UNIVERSITY
of
GLASGOW

Computing Science
Ph.D. Thesis

Minimaximal and maximinimal optimisation problems: a partial order-based approach

David Francis Manlove

Submitted for the degree of

Doctor of Philosophy

in the University of Glasgow

June 1998

Abstract

We study a class of optimisation problems called *minimaximal* and *maximinimal* optimisation problems. Such a problem Π' may be obtained from an optimisation problem Π, called the *source* optimisation problem, by defining a partial order $\prec^x$ on the set of feasible solutions $\mathcal{F}(x)$ for a given instance x of Π. If the objective of Π is to maximise (respectively minimise) the measure over $\mathcal{F}(x)$, then the objective of Π' is to minimise (maximise) the measure over all elements of $\mathcal{F}(x)$ that are maximal (minimal) with respect to $\prec^x$. Many optimisation problems occurring in the literature are minimaximal or maximinimal optimisation problems.

In this thesis, we present the first unifying framework for formulating minimaximal and maximinimal optimisation problems, based on this partial order concept. To accompany this framework, we define a variety of partial orders, an important example being the partial order of set inclusion. By considering various source optimisation problems from the literature, and partial orders from our collection, we use our framework to obtain a range of minimaximal and maximinimal optimisation problems. We study these individual examples mainly from the point of view of algorithmic complexity.

The series of examples begins with the source optimisation problem CHROMATIC NUMBER, whose objective is to minimise the number of colours over all proper colourings of a given graph. A related problem is ACHROMATIC NUMBER, in which we seek to maximise the number of colours over all proper colourings of a given graph G such that each pair of distinct colours occurs at the endpoints of some edge of G. We show that ACHROMATIC NUMBER is a maximinimal counterpart of CHROMATIC NUMBER, by defining a partition-related partial order on the set of all proper colourings of G. A natural refinement of this partial order gives rise to an additional maximinimal optimisation problem concerned with graph colouring, which we call B-CHROMATIC NUMBER. The objective of this problem is to maximise the number of colours over all proper colourings of G such that for each colour i, there is a distinguished vertex of colour i that is adjacent to a vertex of every colour $j \neq i$. Whilst the ACHROMATIC NUMBER problem has been studied for over thirty years, the B-CHROMATIC NUMBER problem is new. We show that the decision version of B-CHROMATIC NUMBER is NP-complete in arbitrary graphs, and also bipartite graphs. However, we prove that B-CHROMATIC NUMBER is solvable in polynomial time for trees, in contrast with ACHROMATIC NUMBER.

Many other examples of minimaximal and maximinimal optimisation problems relating to graph parameters have already been studied in the literature. We focus on graph-theoretic minimaximal and maximinimal optimisation problems that may be obtained from source optimisation problems using the partial order of set inclusion. In particular, we investigate source optimisation problems and their minimaximal or maximinimal counterparts relating to vertex, edge and total covers, and independent sets, matchings and total matchings in a graph. Each of the twelve optimisation problems implied by the previous sentence has been studied in some form in the literature. We review existing complexity results and obtain a number of new results, namely NP-completeness proofs for the decision versions of MAXIMUM MINIMAL TOTAL COVER in planar graphs, MINIMUM MAXIMAL TOTAL MATCHING in bipartite and chordal graphs, and MINIMUM INDEPENDENT DOMINATING SET (or MINIMUM MAXIMAL INDEPENDENT SET) in planar cubic graphs.

We also study source optimisation problems and their minimaximal or maximinimal counterparts relating to strong stable sets, cliques, dominating sets, total dominating sets, edge dominating sets and irredundant sets in a graph. Again, we survey existing algorithmic results, if any, for the twelve implicit optimisation problems, and obtain several new results, namely NP-completeness proofs for the decision versions of MINIMUM MAXIMAL STRONG STABLE SET in planar graphs of maximum degree 3, MINIMUM MAXIMAL CLIQUE in general graphs, and MINIMUM TOTAL DOMINATING SET in planar cubic graphs.

For a graph $G = (V, E)$, a refinement of the notion of an independent set that is maximal with respect to the partial order of set inclusion has been considered in the literature. An independent set $S \subseteq V$ is *k-maximal* ($k \geq 1$) if the removal of any $r - 1$ vertices from S, together with the addition of any r vertices from $V \backslash S$ (for any $r \leq k$), results in a non-independent set. We consider minimaximal optimisation problems related to finding minimum cardinality k-maximal independent sets in a graph. We focus mainly on the case $k = 2$, and prove that the decision version of the problem of determining the minimum size of a 2-maximal independent set is NP-complete, even for planar graphs of maximum degree 3. However, for trees, we give a linear-time algorithm.

Many integer-valued graph parameters have fractional counterparts in the literature. We define the concept of fractional graph optimisation problems, relating to fractional graph parameters, and show how minimaximal and maximinimal versions may be defined, using a partial order on functions. We formulate several examples of such problems using the framework, and by considering appropriate linear programming constructions, we show that the optimal measure (the solution to the evaluation version of the minimaximal or maximinimal fractional graph optimisation problem concerned) is computable, has rational values, and is attained by some function of compact representation which satisfies the feasibility constraint for the minimaximal or maximinimal fractional graph optimisation problem concerned. These three issues have implications for the algorithmic behaviour of a minimaximal or maximinimal fractional graph optimisation problem. We also survey complexity results relating to the source fractional graph optimisation problems and their minimaximal or maximinimal counterparts that we define.

Minimaximal and maximinimal optimisation problems that relate to areas other than the domain of graph theory are also studied in this thesis. By considering source optimisation problems belonging to the Garey and Johnson [*Computers and Intractability*, Freeman, 1979] problem categories of Network Design, Sets and Partitions, Data Storage, Compression and Representation, Mathematical Programming, and Logic, and by imposing various partial orders, we formulate a range of natural minimaximal and maximinimal optimisation problems. These interesting individual examples, most of which are new, are minimaximal or maximinimal versions of the following problems (Garey and Johnson problem numbers in brackets): LONGEST PATH (ND29), 3D-MATCHING (SP1), MINIMUM

TEST SET (SP6), BIN PACKING (SR1), KNAPSACK (MP9), MAXIMUM 2-SAT (LO5), ONE-IN-THREE 3SAT (LO4), LONGEST COMMON SUBSEQUENCE (SR10), SHORTEST COMMON SUPERSEQUENCE (SR8), LONGEST COMMON SUBSTRING (SR10) and SHORTEST COMMON SUPERSTRING (SR9). We survey complexity results for the source optimisation problem Π in each case, and also those, if any, for the minimaximal or maximinimal counterpart(s) of Π. We obtain new NP-completeness results for minimaximal or maximinimal versions of the first seven problems in the above list, and construct a polynomial-time algorithm for a minimaximal counterpart of LONGEST COMMON SUBSTRING.

We also consider the computational complexity of minimaximal and maximinimal optimisation problems from a more general viewpoint. We present conditions under which a Turing reduction from an optimisation problem, Π_1, to another, Π_2, is also a Turing reduction from Π'_1 to Π'_2, where Π'_i is a minimaximal or maximinimal version of Π_i $(i = 1, 2)$. We call Turing reductions satisfying these additional constraints *MM-reductions*. Some examples of MM-reductions are given, involving several partial orders, and involving source optimisation problems from a variety of domains.

Additionally, we discuss the question of how, in general, we may test a feasible solution of an optimisation problem Π for maximality or minimality, and how we may find feasible solutions that are maximal or minimal, with respect to a partial order defined on the feasible solutions of Π for a given instance. In particular, we examine this question when Π is BIN PACKING and when Π is CHROMATIC NUMBER.

We show that, for a natural partition-related partial order $\prec^x$ defined on the feasible solutions for a given instance x of BIN PACKING, the problem of testing a feasible solution for $\prec^x$-minimality is NP-hard, but perhaps surprisingly, the problem of finding a feasible solution that is $\prec^x$-minimal is polynomial time solvable.

In the case of CHROMATIC NUMBER, we consider two families of partition-related partial orders defined on the set of all proper colourings of a given graph G. We investigate the problems of testing a proper colouring for minimality, and finding proper graph colourings that are minimal, with respect to these partial orders. We also consider the complexity of the associated maximinimal optimisation problems in each case. Our algorithmic results for testing and finding show where the thresholds between polynomial-time solvability and NP-hardness lie, within the hierarchy of problems corresponding to the two partial order families. These developments imply possible local search strategies for approximating the chromatic number in certain graph classes.

Finally, we revisit the general framework for minimaximal and maximinimal optimisation problems, considering an alternative definition in which the partial order is defined on the set of all possible (not necessarily feasible) solutions. We show that in some, but not all, cases, the revised definition gives rise to the same maximal and minimal solutions as would be obtained by defining the partial order concerned on the feasible solutions. These observations yield a greater insight into the structure of minimaximal and maximinimal optimisation problems.

Thesis supervisor: Dr. Rob Irving, Senior Lecturer in Computing Science.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Rob Irving for his supervision. He has been a constant source of advice, encouragement and guidance during my research, and has always provided constructive criticism of my work. His enthusiasm for the subject has helped to make this a very enjoyable period of study for me.

I would also like to thank Joe Morris and David Watt, my supervisory committee, for their time and their advice. Also, I would like to express my gratitude to the Department, for providing such a stimulating and friendly environment to work in.

I am grateful to Steve Hedetniemi for valuable discussions regarding some of the graph-theoretic concepts in this thesis.

Thank you too to the very efficient Document Delivery Service of Glasgow University Library, and British Library Document Supply Centre, for their assistance with inter-library loans.

My research was supported by the Engineering and Physical Sciences Research Council, in the form of a Research Studentship, during the period October 1995 to September 1998.

Finally, I would like to thank my parents and Carolynn, for their love and support.

Declaration

This dissertation is submitted in accordance with the regulations for the degree of Doctor of Philosophy in the University of Glasgow. No part of it has been previously submitted by the author for a degree at any other university and all results contained within are claimed as original. The proofs of Theorems 3.4.3, 7.4.1, 9.3.2 and 9.3.3 contain ideas suggested by Rob Irving, and the proof of Theorem 7.4.2 contains ideas suggested by Gerhard Woeginger.

Publications

R.W. Irving and D.F. Manlove. The b-chromatic number of a graph. To appear in *Discrete Applied Mathematics*.
(This paper is based on Sections 3.2, 3.3, 3.4 and 3.6.)

D.F. Manlove. On the algorithmic complexity of twelve covering and independence parameters of graphs. To appear in *Discrete Applied Mathematics*.
(This paper is based on Section 4.2.)

Chapter 1

Introduction and background

1.1 Introduction

Optimisation problems have a well-defined structure: a set of *instances* $\mathcal{I}$ of the problem, and, given an instance $x \in \mathcal{I}$, a set of *feasible solutions* $\mathcal{F}(x)$, some notion of the *measure* $m(x, s)$ of a feasible solution s, and a *goal*, to maximise or minimise. The objective of an optimisation problem Π is to find[1] a feasible solution t whose measure is maximum or minimum (according to the goal) over that of all feasible solutions. Such a feasible solution t is said to be a *globally optimal feasible solution*, and t is said to have *globally optimal measure*. An optimisation problem whose goal is to maximise (respectively minimise) is referred to as a *maximisation* (*minimisation*) problem.

This thesis is concerned with a class of optimisation problems called *minimaximal* and *maximinimal* optimisation problems. Before we introduce this class in Section 1.3, we define some concepts and review literature relating to optimisation problems in general.

1.2 Review of concepts relating to optimisation problems in general

1.2.1 Optimisation problems: three versions in one

In Section 1.1, we defined the *search* version of an optimisation problem. An example of such a problem is CLIQUE, whose objective is to find, given a graph $G = (V, E)$ (a typical instance), a subset V' of V such that every pair of vertices in V' is an edge in G (i.e. V' is feasible), where the cardinality of V' (the measure in this case) is maximum over all such sets. An optimisation problem has two other versions, which we now define.

The objective of the *evaluation version* of an optimisation problem is to compute the globally optimal measure, rather than to find a globally optimal feasible solution.

[1] We have described here the *constructive* or *search* version of Π. There are two other versions of an optimisation problem, namely the *evaluation* and *decision* versions, which will be described in Section 1.2.1.

In the *decision version* of a maximisation (respectively minimisation) problem Π, we ask whether there is a feasible solution whose measure is at least K (at most K), for a given instance and some given positive integer K.

The relationship between the search, evaluation and decision versions of an optimisation problem is discused by Ausiello et al. [10, Section 1.1] and Bovet and Crescenzi [24, Section 6.3].

1.2.2 NP-completeness and NP-hardness

The design of *efficient* algorithms to solve optimisation problems will always be necessary, even as computers become faster [92, pp.7-8]. In particular, the design of *polynomial-time* algorithms (i.e. algorithms whose running time is polynomial in the size of their input) is of paramount importance. The symbol P is used to denote the class of decision problems (i.e. problems whose solution is a 'yes' or 'no' answer) that are polynomial-time solvable[2]. For some optimisation problems, no polynomial-time algorithm has been found. CLIQUE (defined in Section 1.2.1) is an example of an optimisation problem in this category.

This situation prompted Cook [55] to introduce, in his paper entitled *The Complexity of Theorem Proving Procedures* (1971), the class NP of decision problems that can be solved in polynomial time using a nondeterministic algorithm. Cook then demonstrated that a particular problem in NP (known as SATISFIABILITY, problem LO1 of [92]) has the property that every other problem in NP can be polynomially reduced to it[3]. This means that if SATISFIABILITY can be solved in polynomial time then so can any problem in NP, and if any problem Π in NP is *intractable* (i.e. there exists no polynomial-time algorithm to solve Π) then SATISFIABILITY is also intractable.

Following Cook's result, Karp [140] gave, in his paper *Reducibility Among Combinatorial Problems* (1972), examples of many other combinatorial problems in NP that have exactly the same property as SATISFIABILITY mentioned above. Thus the class of *NP-complete* decision problems was born, having the property that if any one of them can be solved in polynomial time, then they all can, and if any one of them is intractable, then they all are. In general, to show that a decision problem Π' is NP-complete, one needs to show that (i) $\Pi' \in$NP, and (ii) there is some NP-complete decision problem Π that may be polynomially reduced to Π'. A decision problem satisfying only condition (ii) is *NP-hard*, in that it is just as hard as the NP-complete problems, and cannot be solved in polynomial time unless P=NP.

The decision version of CLIQUE is just one NP-complete problem appearing in Karp's paper. The list of known NP-complete decision problems quickly grew and these problems were gathered together in the classical work of Garey and Johnson in 1979: *Computers and Intractability* [92]. The field is ever-changing, with new NP-complete problems being discovered regularly. To cope with the need for researchers to keep up to date with the status of open problems, Johnson began a (usually) quarterly column in December

[2] In this thesis, we abuse notation and also use $\Pi \in$P to indicate that the search version Π of an optimisation problem is polynomial-time solvable.

[3] Discovered independently by Levin [153] in 1973.

1981, appearing in the *Journal of Algorithms*, called *The NP-Completeness Column: An Ongoing Guide*[4].

Many NP-complete problems are decision versions of *NP optimisation problems* [24, p.112]. Informally, an NP optimisation problem Π is an optimisation problem such that the set of instances $\mathcal{I}$ is recognisable in deterministic polynomial time, the set of feasible solutions $\mathcal{F}(x)$ for a given instance $x \in \mathcal{I}$ is recognisable in deterministic polynomial time, the measure function is computable in polynomial time, and there exists some polynomial p such that, for any instance x and for any feasible solution $y \in \mathcal{F}(x)$, $|y| \leq p(|x|)$ (where $|y|$ and $|x|$ denote the sizes of y and x respectively under a reasonable encoding scheme for Π). NPO denotes the class of NP optimisation problems. An NP optimisation problem Π has the property that the decision version of Π is in NP [24, Lemma 6.1].

In order to prove that the search version Π' of an optimisation problem is hard to solve, it is sometimes more convenient to consider a *Turing reduction* from a *search problem* Π to Π' (as in Chapter 8, for example), rather than to formulate a polynomial reduction from an NP-hard decision problem Π to the decision version of Π'. Informally, a search problem Π consists of a set $\mathcal{I}$ of instances, and for each instance $x \in \mathcal{I}$, a set $\mathcal{F}(x)$ of feasible solutions for x. An algorithm is said to solve a search problem Π if, given any instance $x \in \mathcal{I}$, the algorithm returns some $y \in \mathcal{F}(x)$ if $\mathcal{F}(x) \neq \emptyset$, or 'no' if $\mathcal{F}(x) = \emptyset$. For example, the search version of an optimisation problem, introduced in Section 1.1, can be viewed as a special case of a search problem. A Turing reduction from a search problem Π to a search problem Π' is a polynomial-time algorithm for solving Π, using one or more calls to a polynomial-time hypothetical subroutine S for solving Π'. If there is a Turing reduction from a search problem Π to a search problem Π', we say that Π is *Turing-reducible* to Π', denoted $\Pi \alpha_T \Pi'$. Any decision problem may be formulated as a search problem (see Garey and Johnson [92, p.110] for further details), and hence we may make the following definition. If Π is an NP-complete problem (Π is a decision problem formulated as a search problem) and Π' is a search problem, where $\Pi \alpha_T \Pi'$, then Π' is said to be *NP-hard*. As in the above definition of NP-hardness for decision problems, Π' is NP-hard implies that Π' cannot be solved by a polynomial-time algorithm unless P=NP. Finally, if Π is an NP-hard search problem, and Π' is a search problem, where $\Pi \alpha_T \Pi'$, then Π' is NP-hard [92, p.113].

See Garey and Johnson [92] for definitions of complexity-theoretic terminology used in this thesis but not defined.

1.2.3 Approximation of NP optimisation problems

Many believe that demonstrating NP-completeness for the decision version of an optimisation problem Π is tantamount to proving that Π is intractable. This situation has motivated researchers to concentrate their efforts instead on finding a polynomial-time algorithm that will give a solution *close* (given some definition of closeness) to optimal, for all instances x of Π.

[4] It appears that this column terminated in 1992, awaiting the second edition of *Computers and Intractability*, not as yet published.

We now formalise this notion of 'closeness' as follows. Given an NPO problem Π, an algorithm A is an *approximation algorithm* for Π if, for any given instance x of Π, A returns a feasible solution $A(x)$. If we denote by $APPROX_A(x)$ the measure of the approximate solution $A(x)$ and $OPT(x)$ the measure of the optimal solution then the *performance ratio of A with respect to x*, for Π a maximisation (respectively minimisation) problem, is:

$$R_A(x) = \frac{OPT(x)}{APPROX_A(x)} \qquad \left(R_A(x) = \frac{APPROX_A(x)}{OPT(x)}\right)$$

Given a constant $c \geq 1$, we say that A is a *c-approximation algorithm* for Π, if, for any instance x, the performance ratio of A with respect to x satisfies $R_A(x) \leq c$. If an NPO problem Π admits a c-approximation algorithm, we say that Π is *approximable within c*. We define the *performance guarantee* of A to be

$$R_A = \inf\{c \geq 1 : \Pi \text{ is approximable within } c\}.$$

Given an arbitrary function $\varepsilon : \mathbb{N} \longrightarrow [1, \infty)$, we say that A is an *$\varepsilon(n)$-approximation algorithm* for Π, if, for any instance x, the performance ratio of A with respect to x satisfies $R_A(x) \leq \varepsilon(|x|)$. If an NPO problem Π admits an $\varepsilon(n)$-approximation algorithm, we say that Π is *approximable within $\varepsilon(n)$*.

There are a variety of classes for NPO problems that depend on their approximability properties. Given an NPO problem Π, Π belongs to the class APX if there is some *constant* $\varepsilon > 1$ such that Π has a polynomial-time ε-approximation algorithm. Π belongs to the class PTAS if there is an algorithm A such that, for any instance x of Π and any real number $\varepsilon > 1$, A produces a feasible solution in time polynomial in $|x|$, and A satisfies $R_A(x) \leq \varepsilon$. Such an algorithm is called a *polynomial-time approximation scheme (ptas)*. Π belongs to the class FPTAS if there is a ptas for Π whose time complexity is bounded by a polynomial in the length of the input and $1/(\varepsilon - 1)$. Such an algorithm is called a *fully polynomial-time approximation scheme (fptas)*. We also have the class PO of NPO problems that can be solved in polynomial time. Finally, it is worth mentioning the problems in NPO\APX: they have no ε-approximation algorithm, for any $\varepsilon \geq 1$, unless P=NP. Clearly

$$\text{PO} \subseteq \text{FPTAS} \subseteq \text{PTAS} \subseteq \text{APX} \subseteq \text{NPO}$$

and all of these inclusions are strict if and only if P$\neq$NP (see Bovet and Crescenzi [24] for further details). Ausiello et al. [10] give examples of NPO problems in each of the above classes.

A major issue in the theory of approximation of NPO problems whose decision versions are NP-complete is that, although all NP-complete problems may be polynomially reduced to one another, two problems in NPO may have quite different properties of approximability. For example, the NPO problems CHROMATIC NUMBER and CLIQUE (whose decision versions are problems GT4 and GT19 of [92] respectively) are not approximable within $n^{\frac{1}{7}-\varepsilon}$ and $n^{\frac{1}{2}-\varepsilon}$ respectively[5], for any $\varepsilon > 0$ (where n is the number of vertices in the given graph), unless P=NP [14, 201], whilst the NPO problem BIN PACKING (whose decision version is problem SR1 of [92]) is approximable within $\frac{3}{2}$ [195]. As a result of this apparent lack of uniformity regarding the approximability of NP-complete problems, much research has been directed towards constructing reductions between NPO problems that do preserve approximability.

Ausiello et al. [11], and Paz and Moran [181], were among the first to define reductions from one NPO problem to another that preserve approximability. Following the work of Orponen and Mannila [176], Crescenzi and Panconesi [60] defined *P-reductions* and *F-reductions* and used them to formulate notions of completeness in the classes APX and PTAS. An *APX-complete* problem Π has the property that Π is in APX, and Π is in PTAS implies that APX=PTAS. Similarly a *PTAS-complete* problem Π has the property that Π is in PTAS, and Π is in FPTAS implies that PTAS=FPTAS. Since the equality of APX and PTAS, or of PTAS and FPTAS, would imply that P=NP, proving that a problem is complete for either of the classes APX or PTAS is convincing evidence that it cannot be approximated by a stronger form of polynomial approximation.

Papadimitriou and Yannakakis [180] define the class *MAX-NP* and its subclass *MAX-SNP* and use *L-reductions* to formulate a notion of *MAX-SNP-completeness*. A MAX-SNP-complete problem Π has the property that:

1. Π is in APX.

2. Π is in PTAS implies MAX-SNP=PTAS.

Arora et al. [8] prove that a MAX-SNP-complete problem does not have a ptas unless P=NP.

More detailed surveys concerning the theory of approximiation algorithms have been carried out by Bruschi et al. [29] and by by Ausiello et al. [10]. In addition, Crescenzi and Kann [59] maintain a list containing NPO problems together with their current approximability status.

1.2.4 Strong NP-completeness

In Section 1.2.3 we described how approximation algorithms can provide a method of coping with NPO problems whose decision versions are NP-complete. However, for some NPO problems Π, we can find an exact algorithm for Π, called a *pseudo-polynomial-time algorithm*. A pseudo-polynomial-time algorithm for an NPO problem Π will, for any instance x of Π, solve Π in time polynomial in $|x|$ and $\max(x)$ (where $\max(x)$ denotes the value of the largest number occurring in x). An NPO problem is *pseudo-polynomial* if it admits a pseudo-polynomial-time algorithm. For example, it is well-known [56, Exercise

[5] These two lower bounds are examples of non-approximability results that follow from the PCP theorem [9, 8] (see [138, 7] for two commentaries on the PCP theorem and its consequences). The PCP revolution has led to an outpouring of non-approximability results similar to the two mentioned for CHROMATIC NUMBER and CLIQUE. Hence it is likely that the non-approximability results of [14, 201] for these two problems (under the assumption P$\neq$NP) will be superseded.

17.2-2] that the NPO problem KNAPSACK (whose decision version is problem MP9 of [92]) is solvable in pseudo-polynomial time using a dynamic programming algorithm.

However there are some NPO problems for which the existence of a pseudo-polynomial-time algorithm is unlikely. Given a decision version Π of an NPO problem and a polynomial p, Π_p denotes the problem obtained by restricting Π to only those instances x for which $\max(x) \leq p(|x|)$. A decision version Π of an NPO problem is *NP-hard in the strong sense* if there is some polynomial p (over the integers) such that Π_p is NP-hard. If, in addition, $\Pi \in$NP, then Π is *NP-complete in the strong sense* or *strongly NP-complete*. A strongly NP-complete problem has the property that it cannot be solved by a pseudo-polynomial-time algorithm unless P=NP. Garey and Johnson [91] give examples of strongly NP-complete problems, two being BIN PACKING (whose decision version is problem SP1 of [92]) and 3-PARTITION (problem SP15 of [92]).

In order to show that a decision version Π of an NPO problem is NP-hard in the strong sense, one might exhibit a specific polynomial p such that Π_p is NP-hard. Alternatively, the same result can be shown to hold by constructing a *pseudo-polynomial transformation* (see [91] for further details) from a known strongly NP-hard problem Π' to Π.

There is also a relationship between strongly NP-complete problems and one of the approximation classes of Section 1.2.3. Let Π be an NPO problem and suppose that, given an instance x of Π, the globally optimal measure of Π is bounded by a polynomial in both $|x|$ and $\max(x)$. Suppose further that the decision version of Π is strongly NP-complete. Then Π cannot be in FPTAS unless P=NP [91].

1.3 Minimaximal and maximinimal optimisation problems

In this section, we introduce the class of optimisation problems that we study in this thesis. Recall from Section 1.1 the informal definition of an optimisation problem. A minimaximal or maximinimal optimisation problem Π' may be obtained from an optimisation problem Π, called the *source* optimisation problem, by defining a partial order $\prec^x$ on the set of feasible solutions $\mathcal{F}(x)$ for a given instance x of Π. Minimaximal and maximinimal optimisation problems are of interest as they have a well-defined structure, in terms of this partial order concept. The instances of Π' are those of Π. A feasible solution of Π' is a member of $\mathcal{F}(x)$ that is locally optimal[6] with respect to $\prec^x$. Local optimality of a feasible solution $s \in \mathcal{F}(x)$ is a local property based on the position of s within a partial order hierarchy, rather than the position of s within a measure hierarchy (as per global optimality). The measure function for an instance of Π' is the restriction of that of Π to the feasible solutions of Π'. If the objective of Π is to maximise (respectively minimise) the measure over $\mathcal{F}(x)$, then the objective of Π' is to minimise (maximise) the measure over all elements of $\mathcal{F}(x)$ that are maximal (minimal) with respect to $\prec^x$.

[6] Given a partial order $\prec^x$ defined on the feasible solutions for a given instance x of an optimisation problem Π, a feasible solution y is *$\prec^x$-maximal*, or *maximal with respect to* $\prec^x$, if there is no feasible solution y' such that $y \prec^x y'$. Similarly y is *$\prec^x$-minimal*, or *minimal with respect to* $\prec^x$, if there is no feasible solution y' such that $y' \prec^x y$. If the goal of Π is to maximise (respectively minimise), then a feasible solution y is *$\prec^x$-optimal*, or *locally optimal with respect to* $\prec^x$, if y is $\prec^x$-maximal ($\prec^x$-minimal).

For example, consider the source optimisation problem MAXIMUM MATCHING. This problem takes a graph $G = (V, E)$ as an instance. The feasible solutions of G are the *matchings* in G (a subset E' of E is a matching if no two edges of E' are adjacent in G), the measure of a feasible solution is its cardinality, and the goal is to maximise. By considering the partial order of set inclusion, we define (as in the previous paragraph) the MINIMUM MAXIMAL MATCHING problem (whose decision version is problem GT10 of [92]). This problem is thus a minimaximal optimisation problem whose objective is to find, given a graph G, a minimum cardinality maximal matching for G, where a matching M is maximal if no proper superset of M is also a matching for G.

1.3.1 Minimaximal and maximinimal optimisation problems in the literature

There appears to be no previous study of minimaximal and maximinimal optimisation problems in general. However, the concept of *minimaximal and maximinimal graph parameters* has received attention. Informally, a minimaximal (respectively maximinimal) graph parameter is the minimum (maximum) cardinality over some set of graph-related structures that are maximal (minimal) with respect to some partial order. More formally, a minimaximal (respectively maximinimal) graph parameter may be defined as the solution to the evaluation version of a suitably defined minimaximal (maximinimal) graph optimisation problem. For example, the minimum maximal matching parameter[7] $\beta_1^-(G)$ is the solution to the evaluation version of MINIMUM MAXIMAL MATCHING, for a given graph G.

Some examples of minimaximal and maximinimal graph parameters are studied by Harary [107], who refers to them as *minimum maximal* and *maximum minimal* invariants. Harary states that for most minimum graph parameters there is a 'maximum minimal' graph parameter, and similarly for maximum graph parameters. However, he does not mention explicitly the concept of a partial order defined on the feasible solutions of the implicit optimisation problems. Rather, he stipulates that the definition of a maximum (respectively minimum) parameter should be altered, in order to accommodate the minimaximal (maximinimal) version. In particular, the modified feasible solutions are the maximal or minimal elements (with respect to some implicit partial order, usually set inclusion) of the usual feasible solutions. This has the drawback that a parameter such as the maximum matching parameter now assumes a less familiar definition as the 'maximum maximal matching' parameter. Also, the range of problems studied is restricted to those relating to graph theory. The paper by Hare et al. [111] follows a similar approach.

In addition to [107, 111], there exist references in which the general concept of minimaximal and maximinimal graph parameters is mentioned in passing. Peters et al. [183] endorse the technique that Harary discusses, before considering minimaximal and maximinimal graph parameters related to vertex and edge connectivity[8]. Majumdar [163] con-

[7] See the footnote on Page 56 for a discussion of our notation for certain graph parameters in this thesis.
[8] These parameters are defined in Section 5.5.

structs a framework for covering and packing parameters of graphs in terms of neighbourhood hypergraphs, and discusses how this framework might accommodate minimaximal and maximinimal graph parameters. McRae [165] introduces a general construction yielding NP-completeness results for graph parameters when restricted to the classes of bipartite and chordal graphs. She demonstrates the applicability of her construction by giving many examples, which include several minimaximal and maximinimal graph parameters, some of which had not been previously studied. Jacobson and Peters [131] demonstrate the existence of a graph parameter γ whose implicit optimisation problem is NP-hard in a certain graph class, yet γ has a maximinimal counterpart Γ that is polynomial-time solvable in the same graph class of graphs. This leads them to ask the question of why it can be difficult to determine the maximum (respectively minimum) value of a parameter, yet relatively 'easy' to solve the corresponding minimum maximal (maximum minimal) problem, and in which other cases this occurs. (We shall return to this question in Section 9.6.)

There are a significant number of references relating to the study of *particular* minimaximal and maximinimal optimisation problems. For example, MAXIMUM MINIMAL DOMINATING SET [39], MINIMUM MAXIMAL INDEPENDENT SET [104] and MINIMUM MAXIMAL MATCHING [92, problem GT10] are just a few minimaximal or maximinimal optimisation problems that have received attention (we define these problems in Chapter 4). However, the majority of minimaximal and maximinimal optimisation problems in the literature are related to graph theory, and the dominant partial order is that of set inclusion. There are few examples in the literature of minimaximal or maximinimal optimisation problems not pertaining to graph theory. However the SHORTEST MAXIMAL COMMON SUBSEQUENCE and LONGEST MINIMAL COMMON SUPERSEQUENCE problems of Fraser et al. [81] are two such examples. (These problems are defined in Section 7.5.) In addition, there are few examples in the literature of minimaximal or maximinimal optimisation problems defined in terms of a partial order besides that of set inclusion. However we show in Section 3.2 that ACHROMATIC NUMBER (whose decision version is problem GT5 of [92]) is in fact a maximinimal optimisation problem, if a suitable partial order is defined on the set of all proper colourings for a graph. In addition, McRae [165] studies the MINIMUM 2-MAXIMAL INDEPENDENT SET and MINIMUM 2-MAXIMAL MATCHING problems, which are minimaximal optimisation problems defined in terms of the partial order of $(1, 2)$-*replacement* (this partial order is defined in Section 2.4, and the minimaximal optimisation problems mentioned are defined in Section 5.3).

1.3.2 Terminology for minimaximal and maximinimal optimisation problems

The words 'minimaximal' and 'maximinimal' abbreviate 'minimum maximal' and 'maximum minimal' respectively. This terminology is used by Peters et al. [183] and we adopt this nomenclature when referring to minimaximal and maximinimal concepts in general. However, in order to be consistent with the literature, we refer to particular minimaximal and maximinimal optimisation problems (such as MINIMUM MAXIMAL MATCHING) without using this abbreviation. Unfortunately, further abbreviation of the words 'minimaximal' and 'maximinimal' to terms such as 'minimax' and 'maximin' respectively, would be undesirable. This is because *minimax* optimisation problems already exist in the literature (see, for example [66]). These problems, distinct from the class of problems that we study in this thesis, involve minimising the maximum value, or maximising the minimum value, over some set of structures.

The words 'minimal' and 'maximal', which we use to denote local optimality, have often been used to denote global optimality: for example, Liu [156, p.287] refers to maximum cardinality matchings as 'maximal' matchings. However, the distinction in the terminology was recognised prior to Liu's book, an example being the paper of Norman and Rabin [174], in which minimum and minimal covers are referred to in the sense used here, and similarly for maximum and maximal matchings.

Some minimaximal and maximinimal graph parameters are referred to in the literature as 'lower' and 'upper' parameters respectively. Here, the words 'lower' and 'upper' refer to the measure hierarchy rather than the partial order hierarchy. For example, the *minimum maximal irredundance* and *maximum minimal domination* numbers are referred to as the *lower irredundance* and *upper domination* numbers respectively [54]. However, perhaps unfortunately, some minimum and maximum graph parameters are also referred to in this way. For example, the *maximum irredundance* and *minimum domination* numbers are referred to as the *upper irredundance* and *lower domination* numbers respectively [54]. This terminology therefore removes the distinction between 'minimum' and 'minimum maximal', and between 'maximum' and 'maximum minimal', in the name of a graph parameter. We therefore use the full terms in this thesis. Sometimes, however, even this is not sufficient. It is clear that the definition of a minimaximal or maximinimal optimisation problem is dependent on the choice of partial order associated with the problem. For many minimaximal and maximinimal optimisation problems, the inherent partial order is evident from the source optimisation problem definition. However, this is not always the case. For example, by defining two partial orders on the set of all feasible solutions to the CHROMATIC NUMBER problem, as in Chapter 3, we obtain two maximinimal optimisation problems that are quite distinct. Thus in some cases, to avoid confusion, it is necessary to include the partial order name in the name of the minimaximal or maximinimal optimisation problem, or to use some other terminological device to distinguish such problems.

1.3.3 Framework for minimaximal and maximinimal optimisation problems

We introduce, in Chapter 2, a framework for defining minimaximal and maximinimal optimisation problems, based on this unifying concept of a partial order defined on the feasible solutions of a source optimisation problem, for a given instance. The framework does not involve changing the definition of the source optimisation problem, and the minimaximal and maximinimal optimisation problems that may be defined using it are not restricted to graph theory. To accompany this framework, we introduce, in Section 2.4, a wide range of partial orders (including the partial orders of set inclusion and *partition merge* that are implicit in Harary's paper [107]) that may be used in order to formulate minimaximal and maximinimal optimisation problems. Every optimisation problem that we have encountered in the literature that is essentially a 'minimum maximal' or 'maximum minimal' problem (with respect to some implicit partial order) may be defined using our framework. In addition, many interesting new examples may be formulated.

We define a variety of source optimisation problems, and using the partial orders in our collection, we formulate many examples of minimaximal and maximinimal optimisation problems in Chapters 3-7. We study these problems from the point of view of algorithmic complexity. In the next section, we define concepts and survey literature relating to subject areas of the source optimisation problems that appear in these chapters.

1.4 A tour of source optimisation problems

1.4.1 Optimisation problems in graph theory

Many of the optimisation problems that we consider in this thesis relate to graph theory. A general introduction to the subject is provided by Berge [15, 17] and Harary [106], for example. These are three classical texts; a more recent work is [35]. Additionally, Gibbons [96] and Golumbic [99] provide an algorithmic emphasis. Standard graph-theoretic terminology used in the following but not defined may be found in Harary [106]. All graphs in this thesis are understood to be finite, undirected graphs with no loops and no multiple edges. Notation for graph parameters follows that of Harary [106] and Haynes et al. [115] unless otherwise indicated.

Graph colouring is an important area of graph theory. Given a graph $G = (V, E)$, we call a partition of V a *colouring* of V. A set of vertices $V' \subseteq V$ is *independent* if no two vertices of V' are adjacent in G. A *proper colouring* is a partition of V into independent sets, or *colours*. If G has a proper colouring of k colours, then G is said to be *k-colourable*. The *chromatic number*, $\chi(G)$, is the minimum number of colours over all proper colourings of G. The CHROMATIC NUMBER problem is the problem of finding a proper colouring of a given graph G using $\chi(G)$ colours. Jensen and Toft [132] review the field of graph colouring in their first chapter, before going on to discuss over two hundred open problems relating to graph colouring.

In Chapter 3, we define partial orders on the set of all proper colourings of a graph, thereby obtaining two maximinimal graph colouring optimisation problems. The partial order of *partition merge* (defined in Section 2.4) gives rise to the ACHROMATIC NUMBER problem (whose decision version is problem GT5 of [92]). This is the problem of finding a proper colouring of a given graph G using the maximum number of colours such that every pair of colours occurs at the endpoints of some edge of G. A natural refinement of partition merge is the partial order of *partition redistribution* (also defined in Section 2.4), giving rise to the B-CHROMATIC NUMBER problem. This is the problem of finding a proper colouring of a given graph G using the maximum number of colours such that, for each colour i, there is a distinguished vertex of colour i that is adjacent to a vertex of every colour $j \neq i$. Whilst ACHROMATIC NUMBER has been studied for over thirty years, B-CHROMATIC NUMBER is new. We show that the decision version of B-CHROMATIC NUMBER is NP-complete in arbitrary graphs and also bipartite graphs. However, we prove that B-CHROMATIC NUMBER is solvable is polynomial-time for trees, in contrast with ACHROMATIC NUMBER.

Two other important notions in graph theory are those of covering and independence [106, Chapter 10][157]. Let $G = (V, E)$ be a graph. A vertex v is defined to cover itself, all edges incident on v and all vertices adjacent to v. An edge $\{u, v\}$ is said to cover itself, vertices u and v, and all edges incident on u or v. Two elements of $V \cup E$ are *independent* if neither covers the other. A *vertex cover* is a subset S of V that covers E, a *dominating set* is a subset S of V that covers V, an *edge cover* is a subset S of E that covers V (assuming that G has no isolated vertices) and an *edge dominating set* is a subset S of E that covers E. A subset C of $V \cup E$ that covers all vertices and edges of G is said to be a *total cover* for G. An *independent set* is a subset S of V whose elements are pairwise independent, and a *matching* is a subset S of E whose elements are pairwise independent. A subset M of $V \cup E$ whose elements are pairwise independent is said to be a *total matching* for G.

In Chapter 4, we consider graph theoretic source optimisation problems and their minimaximal and maximinimal counterparts, defined using the partial order of set inclusion. Such problems associated with vertex, edge and total covers, and independent sets, matchings and total matchings in graphs are considered in Section 4.2. We survey complexity results for these optimisation problems and obtain a number of new results. We prove NP-completeness for the decision versions of the problems of finding a maximum minimal total cover in planar graphs, finding a minimum maximal total matching in bipartite and chordal graphs, and finding a minimum maximal independent set in planar cubic[9] graphs. In addition, we demonstrate that the computational complexities of the problems of finding a maximum minimal vertex cover, a maximum minimal edge cover and a maximum total matching are identical to those of the problems of finding a minimum maximal independent set, a minimum dominating set and a minimum edge dominating set respectively, over all graph classes.

Various other graph-theoretic notions are related to covering and independence. For a graph $G = (V, E)$ and vertex $v \in V$, define the *open neighbourhood* of v to be $N(v) = \{w \in V : \{v, w\} \in E\}$, and define the *closed neighbourhood* of v to be $N[v] = N(v) \cup \{v\}$. For a set $S \subseteq V$, the open neighbourhood of S is the set $N(S) = \cup_{v \in S} N(v)$ and the closed neighbourhood of S is the set $N[S] = \cup_{v \in S} N[v]$. Thus, a dominating set is a subset S of V such that $N[S] = V$. Domination in graphs was first studied by Ore [175] and since then, the field has grown swiftly. Following a volume of *Discrete Mathematics* devoted to the topic of domination [122], two books on the subject have recently been published [114, 115].

The topic of *irredundance* in graphs has also received much attention. A set of vertices

[9] A graph is *cubic* if every vertex has degree 3.

S in a graph $G = (V, E)$ is *irredundant* if, for every vertex v in S, $N[v] \backslash N[S \backslash \{v\}] \neq \emptyset$. Cockayne, Hedetniemi and Miller [49] began the study of irredundance, and literature concerning this concept is surveyed by Hedetniemi et al. [123].

In addition, a *strong stable set* is a subset S of V such that $|N[v] \cap S| \leq 1$ for all $v \in V$. Thus, S is a strong stable set implies that S is an independent set. A *clique* is a subset S of V such that every pair of vertices in S is an edge in G. A *total dominating set* is a subset S of V such that $N(S) = V$.

In Section 4.3, we consider source optimisation problems and their minimaximal and maximinimal counterparts (defined using the partial order of set inclusion) relating to strong stable sets, cliques, domination, total domination, edge domination and irredundance. We survey complexity results for these optimisation problems and obtain several new results. We prove NP-completeness for the decision version of the problems of finding a minimum maximal strong stable set in planar graphs of maximum degree 3, finding a minimum maximal clique in general graphs, and finding a minimum total dominating set in planar cubic graphs.

In Chapter 5, we study minimaximal and maximinimal graph optimisation problems defined in terms of the partial order of $(k-1, k)$*-replacement* $(k \geq 1)$. (This partial order is defined formally in Section 2.4.) We apply this partial order to the set of all independent sets $\mathcal{F}(G)$, for a given graph $G = (V, E)$. An independent set that is maximal with respect to the partial order of $(k-1, k)$-replacement is said to be k*-maximal*. Informally, an independent set $S \subseteq V$ is k*-maximal* $(k \geq 1)$ if the removal of any $r-1$ vertices from S, together with the addition of any r vertices from $V \backslash S$ (for any $r \leq k$), results in a non-independent set. The concept of k-maximal independent sets in graphs was introduced by Bollobás et al. [22], and is a refinement of the notion of a maximal (with respect to the partial order of set inclusion) independent set. We formulate minimaximal optimisation problems whose objective is to find the minimum cardinality k-maximal independent set in a graph. Restricting attention to the case $k = 2$, we prove that the decision problem related to finding a minimum 2-maximal independent set in graph is NP-complete, even for planar graphs of maximum degree 3. In the case of trees, we give a linear time algorithm for computing the cardinality of a minimum 2-maximal independent set.

1.4.2 Fractional graph optimisation problems

Over the last twenty years or so, graph theorists have begun to generalise integer-valued graph parameters to their real-valued counterparts, leading to the concept of *fractional* graph parameters. For example, given a graph $G = (V, E)$, let $\gamma(G)$ denote the *minimum domination number* of G, i.e., the smallest cardinality of a dominating set of G. This parameter has a real-valued generalization, called the *minimum fractional domination number* of G, denoted $\gamma_f(G)$. We may define $\gamma_f(G)$ as follows. A function $f : V \longrightarrow [0, 1]$ is said to be *dominating* for G if, for every $v \in V$, $f(N[v]) \geq 1$, where $f(S) = \sum_{v \in S} f(v)$, for $S \subseteq V$. The value $f(V)$ is said to be the *weight* of f. We then define

$$\gamma_f(G) = \min\{f(V) : f \text{ is a dominating function for } G\}.$$

Note that, although arbitrary real-valued weights in the range $[0, 1]$ may be assigned to the vertices of G, the value $\gamma_f(G)$ is always rational (this follows from the fact that $\gamma_f(G)$ can be expressed as the solution of a linear program – see for example [198]). Thus the parameter γ_f is referred to as a *fractional* parameter. It is clear that many other fractional graph parameters may be formulated in this way. Recently, a research monograph [194] has been published on the subject, and Chapter 3 of [114] is devoted to fractional domination and related fractional graph parameters.

We may recover the definitions of integer-valued graph parameters from their fractional variants. For instance, the above definition of γ_f may be altered, in order to recover the definition of γ, by insisting that $\mathrm{ran}(f) \subseteq \{0, 1\}$ for any dominating function f.

In Chapter 6, we define the concept of *fractional graph optimisation problems.* Solving a fractional graph optimisation problem involves computing the value of an implicit fractional graph parameter. We demonstrate why our usual definition of an optimisation problem is not sufficient to define fractional graph optimisation problems. Using an alternative definition, we show how an appropriate framework for minimaximal and maximinimal fractional graph optimisation problems may be defined, using the *partial order on functions* (defined in Section 6.2). We formulate several examples of such problems using the revised framework, and show that the globally optimal measure (the solution to the evaluation version of the minimaximal or maximinimal fractional graph optimisation problem concerned) is computable, has rational values, and is attained by some function of compact representation which satisfies the feasibility constraint for the minimaximal or maximinimal fractional graph optimisation problem concerned. We also survey algorithmic results relating to the source fractional graph optimisation problems and their minimaximal or maximinimal counterparts that we define. The optimisation problems that we study in Chapter 6 are concerned with domination, total domination, packing, irredundance and vertex and edge covering and independence in graphs.

1.4.3 Optimisation problems concerning strings

Some of the optimisation problems that we consider in this thesis are concerned with strings, and we take this opportunity to introduce some concepts to be used in the following chapters. Gusfield [103], Crochemore and Rytter [61] and Stephen [202] are three recent books concerned with the subject of string algorithms.

Given a set Σ of symbols (an *alphabet*), a *string* over Σ is a function $s : X \longrightarrow \Sigma$, where $X = \varnothing$, $X = \mathbb{Z}^+$, or $X = \{1, 2, \ldots, n\}$, for some $n \in \mathbb{Z}^+$. Σ^* denotes the set of all strings, finite or infinite in length, composed of characters from Σ. The definitions in the remainder of this section apply to finite strings only. The *length* of s, denoted $|s|$, is $|X|$. Let $s : X \longrightarrow \Sigma$ and $s' : Y \longrightarrow \Sigma$ be two strings. We say that s is a *subsequence* of s', denoted $s \underline{\ll} s'$, if $X = \varnothing$, or there is a strictly increasing function $f : X \longrightarrow Y$ such that $s(x) = s'(f(x))$ for all $x \in X$. If $s \underline{\ll} s'$, and in addition, $X \neq \varnothing$ implies that $f(X)$ is an interval, then s is a *substring* of s', denoted $s \overline{\underline{\ll}} s'$. If $s \underline{\ll} s'$ and $s \neq s'$, then we say that s is a *proper subsequence of* s', denoted $s \ll s'$. Similarly, if $s \overline{\underline{\ll}} s'$ and $s \neq s'$, then we say that s is a *proper substring of* s', denoted $s \overline{\ll} s'$. Also, s is a *(proper) supersequence* of s' if and only if s' is a (proper) subsequence of s, and s is a *(proper) superstring* of s' if and only if s' is a (proper) substring of s.

Given a set S of strings, and a string s, we say that s is a *common subsequence* of S, denoted $s \underline{\ll} S$, if $s \underline{\ll} t$ for every $t \in S$. The definitions of *common supersequence* , *common substring* and *common superstring* , denoted $S \underline{\ll} s$, $s \underline{\lll} S$ and $S \underline{\lll} s$ respectively, are analagous.

In Chapter 7, we define minimaximal and maximinimal counterparts of the source optimisation problems concerned with finding a longest common subsequence and a shortest common supersequence of a set S of k strings. Here, a string $s \underline{\ll} S$ is maximal if there does not exist a string t such that $s \ll t$ and $t \underline{\ll} S$; the definition of minimality is analagous. We also define minimaximal and maximinimal counterparts of the source optimisation problems related to finding a longest common substring and a shortest common superstring of a set S of k strings. Here, a string $s \underline{\lll} S$ is maximal if there does not exist a string t such that $s \lll t$ and $t \underline{\lll} S$; the definition of minimality is analagous. We survey complexity results for each of the problems mentioned in this paragraph, and give a polynomial-time algorithm for finding a shortest maximal common substring of a set S of k strings.

1.4.4 Some other optimisation problems

In this section we introduce several additional optimisation problems that have received attention in the literature. These problems belong to the Garey and Johnson [92] subject categories of Network Design, Sets and Partitions, Data Storage, Mathematical Programming, and Logic.

The objective of the LONGEST PATH problem is to find, given a graph $G = (V, E)$ with length $l(e) \in \mathbb{Z}^+$ for each $e \in E$, and distinguished vertices s, t in V, a simple path in G (i.e. a path that visits each vertex at most once) from s to t of maximum length.

The aim of the MAXIMUM 3D-MATCHING problem is to find, given three disjoint sets W, X, Y, each of size q, and a subset $M \subseteq W \times X \times Y$, a maximum matching for M, i.e. a subset M' of M of largest size such that no two elements of M' agree in any co-ordinate.

For the MINIMUM TEST SET problem, an instance is a collection C of subsets of a finite set S. We seek a minimum *test set* for S, i.e. a subset C' of C of minimum size such that, for each pair of distinct elements $u, v \in S$, there is some $c \in C'$ such that $|\{u, v\} \cap c| = 1$.

In the BIN PACKING problem, we are given a finite set U of *items*, each with an associated positive integer (the *size*), and a positive integer (*bin capacity*). The objective is to construct a *bin packing of* U, i.e. to partition U into sets, or *bins*, where the total size of items in each bin does not exceed the bin capacity, such that the number of bins used is minimum over that of every bin packing of U.

For the KNAPSACK problem, we are given a finite set U of *items*, each with two associated positive integers (the *weight* and *value*), and a positive integer (*knapsack capacity*). The objective is to find a subset U' of U such that the total weight of items in U' does not exceed the knapsack capacity (i.e. to construct a *knapsack packing*), and the total value of items in the knapsack packing is maximum over that of every knapsack packing for U.

The SATISFIABILITY decision problem (problem LO1 of [92]) takes a set U of variables and a collection C of clauses over U (see Section 7.7 for precise definitions of logic-related terminology used in this section but not defined) and asks whether there is a satisfying truth assignment for C. SATISFIABILITY has a range of variants which may be expressed as optimisation problems, two of which now follow.

The objective of MAXIMUM ONE-IN-THREE 3SAT is to find, given a set U of variables and a collection C of clauses over U such that each clause in C has size three, a truth assignment that simultaneously satisfies exactly one literal from each clause in C', for some $C' \subseteq C$, and no literals from every clause in $C \backslash C'$, where C' is as large as possible.

