) \), that is, \( f_c(k) = O(2^k) \).

**Proof.** According to Lemma 3.17, for \( k \geq 2 \),

\[
f_c(k) \leq \min_{x+y=k} \{2 \cdot f_c(k - 1) + f_c(x) + f_c(y)\}.
\]

When we choose \( x, y \) in such a way that \( |x - y| \leq 1 \), we have

\[
f_c(k) \leq \begin{cases} 2 \cdot f_c(k - 1) + f_c(\lceil k/2 \rceil) + f_c(\lfloor k/2 \rfloor) & \text{when } k \text{ is odd} \\ 2 \cdot f_c(k - 1) + 2 \cdot f_c(k/2) & \text{when } k \text{ is even} \end{cases}
\]

Defining \( f \) as in Lemma 3.18 with \( f(i) = f_c(i), \quad i = 1, 2 \), we have \( f_c(k) \leq f(k) \) and therefore \( f_c(k) = O(2^k) \).

The construction in Lemma 3.17 does not give a tight upper bound for \( f_c(k) \). For example, the construction uses 13 vertices to construct a 3-cog-chromatic graph from three 2-cog-chromatic graphs and a single vertex. However, as Figure 3.5 shows, \( f_c(3) \leq 10 \). There may be a general
construction, different from but like the one in Lemma 3.17, that for each 
k ≥ 1, uses \( f_c(k) \) vertices. On the other hand, not all extremal graph func-
tions lend themselves to such unified constructions for their corresponding
extremal graphs. The functions may be quite different when constrained to
small inputs as compared to their asymptotic behaviour. Despite this, the
view of the graph \( J \) in Figure 3.6 shows we can construct a 3-cog-chromatic
graph from two \( P_4 \)s and two vertices. If we could construct a \((k + 1)\)-cog-
chromatic graph from two \( k \)-cog-chromatic graphs and two vertices, we have
\( f_c(k + 1) \leq 2f_c(k) + 2 \). This view motivates the following conjecture.

Conjecture 3.20. For \( k \geq 1, f_c(k) = 2^k + 2^{k-1} - 2 \).

3.5 Cog-critical graphs

In the study of classic graph colouring, a graph \( G \) is colour-critical if
\( \chi(H) < \chi(G) \) for any subgraph \( H \) of \( G \). Similarly, we call a graph \( G \)
cog-critical if \( c(H) < c(G) \) for any induced subgraph \( H \) of \( G \). A graph
\( G \) is \( k \)-cog-critical if \( G \) is cog-critical and \( c(G) = k \). We consider induced
subgraphs instead of subgraphs when defining cog-critical graphs because
deleting an edge from a graph may increase its cog-chromatic number. For
example, deleting any edge from a \( C_4 \) results in a \( P_4 \), while \( c(P_4) > c(C_4) \).

3.5.1 Properties of cog-critical graphs

We study several necessary properties of cog-critical graphs.

Remark 3.21. A cog-critical graph cannot be a join of two graphs.

Proof. Suppose there is a cog-critical graph \( H = G_1 \uplus G_2 \). Since the join
of any number of graphs is a cograph if and only if the graphs them-
3.5. Cog-critical graphs

selves are cographs, we have \( c(H) \leq \max\{c(G_1), c(G_2)\} \) by joining pairs of colour classes from cog-colourings of \( G_1 \) and \( G_2 \). Moreover, \( c(H) \geq \max\{c(G_1), c(G_2)\} \) since any cog-colouring of \( H \) induces a cog-colouring of \( c(G_i), i = 1, 2 \). Thus, \( c(H) = \max\{c(G_1), c(G_2)\} \). Suppose \( c(G_1) \geq c(G_2) \). For any vertex \( v \) in \( G_2 \), \( H - v \cong G_1 \oplus (G_2 - v) \) and hence

\[
c(H - v) = \max\{c(G_1), c(G_2 - v)\} = c(G_1) = c(H),
\]

contradicting the assumption that \( H \) is cog-critical. \( \square \)

A module [30] of a graph \( G(V, E) \) is a subset \( X \subseteq V \) satisfying that for any vertex \( v \in V \setminus X \), either \( v \) is adjacent to every vertex in \( X \) or \( v \) is not adjacent to any vertex in \( X \). That is, vertices in \( X \) have the same neighbourhood outside \( X \). \( V(G), \emptyset \), and a single vertex are trivial modules of \( G \). Modules which are not trivial are called nontrivial modules.

Given a graph \( G \) and \( U \subseteq V(G) \), to shrink \( U \) is to replace \( U \) by a single vertex and make it adjacent to all the vertices which were adjacent in \( G \) to at least one vertex in \( U \). The resulting graph is denoted by \( G/U \). Let \( u \) be the vertex in \( G/U \) that corresponds to the \( U \) being shrunk and \( v \) be an arbitrary vertex in \( U \). According to these definitions, if \( U \) is a nontrivial module of \( G \), \( u \) is connected to the same vertices in \( G/U \) as \( v \) is in \( G - (U \setminus v) \). Thus, for any module \( U \) of \( G \) and any vertex \( v \in U, G/U \cong G - (U \setminus v) \).