The aim of MAXIMUM 2-SAT is to find, given a set U of variables and a collection C of clauses over U such that each clause in C has size two, a truth assignment that simultaneously satisfies as many clauses of C as possible.

The decision version of each of the above optimisation problems is NP-complete: see problems ND29, SP1, SP6, SR1, MP9, LO3 and LO5 of [92] respectively. In Chapter 7, we define minimaximal or maximinimal counterparts of each of the above problems, using appropriate partial orders from our collection, and prove that the minimaximal or maximinimal optimisation problem concerned is NP-complete.

This chapter continues with a review of issues regarding the approximation of NP-hard optimisation problems using local search in Section 1.5. Local optimality of feasible solutions is an important theme in this thesis. In Section 1.6, we preview two chapters in this thesis that are closely connected with local search.

1.5 Complexity of local search: literature review

1.5.1 Background and definitions

The NP-completeness of the decision version of an optimisation problem Π naturally leads to the question of the approximability properties of Π. One of the most tried and tested methods of obtaining approximate solutions to hard optimisation problems is called *local search*, described by Papadimitriou and Steiglitz [179, Chapter 19]. Let Π be an optimisation problem, and for a given instance x of Π, let $\mathcal{F}(x)$ be the feasible solutions, and let $m(x, \cdot)$ denote the measure function. Let OPT be max or min according to the goal of Π. A *neighbourhood function* for the instance x of Π is a map

$$N_x : \mathcal{F}(x) \longrightarrow \mathbb{P}(\mathcal{F}(x))$$

which assigns a *neighbourhood* of feasible solutions to any feasible solution $s \in \mathcal{F}(x)$. Given a feasible solution t, the following subroutine *improve* returns a neighbour of better measure (a *local improvement* of t), or 'none' if no such neighbour exists:

$$improve(t) = \begin{cases} \text{any}^{10}\ s \in N_x(t) \text{ such that } m(x,s) > m(x,t), \text{if } s \text{ exists,} \\ \qquad \text{where OPT} = \max \\ \text{any } s \in N_x(t) \text{ such that } m(x,s) < m(x,t), \text{if } s \text{ exists,} \\ \qquad \text{where OPT} = \min \\ \text{`none', otherwise.} \end{cases}$$

The *standard local search algorithm* begins with an arbitrary feasible solution s, and repeatedly calls the subroutine *improve*, to find successive neighbours of better measure, until 'none' is returned. The feasible solution t obtained by the standard local search algorithm is called a *locally optimal* solution. Since the set of feasible solutions of Π for a given instance is always finite, the standard local search algorithm is guaranteed to terminate.

Clearly, the local optimum output by the standard local search algorithm depends on the neighbourhood structure imposed. A more complicated neighbourhood structure may yield a higher quality of locally optimal solutions, but the standard local search algorithm may consequently take longer to converge. Hence there must be a trade-off between these two issues when defining the neighbourhood function.

1.5.2 Local search and the Travelling Salesman Problem

One NP-hard optimisation problem, to which the technique of local search has been applied to yield approximate solutions, is the symmetric TRAVELLING SALESMAN PROBLEM (TSP) (whose decision version is problem ND22 of [92]), which may be defined as follows.

We are given a set $C = \{1, 2, \ldots, n\}$ of *cities*, such that, for each $i, j \in C$ $(i \neq j)$, there is an associated positive integer $d(i,j)$ (a *distance*) between i and j. (Note that in the symmetric case, $d(i,j) = d(j,i)$ for every such pair.) The objective is to construct a *travelling salesman tour* of C of minimum cost, i.e. to find a permutation π of $1, 2, \ldots n$ such that

$$\sum_{i=1}^{i=n} d(\pi(i), \pi(i+1)) + d(\pi(n), \pi(1))$$

is minimum over all tours of C.

Several neighbourhood structures have been imposed on the feasible solutions for a given instance of TSP. In the *k-opt* neighbourhood ($k \geq 2$) [154], two tours are neighbours if and only if they differ in at most k edges. A more complicated structure, known as the *λ-change* or *Lin-Kernighan* neighbourhood has been formulated [155]. Unfortunately, *any* local search algorithm for TSP, having polynomial time complexity per iteration, can generate solutions arbitrarily far from the optimum, unless P=NP [178]. However, despite this, local search heuristics for TSP perform quite well in practice. In particular, the 3-

[10] The quality of the local optimum obtained by the *improve* procedure may be enhanced by imposing some strategy for choosing a neighbour of better measure, in the case that there is more than one such neighbour; such a strategy is known as a *pivoting rule*. In the case of the standard local search algorithm, the choice is nondeterministic.

opt neighbourhood outperforms 2-opt, but the Lin-Kernighan neighbourhood has been described as the 'champion' of local search heuristics. See [135] for further details.

1.5.3 Local search can be provably successful

Despite the empirical success of the technique of local search for generating approximate solutions to TSP, the result of Papadimitriou and Steiglitz [178] rules out the existence of any non-trivial analytical performance guarantee for a polynomial-time local search approximation algorithm for TSP. However, there are several NP-hard optimisation problems to which the technique of local search has been applied successfully, so as to yield, in some cases, the best known performance guarantee for any approximation algorithm currently known for that problem.

One example is MINIMUM DEGREE SPANNING TREE (whose decision version is problem ND1 of [92]), where the objective is to find, given a graph G, a spanning tree for G, whose maximum degree is minimum over that of all spanning trees for G. Let Δ^* denote the maximum degree of a globally optimal spanning tree for this problem. Fürer and Raghavachari [86] show that local search techniques yield an approximation strategy that computes a spanning tree for G with maximum degree either Δ^* or $\Delta^* + 1$. Their result also carries over to the MINIMUM DEGREE STEINER TREE problem, in which one specifies a *required* set of vertices, D, and asks for the maximum degree, over that of all trees which span at least the set D, to be minimised.

Another example is the MAXIMUM LEAF SPANNING TREE problem (whose decision version is problem ND2 of [92]), where the objective is to find, given a graph G, a spanning tree T for G that has the maximum number of *leaves* (i.e. vertices whose degree in T is 1) over all spanning trees for G. Lu and Ravi [158] define two neighbourhoods on the set of all spanning trees for a graph, and show that locally optimal solutions with respect to these neighbourhoods give rise to spanning trees with at worst one fifth and one third[11] of the globally optimal number of leaves, respectively.

Finally, Halldórsson [105] uses local search techniques to obtain the current best performance guarantees for approximating the MAXIMUM INDEPENDENT SET problem (whose decision version is problem GT20 of [92]) in k-*claw-free* graphs and in graphs whose maximum degree is some fixed constant B.

1.5.4 PLS-completeness

For many optimisation problems, the range of values that the measure function can take is bounded by a polynomial in the input, as for example in the MAXIMUM LEAF SPANNING TREE problem described above. Hence, the standard local search algorithm is bound to terminate in a polynomial number of steps (assuming polynomial time complexity per

[11] The local search algorithm yielding a performance guarantee of 3 has $O(n^7)$ time complexity. Lu and Ravi [159] demonstrate an alternative approach which yields a faster (almost linear-time) algorithm with the same performance guarantee. More recently, Solis-Oba [199] obtained a 2-approximation algorithm for the problem.

iteration). However, for other optimisation problems, such as TSP, the range of values that the measure function can take is exponential.

Hence a central issue is to determine, given an instance x of an optimisation problem Π, whether we can find a locally optimal solution for Π with respect to N_x in polynomial time. To formalise this question of finding locally optimal solutions, Johnson et al. [136] define the class PLS (*polynomial local search*) of search problems. The components of a search problem Π in PLS consist of a set of instances $\mathcal{I}$, and, given an instance $x \in \mathcal{I}$, a set of feasible solutions $\mathcal{F}(x)$, a measure function which assigns a natural number to any element of $\mathcal{F}(x)$, a goal (either max or min), and a neighbourhood function $N_x : \mathcal{F}(x) \longrightarrow \mathbb{P}(\mathcal{F}(x))$. Furthermore, a problem in PLS satisies the following properties:

1. We can produce an initial feasible solution in polynomial time.
2. We can compute the measure of a feasible solution in polynomial time.
3. Given a feasible solution s, we can, in polynomial time, test whether s is locally optimal, or else find a neighbour of better measure (i.e. the subroutine *improve(s)* runs in polynomial time).

Given an instance $x \in \mathcal{I}$, the objective of Π is to find a locally optimal solution with respect to N_x.

Johnson et al. [136] define *PLS-reductions* as follows. Let Π_1, Π_2 be two search problems in PLS. For $i = 1, 2$, let $\mathcal{I}_i$ be the set of instances of Π_i, and for $x \in \mathcal{I}_i$, let $\mathcal{F}_i(x)$ be the feasible solutions for the instance x of Π_i, and let $N_{i,x}$ be the neighbourhood function for Π_i. Π_1 is *PLS-reducible* to Π_2 if there exist two polynomial-time computable functions f, g such that:

1. $f : \mathcal{I}_1 \longrightarrow \mathcal{I}_2$.
2. For any $x \in \mathcal{I}_1$, $g(\cdot, x) : \mathcal{F}_2(f(x)) \longrightarrow \mathcal{F}_1(x)$.
3. For any $x \in \mathcal{I}_1$, if $s \in \mathcal{F}_2(f(x))$ is locally optimal with respect to $N_{2,f(x)}$, then $g(s, x)$ is locally optimal with respect to $N_{1,x}$.

The pair $\langle f, g \rangle$ is said to be a *PLS-reduction* from Π_1 to Π_2. A problem Π in PLS is *PLS-complete* if every problem in PLS is PLS-reducible to Π. Thus a PLS-complete problem Π has the property that if a local optimum can be found in polynomial time for Π then local optima can be found in polynomial time for all problems in PLS.

Lueker [160] shows that we may construct instances of the travelling salesman problem, with the 2-opt neighbourhood, such that, by choosing a certain series of improvements, convergence of the initial solution to a local optimum takes an exponential number of steps. Also, Johnson et al. [136] exhibit a problem Π in PLS whose *standard local search algorithm problem* (i.e., given an instance x of Π, find the solution that would be output by the standard local search algorithm) is NP-hard. In each case, the standard local search algorithm is unlikely to yield an efficient method of finding a locally optimal solution for the PLS problem concerned. However, this does not imply that the PLS problem in each case is NP-hard, since the PLS problem asks for *any* locally optimal solution. Indeed, if there were such a problem, then NP=co-NP[12] would hold [136], a situation that is considered to be unlikely.

Johnson et al. [136] show that $\mathrm{P}_S \subseteq \mathrm{PLS} \subseteq \mathrm{NP}_S$, where P_S and NP_S are the search problem analogues of P and NP, respectively. Given the comments in the previous paragraph, it is unlikely that PLS=NP_S. Also, if PLS=P_S, then any polynomial-time algorithm for a PLS-complete problem would give another proof that LINEAR PROGRAMMING (whose decision version is problem OPEN9 of [92]) is in P, since LINEAR PROGRAMMING with the Simplex neighbourhood is in PLS [136]. It is possible that the class PLS lies properly between the classes P_S and NP_S.

Papadimitriou [177] shows that TSP under the Lin-Kernighan neighbourhood, is PLS-complete. Also, Krentel [149] shows that there is a finite (large) $k > 3$ such that TSP under the k-opt neighbourhood is PLS-complete. It is open as to whether TSP under the 2-opt neighbourhood is PLS-complete, though Fischer and Torenvliet [79] provide strong evidence that this is the case.

Yannakakis [211, 212] provides a more detailed survey of complexity issues relating to local search.

1.6 Local search-related issues in this thesis

1.6.1 Minimaximal and maximinimal reductions

In the previous section, we defined the PLS reduction of Johnson et al. [136] – a reduction that preserves the local optimality of feasible solutions in a certain sense. This reduction is relevant to our study of the complexity of minimaximal and maximinimal optimisation problems. For, the notion of local optimality with respect to a neighbourhood relation is closely related to the concept of local optimality with respect to a partial order, when both of these structures are defined on the feasible solutions of an optimisation problem for a given instance (this relationship is discussed in further detail in Section 2.5).

In Chapter 8, we present conditions under which a Turing reduction from an optimisation problem, Π_1, to another, Π_2, is also a Turing reduction from Π'_1 to Π'_2, where Π'_i is a minimaximal or maximinimal version of Π_i ($i = 1, 2$). Turing reductions satisfying these additional constraints are called *MM-reductions*, standing for *minimaximal / maximinimal reductions*. The definition of an MM-reduction bears similarities to that of the PLS reduction, in that there is a local optimality-preserving condition to be satisfied.

Several examples of MM-reductions are given in Chapter 8.

1.6.2 Testing a feasible solution for local optimality, and finding locally optimal feasible solutions

Formulation of the notion of completeness in the class PLS revolves around the fact that, for some optimisation problems and neighbourhood structures, testing a feasible solution

[12] A decision problem Π is in co-NP if and only if the complement of Π, Π^C, is in NP. Π^C is a decision problem which has a 'yes' answer if and only if Π has a 'no' answer, for a given instance.

for local optimality is easy, but finding a locally optimal feasible solution is hard.

In Sections 9.2-9.4, we consider the questions of how we may test a feasible solution for local optimality, and how we may find a locally optimal feasible solution, with respect to a partial order defined on the feasible solutions of an optimisation problem for a given instance. We show that when the partial order of *partition redistribution* (defined in Section 2.4) is imposed on the feasible solutions of BIN PACKING for a given instance x, the resulting problem of testing a feasible solution for minimality is NP-hard, whereas the problem of finding a minimal feasible solution is polynomial-time solvable.

In the case of CHROMATIC NUMBER, we consider two families of partial orders defined on the set of all proper colourings of a given graph G. We investigate the problems of testing a proper colouring for minimality, and finding proper graph colourings that are minimal, with respect to these partial orders. We also consider the complexity of the associated maximinimal optimisation problems in each case. Our algorithmic results for testing and finding show where the thresholds between polynomial-time solvability and NP-hardness lie, within the hierarchy of problems corresponding to the two partial order families. In particular, we show that there is a partial order that may be defined on the set of all proper colourings of a graph, such that both the problems of testing a proper colouring for minimality, and finding a minimal proper colouring are NP-hard.

1.7 Other general issues connected with minimaximal and maximinimal optimisation problems

In Section 9.5, we consider an alternative definition of our framework, showing that in some cases, a partial order defined on the set of all possible (not necessarily feasible) solutions of an optimisation problem gives rise to the same maximal and minimal solutions as would be obtained from defining the partial order concerned on the feasible solutions. We show that this holds for most, but not all, of the optimisation problems defined in this thesis.

Finally, in Section 9.6, we present some general conclusions and directions for further study relating to minimaximal and maximinimal optimisation problems.

Chapter 2

Framework for minimaximal and maximinimal optimisation problems

2.1 Introduction

In this chapter we present a framework for minimaximal and maximinimal optimisation problems, based on the concept of a partial order defined on the feasible solutions $\mathcal{F}(x)$, for a given instance x of a source optimisation problem Π. We begin with several definitions relating to optimisation problems in Section 2.2. In Section 2.3 we define some general concepts concerned with partial orders, leading to the definition of our framework for minimaximal and maximinimal optimisation problems. To accompany this framework, we present a range of partial orders in Section 2.4, which feature in the definition of minimaximal and maximinimal optimisation problems in the following chapters. Finally, in Section 2.5, we discuss the relationships between the concept of a partial order within the context of our framework for minimaximal and maximinimal optimisation problems, and the concept of a neighbourhood relation within the context of local search.

2.2 Formal definition of an optimisation problem

In this section we present several definitions relating to optimisation problems. We begin with a formal definition of an optimisation problem, which is adapted from Bovet and Crescenzi [24, Section 6.1].

Definition 2.2.1 ([24]) An *optimisation problem* Π is a tuple $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, such that:

1. $\mathcal{I}$ is a set of words that encode instances of the problem.

2. $\mathcal{U}$ is a function that maps an instance $x \in \mathcal{I}$ into a non-empty finite set of words that encode the *universal set of possible solutions* of x.

3. π is a predicate such that, for any instance $x \in \mathcal{I}$ and any possible solution $y \in \mathcal{U}(x)$, $\pi(x, y)$ if and only if y is a *feasible solution* (we assume that at least one feasible solution of x exists). We denote the feasible solutions of a given instance $x \in \mathcal{I}$ by $\mathcal{F}(x)$, thus
$$\mathcal{F}(x) = \{y \in \mathcal{U}(x) : \pi(x, y)\}.$$

4. m is a function, called the *objective* or *measure function*, that associates with any instance $x \in \mathcal{I}$ and with any $y \in \mathcal{F}(x)$ a natural number $m(x, y)$ that denotes the *measure* of y.

5. $\textsc{opt} \in \{\max, \min\}$. ■

Note that we have chosen to define an optimisation problem as a tuple $\langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ rather than in the form $\langle \mathcal{I}, \mathcal{F}, m, \textsc{opt} \rangle$. The former definition allows us to consider partial orders defined on $\mathcal{U}(x)$, rather than on $\mathcal{F}(x)$. It will be our usual practice to define a partial order on the set $\mathcal{F}(x)$ (as in Definition 2.3.5), though the implications of defining partial orders on $\mathcal{U}(x)$ are discussed in Chapter 9. Also, the measure function is defined on $\mathcal{F}(x)$, rather than on $\mathcal{U}(x)$. Usually, it is simple to extend the definition of m to the wider class $\mathcal{U}(x)$, but for consistency with the definition in [24], we choose to define m on $\mathcal{F}(x)$ – this definition proves to be practicable for our study of minimaximal and maximinimal optimisation problems in this thesis.

Having defined the components of an optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ according to Definition 2.2.1, we henceforth refer to $\mathcal{F}(x)$, for a given instance x, without explicitly defining the set, since its definition may be obtained from those of $\mathcal{U}(x)$ and π, and Part 3 of Definition 2.2.1.

The next definition introduces three further concepts which follow from Definition 2.2.1.

Definition 2.2.2 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem and suppose $x \in \mathcal{I}$. The range of values that the measure function can take is denoted by $m(x, \mathcal{F}(x))$, i.e.,
$$m(x, \mathcal{F}(x)) = \{m(x, y) : y \in \mathcal{F}(x)\}.$$
The *globally optimal measure*, denoted by $m^*(x)$, is
$$m^*(x) = \textsc{opt}\ m(x, \mathcal{F}(x))$$
and the set of *globally optimal solutions*, denoted by $\mathcal{F}^*(x)$, is
$$\mathcal{F}^*(x) = \{y \in \mathcal{F}(x) : m(x, y) = m^*(x)\}. \blacksquare$$

Where there is no ambiguity, the globally optimal measure function m^* will simply be referred to as the *optimal measure function*.

An optimisation problem has three versions, as is indicated by the following definition.

Definition 2.2.3 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem.

- The objective of the *search version* of Π, Π_s, is to find, given an instance $x \in \mathcal{I}$, a globally optimal solution, i.e., a member of $\mathcal{F}^*(x)$.
- The objective of the *evaluation version* of Π, Π_e, is to find, given an instance $x \in \mathcal{I}$, the optimal measure, i.e., to compute $m^*(x)$.
- The objective of the *decision version* of Π, Π_d, is to determine, given an instance $x \in \mathcal{I}$ and an integer $K \in \mathbb{Z}^+$, whether there exists some $y \in \mathcal{F}(x)$ such that $m(x, y) \geq K$ (if OPT = max) or $m(x, y) \leq K$ (if OPT = min). ■

Throughout this thesis (apart from in Chapter 6), the default version of an optimisation problem Π is the search version of Π, and we thus reason about this version of an optimisation problem Π without subscript. When we give the name of the search version of an optimisation problem Π, the name of the decision version of Π is obtained by appending the word 'DECISION'.

We now define a class *NPO* of optimisation problems, which has implications for the decision versions of optimisation problems.

Definition 2.2.4 ([24]) Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem. Π belongs to the class *NPO* of *NP optimisation problems* if the following conditions hold:

1. The set $\mathcal{I}$ is recognisable in polynomial time.
2. A polynomial p exists such that, for any $x \in \mathcal{I}$ and $y \in \mathcal{U}(x)$, $|y| \leq p(|x|)$.
3. The set $\{(x, y) : x \in \mathcal{I} \wedge y \in \mathcal{U}(x)\}$ is recognisable in polynomial time.
4. The predicate π is decidable in polynomial time.
5. For any $x \in \mathcal{I}$ and $y \in \mathcal{F}(x)$, $m(x, y)$ is computable in polynomial time. ■

Given an optimisation problem Π, $\Pi \in$NPO implies that $\Pi_d \in$NP [24, Lemma 6.1].

2.3 Formal definition of minimaximal and maximinimal optimisation problems

Before presenting our framework for minimaximal and maximinimal optimisation problems, we begin with some concepts relating to partial orders.

Definition 2.3.1 A *strict partial order* $\prec$ defined on a set X is a subset of $X \times X$ such that

1. $\forall\, x \in X \bullet (x, x) \notin \prec$.
2. $\forall\, x, y, z \in X \bullet ((x, y) \in \prec \wedge (y, z) \in \prec) \Rightarrow (x, z) \in \prec$. ■

Henceforth we use the infix notation for partial orders, so that, for $x, x' \in X$, $x \prec x'$ stands for $(x, x') \in \prec$. All partial orders in this thesis are strict, so we use 'partial order' to stand for 'strict partial order'.

The course of our study of minimaximal and maximinimal optimisation problems involves obtaining one or more minimaximal or maximinimal optimisation problems from a *source* optimisation problem Π by introducing one or more partial orders defined on the set of all feasible solutions $\mathcal{F}(x)$, for a given instance x of Π. Since the definition of such a partial order will in general depend on the given instance x of Π, we use the superscript x with the partial order symbols to be used in the following chapters. Several concepts relating to partial orders are now defined.

Definition 2.3.2 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem. For $x \in \mathcal{I}$, suppose that $\prec^x$ is a partial order defined on the set $\mathcal{F}(x)$ and let $t, u, u' \in \mathcal{F}(x)$. Then

- t is *maximal with respect to* $\prec^x$, or $\prec^x$*-maximal*, if $\nexists\, t' \in \mathcal{F}(x) \bullet t \prec^x t'$.
- t is *minimal with respect to* $\prec^x$, or $\prec^x$*-minimal*, if $\nexists\, t' \in \mathcal{F}(x) \bullet t' \prec^x t$.
- t is $\prec^x$*-optimal* if *either*
  - $\text{OPT} = \max$ and t is $\prec^x$-maximal *or*
  - $\text{OPT} = \min$ and t is $\prec^x$-minimal.
- u is a $\prec^x$*-predecessor* of u' if $u \prec^x u'$.
- u' is a $\prec^x$*-successor* of u if and only if u is a $\prec^x$-predecessor of u'.
- u is an *immediate* $\prec^x$*-predecessor* of u' if

$$u \prec^x u' \wedge \nexists\, v \in \mathcal{F}(x) \bullet u \prec^x v \prec^x u'.$$

- u' is an *immediate* $\prec^x$*-successor* of u if and only if u is an immediate $\prec^x$-predecessor of u'. ■

We now give a property that $\prec^x$ must satisfy in order to be considered as a suitable partial order in this thesis.

Definition 2.3.3 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem and for $x \in \mathcal{I}$, let $\prec^x$ be a partial order defined on $\mathcal{F}(x)$. Then $\prec^x$ satisfies *Partial Order Measure Monotonicity (POMM)* with respect to Π if

$$\forall\, y, y' \in \mathcal{F}(x) \bullet y \prec^x y' \Rightarrow m(x, y) < m(x, y'). \;\blacksquare$$

Without POMM, a $\prec^x$-optimal solution would have essentially no significance insofar as measure is concerned. Furthermore, it is intuitively clear that a feasible solution y that is globally optimal should also be locally optimal in the sense that, if y is maximum (respectively minimum) with respect to $m(x, \cdot)$, then y should also be maximal (minimal) with respect to $\prec^x$. The following proposition indicates that the POMM criterion achieves this.

Proposition 2.3.4 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ be an optimisation problem. For an instance $x \in \mathcal{I}$, let $\prec^x$ be a partial order satisfying the POMM criterion in Definition 2.3.3, and let $y \in \mathcal{F}^*(x)$. Then y is $\prec^x$-optimal.*

Proof: Assume $\text{OPT} = \max$; the case $\text{OPT} = \min$ is similar. Suppose y is not $\prec^x$-maximal. Then there exists some $y' \in \mathcal{F}(x)$ such that $y \prec^x y'$. Hence the POMM criterion implies that $m(x, y) < m(x, y')$, contradicting the fact that $m(x, y) = m^*(x)$. Hence y is indeed a $\prec^x$-maximal element of $\mathcal{F}(x)$. ■

We now show how to obtain a minimaximal or maximinimal optimisation problem Π' from a source optimisation problem Π using a suitably defined partial order. The following definition forms the basis of our framework for minimaximal and maximinimal optimisation problems.

Definition 2.3.5 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ be an optimisation problem, called the *source* optimisation problem, and for $x \in \mathcal{I}$, suppose that $\prec^x$ is a partial order defined on $\mathcal{F}(x)$ satisfying POMM with respect to Π. Then we may define an optimisation problem $\Pi' = \langle \mathcal{I}', \mathcal{U}', \pi', m', \text{OPT}'\rangle$, where:

- $\mathcal{I}' = \mathcal{I}$.
- $\mathcal{U}' = \mathcal{U}$.
- $\pi' \Leftrightarrow \pi \wedge \sigma$, where, for $x \in \mathcal{I}'$ and $y \in \mathcal{F}(x)$, $\sigma(x, y)$ if and only if y is $\prec^x$-optimal. For $x \in \mathcal{I}'$, we denote the feasible solutions of x by $\mathcal{F}'(x)$, where
$$\mathcal{F}'(x) = \{y \in \mathcal{U}'(x) : \pi'(x, y)\}.$$
- For $x \in \mathcal{I}'$ and $y \in \mathcal{F}'(x)$, $m'(x, y) = m(x, y)$.
- $\text{OPT}' = \begin{cases} \min\text{, if OPT=max} \\ \max\text{, if OPT=min.} \end{cases}$

If OPT=max then Π' is a *minimaximal optimisation problem*, and if OPT=min then Π' is a *maximinimal optimisation problem*. If, in addition, the set $\{(x, y) : x \in \mathcal{I}' \wedge y \in \mathcal{F}'(x)\}$ is recognisable in deterministic polynomial time, then Π' is in NPO. ■

Some minimaximal and maximinimal optimisation problems are not known to be in NPO even when the source optimisation problem is. For example, when the partial order of *partition redistribution* (to be defined in the following section) is defined on the feasible solutions of the source problem BIN PACKING (whose decision version is [92, problem SR1], and in NP), the decision version of the maximinimal optimisation problem obtained using Definition 2.3.5 is not known to be in NP (see Theorem 9.3.2). However, all minimaximal and maximinimal optimisation problems in this thesis (apart from those of Chapter 6, where an alternative framework is given for such problems) are obtained from source optimisation problems in the class NPO, using an appropriate partial order. Nevertheless, for full generality, we do not stipulate that the source optimisation in Definition 2.3.5 must be in NPO.

Since every partial order is a relation, it is clear that all minimaximal and maximinimal optimisation problems defined in this thesis could be obtained from a modified Definition 2.3.5, in which a general relation satisfying POMM is defined on the feasible solutions for a given instance of an optimisation problem. However, every optimisation problem that we have encountered in the literature that is essentially a 'minimum maximal' or 'maximum minimal' problem (apart from fractional problems of this type, which are modelled using a separate framework, defined in Chapter 6) may be defined using the framework of Definition 2.3.5 as it stands. Thus, we opt for the partial order-based approach, in view of the increased mathematical structure. Indeed, we shall, from time to time, utilise the transitivity property of a partial order when reasoning about the relationships between feasible solutions.

Our final remark following Definition 2.3.5 concerns the partial order $\prec^x$. In our framework for minimaximal and maximinimal optimisation problems, $\prec^x$ is defined on $\mathcal{F}(x)$. In Chapter 9, we discuss the effect of defining $\prec^x$ on $\mathcal{U}(x)$.

2.4 Overview of some partial orders for minimaximal and maximinimal optimisation problems

In this section we introduce several generic partial orders, to be defined on the feasible solutions of source optimisation problems, in order to obtain minimaximal and maximinimal optimisation problems, using the framework of Definition 2.3.5. In the following chapters, it is easy to verify that each of the following partial orders satisfies POMM with respect to the particular optimisation problem to which the partial order is applied.

The first two partial orders correspond to optimisation problems whose feasible solutions are subsets of some fixed set associated with the instance.

Definition 2.4.1 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem. For any $x \in \mathcal{I}$, suppose that there is some set S^x, associated with x, such that $\mathcal{U}(x)$ is a set of subsets of S^x. Define

$$\subset^x = \{(S', S'') \in \mathcal{F}(x) \times \mathcal{F}(x) : S' \subset S''\}.$$

Then $\subset^x$ is the partial order of *set inclusion*. ∎

Definition 2.4.2 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem as in Definition 2.4.1. For $k \geq 1$ and for $x \in \mathcal{I}$, define

$$\sqsubset_k^x = \left\{ (S', S'') \in \mathcal{F}(x) \times \mathcal{F}(x) : \begin{array}{l} \exists\, A \subseteq S' \;\wedge\; |A| \leq k-1 \\ \exists\, B \subseteq S^x \backslash S' \;\wedge\; |B| = |A| + 1 \\ S'' = (S' \backslash A) \cup B \end{array} \right\}.$$

By taking $\subset_k^x = (\sqsubset_k^x)^*$ (the transitive closure of relation $\sqsubset_k^x$), we obtain a partial order that we call $(k-1, k)$-*replacement*. ∎

Intuitively, for $k \geq 1$ and two members S', S'' of $\mathcal{F}(x)$, $S' \sqsubset_k^x S''$ if S'' can be obtained from S' by removing a set A of $r-1$ (where $r \leq k$) elements from S' and adding a set B

of r elements from $S^x \backslash S'$. The partial order admits the useful property that, for $k \geq 2$, $\sqsubset_k^x$ is a refinement of $\sqsubset_{k-1}^x$. This is demonstrated by the next result, which follows by observing that, for two members S', S'' of $\mathcal{F}(x)$, $S' \sqsubset_{k-1}^x S''$ implies $S' \sqsubset_k^x S''$.

Proposition 2.4.3 *Let* $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ *be an optimisation problem as in Definition 2.4.1. Then, for* $k \geq 2$ *and* $x \in \mathcal{I}$, $\sqsubset_{k-1}^x$ *is contained in* $\sqsubset_k^x$.

Corollary 2.4.4 *Let* $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ *be an optimisation problem as in Definition 2.4.1. Then, for* $k \geq 2$, *for* $x \in \mathcal{I}$ *and for* $y \in \mathcal{F}(x)$, y *is* $\sqsubset_k^x$*-optimal implies that* y *is* $\sqsubset_{k-1}^x$*-optimal.*

The partial orders of set inclusion and $(0,1)$-replacement are closely related. Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ be an optimisation problem as in Definition 2.4.1, and suppose $x \in \mathcal{I}$. For any $s, t \in \mathcal{F}(x)$, is clear that $s \sqsubset_1^x t$ implies $s \subset^x t$. However, the converse is also true when a certain condition is satisfied, given by the following definition. The terminology follows that of Dunbar et al. [68].

Definition 2.4.5 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ be an optimisation problem as in Definition 2.4.1. Property π is *hereditary* if, for any $x \in \mathcal{I}$, whenever $s \in \mathcal{U}(x)$, $t \in \mathcal{F}(x)$ and $s \subset t$, then $s \in \mathcal{F}(x)$. Property π is *super-hereditary* if, for any $x \in \mathcal{I}$, whenever $s \in \mathcal{F}(x)$, $t \in \mathcal{U}(x)$ and $s \subset t$, then $t \in \mathcal{F}(x)$. ∎

Given an optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, we say that Π is hereditary or super-hereditary if π is hereditary or super-hereditary, respectively. Each of the optimisation problems considered in Chapter 4 (where we consider the partial order of set inclusion, defined on the feasible solutions of a source graph optimisation problem for a given instance) is either hereditary or super-hereditary. However in Section 5.2, we give an example of an optimisation problem that is neither hereditary nor super-hereditary.

The partial orders of set inclusion and $(0,1)$-replacement are equal when the hereditary or super-hereditary property is satisfied. This is demonstrated by the following result.

Proposition 2.4.6 *Let* $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ *be an optimisation problem as in Definition 2.4.1. In addition, let* π *be hereditary or super-hereditary. Then, for any* $x \in \mathcal{I}$ *and* $s, t \in \mathcal{F}(x)$, $s \subset^x t$ *implies* $s \sqsubset_1^x t$.

Proof: Let $s \subset^x t$, where $t \backslash s = \{r_1, r_2, \ldots, r_n\}$ for some $n \geq 1$. Define $s_0 = s$ and, for $1 \leq i \leq n$, define

$$s_i = s \cup \{r_1, r_2, \ldots, r_i\}.$$

Then, for $1 \leq i \leq n$,

$$s = s_0 \subseteq s_{i-1} \subset s_i \subseteq s_n = t.$$

Thus an easy induction establishes that $s_i \in \mathcal{F}(x)$, for $1 \leq i \leq n$, using either the hereditary or super-hereditrary property throughout. Therefore $s_{i-1} \sqsubset_1^x s_i$ $(1 \leq i \leq n)$, which implies that $s \sqsubset_1^x t$ as required. ∎

This section proceeds with some further partial order definitions. The following two partial orders correspond to optimisation problems whose feasible solutions are strings over some fixed alphabet associated with the instance.

Definition 2.4.7 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem. For any $x \in \mathcal{I}$, suppose that there is an alphabet Σ^x, associated with x, such that $\mathcal{U}(x)$ is a finite set of strings, each composed of symbols of Σ^x. Define

$$\ll^x = \{(s', s'') \in \mathcal{F}(x) \times \mathcal{F}(x) : s' \ll s''\}.$$

Then $\ll^x$ is the *subsequence* partial order. ■

Definition 2.4.8 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem as in Definition 2.4.7. For $x \in \mathcal{I}$, define

$$\overline{\ll}^x = \{(s', s'') \in \mathcal{F}(x) \times \mathcal{F}(x) : s' \overline{\ll} s''\}.$$

Then $\overline{\ll}^x$ is the *substring* partial order. ■

The following two partial orders correspond to optimisation problems whose feasible solutions are partitions of some fixed set associated with the instance.

Definition 2.4.9 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem. For any $x \in \mathcal{I}$, suppose that there is some set S^x, associated with x, such that $\mathcal{U}(x)$ is a set of partitions of S^x. Define

$$\sqsubset_a^x = \left\{ (P, Q) \in \mathcal{F}(x) \times \mathcal{F}(x) : \begin{array}{c} P = \{s_1, s_2, \ldots, s_t\} \wedge \\ Q = \{s'_1, s'_2, \ldots, s'_{t+1}\} \wedge \\ \forall\, 1 \leq i \leq t-1 \bullet s_i = s'_i \end{array} \right\}.$$

By taking $\prec_a^x = (\sqsubset_a^x)^*$, we obtain a partial order that we call *partition merge*. ■

Intuitively, we have that, for two partitions P_1 and P_2 of S, $P_1 \sqsubset_a^x P_2$ if P_1 can be obtained from P_2 by merging two of the constituent sets of P_2.

Definition 2.4.10 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem as in Definition 2.4.9. For $x \in \mathcal{I}$, define

$$\sqsubset_b^x = \left\{ (P, Q) \in \mathcal{F}(x) \times \mathcal{F}(x) : \begin{array}{c} P = \{s_1, s_2, \ldots, s_t\} \wedge \\ Q = \{s'_1, s'_2, \ldots, s'_{t+1}\} \wedge \\ \forall\, 1 \leq i \leq t \bullet s'_i \subseteq s_i \end{array} \right\}.$$

By taking $\prec_b^x = (\sqsubset_b^x)^*$, we obtain a partial order that we call *partition redistribution*. ■

Intuitively, we have that, for two partitions P_1 and P_2 of S, $P_1 \sqsubset_b^x P_2$ if P_1 can be obtained from P_2 by taking a constituent set of P_2 and partitioning its members amongst the remaining sets of P_2. Suppose that $P_1 = \{s_1, s_2, \ldots, s_t\}$ and $P_2 = \{s'_1, s'_2, \ldots, s'_{t+1}\}$ such that, without loss of generality, $s'_i \subseteq s_i$ for $1 \leq i \leq t$. We say that P_1 is a *redistribution* of P_2, and that s'_{t+1} has been *redistributed* among the sets $s'_1, s'_2, \ldots, s'_t$. Note that $\prec_b^x$ is a refinement of $\prec_a^x$. This is demonstrated by the next result, which follows by observing that, for two partitions P_1 and P_2 of $\mathcal{F}(x)$, $P_1 \sqsubset_a^x P_2$ implies $P_1 \sqsubset_b^x P_2$.

Proposition 2.4.11 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $x \in \mathcal{I}$, $\prec_a^x$ is contained in $\prec_b^x$.*

Corollary 2.4.12 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $x \in \mathcal{I}$ and for $y \in \mathcal{F}(x)$, y is $\prec_b^x$-optimal implies that y is $\prec_a^x$-optimal.*

The two families of partial orders given by Definitions 2.4.13 and 2.4.16 generalise Definitions 2.4.9 and 2.4.10 respectively.

Definition 2.4.13 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $k \geq 2$, and for $x \in \mathcal{I}$, define

$$\sqsubset_{a,k}^x = \left\{ (P, Q) \in \mathcal{F}(x) \times \mathcal{F}(x) : \begin{array}{l} P = \{s_1, s_2, \ldots, s_t\} \wedge \\ Q = \{s'_1, s'_2, \ldots, s'_{t+1}\} \wedge \\ \exists\, 2 \leq r \leq k \bullet \\ \quad \forall\, 1 \leq i \leq t+1-r \bullet s_i = s'_i \end{array} \right\}.$$

By taking $\prec_{a,k}^x = (\sqsubset_{a,k}^x)^*$, we obtain a partial order that we call *partition $(k-1,k)$-merge.* ■

Intuitively, for two partitions P_1 and P_2 in $\mathcal{F}(x)$, $P_1 \sqsubset_{a,k}^x P_2$ if P_1 can be obtained from P_2 by merging the elements of r sets in P_2 $(2 \leq r \leq k)$ into $r-1$ new sets, whilst the other sets in P_2 retain their original elements. Note that if $k = 2$ then we obtain the partial order of partition merge. Also, for $k \geq 3$, $\prec_{a,k}^x$ is a refinement of $\prec_{a,k-1}^x$. This is demonstrated by the next result, which follows by observing that, for two partitions P_1, P_2 of $\mathcal{F}(x)$, $P_1 \sqsubset_{a,k-1}^x P_2$ implies $P_1 \sqsubset_{a,k}^x P_2$.

Proposition 2.4.14 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. Then, for $k \geq 3$ and for $x \in \mathcal{I}$, $\prec_{a,k-1}^x$ is contained in $\prec_{a,k}^x$.*

Corollary 2.4.15 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. Then, for $k \geq 3$, for $x \in \mathcal{I}$ and for $y \in \mathcal{F}(x)$, y is $\prec_{a,k}^x$-optimal implies that y is $\prec_{a,k-1}^x$-optimal.*

Definition 2.4.16 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $k \geq 1$ and for $x \in \mathcal{I}$, define

$$\sqsubset_{b,k}^x = \left\{ (P, Q) \in \mathcal{F}(x) \times \mathcal{F}(x) : \begin{array}{l} P = \{s_1, s_2, \ldots, s_t\} \wedge \\ Q = \{s'_1, s'_2, \ldots, s'_{t+1}\} \wedge \\ \exists\, 1 \leq r \leq k \bullet \\ \quad \forall\, 1 \leq i \leq t+1-r \bullet s'_i \subseteq s_i \end{array} \right\}.$$

By taking $\prec_{b,k}^x = (\sqsubset_{b,k}^x)^*$, we obtain a partial order that we call *partition k-redistribution.* ■

Intuitively, for two partitions P_1 and P_2 in $\mathcal{U}(x)$, $P_1 \sqsubset_{b,k}^x P_2$ if P_1 can be obtained from P_2 by distributing the elements of r sets in P_2 $(1 \leq r \leq k)$ amongst the remaining sets in

P_2 plus $r-1$ new sets. Note that if $k=1$ then we obtain the partial order of partition redistribution. Also, for $k \geq 2$, $\prec^x_{b,k}$ is a refinement of $\prec^x_{b,k-1}$. This is demonstrated by the next result, which follows by observing that, for two partitions P_1, P_2 of $\mathcal{F}(x)$, $P_1 \sqsubset^x_{b,k-1} P_2$ implies $P_1 \sqsubset^x_{b,k} P_2$.

Proposition 2.4.17 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. Then, for $k \geq 2$ and for $x \in \mathcal{I}$, $\prec^x_{b,k-1}$ is contained in $\prec^x_{b,k}$.*

Corollary 2.4.18 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. Then, for $k \geq 2$, for $x \in \mathcal{I}$ and for $y \in \mathcal{F}(x)$, y is $\prec^x_{b,k}$-optimal implies that y is $\prec^x_{b,k-1}$-optimal.*

The next result shows that $\prec^x_{b,k}$ is a refinement of $\prec^x_{a,k}$, and follows by observing that, for two partitions P_1 and P_2 of $\mathcal{F}(x)$, $P_1 \sqsubset^x_{a,k} P_2$ implies $P_1 \sqsubset^x_{b,k} P_2$.

Proposition 2.4.19 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $k \geq 2$ and for $x \in \mathcal{I}$, $\prec^x_{a,k}$ is contained in $\prec^x_{b,k}$.*

Corollary 2.4.20 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem as in Definition 2.4.9. For $k \geq 2$, for $x \in \mathcal{I}$ and for $y \in \mathcal{F}(x)$, y is $\prec^x_{b,k}$-optimal implies that y is $\prec^x_{a,k}$-optimal.*

One further partial order that we study is concerned with functions. We introduce the *partial order on functions* in Chapter 6, as the partial order will be defined on the feasible solutions of a *fractional graph optimisation problem*, also to be defined in that chapter.

Finally, the *partial order on truth assignments*, associated with logic-related optimisation problems, is defined in Section 7.7.1, as its definition uses additional terminology that we introduce in that section.

2.5 Relationships between neighbourhood relations and partial orders

In this section, we consider the correspondence between the notion of a partial order within the context of the framework of Definition 2.3.5, and the notion of a neighbourhood relation within the context of the framework of PLS. We show that, despite a relationship between local optimality with respect to a neighbourhood relation and maximality or minimality with respect to a partial order satisfying POMM, the two mathematical structures are distinct entities, reflecting the differing objectives of the frameworks of PLS and Definition 2.3.5 on which the structures are based.

Intuitively, the term 'neighbourhood relation' conjures up an image of the feasible solutions as points in space, in which a pair of feasible solutions are related to each other as neighbours if and only if they are at most distance d apart (as described by Yannakakis [211]). This intuition, along with the dictionary meaning of the word 'neighbour', implies reciprocity, in that s is a neighbour of t implies t is a neighbour of s. However, such symmetry is not a requirement of the classical definition of the neighbourhood relation for a local search problem (see for example [179, p.7]). Nevertheless, most neighbourhood relations that occur in the literature are indeed symmetric.

Considering specific examples, the k-opt neighbourhood for TSP [179, Section 19.2], the *swap* neighbourhood for the so-called GRAPH PARTITIONING problem [179, Section 19.5] and the *bit-flipping* neighbourhood for the so-called FLIP problem [136] are instances of symmetric neighbourhood structures for local search problems that have played a key role in the study of the class PLS [149, 193, 136]. By constrast, the Lin-Kernighan neighbourhood for TSP [155] and the Kernighan-Lin neighbourhood for GRAPH-PARTITIONING [142] are examples of local search problems incorporating non-symmetric neighbourhood relations. These two neighbourhood relations have the common property that, for two feasible solutions s and t, s is a neighbour of t implies that s has better measure than t. Thus non-symmetry is immediate. In fact, every local search problem we have found in the literature, incorporating a neighbourhood relation that is non-symmetric, satisfies the property that a neighbour of a feasible solution s is a local improvement of s.

Thus, for local search problems incorporating such non-symmetric neighbourhoods, we may view the neighbourhood structure as a relation satisfying POMM. By taking the transitive closure of this relation, we may reason about local optimality with respect to a partial order satisfying POMM. However, the same is also possible for local search problems incorporating symmetric neighbourhoods, if we consider only local improvements. The remarks of this paragraph are formalised by the following theorem.

Theorem 2.5.1 *Let* $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ *be an optimisation problem, and for an instance* x *of* Π, *suppose that* N_x *is a neighbourhood relation defined on the feasible solutions* $\mathcal{F}(x)$. *Then we may construct a partial order* $\prec^x$, *defined on* $\mathcal{F}(x)$, *satisfying POMM, such that, given any* $t \in \mathcal{F}(x)$, t *is locally optimal with respect to* N_x *if and only if* t *is* $\prec^x$*-optimal.*

Proof: If $\text{OPT} = \min$, then consider the relation

$$\sqsubset^x = \{(s,t) \in \mathcal{F}(x) \times \mathcal{F}(x) : s \in N_x(t) \wedge m(x,s) < m(x,t)\}.$$

Denote by $\prec^x$ the transitive closure of $\sqsubset^x$. Clearly, $\prec^x$ is a partial order satisfying POMM. Suppose that $t \in \mathcal{F}(x)$ is locally optimal with respect to N_x, but there is some $s \in \mathcal{F}(x)$ such that $s \prec^x t$. Then there is some $n \geq 2$ and $s_i \in \mathcal{F}(x)$ $(1 \leq i \leq n)$ such that

$$s = s_1 \sqsubset^x s_2 \sqsubset^x \ldots \sqsubset^x s_{n-1} \sqsubset^x s_n = t.$$

In particular, $s_{n-1} \in N_x(t)$ and $m(x, s_{n-1}) < m(x,t)$, contradicting the local optimality of t. Conversely, it is clear that a $\prec^x$-minimal element of $\mathcal{F}(x)$ is locally optimal with respect to N_x.

Similarly, if $\text{OPT} = \max$, then consider the relation

$$\sqsubset^x = \{(s,t) \in \mathcal{F}(x) \times \mathcal{F}(x) : t \in N_x(s) \wedge m(x,s) < m(x,t)\}.$$

Again, denote by $\prec^x$ the transitive closure of $\sqsubset^x$. As in the case $\text{OPT} = \min$, it may be verified that $\prec^x$ is a partial order satisfying POMM, and that a feasible solution s is locally optimal with respect to N_x if and only if s is $\prec^x$-maximal. ■

On the other hand, given an optimisation problem Π with a partial order $\prec^x$, satisfying POMM with respect to Π, defined on the feasible solutions of Π for a given instance x, we may construct a symmetric neighbourhood relation N_x with the property that a feasible solution y is $\prec^x$-optimal if and only if y is locally optimal with respect to N_x. This is demonstrated formally by the following result.

Theorem 2.5.2 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem, and for an instance x of Π, suppose that $\prec^x$ is a partial order defined on $\mathcal{F}(x)$, satisfying POMM. Then we may define a symmetric neighbourhood relation N_x such that, given any $t \in \mathcal{F}(x)$, t is $\prec^x$-optimal if and only if t is locally optimal with respect to N_x.*

Proof: We construct a neighbourhood relation as follows. For any $t \in \mathcal{F}(x)$, define

$$\begin{aligned} N_x(t) \quad = \quad & \{s \in \mathcal{F}(x) : s \text{ is an immediate } \prec^x\text{-predecessor of } t\} \cup \\ & \{u \in \mathcal{F}(x) : u \text{ is an immediate } \prec^x\text{-successor of } t\}. \end{aligned}$$

Clearly, this neighbourhood relation is symmetric. Suppose firstly that $\text{OPT} = \min$, some $t \in \mathcal{F}(x)$ is $\prec^x$-minimal, and there is some $s \in N_x(t)$ such that $m(x,s) < m(x,t)$. Then $s \prec^x t$, by POMM. This contradiction to the $\prec^x$-minimality of t implies that t is locally optimal with respect to N_x. Conversely, suppose that t is locally optimal with respect to N_x, and t is not $\prec^x$-minimal. Then t has an immediate $\prec^x$-predecessor, say s. Thus $s \in N_x(t)$, and by POMM, s contradicts the local optimality of t. Hence t is $\prec^x$-minimal.

Similarly, if $\text{OPT} = \max$ and $t \in \mathcal{F}(x)$, then t is $\prec^x$-maximal if and only if t is locally optimal with respect to N_x. ■

Thus, it would appear that there is a correspondence between local optimality with respect to the neighbourhood relation framework of the class PLS, and local optimality with respect to the partial order framework of Definition 2.3.5. However, it is important to draw distinctions between the two mathematical structures.

A neighbourhood relation is not, in general, a partial order: transitivity is not satisfied for any of the neighbourhoods discussed above. Moreover, a symmetric neighbourhood relation does not satisfy POMM. Recall that, in the framework of Definition 2.3.5, the relation defined on the feasible solutions of an optimisation problem must be a partial order satisfying POMM. Reasons for opting for a partial order within our framework were discussed in the remarks following Definition 2.3.5. We also discussed why a partial order should satisfy POMM in the remarks following Definition 2.3.3.

On the other hand, a strict partial order satisfying POMM is not symmetric, and thus we could not define a neighbourhood relation such as k-opt using such a structure without changing the notion of k-opt to one in which only local improvements are allowed (as is done by Theorem 2.5.1). This, in our opinion, would diminish the elegance of the definition of a such a neighbourhood relation, and would also undermine the inherent structure of the neighbourhood concept in such a case.

Chapter 3

Maximinimal graph colouring problems

3.1 Introduction

In this chapter we consider two maximinimal graph colouring optimisation problems that may be obtained from the source CHROMATIC NUMBER problem, defined as follows:

Source problem: CHROMATIC NUMBER=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{G = (V, E) : G$ is a graph$\}$
- $\mathcal{U}(G) = \{P : P$ is a partition of $V\}$, for $G \in \mathcal{I}$
- $\pi(G, P) \Leftrightarrow \forall S \in P \bullet S$ is independent in G, for $G \in \mathcal{I}$ and $P \in \mathcal{U}(G)$
- $m(G, P) = |P|$, for $G \in \mathcal{I}$ and $P \in \mathcal{F}(G)$
- $\text{OPT} = \min$.

We now present some terminology relating to the CHROMATIC NUMBER problem. The *chromatic number*, $\chi(G)$, of a graph G is $m^*(G)$. A *k-colouring* of G is an element P of $\mathcal{U}(G)$ such that $|P| = k$. A *proper k-colouring* of G is an element P of $\mathcal{F}(G)$ such that $m(G, P) = k$. We say that a graph G is *k-colourable* if G has a proper k-colouring. For a graph $G = (V, E)$ and vertex $v \in V$, the *colour*, $c(v)$, assigned to v in any colouring $\{V_1, V_2, \ldots, V_k\}$ is the unique i such that $1 \leq i \leq k$ and $v \in V_i$. The CHROMATIC NUMBER problem has been extensively studied with regard to algorithmic complexity. CHROMATIC NUMBER DECISION is NP-complete [92, problem GT4], even for $K = 3$ and planar graphs of maximum degree 4 [93], though polynomial-time solvable for chordal graphs [95]. The complexity of this problem for many other graph classes is surveyed by Johnson [134].

By considering CHROMATIC NUMBER, together with the partial orders of partition merge and partition redistribution, we obtain from Definition 2.3.5 two maximinimal graph colouring optimisation problems, called MAXIMUM $\prec_a$-MINIMAL CHROMATIC NUMBER and MAXIMUM $\prec_b$-MINIMAL CHROMATIC NUMBER respectively. In Section 3.2, we show that the first problem has in fact been studied previously and is called ACHROMATIC NUMBER in the literature. In Sections 3.3-3.6, the second problem is considered. This problem, which we call B-CHROMATIC NUMBER, is new. We study the computational complexity of B-CHROMATIC NUMBER in arbitrary graphs, bipartite graphs and trees. We obtain NP-completeness results for B-CHROMATIC NUMBER DECISION in general graphs and bipartite graphs in Sections 3.4 and 3.5 respectively. B-CHROMATIC NUMBER is shown to be polynomial-time solvable for trees in Section 3.6.

3.2 Defining the achromatic number

In the previous section, the chromatic number of a graph G was defined to be the minimum number of colours over all proper colourings of G. A related parameter is the *achromatic number*, $\psi(G)$, of a graph G, which involves maximising the number of colours over all proper colourings, rather than minimising. Garey and Johnson [92] define $\psi(G)$ to be the maximum k for which G has a proper colouring $\{V_1, V_2, \ldots, V_k\}$ that also satisfies the following property:

$$\forall\, 1 \leq i < j \leq k \bullet V_i \cup V_j \text{ is not independent.} \tag{3.1}$$

A proper colouring for a graph G satisfying Property 3.1 above is called a *complete* or *achromatic* colouring.

The parameter $\psi(G)$ was first studied by Harary et al. [110], who define a *homomorphism* from a graph G to a graph H as a map Φ from $V(G)$ onto $V(H)$ such that u, v are adjacent in G implies that $u\Phi, v\Phi$ are adjacent in H. A homomorphism Φ is said to be *complete of order* n if $G\Phi$ is isomorphic to K_n. Harary et al. then show that $\chi(G)$ is the smallest order of a complete homomorphism of G. They define the parameter $\psi(G)$ to be the maximum order of all complete homomorphisms of G. The parameter $\psi(G)$ was named the achromatic number of G by Harary and Hedetniemi [109].

The ACHROMATIC NUMBER problem (whose decision version is problem GT5 of [92]) is to find, given a graph G, an achromatic colouring for G of $\psi(G)$ colours.

We now show how to obtain the definition of ACHROMATIC NUMBER in terms of a partial order defined on the set of all proper colourings for a graph G. Let $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ be defined as for the CHROMATIC NUMBER problem in Section 3.1. Consider the partial order $\prec_a^G$ of partition merge, defined on $\mathcal{F}(G)$. Recall that $\prec_a^G$ is the transitive closure of the relation $\sqsubset_a^G$. Intuitively, for two colourings $c_1, c_2 \in \mathcal{F}(G)$, $c_1 \sqsubset_a^G c_2$ if and only if (in order to produce colouring c_1) every vertex belonging to one of the colours i in c_2 is recoloured by one particular colour j chosen from the other colours, while every other vertex retains its original colour. If V_i and V_j denote the vertices coloured i and j in c_2, then for this recolouring to be possible, it is clear that $V_i \cup V_j$ must be independent in G. Thus it is follows that $c \in \mathcal{F}(G)$ is $\prec_a^G$-minimal if and only if Property 3.1 holds. This justifies the following alternative definition of $\psi(G)$:

$$\psi(G) = \max\{|c| : c \in \mathcal{F}(G) \text{ is } \prec_a^G\text{-minimal}\}.$$

We may utilise Definition 2.3.5 to give us the formal definition of the MAXIMUM $\prec_a$-MINIMAL CHROMATIC NUMBER, or ACHROMATIC NUMBER, problem in terms of $\prec_a^G$. Thus ACHROMATIC NUMBER is an optimisation problem in the literature that, perhaps surprisingly, turns out to be a maximinimal optimisation problem within our framework.

The computational complexity of ACHROMATIC NUMBER has been studied for a number of graph classes. ACHROMATIC NUMBER DECISION is shown to be NP-complete by Yannakakis and Gavril [213], even for the complements of bipartite graphs. The problem is also shown to remain NP-complete for bipartite graphs [75] and connected graphs that are simultaneously a *cograph* and an *interval* graph [21]. ACHROMATIC NUMBER DECISION has recently been shown to remain NP-complete for trees [30]. Chaudhary and Vishwanathan [37] obtain the first polynomial-time $o(n)$ approximation algorithm for ACHROMATIC NUMBER.

3.3 Defining the b-chromatic number

We now consider the partial order of partition redistribution, $\prec_b^G$, which is a natural refinement of partition merge, defined on $\mathcal{F}(G)$, for a graph G (where $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ are defined as for the CHROMATIC NUMBER problem in Section 3.1). Recall that $\prec_b^G$ is the transitive closure of the relation $\sqsubset_b^G$. In Section 3.2, we described that intuitively, for two colourings $c_1, c_2 \in \mathcal{F}(G)$, $c_1 \sqsubset_a^G c_2$ if and only if (in order to produce colouring c_1) every vertex belonging to one of the colours i in c_2 is recoloured by one particular colour j chosen from the other colours, while every other vertex retains its original colour. In fact it would be more flexible to allow the recolouring process to pick some colour i of c_2 and redistribute the vertices of colour i among the other colours of c_2, in order to produce c_1. This is exactly the intuition behind $c_1 \sqsubset_b^G c_2$.

We have that a proper colouring $\{V_1, V_2, \ldots, V_k\}$ is $\prec_b^G$-minimal if and only if it is not possible to redistribute the vertices of a colour i $(1 \le i \le k)$ amongst the other colours $1, 2, \ldots, i-1, i+1, \ldots, k$ in order to obtain a proper colouring. Since the vertices of colour i are independent, the choice of new colour given to one vertex of colour i has no bearing on the choice of new colour given to any other vertex of colour i, when making this redistribution. This observation gives rise to the following result, which provides a convenient criterion for a proper colouring to be $\prec_b^G$-minimal.

Proposition 3.3.1 *A proper colouring $\{V_1, V_2, \ldots, V_k\}$ for a graph $G = (V, E)$ is $\prec_b^G$-minimal if and only if*

$$\forall\, 1 \le i \le k \bullet \exists\, v_i \in V_i \bullet \forall\, 1 \le j \ne i \le k \bullet \exists\, w_j \in V_j \bullet \{v_i, w_j\} \in E. \tag{3.2}$$

Intuitively, a proper k-colouring is $\prec_b^G$-minimal if and only if each colour i contains at least one vertex v_i that is adjacent to a vertex of every colour j $(1 \le j \ne i \le k)$. We call such a vertex v_i a *b-chromatic vertex* for colour i. We call a proper k-colouring that satisfies Property 3.2 a *b-chromatic k-colouring.* It follows from Proposition 3.3.1 that testing a proper colouring for $\prec_b^G$-minimality is polynomial-time solvable, and that

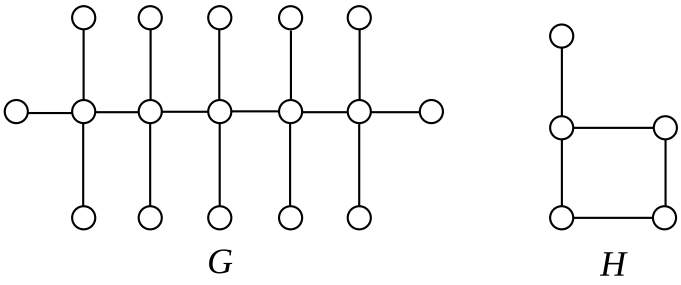

Figure 3.1: Examples to show that $\Gamma(G)$ need not be an upper bound for $\varphi(G)$, or vice versa.

an iteration of the implicit algorithm for testing gives a polynomial-time procedure for finding a $\prec_b^G$-minimal colouring.

We can now make the following definition:

Definition 3.3.2 The *b-chromatic number*, $\varphi(G)$, of a graph $G = (V, E)$ is defined by

$$\varphi(G) = \max\{|c| : c \in \mathcal{F}(G) \text{ is } \prec_b^G\text{-minimal}\}.$$

The MAXIMUM $\prec_b$-MINIMAL CHROMATIC NUMBER, or B-CHROMATIC NUMBER problem is derived from CHROMATIC NUMBER and the partial order $\prec_b^G$, using Definition 2.3.5. ■

Therefore, the b-chromatic number parameter for a graph G is the maximum number of colours for which G has a proper colouring such that every colour contains a vertex adjacent to a vertex of every other colour. This parameter does not appear to have been studied previously in the literature.

Hughes and MacGillivray [127] give an interpretation of the achromatic number parameter $\psi(G)$ as being the largest number of colours in a proper colouring of G, "which does not obviously use unnecessary colours". The definition of the b-chromatic number parameter $\varphi(G)$ therefore incorporates a partial order $\prec_b^G$ that substantially strengthens this notion of not 'wasting' colours.

The parameter $\varphi(G)$ superficially resembles the *Grundy number*[1], $\Gamma(G)$, of G. The Grundy number (first named and studied by Christen and Selkow [40]) is the maximum number of colours k for which G has a *Grundy k-colouring*. A Grundy k-colouring of G is a proper colouring of G using colours $0, 1, \ldots, k-1$ such that every vertex coloured i, for each $0 \leq i < k$, is adjacent to at least one vertex coloured j, for each $0 \leq j < i$. In general it is not the case that the Grundy number is an upper bound for the b-chromatic number, or vice versa, as is demonstrated in Figure 3.1: here $\Gamma(G) = 4$ while $\varphi(G) = 5$, and $\Gamma(H) = 3$ while $\varphi(H) = 2$. Thus in general $\Gamma(G)$ and $\varphi(G)$ are distinct parameters, for a given graph G.

[1]To be consistent with the literature, we must denote by Γ both the Grundy number and the *maximum minimal domination number* (defined in Section 4.3.3) in this thesis. However, all references to the Grundy number are confined to this chapter, and in Chapter 4 onwards, all instances of Γ denote the maximum minimal domination number.

The partial order $\prec_b^G$ is a refinement of $\prec_a^G$ in that a proper colouring that is $\prec_b^G$-minimal is also $\prec_a^G$-minimal. Thus $\varphi(G) \leq \psi(G)$. An immediate lower bound for $\varphi(G)$ is $\chi(G)$, since any proper colouring of G using $\chi(G)$ colours must be b-chromatic. However $\varphi(G)$ may be arbitrarily far away from $\chi(G)$: consider the graph G shown in Figure 3.2(a), that is the complete bipartite graph $K_{n,n}$ minus a perfect matching. Letting $c(u_i) = c(v_i) = i$ for $1 \leq i \leq n$ gives a b-chromatic n-colouring. As each vertex has degree $n-1$, $\varphi(G) = n$, whereas $\chi(G) = 2$.

Harary et al. [110] show (as a consequence of their *Homomorphism Interpolation Theorem*) that an arbitrary graph $G = (V, E)$ has achromatic colourings of any size between $\chi(G)$ and $\psi(G)$. Thus, in the terminology of Harary [107], ψ is an *interpolating invariant*, i.e. for any graph G, the set

$$S = \{k \in \mathbb{Z}^+ : G \text{ has an achromatic colouring of size } k\}$$

is *convex*, that is, every n between $\min(S)$ and $\max(S)$ belongs to S. It turns out that φ is not an interpolating invariant. This may be seen by considering the graph G of Figure 3.2(a) with $n = 4$, illustrated in Figure 3.2(b). We saw previously that G has b-chromatic 2 and 4-colourings, but there is no 3-colouring of G that is b-chromatic, which may be seen as follows. Suppose that G *does* have a b-chromatic 3-colouring, and without loss of generality suppose that $c(u_1) = 1$ and $c(v_4) = 2$. Suppose, again without loss of generality, that u_2 is a b-chromatic vertex for colour 3. Then $c(v_1) = 1$ which in turn forces $c(u_3) = c(u_4) = 3$ and $c(v_2) = c(v_3) = 2$. Neither u_1 nor v_1 is b-chromatic, so we have a contradiction.

In the remainder of this chapter we study $\varphi(G)$ from the point of view of algorithmic complexity.

3.4 The b-chromatic number in general graphs

In this section we prove that determining $\varphi(G)$ for an arbitrary graph G is hard. Firstly, we derive a useful upper bound for $\varphi(G)$. It is clear that, for a graph G to have a b-chromatic colouring of k colours, G must contain at least k vertices, each of degree at least $k-1$. The following definition leads to a closely related, but stronger observation.

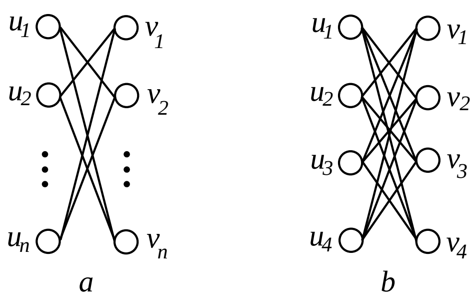

Figure 3.2: Example to show that $\varphi(G)$ can be arbitrarily far from $\chi(G)$.

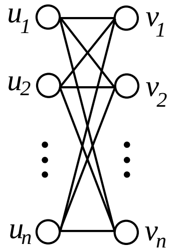

Figure 3.3: Example to show that $\varphi(G)$ can be arbitrarily far from $m(G)$.

Definition 3.4.1 For a graph $G = (V, E)$, suppose that the vertices of G are ordered $v_1, v_2, \ldots, v_n$ such that $d(v_1) \geq d(v_2) \geq \ldots \geq d(v_n)$. Then the *m-degree*, $m(G)$, of G is defined

$$m(G) = \max\{1 \leq i \leq n : d(v_i) \geq i - 1\}. \blacksquare$$

It turns out that $m(G)$ is an upper bound for $\varphi(G)$.

Lemma 3.4.2 *For any graph G, $\varphi(G) \leq m(G)$.*

Proof: The definition of the m-degree implies that there is some set of $m(G)$ vertices of G, each with degree $\geq m(G) - 1$, while the other $|V| - m(G)$ vertices of G each have degree $\leq m(G) - 1$. If $\varphi(G) > m(G)$ then in any b-chromatic colouring of size $\varphi(G)$, there is at least one colour c whose vertices all have degree $\leq m(G) - 1$. For, if not, then there are at least $\varphi(G) > m(G)$ vertices of degree $> m(G) - 1$, a contradiction. Hence all vertices of colour c have degree $< \varphi(G) - 1$, and none of these can be b-chromatic, a contradiction. $\blacksquare$

This upper bound is tight: the graph G of Figure 3.2(b) satisfies $m(G) = 4$, and we have already seen that G has a b-chromatic 4-colouring. On the other hand, $\varphi(G)$ may be arbitrarily far from $m(G)$, as the example of Figure 3.3 shows. For the complete bipartite graph $G = K_{n,n}$ illustrated, $m(G) = n + 1$, whereas $\varphi(G) = 2$, which may be seen as follows. We suppose that G has a b-chromatic colouring of size ≥ 3 and without loss of generality suppose that $c(u_1) = 1$, $c(v_1) = 2$ and v_2 is a b-chromatic vertex for colour 3. Then we have a contradiction, for v_2 cannot be adjacent to a vertex of colour 2.

We now prove that B-CHROMATIC NUMBER DECISION is NP-complete. The proof involves a transformation from the NP-complete problem EXACT COVER BY 3-SETS [92, problem SP2], which may be defined as follows:

> *Name:* EXACT COVER BY 3-SETS (X3C).
> *Instance:* Set $A = \{a_1, a_2, \ldots, a_n\}$ of *elements*, where $n = 3q$ for some q, and a collection $C = \{c_1, c_2, \ldots, c_m\}$ of subsets of A (*clauses*), where $|c_i| = 3$ for each i $(1 \leq i \leq m)$.
> *Question:* Does C contain an exact cover for A, i.e. is there a set C' $(C' \subseteq C)$ of pairwise disjoint sets whose union is A?

Theorem 3.4.3 B-CHROMATIC NUMBER DECISION *is NP-complete.*

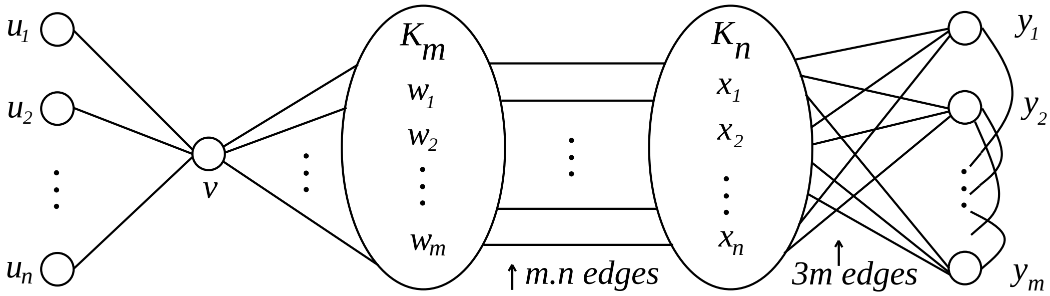

Figure 3.4: Graph G derived from an instance of X3C.

Proof: B-CHROMATIC NUMBER DECISION is certainly in NP, for, given a proper colouring of the vertices we may use the criterion of Proposition 3.3.1 to verify that the colouring is b-chromatic, in polynomial time. To prove NP-hardness, we give a transformation from the X3C problem, as defined above. We suppose that $A = \{a_1, a_2, \ldots, a_n\}$ (where $n = 3q$ for some q), and $C = \{c_1, c_2, \ldots, c_m\}$ (where $c_i \subseteq A$ and $|c_i| = 3$, for each i) is some arbitrary instance of this problem. The X3C problem can easily be transformed to a restricted version of the problem, in which the instance satisfies the following two properties:

1. $\bigcup_{1 \leq i \leq m} c_i = A$
2. $A \neq \emptyset$.

We construct an instance of B-CHROMATIC NUMBER DECISION as follows. Let

$$V = \{u_1, \ldots, u_n, v, w_1, \ldots, w_m, x_1, \ldots, x_n, y_1, \ldots, y_m\},$$

and let E contain the elements

$$\begin{array}{ll} \{u_i, v\} & \text{for } 1 \leq i \leq n, \\ \{v, w_i\} & \text{for } 1 \leq i \leq m, \\ \{w_i, w_j\} & \text{for } 1 \leq i < j \leq m, \\ \{w_i, x_j\} & \text{for } 1 \leq i \leq m, 1 \leq j \leq n, \\ \{x_i, x_j\} & \text{for } 1 \leq i < j \leq n, \\ \{x_i, y_j\} & \text{for } 1 \leq i \leq n, 1 \leq j \leq m \Leftrightarrow a_i \in c_j, \\ \{y_i, y_j\} & \text{for } 1 \leq i < j \leq m \Leftrightarrow c_i \cap c_j \neq \emptyset. \end{array}$$

The resulting graph $G = (V, E)$ is shown in Figure 3.4. We now find $m(G)$ in order to obtain an upper bound for $\varphi(G)$. It may be easily verified that:

- $d(u_i) = 1$ for $1 \leq i \leq n$.
- $d(v) = m + n$.
- $d(w_i) = m + n$ for $1 \leq i \leq m$.
- $d(x_i) \geq m + n$ for $1 \leq i \leq n$ (by Assumption 1 above).

- $d(y_i) \leq 3 + (m-1) < m+n$ (since $n \geq 3$ by Assumption 2 above).

Therefore $m+n+1$ vertices of G have degree at least $m+n$ and all other vertices of G have degree less than $m+n$. Hence $m(G) = m+n+1$ and by Lemma 3.4.2 we have that $\varphi(G) \leq m+n+1$. The claim is that G has a b-chromatic colouring of size $m+n+1$ if and only if C has an exact cover for A.

For, suppose that C has an exact cover $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ for A, where $r \leq m$. We assign $m+n+1$ colours to the vertices of G as follows:

- $c(u_i) = c(x_i) = i$ for $1 \leq i \leq n$,
- $c(v) = m+n+1$,
- $c(w_i) = n+i$ for $1 \leq i \leq m$,
- $c(y_{i_j}) = m+n+1$ for $1 \leq j \leq r$ and
- colour the remaining y_i (i.e. vertices $\{y_1, y_2, \ldots, y_m\} \backslash \{y_{i_1}, y_{i_2}, \ldots, y_{i_r}\}$) by colours $n+1, n+2, \ldots, n+m-r$ respectively.

It remains to show that this colouring is b-chromatic. Certainly the colouring is proper, for the exact cover property gives us that $c_{i_j} \cap c_{i_k} = \varnothing$ for $1 \leq j < k \leq r$, so that $\{y_{i_j}, y_{i_k}\} \notin E$. Also, $m - r < m+1$, so that no y_i such that c_i is not in the exact cover has colour $m+n+1$. We now check that Property 3.2 holds. Take each colour j in turn:

- If $j = m+n+1$ then v is a b-chromatic vertex for colour j.
- If $n+1 \leq j \leq n+m$ then w_{j-n} is a b-chromatic vertex for colour j.
- If $1 \leq j \leq n$ then x_j is adjacent to colours $1, \ldots, j-1, j+1, \ldots, n+m$, plus colour $m+n+1$ by the exact cover property of $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$, so is a b-chromatic vertex for colour j.

Therefore this colouring is b-chromatic and has size $m+n+1$.

Conversely suppose that G has a b-chromatic colouring of size $m+n+1$. Without loss of generality we may assume that $c(x_i) = i$ for $1 \leq i \leq n$ and $c(w_i) = n+i$ for $1 \leq i \leq m$. There is only one remaining vertex of degree at least $m+n$, namely v, so the b-chromatic property forces $c(v) = m+n+1$, and also $u_1, u_2, \ldots, u_n$ must be coloured by some permutation of the colours $\{1, 2, \ldots, n\}$. For each i $(1 \leq i \leq n)$, x_i is the b-chromatic vertex for colour i and hence is adjacent to some y_j such that $c(y_j) = m+n+1$. Thus there is a subcollection $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ for some r $(r \leq m)$ such that, for each j $(1 \leq j \leq n)$, $a_j \in c_{i_k}$ for some k $(1 \leq k \leq r)$. Moreover, the $y_{i_1}, y_{i_2}, \ldots, y_{i_r}$ are all coloured $m+n+1$ so that $\{y_{i_j}, y_{i_k}\} \notin E$ for $1 \leq j < k \leq r$. Hence $c_{i_j} \cap c_{i_k} = \varnothing$ so that $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ forms an exact cover for A. ■

3.5 The b-chromatic number in bipartite graphs

In the previous section, the B-CHROMATIC NUMBER DECISION problem is shown to be NP-complete for arbitary graphs. In fact, the following restricted version of B-CHROMATIC NUMBER DECISION is shown to be NP-complete:

> *Name:* RESTRICTED B-CHROMATIC NUMBER DECISION.
> *Instance:* Graph $G = (V, E)$ and integer $K \in \mathbb{Z}^+$, where $K = m(G)$, G is connected and G has exactly K vertices of degree $\geq K - 1$.
> *Question:* Does G have a b-chromatic K-colouring?

Here we use RESTRICTED B-CHROMATIC NUMBER DECISION to show that the following problem is NP-complete:

> *Name:* BIPARTITE B-CHROMATIC NUMBER DECISION.
> *Instance:* Bipartite graph $G = (V, E)$ and integer $K \in \mathbb{Z}^+$.
> *Question:* Does G have a b-chromatic colouring of $k \geq K$ colours?

Theorem 3.5.1 BIPARTITE B-CHROMATIC NUMBER DECISION *is NP-complete.*

Proof: Let $G = (V, E)$ and $K \in \mathbb{Z}^+$ be an instance of RESTRICTED B-CHROMATIC NUMBER DECISION, where $K = m(G)$, G is connected, and G has exactly K vertices of degree $\geq K - 1$, which we call the *b-chromatic G-vertices.* Let $m = m(G)$ and let $V = \{v_1, v_2, \ldots, v_n\}$ be ordered such that the b-chromatic G-vertices for G are $v_1, v_2, \ldots, v_m$ and suppose that $E = \{e_1, e_2, \ldots, e_q\}$.

We construct an instance $G' = (V', E')$ and $K' \in \mathbb{Z}^+$ of BIPARTITE B-CHROMATIC NUMBER DECISION as follows. We begin with $V' = V \cup \{z_1, z_2, \ldots, z_{2q}\}$ for some new vertices z_r. Corresponding to every edge e_k $(1 \leq k \leq q)$, where $e_k = \{v_i, v_j\}$ for some i, j $(1 \leq i < j \leq n)$, add new vertices a_k, b_k, c_k, d_k and w_k^r, x_k^r for $1 \leq r \leq m+2q-3$. Add the edges $\{v_i, a_k\}, \{b_k, v_j\}, \{v_j, c_k\}, \{d_k, v_i\}, \{v_i, c_k\}, \{v_j, a_k\}$ and $\{a_k, w_k^r\}, \{w_k^r, b_k\}, \{c_k, x_k^r\}, \{x_k^r, d_k\}$, for $1 \leq r \leq m + 2q - 3$, to E'. Corresponding to a b-chromatic G-vertex v_i $(1 \leq i \leq m)$, add the edges $\{v_i, z_r\}$ $(1 \leq r \leq 2q)$ to E'. Finally, set $K' = m + 2q$. Let S denote the set $S = \{1, 2, \ldots, m + 2q\}$. An example component, corresponding to a typical edge of G which is incident to a b-chromatic G-vertex, is shown in Figure 3.5. It may be verified that G' is bipartite.

We firstly show that $m(G') = K'$. Denote the degree of a vertex v in G by $d(v)$, and the degree of a vertex v in G' by $d'(v)$. Then for each i such that $1 \leq i \leq m$,

$$\begin{aligned} d'(v_i) &= 3d(v_i) + 2q \\ &\geq 3(m-1) + 2q \\ &\geq K' - 1. \end{aligned}$$

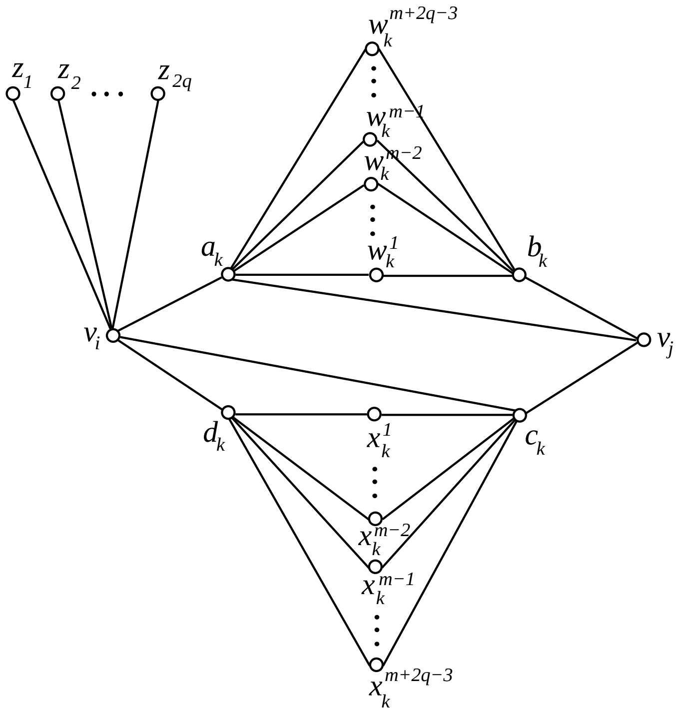

Figure 3.5: Typical edge component in the constructed instance G' of BIPARTITE B-CHROMATIC NUMBER DECISION. The edge component shown corresponds to the edge $e_k = \{v_i, v_j\}$ of G, and here v_i is a b-chromatic G-vertex.

For each i such that $m + 1 \leq i \leq n$,

$$\begin{aligned} d'(v_i) &= 3d(v_i) \\ &< 3(m-1) && \text{(since } v_i \text{ is not a b-chromatic } G\text{-vertex)} \\ &\leq m + 2n - 3 && \text{(since } m \leq n\text{)} \\ &\leq m + 2(q+1) - 3 && \text{(since } G \text{ is connected)} \\ &= K' - 1. \end{aligned}$$

For any k $(1 \leq k \leq q)$, $d'(a_k) = d'(c_k) = K' - 1$, $d'(b_k) = d'(d_k) = K' - 2$, and $d'(w_k^r) = d'(x_k^r) = 2$ for $1 \leq r \leq m + 2q - 3$. Finally, $d'(z_r) = m$ for $1 \leq r \leq 2q$. Hence there are exactly K' vertices in G' of degree $\geq K' - 1$, whilst all other vertices of G' have degree $< K' - 1$. Thus $m(G') = K'$. We call the K' vertices in G' with degree $\geq K' - 1$ the *b-chromatic G'-vertices.*

The claim is that G has a b-chromatic K-colouring if and only if G' has a b-chromatic K'-colouring. Denoting the colour of a vertex v in G by $c(v)$ and the colour of a vertex v in G' by $c'(v)$, we suppose firstly that G has a b-chromatic K-colouring with colours $\{1, 2, \ldots, m\}$. Assign a colouring to G' as follows:

- Set $c'(v_i) = c(v_i)$ for $1 \leq i \leq n$.
- Set $c'(z_i) = m + i$ for $1 \leq i \leq 2q$.

- Set $c'(a_k) = m + 2k - 1$ and $c'(c_k) = m + 2k$ for $1 \leq k \leq q$.
- Set $c'(b_k) = c(v_i)$ and $c'(d_k) = c(v_j)$, for each k $(1 \leq k \leq q)$, where $e_k = \{v_i, v_j\}$, for some i, j $(1 \leq i < j \leq n)$.
- For each k $(1 \leq k \leq q)$, assign colours in G' to the w_k^r, x_k^r $(1 \leq r \leq m + 2q - 3)$ such that
$$\{c'(w_k^1), c'(w_k^2), \ldots, c'(w_k^{m+2q-3})\} = S \backslash \{c(v_i), c(v_j), m + 2k - 1\}$$
and
$$\{c'(x_k^1), c'(x_k^2), \ldots, c'(x_k^{m+2q-3})\} = S \backslash \{c(v_i), c(v_j), m + 2k\},$$
where $e_k = \{v_i, v_j\}$, for some i, j $(1 \leq i < j \leq n)$.

It then follows that this colouring for G' is a b-chromatic K'-colouring.

Conversely, suppose that G' has a b-chromatic K'-colouring. Without loss of generality we may suppose that the b-chromatic G'-vertices are coloured such that $c'(v_i) = i$ for $1 \leq i \leq m$, $c'(a_k) = m + 2k - 1$ and $c'(c_k) = m + 2k$ for $1 \leq k \leq q$.

Now let k $(1 \leq k \leq q)$ be given and suppose that $e_k = \{v_i, v_j\}$ for some i and j $(1 \leq i < j \leq n)$. Vertex a_k is the b-chromatic G'-vertex for colour $m + 2k - 1$, and $d'(a_k) = m + 2q - 1$. Hence $c'(v_j) \neq c'(v_i)$, and
$$\{c'(w_k^1), c'(w_k^2), \ldots, c'(w_k^{m+2q-3})\} = S \backslash \{c'(v_i), c'(v_j), m + 2k - 1\},$$
which implies that $c'(b_k) = c'(v_i)$ or $c'(b_k) = m + 2k - 1$.

Similarly, vertex c_k is the b-chromatic G'-vertex for colour $m + 2k$, and $d'(c_k) = m + 2q - 1$. Hence
$$\{c'(x_k^1), c'(x_k^2), \ldots, c'(x_k^{m+2q-3})\} = S \backslash \{c'(v_i), c'(v_j), m + 2k\},$$
which implies that $c'(d_k) = c'(v_j)$ or $c'(d_k) = m + 2k$.

As each z_i $(1 \leq i \leq 2q)$ is adjacent in G' to every b-chromatic G-vertex, it follows that
$$\{c'(z_1), c'(z_2), \ldots, c'(z_{2q})\} \subseteq \{m + 1, m + 2, \ldots, m + 2q\}.$$
Hence, for each v_i $(1 \leq i \leq m)$ there are $m - 1$ vertices $f_{k_{i,r}}$, for $1 \leq r \leq m - 1$, for some $k_{i,r}$ $(1 \leq k_{i,r} \leq q)$, adjacent in G' to v_i, such that each $f_{k_{i,r}}$ is either $b_{k_{i,r}}$ or $d_{k_{i,r}}$, and
$$\{c'(f_{k_{i,1}}), c'(f_{k_{i,2}}), \ldots, c'(f_{k_{i,m-1}})\} = \{1, \ldots, i - 1, i + 1, \ldots, m\}.$$
Consider such a vertex $f_{k_{i,r}}$, for any i and r $(1 \leq i \leq m,\ 1 \leq r \leq m - 1)$. If $f_{k_{i,r}} = b_{k_{i,r}}$, then $e_{k_{i,r}} = \{v_j, v_i\}$ for some j $(1 \leq j < i)$, where $b_{k_{i,r}}$ is adjacent in G' to v_i. In this case $c'(b_{k_{i,r}}) = c'(v_j)$. If $f_{k_{i,r}} = d_{k_{i,r}}$, then $e_{k_{i,r}} = \{v_i, v_j\}$ for some j $(i < j \leq n)$, where $d_{k_{i,r}}$ is adjacent in G' to v_i. In this case $c'(d_{k_{i,r}}) = c'(v_j)$.

Hence, if we let $c(v_i) = c'(v_i)$ for every i $(1 \leq i \leq n)$ such that $c'(v_i) \leq m$, then we obtain a partial proper colouring of G such that every b-chromatic G-vertex, coloured i (for $1 \leq i \leq m$), is adjacent in G to a vertex coloured j, for each j $(1 \leq j \neq i \leq m)$. For

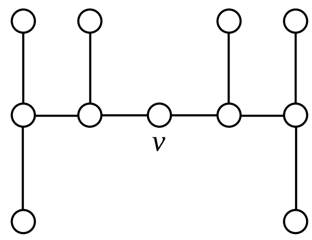

Figure 3.6: Example pivoted tree.

every i $(1 \le i \le n)$ such that $c'(v_i) > m$, v_i is not a b-chromatic G-vertex, so $d(v_i) < m-1$. Hence we may set $c(v_i)$ equal to any colour chosen from the set $\{1, 2, \ldots, m\}$ that has not already been assigned to a neighbour (in G) of v_i, and we have achieved a b-chromatic colouring for G of K colours. ■

3.6 The b-chromatic number in trees

In contrast with the NP-hardness of ACHROMATIC NUMBER for trees [30], we show in this section that the b-chromatic number is polynomial-time computable for trees. In fact, apart from a very special class of exceptions, recognisable in polynomial time, the b-chromatic number of a tree T is equal to the upper bound $m = m(T)$.

Let us call a vertex v of T such that $d(v) \ge m-1$ a *dense* vertex of T. Our methods of finding b-chromatic colourings for trees hinge on colouring firstly vertices adjacent to those in a set $V' = \{v_1, v_2, \ldots, v_m\}$ of dense vertices of T. For trees with more than m dense vertices, we shall demonstrate how V' is to be chosen. We aim to establish a *partial b-chromatic m-colouring* of T, i.e., a partial proper colouring of T using m colours such that each v_i $(1 \le i \le m)$ has colour i and is adjacent to vertices of $m-1$ distinct colours. This approach is applicable for all trees except those satisfying the following criteria.

Definition 3.6.1 A tree $T = (V, E)$ is *pivoted* if T has exactly m dense vertices, and T contains a distinguished vertex v such that:

1. v is not dense.

2. Each dense vertex is adjacent either to v or to a dense vertex adjacent to v.

3. Any dense vertex adjacent to v and to another dense vertex has degree $m-1$.

We call such a vertex v a *pivot* of T. ■

An example of a pivoted tree T is shown in Figure 3.6. Here $m(T) = 4$ and T is pivoted at v. We now establish two properties of pivoted trees.

Proposition 3.6.2 *Let $T = (V, E)$ be a pivoted tree, and let v be a pivot of T. Then*

1. *There are at least two dense vertices of T adjacent to v.*

2. *There is a dense vertex u of T adjacent to v such that u is adjacent to some dense vertex w of T.*

Proof of (1): Suppose not. Let u be the sole dense vertex adjacent to v. Then, as T contains m dense vertices, u is adjacent to $m-1$ dense vertices of T by Property 2 of Definition 3.6.1. Thus $d(u) \geq m$, contradicting Property 3 of Definition 3.6.1.

Proof of (2): Suppose not. Then v is adjacent to every dense vertex of T by Property 2 of Definition 3.6.1. Since there are m dense vertices, $d(v) \geq m$, contradicting Property 1 of Definition 3.6.1. ∎

A pivot is unique if it exists, which we now show.

Proposition 3.6.3 *Let $T = (V, E)$ be a pivoted tree. Then T contains a unique pivot v.*

Proof: Suppose not. Then T has two pivots v_1 and v_2. By Part 2 of Proposition 3.6.2, let u be a dense vertex adjacent to v_1 such that w is a dense vertex adjacent to u. By Part 1 of Proposition 3.6.2, let $u' \neq u$ be a dense vertex adjacent to v_1. Then it is straightforward to verify that v_2 cannot be a non-dense vertex in T at distance at most two from each of u, u' and w, a contradiction to the defining properties of v_2. ∎

The following result demonstrates that pivoted trees may be recognised easily.

Proposition 3.6.4 *Given a tree $T = (V, E)$, we may test for T being pivoted in linear time.*

Proof: Let V' be the set of dense vertices of T. For T to be pivoted, we must have $|V'| = m$. Consider the subtrees $T_1, T_2, \ldots, T_r$ in the connected components of the subgraph T' of T induced by V'. Let $d'(v)$ denote the T'-degree of a vertex $v \in V'$. If T is pivoted, then for each i $(1 \leq i \leq r)$, exactly one of the following three disjoint cases holds:

1. T_i is an isolated vertex u_i.

2. T_i is a K_2 with vertices u_i and v_i.

3. T_i contains a unique vertex u_i such that $d'(u_i) > 1$.

Let u_i, v_i be defined according to Cases 1,2,3 above (v_i is undefined in cases 1,3). Consider T_1 and T_2 (if T is pivoted then $r \geq 2$ by Part 1 of Proposition 3.6.2). If T is pivoted then there exist vertices $w_1 \in V(T_1)$, $w_2 \in V(T_2)$ and a vertex $v \in V \backslash V'$, adjacent in T to both w_1 and w_2, where w_i can be u_i, or v_i if v_i exists $(i = 1, 2)$. (Such a v is unique if it exists). Finally, T is pivoted if and only if v satisfies Properties 2 and 3 of Definition 3.6.1. It is clear that each of these verifications may be carried out in $O(n)$ time. ∎

Returning to the example pivoted tree T of Figure 3.6, in which $m(T) = 4$, it is straightforward to verify that T does not have a b-chromatic 4-colouring, but T has a b-chromatic 3-colouring. The following theorem establishes the b-chromatic number of arbitrary pivoted trees. In addition, the proof of the result shows how to construct a maximum b-chromatic colouring of pivoted trees in polynomial time.

Theorem 3.6.5 *If $T = (V, E)$ is a tree that is pivoted then $\varphi(T) = m(T) - 1$.*

Proof: Denote by v the pivot of T. Let $V = \{v_1, v_2, \ldots, v_n\}$ be ordered so that $V' = \{v_1, v_2, \ldots, v_m\}$ is the set of dense vertices, $v_1, v_2, \ldots, v_p$ (for some $p \leq m$) are the dense vertices adjacent to v, and $v_1, v_2, \ldots, v_q$ (for some $q \leq p$) are the dense vertices adjacent to v each having at least one dense vertex as a neighbour. Then $p \geq 2$ by Part 1 of Proposition 3.6.2 and $q \geq 1$ by Part 2 of Proposition 3.6.2.

Firstly we show that $\varphi(T) < m(T)$. For, suppose that there is a b-chromatic colouring c of T, using m colours, where, without loss of generality, $c(v_i) = i$ $(1 \leq i \leq m)$. As $d(v_j) = m - 1$ for $1 \leq j \leq q$, none of $v_1, v_2, \ldots, v_q$ can be adjacent to more than one vertex of any one colour. Between them, $v_1, v_2, \ldots, v_q$ are adjacent to dense vertices $v_{p+1}, v_{p+2}, \ldots, v_m$. Now v cannot have colour j for $1 \leq j \leq p$, nor colour j for $p+1 \leq j \leq m$, or else some v_k $(1 \leq k \leq q)$ is adjacent to two vertices of that colour. Hence there is no available colour for v, a contradiction.

To establish equality, we construct a b-chromatic colouring c of T using $m-1$ colours. As $p \geq 2$ and $q \geq 1$, the dense vertices v_1, v_2 are adjacent to v and for some r $(p+1 \leq r \leq m)$, there is a dense vertex v_r adjacent to v_1. Set $c(v_i) = i$ for $2 \leq i \leq m$, let $c(v) = r$ and assign $c(v_1) = 2$. All other vertices of V are as yet uncoloured. We now show how to extend this partial colouring into a b-chromatic $(m-1)$-colouring of T, namely a proper $(m-1)$-colouring of V, using colours $2, 3, \ldots, m$, such that every vertex in $V' \backslash \{v_1\}$ is adjacent to vertices of $m-2$ distinct colours, as follows. For $2 \leq i \leq m$, let $R_i = \{2, 3, \ldots, m\} \backslash \{i\}$ (the *required colours* for surrounding v_i), let

$$C_i = \{c(v_j) : 1 \leq j \leq n \wedge v_j \in N(v_i) \wedge v_j \text{ is coloured}\}$$

(the *existing colours* around v_i) and define

$$U_i = \{v_j : m+1 \leq j \leq n \wedge v_j \in N(v_i) \wedge v_j \text{ is uncoloured}\}$$

(the *uncoloured vertices* adjacent to v_i). By construction, v_i is not adjacent to two vertices of the same colour. Hence

$$|C_i| + |U_i| = d(v_i) \geq m - 1 > m - 2 = |R_i|.$$

Hence, as $C_i \subseteq R_i$, it follows that $|U_i| \geq |R_i \backslash C_i|$. Thus if $R_i \backslash C_i = \{r_1^i, \ldots, r_{n_i}^i\}$ (for some $n_i \geq 0$) then we may pick some $\{u_1^i, \ldots, u_{n_i}^i\} \subseteq U_i$ and set $c(u_j^i) = r_j^i$ for $1 \leq j \leq n_i$. This process does not assign the same colour to any two adjacent vertices, since no two adjacent non-dense vertices are both adjacent to dense vertices. Nor does it assign more than one colour to any one vertex, since no two dense vertices have a common non-dense neighbour (except for v, which is already coloured).

For $m+1 \leq i \leq n$, suppose that v_i is uncoloured. As $d(v_i) < m - 1$, not all of colours $2, 3, \ldots, m$ appear on neighbours of v_i. Hence there is some colour available for v_i. It follows that the constructed colouring is a b-chromatic $(m-1)$-colouring of T. ■

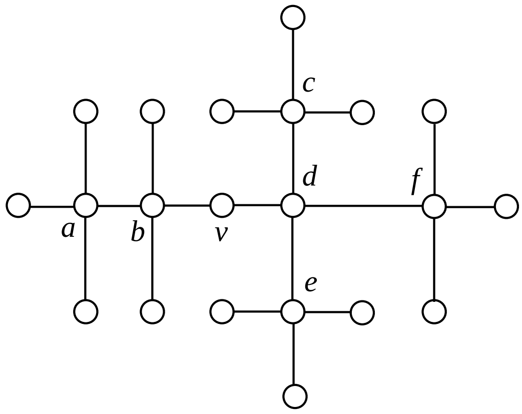

Figure 3.7: Example tree T with $m(T) = \varphi(T) = 5$.

We now show how to construct a b-chromatic colouring of an arbitrary non-pivoted tree, using $m = m(T)$ colours. For trees with more than m dense vertices, we need to be aware of a possible complication. For example, consider the tree T in Figure 3.7. T satisfies $m(T)$=5, but there are six dense vertices: a, b, c, d, e, f. It may be verified that no b-chromatic 5-colouring of T exists in which a, b, c, d, e are the b-chromatic vertices. However, a b-chromatic 5-colouring of T *does* exist, which may be achieved by taking either a, b, c, e, f or a, c, d, e, f to be the b-chromatic vertices.

Hence, as the example of Figure 3.7 shows, a judicious choice of dense vertices may be required in order to achieve a partial b-chromatic m-colouring. In order to assist in making this selection, we formulate the following definition, which is closely related to the concept of a tree being pivoted.

Definition 3.6.6 Let $T = (V, E)$ be a tree, and let V' be the set of dense vertices of T. Suppose that V'' is a subset of V' of cardinality m. Then V'' *encircles* some vertex $v \in V \backslash V''$ if:

1. Each vertex in V'' is adjacent either to v or to some vertex in V'' adjacent to v.

2. Any vertex in V'' adjacent to v and to another vertex in V'' has degree $m - 1$.

We refer to v as an *encircled vertex* with respect to V''. ■

We now give an additional definition that incorporates this concept of encirclement.

Definition 3.6.7 Let $T = (V, E)$ be a tree, and let V' be the set of dense vertices of T. Suppose that V'' is a subset of V' of cardinality m. Then V'' is a *good set* with respect to T if:

(a) V'' does not encircle any vertex in $V \backslash V''$.

(b) Any vertex $u \in V \backslash V''$ such that $d(u) \geq m$ is adjacent to some $v \in V''$ with $d(v) = m - 1$. ■

In the example of Figure 3.7, the set $\{a, b, c, d, e\}$ encircles vertex v. However, either of $\{a, b, c, e, f\}$ or $\{a, c, d, e, f\}$ is a good set with respect to T. In general, our aim is to build up a b-chromatic m-colouring by choosing a good set with respect to the given tree T. The following lemma describes how we make this choice, and also shows that such a choice is always possible in non-pivoted trees.

Lemma 3.6.8 *Let $T = (V, E)$ be a tree that is not pivoted. Then we may construct a good set for T.*

Proof: Let V' be the set of dense vertices of T. By the definition of $m(T)$, we may choose a subset V'' of V', with $|V''| = m$, so that every vertex in $V \backslash V''$ has degree less than m. Let $V = \{v_1, v_2, \ldots, v_n\}$ be ordered so that $V'' = \{v_1, v_2, \ldots, v_m\}$. Suppose that V'' encircles some vertex $v \in V \backslash V''$ (for if not, we set $W = V''$ and we are done, since W satisfies Properties (a) and (b) of Definition 3.6.7). Without loss of generality, suppose that $v_1, v_2, \ldots, v_p$ (for some $p \leq m$) are the members of V'' adjacent to v, and $v_1, v_2, \ldots, v_q$ (for some $q \leq p$) are the members of V'' adjacent to v, each having at least one other member of V'' as a neighbour. Now $p \geq 2$, for otherwise $d(v_1) \geq m$ as each of $v_2, \ldots, v_m$ is adjacent to v_1 by Property 1 of Definition 3.6.6, contradicting Property 2 of Definition 3.6.6. Also $q \geq 1$, for otherwise $p = m$ by Property 1 of Definition 3.6.6, so $d(v) \geq m$, a contradiction to the choice of V''. Thus there is a vertex $v_r \in V''$, for some r $(p + 1 \leq r \leq m)$, adjacent to v_1. We consider two cases.

Case (i): v is dense. Then $d(v) = m - 1$ by the choice of V''. Let $W = (V'' \backslash \{v_2\}) \cup \{v\}$. Also by the choice of V'', the only vertex not in W that can have degree at least m is v_2. But v_2 is adjacent to $v \in W$, and $d(v) = m - 1$, so that W satisfies Property (b) of Definition 3.6.7. W also satisfies Property (a) of Definition 3.6.7. For no vertex $w \in V \backslash (W \cup \{v_2\})$, adjacent to v_j, for some j $(p + 1 \leq j \leq m)$, may be encircled by W, since v is at distance 3 from w. Also, no vertex $w \in V \backslash W$, adjacent to v_j, for some j $(1 \leq j \leq p)$, may be encircled by W, since there is some $v_k \in V''$ $(1 \leq k \leq p)$, adjacent to v, at distance 3 from w (as $p \geq 2$). Finally, no vertex w of $V \backslash W$, adjacent to v, may be encircled by W, since v_r is at distance 3 from w.

Case (ii): v is not dense. If $|V'| = m$ then T is pivoted at vertex v, a contradiction. Hence $|V'| > m$, so there is some $u \in V' \backslash V''$. Let $W = (V'' \backslash \{v_1\}) \cup \{u\}$. Now suppose that W encircles some vertex x. At most one vertex not in W lies on the path between any pair of non-adjacent vertices in W, namely x. But $v_1 \notin W$ and $v \notin W$ lie on the path between $v_2 \in W$ and $v_r \in W$. This contradiction implies that W satisfies Property (a) of Definition 3.6.7. Also, W satisfies Property (b) of Definition 3.6.7, since $d(v) < m - 1$ and $d(v_1) = m - 1$, and therefore every vertex outside W has degree less than m. ■

The following theorem establishes the b-chromatic number of trees that are not pivoted. In addition, the proof of the result shows how to construct a maximum b-chromatic colouring of non-pivoted trees in polynomial time.

Theorem 3.6.9 *If $T = (V, E)$ is a tree that is not pivoted, then $\varphi(T) = m(T)$.*

Proof: As T is not pivoted then, by Lemma 3.6.8, we may choose a good set W of the dense vertices of T. Henceforth we assume that the vertices of T are ordered $v_1, v_2, \ldots, v_n$ so that $W = \{v_1, v_2, \ldots, v_m\}$. We refer to the vertices in W as the *b-chromatic candidates* of T. Initially, we let $c(v_i) = i$ for $1 \leq i \leq m$; each vertex v_i, for $m+1 \leq i \leq n$, is as yet uncoloured.

A partial b-chromatic m-colouring of T could easily be established if each pair of distinct b-chromatic candidates were at distance at least 4. We deal with the general case, where some pairs are separated by a distance of at most 3, as follows. Let T' be the subgraph of T induced by W and the vertices on paths of length at most 3 between any pair of distinct b-chromatic candidates. Let $T_1, T_2, \ldots, T_r$ be the connected components of the forest T'.