**Remark 3.22.** Every nontrivial module \( M \) of a cog-critical graph \( G \) satisfies \( c(G/M) < c(G) \).

**Proof.** Let \( v \) be an arbitrary vertex in \( M \). Since \( G/M \cong G - (M \setminus \{v\}) \), \( c(G/M) = c(G - (M \setminus \{v\})) < c(G) \). \( \square \)

**Lemma 3.23.** If \( M \) is a module of \( G \), then \( c(G) \leq c(G/M)) + c(G[M]) - 1 \).

**Proof.** Let \( M = V(G) \setminus M \). For any vertex \( u \in M \), since

\[
G[M \cup \{u\}] = G - (M \setminus \{u\}) \cong G/M,
\]

there is a cog-colouring \( \sigma \) of \( G[M \cup \{u\}] \) using \( c(G/M) \) colours. Suppose \( u \) is coloured red by \( \sigma \). By extending \( \sigma \) to colour \( G \) such that \( M \) are coloured all red, we get a colouring \( \sigma' \) where all possible monochromatic \( P_4 \)s are subgraphs of \( G[M] \). We then use \( c(G[M]) - 1 \) new colours not used by \( \sigma' \) and red to cog-colour \( M \). The resulting colouring is a cog-colouring of \( G \). Therefore, \( c(G) \leq c(G/M)) + c(G[M]) - 1 \). \( \square \)

**Corollary 3.24.** A cog-critical graph \( G \) cannot have a nontrivial module \( M \) such that \( G[M] \) is a cograph.
3.5. Cog-critical graphs

A restatement of Corollary 3.24 is that no module of a cog-critical graph can have two vertices. Thus, for any two vertices $u, v$ of a cog-critical graph $G$, there must be another vertex $x$ of $G$ such that $x$ is adjacent to exactly one of $uv$.

**Lemma 3.25.** Let $M$ be a nontrivial module of $G$ and let $d(M)$ be the number of vertices not in $M$ but adjacent to vertices in $M$. If $c(G[M]) + d(M) \leq c(G/M)$, then $c(G) = c(G/M)$.

**Proof.** Let $N(M) = \bigcap_{v \in M} N(v)$. Since $M$ is a module, every vertex not in $N(M)$ is either in $M$ or not adjacent to any vertex in $M$. Consider a cog-colouring $\sigma$ of $G/M$. As in the proof of Lemma 3.23, $\sigma$ can be extended to a cog-colouring of $G$ by introducing $c(G[M]) - 1$ more colours. Suppose $\sigma$ colours $N(M)$ with $x$ colours. If $x + c(G[M]) \leq c(G/M)$, then we can reuse colours that $\sigma$ used to colour vertices in $G/M - N(M)$. This is possible because every vertex in $G/M - N(M)$ is not adjacent to any vertex in $M$. Thus, we do not need to introduce any new colours and hence $c(G) \leq c(G/M)$. Note that $x \leq d(M)$. Therefore, if $c(G[M]) + d(M) \leq c(G/M)$, then $x + c(G[M]) \leq c(G/M)$ and $c(G) \leq c(G/M)$. Finally $c(G) = c(G/M)$ because $c(G) \geq c(G/M)$.

**Corollary 3.26.** A cog-critical graph $G$ cannot have a nontrivial module $M$ such that $c(G[M]) + d(M) \leq c(G/M)$.

3.5.2 Arbitrarily large $k$-cog-critical graphs

According to the definition of cog-critical graphs, $K_1$ is the only 1-cog-critical graph and $P_4$ is the only 2-cog-critical graph. In this section, for any fixed $k \geq 3$, we describe $k$-cog-critical graphs with an arbitrarily large number of vertices. The constructions in Lemma 3.16 and Lemma 3.17 both generate cog-critical graphs when the parts of the construction are cog-critical graphs. Thus, we can use them to create cog-critical graphs with arbitrary number of vertices. However, both constructions increase the cog-chromatic number by one while increasing the number of vertices by two or three times. Therefore, both constructions do not provide arbitrarily large $k$-cog-critical graphs with $k$ fixed. In this section, we describe two different constructions that complete this task. Moreover, we provide a way to construct arbitrarily large $k$-cog-critical planar graphs where $k = 3$ and 4.

**Definition 3.27 (J-construction).** Let $F$ be a graph with $n$ vertices $\{v_1, \cdots, v_n\}$, and $\mathcal{H} = \{H_1, \cdots, H_n\}$ be a set of disjoint graphs. Define $G$ to be
3.5. Cog-critical graphs

the graph with \( V(G) = V(F) \cup \bigcup_{i=1}^{n} V(H_i) \) and edges \( E(G) = E(F) \cup (\bigcup_{i=1}^{n} E(H_i \oplus \{v_i\})) \). We call \( G \) the J-construction from pair \((F, \mathcal{H})\), \( F \) the inner graph of \( G \), and \( \mathcal{H} \) the set of outer graphs of \( G \).

In Lemma 3.11, the gadget (Figure 3.4) used in the polynomial reduction is a J-construction where the outer graphs are \( P_4 \)s.

**Lemma 3.28.** Let \( F \) be a \((k + 1)\)-colour-critical graph and each \( H_i \in \mathcal{H} \) be a \( k \)-cog-critical graph. The J-construction \( G \) from pair \((F, \mathcal{H})\) is a \((k + 1)\)-cog-critical graph.