Recall that, for any $v \in V$, $d(v)$ is the degree of v in T. For any $v \in V(T')$, we define $d'(v)$ to be the degree of v in T'. Suppose we can establish a *partial b-chromatic subtree colouring* for each T_j $(1 \leq j \leq r)$, that is a proper colouring of T_j, such that if $v_i \in V_j$ (for some i where $1 \leq i \leq m$),

- $d(v_i) = m - 1$ implies that all vertices adjacent to v_i in T_j have distinct colours.
- $d(v_i) > m - 1$ implies that at most two vertices adjacent to v_i in T_j have the same colour.

Then we may obtain a partial b-chromatic m-colouring of T, and hence a full proper b-chromatic m-colouring of T. For, suppose that v_i $(1 \leq i \leq m)$ is any b-chromatic candidate. Let $R_i = \{1, 2, \ldots, m\} \backslash \{i\}$ (the *required colours* for surrounding v_i), let

$$C_i = \{c(v_j) : 1 \leq j \leq n \wedge v_j \in N(v_i) \wedge v_j \text{ is coloured}\}$$

(the *existing colours* around v_i) and define

$$U_i = \{v_j : m+1 \leq j \leq n \wedge v_j \in N(v_i) \wedge v_j \text{ is uncoloured}\}$$

(the *uncoloured vertices* adjacent to v_i). By definition of the partial b-chromatic subtree colouring, $U_i \subseteq V \backslash V(T')$. If $d(v_i) = m - 1$ then by construction, all vertices adjacent to v_i in T_j have distinct colours. Hence

$$|C_i| + |U_i| = d(v_i) = m - 1 = |R_i|.$$

If $d(v_i) > m - 1$ then by construction, v_i is adjacent in T_j to at most two vertices of the same colour. Hence

$$|C_i| + |U_i| \geq d(v_i) - 1 \geq m - 1 = |R_i|.$$

Hence in both cases, $|U_i| \geq |R_i \backslash C_i|$, as $C_i \subseteq R_i$ and $C_i \cap U_i = \varnothing$. Thus if $R_i \backslash C_i = \{r_1^i, \ldots, r_{n_i}^i\}$ (for some $n_i \geq 0$) then we may pick some $\{u_1^i, \ldots, u_{n_i}^i\} \subseteq U_i$ and set $c(u_j^i) = r_j^i$ for $1 \leq j \leq n_i$. This colouring is proper, for each u_j^i is adjacent to v_i, but $u_j^i \notin V(T')$, so that u_j^i is at distance at least 3 from any b-chromatic candidate v_r $(r \neq i)$. Also, the colouring process does not assign more than one colour to any one vertex, since no two b-chromatic candidates have a common neighbour in U_i, for any i, since $U_i \subseteq V \backslash V(T')$.

For $m+1 \leq i \leq n$, suppose that v_i is uncoloured. If $d(v_i) \geq m$, then by construction of W, v_i is adjacent to some b-chromatic candidate of degree $m-1$, and hence v_i has already received a colour. Thus $d(v_i) < m$, and not all of colours $1, 2, \ldots, m$ appear on neighbours of v_i. Hence there is some colour available for v_i. It follows that the constructed colouring is a b-chromatic m-colouring of T.

Therefore it remains to establish a partial b-chromatic subtree colouring of each T_j. The colouring process is in two stages, and identical for each subtree, so we assume that a fixed j $(1 \leq j \leq r)$ is given. Suppose that V_j contains b-chromatic candidates $v_{j,1}, v_{j,2}, \ldots, v_{j,n_j}$ for some $n_j > 0$.

Stage 1 *(for each $k = 1, 2, \ldots, n_j$ in turn)*. Let S be the set of uncoloured vertices adjacent to $v_{j,k}$ in T_j. If $S = \varnothing$ then no vertices are coloured at this step. Now suppose that $S = \{w\}$. If there is some path of length 2 in T_j from w to a b-chromatic candidate u then we set $c(w) = c(u)$. Otherwise (if $S = \{w\}$) no vertices are coloured at this step. Finally suppose that $S = \{w_1, w_2, \ldots, w_s\}$ for some $s > 1$. For each i $(1 \leq i \leq s)$, w_i is on a path of ≤ 3 edges in T_j from $v_{j,k}$ to some other b-chromatic candidate: let c_i be the colour of one such b-chromatic candidate. Set $c(w_i) = c_{i+1}$ for $1 \leq i \leq s-1$ and $c(w_s) = c_1$.

Stage 2 *(for each uncoloured vertex u in T_j)*. Vertex u was not coloured by Stage 1 because every vertex adjacent to u in T_j is a b-chromatic candidate (if there was a non b-chromatic candidate adjacent to u then u would have been coloured when Stage 1 considered some b-chromatic candidate adjacent to u). Also, for any $v \in N(u) \backslash V(T_j)$, $v \notin V(T')$. Let $N'(u)$ denote the vertices adjacent to u in T'. It is clear that, for any $v \in N'(u)$ and $w \in N'(v) \backslash \{u\}$, w is coloured. For if not, both u and w would have been coloured cyclically when Stage 1 considered the b-chromatic vertex v, a contradiction. Hence define

$$S = \{1, 2, \ldots, m\} \backslash \left(\left(\bigcup_{v \in N'(u)} \{c(v)\} \right) \cup \left(\bigcup_{v \in N'(u)} \left\{ c(w) : \begin{array}{c} w \in N'(v) \backslash \{u\} \\ \wedge\ d(v) = m-1 \end{array} \right\} \right) \right).$$

Choose $c(u)$ to be an arbitrary member of S (we show later that S is always nonempty).

We prove that the algorithm produces a partial b-chromatic subtree colouring. It may be verified that Stage 1 yields a partial colouring of T such that no vertex is adjacent to a vertex of the same colour and that no b-chromatic candidate is adjacent to more than one vertex of a certain colour. Clearly if every set S constructed by Stage 2 is nonempty then we would terminate with the desired colouring. For, at most one vertex adjacent to any b-chromatic candidate is coloured by Stage 2 (or else Stage 1 would have coloured these vertices). Hence, by definition of S, any b-chromatic candidate of degree more than $m-1$ is adjacent in T' to at most two vertices of the same colour, and all vertices adjacent in T' to any b-chromatic candidate of degree exactly $m-1$ have distinct colours.

Hence suppose that there is some step of Stage 2 where the constructed set S for some uncoloured vertex u is empty. Then u is adjacent in T' to b-chromatic candidates $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ (for some i_r and k), such that (without loss of generality), $d(v_{i_r}) = m-1$ for $1 \leq r \leq l$ (for some $l \geq 0$) and

$$\bigcup_{1\leq r\leq k} \{c(v_{i_r})\} \cup \left(\bigcup_{1\leq r\leq l} \{c(w) : w \in N'(v_{i_r})\backslash\{u\}\} \right) = \{1, 2, \ldots, m\}. \tag{3.3}$$

Considering the cardinalities of the sets on both sides of Equation 3.3 gives

$$d'(v_{i_1}) + d'(v_{i_2}) + \ldots + d'(v_{i_l}) + (k-l) \geq m. \tag{3.4}$$

For each r $(1 \leq r \leq k)$, there are at least $d'(v_{i_r}) - 1$ paths, not passing through u, leading from v_{i_r} to a b-chromatic candidate, by definition of T_j. But there are only $m-k$ b-chromatic candidates remaining. Hence

$$d'(v_{i_1}) + d'(v_{i_2}) + \ldots + d'(v_{i_k}) - k \leq m - k. \tag{3.5}$$

Combining Inequalities 3.4 and 3.5 gives

$$d'(v_{i_{l+1}}) + d'(v_{i_{l+2}}) + \ldots + d'(v_{i_k}) \leq k - l. \tag{3.6}$$

But $d'(v_{i_r}) \geq 1$ for all r such that $1 \leq r \leq k$. Hence, from Inequality 3.6 we deduce that $d'(v_{i_r}) = 1$ for all r such that $l+1 \leq r \leq k$. Substituting into Inequality 3.5 and again combining Inequalities 3.4 and 3.5 gives

$$d'(v_{i_1}) + d'(v_{i_2}) + \ldots + d'(v_{i_l}) + (k-l) = m. \tag{3.7}$$

Let $p(v_{i_r})$ $(1 \leq r \leq l)$ denote the number of paths, not passing through u, leading from v_{i_r} to distinct b-chromatic candidates. Since $p(v_{i_1}) + p(v_{i_2}) + \ldots + p(v_{i_l}) \leq m-k$, and $p(v_{i_r}) \geq d'(v_{i_r}) - 1$ for each r $(1 \leq r \leq l)$, we can deduce (from Equation 3.7) that

$$p(v_{i_1}) + p(v_{i_2}) + \ldots + p(v_{i_l}) = d'(v_{i_1}) + d'(v_{i_2}) + \ldots + d'(v_{i_l}) - l.$$

But $p(v_{i_r}) \geq d'(v_{i_r}) - 1$ and hence $p(v_{i_r}) = d'(v_{i_r}) - 1$, for each r $(1 \leq r \leq l)$. Hence, for each $1 \leq r \leq l$, there are exactly $d'(v_{i_r}) - 1$ distinct paths, of length p_s^r, not passing through u, each leading from v_{i_r} to another b-chromatic candidate, where $1 \leq s \leq d(v_{i_r}) - 1$ and $1 \leq p_s^r \leq 3$, by definition of E_j. As already discussed, it is impossible that two or more vertices adjacent to a v_{i_r} will be coloured by Stage 2. Also, no vertex adjacent to a v_{i_r} and distinct from u was coloured by Stage 1 (or else u would also have been coloured cyclically by Stage 1). Thus $p_s^r = 1$ for each r $(1 \leq r \leq l)$ and s $(1 \leq s \leq d(v_{i_r}) - 1)$. Therefore u is encircled by W, contradicting the choice of W. ■

3.7 Conclusion and further study relating to the b-chromatic number

In this chapter we have defined two partial orders on the set of all proper colourings for a graph, giving rise to the ACHROMATIC NUMBER and B-CHROMATIC NUMBER problems. The polynomial-time solvability result for B-CHROMATIC NUMBER in trees may be contrasted with the NP-completeness of ACHROMATIC NUMBER DECISION in trees. Thus CHROMATIC NUMBER is an example of optimisation problem such that two partial orders, namely $\prec_a^G$ and $\prec_b^G$, defined on $\mathcal{F}(G)$, admit maximinimal problems of differing complexity, for a particular restriction on the instance G.

The complexity of ACHROMATIC NUMBER has been studied for a number of graph classes, though there is much scope for further study of the complexity of B-CHROMATIC NUMBER when restricted to certain classes of graph. For example, the complexity of B-CHROMATIC NUMBER is open for planar graphs, although we conjecture this problem to be NP-complete also.

Attempts were made to find an approximation algorithm for B-CHROMATIC NUMBER with a constant performance guarantee by trying to relate $\varphi(G)$ to $m(G)$. Although the example of Figure 3.3 ruled out hopes of this method succeeding, the existence of such an approximation algorithm is open.

Chapter 4

Minimaximal and maximinimal graph problems based on the partial order of set inclusion

4.1 Introduction

In this chapter we consider minimaximal and maximinimal graph optimisation problems whose definitions incorporate the partial order of set inclusion. We study these problems from the point of view of algorithmic complexity. The chapter is arranged as follows: in Section 4.2 we consider twelve covering and independence graph problems, of which six are minimaximal or maximinimal optimisation problems, whilst in Section 4.3 we examine twelve strong stability, clique, domination and irredundance graph problems, of which six are minimaximal or maximinimal optimisation problems.

The twelve covering and independence graph problems are grouped together because of a framework suggested by four of these parameters: the *total covering* and *total matching* parameters. Total coverings and total matchings of graphs (defined in Section 4.2) are an important concept as they fuse together the notions of vertex and edge covering, yet they have not been extensively studied, relative to the concepts of vertex and edge covering and independence. We briefly survey the non-algorithmic work to date on total coverings and total matchings of graphs, and present the framework, due to Nordhaus [172], that can be obtained from the four associated graph parameters. We study the algorithmic complexity of the twelve optimisation problems related to the parameters in this framework, over several classes of graphs. The classes that we consider include, in each case, four extensively studied classes of graphs, namely planar, bipartite and chordal[1] graphs, and trees[2]. Definitions of other graph classes mentioned here but not defined may

[1] A graph G is *chordal* if every cycle in G of length four or more contains a *chord*, i.e. an edge connecting two non-adjacent points on the cycle

[2] Many NP-complete graph problems become polynomial-time solvable when restricted to trees. In addition, many graph problems that are solvable in polynomial time for trees are also solvable in polynomial time for graphs of bounded treewidth (i.e., partial k-trees) [6, 5, 204].

be found in [99] and [134]. Henceforth we refer to 'the complexity of α' when we mean 'the complexity of the decision problem related to graph parameter α'. We survey briefly known results for graph classes that include at least the four mentioned above, and obtain new NP-completeness results for the following parameters:

- Maximum minimal total cover in planar graphs.
- Minimum maximal total matching in bipartite and chordal graphs.
- Minimum independent dominating set in planar cubic graphs.

In addition, we demonstrate that the complexities of the maximum minimal edge cover, maximum minimal vertex cover and maximum total matching parameters are identical to the complexities of the minimum dominating set, minimum independent dominating set and minimum edge dominating set parameters respectively, over all graph classes. These results do not appear to have been noted explicitly in the literature previously.

The remaining twelve graph parameters studied algorithmically in this chapter are connected with strong stability, cliques, domination and irredundance, and are considered in Section 4.3. We survey their algorithmic complexity for several graph classes including, again, planar, bipartite and chordal graphs, and trees. We obtain new NP-completeness results for the following parameters:

- Minimum maximal strong stable set in planar graphs of maximum degree 3.
- Minimum maximal clique in general graphs.
- Minimum total dominating set in planar cubic graphs.

As mentioned above, all of the minimaximal and maximinimal optimisation problems studied in this chapter are formulated using the partial order of set inclusion, defined on the feasible solutions for a given instance of a source optimisation problem. Thus all occurrences of the words 'maximal' and 'minimal' in this chapter refer to maximality and minimality with respect to the partial order of set inclusion. It may be verified that, for each source optimisation problem in this chapter, the set of feasible solutions for a given instance is either hereditary or super-hereditary. Thus, by Proposition 2.4.6, each concept of maximality or minimality with respect to the partial order of set inclusion is equivalent to maximality or minimality with respect to the partial order of $(0,1)$-replacement. However, in the literature, definitions of maximality or minimality for classical graph-theoretic concepts, such as vertex covers or independent sets, are usually formulated in terms of the partial order of set inclusion (see Berge [17, p.10], for example). Thus, to be consistent with these definitions, we choose the partial order of set inclusion.

For each source optimisation problem, we have $\mathcal{I} = \{G = (V, E) : G \text{ is a graph}\}$, and for $G \in \mathcal{I}$, $m(G, y) = |y|$ for each $y \in \mathcal{F}(G)$. Thus each source problem is defined completely by supplying the definitions of $\mathcal{U}, \pi$ and OPT. When defining $\mathcal{U}$, we assume that $G \in \mathcal{I}$ is given, and when defining π, we assume that $G \in \mathcal{I}$ and some $S \in \mathcal{U}(G)$ have been given.

4.2 Minimaximal and maximinimal covering and independence graph problems

4.2.1 Introduction

In graph theory, the notion of covering vertices or edges of graphs by other vertices or edges has been extensively studied. For instance, covering vertices by other vertices leads to parameters concerned with vertex domination [115]. When edges are to be covered by vertices we obtain parameters connected with the classical vertex covering problem [106, p.94]. Covering vertices by edges, i.e. finding edge covers, is considered by Norman and Rabin [174]. Finally, when edges are to cover other edges, we obtain parameters associated with edge domination (introduced by Mitchell and Hedetniemi [168]). Independent sets of vertices [106, p.95] correspond to the case where vertices are chosen so as *not* to cover one another, and matchings [157] of a graph correspond to the similar restriction involving edges.

It is natural to extend this notion of covering by vertices and edges. Nordhaus [172], and also Alavi et al. [2], define the *elements* of a graph $G = (V, E)$ to be the set $V \cup E$. A vertex v is defined to cover itself, all edges incident on v and all vertices adjacent to v. An edge $\{u, v\}$ is said to cover itself, vertices u and v, and all edges incident on u or v. Two elements of $V \cup E$ are *independent* if neither covers the other. Thus, a *vertex cover* is a subset S of V that covers E, a *dominating set* is a subset S of V that covers V (in this chapter, the term *dominating set* will only apply to a set of vertices), an *edge dominating set* is a subset S of E that covers E, and an *edge cover* is a subset S of E that covers V (assuming that G has no isolated vertices). A subset C of $V \cup E$ that covers all elements of G is said to be a *total cover* for G. Also, an *independent set* is a subset S of V whose elements are pairwise independent (in this chapter, the term *independent set* will only apply to a set of vertices), and a *matching* is a subset S of E whose elements are pairwise independent (in this chapter, the term *matching* will only apply to a set of edges). A subset M of $V \cup E$ whose elements are pairwise independent is said to be a *total matching* for G.

Suppose that $\mathcal{P}$ is some collection of sets. Denote by $\mathcal{P}^+$ the maximal elements of $\mathcal{P}$, i.e. $S \in \mathcal{P}^+$ if and only if $S \in \mathcal{P}$ and no proper superset of S is a member of $\mathcal{P}$. Similarly, denote by $\mathcal{P}^-$ the minimal elements of $\mathcal{P}$, i.e. $S \in \mathcal{P}^-$ if and only if $S \in \mathcal{P}$ and no proper subset of S is a member of $\mathcal{P}$. Let

$$\mathcal{C}(G) = \{C \subseteq V \cup E : C \text{ is a total cover for } G\}$$

and

$$\mathcal{M}(G) = \{M \subseteq V \cup E : M \text{ is a total matching for } G\}.$$

Nordhaus [172] and also Alavi et al. [2] define the following parameters[3]:

$$\alpha_2(G) = \min\{|C| : C \in \mathcal{C}^-(G)\}, \qquad \alpha_2^+(G) = \max\{|C| : C \in \mathcal{C}^-(G)\},$$
$$\beta_2^-(G) = \min\{|M| : M \in \mathcal{M}^+(G)\}, \quad \beta_2(G) = \max\{|M| : M \in \mathcal{M}^+(G)\}.$$

4.2.2 Survey of non-algorithmic total covering and total matching results

Some upper and lower bounds involving each of these parameters separately are derived by Gupta [102], Nordhaus [172], Alavi et al. [2], Meir [166], Kulli et al. [150], Zhang et al. [215], Alavi et al. [4] and Gimbel and Vestergaard [98]. In particular, it is known [2] that

$$\alpha_2(G) \le \beta_2^-(G) \le \beta_2(G) \le \alpha_2^+(G).$$

Peled and Sun [182] derive exact values for these parameters in threshold graphs. Also, Alavi et al. [4] consider properties of those connected graphs on n vertices having $\alpha_2(G) = \lceil \frac{n}{2} \rceil$. Bounds for $\alpha_2(G) + \beta_2(G)$ are considered by Alavi et al. [2], Erdös and Meir [72] and Meir [166]. In addition, some Nordhaus-Gaddum [173] type results have been obtained, involving each of α_2 and β_2 [72, 166], and involving β_2^- [98]. Finally, Topp and Vestergaard [208] characterise those graphs in which every maximal total matching is maximum, and Topp [206] studies those graphs having a unique maximum total matching. The survey by Hedetniemi et al. [117] describes the inequalities involving the total covering and total matching parameters in more detail.

The terminology for total covers and total matchings does not seem to be universally agreed upon in the literature. Nordhaus [172] and Alavi et al. [2], who introduced these concepts, define a subset C of $V \cup E$ to be a total cover if C covers G and C is minimal. Similarly, they define C to be a total matching if the elements of C are pairwise independent and C is maximal. However, several authors [4, 98, 208] have defined total covers and total matchings without the minimality or maximality requirement, respectively, as is done here. This can be advantageous, for example, when reasoning about a subset C of $V \cup E$ whose elements are pairwise independent, but C is not maximal. Following the terminology of Nordhaus [172], such a set is not a total matching. Referring to C as an *independent set* or a *matching* coincides with the usual notion of an independent set or matching when applied to sets containing vertices or edges only, respectively. Thus we choose to follow the terminology of [4, 98, 208]. We note in passing that total covers (as defined here) are referred to as *mixed dominating sets* by Hedetniemi et al. [117],

[3]The notation of the covering and independence parameters studied in this thesis follows that of Harary [106] and Alavi et al. [2]. The convention these authors follow is that the α and β symbols refer respectively to covering and independence properties that are to be satisfied. The subscript of the parameter symbol is 0, 1, 2 according to whether the elements of the set $\mathcal{U}(G)$ (as will be defined in the following sections, for each optimisation problem associated with the parameter concerned) are vertices, edges, or both, respectively. A superscript of '+' in the case of an α parameter refers to the 'maximum minimal' objective, and a superscript of '−' in the case of a β parameter refers to the 'minimum maximal' objective. When this superscript is missing from an α symbol, the objective is to minimise, and the objective is to maximise in the case that a β symbol is without superscript.

entire dominating sets by Kulli et al. [150] and *total dominating sets* by Gimbel et al. [97]. The latter definition is quite distinct from the more widely accepted notion of a total dominating set, due to Cockayne et al. [41] (see Section 4.3.4).

4.2.3 More covering and independence parameters

Nordhaus [172] shows how we may use $\mathcal{C}$ and $\mathcal{M}$ to derive some existing graph parameters. Define

$$\mathcal{C}_0(G) = \{C \in \mathcal{C}(G) : C \subseteq V\} \text{ and } \mathcal{C}_1(G) = \{C \in \mathcal{C}(G) : C \subseteq E\}$$

and similarly define

$$\mathcal{M}_0(G) = \{M \in \mathcal{M}(G) : M \subseteq V\} \text{ and } \mathcal{M}_1(G) = \{M \in \mathcal{M}(G) : M \subseteq E\}.$$

Then we obtain, as in [172],

$$\begin{array}{ll} \alpha_0(G) = \min\{|C| : C \in \mathcal{C}_0^-(G)\} - I_G, & \alpha_0^+(G) = \max\{|C| : C \in \mathcal{C}_0^-(G)\} - I_G, \\ \beta_0^-(G) = \min\{|M| : M \in \mathcal{M}_0^+(G)\}, & \beta_0(G) = \max\{|M| : M \in \mathcal{M}_0^+(G)\}, \end{array}$$

where α_0 and α_0^+ are the minimum and maximum over all minimal vertex covers of G respectively, and β_0^- and β_0 are the minimum and maximum over all maximal independent sets of G respectively, and I_G denotes the number of isolated vertices of G. Similarly we obtain[4]

$$\begin{array}{ll} \alpha_1(G) = \min\{|C| : C \in \mathcal{C}_1^-(G)\}, & \alpha_1^+(G) = \max\{|C| : C \in \mathcal{C}_1^-(G)\}, \\ \beta_1^-(G) = \min\{|M| : M \in \mathcal{M}_1^+(G)\}, & \beta_1(G) = \max\{|M| : M \in \mathcal{M}_1^+(G)\}, \end{array}$$

where α_1 and α_1^+ are the minimum and maximum over all minimal edge covers of G respectively, and β_1^- and β_1 are the minimum and maximum over all maximal matchings of G respectively. Thus definitions relating to the total covering and total matching parameters $\alpha_2, \alpha_2^+, \beta_2^-, \beta_2$ can be restricted, in order to obtain the eight covering and independence parameters $\alpha_i, \alpha_i^+, \beta_i^-, \beta_i$ for $i = 0, 1$. This implies a possible framework for twelve covering and independence parameters of graphs.

Nordhaus [172] investigates relations between the parameters $\alpha_2, \alpha_2^+, \beta_2^-, \beta_2$ and $\alpha_i, \alpha_i^+, \beta_i^-, \beta_i$ for $i = 0, 1$, and obtains the inequalities

$$\alpha_2(G) \leq \alpha_i(G) \leq \alpha_i^+(G) \leq \alpha_2^+(G)$$

for $i = 0, 1$, and

$$\beta_2(G) \geq \max\{\beta_0(G), \beta_1(G)\} \text{ and } \beta_2^-(G) \geq \max\{\beta_0^-(G), \beta_1^-(G)\}.$$

Let $\gamma(G)$ and $\Gamma(G)$ denote respectively the minimum and maximum over all minimal dominating sets of a graph G. For a graph $G = (V, E)$, let $T(G)$ denote the *total graph*

[4] In the case of α_1 and α_1^+, we assume that G has no isolated vertices, for the concept of edge covering is undefined for graphs with isolated vertices.

of G – this is the graph with vertex set $V \cup E$, and two vertices are adjacent in $T(G)$ if and only if the corresponding elements are adjacent or incident as vertices or edges of G. It is clear that $\alpha_2(G) = \gamma(T(G))$, $\alpha_2^+(G) = \Gamma(T(G))$, $\beta_2^-(G) = \beta_0^-(T(G))$ and $\beta_2(G) = \beta_0(T(G))$.

In the following sections, we consider the computational complexity of the graph optimisation problems related to each of the twelve covering and independence parameters introduced above. The total covering and total matching parameters are discussed in Sections 4.2.4 and 4.2.5 respectively, as their definition gives rise to the framework for the remaining parameters. Then, in Sections 4.2.6, 4.2.7 and 4.2.8, we consider the vertex covering and independence parameters, the edge covering parameters, and the matching parameters, respectively. Finally, we present some concluding remarks in Section 4.2.9.

4.2.4 Total covering

The source MINIMUM TOTAL COVER problem has components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, such that:

- $\mathcal{U}(G) = \mathbb{P}(V \cup E)$
- $\pi(G, S) \Leftrightarrow S$ is a total cover for G
- $\text{OPT} = \min$.

The maximinimal problem name is MAXIMUM MINIMAL TOTAL COVER.

Majumdar [163, p.52] shows that α_2 is NP-complete for general graphs, using a transformation from 3-SAT [92, problem LO2], and gives a linear-time algorithm for trees. Hedetniemi et al. [117] show that α_2 remains NP-complete for bipartite and chordal graphs.

Investigating the computational complexity of α_2^+ is given as an open problem by Hedetniemi et al. [117]. We show that MAXIMUM MINIMAL TOTAL COVER DECISION is NP-complete for planar graphs. The proof involves a transformation from a restricted version of the EXACT COVER BY 3-SETS (X3C) problem, defined in Section 3.4. The restriction of X3C known as PLANAR EXACT COVER BY 3-SETS (PX3C) demands that the graph $G = (V, E)$, associated with an instance (A, C) of X3C, with vertex set $V = A \cup C$ and edge set $E = \{(a, c) : a \in c \in C\}$, is planar. PX3C is NP-complete [69].

Theorem 4.2.1 MAXIMUM MINIMAL TOTAL COVER DECISION *is NP-complete, even for planar graphs.*

Proof. Clearly MAXIMUM MINIMAL TOTAL COVER DECISION is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$ and a set S of at least K elements, it is straightforward to verify in polynomial time that S is a minimal total cover of G.

To show NP-hardness, we give a transformation from PX3C, defined above. Given an arbitrary instance of PX3C, we construct a planar graph G, with the property that there exists an exact cover for the PX3C instance if and only if there exists a minimal total cover of G with at least K elements, for a particular $K \in \mathbb{Z}^+$.

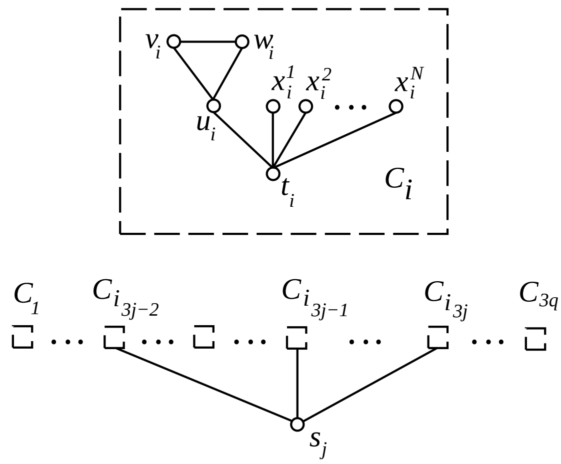

Figure 4.1: Part of the graph G constructed as an instance of MAXIMUM MINIMAL TOTAL COVER DECISION, showing typical subset and element components.

Suppose that a set of elements $A = \{a_1, a_2, a_3, \ldots, a_{3q}\}$ and a collection of clauses $C = \{c_1, c_2, c_3, \ldots, c_m\}$ (for some $q, m \in \mathbb{Z}^+$) is an arbitrary instance of PX3C. Suppose further that, for each j ($1 \leq j \leq m$), $c_j = \{a_{i_{3j-2}}, a_{i_{3j-1}}, a_{i_{3j}}\}$, where $i_1, i_2, i_3, \ldots, i_{3m}$ is some sequence of integers such that

$$\{i_1, i_2, i_3, \ldots, i_{3m}\} = \{1, 2, 3, \ldots, 3q\}.$$

Construct an instance – graph $G = (V, E)$ and positive integer K – of MAXIMUM MINIMAL TOTAL COVER DECISION as follows:

- *Subset vertices*: For each j ($1 \leq j \leq m$) create a *subset vertex* s_j.
- *Communication edges*: For each j ($1 \leq j \leq m$), add three *communication edges*, $\{s_j, t_{i_{3j-2}}\}, \{s_j, t_{i_{3j-1}}\}, \{s_j, t_{i_{3j}}\}$, where $c_j = \{a_{i_{3j-2}}, a_{i_{3j-1}}, a_{i_{3j}}\}$.
- *Element components*: For each i ($1 \leq i \leq 3q$), create an *element vertex* t_i. Form a clique among three vertices u_i, v_i, w_i, and join u_i to t_i. Create N (where N is to be defined) *leaf vertices* x_i^r, and join each x_i^r to t_i, for $1 \leq r \leq N$.
- *Target value*: Set $K = m + (3N + 8)q$.

Denote by S_i the following elements in the ith element component:

$$S_i = \{t_i, \{t_i, u_i\}, u_i, \{u_i, v_i\}, v_i, \{v_i, w_i\}, w_i, \{w_i, u_i\}\}.$$

Apart from the leaf vertices and their incident edges, G has a total of $12q + m$ vertices and $12q + 3m$ edges. Set N to be the sum of these totals, i.e. $N = 24q + 4m$. The construction is partly illustrated in Figure 4.1. Clearly, this construction is polynomial with respect to the size of the PX3C instance, and preserves the planarity of the graph constructed from this instance. First we show that if the PX3C instance has an exact cover, then G has a minimal total cover S, with $|S| = K$. From an exact cover C' for the PX3C instance, we construct a set S as follows. For each j $(1 \leq j \leq m)$:

- If $c_j \in C'$, add to S the three edges $\{s_j, t_{i_{3j-2}}\}$, $\{s_j, t_{i_{3j-1}}\}$ and $\{s_j, t_{i_{3j}}\}$.
- If $c_j \notin C'$, add to S the vertex s_j.

For each i $(1 \leq i \leq 3q)$:

- Add to S the vertices v_i, w_i.
- Add to S the vertices x_i^r for $1 \leq r \leq N$.

Now S is a total cover, for, clearly the leaf vertices cover themselves, their incident edges and t_i, for $1 \leq i \leq 3q$. Also s_j is covered either by itself or by an incident edge, for each j $(1 \leq j \leq m)$. As C' is an exact cover, then for each i $(1 \leq i \leq 3q)$, all edges incident on t_i are covered by some communication edge of S. Finally, all other vertices and edges in each element component are clearly covered.

S is minimal, for it is clear that each of the leaf vertices are are covered by no other element of S. Also $S \backslash \{v_i\}$ does not cover the edge $\{u_i, v_i\}$, and $S \backslash \{w_i\}$ does not cover the edge $\{u_i, w_i\}$, for any i $(1 \leq i \leq 3q)$. If $s_j \in S$ for any j $(1 \leq j \leq m)$, then no communication edge of S is incident on s_j, so that $S \backslash \{s_j\}$ does not cover s_j. Finally, if a communication edge $\{s_j, t_i\}$ is in S, for any i and j $(1 \leq i \leq 3q, 1 \leq j \leq m)$, then $S \backslash \{\{s_j, t_i\}\}$ does not cover $\{t_i, u_i\}$, since C' is an exact cover.

By construction of S, all $3q$ of the element vertices are covered by exactly one communication edge. As C' is an exact cover, these edges cover exactly q subset vertices. There are then $m - q = |C \backslash C'|$ subset vertices in S. Each element component contributes $N + 2$ vertices and no edges. Thus

$$\begin{aligned} |S| &= 3q + (m - q) + 3q(N + 2) \\ &= K \end{aligned}$$

as required.

Conversely, suppose that G has a minimal total cover S such that $|S| \geq K$. We show that the PX3C instance has an exact cover C'. From all minimal total covers for G with cardinality at least K, choose S to be such a set that has the fewest number of communication edges. We now establish a number of facts about the elements that S contains.

1. *S does not contain t_i, for any i $(1 \leq i \leq 3q)$.* For, suppose $t_i \in S$ for some i $(1 \leq i \leq 3q)$. Then by minimality $x_i^r \notin S$ for $1 \leq r \leq N$ and $\{t_i, x_i^r\} \notin S$ for $1 \leq r \leq N$. Thus, an upper bound for S in this case must be:

$$\begin{aligned} |S| &\leq N + (3q - 1)N \\ &< K \end{aligned}$$

which is a contradiction.

2. *There are $3qN$ elements in S such that each element is either a leaf vertex or is an edge incident on a leaf vertex. Furthermore, these elements cover each of the vertices t_i, for $1 \le i \le 3q$.* This observation follows from Fact 1.

3. *$|S \cap S_i| = 2$, for any i ($1 \le i \le 3q$).* For, let $1 \le i \le 3q$ be given. From Fact 1, $t_i \notin S$. Suppose $\{t_i, u_i\} \in S$. If $S \backslash \{\{t_i, u_i\}\}$ does not cover some edge $\{s_j, t_i\}$, for some j $(1 \le j \le m)$ then S does not cover s_j, since $t_k \notin S$, for any k $(1 \le k \le 3q)$, a contradiction. Thus $S \backslash \{\{t_i, u_i\}\}$ covers all communication edges of G, but does not cover some element of S_i. It follows that exactly one more element of S_i is in S. In the case that $\{t_i, u_i\} \notin S$, it may easily be verified that exactly two elements of S_i belong to S.

4. *S does not contain an edge $\{s_j, t_i\}$ together with vertex s_j, for any i and j ($1 \le i \le 3q$ and $1 \le j \le m$).* For, suppose S did. Since, by Fact 2, each of $t_{i_{3j-2}}, t_{i_{3j-1}}, t_{i_{3j}}$ is covered by a leaf vertex or an edge incident on a leaf vertex, then $S \backslash \{s_j\}$ also covers G, contradicting the minimality of S.

5. *S does not contain more than one communication edge incident on a vertex t_i, for any i ($1 \le i \le 3q$).* For, suppose S did – let $\{s_j, t_i\}$ and $\{s_k, t_i\}$ be two such edges, for some j, k $(1 \le j \ne k \le m)$ and i $(1 \le i \le 3q)$. Then by Fact 4, $s_k \notin S$, and by minimality, no edge incident on s_k other than $\{s_k, t_i\}$ is in S. Since, by Fact 2, each of $t_{i_{3k-2}}, t_{i_{3k-1}}, t_{i_{3k}}$ is already covered by a leaf vertex or an edge incident on a leaf vertex, then $S' = (S \backslash \{\{s_k, t_i\}\}) \cup \{s_k\}$ is a minimal total cover of G, with one fewer communication edge, and satisfies $|S'| = |S|$, contradicting the choice of S.

Let there be l communication edges in S. Then Fact 5 implies that these l edges are incident on exactly l of the element vertices, so that $l \le 3q$. Suppose that S contains r subset vertices. Now suppose that the l communication edges in S are incident on a total of s subset vertices. Then $3s \ge l$ and by Fact 4, these s subset vertices are all distinct from the r subset vertices defined above. Thus $r + s \le m$. But $r + s = m$, or else some s_j $(1 \le j \le m)$ is not covered, since $t_i \notin S$, for any i $(1 \le i \le 3q)$, by Fact 1. Finally, Facts 2 and 3 imply that S contains $N + 2$ elements from each of the $3q$ element components. Hence, having accounted for all the elements in S,

$$\begin{aligned} |S| &= r + l + 3q(N+2) \\ &= m + l - s + 3q(N+2) \qquad (\text{since } r + s = m). \end{aligned} \tag{4.1}$$

Assume firstly that $s < q$. Then by Equation 4.1,

$$\begin{aligned} |S| &\le m + 2s + 3q(N+2) & (\text{since } 3s \ge l) \\ &< K & (\text{since } s < q) \end{aligned}$$

which is a contradiction. Thus $s \geq q$. Now assume for a contradiction that $l < 3q$. Then by Equation 4.1,

$$\begin{array}{rcll} |S| & < & m + 3q - s + 3q(N+2) & (\text{since } l < 3q) \\ & \leq & K & (\text{since } s \geq q) \end{array}$$

which is also a contradiction. Hence $l = 3q$. Finally, assume for a contradiction that $s > q$. Then by Equation 4.1,

$$\begin{array}{rcll} |S| & = & m + 3q - s + 3q(N+2) & (\text{since } l = 3q) \\ & < & K & (\text{since } s > q) \end{array}$$

which gives a contradiction. Hence $s = q$ and $r = m - q$, so that exactly q of the subset vertices are covered by communication edges. Also, each of the $3q$ element vertices is covered by exactly one edge. Thus, for exactly q of the the subset vertices s_j $(1 \leq j \leq m)$, we have $\{s_j, t_{i_{3j-2+r}}\} \in S$, for $0 \leq r \leq 2$; let C' contain the q corresponding c_j triples. Since the $m - q$ other subset vertices cover themselves, then C' is an exact cover. ■

As pointed out in Section 4.2.3, $\alpha_2^+(G) = \Gamma(T(G))$ for a graph G. Yannakakis and Gavril [213] show that a connected graph is a tree if and only if its total graph is chordal. Jacobson and Peters [131] show that $\Gamma = \beta_0$ for chordal graphs. Hence, as β_0 is polynomial-time solvable for this class of graphs [95], the same is true for Γ, so that α_2^+ is polynomial-time solvable for trees. In addition, the remarks of this paragraph also imply that $\alpha_2^+ = \beta_2$ for trees.

4.2.5 Total matching

The components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ of the source MAXIMUM TOTAL MATCHING problem are defined as follows:

- $\mathcal{U}(G) = \mathbb{P}(V \cup E)$
- $\pi(G, S) \Leftrightarrow S$ is a total matching in G
- $\text{OPT} = \max$.

The minimaximal problem name is MINIMUM MAXIMAL TOTAL MATCHING.

The total matching parameter β_2 is related to β_1^-: Gupta [102] shows that $\beta_2(G) + \beta_1^-(G) = n$ for any graph $G = (V, E)$, where $n = |V|$. Therefore we have the following result, which does not seem to have been explicitly noted in the literature previously.

Theorem 4.2.2 *The complexities of β_2 and β_1^- are identical, for any graph class.*

It is interesting to consider how we may construct a maximum total matching from a minimum maximal matching, and vice versa. Since Gupta's result is stated without proof, we provide, for completeness, one possible method. We use the following result, whose proof is straightforward, and is omitted.

Proposition 4.2.3 *Let $G = (V, E)$ be a graph and let $M \subseteq V \cup E$ be a total matching. Then M is a maximal total matching if and only if M is a total cover.*

Proposition 4.2.4 *Let $G = (V, E)$ be a graph, where $n = |V|$. Then if $M \subseteq E$ is a maximal matching for G, where $m = |M|$, we may find a maximal total matching $M' \subseteq V \cup E$ for G, where $|M'| = n - m$, in polynomial time. Conversely, if $M \subseteq V \cup E$ is a maximum total matching for G, where $m = |M|$, we may find a maximal matching $M' \subseteq E$ for G, where $|M'| = n - m$, in polynomial time.*

Proof: Suppose that $M \subseteq E$ is a maximal matching for G, where $m = |M|$. Then M covers $2m$ vertices of V, so that there is a set V' of vertices not covered by M, where $|V'| = n - 2m$. Set $M' = M \cup V'$. Then M' is a total matching, since by maximality of M, no pair of vertices in V' are adjacent in G. Also M' is maximal by Proposition 4.2.3, since M' is a total cover of G. Finally $|M'| = m + (n - 2m) = n - m$.

Conversely, suppose that $M \subseteq V \cup E$ is a maximum total matching for G, so that $|M| = \beta_2(G)$. We may construct a set $M'' \subseteq V \cup E$, where $|M''| = |M|$, such that M'' is a total matching for G and for every edge $\{u, v\}$ of E, some edge of M'' is incident on u, or incident on v, or $\{u, v\} \in M''$. For, suppose there is an edge $\{u, v\}$ such that no edge of M is incident on u or v. Then as M is maximal, M covers the edge $\{u, v\}$, by Proposition 4.2.3. Thus, without loss of generality $u \in M$. Hence we may replace u with $\{u, v\}$ in M. Repeating this procedure with every such edge gives rise to M'', which is clearly a total matching, and must be maximal, since $|M''| = \beta_2(G)$. Now let $M' = M'' \cap E$. Then $M' \subseteq E$ is a matching and is maximal, since no two vertices that are not covered by M' are adjacent in G, by construction of M''. Let $|M'| = n - m$, for some $m > 0$. Then M' covers $2n - 2m$ vertices of G. Thus there are $2m - n$ elements (all vertices) in $M'' \backslash M'$, since M'' is a total cover of G. Thus $|M| = |M''| = (n - m) + (2m - n) = m$. ∎

Corollary 4.2.5 *There is a polynomial time algorithm to transform a minimum maximal matching into a maximum total matching, and vice versa.*

In order to resolve the complexity of β_2^-, we make the following definition. Given an arbitrary graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, construct the *pendant graph* $G' = (V', E')$ of G by adding two new vertices u_i and w_i to V, for each i $(1 \leq i \leq n)$, and two new edges $\{u_i, v_i\}$ and $\{w_i, v_i\}$ to E, for each i $(1 \leq i \leq n)$.

Theorem 4.2.6 (Gimbel and Vestergaard [98]) *Given a graph $G = (V, E)$, where $n = |V|$,*

$$\beta_2^-(G') = 2n - \beta_0(G)$$

where $G' = (V', E')$ is the pendant graph of G.

By Theorem 4.2.6 and the complexity of β_0 (discussed in Section 4.2.6), we deduce that β_2^- is NP-complete for an arbitrary graph. In fact, as β_0 remains NP-complete for planar cubic graphs (see Section 4.2.6), we may deduce that β_2^- remains NP-complete for planar graphs of maximum degree 5.

We also note that it is possible to use the transformation of Hedetniemi et al. [117], showing NP-completeness for α_2 in bipartite and chordal graphs, in order to obtain NP-completeness for β_2^- in the same two classes of graphs.

Theorem 4.2.7 MINIMUM MAXIMAL TOTAL MATCHING DECISION *is NP-complete for bipartite and chordal graphs.*

Proof: Clearly, the problem is in NP for both graph classes. To show NP-hardness, we focus on the transformation of Hedetniemi et al. [117], showing NP-completeness for α_2 in bipartite or chordal graphs. The reduction begins from the NP-complete problem X3C, defined in Section 3.4. A bipartite/chordal graph G is constructed, and an integer K is defined, with the property that the X3C instance has an exact cover if and only if G has a total cover of size at most K.

Corresponding to an exact cover for the X3C instance, the total cover constructed by Hedetniemi et al. [117] is in fact a total matching, and hence a maximal total matching by Proposition 4.2.3. Conversely, if G has a maximal total matching M of size at most K, then M is a total cover for G by Proposition 4.2.3, and the corresponding argument of Hedetniemi et al. [117] shows that the X3C instance has an exact cover.

Thus the same reduction may be used to prove NP-completeness for β_2^- in bipartite or chordal graphs. ■

As pointed out in Section 4.2.3, $\beta_2^-(G) = \beta_0^-(T(G))$ for a graph G. Majumdar [163, p.26] shows that a connected graph is a tree if and only if its total graph is strongly chordal[5]. Farber [74] shows that β_0^- is polynomial-time solvable for strongly chordal graphs. Hence β_2^- is polynomial-time solvable for trees.

4.2.6 Vertex covering and vertex independence

We firstly define the MINIMUM VERTEX COVER and MAXIMUM INDEPENDENT SET problems (whose decision versions are problems GT1 and GT20 respectively in [92]).

Source problem: MINIMUM VERTEX COVER=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is a vertex cover for E
- $\text{OPT} = \min$.

Maximinimal problem name: MAXIMUM MINIMAL VERTEX COVER.

Source problem: MAXIMUM INDEPENDENT SET=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is an independent set in G

[5] A graph G is *strongly chordal* if G is chordal and every cycle of length at least six has an 'odd' chord, i.e., a chord joining two vertices that are separated by an odd number of edges in the cycle.

- OPT = max.

Minimaximal problem name: MINIMUM MAXIMAL INDEPENDENT SET.

The complexities of α_0 and β_0 for any class of graphs are identical, as is indicated by the following proposition, whose proof is trivial.

Proposition 4.2.8 *Given a graph $G = (V, E)$ and a set $V' \subseteq V$, V' is a vertex cover for G if and only if $V \backslash V'$ is an independent set for G.* ∎

From Proposition 4.2.8, we deduce the classical result of Gallai [87], namely that for a graph G with n vertices, $\alpha_0(G) + \beta_0(G) = n$. The parameter β_0 is NP-complete, even for planar cubic graphs. This fact may be deduced from separate results due to Garey et al. [93], Garey and Johnson [89], and Maier and Storer [162]. On the other hand, β_0 is polynomial-time solvable for bipartite graphs (by matching – see Harary [106], for example), chordal graphs [95] and trees [62]. Many other classes of graphs for which β_0 remains NP-complete and for which β_0 is polynomial-time solvable are discussed in [92, problem GT20] and [134].

Similarly the complexities of α_0^+ and β_0^- are identical, as the following result shows. Again the proof is simple, and is omitted.

Lemma 4.2.9 *Given a graph $G = (V, E)$ and a set $V' \subseteq V$, V' is a minimal vertex cover for G if and only if $V \backslash V'$ is a maximal independent set for G.* ∎

From Lemma 4.2.9 we may deduce another Gallai type identity, that for a graph G with n vertices, $\alpha_0^+(G) + \beta_0^-(G) = n$, as observed by McFall and Nowakowski [164]. In fact the complexities of α_0^+ and β_0^- are related to that of i, the minimum *independent dominating set* parameter. A set of vertices S is an independent dominating set for a graph G if S is both an independent set and a dominating set for G. Independent dominating sets are related to maximal independent sets, as the following lemma demonstrates.

Lemma 4.2.10 (Berge [17, Thm.2, p.309]) *Given a graph $G = (V, E)$ and a subset V' of V, V' is a maximal independent set if and only if V' is an independent dominating set.*

Thus Lemma 4.2.10 implies that $i(G) = \beta_0^-(G)$ for any graph G. Lemmas 4.2.9 and 4.2.10 together give the following result.

Theorem 4.2.11 *α_0^+, β_0^-, i each have the same complexity, over every graph class.*

The parameter i is NP-complete for bipartite graphs [57, 129] and dually chordal graphs [25], though polynomial-time algorithms have been constructed for chordal graphs [73], interval and circular-arc graphs [34], permutation graphs [76, 27], cocomparability graphs [147], asteroidal triple-free graphs [28], k-polygon graphs (for fixed k) [71], series-parallel graphs [184, 101], partial k-trees (for fixed k) [204] and trees [20].

The complexity of i for planar graphs does not seem to be mentioned explicitly in the literature. However, the transformation of Corneil and Perl [57], showing NP-completeness

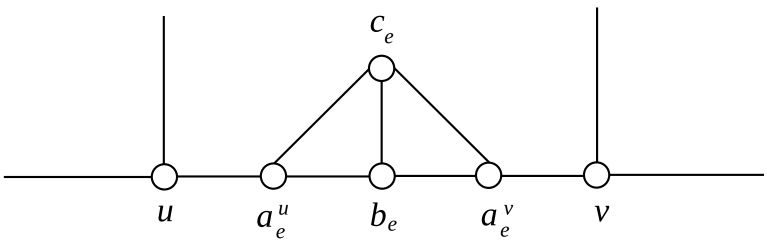

Figure 4.2: A typical edge component from the constructed instance of MINIMUM INDEPENDENT DOMINATING SET DECISION.

for i in bipartite graphs, begins from MINIMUM DOMINATING SET DECISION (which is the decision problem associated with γ, taking a graph G and integer $K \in \mathbb{Z}^+$ as input, and asking whether $\gamma(G) \leq K$) in general graphs and preserves planarity. By transforming from the NP-complete restriction of MINIMUM DOMINATING SET DECISION to planar cubic graphs [144], we obtain NP-completeness for i in planar bipartite graphs, where all vertices in one part have degree at most 3, and all vertices in the other part have degree at most 2. An alteration to the transformation of Corneil and Perl gives NP-completeness for i in planar cubic graphs. To aid exposition, we present the proof in its entirety. (In what follows, we refer to the MINIMUM INDEPENDENT DOMINATING SET DECISION problem, which takes a graph G and integer $K \in \mathbb{Z}^+$ as input and asks whether $i(G) \leq K$.)

Theorem 4.2.12 MINIMUM INDEPENDENT DOMINATING SET DECISION *is NP-complete, even for planar cubic graphs.*

Proof: Clearly MINIMUM INDEPENDENT DOMINATING SET DECISION is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$ and a set S of at most K vertices, it is straightforward to verify in polynomial time that S is an independent dominating set of G.

To show NP-hardness, we give a transformation from the NP-complete MINIMUM DOMINATING SET DECISION problem for planar cubic graphs, as discussed above. Hence let $G = (V, E)$ (a planar cubic graph) and K (a positive integer) be an instance of MINIMUM DOMINATING SET DECISION. Assume that $|E| = m$. We construct an instance $G' = (V', E')$ (planar cubic graph) and K' (positive integer) of MINIMUM INDEPENDENT DOMINATING SET DECISION. Corresponding to every edge $e = \{u, v\}$ of E, construct an *edge component* of G' as follows: replace the edge e by a path on five vertices, namely u, a_e^u, b_e, a_e^v, v, connected in that order. Create an additional vertex, c_e, adjacent to a_e^u, b_e, a_e^v. It may be verified that G' is planar and cubic. An example edge component is shown in Figure 4.2. Denote by V_e the vertices in the edge component corresponding to edge e, i.e.

$$V_e = \{u, a_e^u, a_e^v, b_e, c_e, v\}.$$

Denote by X_e the internal vertices in this edge component, i.e.

$$X_e = \{a_e^u, a_e^v, b_e, c_e\}.$$

Set $K' = K + m$. We claim that G has a dominating set of cardinality at most K if and only if G' has an independent dominating set of cardinality at most K'.