**Proof.** We first categorize induced \( P_4 \)s in \( G \) into two types.

1. Type-1 \( P_4 \): The four vertices of the \( P_4 \) are contained in \( H_i \), for some \( 1 \leq i \leq n \).
2. Type-2 \( P_4 \): The \( P_4 \) contains some edge of \( F \).

Let \( C_l = \{c_1, \cdots, c_l\} \) be a set of \( l \) colours for \( l \geq 1 \). We complete the proof in four claims.

**Claim 1:** \( c(G) \leq k + 1 \).

Let \( \sigma \) be a \( K_2 \)-free colouring of \( F \) using colours from \( C_{k+1} \). Extend \( \sigma \) to all of \( G \) by obtaining a \( k \)-cog-colouring of \( H_i \) that uses colours \( C_k \), for each \( 1 \leq i \leq n \). Neither a Type-1 \( P_4 \) nor a Type-2 \( P_4 \) is monochromatic under \( \sigma \), because each \( H_i \) is cog-coloured and there is no monochromatic edge in \( F \). Thus, \( \sigma \) is a \((k + 1)\)-cog-colouring of \( G \) and Claim 1 is proved.

**Claim 2:** \( c(G) > k \).

Suppose \( \sigma \) is a \( k \)-cog-colouring of \( G \) using colours from \( C_k \). Since \( F \) is not \( k \)-colourable, there is a monochromatic edge \( u_iv_j \) in \( E(F) \). Suppose \( u_i, v_j \) are coloured \( c_1 \). Since \( c(H_i) = c(H_j) = k \), some vertex \( u_i \) of \( H_i \) as well as some vertex \( u_j \) of \( H_j \) are coloured \( c_1 \). Thus, \( u_iv_iu_jv_j \) is a monochromatic induced \( P_4 \), contradicting the assumption that \( \sigma \) is a cog-colouring. Therefore, \( G \) is not \( k \)-cog-colourable and Claim 2 is proved.

**Claim 3:** For any vertex \( v \) of \( F \), \( c(G - v) \leq k \).

Since \( F \) is \((k + 1)\)-colour-critical, there is a \( K_2 \)-free colouring \( \sigma \) of \( F - v \). Extend \( \sigma \) with any \( k \)-cog-colouring of \( H_i \) using colours \( C_k \) for \( 1 \leq i \leq n \). Neither a Type-1 \( P_4 \) nor a Type-2 \( P_4 \) is monochromatic under \( \sigma \), because each \( H_i \) is cog-coloured and there is no monochromatic edge in \( F \). Thus, \( \sigma \) is a \( k \)-cog-colouring of \( G - v \), and Claim 3 is proved.

**Claim 4:** For any vertex \( v \) in \( \mathcal{H} \), \( c(G - v) \leq k \).

Since \( v \) is a vertex in \( \mathcal{H} \), \( v \) is a vertex of \( H_i \), for some \( 1 \leq i \leq n \). Since \( F \) is \((k + 1)\)-colour-critical, \( k \geq 1 \), \( v_i \) is not an isolated vertex in \( F \). Let
3.5. Cog-critical graphs

$v_j$ be an arbitrary vertex which is adjacent to $v_i$ in $F$. Let $e = v_iv_j$, and let $\sigma$ be a $K_2$-free colouring of $F - e$ using colours from $C_k$. Note that $e$ is monochromatic under $\sigma$ since otherwise $F$ is $k$-colourable. Suppose the colour of $v_i$ is $c_1$. We then extend $\sigma$ to cog-colour $H_l$ with colours from $C_k$, $l \neq i$. Since $H_l$ is $k$-cog-critical, we can further extend $\sigma$ to all of $G - v$, with any $(k-1)$-cog-colouring of $H_l - v$ using the $k-1$ colours in $C_k \setminus \{c_1\}$. Then every Type-2 $P_4$ containing $v_iv_j$ cannot be monochromatic under $\sigma$, because none of $N_{G}(v_i) \setminus \{v_j\}$ is coloured $c_1$, none of $N_{F}(v_j) \setminus \{v_i\}$ is coloured $c_1$ and this $P_4$ can have at most one vertex from $H_j$. Every other Type-2 $P_4$ contains an edge in $F$ with ends of different colours and thus is not monochromatic. Moreover, there is no monochromatic Type-1 $P_4$ because every $H_j$ is cog-coloured, $j \neq i$, and $H_j - v$ is cog-coloured. Thus, $\sigma$ is a $k$-cog-colouring of $G$ and Claim 4 is proved.

By Claims 1 through 4, $G$ is a $(k+1)$-cog-critical graph. 

**Theorem 3.29** (Theorem 5 in [52]). There exists a $k$-colour-critical graph, $k \geq 4$, with $n$ vertices if and only if $n \geq k$ and $n \neq k+1$. There exists a $k$-colour-critical $r$-uniform hypergraph, $k \geq 3, r \geq 3$, with $n$ vertices if and only if $n \geq (r-1)(k-1) + 1$.

**Theorem 3.30.** For any $k \geq 3$ and $n > 0$, there exists a $k$-cog-critical graph $G$ with more than $n$ vertices.

**Proof.** We prove by induction on $k$.

Base Step ($k = 3$): Let $F$ be a cycle of $2n+1$ vertices and $\mathcal{H}$ be a set of $2n+1$ disjoint $P_4$s. Since $F$ is 3-colour-critical and $P_4$ is 2-cog-critical, according to Lemma 3.28, the J-construction $G$ from $(F, \mathcal{H})$ is a 3-cog-critical graph with more than $n$ vertices.

Inductive Step: Suppose the statement of the theorem is true for some $k$, $k \geq 3$, and we consider the case $k+1$. According to Theorem 3.29, there is a $(k+1)$-colour-critical graph $F$ with max$\{n, k+2\}$ vertices. Also, according to the inductive hypothesis, there is a $k$-cog-critical graph $H$ with more than $n$ vertices. Let $\mathcal{H}$ be a set of $|V(F)|$ disjoint copies of $H$. According to Lemma 3.28, the J-construction $G$ from $(F, \mathcal{H})$ is a $(k+1)$-cog-critical graph with more than $n$ vertices.

A graph is planar if it can be drawn in the plane so that its edges intersect only at their ends. Such a drawing is called a planar embedding of the graph. A graph is outerplanar if it has a planar embedding in which all vertices lie on the boundary of its outer face. If $G$ is outerplanar, then $G \oplus K_1$ is planar.
Corollary 3.31. For \( k = 3 \) and \( 4 \), and any \( n > 0 \), there is a planar \( k \)-cog-critical graph with more than \( n \) vertices.

Proof. Let \( F \) be a cycle of \( 2n + 1 \) vertices and \( H \) be a set of \( 2n + 1 \) disjoint \( P_3 \)s. As in the proof of Theorem 3.30, the J-construction \( G \) from \((F, H)\) is 3-cog-critical. Note that, \( G \) is also outerplanar (for \( n = 1 \), see \( G' \) in Figure 3.4). Thus, \( G \) is a 3-cog-critical planar graph with more than \( n \) vertices.

Let \( G \) be a set of four disjoint copies of the graph \( G \) in the above paragraph. Since \( G \) is outerplanar and \( K_4 \) is planar, the J-construction \( G' \) from \((K_4, G)\) is planar. Moreover, \( K_4 \) is 4-colour-critical and \( G \) is 3-cog-critical. According to Lemma 3.28, \( G' \) is 4-cog-critical. Thus, \( G' \) is a 4-cog-critical planar graph with more than \( n \) vertices.

Since planar graphs are 4-colourable, they are 4-cog-colourable. Thus, there does not exist a 5-cog-critical planar graph. The existence of 4-cog-chromatic planar graphs is proved in [25]. Corollary 3.31 provides a construction of 4-cog-chromatic planar graphs which are also 4-cog-critical. Koester graph [39] is a 4-regular 4-colour-critical planar graph. However, according to Remark 3.13, graphs with maximum degree 4 are 3-cog-colourable and hence there is no 4-regular 4-cog-critical planar graph. It is interesting to ask whether there is a 4-cog-critical graph with minimum degree 4.

The girth of a graph \( G \) with at least one cycle is the length of a shortest cycle in \( G \). A graph is triangle-free if it does not contain a cycle or its girth is larger than 3. Planar graphs with girth at least 11 are shown to be 2-cog-colourable [25]. Moreover, it is NP-complete to decide whether a triangle-free planar graph is 2-cog-colourable [16]. Thus, it is interesting to ask whether planar graphs with girth at least 5 are 2-cog-colourable.

Theorem 3.30 relies on the J-construction and the J-construction relies on the fact there is an arbitrarily large \( k \)-colour-critical graphs. In the following, we describe a new construction that does not rely on the existence of arbitrarily large \( k \)-colour-critical graphs, but instead relies on the fact that there are arbitrarily large \( k \)-colour-critical hypergraphs, as proved by Theorem 3.29.

Definition 3.32 (D-construction). Let \( H \) be a graph on \( n \) vertices with \( c(H) = k \). Let \( F \) be an \( n \)-uniform \((k + 1)\)-colour-critical hypergraph. For each edge \( F \in \mathcal{F} \), let \( H_F^1, H_F^2, H_F^3 \) be three disjoint copies of \( H \) and define \( H_F = (H_F^1 \oplus H_F^2) \cup (H_F^2 \oplus H_F^3) \). Let \( G^* = \bigcup_{A, B \in \mathcal{F}} (H_A \oplus H_B) \). For each edge \( F \in \mathcal{F} \), let \( M_F \) be a matching between the vertices of \( F \) and \( V(H_F^1) \). Let \( G \) be a graph defined by \( V(G) = V(\mathcal{F}) \cup V(G^*) \) and \( E(G) = E(G^*) \cup \)
We call $G$ the $D$-construction from $F$ and $H$, denoted by $D(F, H)$. For each edge $F \in F$, we call $H_F$ the satellite graph of $F$.

We will choose the graph $H$ in our $D$-construction so that proper $k$-colourings of subhypergraphs of $F$ can be extended to $(k+1)$-cog-colourings of $G$. The following definition and lemma provide the needed properties of $H$.

**Definition 3.33** ($T_k$ graph). Define $T_2 = P_4$. For any $k > 2$, define $T_k = T(T_{k-1})$, where the $T(G)$ is the 3+1 construction from $G$, defined in Definition 3.15.

**Lemma 3.34.** For any $k \geq 2$, $T_k$ is $k$-cog-critical.

**Proof.** Since $c(T_2) = 2$ and $c(T_{k+1}) = c(T_k) + 1$ for any $k \geq 2$, proved by Lemma 3.16, we have $c(T_k) = k$ for any $k \geq 2$. We prove that $T_k$ is cog-critical for any $k \geq 2$ by induction on $k$.