For, suppose that D is a dominating set for G, where $|D| = k \leq K$. We construct an independent dominating set D' for G'. Initially, let D' contain the vertices of D. For any edge $e = \{u, v\}$ of G, we add vertices to D', according to four cases:

1. $u \notin D, v \notin D$. Add the vertex b_e to D'.

2. $u \in D, v \notin D$. Add the vertex a_e^v to D'.

3. $u \notin D, v \in D$. Add the vertex a_e^u to D'.

4. $u \in D, v \in D$. Add the vertex b_e to D'.

It may be verified that D' is an independent dominating set for G', and $|D'| = k+m \leq K'$.

Conversely, suppose that D' is an independent dominating set for G' of size at most K'. We construct a set D'' as follows. Initially let $D'' = D'$. For any edge $e = \{u, v\}$ of G, consider the elements of $Q_e = V_e \cap D'$. By domination, $|Q_e| \geq 1$, and if $|Q_e| = 1$, then $Q_e = \{b_e\}$ or $Q_e = \{c_e\}$. By independence, $|Q_e| \leq 3$, and if $|Q_e| = 3$, then $Q_e = \{u, b_e, v\}$ or $Q_e = \{u, c_e, v\}$. If $|Q_e| = 2$, then either $|Q_e \cap X_e| = 1$, or $Q_e = \{a_e^u, a_e^v\}$. In the latter case, replace a_e^v by v in D''.

It may be verified that D'' is a dominating set for G', and $|D''| \leq |D'|$. Now let $D = D'' \cap V$. We claim that D is a dominating set for G. For, suppose that $u \in V \backslash D$. Then $u \notin D''$, so by the domination property of D'', there is some $e = \{u, v\} \in E$ such that $a_e^u \in D''$. By construction of D'', $|D'' \cap X_e| = 1$. Hence, $a_e^v \notin D''$, but as a_e^v must be dominated by D'', the only outcome is $v \in D''$. Hence $v \in D$ as required. Finally, $|D| = |D''| - m \leq |D'| - m \leq K' - m = K$. ■

4.2.7 Edge covering

In this section, we consider only graphs with no isolated vertices, since the concept of edge covering is undefined for graphs with isolated vertices.

The source MINIMUM EDGE COVER problem has components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where:

- $\mathcal{U}(G) = \mathbb{P}(E)$
- $\pi(G, E') \Leftrightarrow E'$ is an edge cover for V
- $\text{OPT} = \min$.

The maximinimal problem name is MAXIMUM MINIMAL EDGE COVER.

Norman and Rabin [174] demonstrate that there is a polynomial time algorithm to transform a maximum matching to a minimum edge cover, and vice versa. Hence the complexity of α_1 is identical to that of β_1. The proof of this result also demonstrates that a further Gallai type identity holds, i.e. for a graph G with n vertices, $\alpha_1(G) + \beta_1(G) = n$.

The parameter $\alpha_1^+(G)$, the cardinality of a maximum minimal edge cover, seems to have received relatively little attention in the literature. However, the parameter is considered by Hedetniemi [119], who shows that $\alpha_1^+(G) = \varepsilon(G)$ for a non-trivial connected graph G, where $\varepsilon(G)$ denotes the maximum number of *pendant edges* among all spanning forests for G. (Given a spanning forest F for G, $\{u, v\} \in F$ is a pendant edge for F if the degree of u or v in F is one.) Nieminen [171] shows that, for a non-trivial connected graph G with n vertices,

$$\gamma(G) + \varepsilon(G) = n \tag{4.2}$$

and hence $\gamma(G) + \alpha_1^+(G) = n$. It is clear that these results extend to arbitrary graphs with no isolated vertices. Hence we obtain the following theorem.

Theorem 4.2.13 *For graphs with no isolated vertices, the complexity of α_1^+ is identical to that of γ.*

The complexity of γ for several graph classes is surveyed in Section 4.3.3.

It is also of interest to consider how we may construct a maximum minimal edge cover from a minimum dominating set, and vice versa. For a given graph G and a spanning forest F of G, let $e(F)$ denote the number of pendant edges of F. A spanning forest F of G such that $e(F) = \varepsilon(G)$ is called a *maximum spanning forest* of G. Nieminen's proof of Equation 4.2 involves constructing in polynomial time a maximum spanning forest $F(D)$ from a minimum dominating set D, where $e(F(D)) = |V| - |D|$. Hedetniemi's proof of $\alpha_1^+(G) = \varepsilon(G)$ involves constructing in polynomial time a maximum minimal edge cover from a maximum spanning forest. Together, these two constructions give a polynomial-time procedure for transforming a minimum dominating set into a maximum minimal edge cover. For the converse, we make the following observation about minimal edge covers (the proof is straightforward, and is omitted):

Proposition 4.2.14 *Let G be a graph with no isolated vertices and let $S \subseteq V \cup E$. Then S is a minimal edge cover if and only if S is a spanning forest for G that satisfies the following two properties:*

1. *$S \subseteq E$.*
2. *Every edge of S is a pendant edge.*

Thus a minimal edge cover of G is a spanning forest S such that each connected component of S is a non-trivial star (i.e. is a $K_{1,r}$ for some $r \geq 1$).

Given a graph $G = (V, E)$ with no isolated vertices, and a maximum minimal edge cover S of G, we construct a set of vertices $P \subseteq V$ as follows. For each edge $e \in S$, we know that e is a pendant edge, so that at least one endpoint vertex u of e has degree one in S; add u to P. Thus P contains exactly one vertex corresponding to every edge of S, so that $|P| = |S|$. Let $D = V \backslash P$. Then $|D| = \gamma(G)$, and it may be verified that D is a dominating set for G, by Proposition 4.2.14. Thus, we have the following result.

Theorem 4.2.15 *There is a polynomial time algorithm to construct a maximum minimal edge cover from a minimum dominating set and vice versa, for arbitrary graphs with no isolated vertices.*

4.2.8 Matching

The source MAXIMUM MATCHING problem has components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, such that:

- $\mathcal{U}(G) = \mathbb{P}(E)$
- $\pi(G, E') \Leftrightarrow E'$ is a matching in G
- $\text{OPT} = \max$.

The minimaximal problem name is MINIMUM MAXIMAL MATCHING (whose decision version is problem GT10 of [92]).

Computation of $\beta_1(G)$ is the well-known problem of finding a maximum matching of a graph. The famous algorithm due to Edmonds [70] is described in detail by Lovász and Plummer [157], for example. The parameter β_1^-, the cardinality of a minimum maximal matching, is in fact equal to γ', the cardinality of a minimum edge dominating set, as we now show. Two propositions follow, the proof of the first of which is trivial. Both propositions involve the concept of an *independent edge dominating set*, which is a set of edges that is both a matching and an edge dominating set.

Proposition 4.2.16 *Given a graph $G = (V, E)$ and a set $E' \subseteq E$, E' is a maximal matching for G if and only if E' is an independent edge dominating set for G.*

Proposition 4.2.17 (Yannakakis and Gavril [213]) *Given a graph $G = (V, E)$ and an edge dominating set E' for G, we may construct, in polynomial time, an independent edge dominating set E'' for G, with $|E''| \leq |E'|$.*

Proof: Suppose that E' is edge dominating with adjacent edges $\{u, v\}$ and $\{v, w\}$. If $E' \backslash \{\{v, w\}\}$ is edge dominating then we may delete $\{v, w\}$ from E'. Otherwise there is some x such that $\{w, x\}$ is dominated only by $\{v, w\}$, so we may replace $\{v, w\}$ by $\{w, x\}$ in E'. In either case, the resultant set is edge dominating with one fewer pair of adjacent edges. We may continue this process until we obtain an independent edge dominating set E'', which is a maximal matching by Proposition 4.2.16, and it is clear that $|E''| \leq |E'|$. ■

From Propositions 4.2.16 and 4.2.17, it follows that $\beta_1^-(G) = \gamma'(G)$ for any graph G, which implies that the complexities of β_1^- and γ' are identical. The complexity of γ' for several graph classes is surveyed in Section 4.3.5.

Propositions 4.2.16 and 4.2.17 also indicate how we may construct a minimum maximal matching from a minimum edge dominating set in polynomial time. The converse is trivial, since any minimum maximal matching is, of course, a minimum edge dominating set.

4.2.9 Conclusion and further study relating to the twelve covering and independence parameters

Relatively speaking, the parameters $\alpha_2, \alpha_2^+, \beta_2, \beta_2^-$ have not been extensively studied, despite their very natural definitions. In particular, there is scope for investigating whether Gallai type identities hold [87]. A survey of such results involving the parameters α_i, α_i^+,

β_i, β_i^- for $i = 0, 1$ appears in [48]. As mentioned in Section 4.2.2, bounds for $\alpha_2(G)+\beta_2(G)$ have been investigated [2, 72, 166], and the identity $\beta_2(G) + \beta_1^-(G) = n$ holds [102], but it is open as to whether bounds exist involving $\alpha_2^+(G) + \beta_2^-(G)$ that improve on those obtained by simply considering the sum of known upper and lower bounds for α_2^+ and β_2^- separately.

Similarly, the existence of Nordhaus-Gaddum [173] type inequalities are of interest. Such results have been obtained for the parameters β_0 and β_1 [36], β_0^- [51, 42, 45, 113], γ and γ' (see [108] for a survey). As reported in Section 4.2.2, Nordhaus-Gaddum inequalities involving α_2, β_2 and β_2^- have been obtained [72, 166, 98], but there is still scope for investigating such bounds involving the other parameters treated in this paper.

Regarding the algorithmic complexity of these parameters, one perhaps significant open problem is the complexity of β_1^- for chordal graphs – that this problem is open is noted by Horton and Kilakos [126].

The NP-completeness results for the parameters considered here imply that their properties of approximability are of interest. Results have been obtained for the parameters $\alpha_0, \beta_0, \beta_0^-$ and are surveyed in [59]. Regarding the approximability of β_1^-, any maximal matching is a 2-approximation to β_1^- [145]. Proposition 4.2.17 implies that we may construct, in polynomial time, a maximal matching E'' from an edge dominating set E', such that $|E''| \leq |E'|$. Thus, since $\beta_1^- = \gamma'$, and MINIMUM EDGE DOMINATING SET admits a ptas for planar graphs [12], then MINIMUM MAXIMAL MATCHING also admits a ptas for planar graphs. Also, β_1^- is APX-complete, even for graphs of maximum degree 3 [216]. However it appears that the approximability of the parameters $\alpha_0^+, \alpha_1^+, \alpha_2, \alpha_2^+, \beta_2^-, \beta_2$ is open.

4.3 Minimaximal and maximinimal strong stability, clique, domination and irredundance graph problems

In this section we investigate twelve parameters concerned with strong stability, cliques, domination, total domination, edge domination, and irredundance, in Sections 4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.3.5 and 4.3.6, respectively.

4.3.1 Strong stability

The components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ of the source MAXIMUM STRONG STABLE SET problem are:

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is a strong stable set for G, i.e., $\forall\, v \in V \bullet |N[v] \cap V'| \leq 1$.
- $\text{OPT} = \max$.

The minimaximal problem name is MINIMUM MAXIMAL STRONG STABLE SET.

Following the notation of Domke et al. [63], let $\beta_{SS}^-(G)$ and $\beta_{SS}(G)$ denote respectively the minimum and maximum over all maximal strong stable sets of a given graph G.

The concept of a strong stable set was first defined by Hochbaum and Shmoys [124]. Meir and Moon [167] define a set of vertices S of a graph $G = (V, E)$ to be a *2-packing* if $d(u, v) > 2$ for each pair of distinct vertices $u, v \in S$. The following proposition, whose proof is straightforward, demonstrates the equivalence of these two concepts.

Proposition 4.3.1 *Let $G = (V, E)$ be a graph and let $S \subseteq V$. Then S is a strong stable set if and only if S is a 2-packing.*

Let $\rho(G)$ denote the maximum cardinality of a 2-packing of G (also denoted $P_2(G)$ by Meir and Moon [167]); this notation follows Haynes et al. [115, p.348]. It follows by Proposition 4.3.1 that $\beta_{SS}(G) = \rho(G)$.

In general, the parameter β_{SS} is NP-complete [124]. Also, Chang and Nemhauser [33] show that β_{SS} is NP-complete for bipartite and chordal graphs. The NP-completeness result for bipartite graphs involves a transformation from the MAXIMUM INDEPENDENT SET DECISION problem for general graphs, and preserves planarity. By considering the NP-complete restriction of β_0 to planar cubic graphs (see Section 4.2.6), we may use the same transformation to obtain NP-completeness for β_{SS} in planar bipartite graphs of maximum degree 3, as observed by Horton and Kilakos [126]. However, for a tree T, Meir and Moon [167] show that $\rho(T) = \gamma(T)$, and hence β_{SS} is polynomial-time solvable for trees (see Section 4.3.3). Also β_{SS} is polynomial-time solvable for dually chordal graphs [25].

The minimum maximal strong stability number, β_{SS}^- has been studied by McRae [165], who shows that β_{SS}^- is NP-complete for bipartite and chordal graphs. We now resolve the complexity of this parameter in planar graphs.

Theorem 4.3.2 MINIMUM MAXIMAL STRONG STABLE SET DECISION *is NP-complete, even for planar graphs of maximum degree 3.*

Proof: Clearly MINIMUM MAXIMAL STRONG STABLE SET DECISION is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$ and a set S of at most K vertices, it is straightforward to verify in polynomial time that S is a maximal strong stable set of G.

To show NP-hardness, we give a transformation from the MINIMUM VERTEX COVER DECISION problem for planar cubic graphs. That this problem remains NP-complete for planar cubic graphs is discussed in Section 4.2.6. Hence let $G = (V, E)$ (a planar cubic graph) and K (a positive integer) be an instance of MINIMUM VERTEX COVER DECISION. Assume that $|E| = m$. We construct an instance $G' = (V', E')$ (planar graph of maximum degree 3) and K' (positive integer) of MINIMUM MAXIMAL STRONG STABLE SET DECISION. Corresponding to every edge $e = \{u, v\} \in E$, construct an *edge component* of G', comprising nine vertices, of which two are u, v, and the other seven are new, as follows:

- Vertices $u, v, a_e^u, a_e^v, b_e^u, b_e^v, c_e^u, c_e^v, d_e$.
- Edges

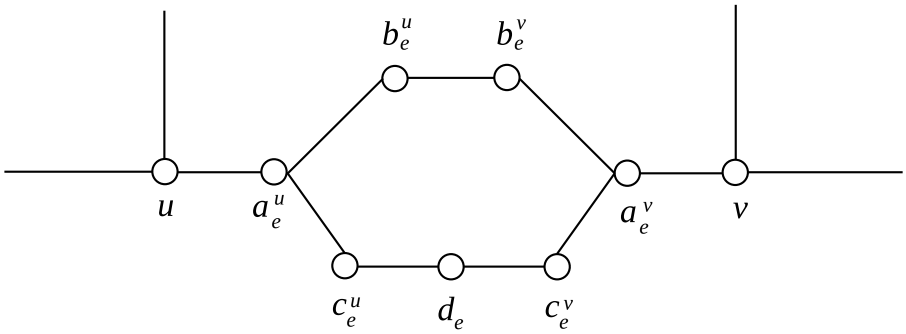

Figure 4.3: A typical edge component from the constructed instance of MINIMUM MAXIMAL STRONG STABLE SET DECISION.

- $\{u, a_e^u\}, \{a_e^v, v\}$,
- $\{a_e^u, b_e^u\}, \{b_e^u, b_e^v\}, \{b_e^v, a_e^v\}$,
- $\{a_e^u, c_e^u\}, \{c_e^u, d_e\}, \{d_e, c_e^v\}, \{c_e^v, a_e^v\}$.

It is clear that the graph G' constructed is planar of maximum degree 3. A typical edge component of G' is illustrated in Figure 4.3. Denote by V_e the internal vertices in the edge component corresponding to the edge $e = \{u, v\}$ of G, i.e.

$$V_e = \{a_e^u, a_e^v, b_e^u, b_e^v, c_e^u, c_e^v, d_e\}.$$

Set $K' = K + m$. We now show that G has a vertex cover of cardinality at most K if and only if G' has a maximal strong stable set with cardinality at most K'.

For, suppose that C is a vertex cover for G, where $|C| \leq K$. We construct a set S as follows. Initially let $S = C$. For each edge $e = \{u, v\}$, at least one of u, v is in C. If $u \in C$ and $v \notin C$, then we add c_e^v to S. Similarly, if $u \notin C$ and $v \in C$, then we add c_e^u to S. In the case that $u \in C$ and $v \in C$, then we add d_e to S.

It is straightforward to verify that S is a strong stable set. In addition, S is maximal. For, let $e = \{u, v\}$ be given. It is clear that, in each of the three cases $u \in C, v \notin C$ and $u \notin C, v \in C$ and $u \in C, v \in C$, each member of $V_e \backslash S$ is at distance at most two from some member of S. Now suppose that $u \in V \backslash S$. Pick any $v \in V$ such that $e = \{u, v\} \in E$. Then $u \in V \backslash C$ implies that $v \in C$. By construction of S, we have $c_e^u \in S$. As u is at distance two from c_e^u, then S is maximal. Moreover, $|S| = |C| + m \leq K + m = K'$, as required.

Conversely, suppose that S is a maximal strong stable set for G', where $|S| \leq K'$. Let $e = \{u, v\} \in E$ be given. It is straightforward to observe that in each of the three cases $u \in S, v \notin S$ and $u \notin S, v \in S$ and $u \in S, v \in S$, we have $|S \cap V_e| \geq 1$. Moreover, it may verified that in the case $u \notin S$ and $v \notin S$, we have $|S \cap V_e| \geq 2$. Define

$$X = \{e \in E : e = \{u, v\} \wedge u \notin S \wedge v \notin S\}.$$

Let $x = |X|$. Then, apart from the vertices in $S \cap V$, S contains at least two vertices from x edge components of G' and at least one vertex from $(m - x)$ edge components of G'. Thus

$$\begin{aligned} |S| &\geq |S \cap V| + 2x + (m - x) \\ &= |S \cap V| + m + x \end{aligned} \tag{4.3}$$

Define a set S' as follows. Initially let $S' = S$. For each $e = \{u, v\} \in X$, there is some vertex $w \in V_e \cap S$. Replace w by u in S'. Now let $C = S' \cap V$. Clearly, C is a vertex cover for G. Finally, $|C| \leq |S \cap V| + x$, so that, by Inequality 4.3, $|C| \leq |S| - m \leq K' - m = K$ as required. ■

4.3.2 Clique

The source MAXIMUM CLIQUE problem (whose decision version is problem GT19 of [92]) has components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, such that:

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is a clique in G, i.e., $\forall\, v, w \in V' \bullet \{v, w\} \in E$
- $\text{OPT} = \max$.

The minimaximal problem name is MINIMUM MAXIMAL CLIQUE.

Let $\omega^-(G)$ and $\omega(G)$ denote respectively the minimum and maximum over all maximal cliques of a given graph G.

The complexity of the maximum clique number, ω, is as for β_0 on the complementary graph, as is shown by the following proposition, whose proof is trivial.

Proposition 4.3.3 *Given a graph $G = (V, E)$ and a set $V' \subseteq V$, V' is a clique for G if and only if V' is an independent set for G^C.*

Therefore, as β_0 is NP-complete for planar graphs, ω is NP-complete, even for the complements of planar graphs. However, ω is polynomial-time solvable for chordal graphs [95]. Determining the value of ω in bipartite or planar graphs is trivial, since $\omega(G) \leq 2$ for a bipartite graph G, and $\omega(G) \leq 4$ for a planar graph G.

Maximal cliques in a graph correspond to maximal independent sets in the complementary graph, as is shown by the following result, whose proof is also simple, and is omitted.

Lemma 4.3.4 *Given a graph $G = (V, E)$ and a set $V' \subseteq V$, V' is a maximal clique for G if and only if V' is a maximal independent set for G^C.*

Lemma 4.3.4 leads on to the following result, which relates the complexity of the minimum maximal clique number, ω^-, to a parameter that we have already defined.

Theorem 4.3.5 *The complexity of ω^- is identical to the complexity of β_0^- on the complementary graph.*

Therefore, as β_0^- is NP-complete for planar or bipartite graphs, ω^- is NP-complete, even for the complements of planar or bipartite graphs. As is the case for ω, determining the value of ω^- in bipartite or planar graphs is trivial.

4.3.3 Domination

The source MINIMUM DOMINATING SET problem (whose decision version is GT2 of [92]) has components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, such that:

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is a dominating set for V, i.e. $N[V'] = V$
- $\text{OPT} = \min$.

The maximinimal problem name is MAXIMUM MINIMAL DOMINATING SET or UPPER DOMINATION.

Let $\gamma(G)$ and $\Gamma(G)$ denote respectively the minimum and maximum over all minimal dominating sets of a given graph G.

The domination number, γ, remains NP-complete for planar cubic graphs [90, 144], bipartite graphs [19] and undirected path graphs (a subclass of chordal graphs) [23], though γ is polynomial-time solvable for strongly chordal graphs [74] and trees [46]. Polynomial-time algorithms and NP-completeness results for γ have been obtained for many other classes of graphs. Chapter 8 of [114] and Chapter 12 of [115] contain two recent algorithmic surveys of γ in various graph classes. See also [92, problem GT2] and [133, 134, 58].

The maximum minimal domination number, Γ, is NP-complete for arbitrary graphs [39]. Fellows et al. [77] report that the parameter remains NP-complete for planar graphs. However, for a chordal graph G, Jacobson and Peters [131] show that $\Gamma(G) = \beta_0(G)$, and β_0 is polynomial-time solvable for chordal graphs [95]. Similarly, for a bipartite graph G, Cockayne et al. [44] show that $\Gamma(G) = \beta_0(G)$, and β_0 is polynomial-time solvable for bipartite graphs [106]. Thus, for a tree T, $\Gamma(T) = \beta_0(T)$. A linear time algorithm to find $\beta_0(T)$ is given by Daykin and Ng [62].

4.3.4 Total domination

The components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ of the source MINIMUM TOTAL DOMINATING SET problem are defined as follows:

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is a total dominating set for V, i.e., $N(V') = V$.
- $\text{OPT} = \min$.

The maximinimal problem name is MAXIMUM MINIMAL TOTAL DOMINATING SET or UPPER TOTAL DOMINATION.

Let $\gamma_t(G)$ and $\Gamma_t(G)$ denote respectively the minimum and maximum over all minimal total dominating sets of a given graph G.

The total domination number, γ_t was first defined and studied by Cockayne et al. [41]. The parameter remains NP-complete for bipartite graphs [185], undirected path graphs (a subclass of chordal graphs) [152], split graphs (also a subclass of chordal graphs) [151], 2-CUBs [58] and circle graphs [141], though polynomial-time solvable for strongly chordal graphs [31], interval and circular-arc graphs [34], permutation graphs [26, 58], cocomparability graphs [147], asteroidal triple-free graphs [146], series-parallel graphs [184], partial k-trees (for fixed k) [204] and trees [152]. Polynomial-time algorithms to compute γ_t in dually chordal graphs, distance hereditary graphs, and k-polygon graphs (for fixed k) follow from [148] and the polynomial-time algorithms to compute γ in the same graph classes (see Chapter 8 of [114] for further details).

The complexity of γ_t in planar graphs does not seem to have been discussed in the literature. We show that NP-completeness holds for planar graphs of maximum degree 3.

Theorem 4.3.6 MINIMUM TOTAL DOMINATING SET DECISION *is NP-complete, even for planar graphs of maximum degree 3.*

Proof: Clearly MINIMUM TOTAL DOMINATING SET DECISION is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$ and a set S of at most K vertices, it is straightforward to verify in polynomial time that S is a total dominating set.

To show NP-hardness, we give a transformation from the MINIMUM DOMINATING SET DECISION problem for planar cubic graphs. That this problem remains NP-complete for planar cubic graphs is shown by Kikuno et al. [144]. Hence let $G = (V, E)$ (a planar cubic graph) and K (a positive integer) be an instance of MINIMUM DOMINATING SET DECISION. Assume that $|E| = m$. We construct an instance $G' = (V', E')$ (planar graph of maximum degree 3) and K' (positive integer) of MINIMUM TOTAL DOMINATING SET DECISION. Corresponding to every edge $e = \{u, v\}$ of E, construct an *edge component* of G' as follows: replace the edge e by a path on seven vertices, namely $u, a_e^u, c_e^u, d_e, c_e^v, a_e^v, v$, connected in that order. Create two additional vertices b_e^u and b_e^v; join b_e^u to a_e^u and c_e^u, and join b_e^v to a_e^v and c_e^v. It may be verified that G' is planar and of maximum degree 3. An example edge component is shown in Figure 4.4. Denote by V_e the internal vertices in the edge component corresponding to edge e of G, i.e.

$$V_e = \{a_e^u, b_e^u, c_e^u, d_e, c_e^v, b_e^v, a_e^v\}.$$

Denote by X_e the following vertices in V_e:

$$X_e = \{b_e^u, c_e^u, d_e, c_e^v, b_e^v\}.$$

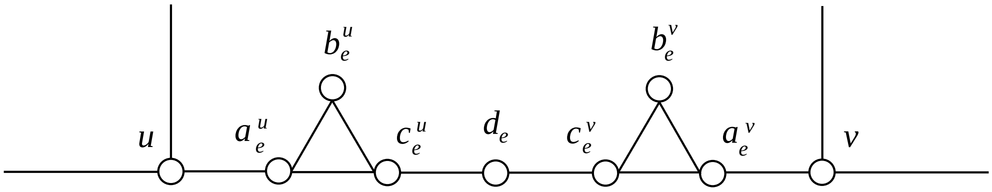

Figure 4.4: A typical edge component from the constructed instance of MINIMUM TOTAL DOMINATING SET DECISION.

Define $X = \bigcup_{e \in E} X_e$. Set $K' = K + 3m$. We claim that G has a dominating set of cardinality at most K if and only if G' has a total dominating set of cardinality at most K'.

For, suppose that D is a dominating set for G, where $|D| = k \leq K$. We construct a total dominating set D' for G'. Initially, let D' contain the vertices of D. For any edge $e = \{u, v\}$ of G, we add vertices to D', according to four cases:

1. $u \notin D, v \notin D$. Add vertices c_e^u, d_e, c_e^v to D'.
2. $u \in D, v \notin D$. Add vertices a_e^u, c_e^v, a_e^v to D'.
3. $u \notin D, v \in D$. Add vertices a_e^u, c_e^u, a_e^v to D'.
4. $u \in D, v \in D$. Add vertices a_e^u, c_e^u, a_e^v to D'.

It may be verified that D' is a total dominating set for G', and $|D'| = k + 3m \leq K'$.

Conversely, suppose that G' has a total dominating set of size at most K'. Choose D' to be such a set which minimises $|D' \cap X|$. Let e be any edge of E and define $Q_e = V_e \cap D'$. It may be verified that $|Q_e| \geq 3$; now suppose that $|Q_e| > 3$. Then $|D' \cap X_e| \geq 2$. Define a set D'' as follows:

$$D'' = (D' \backslash Q_e) \cup \{u, a_e^u, c_e^v, a_e^v\}.$$

Then it is straightforward to check that D'' is a total dominating set for G', $|D''| \leq |D'| \leq K'$, and $|D'' \cap X_e| = 1$. Thus $|D'' \cap X| < |D' \cap X|$, contradicting the choice of D'. Hence $|Q_e| = 3$.

Now let $D = D' \cap V$. We claim that D is a dominating set for G. For, suppose that $u \in V \backslash D$. Then $u \notin D'$, so by the domination property of D', there is some $e = \{u, v\} \in E$ such that $a_e^u \in D'$. By the total domination property of D', either $b_e^u \in D'$ or $c_e^u \in D'$. But $|Q_e| = 3$, so that $|Q_e \cap \{d_e, c_e^v, b_e^v, a_e^v\}| = 1$. Since b_e^v and c_e^v must be dominated by this single vertex, the total domination property forces $a_e^v \in D'$ and $v \in D'$. Hence $v \in D$, so that D is a dominating set for G. Finally, $|D| = |D'| - 3m \leq K' - 3m = K$. ∎

The construction of Theorem 4.3.6 can be extended in order to show that γ_t remains NP-complete for planar cubic graphs.

Theorem 4.3.7 MINIMUM TOTAL DOMINATING SET DECISION *is NP-complete for planar cubic graphs.*

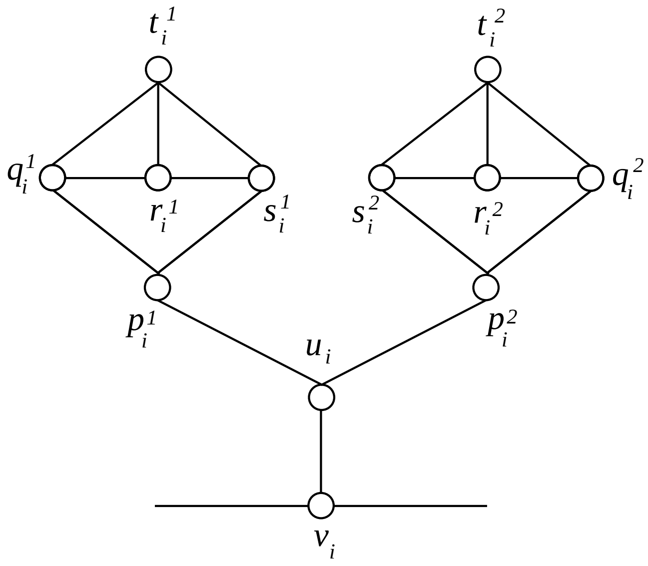

Figure 4.5: A typical degree two attachment from the constructed instance of MINIMUM TOTAL DOMINATING SET DECISION.

By considering the construction of Theorem 4.3.6, it may be verified that MINIMUM TOTAL DOMINATING SET DECISION is NP-complete for planar graphs, where each vertex has degree two or three. To show that NP-completeness also holds for planar cubic graphs, we give a reduction from this problem. Hence let $G = (V, E)$ (a planar graph, where each vertex has degree two or three) and K (a positive integer) be an instance of MINIMUM TOTAL DOMINATING SET DECISION. Assume that $V = \{v_1, v_2, \ldots, v_n\}$. Let D_2 be the set of vertices in G of degree two, and let $n_2 = |D_2|$. We construct an instance $G' = (V', E')$ (planar cubic graph) and K' (positive integer) of MINIMUM TOTAL DOMINATING SET DECISION as follows. Corresponding to every $v_i \in D_2$, contruct a *degree two attachment* as follows:

- Vertices $p_i^j, q_i^j, r_i^j, s_i^j, t_i^j, u_i$ $(j = 1, 2)$.
- Edges $\{v_i, u_i\}, \{u_i, p_i^j\}, \{p_i^j, q_i^j\}, \{p_i^j, s_i^j\}, \{q_i^j, r_i^j\}, \{r_i^j, s_i^j\}, \{q_i^j, t_i^j\}, \{r_i^j, t_i^j\}, \{s_i^j, t_i^j\}$ $(j = 1, 2)$.

It is clear that the graph G' constructed is planar and cubic. A typical degree two attachment of G' is illustrated in Figure 4.5. The construction bears similarities to one used by Kikuno et al. [144], proving NP-completeness for γ in planar cubic graphs. For $v_i \in D_2$, denote by V_i the following vertices in the corresponding degree two attachment:

$$V_i = \{p_i^j, q_i^j, r_i^j, s_i^j, t_i^j : j = 1, 2\}.$$

Set $K' = K + 4n_2$. We now show that G has a total dominating set of cardinality at most K if and only if G' has a total dominating set of cardinality at most K'.

For, suppose that D is a total dominating set for G, where $|D| \le K$. Let

$$D' = D \cup \{p_i^1, s_i^1, p_i^2, s_i^2 : v_i \in D_2\}.$$

Then it may be verified that D' is a total dominating set for G' and $|D'| = |D| + 4n_2 \le K + 4n_2 = K'$ as required.

Conversely, suppose that D' is a total dominating set for G', where $|D'| \le K'$. Suppose that $v_i \in D_2$, for some i $(1 \le i \le n)$. Then $|D' \cap V_i| \ge 4$. For, some vertex w_j satisfies $w_j \in D' \cap \{q_i^j, r_i^j, s_i^j\}$, in order to dominate t_i^j $(j = 1, 2)$. Furthermore, some additional vertex x_j satisfies $x_j \in D' \cap \{p_i^j, q_i^j, r_i^j, s_i^j, t_i^j\}$, in order to dominate w_j $(j = 1, 2)$. Define

$$\begin{aligned} X &= \{v_i \in D_2 : v_i \in D' \wedge u_i \in D'\} \\ Y &= \{v_i \in D_2 : v_i \in D' \wedge u_i \notin D'\} \\ Z &= (V \backslash D_2) \cap D'. \end{aligned}$$

Let $x = |X|$, $y = |Y|$ and $z = |Z|$. Then

$$\begin{aligned} |D'| &\ge 6x + 5y + 4(n_2 - x - y) + z \\ &= 4n_2 + 2x + y + z \\ &= |D' \cap V| + 4n_2 + x \end{aligned} \tag{4.4}$$

Define a set D'' as follows. Initially let $D'' = D'$. Corresponding to each $v_i \in X$, replace u_i in D'' by some v_j adjacent in G to v_i. Now let $D = D'' \cap V$. Clearly, D is a total dominating set for G. Finally, $|D| \le |D' \cap V| + x$, so that, by Inequality 4.4, $|D| \le |D'| - 4n_2 \le K' - 4n_2 = K$ as required. ■

The maximum minimal total domination number, Γ_t, is studied by Fricke et al. [83], who show that the parameter is NP-complete for bipartite graphs, though they give a linear-time algorithm for trees.

4.3.5 Edge domination

The components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ of the source MINIMUM EDGE DOMINATING SET problem are defined as follows:

- $\mathcal{U}(G) = \mathbb{P}(E)$
- $\pi(G, E') \Leftrightarrow E'$ is an edge dominating set for E
- $\textsc{opt} = \min$.

The maximinimal problem name is MAXIMUM MINIMAL EDGE DOMINATING SET.

Let $\gamma'(G)$ and $\Gamma'(G)$ denote respectively the minimum and maximum over all minimal edge dominating sets of a given graph G.

The minimum edge domination parameter, γ', was first studied by Mitchell and Hedetniemi [168]. The parameter remains NP-complete for planar or bipartite graphs of maximum degree 3 [213], planar bipartite graphs, their subdivision, line and total graphs[6], perfect claw-free graphs, planar cubic graphs and iterated total graphs [126]. The problem of computing γ' is polynomial-time solvable for bipartite permutation graphs and cotriangulated graphs [200], trees [168, 213], k-outerplanar graphs [12] and a number of other classes of graphs including claw-free chordal graphs [126].

It would appear that the only complexity result for the maximum minimal edge domination parameter, Γ', is an NP-completeness result for bipartite graphs, due to McRae [165].

4.3.6 Irredundance

The components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ of the source MAXIMUM IRREDUNDANT SET or UPPER IRREDUNDANCE problem are defined as follows:

- $\mathcal{U}(G) = \mathbb{P}(V)$
- $\pi(G, V') \Leftrightarrow V'$ is irredundant in G, i.e. $\forall\, v \in V' \bullet N[v] \backslash N[V' \backslash \{v\}] \neq \varnothing$
- $\text{OPT} = \max$.

The minimaximal problem name is MINIMUM MAXIMAL IRREDUNDANT SET or LOWER IRREDUNDANCE.

Let $ir(G)$ and $IR(G)$ denote respectively the minimum and maximum over all maximal irredundant sets of a given graph G.

The concept of irredundance was introduced by Cockayne et al. [49]. The maximum (or upper) irredundance number, IR, is NP-complete for arbitrary graphs [77]. Fellows et al. [77] also report that IR is NP-complete for planar graphs. However, for a chordal graph G, Jacobson and Peters [131] show that $IR(G) = \beta_0(G)$, and β_0 is polynomial-time solvable for chordal graphs [95]. Similarly, for a bipartite graph G, Cockayne et al. [44] show that $IR(G) = \beta_0(G)$, and β_0 is polynomial-time solvable for bipartite graphs [106]. Thus, for a tree T, $IR(T) = \beta_0(T)$. A linear time algorithm to find $\beta_0(T)$ is given by Daykin and Ng [62].

The minimum maximal (or lower) irredundance number, ir, is NP-complete for bipartite graphs [185] and chordal graphs [151], though solvable in linear time for trees [18].

[6] Given a graph $G = (V, E)$, the *line graph* of G, $L(G)$, has vertex set E, and two vertices of $L(G)$ are adjacent in $L(G)$ if and only if the corresponding edges are adjacent in G. To form the *subdivision graph* of G, $S(G)$, we subdivide each edge $\{v, w\}$ of G, i.e. we add vertices $u_{v,w}$ and edges $\{v, u_{v,w}\}$ and $\{u_{v,w}, w\}$ for each $\{v, w\} \in E$. The *total graph* of G, $T(G)$ has vertex set $V \cup E$, and two vertices of $T(G)$ are adjacent if and only if they are incident or adjacent as vertices or edges of G.

The following inequality chain involving six parameters studied in Sections 4.2.6, 4.3.3 and 4.3.6, has received much attention:

$$ir(G) \leq \gamma(G) \leq i(G) \leq \beta_0(G) \leq \Gamma(G) \leq IR(G).$$

The inequality chain was first observed by Cockayne et al. [49]; further details may be found in [115, Section 3.5].

4.3.7 Conclusion and further study regarding minimaximal and maximinimal optimisation problems in this section

In this section we have studied complexity results relating to parameters concerned with strong stability, cliques, domination, total domination, edge domination and irredundance. As discussed in Section 4.2.9, the complexity of γ' is open for chordal graphs. Also, no reference was found for the complexity of ω^- in chordal graphs.

Chapter 5

Minimaximal and maximinimal graph problems based on the partial order of $(k-1, k)$-replacement

5.1 Introduction

In this chapter we consider a number of examples of minimaximal and maximinimal graph optimisation problems that may be derived from a source optimisation problem Π using the partial order of $(k-1, k)$-replacement, $\subset_k^G$, defined on $\mathcal{F}(G)$, the feasible solutions of Π, for a given graph G. An element of $\mathcal{F}(G)$ that is $\subset_k^G$-maximal or $\subset_k^G$-minimal will be referred to as *k-maximal* or *k-minimal*, respectively. The minimaximal and maximinimal optimisation problems that we formulate in this chapter have been studied previously in the literature in some form.

We begin in Section 5.2 by considering the case $k = 1$, and take Π to be the source MAXIMUM NEARLY PERFECT SET and MINIMUM NEARLY PERFECT SET problems. We discuss the computational complexity of these problems, together with their respective minimaximal or maximinimal counterparts, namely MINIMUM 1-MAXIMAL NEARLY PERFECT SET and MAXIMUM 1-MINIMAL NEARLY PERFECT SET.

In Section 5.3, we consider the case $k \geq 1$, and take Π to be the source MAXIMUM INDEPENDENT SET problem. We place particular emphasis on the case $k = 2$, focusing on the resulting MINIMUM 2-MAXIMAL INDEPENDENT SET problem. We study the algorithmic complexity of this problem in trees and planar graphs.

As this chapter concludes our study of minimaximal and maximinimal graph optimisation problems in this thesis, we summarise in Section 5.4, in table format, the algorithmic results that we have obtained and surveyed for the parameters in Chapters 3, 4 and 5, over arbitrary, planar, bipartite and chordal graphs, and trees.

There are many other minimaximal and maximinimal graph optimisation problems that can be obtained by applying a partial order satisfying POMM to the feasible solutions of problems of this type. Some of these minimaximal and maximinimal problems have been introduced in the literature, though none has been studied in detail with regard to computational complexity. In Section 5.5, we list some further minimaximal and maximinimal graph optimisation problems that might deserve attention.

5.2 1-maximal and 1-minimal nearly perfect sets

Let $G=(V,E)$ be a graph, and let $S \subseteq V$. S is a *nearly perfect set* of G if, for every $v \in V \backslash S$, $|N(v) \cap S| \leq 1$, i.e. every vertex outside S is adjacent to at most one vertex of S. Nearly perfect sets were first studied by Dunbar et al. [68]. We may define two optimisation problems relating to this concept as follows:

Source problem: MAXIMUM (MINIMUM) NEARLY PERFECT SET$=\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, where

- $\mathcal{I}=\{G=(V,E) : G \text{ is a graph}\}$
- $\mathcal{U}(G)=\mathbb{P}(V)$, for $G \in \mathcal{I}$
- $\pi(G, V') \Leftrightarrow V'$ is a nearly perfect set of G, for $G \in \mathcal{I}$ and $V' \in \mathcal{U}(G)$
- $m(G, V')=|V'|$, for $G \in \mathcal{I}$ and $V' \in \mathcal{F}(G)$
- OPT $=\max(\min)$.

Both MAXIMUM NEARLY PERFECT SET and MINIMUM NEARLY PERFECT SET are trivially in P, since both V and $\varnothing$ are nearly perfect sets of G. Similarly, $\subset^G$-maximal nearly perfect sets and $\subset^G$-minimal nearly perfect sets are not particularly interesting concepts, since these also correspond to V and $\varnothing$, respectively. However, it is a consequence of the fact that nearly perfect-ness is not a hereditary or super-hereditary property that we can define a meaningful concept of local optimality, using the partial order of $(0,1)$-replacement. It is straightforward to see that the property of nearly perfect-ness is neither hereditary nor super-hereditary. Consider the triangle K_3, with vertices u, v, w. The inclusions $\{u\} \subset \{u,v\} \subset \{u,v,w\}$ hold, and $\{u\}, \{u,v,w\}$ are nearly perfect sets, but $\{u,v\}$ is not.

In fact, both 1-maximal and 1-minimal nearly perfect sets are interesting notions. Using the partial order $\subset_1^G$, defined on $\mathcal{F}(G)$, the set of all nearly perfect sets for a graph G, we obtain the MINIMUM 1-MAXIMAL NEARLY PERFECT SET and MAXIMUM 1-MINIMAL NEARLY PERFECT SET problems, using the framework of Definition 2.3.5. Dunbar et al. [68] denote by $n_p(G)$ and $N_p(G)$ the cardinality of a minimum 1-maximal and maximum 1-minimal nearly perfect set, respectively. They show that n_p is NP-complete in bipartite and chordal graphs, though solvable in linear time for trees, whereas the parameter N_p is linear-time computable for all graphs.

5.3 k-maximal independent sets ($k \geq 1$)

5.3.1 Introduction

Independent sets in graphs have been extensively studied. The principal algorithmic focus has been the investigation of how we may efficiently determine maximum independent sets in a graph. The computational complexity of the problem of computing $\beta_0(G)$, the cardinality of a maximum independent set in a graph G has been studied extensively, for a variety of classes of graphs (see Section 4.2.6).

Independent sets that are maximal with respect to the partial order of set inclusion have also been of interest. For instance, the problem of counting the number of maximal independent sets in a graph has been shown to be #P-complete [210]. Also, a graph is *well-covered* if every maximal independent set is maximum – results in this area are surveyed by Plummer [186]. A much-studied parameter is $\beta_0^-(G)$, the cardinality of a minimum maximal independent set in G. The parameter β_0^- is also referred to as the minimum independent domination parameter i. Independent domination was first studied by Cockayne and Hedetniemi [47], and has also been the focus of much algorithmic activity (see Section 4.2.6).

For a graph G, we saw in Section 4.2.6 that the definition of $\beta_0^-(G)$ can be obtained by defining the partial order $\subset^G$ (the partial order of set inclusion) on $\mathcal{F}(G)$, the set of all independent sets in G, and by considering the minimum over all $\subset^G$-maximal elements of $\mathcal{F}(G)$. However, it is of interest to consider other partial orders that may be defined on $\mathcal{F}(G)$, and the corresponding minimaximal optimisation problems that result from their definition.

In Section 5.3 we define the partial order $\subset_k^G$ on $\mathcal{F}(G)$, for $k \geq 1$, and consider the k-maximal members of $\mathcal{F}(G)$, for a given graph $G = (V, E)$. Recall that an independent set S is k-maximal if, for all subsets A of S (where $|A| \leq k-1$), and all subsets B of $V \backslash S$ (where $|B| = |A|+1$), $(S \backslash A) \cup B$ is non-independent. The parameter $\beta_{0,k}^-(G)$[1] will denote the minimum over all k-maximal independent sets of G. The concept of k-maximal independence in graphs was introduced by Bollobás et al. [22], and several non-algorithmic results concerning $\beta_{0,k}^-$, for $k \geq 1$, have been obtained [169, 50]. The related concept of *k-minimal domination* was also introduced by Bollobás et al. [22], and further details may be found in [170, 52, 53]. Halldórsson's approximation algorithms for MAXIMUM INDEPENDENT SET in various graph classes [105] involve constructing k-maximal independent sets. Nevertheless, k-maximal independent sets are interesting in their own right.

Investigating the computational complexity of $\beta_{0,k}^-$ for $k > 1$ was given as an open problem by Cockayne et al. [50]. However, the parameter $\beta_{0,2}^-$ has been studied by McRae [165], from an algorithmic point of view. She shows that the decision problem related to

[1] Mynhardt [169] and Cockayne et al. [50] refer to $\beta_{0,k}^-$ as β_k. However, for $k = 1$, this choice coincides with the maximum matching parameter (see Harary [106], for example), and for $k = 2$, this choice coincides with the maximum total matching parameter of Alavi et al. [2]. In our notation, the subscript '0' of $\beta_{0,k}^-$ refers to *vertex independence* (as in Harary [106]), the subscript 'k' refers to k-maximality, and the superscript '-' refers to the minimum cardinality requirement.

determining $\beta_{0,2}^-$ is NP-complete for bipartite graphs and line graphs of bipartite graphs. In this section we give a linear algorithm for computing the minimum 2-maximal independence number of a tree. The algorithm is based on that of Beyer et al. [20] for computing $\beta_0^-(T)$. We also demonstrate that the decision problem related to computing $\beta_{0,2}^-$ is NP-complete for planar graphs of maximum degree 3.

The remainder of Section 5.3 is organised as follows. In Section 5.3.2, we define several notions related to k-maximal independence. In Section 5.3.3 we give the main theorem, on which the algorithm is based, and in Section 5.3.4, we present the algorithm itself. The NP-completeness result for planar graphs is given in Section 5.3.5. Finally, in Section 5.3.6, we discuss some possible directions for further study, based on a hierarchy of k-maximal independence parameters.

5.3.2 Definitions related to k-maximal independence

Recall from Section 4.2.6 the definitions of the components $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ for the MAXIMUM INDEPENDENT SET problem. For $k \geq 1$, consider the partial order of $(k-1,k)$-replacement, $\subset_k^G$, defined on $\mathcal{F}(G)$, the set of all independent sets of a graph G. The resulting minimaximal optimisation problem is called MINIMUM k-MAXIMAL INDEPENDENT SET. We may now formally define the parameters $\beta_{0,k}^-$, as follows:

$$\beta_{0,k}^-(G) = \min\{|S| : S \in \mathcal{F}(G) \wedge S \text{ is } k\text{-maximal}\}.$$

Since independence is a hereditary property, a 1-maximal member of $\mathcal{F}(G)$ is maximal by Proposition 2.4.6. Thus $\beta_{0,1}^- = \beta_0^-$. By Proposition 2.4.3, $\beta_{0,k-1}^-(G) \leq \beta_{0,k}^-(G)$ for a graph G and $k \geq 2$. Also, $\beta_{0,k}^-(G) \geq k$ for $1 \leq k \leq \beta_0(G)$ [169].

For the remainder of this section, and in Sections 5.3.3-5.3.5, we are concerned with the case $k = 2$. We have already noted that $\beta_0^-(G) \leq \beta_{0,2}^-(G)$. A simple example of where strict inequality can occur is provided by P_3: $\beta_0^-(P_3) = 1$, whereas $\beta_{0,2}^-(P_3) = 2$.

The following definitions relating to graphs may be used to obtain a convenient criterion for an independent set to be 2-maximal. Let $G = (V, E)$ be a graph. For a set of vertices $S \subseteq V$ and a vertex $v \in V$, the *private S-neighbours of v* are those vertices in the set $N[v]\backslash N[S\backslash\{v\}]$. We say that *$S$ admits an augmenting P_3 in G* if there exist vertices w, x and y of V such that $w \in S$, $x, y \notin S$, $\{x, y\} \notin E$ and x, y are private S-neighbours of w. It turns out that a maximal independent set S is 2-maximal if and only if S does not admit an augmenting P_3 in G, as the following result, due to McRae [165], demonstrates. We include her proof for completeness.

Lemma 5.3.1 (McRae [165]) *Let $G = (V, E)$ be a graph. A maximal independent set of vertices $S \subseteq V$ is 2-maximal if and only if S does not admit an augmenting P_3 in G.*

Proof: Suppose $S \subseteq V$ is a maximal independent set. If there exist vertices w, x and y of V such that $w \in S$, $x, y \notin S$, $\{x, y\} \notin E$ and x, y are private S-neighbours of w, then it is clear that S is not 2-maximal. Conversely suppose that S is not 2-maximal. Then there exist vertices w, x and y of V such that $w \in S$, $x, y \notin S$ and $(S\backslash\{w\}) \cup \{x, y\}$ is independent. As S is maximal independent, $S\backslash\{w\}$ dominates $V\backslash N[w]$. Thus $x, y \in$

$N(w)$, and x, y were private S-neighbours of w. Clearly also $\{x, y\} \notin E$. ■

The result of Lemma 5.3.1 is utilised by our algorithm to find the cardinality of a minimum 2-maximal independent set of a tree.

5.3.3 The main theorem for trees

Before presenting the main result, we make some further definitions relating to trees. Our algorithm is adapted from the one used by Beyer et al. [20] to calculate $i(T)$ for a tree T, and hence we use similar notation.

Given a rooted tree T and a vertex v of T, define T_v to be the subtree of T with root v. For a vertex v of T and a set of vertices $S \subseteq V(T)$, v is said to be *bad* with respect to S if v is a leaf node of T and $v \notin S$, *or*, v is a non-leaf node of T, $v \notin S$ and no child of v is in S. As a result of Lemma 5.3.1, the following necessary condition for an independent set S of a tree T to be 2-maximal can easily be verified.

Lemma 5.3.2 *Let T be a rooted tree and let $S \subseteq V(T)$ be independent. Then S is 2-maximal implies that every v in S has at most one bad child with respect to S.* ■

In view of this observation, we define the following five functions:

$INBC_T(v)$: The smallest number of vertices in a 2-maximal independent set $S \subseteq V(T_v)$ for T_v which includes v and is such that v has one bad child with respect to S.

$INN_T(v)$: The smallest number of vertices in a 2-maximal independent set $S \subseteq V(T_v)$ for T_v which includes v and is such that v has no bad children with respect to S.