Base Step ($k = 2$): Since any graph of less than 4 vertices is a cograph, $T_2$ is cog-critical.

Inductive Step: Suppose $T_k$ is cog-critical for some $k \geq 2$. We prove that $T_{k+1}$ is cog-critical. Let $T_{k+1} = (L \oplus M) \cup (M \oplus R) \cup (R \oplus \{s\})$, where $L, M, R$ are three copies of $T_k$ and $s$ is the single vertex. As in the proof of Lemma 3.16, every induced $P_4$ of $T_{k+1}$ is either of Type-1 (four vertices contained in one of $L, M$ and $R$) or Type-2 ($s$ and one vertex from $L,M$ and $R$ respectively). Let $v$ be an arbitrary vertex of $T_{k+1}$. We prove that $c(T_{k+1} - v) = k$. Let $C_l = \{c_1, \ldots, c_l\}$ be a set of $l$ colours for any $l \geq 1$.

We perform a case analysis on $v$.

Case 1: $v = s$.

There is no Type-2 $P_4$ in $T_{k+1} - v$. Since $c(T_k) = k$, there is a $k$-colouring $\sigma$ that cog-colours $L$, $M$ and $R$ individually, and hence can be extended to $T_{k+1}$. Thus, every Type-1 $P_4$ is not monochromatic under $\sigma$ and $c(T_{k+1} - v) = k$.

Case 2: $v \in V(A)$, where $A \in \{L, M, R\}$.

Without loss of generality, we suppose $v \in V(L)$. According to the inductive hypothesis, $L$ is $k$-cog-critical and thus there is a $(k-1)$-cog-colouring $\sigma$ of $L - v$ using colours from $C_{k-1}$. Extend $\sigma$ to cog-colour $M$ and $R$ using colours from $C_k$ and set $\sigma(s) = c_k$. There is no monochromatic Type-1 $P_4$ because $\sigma$ cog-colours $L - v$, $M$ and $R$. Every Type-2 $P_4$ is not monochromatic because every vertex in $L - v$ is not coloured $c_k$. Thus, $\sigma$ is a $k$-cog-colouring of $T_{k+1} - v$ and $c(T_{k+1} - v) = k$.

Therefore, $T_{k+1}$ is $(k+1)$-cog-critical. \qed
3.5. Cog-critical graphs

Let $\tau$ be a function with domain $D$ and $A \subseteq D$. Define $\tau_A$ to be the restriction of $\tau$ to the set $A$. That is, $\tau_A$ is a function with domain $A$ such that $\tau_A(a) = \tau(a)$ for all $a \in A$. If $\{A_1, \ldots, A_l\}$ is a partition of $D$, then $\tau$ is completely determined by $\{\tau_{A_1}, \ldots, \tau_{A_l}\}$. Given a graph $G$ and an induced subgraph $H$ of $G$, we simplify the notation of $\sigma_{V(H)}$ as $\sigma_H$, where $\sigma$ is a function defined on $V(G)$. A function $f : X \rightarrow Y$ is constant if all inputs have the same output. That is, for any $x_1, x_2 \in X$, $f(x_1) = f(x_2)$.

**Lemma 3.35.** For any $k \geq 2$, let $C_k = \{c_1, \ldots, c_k\}$ be a set of $k$ colours. For any function $m : V(T_k) \rightarrow C_k$, if $m$ is not constant, then $T_k$ has a $k$-cog-colouring $\sigma$ using colours from $C_k$ such that $\sigma(v) \neq m(v)$ for all $v \in V(T_k)$.

**Proof.** We prove this lemma by induction on $k$. We call two functions $f$ and $g$ contrary to each other, denoted by $f \sim g$, if they have different values for every input common to both domains. The $\sigma$ of the lemma we are constructing will be contrary to the given $m$.

Base Step (k=2): Let $\sigma$ be a colouring of $T_2$ such that $\sigma(v) = c_1$ if $m(v) = c_2$ and $\sigma(v) = c_2$ otherwise. Since $T_2$ is a $P_4$ and $m$ is not constant, $\sigma$ is a 2-cog-colouring of $T_2$ and $\sigma \sim m$.

Inductive Step: Let $k \geq 2$ and suppose the statement of the lemma is true for $k$. We now prove the statement is true for $k+1$. Let $L, M, R$ be the three copies of $T_k$ and $s$ be the single vertex such that $T_{k+1} = (L \oplus M) \cup (M \oplus R) \cup (R \oplus \{s\})$.

Claim: For any $m : V(T_{k+1}) \rightarrow C_{k+1}$ and any $A \in \{L, M, R\}$, there is always a colouring $\tau$ such that $\tau_A$ is a $k$-cog-colouring of $A$ and $\tau_A \sim m_A$.

Proof of claim: Let $A \in \{L, M, R\}$. We consider three cases depending on the size of the range of $m_A$.

Case 1. $m_A$ is constant.

Suppose the range of $m_A$ is $\{c_1\}$. Since $A$ is $k$-cog-critical, there is a $k$-cog-colouring $\tau_A$ of $A$ using colours from $C_{k+1} \setminus \{c_1\}$. It is clear that $\tau_A \sim m_A$.