$OUTC_T(v)$: The smallest number of vertices in a 2-maximal independent set $S \subseteq V(T_v)$ for T_v which does not include v but is such that *either*

1. v has more than one child in S *or*
2. v has exactly one child w in S, and w has no bad children with respect to S.

$OUTO_T(v)$: The smallest number of vertices in a maximal independent set $S \subseteq V(T_v)$ for T_v which does not include v but is such that v has exactly one child w in S, w has one bad child x with respect to S, and S admits exactly one augmenting P_3 in T_v (namely the P_3 involving vertices v, w, x).

$OUTN_T(v)$: The smallest number of vertices in an independent set $S \subseteq V(T_v)$ for T_v such that S dominates $V(T_v) \backslash \{v\}$, S does not admit an augmenting P_3 in T_v, and v is bad with respect to S.

Intuitively, for $OUTO_T(v)$ and $OUTN_T(v)$, v must be a non-root node, in order to be dominated by its parent in T.

One or more of these functions may be undefined for some $v \in V(T)$. In particular, if v is a leaf node of T, then $INBC_T(v)$, $OUTC_T(v)$ and $OUTO_T(v)$ are all undefined. These cases are dealt with in Section 5.3.4.

The algorithm for finding the minimum 2-maximal independence number of a rooted tree T utilises the observation that, if S is a 2-maximal independent set of T, then for each vertex $v \in S$, v has at most one bad child by Lemma 5.3.2, and also S does not admit an augmenting P_3 in T involving the the parent of v, v itself, and a child of v (assuming that such vertices exist). The dynamic programming approach is based on the following theorem, which demonstrates relationships between the above five functions for adjacent vertices in a tree, and which also proves the correctness of the algorithm.

Theorem 5.3.3 *Let $T' = (V', E')$ be a tree rooted at some vertex u, and let $T'' = (V', E'')$ be another tree rooted at some vertex v. Let $T = (V, E)$ be the tree rooted at u, with vertices $V = V' \cup V''$ and edges $E = E' \cup E'' \cup \{\{u, v\}\}$. Then*

1. $INBC_T(u) = \min \left\{ \begin{array}{l} INBC_{T'}(u) + OUTC_{T''}(v), \\ INBC_{T'}(u) + OUTO_{T''}(v), \\ INN_{T'}(u) + OUTN_{T''}(v) \end{array} \right\}$

2. $INN_T(u) = INN_{T'}(u) + \min\{OUTC_{T''}(v), OUTO_{T''}(v)\}$

3. $OUTC_T(u) = \min \left\{ \begin{array}{l} OUTC_{T'}(u) + OUTC_{T''}(v), \\ OUTN_{T'}(u) + INN_{T''}(v), \\ OUTC_{T'}(u) + INBC_{T''}(v), \\ OUTC_{T'}(u) + INN_{T''}(v), \\ OUTO_{T'}(u) + INBC_{T''}(v), \\ OUTO_{T'}(u) + INN_{T''}(v) \end{array} \right\}$

4. $OUTO_T(u) = \min \left\{ \begin{array}{l} OUTO_{T'}(u) + OUTC_{T''}(v), \\ OUTN_{T'}(u) + INBC_{T''}(v) \end{array} \right\}$

5. $OUTN_T(u) = OUTN_{T'}(u) + OUTC_{T''}(v)$.

Proof: For Cases 1-3, let S be a minimum 2-maximal independent set for T and define $S' = S \cap V(T')$ and $S'' = S \cap V(T'')$. We consider each of Cases 1-5 separately:

(1). Suppose that S contains u, and u has one bad child w with respect to S in T. Then either (i) $w \in S'$, or (ii) w is v.

In (i), $|S'| = INBC_{T'}(u)$. Also, v has at least one child in S. The existence of u implies that if v has only one child x in S, then x is allowed to have a bad child with respect to S'' in T''. Thus $|S''| = \min\{OUTC_{T''}(v), OUTO_{T''}(v)\}$.

In (ii), u has no bad child with respect to S' in T', so $|S'| = INN_{T'}(u)$. Also, v has no children in S, so $|S''| = OUTN_{T''}(v)$.

(2). Suppose that S contains u, and u has no bad children with respect to S in T. Then u has no bad children with respect to S' in T'. Hence $|S'| = INN_{T'}(u)$. Also, v has at least one child in S. As in (1), the existence of u implies that if v has only one child w in S, then w is allowed to have a bad child with respect to S'' in T''. Thus $|S''| = \min\{OUTC_{T''}(v), OUTO_{T''}(v)\}$.

(3). Suppose that S does not contain u, but is such that either u has more than one child in S, or u has exactly one child w in S and w has no bad children with respect to S in T. We consider the cases (i) u has a child in S', or (ii) $v \in S$, or (iii) both (i) and (ii).

In (i), u cannot have a sole child w in S' such that w has a bad child with respect to S', since $v \notin S$. Thus $|S'| = OUTC_{T'}(u)$. Also, v must have at least one child in S, or else $S \cup \{v\}$ is independent, a contradiction. If v has only one child x which has a bad child with respect to S'', then S is not 2-maximal, since $u \notin S$. Hence $|S''| = OUTC_{T''}(v)$.

In (ii), $|S'| = OUTN_{T'}(u)$. Also, v cannot have a bad child with respect to S'' in T'', for then S would not be 2-maximal, as no child of u is in S'. Hence $|S''| = INN_{T''}(v)$.

In (iii), the existence of a child of u in S' means that v is permitted to have a bad child with respect to S'' in T''. Hence $|S''| = \min\{INBC_{T''}(v), INN_{T''}(v)\}$. Also, the existence of $v \in S$ means that it is permissible for u to have a sole child x in S' such that x has a bad child with respect to S'. Hence $|S'| = \min\{OUTC_{T'}(u), OUTO_{T'}(u)\}$.

(4). Let S be a minimum maximal independent set for T such that S does not include u, but exactly one child w of u is in S, w has one bad child x with respect to S, and S admits exactly one augmenting P_3 in T (namely the P_3 involving the vertices u, w, x). Let $S' = S \cap V(T')$ and $S'' = S \cap V(T'')$. Then either (i) $w \in S'$ or (ii) w is v.

In (i), $|S'| = OUTO_{T'}(u)$. Also, v must have a child in S, or else $S \cup \{v\}$ is independent, a contradiction. If v has only one child y which has a bad child z with respect to S'', then S admits an augmenting P_3 in T involving vertices v, y, z, since $u \notin S$, a contradiction. Hence $|S''| = OUTC_{T''}(v)$.

In (ii), $|S'| = OUTN_{T'}(u)$. Also, $|S''| = INBC_{T''}(v)$.

(5). Let S be a minimum independent set for T such that S dominates $V(T)\backslash\{u\}$, S does not admit an augmenting P_3 in T, and u is bad with respect to S. Let $S' = S \cap V(T')$ and $S'' = S \cap V(T'')$.

Then $|S'| = OUTN_{T'}(u)$. Also, v must have a child in S, or else S does not dominate v, a contradiction. If v has only one child w which has a bad child with respect to S'', then S admits an augmenting P_3 in T, since $u \notin S$, a contradiction. Hence $|S''| = OUTC_{T''}(v)$. ∎

5.3.4 Linear algorithm for trees

The class of rooted trees may be constructed recursively from copies of the singleton vertex K_1, together with the rule of composition indicated in the statement of Theorem 5.3.3 (that is, take a tree T', rooted at some vertex u, and another tree T'', rooted at some vertex v, and let T be the tree with root u, constructed from T' and T'' by joining u and v by an edge). Hence, by providing suitable initialisations for the singleton subtrees of a given tree T rooted at some vertex u, we may build up T from the singleton subtrees, using this rule of composition successively at each stage, together with Theorem 5.3.3, to obtain the values of the five functions at vertices of T.

The values $OUTO_T(u)$ and $OUTN_T(u)$ must be disregarded, since a set that satisfies the defining properties for either of the functions $OUTO_T$ or $OUTN_T$ must be covered from above in T in order to be 2-maximal. Since this is impossible in the case of the root vertex, we have:

$$\beta^-_{0,2}(T) = \min\{INBC_T(u), INN_T(u), OUTC_T(u)\}.$$

Thus it remains to supply appropriate initialisations for the singleton subtrees of T. For a singleton tree T, consisting only of some vertex v, it is clear that $INN_T(v) = 1$ and $OUTN_T(v) = 0$. However, as mentioned in the previous section, the values of the other three functions $INBC_T(v)$, $OUTO_T(v)$ and $OUTC_T(v)$ are undefined. They are therefore given value $N+1$ (where $N = |V(T)|$), which is large enough not to affect the remainder of the procedure in computing $\beta^-_{0,2}(T)$. For a vertex v of an arbitrary tree T, it follows by Theorem 5.3.3 that there is no set satisfying the defining property of $S_T(v)$ if and only if $S_T(v) \geq N+1$, where S is one of the five classes defined above.

The pseudocode of our algorithm to find $\beta^-_{0,2}(T)$ for a tree T is shown in Figure 5.1. We assume that an arbitrary vertex is chosen to be the root of T. Also, we suppose that the vertices are ordered $1, 2, \ldots, N$ such that vertex 1 is the root, and the remaining vertices are numbered breadth-first from the root. It follows from Theorem 5.3.3 and the discussion in this section that this algorithm is correct.

It may be verified that the algorithm in Figure 5.1 requires $O(N)$ time for execution. The initialisation is clearly $O(N)$, and the main loop is also $O(N)$, since the values of the five functions may be computed in a constant number of steps, for each iteration. It is also clear that $O(N)$ space is required.

In order to construct a minimum 2-maximal independent set of a tree T, it is necessary to store extra information during the main loop of the algorithm in Figure 5.1. This extra information is then used in a final pass over the vertices in the order $1, 2, \ldots, N$, during which a minimum 2-maximal independent set is constructed. This is a standard procedure for problems of this type, taking $O(N)$ time and space – further details may be found in [18].

5.3.5 2-maximality in planar graphs

In this section we prove that $\beta^-_{0,2}$ is NP-complete, even for planar graphs of maximum degree 3.

Theorem 5.3.4 MINIMUM 2-MAXIMAL INDEPENDENT SET DECISION *is NP-complete, even for planar graphs of maximum degree 3.*

Proof: Clearly MINIMUM 2-MAXIMAL INDEPENDENT SET DECISION is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$ and a set S of at most K elements, it is straightforward to verify in polynomial time that S is independent and 2-maximal in G.

To show NP-hardness, we give a transformation from the MAXIMUM INDEPENDENT SET DECISION problem for planar cubic graphs. That this problem remains NP-complete

procedure *tree-min-2-max-ind-number* (**value** *parent* : **array** $[2..N]$ **of** $[1..N]$)
(**result** $\beta^-_{0,2} : \mathbb{N}$) $\widehat{=}$

|[
{ Given a tree T with vertices $1, 2, \ldots, N$, calculate $\beta^-_{0,2}(T)$. T is rooted at vertex 1,}
{ and the vertices are numbered breadth-first from the root. T is represented by a }
{ *parent* array, i.e., $i = parent[j]$ if and only if i is the parent of j in T. }

var $i, j : \mathbb{Z}^+$;
$INBC_T, INN_T, OUTC_T, OUTN_T, OUTO_T$: **array** $[1..N]$ **of** $\mathbb{N}$ •
$j := 1$;
while $j \leq N$ **do** {Initialisation}
$INBC_T[j] := N + 1$; {Indicates value undefined in this case}
$INN_T[j] := 1$;
$OUTC_T[j] := N + 1$; {Indicates value undefined in this case}
$OUTO_T[j] := N + 1$; {Indicates value undefined in this case}
$OUTN_T[j] := 0$;
$j := j + 1$
od;
$j := N$;
while $j \geq 2$ **do** {Propagate values towards root}
$i := parent[j]$;

$$INBC_T[i] := \min \left\{ \begin{array}{l} INBC_T[i] + OUTC_T[j], \\ INBC_T[i] + OUTO_T[j], \\ INN_T[i] + OUTN_T[j] \end{array} \right\};$$

$INN_T[i] := INN_T[i] + \min\{OUTC_T[j], OUTO_T[j]\}$;

$$OUTC_T[i] := \min \left\{ \begin{array}{l} OUTC_T[i] + OUTC_T[j], \\ OUTN_T[i] + INN_T[j], \\ OUTC_T[i] + INBC_T[j], \\ OUTC_T[i] + INN_T[j], \\ OUTO_T[i] + INBC_T[j], \\ OUTO_T[i] + INN_T[j] \end{array} \right\};$$

$$OUTO_T[i] := \min \left\{ \begin{array}{l} OUTO_T[i] + OUTC_T[j], \\ OUTN_T[i] + INBC_T[j] \end{array} \right\};$$

$OUTN_T[i] := OUTN_T[i] + OUTC_T[j]$;
$j := j - 1$
od;
$\beta^-_{0,2} := \min\{INBC_T[1], INN_T[1], OUTC_T[1]\}$
]|

Figure 5.1: Algorithm to find the minimum 2-maximal independence number of a tree.

for planar cubic graphs is discussed in Section 4.2.6. Hence let $G = (V, E)$ (a planar cubic graph) and K (a positive integer) be an instance of MAXIMUM INDEPENDENT SET DECISION. Assume that $V = \{v_1, v_2, \ldots, v_n\}$. We construct an instance $G' = (V', E')$ (planar graph of maximum degree 3) and K' (positive integer) of MINIMUM 2-MAXIMAL INDEPENDENT SET DECISION. Corresponding to every vertex $v_i \in V$ $(1 \leq i \leq n)$, construct a *vertex component* C_i of G' as follows:

- Vertices $p_i, q_i, r_i, x_i^j, y_i^j, z_i^j$, for $1 \leq j \leq 4$.
- Edges
  - $\{x_i^4, x_i^1\}$ and $\{x_i^j, x_i^{j+1}\}$, for $1 \leq j \leq 3$.
  - $\{y_i^4, y_i^1\}$ and $\{y_i^j, y_i^{j+1}\}$, for $1 \leq j \leq 3$.
  - $\{z_i^4, z_i^1\}$ and $\{z_i^j, z_i^{j+1}\}$, for $1 \leq j \leq 3$.
  - $\{x_i^3, p_i\}, \{y_i^3, q_i\}, \{z_i^3, r_i\}$.
  - $\{p_i, q_i\}, \{p_i, r_i\}, \{q_i, r_i\}$.

We denote by V_i the vertices in C_i. To each of the vertices x_i^1, y_i^1 and z_i^1 in G', we attach one of the three edges incident on v_i in G. There is obviously a degree of freedom involved in making such attachments; however the actual choice of assignment does not affect the planarity of G', nor the remainder of the proof, as we shall see. One possible procedure for assigning edges to these vertices is now given.

Let $S_i = \{s_i^1, s_i^2, s_i^3\}$ start out as a set of three integers, for each i $(1 \leq i \leq n)$, such that $s_i^1 < s_i^2 < s_i^3$ and $\{v_i, v_{s_i^j}\} \in E$ for each j $(1 \leq j \leq 3)$. Define the *next free vertex of component* r $(1 \leq r \leq n)$ to be x_r^1 if $|S_r| = 3$, y_r^1 if $|S_r| = 2$, or z_r^1 if $|S_r| = 1$. For each S_i in turn, for $i = 1$ to n, perform the following:

- If $S_i = \{a, b, c\}$, connect x_i^1, y_i^1 and z_i^1 to the next free vertices of components a, b and c respectively. Delete i from S_a, S_b and S_c.
- If $S_i = \{b, c\}$, connect y_i^1 and z_i^1 to the next free vertices of components b and c respectively. Delete i from S_b and S_c.
- If $S_i = \{c\}$, connect z_i^1 to the next free vertex of component c. Delete i from S_c.
- If $S_i = \{\}$, do nothing at this step.

It is clear that the graph G' constructed is planar of maximum degree 3. A typical vertex component of G' is illustrated in Figure 5.2. Set $K' = 7n - K$. We now show that G has an independent set of cardinality at least K if and only if G' has a 2-maximal independent set D with cardinality at most K'.

For, suppose that I is an independent set for G, where $|I| = k \geq K$. We construct a set S as follows. For each i $(1 \leq i \leq n)$, if $v_i \in I$, add the vertices $x_i^1, x_i^3, y_i^1, y_i^3, z_i^1, z_i^3$ to S. If $v_i \notin I$, add the vertices $p_i, x_i^2, x_i^4, y_i^2, y_i^4, z_i^2, z_i^4$ to S.

S is independent in G', for if $\{s_i^1, t_j^1\} \in E'$, for any i, j $(1 \leq i \neq j \leq n)$, where s is x, y or z, and t is x, y or z, then $\{v_i, v_j\} \in E$. As I is independent in G then without

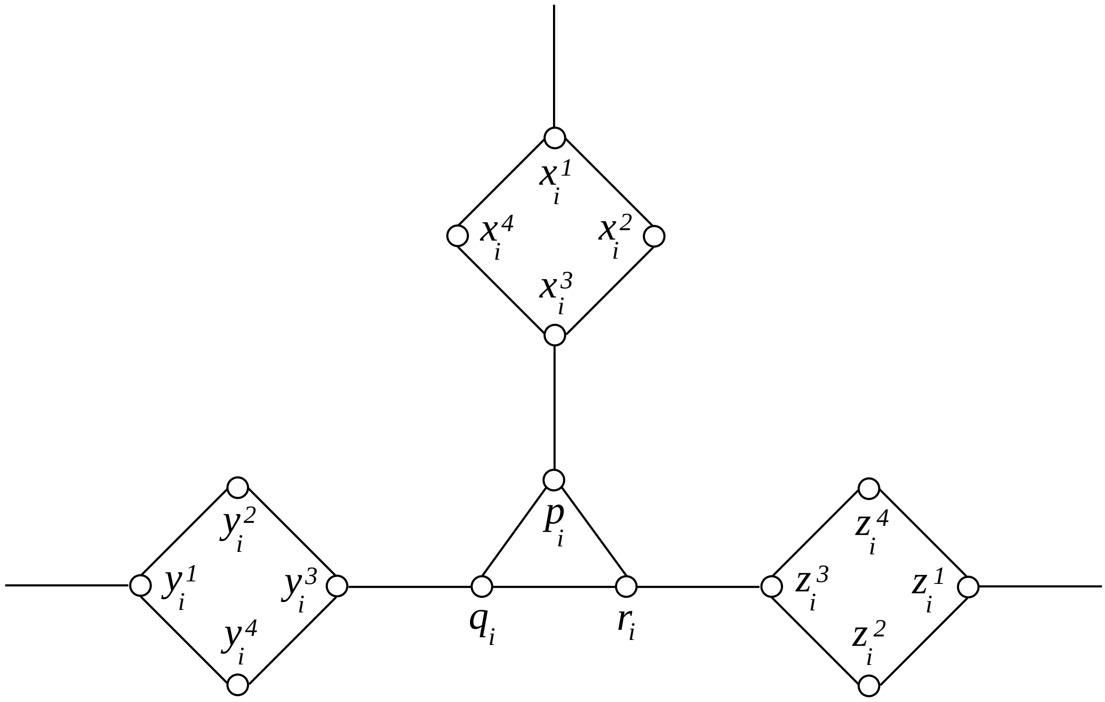

Figure 5.2: A typical vertex component from the constructed instance of MINIMUM 2-MAXIMAL INDEPENDENT SET DECISION.

loss of generality $v_i \notin I$, so that none of x_i^1, y_i^1 or z_i^1 is in S. Also, S is 2-maximal in G', for S is certainly maximal, and moreover, S does not admit an augmenting P_3 in G'. For, if $v_i \in I$ $(1 \leq i \leq n)$ then any P_3 in G' that augments s_i^1 or s_i^3 (where s is x, y or z) must include at least one of the vertices s_i^2, s_i^4, neither of which is available. If $v_i \notin I$ $(1 \leq i \leq n)$ then similarly any P_3 in G' that augments s_i^2 or s_i^4 (where s is x, y or z) must include at least one of the vertices s_i^1, s_i^3, neither of which is available. In addition, any P_3 in G' that augments p_i must include x_i^3, which is not available. Finally,

$$\begin{aligned} |S| &= 6k + 7(n-k) \\ &\leq 7n - K \\ &= K' \end{aligned}$$

as required.

Conversely suppose that S is a 2-maximal independent set for G', where $|S| \leq K'$. For a given i $(1 \leq i \leq n)$, we consider the elements of $S \cap V_i$. By the maximality of S, we see that the vertices s_i^2, s_i^4 must be dominated by vertices of V_i (where s is x, y or z). Since S is 2-maximal, we have that $s_i^1 \in S$ if and only if $s_i^3 \in S$, where s is x, y or z. Also, the maximality of S implies that $s_i^2 \in S$ if and only if $s_i^4 \in S$, where s is x, y or z. Thus $|S \cap V_i| \geq 6$.

It may be verified that $|S \cap V_i| = 6$ if and only if $S \cap V_i = W_i$, where

$$W_i = \{x_i^1, x_i^3, y_i^1, y_i^3, z_i^1, z_i^3\}.$$

Moreover, if $S \cap V_i \neq W_i$, then by the comments in the preceding paragraph, it is straightforward to check that $|S \cap V_i| = 7$. Define

$$I = \{v_i \in V : S \cap V_i = W_i\}.$$

We firstly claim that I is independent in G. For, if $\{v_i, v_j\} \in E$ for some i, j $(1 \leq i < j \leq n)$, then $\{s_i^1, t_j^1\} \in E'$, where s is x, y or z and t is x, y or z. If $v_i \in I$ then $x_i^1, y_i^1, z_i^1 \in S$. But S is independent in G', so that $t_j^1 \notin S$. Thus by construction of I, $v_j \notin I$ as required. Now let $k = |I|$ and suppose for a contradiction that $k < K$. Then

$$\begin{aligned} |S| &= 6k + 7(n-k) \\ &> 7n - K \\ &= K' \end{aligned}$$

which is a contradiction. Hence $k \geq K$ as required. ∎

5.3.6 Conclusions and further study relating to k-maximal independence

The complexity results for $\beta_{0,2}^-$ in trees and planar graphs presented here leave open the algorithmic complexity of $\beta_{0,2}^-$ in other classes of graphs, for example chordal graphs.

It is also interesting to consider the partial orders $\subset_k^G$ for $k > 2$, and the corresponding parameters $\beta_{0,k}^-$ for $k > 2$. We have already seen that, for $k = 1, 2$, the decision problem related to finding $\beta_{0,k}^-$ is NP-complete in bipartite and planar graphs, but polynomial-time solvable for trees, and we conjecture that this is the case for each fixed $k > 2$.

As a variation on the above hierarchy of parameters, consider the following. For a graph G and integer $k \leq \beta_0(G) - 1$, let $\beta_{0,\beta_0(G)-k}^-(G)$ denote the smallest order of a $(\beta_0(G)-k)$-maximal independent set of G. Now $\beta_0(G) - k \leq \beta_{0,\beta_0(G)-k}^-(G) \leq \beta_0(G)$ [169], so that $\beta_{0,\beta_0(G)}^-(G) = \beta_0(G)$ in the case $k = 0$. Thus finding $\beta_{0,\beta_0(G)}^-(G)$ is polynomial-time solvable for G a bipartite graph [106]. However, for general k, we conjecture that the associated decision problem is NP-complete for bipartite graphs.

We may also consider the parameters $\beta_{0,k}^-$ for line graphs $L(G)$ of general graphs G, for $k \geq 1$. The analagous parameter to $\beta_{0,k}^-$ in line graphs is $\beta_{1,k}^-$, the *minimum k-maximal matching* parameter. Thus $\beta_{1,k}^-(G) = \beta_{0,k}^-(L(G))$. For $k = 1, 2$, the decision problem related to finding $\beta_{1,k}^-$ in bipartite graphs is NP-complete [213, 165], and we conjecture that this is the case for each fixed $k > 2$. As above, we may consider, for a graph G and integer $k \leq \beta_1(G) - 1$, the parameter $\beta_{1,\beta_1(G)-k}^-(G)$. As before, $\beta_{1,\beta_1(G)}^-(G) = \beta_1(G)$, so finding $\beta_{1,\beta_1(G)}^-(G)$ is polynomial-time solvable for arbitrary graphs [70]. But for general k, we conjecture that the related decision problem is again NP-complete.

5.4 Summary of complexity results for graph parameters considered in this thesis

Table 5.1 summarises the complexity results for the decision problems associated with each of the parameters discussed in Chapters 3, 4 and 5. In a table entry, 'N' denotes

NP-completeness, 'P' denotes polynomial-time solvability and 'T' denotes the fact that the problem becomes trivially solvable in polynomial time when restricted to the particular graph class. References are indicated, where appropriate. The symbol '†' denotes the fact that either NP-completeness follows by restriction from another result in the same table row, or polynomial-time solvability follows by noting polynomial-time solvability from a class of graphs that contain the class in question. An asterisk indicates that the result is new and the proof is presented here for the first time, and a question mark indicates that the corresponding problem is open. When more than one reference is given within square brackets, the relevant complexity result is obtained from separate results to be found in each of the references concerned, with the most important one listed first. The classes of graphs dealt with in the table are of course far from exhaustive, but extending our attention beyond planar, bipartite and chordal graphs and trees would give rise to many additional open problems. A reminder of each of the parameter names, together with a page number where the parameter concerned is defined, is given in Table 5.2.

A number of table entries require further explanation. These are the complexity results corresponding to the parameters $\beta^-_{1,2}, \alpha_2, \Gamma', n_p$ and ir in planar graphs. The first reference indicated in the table entry for each of these parameters in the class of planar graphs contains a transformation which may be used to show NP-completeness for the parameter concerned in *bipartite graphs.* In each case, the transformation begins from X3C, defined in Section 3.4. However, by considering a restricted version of X3C, we may obtain NP-completeness for these parameters in planar bipartite graphs.

The restriction of X3C known as PLANAR EXACT COVER BY 3-SETS (PX3C) demands that the graph $G = (V, E)$, associated with an instance (A, C) of X3C, with vertex set $V = A \cup C$ and edge set $E = \{(a, c) : a \in c \in C\}$, is planar. PX3C is NP-complete [69], even if each element occurs in either two or three clauses. In the case of each of the parameters $\beta^-_{1,2}, \alpha_2, \Gamma', n_p, ir$, the transformations contained in [165, 117, 165, 68, 165] respectively, showing NP-completeness for the parameter concerned in bipartite graphs, preserves the planarity of this graph G. Moreover, in each case apart from Γ', the maximum degree of the graph constructed is 4, if we consider the case that each element occurs in either two or three clauses. Thus, by considering the same transformation in each case, but from PX3C (where each element occurs in two or three clauses) rather than from X3C, we obtain the following result.

Theorem 5.4.1 *Each of $\beta^-_{1,2}, \alpha_2, n_p, ir$ is NP-complete for planar bipartite graphs of maximum degree 4. Also, Γ' is NP-complete for planar bipartite graphs.*

Note that this observation also applies to many other transformations contained in McRae's PhD thesis [165]. Thus one may obtain a number of NP-completeness results for graph parameters in planar bipartite graphs. However, a drawback of using this method is that the maximum degree of the graph constructed in each case is at least 4. In order to prove NP-completeness for planar graphs of maximum degree 3, one has to resort to other methods. In Table 5.1, the references shown for an NP-completeness result for a given graph parameter in planar graphs corresponds to the reference containing the lowest degree complexity result that we are aware of. Furthermore, our NP-completeness results for

β_0^-, $\beta_{0,2}^-$, γ_t and β_{SS}^- in planar graphs show NP-completeness for planar graphs of maximum degree 3 (see Theorems 4.2.12, 5.3.4, 4.3.6 and 4.3.2 respectively).

5.5 Further minimaximal and maximinimal graph optimisation problems

There are many other minimaximal and maximinimal graph optimisation problems that can be studied, in addition to those that we have investigated in Chapters 3, 4 and 5. Here we catalogue some of those problems that have been introduced in the literature, but have not been studied extensively with regard to algorithmic complexity.

The obvious analogue of edge domination for irredundance is *edge irredundance*. The maximum edge irredundance and minimum maximal edge irredundance parameters (denoted IR' and ir' respectively) are both shown to be NP-complete for bipartite graphs in [165]. In addition, there are variants of the irredundance parameters ir and IR, according to the criterion that a set of vertices V' must satisfy – these are summarised in Table 5.3. Some complexity results for $ooir$, oir, $coir$, $OOIR$, OIR, $COIR$ are obtained, and others are surveyed, by McRae [165].

One could also study analogues of Γ for other forms of domination, for example *connected domination* [121], *perfect domination* [214] and *private domination* [116] to name only a few variants of the original domination problem. Each of these forms of domination is defined in Table 5.4. There is no distinct maximinimal version for the *efficient domination* problem [13] (a set of vertices D is an efficient dominating set if, for every $v \in V$, $|N[v] \cap D| = 1$), for, if a given graph G has an efficient dominating set S (and the problem of deciding whether this is the case is NP-complete), then $|S| = \gamma(G)$ [13].

Two, perhaps interesting, maximinimal parameters that do not appear to have been studied with respect to algorithmic complexity are the *maximum minimal vertex connectivity number* κ_0^+ and the *maximum minimal edge connectivity number* κ_1^+. The partial order here is that of set inclusion. The vertex and edge connectivity numbers of a graph G, κ_0, κ_1 respectively[2], are the minimum sizes of a *separating set* for G containing only vertices and edges respectively. A *separating set* for G is a subset of $V(G) \cup E(G)$ whose removal from G results in a disconnected graph or a single vertex. Both κ_0 and κ_1 are polynomial-time solvable for arbitrary graphs by Menger's theorem (see for example Harary [106, Chapter 5]). The parameters κ_0^+ and κ_1^+ were introduced by Alavi et al. [3], and are also studied by Peters et al. [183]. Both of these parameters are conjectured to be NP-complete in general by Hedetniemi [120].

One might also consider the *neighbourhood numbers* n_0, n_0^+ and *line neighbourhood numbers* $n_0', n_0'^+$, studied by Sampathkumar and Neeralagi [190, 191] and Kale and Deshpande [137]. These have received some initial attention [32] from an algorithmic point of

[2] These parameters have also been denoted κ and λ respectively, in the literature.

Graph class $\Rightarrow$	Arbitrary	Planar	Bipartite	Chordal	Tree
Parameter $\Downarrow$					
α_0	as for β_0				
α_0^+	as for β_0^-				
β_0^-	N(†)	N(*)	N[57]	P[73]	P[20]
$\beta_{0,2}^-$	N(†)	N(*)	N[165]	?	P(*)
β_0	N(†)	N[93]	P[106]	P[95]	P[62]
α_1	as for β_1				
α_1^+	as for γ				
β_1^-	as for γ'				
$\beta_{1,2}^-$	N(†)	N[165, 69]	N[165]	?	?
β_1	P[70]	P(†)	P(†)	P(†)	P(†)
β_{SS}^-	N(†)	N(*)	N[165]	N[165]	?
β_{SS}	N[124])	N[33, 126]	N[33]	N[33]	P[167, 46]
α_2	N[163]	N[117, 69]	N[117]	N[117]	P[163]
α_2^+	N(†)	N(*)	?	?	P[131, 95]
β_2^-	N(†)	N[98]	N[117, *]	N[117, *]	P[74]
β_2	as for β_1^-				
γ	N(†)	N[92]	N[19]	N[23]	P[46]
Γ	N[39]	N[77]	P[44, 106]	P[131, 95]	P[44, 62]
γ'	N(†)	N[213]	N[213]	?	P[168]
Γ'	N(†)	N[165, 69]	N[165]	?	?
γ_t	N(†)	N(*)	N[185]	N[152]	P[152]
Γ_t	N(†)	?	N[83]	?	P[83]
n_p	N(†)	N[68, 69]	N[68]	N[68]	P[68]
N_p	P[68]	P(†)	P(†)	P(†)	P(†)
ir	N(†)	N[165, 69]	N[185]	N[151]	P[18]
IR	N[77]	N[77]	P[44, 106]	P[131, 95]	P[44, 62]
ω^-	N(*)	P(T)	P(T)	?	P(T)
ω	N[92]	P(T)	P(T)	P[95]	P(T)
χ	N[140]	N[93]	P(T)	P[95]	P(†)
ψ	N[213]	N(†)	N[75]	N(†)	N[30]
φ	N(*)	?	N(*)	?	P(*)

Table 5.1: Complexity results for graph parameters considered in Chapters 3, 4 and 5, restricted to certain graph classes.

Graph Parameter	Name	Page
α_0	minimum vertex covering number	57
α_0^+	maximum minimal vertex covering number	57
β_0^- *or* i	minimum maximal independence number *or* minimum independent domination number	57
$\beta_{0,2}^-$	minimum 2-maximal independence number	84
β_0	maximum independence number	57
α_1	minimum edge covering number	57
α_1^+	maximum minimal edge covering number	57
β_1^-	minimum maximal matching number	57
$\beta_{1,2}^-$	minimum 2-maximal matching number	92
β_1	maximum matching number	57
β_{SS}^-	minimum maximal strong stable set number	70
β_{SS}	maximum strong stable set number	70
α_2	minimum total covering number	56
α_2^+	maximum minimal total covering number	56
β_2^-	minimum maximal total matching number	56
β_2	maximum total matching number	56
γ	minimum domination number	74
Γ	maximum minimal (upper) domination number	74
γ'	minimum edge domination number	78
Γ'	maximum minimal (upper) edge domination number	78
γ_t	minimum total domination number	75
Γ_t	maximum minimal total domination number	75
n_p	minimum 1-maximal nearly perfect set number	82
N_p	maximum 1-minimal nearly perfect set number	82
ir	minimum maximal (lower) irredundance number	79
IR	maximum (upper) irredundance number	79
ω^-	minimum maximal clique number	73
ω	maximum clique number	73
χ	chromatic number	33
ψ	achromatic number	34
φ	b-chromatic number	36

Table 5.2: Graph parameters considered in Chapters 3, 4 and 5: for each parameter, a short description, together with a page number where the parameter is defined, is given.

Criterion	Concept	Minimaximal parameter	Maximum parameter
$\forall\, v \in V' \bullet N(v) \backslash N(V' \backslash \{v\}) \neq \varnothing$	open-open irredundance	*ooir*	*OOIR*
$\forall\, v \in V' \bullet N(v) \backslash N[V' \backslash \{v\}] \neq \varnothing$	open irredundance	*oir*	*OIR*
$\forall\, v \in V' \bullet N[v] \backslash N(V' \backslash \{v\}) \neq \varnothing$	closed-open irredundance	*coir*	*COIR*
$\forall\, v \in V' \bullet N[v] \backslash N[V' \backslash \{v\}] \neq \varnothing$	irredundance	*ir*	*IR*

Table 5.3: Summary of variations of irredundance, and associated parameters.

Name	Extra criterion for dominating set D to satisfy
Connected domination	Subgraph induced by D is connected
Perfect domination	$\forall\, v \in V \backslash D \bullet |N(v) \cap D| = 1$
Private domination	$\forall\, v \in D \bullet v$ has a private D-neighbour

Table 5.4: Summary of variations of domination.

view. The neighbourhood numbers n_0, n_0^+ are respectively the minimum and maximum over all minimal (with respect to set inclusion) *neighbourhood sets*. A set S of vertices of a graph G is a neighbourhood set for G if $G = \cup_{v \in S} \langle N[v] \rangle$. The line neighbourhood numbers $n_0', n_0'^+$ are respectively the minimum and maximum over all minimal (with respect to set inclusion) *line neighbourhood sets*. A set T of edges of a graph G is a line neighbourhood set for G if $G = \cup_{\{v,w\} \in T} \langle N(v) \cup N(w) \rangle$.

Chapter 6

Minimaximal and maximinimal fractional graph optimisation problems

6.1 Introduction

This chapter is concerned with *minimaximal and maximinimal fractional graph optimisation problems*. In Section 6.2, we begin by defining a *fractional graph optimisation problem*, showing why Definition 2.2.1 must be adapted in order to define such problems. Solving a fractional graph optimisation problem involves computing the value of a fractional graph parameter (see Section 1.4.2 for a brief introduction to fractional graph parameters). By defining a partial order, called the *partial order on functions*, we give a framework for minimaximal and maximinimal fractional graph optimisation problems, following on from the definition of a fractional graph optimisation problem.

In Section 6.3, we define, in terms of our definition of a fractional graph optimisation problem, optimisation problems associated with a series of fractional graph parameters that have appeared in the literature. These parameters are connected with domination, packing, irredundance and vertex and edge covering and independence in graphs. We also give a brief literature survey of concepts relating to these fractional graph parameters.

In Section 6.4, we consider several examples of minimaximal and maximinimal fractional graph optimisation problems, again associated with the same graph-theoretic concepts that were detailed in the previous paragraph. A number of fractional graph parameters that involve maximality or minimality criteria have appeared in the literature. We show that, in each case, the maximality or minimality condition is equivalent to maximality or minimality with respect to the partial order on functions. Thus, our framework for minimaximal and maximinimal fractional graph optimisation problems may be used in order to define optimisation problems associated with such fractional graph parameters. We also show that, for each minimaximal and maximinimal fractional graph optimisation problem in Section 6.4, the optimal measure function is computable and has rational values. We investigate, in each case, the question of whether the optimal measure may be attained by some function of compact representation which satisfies the feasibility criteria for the minimaximal or maximinimal fractional graph optimisation problem concerned. The three issues mentioned in the two previous sentences have important implications for the algorithmic complexity of minimaximal and maximinimal fractional graph optimisation problems. For each minimaximal and maximinimal fractional graph optimisation problem studied in this chapter, we review complexity results in the literature.

Finally, we present some closing remarks to this chapter in Section 6.5.

6.2 Framework for minimaximal and maximinimal fractional graph optimisation problems

Recall from Section 4.3.3 the definition of the integer-valued graph parameter $\gamma(G)$, the minimum domination number of a graph G. The value of this parameter is equal to $m^*(G)$, when $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$ is defined for the MINIMUM DOMINATING SET problem, as in Section 4.3.3. In Section 1.4.2, we discussed a fractional version of γ, namely $\gamma_f(G)$[1], the minimum fractional domination number of a graph G. However it is not possible to define $\gamma_f(G)$ as the optimal measure function of some optimisation problem, for a given graph G, in terms of our current definition of an optimisation problem (Definition 2.2.1). This is because:

1. For any instance x, $\mathcal{U}(x)$ must be finite. Yet the set of all dominating functions for a given graph is not, in general, of a finite size.

2. For any instance x and any $y \in \mathcal{U}(x)$, $m(x, y)$ must be a natural number. Yet the weight of a dominating function will not, in general, be integral.

3. For any instance x, the optimal measure is defined to be the value

$$m^*(x) = \text{OPT}\{m(x, y) : y \in \mathcal{F}(x)\}.$$

   When $\mathcal{F}(x)$ is finite, it is clear that $m^*(x)$ exists. Also, if m is a computable function, then so is m^*. However, when $\mathcal{F}(x)$ is not finite, as is the case when $\mathcal{F}(x)$ is the set of all dominating functions for example, it is no longer clear that $m^*(x)$ exists[2], or that m^* is computable, even if m is.

In this section, we aim to show that our framework for minimaximal and maximinimal optimisation problems extends to such optimisation problems related to fractional graph parameters. However, in order to demonstrate this, it follows from the remarks above that we must provide an alternative definition of an optimisation problem from Definition 2.2.1 in order to model fractional graph optimisation problems. Such a definition is now given.

[1] The 'f' in $\gamma_f(G)$ refers to the fractional property, and is not connected with any particular dominating function f that may be defined. This applies to all fractional graph parameters introduced in this chapter, and also to the partial order $\prec_f^G$, to be defined.

[2] In fact it is possible to prove that, for a given graph G, the set of weights of all dominating functions for G does have a minimum value, which is attained by some dominating function f^* (see Section 6.3). However, this is not the case for *maximal irredundant functions*, for example (see Section 6.4.4).

Definition 6.2.1 A *fractional graph optimisation problem*, Π, is a tuple $\langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$, such that:

1. $\mathcal{I} \subseteq \{G : G \text{ is a graph}\}$.
2. $\mathcal{U}$ is a function that maps a graph $G = (V, E) \in \mathcal{I}$ into a non-empty set of functions that encode the *universal set of possible solutions* of G, such that:
   (a) $f \in \mathcal{U}(G)$ implies that $\text{dom}(f) \subseteq V \cup E$.
   (b) $f, g \in \mathcal{U}(G)$ implies that $\text{dom}(f) = \text{dom}(g)$.
   (c) $f \in \mathcal{U}(G)$ implies that $\text{ran}(f) \subseteq \mathbb{R}$.
3. π is a predicate such that, for any instance $G \in \mathcal{I}$ and any possible solution $f \in \mathcal{U}(G)$, $\pi(G, f)$ if and only if f is a *feasible solution* (we assume that at least one feasible solution of G exists). We denote the feasible solutions of a given instance $G \in \mathcal{I}$ by $\mathcal{F}(G)$, thus
$$\mathcal{F}(G) = \{f \in \mathcal{U}(G) : \pi(G, f)\}.$$
4. $\text{OPT} \in \{\inf, \sup\}$. ■

The definition of a fractional graph optimisation problem differs from Definition 2.2.1 in that the measure function m is not a member of the defining tuple in Definition 6.2.1. This is due to the fact that, given a fractional graph optimisation problem, $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$, a graph $G \in \mathcal{I}$, and a function $f \in \mathcal{F}(G)$, the measure of f always has the real-number value
$$m(G, f) = \sum_{s \in \text{dom}(f)} f(s).$$
Of course, for a given fractional graph optimisation problem defined in terms of Definition 6.2.1, the value $m^*(G)$ may be $-\infty$ or ∞ for a particular graph G. For full generality, this situation must be permitted, so that Definition 6.2.1 may cope with functions with unbounded range defined on a graph. However, fractional graph problems involving functions over unbounded ranges have received comparatively little attention in the literature. (See Section 2.4 in Chapter 2 of [114] for an introduction to dominating functions of a graph with range $\mathbb{R}, \mathbb{Q}$ and $\mathbb{Z}$.) In this chapter, we restrict attention to the more extensively studied case where $\mathcal{U}(G)$ satisfies

2. (d) $\displaystyle\bigcup_{f \in \mathcal{U}(G)} \text{ran}(f)$ is a bounded subset of $\mathbb{R}$.

Henceforth we therefore assume that a fractional graph optimisation problem satisfies the criteria in Definition 6.2.1 plus 2(d) above.

Note that, as G is finite and $\displaystyle\bigcup_{f \in \mathcal{F}(G)} \text{ran}(f)$ is a bounded subset of $\mathbb{R}$, then $m(G, \mathcal{F}(G))$ is a bounded subset of $\mathbb{R}$. Hence $m^*(G)$ exists, by the completeness axiom. Thus, the *evaluation version* of Π, Π_e, is well-defined according to Definition 2.2.3. Similarly, we may define the *decision version* of Π, Π_d, as in Definition 2.2.3. However, instead of adding a positive integer to the instance of Π, we add an arbitrary real number $r \in \mathbb{R}$, in order to give Π_d. If $\mathrm{ran}(m^*) \subseteq \mathbb{Q}$ then $r \in \mathbb{Q}$.

The *search version* of Π, Π_s, may not always be well-defined according to Definition 2.2.3. This is because, given $G \in \mathcal{I}$, it may not be possible to find some $f \in \mathcal{F}(G)$ such that $m(G,f) = m^*(G)$. An example of such an optimisation problem is MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE (see Section 6.4.4). We now give a definition which provides terminology for a necessary and sufficient condition for the search version of a fractional graph optimisation problem to be well-defined.

Definition 6.2.2 Let $\Pi = \langle I, \mathcal{U}, \pi, \mathrm{OPT}\rangle$ be a fractional graph optimisation problem. Π is *compact* if, for every $G \in \mathcal{I}$, there is some $f \in \mathcal{F}(G)$ such that $m(G,f) = m^*(G)$ (i.e., the optimal measure is attained by some feasible function). ■

In this chapter, the default version of a fractional graph optimisation problem Π is Π_s, provided Π is compact, or Π_e otherwise.

One might prove that a fractional graph optimisation problem is compact using the theory of compact metric and topological spaces [203] (hence the terminology *compact* optimisation problem), as follows. Suppose that $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \mathrm{OPT}\rangle$ is a fractional graph optimisation problem, let $G \in \mathcal{I}$ and let $f \in \mathcal{F}(G)$. Define $D = \mathrm{dom}(f)$, $n = |D|$ and $R = \bigcup_{f \in \mathcal{F}(G)} \mathrm{ran}(f)$. Consider the space $M = \mathbb{R}^n$, together with a suitable metric defined on M. The map

$$d : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$$

defined by

$$d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq n} |x_i - y_i|$$

where $\mathbf{x} = \{x_1, \ldots, x_n\}, \mathbf{y} = \{y_1, \ldots, y_n\} \in \mathbb{R}^n$, is one possible choice (it may easily be verifed that (M, d) is a metric space). Now suppose that each function $f \in \mathcal{F}(G)$ (recall that $f : D \longrightarrow R$) is represented by an n-tuple in M; denote by C the set of all such n-tuples. Clearly C is bounded (as R is). It is sufficient to prove that C is closed. In order to prove this, we might consider a sequence of points, $(\mathbf{x}_r) \in C$ (for $r \geq 1$), which converges to some $\mathbf{x} \in M$, and show that $\mathbf{x} \in C$. Thus C, as a closed, bounded subspace of $\mathbb{R}^n$, is compact [203, Theorem 5.7.1]. Now consider the map $w : C \longrightarrow \mathbb{R}$ (where $\mathbb{R}$ has the usual metric) defined by

$$w(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i.$$

As w is a sum of projections, each of which is continuous [203, Proposition 3.5.3], then w is itself continuous [203, Exercise 2.6.14]. Hence w is bounded on C [203, Corollary 5.5.3]. In addition, w attains its bounds on C [203, Corollary 5.5.4]. Since $w(x_1, \ldots, x_n)$ is the weight of the function represented by $(x_1, \ldots, x_n) \in \mathbb{R}^n$, then Π is compact.

The above method is utilised by Cheston et al. [39] to prove that the optimisation problem related to the fractional version of the maximum minimal domination number is compact. However, it is often easier to show that an optimisation problem is compact by considering appropriate linear programming constructions (see Theorem 6.4.4, for example) – such a method also establishes the computablity of m^*, demonstrates that m^* has rational values, and proves that m^* has a *compact representation*. This latter term corresponds to the property that there exists a polynomial p such that, for any graph G, there exists a function $f \in \mathcal{F}(G)$ such that $m(G,f) = m^*(G)$, f has rational values, and the length of the representation of f is bounded by $p(|G|)$. The notion of m^* having a compact representation clearly subsumes the concept of Π being compact.

Given a fractional graph optimisation $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$, the computability of m^*, the rationality of m^*, and m^* having a compact representation are three attributes of Π that are important when reasoning about the behaviour of Π from an algorithmic point of view. However, in order to be consistent with Definition 2.2.1, we have not included these attributes as standard properties of a fractional graph optimisation problem in Definition 6.2.1. We have not found an example in the literature of a fractional graph optimisation problem, associated with a fractional graph parameter, that does not have a computable m^* function, or has an m^* function taking non-rational values, However, as mentioned previously, MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE is an example of a non-compact fractional graph optimisation problem (see Section 6.4.4).

Our framework for minimaximal and maximinimal fractional graph optimisation problems follows naturally from Definition 6.2.1 in a manner similar to Definition 2.3.5. Before presenting this, we make the following definition, which introduces the partial order that is incorporated in the definitions of the minimaximal and maximinimal fractional graph optimisation problems in this chapter.

Definition 6.2.3 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$ be a fractional graph optimisation problem. Let $G \in \mathcal{I}$, and for any $f \in \mathcal{U}(G)$, let $D = \text{dom}(f)$. Define

$$\prec_f^G = \{(f', f'') \in \mathcal{F}(G) \times \mathcal{F}(G) : f' \neq f'' \wedge \forall\, x \in D \bullet f'(x) \leq f''(x)\}.$$

Then $\prec_f^G$ is the *partial order on functions*. ■

For any fractional graph optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$, it is clear that $\prec_f^G$ satisfies POMM with respect to Π, for any $G \in \mathcal{I}$. The definition of a minimaximal or maximinimal fractional graph optimisation problem is now given.

Definition 6.2.4 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$ be a fractional graph optimisation problem, called the *source* fractional graph optimisation problem. Then we may define a fractional graph optimisation problem $\Pi' = \langle \mathcal{I}', \mathcal{U}', \pi', \text{OPT}' \rangle$, where:

- $\mathcal{I}' = \mathcal{I}$
- $\mathcal{U}' = \mathcal{U}$
- $\pi' \Leftrightarrow \pi \wedge \sigma$, where $\begin{cases} \sigma(G,f) \Leftrightarrow f \text{ is } \prec_f^G\text{-minimal, if OPT=inf, } \textit{or} \\ \sigma(G,f) \Leftrightarrow f \text{ is } \prec_f^G\text{-maximal, if OPT=sup,} \\ \quad\text{for any } G \in \mathcal{I}' \text{ and } f \in \mathcal{F}(G). \end{cases}$

For $G \in \mathcal{I}$, we denote the feasible solutions of G by $\mathcal{F}'(G)$, where

$$\mathcal{F}'(G) = \{f \in \mathcal{U}'(G) : \pi'(G, f)\}.$$

- $\textsc{opt}' = \begin{cases} \sup, \text{ if } \textsc{opt}=\inf \\ \inf, \text{ if } \textsc{opt}=\sup. \end{cases}$

If $\textsc{opt}=\sup$, then Π' is a *minimaximal fractional graph optimisation problem*, and if $\textsc{opt}=\inf$, then Π' is a *maximinimal fractional graph optimisation problem.* ■

As in Definition 6.2.1, the measure function is not a member of the defining tuple for a minimaximal or maximinimal fractional graph optimisation problem. In the context of the above definition, the measure of a function $f \in \mathcal{F}'(G)$, for some $G \in \mathcal{I}'$, is always the weight of f.

6.3 Definitions and literature relating to fractional graph parameters

In this section, we define a series of fractional graph optimisation problems in terms of Definition 6.2.1. The problems that we consider relate to fractional graph parameters that have been studied in the literature. Attention seems to have focused mainly on fractional graph parameters concerning domination, packing, irredundance, and vertex and edge covering and independence. We consider fractional graph optimisation problems relating to each of these concepts.

Minimaximal and maximinimal versions of these fractional graph parameters have also appeared in the literature. We consider optimisation problems relating to these parameters in Section 6.4. In this section, we consider the maximum and minimum parameters, giving a brief literature survey.

We begin with some definitions. Define $H(v)$ to be the set of all edges of G incident on v, i.e.,

$$H(v) = \{\{v, w\} \in E : w \in N(v)\}.$$

A real-valued function $f : V \longrightarrow [0, 1]$ is:

1. *dominating* if $\forall\, v \in V \bullet f(N[v]) \geq 1$.
2. *total dominating* if $\forall\, v \in V \bullet f(N(v)) \geq 1$.
3. *packing* if $\forall\, v \in V \bullet f(N[v]) \leq 1$.
4. *irredundant* if $\forall\, v \in V \bullet f(v) > 0 \Rightarrow \exists\, u \in N[v] \bullet f(N[u]) = 1$.
5. *vertex covering* if $\forall \{u, v\} \in E \bullet f(u) + f(v) \geq 1$.
6. *vertex independent* if $\forall \{u, v\} \in E \bullet f(u) + f(v) \leq 1$.
7. *independent dominating* if f is both vertex independent and dominating.

A real-valued function $f : E \longrightarrow [0, 1]$ is:

8. *edge covering* if $\forall\, v \in V \bullet f(H(v)) \geq 1$.

9. *edge independent (matching)* if $\forall\, v \in V \bullet f(H(v)) \leq 1$.

It appears that the domination number was the first graph parameter to be generalised into a fractional version. Farber [74] introduced the concept when investigating conditions under which linear programming formulations of the domination problem have (0,1) integral solutions. However, fractional domination was first defined and studied by Hedetniemi et al. [118]. The concept of fractional independent domination is considered by Slater [198], whilst fractional irredundance was introduced by Domke et al. [64]. Fractional packing, and also fractional vertex and edge covering and independence were introduced independently by Domke et al. [64] and Grinstead and Slater [100].

By considering Definition 6.2.1, we may define a fractional graph optimisation problem, Π, for each of Concepts 1-9, as follows. In each case, $\mathcal{I} = \{G : G \text{ is a graph}\}$. For each of Concepts 1-7 above, we have $\mathcal{U}(G) = \{f : V \longrightarrow [0, 1]\}$, for $G = (V, E) \in \mathcal{I}$. For both of Concepts 8 and 9 above, we have $\mathcal{U}(G) = \{f : E \longrightarrow [0, 1]\}$, for $G = (V, E) \in \mathcal{I}$. The predicate π for each fractional graph parameter is given by the corresponding condition indicated in 1-9 above. Finally, for Concepts 1, 2, 5, 7 and 8, we have $\textsc{opt} = \inf$, and for Concepts 3, 4, 6, and 9, we have $\textsc{opt} = \sup$.

We refer to the fractional graph optimisation problems corresponding to Concepts 1-9 as MINIMUM FRACTIONAL DOMINATION, MINIMUM FRACTIONAL TOTAL DOMINATION, MAXIMUM FRACTIONAL PACKING, MAXIMUM FRACTIONAL IRREDUNDANCE, MINIMUM FRACTIONAL VERTEX COVER, MAXIMUM FRACTIONAL VERTEX INDEPENDENCE, MINIMUM FRACTIONAL INDEPENDENT DOMINATION, MINIMUM FRACTIONAL EDGE COVER and MAXIMUM FRACTIONAL MATCHING respectively.

For each of the fractional graph optimisation problems relating to Concepts 1-9, except for Concept 4 (irredundance), the optimal measure $m^*(G)$ may be viewed as the solution of a linear program, for a given arbitrary graph G. These results follow from the linear programs defined by Slater [198], when the constraints $x_i \in \{0, 1\}$ in his formulations are relaxed to the constraints $0 \leq x_i \leq 1$. Thus, in each case, it is immediate that m^* is computable, has rational values and has a compact representation.

In the case of irredundance, Fricke [82] has shown that

$$m^*(G) = \sup\{f(V) : f \text{ is an irredundant function for } G\} = IR(G)$$

for any graph $G = (V, E)$, which is a remarkable result. Thus, m^* is computable, has integral values and has a compact representation.

Let $\gamma_f, \gamma_{t,f}, \rho_f, IR_f, \alpha_{0,f}, \beta_{0,f}, i_f, \alpha_{1,f}, \beta_{1,f}$ be the symbols[3] denoting the optimal measure function m^* when considering the fractional graph optimisation problems relating to Concepts 1-9, respectively. By the comments above, the complexity of IR_f is identical to

[3] The maximum weight over all packing functions of a graph G, here denoted $\rho_f(G)$, is also denoted $P_f(G)$ by Domke et al. [64].

that of IR (reviewed in Section 4.3.6), and each of the other parameters may be computed in polynomial time: Khachiyan [143] was the first to prove that LINEAR PROGRAMMING is polynomial-time solvable; Karmarkar [139] provides an alternative (faster) algorithm.

In fact, the linear programs corresponding to the parameters γ_f and ρ_f are *dual* [94, Section 2.9], so that $\gamma_f(G) = \rho_f(G)$ for any graph G, as observed by Domke et al. [64]. The same is true for the parameters $\alpha_{0,f}$ and $\beta_{1,f}$, and also in the case of the parameters $\alpha_{1,f}$ and $\beta_{0,f}$, as observed by Slater [198]. Hence, $\alpha_{0,f}(G) = \beta_{1,f}(G)$ for any graph G, and $\alpha_{1,f}(G) = \beta_{0,f}(G)$ for any graph G with no isolated vertices.

Recall from Sections 4.2.6 and 4.2.7 that, for a graph $G = (V, E)$, where $n = |V|$, $\alpha_0(G)+\beta_0(G) = n$, and for a graph $G = (V, E)$ with no isolated vertices, where $n = |V|$, $\alpha_1(G) + \beta_1(G) = n$. Fractional versions of these Gallai type identities appear in [100]. Thus, for a graph $G = (V, E)$ where $n = |V|$,

$$\alpha_{0,f}(G) + \beta_{0,f}(G) = n$$

and for a graph $G = (V, E)$ with no isolated vertices, where $n = |V|$,

$$\alpha_{1,f}(G) + \beta_{1,f}(G) = n.$$

A final interesting result concerns fractional matching and states that, for any graph $G = (V, E)$, there exists a matching function $g : E \longrightarrow [0, 1]$, such that $g(E) = \beta_{1,f}(G)$, and $\mathrm{ran}(g) \subseteq \{0, \frac{1}{2}, 1\}$ [94, p.85][187].

6.4 Minimaximal and maximinimal fractional graph optimisation problems

In the previous section, we formulated fractional graph optimisation problems relating to minimum domination, minimum total and independent domination, minimum vertex and edge covering, maximum packing, maximum irredundance and maximum vertex and edge independence, using Definition 6.2.1. In this section, we introduce minimaximal and maximinimal fractional graph optimisation problems relating to these concepts, using Definition 6.2.4.

Each minimaximal or maximinimal fractional graph optimisation problem that we study in this section relates to a minimaximal or maximinimal fractional graph parameter that has been defined in the literature. This literature definition incorporates some notion of maximality or minimality with respect to a property P, such that the characteristic function of a maximal (respectively minimal) P-set is a maximal (respectively minimal) P-function. For example, the characteristic function of a maximal irredundant set is a maximal irredundant function [84, Lemma 1].

For some of the minimaximal and maximinimal fractional parameters defined in the literature, this notion of maximality or minimality is formulated using the partial order on functions (for example, maximal irredundant functions fall into this category [84]). For other minimaximal and maximinimal fractional parameters defined in the literature, this concept of maximality or minimality is given as a property not in terms of the partial order on functions (for example, Domke et al. [64] define a packing function $g : V \longrightarrow [0, 1]$ of a graph $G = (V, E)$ to be maximal if, for every $v \in V$ with $g(v) < 1$ there is some $u \in N[v]$ such that $g(N[u]) = 1$). For the minimaximal and maximinimal fractional graph parameters in the second category, we show in this section that each of these notions of maximality or minimality is equivalent to maximality or minimality with respect to the partial order on functions. Therefore, the framework of Definition 6.2.4 may be used to define minimaximal and maximinimal fractional graph optimisation problems relating to minimaximal and maximinimal fractional graph parameters in both categories.

We study minimaximal and maximinimal fractional graph optimisation problems relating to domination, total domination, packing, irredundance, and vertex and edge covering and independence separately, in the remainder of this section. In addition to defining the problem Π using Definition 6.2.4, we show that the optimal measure function m^* is computable and takes rational values in each case, and that m^* has a compact representation in each case apart from irredundance. In addition, we review complexity results relating to Π.

6.4.1 Minimal dominating functions

Minimal dominating functions are studied by Cheston et al. [39], who obtain the following result.

Proposition 6.4.1 (Cheston et al. [39]) *Let $G = (V, E)$ be a graph and let $f : V \longrightarrow [0, 1]$ be a dominating function. Then f is $\prec_f^G$-minimal if and only if*

$$\forall\, v \in V \bullet f(v) > 0 \Rightarrow \exists\, u \in N[v] \bullet f(N[u]) = 1. \tag{6.1}$$

Let MAXIMUM MINIMAL FRACTIONAL DOMINATION be the maximinimal version of MINIMUM FRACTIONAL DOMINATION, formulated using Definition 6.2.4. Denote by Γ_f the optimal measure function m^* for MAXIMUM MINIMAL FRACTIONAL DOMINATION. Cheston et al. [39] show, using appropriate linear programming constructions, that Γ_f is computable and has rational values. Also, using the theory of metric and topological spaces (as discussed on Page 101), they show that MAXIMUM MINIMAL FRACTIONAL DOMINATION is compact. This result, together with the linear programming constructions, yields the outcome that m^* has a compact representation.

The decision version of MAXIMUM MINIMAL FRACTIONAL DOMINATION is NP-complete for arbitrary graphs [39]. If G is a *strongly perfect graph*, then $\Gamma_f(G) = \beta_0(G) = \Gamma(G) = IR(G)$ [38]. The class of strongly perfect graphs includes trees, bipartite graphs and chordal graphs, over each of which β_0 is polynomial-time solvable (see Section 4.2.6).

6.4.2 Minimal total dominating functions

Minimal total dominating functions are considered by Fricke et al. [83], who prove the following result.

Proposition 6.4.2 (Fricke et al. [83]) *Let $G = (V, E)$ be a graph and let $f : V \longrightarrow [0, 1]$ be a total dominating function. Then f is $\prec_f^G$-minimal if and only if*

$$\forall\, v \in V \bullet f(v) > 0 \Rightarrow \exists\, u \in N(v) \bullet f(N(u)) = 1. \tag{6.2}$$

Let MAXIMUM MINIMAL FRACTIONAL TOTAL DOMINATION be the maximinimal version of MINIMUM FRACTIONAL TOTAL DOMINATION, formulated using Definition 6.2.4. Denote by $\Gamma_{t,f}$ the optimal measure function m^* for MAXIMUM MINIMAL FRACTIONAL TOTAL DOMINATION. Fricke et al. [83] show, also using appropriate linear programming constructions, that $\Gamma_{t,f}$ is computable, has rational values and has a compact representation.

The decision version of MAXIMUM MINIMAL FRACTIONAL TOTAL DOMINATION is NP-complete, even for bipartite graphs [83]. Fricke et al. [83] prove that, for a tree T, $\Gamma_{t,f}(T) = \Gamma_t(T)$. Γ_t is polynomial-time solvable for trees (see Section 4.3.4).

6.4.3 Maximal packing functions

The concept of maximal packing functions was introduced by Domke et al. [64]. Cockayne et al. [43] give a simpler criterion for a packing function to be maximal, and prove its equivalence to $\prec_f$-maximality.

Proposition 6.4.3 (Cockayne et al. [43]) *Let $G = (V, E)$ be a graph and let $f : V \longrightarrow [0, 1]$ be a packing function. Then f is $\prec_f^G$-maximal if and only if*

$$\forall\, v \in V \bullet \exists\, u \in N[v] \bullet f(N[u]) = 1. \tag{6.3}$$

Let MINIMUM MAXIMAL FRACTIONAL PACKING be the minimaximal version of MAXIMUM FRACTIONAL PACKING, formulated using Definition 6.2.4. Denote by p_f the optimal measure function m^* for MINIMUM MAXIMAL FRACTIONAL PACKING. We prove that p_f is computable, takes rational values and has a compact representation. The method of proof is similar to that employed by Fricke et al. [83] for MAXIMUM MINIMAL FRACTIONAL TOTAL DOMINATION.

Theorem 6.4.4 *The parameter p_f is computable, has rational values and has a compact representation.*

Proof: Let $\langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT}\rangle$ be defined for MAXIMUM FRACTIONAL PACKING, and denote by $\langle \mathcal{I}', \mathcal{U}', \pi', \text{OPT}'\rangle$ the components of MINIMUM MAXIMAL FRACTIONAL PACKING, formulated using Definition 6.2.4. Suppose that $G = (V, E) \in \mathcal{I}$ is given, so that $\mathcal{F}(G)$ denotes the set of all packing functions for G, and $\mathcal{F}'(G)$ denotes the set of all maximal packing functions for G. Assume that $V = \{v_1, \ldots, v_n\}$. Define

$$\mathcal{C} = \{S \subseteq V : N[S] = V\}$$

and, for some $S \in \mathcal{C}$, define

$$\mathcal{F}'_S(G) = \{f \in \mathcal{F}(G) : S \subseteq \{u \in V : f(N[u]) = 1\}\}.$$

We firstly show that $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$. For, let $S \in \mathcal{C}$ and suppose that $f \in \mathcal{F}'_S(G)$. For any $v \in V$, $v \in N[S]$ by definition of $\mathcal{C}$. Hence there is some $u \in N[v]$ such that $f(N[u]) = 1$. Thus $f \in \mathcal{F}'(G)$ by Proposition 6.4.3.

On the other hand, if $f \in \mathcal{F}'(G)$, then define

$$S_f = \{u \in V : f(N[u]) = 1\}.$$

Let $v \in V$. By maximality of f, $v \in N[S_f]$. Hence $N[S_f] = V$, so that $S_f \in \mathcal{C}$. Thus $f \in \mathcal{F}'_{S_f}(G)$.

For any $S \in \mathcal{C}$, consider the following linear program:

$$\text{minimise} \sum_{i=1}^{n} x_i \text{ subject to} \tag{6.4}$$

$$\sum_{v_j \in N[v_i]} x_j = 1, \quad v_i \in S \quad (1 \le i \le n) \tag{6.5}$$

$$\sum_{v_j \in N[v_i]} x_j \le 1, \quad v_i \in V \backslash S \quad (1 \le i \le n) \tag{6.6}$$

$$0 \le x_i \le 1, \quad (1 \le i \le n) \tag{6.7}$$

It is straightforward to verify that $x_1, \ldots, x_n$ satisfies Constraints 6.5-6.7 if and only if $f \in \mathcal{F}'_S(G)$, where $f : V \longrightarrow [0,1]$ and $f(v_i) = x_i$ $(1 \le i \le n)$.

Hence $\mathcal{F}'_S(G) \neq \emptyset$ if and only if the linear program defined by S and Constraints 6.4-6.7 has a solution. Moreover, if $\mathcal{F}'_S(G) \neq \emptyset$, then $\min\{f(V) : f \in \mathcal{F}'_S(G)\}$ exists and may be computed in polynomial time. But $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$, and $\mathcal{C}$ is finite. Hence

$$p_f(G) = \min\{\min\{f(V) : f \in \mathcal{F}'_S(G)\} : S \in \mathcal{C} \wedge \mathcal{F}'_S(G) \neq \emptyset\}.$$

Thus MINIMUM MAXIMAL FRACTIONAL PACKING is compact. Also p_f is computable, and must be rational, since each individual linear programming problem involves only rational constraints. Finally, p_f has a compact representation, since each individual problem is bounded by a polynomial in $|G|$, and linear programming is polynomial-time solvable. ■

Corollary 6.4.5 MINIMUM MAXIMAL FRACTIONAL PACKING DECISION *is in NP.*

The following inequalities, in part due to Domke et al. [64], relate p_f to some other graph parameters:

$$p_f(G) \le p_2(G) \le \rho(G) \le \rho_f(G) = \gamma_f(G) \le \gamma(G) \le \Gamma(G) \le \Gamma_f(G)$$

for an arbitrary graph G, where the parameters p_2, ρ are the integer-valued versions of p_f, ρ_f respectively, i.e. the minimum and maximum over all maximal 2-packings of G, or equivalently, the minimum and maximum over all maximal strong stable sets of G (see Section 4.3.1).

We know of no algorithmic results concerning p_f.

6.4.4 Maximal irredundant functions

Maximal irredundant functions were introduced by Domke et al. [64], and are also studied by Fricke et al. [84]. There is no straightforward criterion for an irredundant function to be maximal, in contrast to the functions studied in the previous sections. Indeed, the definition of maximality for an irredundant function is stated in terms of $\prec_f^G$-maximality. However, it turns out that an irredundant function *can* be tested for maximality in polynomial time: Fricke et al. [84] present an algorithm for performing this task.

Let MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE be the minimaximal version of MAXIMUM FRACTIONAL IRREDUNDANCE, formulated using Definition 6.2.4. In contrast to the minimaximal and maximinimal fractional graph optimisation problems of the previous sections, MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE is not compact: Fricke et al. [84] demonstrate the existence of a graph G which has maximal irredundant functions with weight arbitrarily close to some fixed c, but G has no maximal irredundant function of weight c. Denote by ir_f the optimal measure function m^* for MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE. Despite the non-compactness of this optimisation problem, Fricke et al. show that ir_f is computable and has rational values.

We know of no algorithmic results for the parameter ir_f. However, Fricke et al. [84] state that it is likely (though not known at the time of writing) that MINIMUM MAXIMAL FRACTIONAL IRREDUNDANCE DECISION is in NP.

6.4.5 Minimal vertex covering functions

The concept of a minimal vertex covering function was introduced by Domke et al. [64]. It would appear that their criterion for a vertex covering function to be minimal fails to take account of isolated vertices. Here, we present a slightly modified version, and prove that our criterion is equivalent to $\prec_f^G$-minimality.

Proposition 6.4.6 *Let $G = (V, E)$ be a graph and let $f : V \longrightarrow [0, 1]$ be a vertex covering function. Then f is $\prec_f^G$-minimal if and only if*

$$\forall\, v \in V \bullet f(v) > 0 \Rightarrow \exists\, u \in N(v) \bullet f(u) + f(v) = 1. \tag{6.8}$$

Proof: Let f be non-$\prec_f^G$-minimal, and suppose for a contradiction that f satisfies Criterion 6.8. As f is non-$\prec_f^G$-minimal, there exists some $f' \neq f$ such that f' is a vertex covering function and $f'(w) \leq f(w)$ for every $w \in V$. As $f' \neq f$ there is some $v \in V$ such that $0 \leq f'(v) < f(v)$. Then, since $f(v) > 0$, there is some $u \in N(v)$ such that $f(u)+f(v) = 1$. Hence $f'(u) + f'(v) < f(u) + f(v) = 1$, contradicting the fact that f' is a vertex covering function.

Conversely let f be $\prec_f^G$-minimal and suppose for a contradiction that f does not satisfy Criterion 6.8. Let $v \in V$ satisfy $f(v) > 0$. Clearly v is not an isolated vertex, or else the function $f' : V \longrightarrow [0, 1]$ defined by

$$f'(w) = \begin{cases} f(w), & w \neq v \\ 0, & w = v \end{cases}$$

satisfies $f'(w) \le f(w)$, for all $w \in V$, $f'(v) < f(v)$, and f' is a vertex covering function, contradicting the $\prec_f^G$-minimality of f.

As f does not satisfy Criterion 6.8, then $f(u) + f(v) > 1$, for all $u \in N(v)$, since f is a vertex covering function. Define

$$\varepsilon = \min\{f(u) + f(v) - 1 : u \in N(v)\}.$$

Then $\varepsilon > 0$. For any $u \in N(v)$,

$$\begin{aligned} f(v) - \varepsilon &\ge f(v) - (f(u) + f(v) - 1) \\ &= 1 - f(u) \\ &\ge 0. \end{aligned}$$

Hence we may define a function $f' : V \longrightarrow [0, 1]$ as follows:

$$f'(w) = \begin{cases} f(w), & w \ne v \\ f(w) - \varepsilon, & w = v. \end{cases}$$

Then $f'(w) \le f(w)$ for all $w \in V$, and $f'(v) < f(v)$. If u, w are distinct vertices of V, neither equal to v, where $\{u, w\} \in E$, then clearly $f'(u) + f'(w) \ge 1$ holds. Now suppose that $u \in N(v)$. Then

$$\begin{aligned} f'(u) + f'(v) &= f(u) + f(v) - \varepsilon \\ &\ge f(u) + f(v) - (f(u) + f(v) - 1) \\ &= 1. \end{aligned}$$

Hence f' is a vertex covering function, contradicting the $\prec_f^G$-minimality of f. ■

Let MAXIMUM MINIMAL FRACTIONAL VERTEX COVER be the maximinimal version of MINIMUM FRACTIONAL VERTEX COVER, formulated using Definition 6.2.4. Denote by $\alpha_{0,f}^+$ the optimal measure function m^* for MAXIMUM MINIMAL FRACTIONAL VERTEX COVER. We prove that $\alpha_{0,f}^+$ is computable, takes rational values and has a compact representation.

Theorem 6.4.7 *The parameter $\alpha_{0,f}^+$ is computable, has rational values and has a compact representation.*

Proof: Let $\langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$ be defined for MINIMUM FRACTIONAL VERTEX COVER, and denote by $\langle \mathcal{I}', \mathcal{U}', \pi', \text{OPT}' \rangle$ the components of MAXIMUM MINIMAL FRACTIONAL VERTEX COVER, formulated using Definition 6.2.4. Suppose that $G = (V, E) \in \mathcal{I}$ is given, so that $\mathcal{F}(G)$ denotes the set of all vertex covering functions for G, and $\mathcal{F}'(G)$ denotes the set of all minimal vertex covering functions for G. Assume that $V = \{v_1, \ldots, v_n\}$, and denote by I the set of isolated vertices of G. For a set of edges $S \subseteq E$, let $V(S)$ denote the vertices of S, i.e., $V(S) = \{v, w : \{v, w\} \in S\}$. Define

$$\mathcal{C} = \{S \subseteq E : N[V(S)] = V \backslash I\}$$

and, for some $S \in \mathcal{C}$, define

$$\mathcal{F}'_S(G) = \left\{ f \in \mathcal{F}(G) : \begin{array}{l} S \subseteq \{\{v, w\} \in E : f(v) + f(w) = 1\} \wedge \\ V \backslash V(S) \subseteq \{v \in V : f(v) = 0\} \end{array} \right\}.$$

We firstly show that $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$. For, let $S \in \mathcal{C}$ and suppose that $f \in \mathcal{F}'_S(G)$. For any $v \in V$, suppose that $f(v) > 0$. Then $v \in V(S)$, so that $f(u) + f(v) = 1$ for some $u \in N(v)$. Hence $f \in \mathcal{F}'(G)$ by Proposition 6.4.6.

On the other hand, if $f \in \mathcal{F}'(G)$, then define

$$S_f = \{\{v, w\} \in E : f(v) + f(w) = 1\}.$$

Let $u \in V \backslash I$. As u is not isolated, there is some $v \in N[u]$ such that $f(v) > 0$, by the vertex covering property. By minimality, there is some $w \in N(v)$ such that $f(v) + f(w) = 1$. Hence $\{v, w\} \in S_f$, $v \in V(S_f)$ and $u \in N[V(S_f)]$. Clearly also $N[V(S_f)] \subseteq V \backslash I$. Hence $N[V(S_f)] = V \backslash I$, so that $S_f \in \mathcal{C}$. Finally, $f \in \mathcal{F}'_{S_f}(G)$. For, if $v \in V$ satisfies $f(v) > 0$, then $v \in V(S_f)$ by minimality of f.

For any $S \in \mathcal{C}$, consider the following linear program:

$$\text{maximise} \sum_{i=1}^{n} x_i \text{ subject to} \tag{6.9}$$

$$0 \le x_i \le 1, \quad v_i \in V(S) \quad (1 \le i \le n) \tag{6.10}$$

$$x_i = 0, \quad v_i \in V \backslash V(S) \quad (1 \le i \le n) \tag{6.11}$$

$$x_i + x_j = 1, \quad \{v_i, v_j\} \in S \quad (1 \le i < j \le n) \tag{6.12}$$

$$x_i + x_j \ge 1, \quad \{v_i, v_j\} \in E \backslash S \quad (1 \le i < j \le n) \tag{6.13}$$

It is straightforward to verify that $x_1, \ldots, x_n$ satisfies Constraints 6.10-6.13 if and only if $f \in \mathcal{F}'_S(G)$, where $f : V \longrightarrow [0, 1]$ and $f(v_i) = x_i$ $(1 \le i \le n)$.

Hence $\mathcal{F}'_S(G) \ne \varnothing$ if and only if the linear program defined by S and Constraints 6.9-6.13 has a solution. Moreover, if $\mathcal{F}'_S(G) \ne \varnothing$, then $\max\{f(V) : f \in \mathcal{F}'_S(G)\}$ exists and may be computed in polynomial time. But $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$, and $\mathcal{C}$ is finite. Hence

$$\alpha^+_{0,f}(G) = \max\{\max\{f(V) : f \in \mathcal{F}'_S(G)\} : S \in \mathcal{C} \wedge \mathcal{F}'_S(G) \ne \varnothing\}.$$

Thus MAXIMUM MINIMAL FRACTIONAL VERTEX COVER is compact. Also $\alpha^+_{0,f}$ is computable, and must be rational, since each individual linear programming problem involves only rational constraints. Finally, $\alpha^+_{0,f}$ has a compact representation, since each individual problem is bounded by a polynomial in $|G|$, and linear programming is polynomial-time solvable. ∎

Corollary 6.4.8 MAXIMUM MINIMAL FRACTIONAL VERTEX COVER DECISION *is in NP.*

Note that in the proof of Theorem 6.4.7, the constraint that $S \in \mathcal{C}$ if and only if $N[V(S)] = V \backslash I$ is not required in order to prove that a member of $\mathcal{F}'_S(G)$ is a minimal vertex covering function. However, the contraint reduces the number of linear programs to be formulated.

We know of no results regarding the computational complexity of $\alpha^+_{0,f}$.

6.4.6 Maximal vertex independent functions

As in the case of minimal vertex covering functions, Domke et al. [64] introduce maximal vertex independent functions. However, again their definition overlooks isolated vertices. The criterion we state here is therefore slightly different from that of Domke et al. The proof that our criterion is equivalent to $\prec_f^G$-maximality is similar to that of Proposition 6.4.6.

Proposition 6.4.9 *Let $G = (V, E)$ be a graph and let $f : V \longrightarrow [0, 1]$ be a vertex independent function. Then f is $\prec_f^G$-maximal if and only if*

$$\forall\, v \in V \bullet f(v) < 1 \Rightarrow \exists\, u \in N(v) \bullet f(u) + f(v) = 1. \tag{6.14}$$

Proof: Let f be non-$\prec_f^G$-maximal, and suppose for a contradiction that f satisfies Criterion 6.14. As f is non-$\prec_f^G$-maximal, there exists some $f' \neq f$ such that f' is a vertex independent function and $f'(w) \geq f(w)$ for every $w \in V$. As $f' \neq f$ there is some $v \in V$ such that $f(v) < f'(v) \leq 1$. Then, since $f(v) < 1$, there is some $u \in N(v)$ such that $f(u)+f(v) = 1$. Hence $f'(u)+f'(v) > f(u)+f(v) = 1$, contradicting the fact that f' is a vertex independent function.

Conversely let f be $\prec_f^G$-maximal and suppose for a contradiction that f does not satisfy Criterion 6.14. Let $v \in V$ satisfy $f(v) < 1$. Clearly v is not an isolated vertex, or else the function $f' : V \longrightarrow [0, 1]$ defined by

$$f'(w) = \begin{cases} f(w), & w \neq v \\ 1, & w = v \end{cases}$$

satisfies $f'(w) \geq f(w)$, for all $w \in V$, $f'(v) > f(v)$, and f' is a vertex independent function, contradicting the $\prec_f^G$-maximality of f.

As f does not satisfy Criterion 6.14, then $f(u) + f(v) < 1$, for all $u \in N(v)$, since f is a vertex independent function. Define

$$\varepsilon = \min\{1 - f(u) + f(v) : u \in N(v)\}.$$

Then $\varepsilon > 0$. For any $u \in N(v)$,

$$\begin{aligned} f(v) + \varepsilon &\leq f(v) + (1 - f(u) - f(v)) \\ &= 1 - f(u) \\ &\leq 1. \end{aligned}$$

Hence we may define a function $f' : V \longrightarrow [0, 1]$ as follows:

$$f'(w) = \begin{cases} f(w), & w \neq v \\ f(w) + \varepsilon, & w = v. \end{cases}$$

Then $f'(w) \geq f(w)$ for all $w \in V$ and $f'(v) > f(v)$. If u, w are distinct vertices of V, neither equal to v, where $\{u, w\} \in E$, then clearly $f'(u) + f'(w) \leq 1$ holds. Now suppose that $u \in N(v)$. Then

$$\begin{aligned} f'(u) + f'(v) &= f(u) + f(v) + \varepsilon \\ &\leq f(u) + f(v) + (1 - f(u) - f(v)) \\ &= 1. \end{aligned}$$

Hence f' is a vertex independent function, contradicting the $\prec_f^G$-maximality of f. ∎

Let MINIMUM MAXIMAL FRACTIONAL VERTEX INDEPENDENCE be the minimaximal version of MAXIMUM FRACTIONAL VERTEX INDEPENDENCE, formulated using Definition 6.2.4. Denote by $\beta_{0,f}^-$ the optimal measure function m^* for MINIMUM MAXIMAL FRACTIONAL VERTEX INDEPENDENCE. Domke et al. [64] prove that $\alpha_{0,f}^+(G) + \beta_{0,f}^-(G) = n$, for any graph $G = (V, E)$, where $n = |V|$. The proof demonstrates how to obtain a maximal vertex independent function of minimum weight from a minimal vertex covering function of maximum weight, and vice versa. Hence, by Theorem 6.4.7, we obtain the following result.

Theorem 6.4.10 *The parameter $\beta_{0,f}^-$ is computable, has rational values and has a compact representation.*

Furthermore, the computational complexity of $\beta_{0,f}^-$ is identical to that of $\alpha_{0,f}^+$.

Despite the nice structure afforded by the Gallai type correspondence between $\alpha_{0,f}^+$ and $\beta_{0,f}^-$, the definition of Domke et al. for maximal vertex independent functions does not seem to meet with universal approval. Fricke et al. [84] claim that any maximal vertex independent function ought to be a minimal dominating function. However this is not the case with the definition of Domke et al. To see this, consider the graph $G = K_3$ with vertices x, y, z. It is clear that the function defined by $f(x) = f(y) = f(z) = \frac{1}{2}$ is a maximal vertex independent function. Although f is dominating, f is not minimal dominating; we may decrease the values $f(x)$, $f(y)$ and $f(z)$ to $\frac{1}{3}$, to obtain a function that dominates the graph. There does not, as yet, appear to be a definition of a maximal vertex independent function that meets the criterion of Fricke et al. [84].

6.4.7 Minimal fractional edge covering

Domke et al. [64] mention in passing the concept of minimal edge covering functions, but do not define a criterion for minimality. We present a criterion here, and prove its equivalence to $\prec_f^G$-minimality. In what follows, we assume that G contains no isolated vertices, since the concept of fractional edge covering is undefined for a graph with isolated vertices.

Proposition 6.4.11 *Let $G = (V, E)$ be a graph with no isolated vertices, and let $f : E \longrightarrow [0, 1]$ be an edge covering function. Then f is $\prec_f^G$-minimal if and only if*

$$\forall \{u, v\} \in E \bullet f(\{u, v\}) > 0 \Rightarrow f(H(u)) = 1 \vee f(H(v)) = 1. \tag{6.15}$$

Proof: Let f be non-$\prec_f^G$-minimal, and suppose for a contradiction that f satisfies Criterion 6.15. As f is non-$\prec_f^G$-minimal, there exists some $f' \neq f$ such that f' is an edge covering function and $f'(e) \leq f(e)$ for every $e \in E$. As $f' \neq f$ there is some $e_0 \in E$ such that $0 \leq f'(e_0) < f(e_0)$. Then, since $f(e_0) > 0$, without loss of generality $f(H(u_0)) = 1$, where $e_0 = \{u_0, v_0\}$. Hence $f'(H(u_0)) < f(H(u_0)) = 1$, contradicting the fact that f' is an edge covering function.

Conversely let f be $\prec_f^G$-minimal and suppose for a contradiction that f does not satisfy Criterion 6.15. Then there is some $e_0 = \{u_0, v_0\} \in E$ such that $f(e_0) > 0$, $f(H(u_0)) > 1$ and $f(H(v_0)) > 1$, since f is an edge covering function. Define

$$\varepsilon = \min\{f(e_0), f(H(u_0)) - 1, f(H(v_0)) - 1\}.$$

Then $\varepsilon > 0$. Hence we may define a function $f' : E \longrightarrow [0, 1]$ as follows:

$$f'(e) = \begin{cases} f(e), & e \neq e_0 \\ f(e) - \varepsilon, & e = e_0. \end{cases}$$

Clearly $f'(e_0) \geq 0$ as $\varepsilon \leq f(e_0)$. Also, $f'(e) \leq f(e)$ for all $e \in E$ and $f'(e_0) < f(e_0)$. If w is any vertex of V not equal to u_0 or v_0 then clearly $f'(H(w)) \geq 1$ still holds. Now suppose without loss of generality that w is u_0 or v_0. Then

$$\begin{aligned} f'(H(w)) &= f(H(w)) - \varepsilon \\ &\geq f(H(w)) - (f(H(w)) - 1) \\ &= 1. \end{aligned}$$

Hence f' is an edge covering function, contradicting the $\prec_f^G$-minimality of f. ■

Let MAXIMUM MINIMAL FRACTIONAL EDGE COVER be the maximinimal version of MINIMUM FRACTIONAL EDGE COVER, formulated using Definition 6.2.4. Denote by $\alpha_{1,f}^+$ the optimal measure function m^* for MAXIMUM MINIMAL FRACTIONAL EDGE COVER. We prove that $\alpha_{1,f}^+$ is computable, takes rational values and has a compact representation.

Theorem 6.4.12 *The parameter $\alpha_{1,f}^+$ is computable, has rational values and has a compact representation.*

Proof: Let $\langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT}\rangle$ be defined for MINIMUM FRACTIONAL EDGE COVER, and denote by $\langle \mathcal{I}', \mathcal{U}', \pi', \text{OPT}'\rangle$ the components of MAXIMUM MINIMAL FRACTIONAL EDGE COVER, formulated using Definition 6.2.4. Suppose that $G = (V, E) \in \mathcal{I}$ is given, so that $\mathcal{F}(G)$ denotes the set of all edge covering functions for G, and $\mathcal{F}'(G)$ denotes the set of all minimal edge covering functions for G. Assume that $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Define

$$\mathcal{C} = \{S \subseteq V : N[S] = V\}$$

and, for some $S \in \mathcal{C}$, define

$$\mathcal{F}'_S(G) = \left\{ f \in \mathcal{F}(G) : \begin{array}{l} \{\{u,v\} \in E : u \in V\backslash S \wedge v \in V\backslash S\} \subseteq \{e \in E : f(e) = 0\} \wedge \\ S \subseteq \{v \in V : f(H(v)) = 1\} \end{array} \right\}.$$

We firstly show that $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$. For, let $S \in \mathcal{C}$ and suppose that $f \in \mathcal{F}'_S(G)$. For any $\{u,v\} \in E$, suppose that $f(\{u,v\}) > 0$. Then without loss of generality, $u \in S$, so that $f(H(u)) = 1$. Hence $f \in \mathcal{F}'(G)$ by Proposition 6.4.11.

On the other hand, if $f \in \mathcal{F}'(G)$, then define

$$S_f = \{v \in V : f(H(v)) = 1\}.$$

Let $v \in V$. As v is not isolated, there is some $w \in N(v)$ such that $f(\{v,w\}) > 0$, by the edge covering property. By minimality, $f(H(v)) = 1$ or $f(H(w)) = 1$. Hence $v \in S_f$ or $w \in S_f$, so that $v \in N[S_f]$. Thus $N[S_f] = V$, so that $S_f \in \mathcal{C}$. Finally, $f \in \mathcal{F}'_{S_f}(G)$. For, if $e = \{u,v\} \in E$ satisfies $f(e) > 0$, then $u \in S_f$ or $v \in S_f$ by minimality of f.

For any $S \in \mathcal{C}$, consider the following linear program:

$$\text{maximise} \sum_{i=1}^{m} x_i \text{ subject to} \tag{6.16}$$

$$0 \le x_i \le 1, \quad \left\{ \begin{array}{l} e_i = \{u,v\} \\ \text{for some } u,v \in V, \\ \text{where } u \in S \text{ or } v \in S \end{array} \right\} \quad (1 \le i \le m) \tag{6.17}$$

$$x_i = 0, \quad \left\{ \begin{array}{l} e_i = \{u,v\} \\ \text{for some } u,v \in V, \\ \text{where } u \in V\backslash S \text{ and } v \in V\backslash S \end{array} \right\} \quad (1 \le i \le m) \tag{6.18}$$

$$\sum_{e_j \in H(v_i)} x_j = 1, \quad v_i \in S \quad (1 \le i \le n) \tag{6.19}$$

$$\sum_{e_j \in H(v_i)} x_j \ge 1, \quad v_i \in V\backslash S \quad (1 \le i \le n) \tag{6.20}$$

It is straightforward to verify that $x_1, \ldots, x_m$ satisfies Constraints 6.17-6.20 if and only if $f \in \mathcal{F}'_S(G)$, where $f : E \longrightarrow [0,1]$ and $f(e_i) = x_i$ $(1 \le i \le m)$.

Hence $\mathcal{F}'_S(G) \ne \varnothing$ if and only if the linear program defined by S and Constraints 6.16-6.20 has a solution. Moreover, if $\mathcal{F}'_S(G) \ne \varnothing$, then $\max\{f(V) : f \in \mathcal{F}'_S(G)\}$ exists and may be computed in polynomial time. But $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$, and $\mathcal{C}$ is finite. Hence

$$\alpha^+_{1,f}(G) = \max\{\max\{f(E) : f \in \mathcal{F}'_S(G)\} : S \in \mathcal{C} \wedge \mathcal{F}'_S(G) \ne \varnothing\}.$$

Thus MAXIMUM MINIMAL FRACTIONAL EDGE COVER is compact. Also $\alpha^+_{1,f}$ is computable, and must be rational, since each individual linear programming problem involves only rational constraints. Finally, $\alpha^+_{1,f}$ has a compact representation, since each individual problem is bounded by a polynomial in $|G|$, and linear programming is polynomial-time solvable. ■

Corollary 6.4.13 MAXIMUM MINIMAL FRACTIONAL EDGE COVER DECISION *is in NP.*

As for Theorem 6.4.7, note that in the proof of Theorem 6.4.12, the constraint that $S \in \mathcal{C}$ if and only if $N[S] = V$ is not required in order to prove that a member of $\mathcal{F}'_S(G)$ is a minimal edge covering function. However, the constraint similarly reduces the number of linear programs to be formulated.

The following inequalities relate $\alpha^+_{1,f}$ to some other graph parameters:

$$\beta^-_{0,f}(G) \le \beta^-_0(G) \le \beta_0(G) \le \beta_{0,f}(G) = \alpha_{1,f}(G) \le \alpha_1(G) \le \alpha^+_1(G) \le \alpha^+_{1,f}(G)$$

for an arbitrary graph G. We know of no results regarding the computational complexity of $\alpha^+_{1,f}$. However, recall from Section 4.2.7, that for a graph $G = (V, E)$ without isolated vertices, where $n = |V|$, we have $\gamma(G) + \alpha^+_1(G) = n$. Also, Slater [196] shows that

$$\gamma(G) + \beta^+_c(G) = n \tag{6.21}$$

which implies that $\alpha^+_1(G) = \beta^+_c(G)$, where $\beta^+_c(G)$ denotes the maximum weight over all *enclaveless* functions with range $\{0, 1\}$. In general, a function $f : V \longrightarrow [0, 1]$ is *enclaveless* if $f(N[v]) \le |N(v)|$ for every $v \in V$. Let $\beta^+_{c,f}$ denote the maximum weight over all enclaveless functions. Slater [198] shows that $\beta^+_{c,f}(G)$ may be formulated as a linear program with rational constraints. Hence $\beta^+_{c,f}$ is polynomial-time computable and takes rational values. Thus, if it were to follow that $\beta^+_{c,f}(G) = \alpha^+_{1,f}(G)$, then the parameter $\alpha^+_{1,f}$ would also be polynomial-time solvable. Some evidence for this may be obtained from the fact that the fractional version of Equation 6.21 holds, i.e., Slater [197] has shown that $\gamma_f(G) + \beta^+_{c,f}(G) = n$.

6.4.8 Maximal edge independent functions

Domke et al. [64] mention in passing the concept of maximal edge independent functions, but do not define a criterion for maximality. We present a criterion here, and prove its equivalence to $\prec^G_f$-maximality.

Proposition 6.4.14 *Let $G = (V, E)$ be a graph and let $f : E \longrightarrow [0, 1]$ be an edge independent function. Then f is $\prec^G_f$-maximal if and only if*

$$\forall \{u, v\} \in E \bullet f(H(u)) = 1 \vee f(H(v)) = 1. \tag{6.22}$$

Proof: Let f be non-$\prec^G_f$-maximal, and suppose for a contradiction that f satisfies Criterion 6.22. As f is non-$\prec^G_f$-maximal, there exists some $f' \neq f$ such that f' is an edge independent function and $f'(e) \ge f(e)$ for every $e \in E$. As $f' \neq f$ there is some $e_0 \in E$ such that $1 \ge f'(e_0) > f(e_0)$. Without loss of generality, $f(H(u_0)) = 1$, where $e_0 = \{u_0, v_0\}$. Hence $f'(H(u_0)) > f(H(u_0)) = 1$, contradicting the fact that f' is an edge independent function.

Conversely let f be $\prec^G_f$-maximal and suppose for a contradiction that f does not satisfy Criterion 6.22. Then there is some $e_0 = \{u_0, v_0\} \in E$ such that $f(H(u_0)) < 1$ and

$f(H(v_0)) < 1$, since f is an edge independent function. Define

$$\varepsilon = 1 - \min\{f(H(u_0)), f(H(v_0))\}.$$

Then $\varepsilon > 0$ and

$$\begin{aligned} f(e_0) + \varepsilon &\leq \min\{f(H(u_0)), f(H(v_0))\} + \varepsilon \\ &= 1. \end{aligned}$$

Hence we may define a function $f' : E \longrightarrow [0,1]$ as follows:

$$f'(e) = \begin{cases} f(e), & e \neq e_0 \\ f(e) + \varepsilon, & e = e_0. \end{cases}$$

Then $f'(e) \geq f(e)$ for all $e \in E$ and $f'(e_0) > f(e_0)$. If w is any vertex of V not equal to u_0 or v_0 then clearly $f'(H(w)) \leq 1$ still holds. Now suppose that w is u_0 or v_0. Then

$$\begin{aligned} f'(H(w)) &= f(H(w)) + \varepsilon \\ &\leq f(H(w)) + 1 - f(H(w)) \\ &= 1. \end{aligned}$$

Hence f' is an edge independent function, contradicting the $\prec_f^G$-maximality of f. ■

Let MINIMUM MAXIMAL FRACTIONAL MATCHING be the minimaximal version of MAXIMUM FRACTIONAL MATCHING, formulated using Definition 6.2.4. Denote by $\beta_{1,f}^-$ the optimal measure function m^* for MINIMUM MAXIMAL FRACTIONAL MATCHING. We prove that $\beta_{1,f}^-$ is computable, takes rational values and has a compact representation.

Theorem 6.4.15 *The parameter $\beta_{1,f}^-$ is computable, has rational values and has a compact representation.*

Proof: Let $\langle \mathcal{I}, \mathcal{U}, \pi, \text{OPT} \rangle$ be defined for MAXIMUM FRACTIONAL MATCHING, and denote by $\langle \mathcal{I}', \mathcal{U}', \pi', \text{OPT}' \rangle$ the components of MINIMUM MAXIMAL FRACTIONAL MATCHING, formulated using Definition 6.2.4. Suppose that $G = (V, E) \in \mathcal{I}$ is given, so that $\mathcal{F}(G)$ denotes the set of all matching functions for G, and $\mathcal{F}'(G)$ denotes the set of all maximal matching functions for G. Assume that $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Define

$$\mathcal{C} = \{S \subseteq V : S \text{ is a vertex cover for } E\}$$

and, for some $S \in \mathcal{C}$, define

$$\mathcal{F}'_S(G) = \{f \in \mathcal{F}(G) : S \subseteq \{v \in V : f(H(v)) = 1\}\}.$$

We firstly show that $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$. For, let $S \in \mathcal{C}$ and suppose that $f \in \mathcal{F}'_S(G)$. For any $\{v, w\} \in E$, without loss of generality $v \in S$, by definition of $\mathcal{C}$. Hence $f(H(v)) = 1$ so that $f \in \mathcal{F}'(G)$ by Proposition 6.4.14.

On the other hand, if $f \in \mathcal{F}'(G)$, then define

$$S_f = \{v \in V : f(H(v)) = 1\}.$$

Let $e = \{v, w\} \in E$. By maximality, $f(H(v)) = 1$ or $f(H(w)) = 1$. Hence S_f is a vertex cover for E, i.e., $S_f \in \mathcal{C}$. Thus $f \in \mathcal{F}'_{S_f}(G)$.

For any $S \in \mathcal{C}$, consider the following linear program:

$$\text{minimise} \sum_{i=1}^{m} x_i \text{ subject to} \tag{6.23}$$
$$0 \leq x_i \leq 1, \qquad (1 \leq i \leq m) \tag{6.24}$$
$$\sum_{e_j \in H(v_i)} x_j = 1, \quad v_i \in S \quad (1 \leq i \leq n) \tag{6.25}$$
$$\sum_{e_j \in H(v_i)} x_j \leq 1, \quad v_i \in V \backslash S \quad (1 \leq i \leq n) \tag{6.26}$$

It is straightforward to verify that $x_1, \ldots, x_m$ satisfies Constraints 6.24-6.26 if and only if $f \in \mathcal{F}'_S(G)$, where $f : E \longrightarrow [0, 1]$ and $f(e_i) = x_i$ $(1 \leq i \leq m)$.

Hence $\mathcal{F}'_S(G) \neq \varnothing$ if and only if the linear program defined by S and Constraints 6.23-6.26 has a solution. Moreover, if $\mathcal{F}'_S(G) \neq \varnothing$, then $\min\{f(V) : f \in \mathcal{F}'_S(G)\}$ exists and may be computed in polynomial time. But $\mathcal{F}'(G) = \bigcup_{S \in \mathcal{C}} \mathcal{F}'_S(G)$, and $\mathcal{C}$ is finite. Hence

$$\beta^-_{1,f}(G) = \min\{\min\{f(E) : f \in \mathcal{F}'_S(G)\} : S \in \mathcal{C} \wedge \mathcal{F}'_S(G) \neq \varnothing\}.$$

Thus MINIMUM MAXIMAL FRACTIONAL MATCHING is compact. Also $\beta^-_{1,f}$ is computable, and must be rational, since each individual linear programming problem involves only rational constraints. Finally, $\beta^-_{1,f}$ has a compact representation, since each individual problem is bounded by a polynomial in $|G|$, and linear programming is polynomial-time solvable. ■

Corollary 6.4.16 MINIMUM MAXIMAL FRACTIONAL MATCHING DECISION *is in NP.*

The following inequalities relate $\beta^-_{1,f}$ to some other graph parameters of this chapter:

$$\beta^-_{1,f}(G) \leq \beta^-_1(G) \leq \beta_1(G) \leq \beta_{1,f}(G) = \alpha_{0,f}(G) \leq \alpha_0(G) \leq \alpha^+_0(G) \leq \alpha^+_{0,f}(G)$$

for an arbitrary graph G.

We know of no results regarding the computational complexity of $\beta^-_{1,f}$.

6.5 Concluding remarks

In this chapter, we have studied fractional graph optimisation problems relating to the fractional graph parameters γ_f, $\gamma_{t,f}$, ρ_f, IR_f, $\alpha_{0,f}$, $\beta_{0,f}$, i_f, $\alpha_{1,f}$ and $\beta_{1,f}$, Recall from Section 1.4.2 that for each of the integer-valued parameters ϕ, where ϕ is γ, γ_t, ρ, IR, α_0, β_0, i, α_1 or β_1, the parameter $\phi(G)$ for a given graph $G = (V, E)$ may be expressed as the maximum or minimum weight over all functions $f : D \longrightarrow \{0, 1\}$, each satisfying some property π, where $D \subseteq V \cup E$. The definition of $\phi_f(G)$ may then be obtained by considering the maximum or minimum weight over all functions $f : D \longrightarrow [0, 1]$ satisfying π. However, it is also possible to generalise the definition of ϕ in an integer-valued sense: we may consider functions $f : D \longrightarrow Y$, each satisfying some predicate π_Y (where π_Y is related to π and depends on Y), where Y is some finite subset of $\mathbb{Z}$. Some particular cases of Y have received attention from an algorithmic point of view:

- *k-domination* [65]. Here $Y = \{0, 1, \ldots, k\}$ for some $k \geq 1$. A function $f : V \longrightarrow Y$ is *k-dominating* if, for every $v \in V$, $f(N[v]) \geq k$, for a given graph $G = (V, E)$. Similarly, *k-packing* and *k-irredundant* functions may be defined. Domke et al. [65] give a criterion, similar to (6.1), for a k-dominating function to be minimal:

$$\forall\, v \in V \bullet f(v) > 0 \Rightarrow \exists\, u \in N[v] \bullet f(N[u]) = k. \tag{6.27}$$

  It may be verified that Criterion 6.27 is equivalent to $\prec_f^G$-minimality. In addition, a criterion similar to (6.3) for the maximality of a k-packing function is given, which is equivalent to $\prec_f^G$-maximality. The maximality of a k-irredundant function is given in terms of $\prec_f^G$-maximality.

- *Minus domination* [67]. Here $Y = \{-1, 0, 1\}$. A function $f : V \longrightarrow Y$ is *minus dominating* if, for every $v \in V$, $f(N[v]) \geq 1$, for a given graph $G = (V, E)$. Dunbar et al. [67] give a criterion for a minus dominating function to be minimal:

$$\forall\, v \in V \bullet f(v) \geq 0 \Rightarrow \exists\, u \in N[v] \bullet f(N[u]) = 1. \tag{6.28}$$

  It may be verified that Criterion 6.28 is equivalent to $\prec_f^G$-minimality.

- *Signed domination* [112]. Here $Y = \{-1, 1\}$. A function $f : V \longrightarrow Y$ is *signed dominating* if, for every $v \in V$, $f(N[v]) \geq 1$, for a given graph $G = (V, E)$. Hattingh et al. [112] give a criterion for a signed dominating function to be minimal:

$$\forall\, v \in V \bullet f(v) = 1 \Rightarrow \exists\, u \in N[v] \bullet f(N[u]) \in \{1, 2\}. \tag{6.29}$$

  It may be verified that Criterion 6.29 is equivalent to $\prec_f^G$-minimality.