Case 2. The range of $m_A$ has more than one but less than $k+1$ colours.

Suppose the range of $m_A$ is a subset of $C_k$. Since $A$ is a copy of $T_k$, according to the inductive hypothesis, there is a $k$-cog-colouring $\tau_A$ of $A$ using $k$ colours from $C_k$ and $\tau_A \sim m_A$.

Case 3. The range of $m_A$ has $k+1$ colours.

Let $m'_A : V(A) \rightarrow C_k$ be a function such that $m'_A(v) = m_A(v)$ if $m_A(v) \neq c_{k+1}$ and $m'_A(v) = c_1$ otherwise. Now $m'_A$ has range of $k \geq 2$ colours from $C_k$. According to the inductive hypothesis, there is a $k$-cog-colouring $\tau_A$ of $A$ using $k$ colours from $C_k$ such that $\tau_A \sim m'_A$. For all $v$ with $m_A(v) \neq c_{k+1}$,
m_A(v) = m'_A(v) \neq \tau_A(v)$. For all $v$ with $m_A(v) = c_{k+1}$, $m_A(v) \neq \tau_A(v)$ because $\tau_A(v) \neq c_{k+1}$. Thus, $\tau_A$ is contrary to $m_A$.

The claim is now proved.

Let $\sigma$ be a colouring of $T_{k+1} - s$ such that $\sigma(v) = \tau(v)$ for every $v \in V(T_{k+1} - s)$. We complete the proof of the inductive step for case $k+1$ with a case analysis of the colours used by $\sigma_L$, $\sigma_M$ and $\sigma_R$.

Case 1. $\sigma_L$, $\sigma_M$ and $\sigma_R$ do not use the same $k$ colours.

Without loss of generality, we suppose the colours used by $\sigma_L$ is different from the colours used by $\sigma_M$. Since $L$ and $M$ are both $k$-cog-critical, the range of $\sigma_L$ and the range of $\sigma_M$ have exactly $(k-1)$ colours in common. Thus, there are at least two colours, $c_i$ and $c_j$, such that $c_i$ is used by $\sigma_L$ but not by $\sigma_M$, and $c_j$ is used by $\sigma_M$ but not by $\sigma_L$. If $m(s) \neq c_i$, we set $\sigma(s)$ to $c_i$. If $m(s) = c_i$, we set $\sigma(s)$ to $c_j$. As in the proof of Lemma 3.16 and since $\sigma_A$ is a $k$-cog-colouring of $A$ for all $A \in \{L, M, R\}$, every possible induced $P_4$ of $T_{k+1}$ must contain $s$ and have exactly one vertex from $L, M$ and $R$ respectively. Every such $P_4$ is not monochromatic because either the vertex from $L$ or the vertex from $M$ is not coloured $\sigma(s)$. Therefore, $\sigma$ is a $(k+1)$-cog-colouring of $T_{k+1}$. By construction, $\sigma \approx m$.

Case 2: $\sigma_L$, $\sigma_M$ and $\sigma_R$ all use the same $k$ colours.

Suppose they all use $C_k$. If $m(s) \neq c_{k+1}$, we set $\sigma(s) = c_{k+1}$. Then $\sigma$ is a $(k+1)$-cog-colouring of $T_{k+1}$ with $\sigma \approx m$. If $m(s) = c_{k+1}$, there must be some colour $c_i \neq c_{k+1}$ in the range of $m$ because $m$ is not constant. We then redefine $\sigma(v) = c_{k+1}$ for each $v$ with $m(v) = c_i$, and set $\sigma(s) = c_i$. For each $A \in \{L, M, R\}$, $\sigma_A$ is a $k$-cog-colouring of $A$ because the redefinition just switches $c_i$ with $c_{k+1}$. As in the proof of Lemma 3.16, every possible induced $P_4$ of $T_{k+1}$ must contain $s$ and have exactly one vertex from $L, M$ and $R$ respectively. Every such $P_4$ is not monochromatic because $s$ is the only vertex coloured $c_i$. The resulting $\sigma$ is a $(k+1)$-cog-colouring of $T_{k+1}$ and $\sigma \approx m$.

Thus, for any $m : V(T_{k+1}) \to C_{k+1}$, if $m$ is not constant, there is a $(k+1)$-cog-colouring $\sigma$ of $T_{k+1}$ such that $\sigma \approx m$.

Therefore the statement of the lemma is true for $k+1$. 

\textbf{Corollary 3.36.} For any $k \geq 2$, let $F$ be an empty graph with $|V(T_k)|$ vertices, $H^1_F, H^2_F, H^3_F$ be three disjoint copies of $T_k$, $M_F$ be a matching between $V(F)$ and $V(H^1_F)$. For any vertex $v \in V(F)$, let $v'$ be the vertex in $H^1_F$ such that $vv' \in M_F$. Define $H_F = (H^1_F \oplus H^2_F) \cup (H^2_F \oplus H^3_F) \cup (H^3_F \oplus K_1)$ and the graph $J$ with $V(J) = V(F) \cup V(H_F)$ and $E(J) = M_F \cup E(H_F)$. Let $C_k = \{c_1, \ldots, c_k\}$ be a set of $k$ colours. Let $\tau$ be a $k$-colouring of $F$. If $\tau$ is constant, then for any vertex $v \in V(J)$, there is a $k$-cog-colouring $\sigma$ of $J - v$
such that $\sigma_F = \tau$ and there is only one monochromatic edge in $M_F$. If $\tau$ is not constant, then there is a $k$-cog-colouring $\sigma$ of $J$ such that $\sigma_F = \tau$ and there is no monochromatic edge in $M_F$.

Proof. We first prove the first part of the lemma. Suppose $\tau$ is constant with colour, say $c_1$. Let $v$ be a vertex of $G$. We consider three cases depending on the part of $G$ to which the $v$ belongs.

Case 1: $v \in V(F)$.

We $(k-1)$-cog-colour $H^1_F - v'$ using colours from $C_k \setminus \{c_1\}$ and set $\sigma(v') = c_1$. We then $k$-cog-colour $H^2_F$ and $H^3_F$ with colours from $C_k$. For every $P_4$ in $J - v$, it is either contained in one of $\{H^1_F, H^2_F, H^3_F\}$ or it has an edge from $M_F - vv'$. Since each one of $\{H^1_F, H^2_F, H^3_F\}$ is cog-coloured and every edge in $M_F$ except $vv'$ is not monochromatic, we get a $k$-cog-colouring of $J - v$ and $vv'$ is the only monochromatic edge in $M_F$.

Case 2: $v \in V(H^2_F) \cup V(H^3_F)$.

We $(k-1)$-cog-colour $H^1_F - v$ with colours from $C_k \setminus \{c_1\}$. We then $k$-cog-colour $H^2_F$ and $H^3_F$ with colours from $C_k$. For every $P_4$ in $J - v$, it is either contained in one of $\{H^1_F - v, H^2_F, H^3_F\}$ or it has an edge from $M_F$ different than $vv'$. Since each one of $\{H^1_F - v, H^2_F, H^3_F\}$ is cog-coloured and $vv'$ is the only monochromatic edge in $M_F$, $J - v$ is $k$-cog-colourable.

Case 3: $v \in V(H^2_F) \cup V(H^3_F)$.

Suppose, without loss of generality, $v \in V(H^2_F)$. We $(k-1)$-cog-colour $H^3_F - v$ with colours from $C_k \setminus \{c_1\}$ and $k$-cog-colour $H^2_F$ with colours from $C_k$. Then, we choose an arbitrary vertex $u'$ of $H^1_F$, colour $u'$ with $c_1$ and $(k-1)$-cog-colour $H^1_F - u'$ with colours from $C_k \setminus \{c_1\}$. For every $P_4$ in $J - v$, it is either contained in one of $\{H^1_F, H^2_F, H^3_F - v\}$ or it has an edge from $M_F$ different than $vu'$. Since each one of $\{H^1_F, H^2_F, H^3_F - v\}$ is cog-coloured and $vu'$ is the only monochromatic edge in $M_F$, $J - v$ is $k$-cog-colourable.

Now suppose $\tau$ is not constant. Since a colouring of $H^1_F$ is a colouring of $T_k$, by using $\tau$ as the function in Lemma 3.35, we have a $k$-cog-colouring of $H^1_F \cup M_F$ using colours from $C_k$ such that there is no monochromatic edge in $M_F$. We can then arbitrarily cog-colour $H^2_F$ and $H^3_F$ using colours from $C_k$. For every $P_4$ in $J$, it is either contained in one of $\{H^1_F, H^2_F, H^3_F\}$ or it has an edge from $M_F$. Since $\{H^1_F, H^2_F, H^3_F\}$ are cog-coloured and every edge in $M_F$ is not monochromatic, we get a $k$-cog-colouring of $G$ with no monochromatic edge in $M_F$. \hfill $\Box$

Theorem 3.37. Let $\mathcal{F}$ be a $n$-uniform $(k+1)$-colour-critical hypergraph, where $n = |V(T_k)|$. The graph $G = D(\mathcal{F}, T_k)$ is $(k + 1)$-cog-critical.
3.5. Cog-critical graphs

Proof. We first categorize the $P_4$s in $G$. For any two hyperedges $A, B \in \mathcal{F}$, $A \neq B$, $H_A$ and $H_B$ form a join in $G$. Thus, there is no $P_4$ which has vertices in more than two different satellite graphs. If $u_1u_2u_3u_4$ is a $P_4$ with vertices from two different satellite graphs $H_A$ and $H_B$, then $u_2 \in V(H_A), u_3 \in V(H_B)$ and $u_1, u_4 \in A \cup B$. If not, then a triangle is formed between the satellite graphs. Therefore, any $P_4$, say $u_1u_2u_3u_4$ of $G$, can be either one of the following three types.

1. **Type-1 $P_4$:** The four vertices are contained in a $M_F \cup H_F$ for some hyperedge $F$.

2. **Type-2 $P_4$:** $u_1 \in A, u_4 \in A, u_2 \in V(H_A), u_3 \in V(H_B)$, where $A$ and $B$ are two different hyperedges.

3. **Type-3 $P_4$:** $u_1 \in A, u_4 \in B, u_2 \in V(H_A), u_3 \in V(H_B)$, where $A$ and $B$ are two different hyperedges.

Let $C_k = \{c_1, \ldots, c_k\}$ be a set of $k$ colours. We break up the proof of the theorem into four claims.

Claim 1: $c(G) \leq k + 1$.

Let $\sigma$ be a proper colouring of $\mathcal{F}$ using colours from $C_{k+1}$. Since there is no monochromatic hyperedge under $\sigma$, according to Corollary 3.36, we can extend $\sigma$ to a $k$-cog-colouring $G^*$ such that ends of each edge in the matchings, $\bigcup_{F \in \mathcal{F}} M_F$, are coloured differently. Thus, a $P_4$ of any type must not be monochromatic under $\sigma$. Therefore, $\sigma$ is a $k$-cog-colouring of $G$ and Claim 1 is proved.

Claim 2: $c(G) > k$.

Suppose, to the contrary, $G$ has a $k$-cog-colouring $\sigma$ using colours from $C_k$. Since $\mathcal{F}$ is $(k + 1)$-colour-critical, there is at least one monochromatic hyperedge $F$ in $\mathcal{F}$ coloured by $\sigma$. Suppose the vertices of $F$ are coloured $c_1$. Since $H_F^1, H_F^2, H_F^3$ are all $k$-cog-critical graphs, $c_1$ must be used by some vertex $u_1$ in $H_F^1$, $u_2$ in $H_F^2$ and $u_3$ in $H_F^3$. But then $xu_1u_2u_3$ is a monochromatic $P_4$, where $xu_2 \in M_F$ for some $x \in F$, a contradiction. Therefore, $c(G) > k$ and Claim 2 is proved.

Claim 3: For any $u \in \mathcal{F}, c(G - u) \leq k$.

Since $\mathcal{F}$ is $(k+1)$-colour-critical, there is a proper $k$-colouring $\sigma$ of $\mathcal{F} - u$. According to Corollary 3.36, we can extend $\sigma$ to a $k$-cog-colour $G^*$ such that ends of each edge in the matchings are coloured differently. Thus, a $P_4$ of either type must not be monochromatic under $\sigma$. Therefore, $\sigma$ is a $k$-cog-colouring of $G$ and Claim 3 is proved.

Claim 4: For any $u \in G^*$, $c(G - u) \leq k$. 

50
3.5. Cog-critical graphs

Let $A$ be the hyperedge such that $u \in V(H_A)$. Since $\mathcal{F}$ is $(k + 1)$-colour-critical, there is a proper $k$-colouring $\sigma$ of $\mathcal{F} - A$. We then extend $\sigma$ to $k$-cog-colour $G^* - V(H_A)$ using colours from $C_k$ such that ends of edges in $\bigcup_{F \in \mathcal{F}, F \neq A} M_F$ are coloured differently. Because $A$ must be monochromatic under $\sigma$, according to Corollary 3.36, we can extend $\sigma$ to $k$-cog-colour $M_A \cup H_A$ such that there is only one monochromatic edge in $M_A$. For any $F \in \mathcal{F}$, $M_F \cup H_F$ is $k$-cog-coloured. Thus, $P_4$s of Type-1 are not monochromatic. Since there is only one monochromatic edge in $\bigcup_{F \in \mathcal{F}, M_F}$, $P_4$s of Type-2 and Type-3 are not monochromatic. Therefore, $\sigma$ is $k$-cog-colouring of $G$ and Claim 4 is proved.

By Claim 1 through 4, $G$ is $(k + 1)$-cog-critical.

Using Theorem 3.37, we can have a different proof of Theorem 3.30 based on the D-construction.

Proof. Let $k$ be any fixed integer larger than 2. Then $|V(T_{k-1})| > 3$. For any $n > 0$, according to Theorem 3.29, there is a $k$-colour-critical $|V(T_{k-1})|$-uniform hypergraph $\mathcal{F}$ of max\{$n, |V(T_{k-1})| \cdot k\}$ vertices. According to Theorem 3.37, the D-construction $G$ from $(\mathcal{F}, T_k)$ is $k$-cog-critical with more than $n$ vertices.
Chapter 4

Conclusion

To the best of our knowledge, we have conducted (for the first time) a probabilistic analysis of super solutions of random instances of SAT and CSPs. While we have focused on the special (but already challenging) case of (1,0)-super solutions, some of our analysis extends to the case of \((a,0)\)-super solutions for \(a > 1\). For random instances of CSPs, new analytical methods and ideas are needed to obtain a more detailed characterization of the behaviour of the super solutions, and we leave this as a future work. It is also highly interesting to conduct a systematic empirical analysis to fully understand the hardness of solving random instances of (1,0)-k-SAT as well as the hardness of solving the projected standard SAT instances, which may serve as suite of SAT benchmark with a unique structural properties. We wonder if our analysis can be extended to random instances of other problems such as graphical games where solution concepts similar to super solutions have been used.

For the problem of partitioning graphs into cographs, we have studied several problems related to cog-chromatic number of graphs which have not been studied before, as far as we know. For example, the order of minimum \(k\)-cog-chromatic graphs is studied in Section 3.4. Moreover, methods that construct arbitrarily large graphs with required properties are given in Section 3.4 and Section 3.5. We are very interested in answering conjectures made in Section 3.3 and Section 3.4. For example, does the smallest \(k\)-cog-chromatic graphs have \(2^k + 2^{k-1} - 2\) vertices? We would also like to spend more time on studying approximation algorithms for \(c(G)\) and the hardness of approximating \(c(G)\), where \(G\) is an arbitrary graph.
Bibliography


Bibliography


Bibliography