Thus, the partial order on functions may be used, together with the framework of Definition 2.3.5 (where the measure is now an integer rather than a natural number), in order to define the minimaximal and maximinimal optimisation problems suggested by the above list.

As is evident from the review of individual minimaximal and maximinimal fractional graph optimisation problems in Section 6.4, algorithmic results for these problems are scarce. Indeed, the transformation used to show NP-completeness for the decision version of MAXIMUM MINIMAL FRACTIONAL DOMINATION in arbitrary graphs [39] is quite sophisticated. Even the task of finding a graph G for which $\Gamma_f(G) > \Gamma(G)$ [39] is somewhat less than straightforward. Hence there are many open questions regarding the computational complexity of minimaximal and maximinimal fractional graph optimisation problems that we have defined in this chapter, and others that may be obtained using the framework of Definition 6.2.4.

However, for a given graph G, the technique of formulating a series of linear programs P_S (for each $S \in \mathcal{C}$, for some collection $\mathcal{C}$) for a fractional graph parameter $\xi_f(G)$ (as used in Theorem 6.4.15, for example), can yield a variety of useful results. As well as establishing that the parameter ξ_f is computable, takes rational values and has a compact representation, we may use certain properties of the constraint matrix A_S in each P_S to obtain complexity results, as in [83]. A $(0,1)$ matrix A is *balanced* [16] if A contains no submatrix that is the (edge-vertex) incidence matrix of an odd cycle. Equivalently, A is balanced if A contains no square submatrix of odd order whose rows and columns each sum to exactly two. Fulkerson et al. [85], prove that, for a balanced $(0,1)$ matrix A, the extreme points of the polyhedra defined by

$$P(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \geq \mathbf{1} \wedge \mathbf{x} \geq \mathbf{0}\} \tag{6.30}$$

and

$$Q(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{1} \wedge \mathbf{x} \geq \mathbf{0}\} \tag{6.31}$$

have (0,1) coordinates. By demonstrating that the linear program defined by P_S takes the form of (6.30) or (6.31) above, the theory establishes that the parameter ξ_f is optimised by some function f with range $\{0,1\}$, and hence $\xi_f = \xi$ (where ξ is the integer (0,1) form of ξ_f), under the conditions that the constraint matrix A_S is balanced. See [83] for further details.

A similar theory exists for constraint matrices that are *totally unimodular*. A matrix A is *totally unimodular* [94, p.67] if every square submatrix of A has determinant $-1, 0$ or 1[4]. Hoffman and Kruskal [125] prove that if a $(0,1)$ matrix A is totally unimodular then the extreme points of the polyhedron defined by

$$R(A) = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{1} \wedge \mathbf{x} \geq \mathbf{0}\} \tag{6.32}$$

have integral coordinates. Hence, if A is a nonsingular $(0,1)$ matrix, then A is totally unimodular implies that the extreme points of $R(A)$ have $(0,1)$ co-ordinates. Thus, as in the case of balanced matrices, by demonstrating that the linear program defined by P_S takes the form of (6.32), the theory establishes that the parameter ξ_f is optimised by some function f with range $\{0,1\}$, and hence $\xi_f = \xi$ (where ξ is the integer (0,1) form of ξ_f), under the conditions that the constraint matrix A_S is nonsingular and totally unimodular.

[4] A totally unimodular matrix is balanced, though the converse is not true.

Chapter 7

Minimaximal and maximinimal non graph-theoretic optimisation problems

7.1 Introduction

In this chapter, we study several minimaximal and maximinimal optimisation problems that relate to areas other than graph theory, according to the source optimisation problem classification of Garey and Johnson [92]. Following their subject categorisation, we study selected problems relating to Network Design, Sets and Partitions, Data Storage, Compression and Representation, Mathematical Programming, and Logic, in Sections 7.2, 7.3, 7.4, 7.5, 7.6 and 7.7 respectively. More specifically, we formulate minimaximal or maximinimal versions of the following problems (Garey and Johnson problem numbers in brackets): LONGEST PATH (ND29), 3D-MATCHING (SP1), MINIMUM TEST SET (SP6), BIN PACKING (SR1), KNAPSACK (MP9), MAXIMUM 2-SAT (LO5), ONE-IN-THREE 3SAT (LO4), LONGEST COMMON SUBSEQUENCE (SR10), SHORTEST COMMON SUPERSEQUENCE (SR8), LONGEST COMMON SUBSTRING (SR10) and SHORTEST COMMON SUPERSTRING (SR9). All minimaximal and maximinimal optimisation problems that we study in this chapter are new problems, with the exception of the minimaximal and maximinimal versions of LONGEST COMMON SUBSEQUENCE and SHORTEST COMMON SUPERSEQUENCE respectively.

As always, each minimaximal or maximinimal optimisation problem considered is obtained from a source optimisation problem Π via a partial order defined on the feasible solutions $\mathcal{F}(x)$ for a given instance x of Π, satisfying POMM with respect to Π. The format of our treatment of the problems in this chapter is as follows. For each source problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$, we define the components $\mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt}$. When defining $\mathcal{U}, \pi$ and m, we assume that $x \in \mathcal{I}$ has been given, and when defining π and m, we also assume that $y \in \mathcal{U}(x)$ and $y \in \mathcal{F}(x)$ have been given, respectively. We then survey complexity results relating to the source problem. The relevant partial order that will be used to define the minimaximal or maximinimal optimisation problem is then stated. After giving the name of the minimaximal or maximinimal optimisation problem Π', we survey existing results, if any, regarding the complexity of Π', or else present a new complexity result for Π'. The new results are NP-completeness proofs for minimaximal or maximinimal versions of the first seven problems in the above list, and a polynomial-time algorithm for a minimaximal version of LONGEST COMMON SUBSTRING.

The complexities of the optimisation problems that we consider in this chapter are summarised in Section 7.8.

7.2 Network design

In the following two sections, we consider two variants of a source optimisation problem concerned with finding maximum length paths in a weighted graph. The first version specifies the endpoint vertices for the path in the input, whilst the second version does not involve such a restriction.

7.2.1 Longest path between two specified vertices

Source problem: LONGEST PATH=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, where

- $\mathcal{I} = \left\{ \langle G = (V, E), l, s, t\rangle : \begin{array}{l} G \text{ is a graph } \wedge \\ s, t \in V \wedge l : E \longrightarrow \mathbb{Z}^+ \end{array} \right\}$
- $\mathcal{U}(x) = \bigcup_{r=1}^{n} \mathbf{seq}_r\{v_1, v_2, \ldots, v_n\}$, where $V = \{v_1, v_2, \ldots, v_n\}$
- $\pi(x, \langle v_{i_1} v_{i_2} \ldots v_{i_r}\rangle) \Leftrightarrow |\{i_1, i_2, \ldots, i_r\}| = r \wedge s = v_{i_1} \wedge t = v_{i_r} \wedge$ $\forall\, 1 \leq j \leq r-1 \bullet \{v_{i_j}, v_{i_{j+1}}\} \in E$
- $m(x, \langle v_{i_1} v_{i_2} \ldots v_{i_r}\rangle) = \sum_{j=1}^{r-1} l(\{v_{i_j}, v_{i_{j+1}}\})$
- OPT = max.

An element $y \in \mathcal{U}(x)$ such that $\pi(x, y)$ holds is called a *simple path* in G.

Complexity of source problem: LONGEST PATH DECISION is NP-complete, even if $l(e) = 1$, for every $e \in E$ [92, problem ND29].

Partial order: substring

Minimaximal problem name: SHORTEST $\lll$-MAXIMAL PATH.

Complexity of minimaximal problem: Given any instance x of LONGEST PATH, we have that $\lll^x = \varnothing$. For, if $p \in \mathcal{F}(x)$ is a simple path with endpoint vertices s and t, then it is clear that p cannot be extended at either end to give rise to a simple path q with endpoint vertices s and t. Thus *any* simple path between s and t is $\lll^x$-maximal, which implies that the shortest $\lll^x$-maximal simple path between s and t may be found using Dijkstra's algorithm, for instance (see, for example Aho et al. [1, p.207]). Hence SHORTEST $\lll$-MAXIMAL PATH is polynomial-time solvable.

Partial order: subsequence

Minimaximal problem name: SHORTEST $\ll$-MAXIMAL PATH.

Complexity of minimaximal problem: The complexity of SHORTEST $\ll$-MAXIMAL PATH is resolved by the following theorem. The proof involves a transformation from the NP-complete problem HAMILTONIAN PATH BETWEEN TWO VERTICES [92, problem GT39], which may be defined as follows:

> *Name:* HAMILTONIAN PATH BETWEEN TWO VERTICES.
> *Instance:* Graph $G = (V, E)$ and distinguished vertices $s, t \in V$.
> *Question:* Does G contain a Hamiltonian path (i.e., a simple path that includes all the vertices of V) with endpoint vertices s and t?

Theorem 7.2.1 SHORTEST $\ll$-MAXIMAL PATH DECISION *is NP-complete, even if* $l(e) = 1$*, for every* $e \in E$*.*

Proof: SHORTEST $\ll$-MAXIMAL PATH DECISION is in NP. For, a simple path $p = \langle x_1 \dots x_k \rangle$ from s to t in G is $\ll$-maximal if and only if there is no simple path in G from x_j to x_{j+1} $(1 \le j \le k-1)$ not passing through $\{x_j, x_{j+1}\}$ and not passing through x_i $(1 \le i \le k)$. Since testing for the existence of such a path for each j $(1 \le j \le k-1)$ may clearly be accomplished in polynomial time, membership of the problem in NP follows.

To show NP-hardness, we give a transformation from HAMILTONIAN PATH BETWEEN TWO VERTICES, defined above. Suppose we have an instance of HAMILTONIAN PATH BETWEEN TWO VERTICES: graph $G = (V, E)$ and distinguished vertices $s, t \in V$, where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. Without loss of generality, suppose that $s = v_1$ and $t = v_n$. We construct an instance x of SHORTEST $\ll$-MAXIMAL PATH DECISION as follows. Let

$$\begin{aligned} V' &= V \cup \{u\} \cup \{w_r : 1 \le r \le m\} \\ &\quad \cup \{p_i^j, q_i^j : 2 \le i \le n-1 \wedge 1 \le j \le 2n\} \end{aligned}$$

for new vertices u, w_r $(1 \le r \le m)$, p_i^j, q_i^j $(2 \le i \le n-1, 1 \le j \le 2n)$ and let

$$\begin{aligned} E' &= \{\{v_i, w_r\}, \{w_r, v_j\} : 1 \le i < j \le n \wedge 1 \le r \le m \wedge e_r = \{v_i, v_j\}\} \\ &\quad \cup \{\{t, u\}\} \\ &\quad \cup \{\{p_i^j, p_i^{j+1}\}, \{q_i^j, q_i^{j+1}\} : 2 \le i \le n-1 \wedge 1 \le j \le 2n-1\} \\ &\quad \cup \{\{t, p_i^1\}, \{u, q_i^1\}, \{p_i^{2n}, v_i\}, \{q_i^{2n}, v_i\} : 2 \le i \le n-1\}. \end{aligned}$$

Let G' be the graph $G' = (V', E')$. Define a measure function $l : E' \longrightarrow \mathbb{Z}^+$ by $l(e) = 1$, for every $e \in E'$. We claim that G has a Hamiltonian path between s and t if and only if G' has a $\ll^x$-maximal simple path between s and u, of length at most $2n - 1$.

For, suppose that G has a Hamiltonian path $\langle v_{i_1} v_{i_2} \dots v_{i_n} \rangle$, where $|\{i_1, i_2, \dots, i_n\}| = n$, $\{v_{i_r}, v_{i_{r+1}}\} \in E$ for each r $(1 \le r \le n-1)$, $v_{i_1} = v_1 = s$ and $v_{i_n} = v_n = t$. Let j_r

$(1 \leq r \leq n-1)$ be a sequence of integers such that $\{v_{i_r}, v_{i_{r+1}}\} = e_{j_r}$. Define a simple path p in G', where

$$p = \langle s w_{j_1} v_{i_2} w_{j_2} v_{i_3} w_{j_3} \dots v_{i_{n-1}} w_{j_{n-1}} t u \rangle.$$

Then p is a simple path from s to u in G' of length $2n-1$. Suppose that p is not $\ll^x$-maximal. Then there is some simple path q from s to u in G' such that $p \ll^x q$. Hence there is some vertex $v \in V'$ such that $\langle v \rangle \lll q$ but $\langle v \rangle \not\lll p$ as p, q are simple paths. Then $v = w_k$ for some k $(1 \leq k \leq m)$, or $v = p_i^j$ for some i, j $(2 \leq i \leq n-1, 1 \leq j \leq 2n)$ or $v = q_i^j$ for some i, j $(2 \leq i \leq n-1, 1 \leq j \leq 2n)$.

Suppose firstly that $v = w_k$ for some k $(1 \leq k \leq m)$. Then $k \neq j_r$ for any r $(1 \leq r \leq n-1)$. As p visits every vertex of V, then $e_k = \{v_{i_a}, v_{i_b}\}$ for some a, b $(1 \leq a < b \leq n)$. Now $\{v_{i_a}, w_k\}, \{w_k, v_{i_b}\}$ are the only edges incident to w_k in G'. As w_k is not an endvertex of q, and $\langle v_{i_a} v_{i_b} \rangle \lll p \ll q$, then $\langle v_{i_a} w_k v_{i_b} \rangle \lesseqqgtr q$. But this is impossible, since

$$\langle v_{i_a} w_{j_a} v_{i_{a+1}} \rangle \lesseqqgtr q$$

follows from $p \ll q$, and q must visit v_{i_a} at most once (recall that $k \neq j_a$).

Now suppose that $v = p_i^j$ for some i, j $(2 \leq i \leq n-1, 1 \leq j \leq 2n)$. As p visits every vertex of V, then $i = i_r$ for some r $(2 \leq r \leq n-1)$. Also, since the degree of p_i^j in G' is two, and p_i^j is not an endvertex of q, then either $\langle v_{i_r} p_i^{2n} \rangle \lesseqqgtr q$ or $\langle p_i^{2n} v_{i_r} \rangle \lesseqqgtr q$. But it is impossible for either of these two cases to hold, since

$$\langle v_{i_{r-1}} w_{j_{r-1}} v_{i_r} w_{j_r} v_{i_{r+1}} \rangle \lesseqqgtr q$$

follows from $p \ll q$, and q must visit v_{i_r} at most once. The case $v = q_i^j$ is similar.

Hence p is indeed $\ll^x$-maximal, as required.

Conversely, suppose that G' has a $\ll^x$-maximal simple path $p = \langle x_1 x_2 \dots x_k \rangle$ from s to u, where $k \leq 2n$. Then $\{x_r, x_{r+1}\} \in E'$ for each r $(1 \leq r \leq k-1)$, $x_1 = v_1 = s$ and $x_k = u$. Also, $\langle p_i^j \rangle \not\lll p$ for any i, j $(2 \leq i \leq n-1, 1 \leq j \leq 2n)$, for otherwise $\langle p_i^j \rangle \lll p$ for each j $(1 \leq j \leq 2n)$, and then p would be too long. Similarly $\langle q_i^j \rangle \not\lll p$ for any i, j $(2 \leq i \leq n-1, 1 \leq j \leq 2n)$.

Thus $x_{k-1} = v_n = t$. We claim that p visits every vertex of V. For, suppose not. Then there is some $v_i \in V$ $(2 \leq i \leq n-1)$ such that $\langle v_i \rangle \not\lll p$. Define a simple path q, where

$$q = \langle s x_2 x_3 \dots x_{k-2} t p_i^1 p_i^2 \dots p_i^{2n} v_i q_i^{2n} q_i^{2n-1} \dots q_i^1 u \rangle.$$

Then $p \ll^x q$, contradicting the $\ll^x$-maximality of p. Hence p does indeed visit every vertex of V, so that $k = 2n$, and $x_{2r+1} \in V$ for each r $(0 \leq r \leq n-1)$. Thus $\langle s x_3 x_5 \dots x_{2n-3} t \rangle$ is a Hamiltonian Path from s to t in G. ■

Thus LONGEST PATH is an example of an optimisation problem that admits two minimaximal optimisation problems of contrasting algorithmic complexity, when two partial orders, namely $\lesseqqgtr$ and $\ll$, are defined on the feasible solutions. However this example is perhaps not as significant as CHROMATIC NUMBER for trees, with the partial orders $\prec_a$ and $\prec_b$ (see Section 3.6), since the partial order $\lesseqqgtr^x$ is empty, for any instance x of LONGEST PATH.

7.2.2 Unconstrained longest path

We now consider an alternative version of the LONGEST PATH problem of the previous section. Define UNCONSTRAINED LONGEST PATH to be the optimisation problem with components similar to those of LONGEST PATH, except that an instance no longer contains distinguished vertices s, t, and now we seek the longest simple path over all pairs of vertices in the graph. It is straightforward to see that UNCONSTRAINED LONGEST PATH DECISION is NP-complete. This follows by considering the NP-complete problem HAMILTONIAN PATH [92, problem GT39], which may be defined as follows:

> *Name:* HAMILTONIAN PATH.
> *Instance:* Graph $G = (V, E)$.
> *Question:* Does G contain a Hamiltonian path?

Given a graph $G = (V, E)$, where $n = |V|$, as an instance of HAMILTONIAN PATH, we may define a measure function $l : E \longrightarrow \mathbb{Z}^+$ by $l(e) = 1$ for all $e \in E$. Then G has a Hamiltonian path if and only if G has a simple path of length at least $n - 1$.

Now consider the substring partial order defined on the feasible solutions of UNCONSTRAINED LONGEST PATH. We resolve the complexity of the associated UNCONSTRAINED SHORTEST $\overline{\ll}$-MAXIMAL PATH problem.

Theorem 7.2.2 UNCONSTRAINED SHORTEST $\overline{\ll}$-MAXIMAL PATH DECISION *is NP-complete, even if $l(e) = 1$, for every $e \in E$.*

Proof: Clearly UNCONSTRAINED SHORTEST $\overline{\ll}$-MAXIMAL PATH DECISION is in NP. For, given a simple path p of a graph $G = (V, E)$, it is straightforward to verify in polynomial time that $p +\!\!+ \langle v \rangle$ and $\langle v \rangle +\!\!+ p$ is not a simple path in G for any $v \in V$.

To show NP-hardness, we give a transformation from HAMILTONIAN PATH BETWEEN TWO VERTICES, defined on Page 123. Suppose we have an instance of HAMILTONIAN PATH BETWEEN TWO VERTICES: graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, and distinguished vertices $s, t \in V$. Without loss of generality, suppose that $s = v_1$ and $t = v_n$. We construct an instance of x UNCONSTRAINED SHORTEST $\overline{\ll}$-MAXIMAL PATH DECISION as follows. Let

$$V' = V \cup \{v_{n+1}, u\} \cup \{p_i^j : 2 \leq i \leq n+1, 1 \leq j \leq n+2\}$$

for new vertices v_{n+1}, u, p_i^j $(2 \leq i \leq n+1, 1 \leq j \leq n+2)$. Let

$$\begin{aligned} E' \;=\; & E \cup \{\{v_n, v_{n+1}\}, \{v_{n+1}, u\}\} \cup \{\{v_i, u\} : 2 \leq i \leq n\} \\ & \cup \{\{v_i, p_i^1\} : 2 \leq i \leq n+1\} \\ & \cup \{\{p_i^j, p_i^{j+1}\} : 2 \leq i \leq n+1, 1 \leq j \leq n+1\} \end{aligned}$$

and define G' to be the graph $G' = (V', E')$. Define a measure function $l : E' \longrightarrow \mathbb{Z}^+$ by $l(e) = 1$, for every $e \in E'$. We claim that G has a Hamiltonian path between s and t if and only if G' has a $\overline{\ll}^{\,x}$-maximal simple path, of length at most $n + 1$.

For, suppose that G has a Hamiltonian path $\langle v_{i_1}v_{i_2}\ldots v_{i_n}\rangle$, where $|\{i_1,i_2,\ldots,i_n\}|=n$, $\{v_{i_r},v_{i_{r+1}}\}\in E$ for each r $(1\le r\le n-1)$, $v_{i_1}=v_1=s$ and $v_{i_n}=v_n=t$. Then it may be verified that $\langle sv_{i_2}v_{i_3}\ldots v_{i_{n-1}}tv_{n+1}u\rangle$ is a $\overline{\lll}^{\,x}$-maximal simple path for G', of length $n+1$.

Conversely, suppose that G' has a $\overline{\lll}^{\,x}$-maximal simple path $p=\langle x_1x_2\ldots x_k\rangle$, where $k\le n+2$. Then $\{x_r,x_{r+1}\}\in E'$ for each r $(1\le r\le k-1)$. By $\overline{\lll}^{\,x}$-maximality, p has two endpoints from the set

$$\{s,u\}\cup\{p_i^{n+2}:2\le i\le n+1\}.$$

But if p has endpoint p_i^{n+2} for some i $(2\le i\le n+1)$, then $\langle p_i^j\rangle\lll p$ for each j $(1\le j\le n+2)$ and hence p is too long. Thus, without loss of generality, $x_1=v_1=s$ and $x_k=u$. Now p visits each vertex v_i $(2\le i\le n+1)$, for suppose not. Then there is some v_i $(2\le i\le n+1)$ such that $\langle v_i\rangle\not\lll p$. Define a simple path q, where

$$q=\langle sx_2x_3\ldots x_{k-1}uv_i\rangle.$$

Then $p\,\overline{\lll}^{\,x}q$, contradicting the $\overline{\lll}^{\,x}$-maximality of p. Thus p indeed visits each vertex v_i $(2\le i\le n+1)$, so that $k=n+2$. In particular, p visits v_{n+1}. But the only edges incident to v_{n+1} in G' are $\{p_{n+1}^1,v_{n+1}\}$, $\{t,v_{n+1}\}$ and $\{v_{n+1},u\}$. Now $\langle p_{n+1}^1\rangle\not\lll p$. Thus, as u is an endpoint vertex of p, then $\langle tv_{n+1}u\rangle\,\overline{\lll}\,p$, which implies that $x_{k-1}=v_{n+1}$. Thus $\langle sx_2x_3\ldots x_{k-3}t\rangle$ is a Hamiltonian path for G between s and t. ■

It may be verified that the above transformation can also be used to show that UNCONSTRAINED SHORTEST $\lll$-MAXIMAL PATH DECISION is NP-complete, where this problem incorporates the subsequence partial order. (The proof that the problem is in NP is similar to the argument for SHORTEST $\lll$-MAXIMAL PATH DECISION.)

7.3 Sets and partitions

7.3.1 3D-matching

Let W,X,Y be pairwise disjoint sets, each of size q, and let M be a subset of $W\times X\times Y$. A *3D-matching* for M is a subset M' of M, such that no two elements of M' agree in any coordinate. In this section, we consider a source optimisation problem concerned with finding maximum cardinality 3D-matchings of M.

Source problem: MAXIMUM 3D-MATCHING$=\langle\mathcal{I},\mathcal{U},\pi,m,\textsc{opt}\rangle$, where

- $\mathcal{I}=\left\{\langle W,X,Y,q,M\rangle:\begin{array}{l}W,X,Y\text{ are pairwise disjoint,}\\ \text{each of size }q\text{, and }M\subseteq W\times X\times Y\end{array}\right\}$
- $\mathcal{U}(x)=\mathbb{P}(M)$
- $\pi(x,M')\Leftrightarrow M'$ is a 3D-matching for M
- $m(x,M')=|M'|$

- OPT = max.

Complexity of source problem: MAXIMUM 3D-MATCHING DECISION is NP-complete, for if we restrict the target value to satisfy $K = q$ then we obtain [92, problem SP1].

Partial order: set inclusion

Minimaximal problem name: MINIMUM MAXIMAL 3D-MATCHING.

Complexity of minimaximal problem: The complexity of MINIMUM MAXIMAL 3D-MATCHING DECISION is resolved by the following theorem.

Theorem 7.3.1 MINIMUM MAXIMAL 3D-MATCHING DECISION *is NP-complete.*

Proof: Clearly MINIMUM MAXIMAL 3D-MATCHING DECISION is in NP. For, given $K \in \mathbb{Z}^+$ and a set S of at most K vertices, it is straightforward to verify in polynomial time that S is a maximal 3D-matching.

To show NP-hardness, we give a transformation from MINIMUM MAXIMAL MATCHING DECISION for bipartite graphs with no isolated vertices [92, problem GT10]. Hence let $G = (V, E)$ (bipartite graph with no isolated vertices) and K (positive integer) be an instance of MINIMUM MAXIMAL MATCHING DECISION. Suppose that $E = \{e_1, e_2, \ldots, e_m\}$, and G has bipartition $V = V_1 \cup V_2$, where $n_1 = |V_1|$ and $n_2 = |V_2|$. As G has no isolated vertices, then $m \geq \max\{n_1, n_2\}$. We construct an instance of MINIMUM MAXIMAL 3D-MATCHING DECISION as follows. Let

$$W = V_1 \cup \{w_1, w_2, \ldots, w_{m-n_1}\},$$

$$X = V_2 \cup \{x_1, x_2, \ldots, x_{m-n_2}\} \text{ and}$$

$$Y = \{y_1, y_2, \ldots, y_m\},$$

where the w_i, x_i and y_i are new vertex names. Then $|W| = |X| = |Y|$. Set

$$M = \{(v_1, v_2, y_i) : v_1 \in V_1 \wedge v_2 \in V_2 \wedge e_i = \{v_1, v_2\}\}.$$

The transformation is clearly polynomial, and we claim that G has a maximal matching of size at most K if and only if M has a maximal 3D-matching of size at most K.

For, suppose that M has a maximal 3D-matching M', where $|M'| \leq K$. Set

$$E' = \{\{v_1, v_2\} : v_1 \in V_1 \wedge v_2 \in V_2 \wedge (v_1, v_2, y_i) \in M' \text{ for some } y_i \ (1 \leq i \leq m)\}.$$

Then E' is a maximal matching for G, and $|E'| = |M'|$.

Conversely, suppose that G has a maximal matching E', where $|E'| \leq K$. Set

$$M' = \{(v_1, v_2, y_i) : v_1 \in V_1 \wedge v_2 \in V_2 \wedge \{v_1, v_2\} = e_i \text{ for some } i \ (1 \leq i \leq m)\}.$$

Then M' is a maximal 3D-matching for M, and $|M'| = |E'|$. ■

7.3.2 Test set

Let C be a collection of subsets of a finite set S. A *test set* for S is a subset C' of C such that, for every pair of distinct elements $u, v \in S$, there is some $c \in C'$ such that $|\{u, v\} \cap c| = 1$. In this section, we consider a source optimisation problem whose objective is to find a minimum cardinality test set for S.

Source problem: MINIMUM TEST SET=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle S, C \rangle : C \subseteq \mathbb{P}(S)\}$
- $\mathcal{U}(x) = \mathbb{P}(C)$
- $\pi(x, C') \Leftrightarrow C'$ is a test set for S
- $m(x, C') = |C'|$
- $\text{OPT} = \min$.

Complexity of source problem: MINIMUM TEST SET DECISION is NP-complete [92, problem SP6].

Partial order: set inclusion

Maximinimal problem name: MAXIMUM MINIMAL TEST SET.

Complexity of maximinimal problem: In order to resolve the complexity of MAXIMUM MINIMAL TEST SET, we introduce a further optimisation problem.

Let $G = (V, E)$ be a graph. A *restricted edge total dominating set* of G is a subset E' of E such that, for every $e \in E$, there is some $e' \in E'$ such that e and e' are adjacent as edges in G, and in addition, E' covers at least $|V| - 1$ vertices of V. A *minimal restricted edge total dominating set* E' of G is a restricted edge total dominating set of G such that no proper subset of E' is a restricted edge total dominating set of G. We now define the MAXIMUM MINIMAL RESTRICTED EDGE TOTAL DOMINATING SET DECISION (MMRETDSD) problem.

Name: MMRETDSD.
Instance: Graph G, positive integer K.
Question: Does G have a minimal restricted edge total dominating set S, with $|S| \geq K$?

It turns out that restricted edge total dominating sets are related to test sets.

Proposition 7.3.2 *Let $G = (V, E)$ be a graph and put $S = V$ and $C = E$. Then a restricted edge total dominating set for G is a test set for S, and vice versa.*

Proof: Suppose that G has a restricted edge total dominating set E'. Let u, v be two distinct members of $S = V$. If $\{u, v\} \in E$ then there is some $c \in E'$ such that $|\{u, v\} \cap c| = 1$, as E' is edge total dominating. If $\{u, v\} \notin E$ then there is some $c \in E'$ such that $|\{u, v\} \cap c| = 1$, since E' covers at least $|V| - 1$ vertices of V. Hence $C' = E'$ is a test set for S.

Conversely suppose that C' is a test set for S. Let $\{u, v\} \in E$. Then there is some $c \in C'$ such that $|\{u, v\} \cap c| = 1$, so that $E' = C'$ is an edge total dominating set for G. Now suppose that E' does not cover at least $|V| - 1$ vertices of V. Let u, v be two distinct vertices not covered by E'. Then $|\{u, v\} \cap c| = 0$ for all $c \in E'$, contradicting the fact that C' is a test set for S. Thus E' is a restricted edge total dominating set for G. ∎

The following corollary of the above theorem is then easily established.

Corollary 7.3.3 *Let $G = (V, E)$ be a graph and put $S = V$ and $C = E$. Then a minimal restricted edge total dominating set for G is a minimal test set for S, and vice versa.*

It is clear that a graph G has a restricted edge total dominating set if and only if G contains at most one isolated vertex. We use this observation when proving the following result, which resolves the complexity of MMRETDSD.

Theorem 7.3.4 MMRETDSD *is NP-complete.*

Proof. Clearly MMRETDSD is in NP. For, given a graph G, an integer $K \in \mathbb{Z}^+$, and a set S of at least K elements, it is straightforward to verify in polynomial time that S is an edge total dominating set, covering all but possibly one of the vertices of G.

To show NP-hardness, we give a transformation from X3C, defined in Section 3.4. Given an arbitrary instance of X3C, we construct a graph G, with the property that there exists an exact cover for the X3C instance if and only if there exists a minimal restricted edge total dominating set of G with at least K edges, for a particular $K \in \mathbb{Z}^+$.

Suppose that a set of elements $A = \{a_1, a_2, a_3, \ldots, a_{3q}\}$ and a collection of clauses $C = \{c_1, c_2, c_3, \ldots, c_m\}$ (for some q and m) is an arbitrary instance of X3C. Suppose further that, for each j $(1 \le j \le m)$, $c_j = \{a_{i_{3j-2}}, a_{i_{3j-1}}, a_{i_{3j}}\}$, where $i_1, i_2, i_3, \ldots, i_{3m}$ is some sequence of integers, such that

$$\{i_1, i_2, i_3, \ldots, i_{3m}\} = \{1, 2, 3, \ldots, 3q\}.$$

Construct an instance – graph $G = (V, E)$ and positive integer K – of MMRETDSD as follows:

- *Element vertices*: For each element a_i in the X3C instance $(1 \le i \le 3q)$, create a path on two vertices, namely x_i and y_i.
- *Subset components*: For each subset c_j in the X3C instance $(1 \le j \le m)$, create a path on four vertices, namely t_j, u_j, v_j, w_j, connected in that order.
- *Communication edges*: For each j $(1 \le j \le m)$, join w_j to $x_{i_{3j-2}}$, $x_{i_{3j-1}}$ and $x_{i_{3j}}$.
- *Isolated vertex*: Add a single isolated vertex z.
- *Target value*: Set $K = 3m + 5q$.

Denote by S_j the edges in the jth subset component, i.e.

$$S_j = \{\{t_j, u_j\}, \{u_j, v_j\}, \{v_j, w_j\}\}.$$

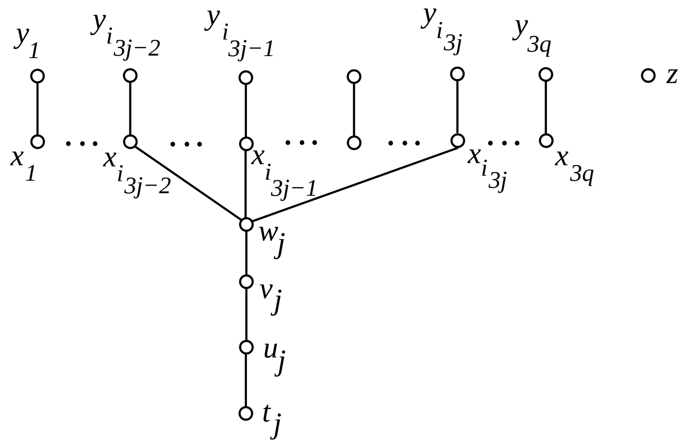

Figure 7.1: Part of the graph G constructed as an instance of MMRETDSD, showing a typical subset component.

The construction is partly illustrated in Figure 7.1. Clearly, this construction is polynomial with respect to the size of the X3C instance. First we show that if the X3C instance has an exact cover, then G has a minimal restricted edge total dominating set S, with $|S| = K$. From an exact cover C' for the X3C instance, we construct a set S as follows. For each j $(1 \le j \le m)$:

- Add to S the edges $\{t_j, u_j\}$ and $\{u_j, v_j\}$ from each subset component path.
- If $c_j \in C'$, add to S the three edges $\{w_j, x_{i_{3j-2}}\}$, $\{w_j, x_{i_{3j-1}}\}$ and $\{w_j, x_{i_{3j}}\}$.
- If $c_j \notin C'$, add to S the edge $\{v_j, w_j\}$.

For each i $(1 \le i \le 3q)$, add to S the edge $\{x_i, y_i\}$.

Clearly S covers every vertex of G except z. It may also be verified that S is an edge total dominating set, and hence a restricted edge total dominating set. Further, it may be verified that S is minimal with respect to this property, and that

$$\begin{aligned} |S| &= 2m + 3q + (m - q) + 3q \\ &= K \end{aligned}$$

as required.

Conversely, suppose that G has a minimal restricted edge total dominating set S of size at least K. We show that the X3C instance has an exact cover. From all minimal restricted edge total dominating sets for G with cardinality at least K, choose S to be such a set that has the fewest number of communication edges.

Since z is isolated, S does not cover z. Thus S covers every other vertex of G. Hence $\{x_i, y_i\} \in S$ for each i $(1 \le i \le 3q)$ to cover y_i, and $\{t_j, u_j\} \in S$ for each j $(1 \le j \le m)$ to cover t_j. Also, $\{u_j, v_j\} \in S$ for each j $(1 \le j \le m)$ to dominate $\{t_j, u_j\}$. Furthermore, the following facts may be established:

1. *If w_j $(1 \le j \le m)$ is incident to some communication edge of S, then $|S \cap S_j| = 2$.* For, $\{x_i, y_i\} \in S$ for each i $(1 \le i \le 3q)$, so each communication edge is dominated by one such edge $\{x_i, y_i\}$. Hence, by minimality, $\{v_j, w_j\} \notin S$.

2. *If w_j $(1 \le j \le m)$ is incident to no communication edge of S, then $|S \cap S_j| = 3$.* For, $\{v_j, w_j\} \in S$, in order to cover w_j.

3. *Each x_i $(1 \le i \le 3q)$ is incident to at least one communication edge of S, in order to dominate $\{x_i, y_i\}$.*

4. *Each x_i $(1 \le i \le 3q)$ is incident to at most one communication edge of S.* For, suppose not. Then $\{w_j, x_i\} \in S$ and $\{w_k, x_i\} \in S$ for some j, k $(1 \le j < k \le m)$. By Fact 1, $\{v_k, w_k\} \notin S$. We claim that $\{w_k, x_i\}$ is the only communication edge of S incident to w_k. For, suppose not. Then $\{w_k, x_l\} \in S$ for some l $(1 \le l \ne i \le 3q)$. Since $\{w_k, x_l\}$ is dominated by $\{x_l, y_l\}$, and $\{w_j, x_i\}$ is dominated by $\{x_i, y_i\}$, then $S \backslash \{\{w_k, x_i\}\}$ is a restricted edge total dominating set for G, contradicting the minimality of S. Thus the claim is established. Set

$$S' = (S \backslash \{\{w_k, x_i\}\}) \cup \{\{v_k, w_k\}\}.$$

   Then it may be verified that S' is a minimal restricted edge total dominating set for G with $|S'| = |S|$ and one fewer communication edge, contradicting the choice of S. Thus x_i is indeed incident to at most one communication edge of S.

Let there be l communication edges in S. Then Fact 3 implies that $l \ge 3q$, and Fact 4 implies that $l \le 3q$, so that $l = 3q$. Now suppose that the communication edges of S are incident to a total of s vertices of the form w_j. Then $s \ge q$. By Facts 1 and 2, S contains exactly $(m - s)$ edges of the form $\{v_j, w_j\}$. Finally, S contains all $3q$ edges of the form $\{x_i, y_i\}$ $(1 \le i \le 3q)$ and all $2m$ edges of the form $\{t_j, u_j\}$ and $\{u_j, v_j\}$ $(1 \le j \le m)$. Hence, having accounted for all the elements in S,

$$\begin{aligned} |S| &= 3q + 3q + (m - s) + 2m \\ &= 3m + 6q - s. \end{aligned} \tag{7.1}$$

Suppose that $s > q$. Then by Equation 7.1, $|S| < K$, a contradiction. Thus $s = q$. Hence there are $3q$ communication edges, incident to exactly q vertices of the form w_j $(1 \le j \le m)$. Thus the set

$$\{c_j : 1 \le j \le m \;\wedge\; w_j \text{ is incident to three communication edges of } S\}$$

is an exact cover for A. ■

Corollary 7.3.5 MAXIMUM MINIMAL TEST SET DECISION *is NP-complete, even if $|c| = 2$, for all $c \in C$.*

Proof: Clearly that MAXIMUM MINIMAL TEST SET DECISION is in NP. NP-hardness follows from the NP-completenes of MMRETDSD and Corollary 7.3.3. It is clear that, in the implicit constructed instance of MAXIMUM MINIMAL TEST SET DECISION, all $c \in C$ satisfy $|c| = 2$. ∎

7.4 Data Storage

7.4.1 Bin packing

In this section, we consider two maximinimal versions of a source optimisation problem related to the well-known BIN PACKING problem [92, problem SR1] (defined informally in Section 1.4.4).

Source problem: MINIMUM BIN PACKING$= \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle U, s, B \rangle : |U| < \infty \ \wedge \ s : U \longrightarrow \mathbb{Z}^+ \ \wedge \ B \in \mathbb{Z}^+\}$
- $\mathcal{U}(x) = \{P : P \text{ is a partition of } U\}$
- $\pi(x, P) \Leftrightarrow \forall\, S \in P \bullet \sum_{u \in S} s(u) \leq B$
- $m(x, P) = |P|$
- $\text{OPT} = \min$.

Given an instance x of MINIMUM BIN PACKING, we call an element P of $\mathcal{F}(x)$ a *bin packing* for the instance x.

Complexity of source problem: MINIMUM BIN PACKING DECISION is NP-complete, for this problem is equivalent to problem SP1 of [92].

Partial order: partition merge

Maximinimal problem name: MAXIMUM $\prec_a$-MINIMAL BIN PACKING.

Given an instance x of MINIMUM BIN PACKING, intuitively a bin packing P is $\prec_a^x$-minimal if it is not possible to merge the contents of any two bins of P into a single bin, without exceeding the bin capacity.

Complexity of maximinimal problem: The complexity of MAXIMUM $\prec_a$-MINIMAL BIN PACKING DECISION is resolved by the following theorem. The proof involves a transformation from the NP-complete problem 3-PARTITION [92, problem SP15], which may be defined as follows:

> *Name:* 3-PARTITION.
> *Instance:* Positive integers n, S and collection of $3n$ elements, each element a_i with associated size $s(a_i) \in \mathbb{Z}^+$, such that $S/4 < s(a_i) < S/2$ $(1 \leq i \leq 3n)$ and $\sum_{i=1}^{3n} s(a_i) = nS$.
> *Question:* Whether there is a 3-partition of the a_i, i.e., a permutation $i_1, i_2, \ldots, i_{3n}$ of $1, 2, \ldots, 3n$ such that $\sum_{k=1}^{3} s(a_{i_{3(j-1)+k}}) = S$ for each j $(1 \leq j \leq n)$.

Theorem 7.4.1 MAXIMUM MINIMAL $\prec_a$-BIN PACKING *is NP-complete.*

Proof: MAXIMUM $\prec_a$-MINIMAL BIN PACKING DECISION is in NP, for, given a partition $P = \{U_1, U_2, \ldots, U_k\}$ of a set U, we may verify, in polynomial time, that P is a bin packing, and that no pair of sets in P can be merged without exceeding the bin capacity.

To show NP-hardness, we give a transformation from 3-PARTITION, defined on Page 132. Suppose we have an instance of 3-PARTITION: elements $a_1, a_2, \ldots, a_{3n}$ for some n, each with associated size $s(a_i) \in \mathbb{Z}^+$, such that $S/4 < s(a_i) < S/2$ and $\sum_{i=1}^{3n} s(a_i) = nS$, for some S. Clearly, we lose no generality by assuming that $n \geq 3$. We construct the following instance x of MAXIMUM $\prec_a$-MINIMAL BIN PACKING DECISION: objects $U = \{u_1, u_2, \ldots, u_{3n}\}$ where $s(u_i) = s(a_i)$ $(1 \leq i \leq 3n)$, bin capacity $B = 2S - 1$ and target number of bins $K = n$. The transformation is clearly polynomial, and the claim is there is a 3-partition of the a_i if and only if there is a $\prec_a^x$-minimal packing of U into $k \geq K$ bins.

For, suppose that there is some permutation $i_1, i_2, \ldots, i_{3n}$ of $1, 2, \ldots, 3n$ such that $\sum_{k=1}^{3} s(a_{i_{3(j-1)+k}}) = S$ for each j $(1 \leq j \leq n)$. We construct a packing into n bins as follows. Into each bin B_j $(1 \leq j \leq n)$ we place $a_{i_{3(j-1)+k}}$, for $k = 1, 2, 3$. Each bin has objects of total size S, and hence no bin is overfilled. Clearly no pair of bins may be merged, so that the constructed packing is $\prec_a^x$-minimal.

Conversely suppose that there is a $\prec_a^x$-minimal packing of U into bins $B_1, B_2, \ldots, B_k$ for some $k \geq K$. If $k > n$ then at least two bins are filled to total weight $< S$ and hence may be merged, a contradiction. Thus $k = n$. The claim is that each bin is filled to total weight exactly S. For suppose not. Then there is some bin B_j $(1 \leq k \leq n)$ such that B_j has total weight $< S$. Without loss of generality suppose that $j = 1$. Let w_i denote the total size of objects in B_i in this packing. We claim that there is some bin B_k $(2 \leq k \leq n)$ such that $w_1 + w_k \leq B$. For suppose not. Then for each k $(2 \leq k \leq n)$,

$$w_1 + w_k \geq 2S \tag{7.2}$$

and hence

$$\begin{aligned} w_k &\geq 2S - w_1 \\ &> S. \end{aligned} \tag{7.3}$$

Thus the total weight of objects packed is equal to

$$\begin{aligned} \sum_{i=1}^{n} w_i &= w_1 + w_2 + \sum_{i=3}^{n} w_i \\ &\geq 2S + \sum_{i=3}^{n} w_i && \text{By Inequality 7.2} \\ &> 2S + (n-2)S && \text{By Inequality 7.3, and since } n \geq 3 \\ &= nS \end{aligned}$$

which is a contradiction. Thus there is some bin B_k $(k > 1)$ that may be merged with B_1. This contradiction shows that $w_i = S$ $(1 \leq i \leq n)$. Let $i_1, i_2, \ldots, i_{3n}$ be a permutation of $1, 2, \ldots, 3n$ such that $\{u_{i_{3r+1}}, u_{i_{3r+2}}, u_{i_{3r+3}}\} \in B_{r+1}$ for $0 \leq r \leq n-1$. Then

$$\{\{a_{i_{3r+1}}, a_{i_{3r+2}}, a_{i_{3r+3}}\} : 0 \leq r \leq n-1\}$$

is a 3-partition of the a_i. ■

Note that 3-PARTITION is strongly NP-complete [92, problem SP15], and the transformation given in Theorem 7.4.1 is pseudo-polynomial. Hence MAXIMUM $\prec_a$-MINIMAL BIN PACKING is strongly NP-complete.

Partial order: partition redistribution

Maximinimal problem name: MAXIMUM $\prec_b$-MINIMAL BIN PACKING.

Given an instance x of MINIMUM BIN PACKING, intuitively a bin packing P is $\prec_b^x$-minimal if it is not possible to redistribute the contents of a bin of P amongst the remaining bins, without exceeding the bin capacity.

Complexity of maximinimal problem: The complexity of MAXIMUM $\prec_b$-MINIMAL BIN PACKING DECISION is resolved by the following theorem.

Theorem 7.4.2 MAXIMUM $\prec_b$-MINIMAL BIN PACKING DECISION *is NP-hard.*

Proof: We give a transformation from 3-PARTITION, defined on Page 132. Suppose we have an instance of 3-PARTITION: elements $a_1, a_2, \ldots, a_{3n}$ for some n, each with associated size $s(a_i) \in \mathbb{Z}^+$, such that $S/4 < s(a_i) < S/2$ and $\sum_{i=1}^{3n} s(a_i) = nS$, for some S. Clearly, we lose no generality by assuming that $n \geq 2$. We construct the following instance x of MAXIMUM $\prec_b$-MINIMAL BIN PACKING DECISION: bin capacity $B = 4nS$, target number of bins $K = n + 1$, and elements of the following type:

- 'small' elements $w_1, w_2, \ldots, w_{3n}$, where $s(w_i) = s(a_i)$, for $1 \leq i \leq 3n$.
- 'x' element, where $s(x) = 2nS - S$.
- 'y' elements $y_1, y_2, \ldots, y_n$, where $s(y_i) = 2nS + 1$.
- 'z' element, where $s(z) = S$.

The claim is that there a $\prec_b^x$-minimal packing of the elements into $k \geq K$ bins if and only if there is a 3-partition of the a_i.

For, suppose that there is some permutation $i_1, i_2, \ldots, i_{3n}$ of $1, 2, \ldots, 3n$ such that $\sum_{k=1}^{3} s(a_{i_{3(j-1)+k}}) = S$ for each j $(1 \leq j \leq n)$. We construct a packing into $n+1$ bins as follows. Into each bin B_j, for $1 \leq j \leq n$, we place y_j and $a_{i_{3(j-1)+k}}$, for $k = 1, 2, 3$. Into bin B_{n+1} we place x and z.

This packing clearly overfills no bin. Also, the packing is $\prec_b^x$-minimal, which may be seen as follows. No bin B_j $(1 \le j \le n)$ can be redistributed among the other bins, for no bin B_k $(1 \le k \ne j \le n+1)$ has room for y_j in particular. Also, B_{n+1} cannot be redistributed among the other bins, for no bin B_k $(1 \le k \le n)$ has room for x in particular.

Conversely suppose that there is a $\prec_b^x$-minimal packing of the objects into $k \ge K$ bins. Clearly each of the elements $y_1, y_2, \ldots, y_n$ must be in different bins. The remaining elements to be packed have weight totalling $3nS$, and hence $k = n+1$. Without loss of generality we may assume that y_j is packed in bin B_j for $1 \le j \le n$. The remainder of the proof is split into cases, according to where x and z have been packed.

Case 1: $x \in B_i, z \in B_j$ $(1 \le i, j \le n)$. Then $i \ne j$. But then

$$\begin{aligned} s(y_j) + s(z) + \sum_{w_r \in B_j} s(w_r) + \sum_{w_r \in B_{n+1}} s(w_r) &\le (2nS+1) + S + nS \\ &\le B \end{aligned}$$

so that B_{n+1} can be merged with B_j, a contradiction. Hence this case cannot occur.

Case 2: $x \in B_i$ $(1 \le i \le n), z \in B_{n+1}$. Then for any j $(1 \le j \ne i \le n)$, we have that

$$\begin{aligned} s(y_j) + s(z) + \sum_{w_r \in B_j} s(w_r) + \sum_{w_r \in B_{n+1}} s(w_r) &\le (2nS+1) + S + nS \\ &\le B \end{aligned}$$

so that B_{n+1} can be merged with B_j, a contradiction. Hence this case cannot occur.

Case 3: $x \in B_{n+1}, z \in B_i$ $(1 \le i \le n)$. Then suppose, without loss of generality, that $i = 1$. The claim is each of the bins $B_2, B_3, \ldots, B_{n+1}$ has 'small' elements of weight totalling $\ge S$. For, suppose that some B_j does not. We consider two subcases.

Subcase 3a: $2 \le j \le n$. Then

$$\begin{aligned} s(x) + s(y_j) + \sum_{w_r \in B_j} s(w_r) &\le (2nS - S) + (2nS+1) + (S-1) \\ &= B. \end{aligned}$$

Also, we have

$$\begin{aligned} s(y_1) + s(z) + \sum_{w_r \in B_1} s(w_r) + \sum_{w_r \in B_{n+1}} s(w_r) &\le (2nS+1) + S + nS \\ &\le B \end{aligned}$$

so that bin B_{n+1} can be redistributed over bins B_1 and B_j, a contradiction.

Subcase 3b: $j = n+1$. Then

$$s(x) + s(y_1) + \sum_{w_r \in B_{n+1}} s(w_r) \le (2nS - S) + (2nS+1) + (S-1)$$

$$= \quad B.$$

Also, we have, for any $2 \le k \le n$,

$$\begin{aligned} s(y_k) + s(z) + \sum_{w_r \in B_1} s(w_r) + \sum_{w_r \in B_k} s(w_r) &\le (2nS+1) + S + nS \\ &\le B. \end{aligned}$$

Hence bin B_1 can be redistributed over bins B_k and B_{n+1}, a contradiction.

Thus each bin B_j $(2 \le j \le n+1)$ must have 'small' elements of weight totalling exactly S. Let $i_1, i_2, \ldots, i_{3n}$ be a permutation of $1, 2, \ldots, 3n$ such that $\{w_{i_{3r+1}}, w_{i_{3r+2}}, w_{i_{3r+3}}\} \in B_{r+2}$ for $0 \le r \le n-1$. Then

$$\{\{a_{i_{3r+1}}, a_{i_{3r+2}}, a_{i_{3r+3}}\} : 0 \le r \le n-1\}$$

is a 3-partition of the a_i.

Case 4: $x \in B_{n+1}, z \in B_{n+1}$. Then for each j $(1 \le j \le n)$, B_j must have 'small' elements of weight totalling $\ge S$. For, if not, then there is some bin B_j $(1 \le j \le n)$ such that

$$\begin{aligned} s(x) + s(y_j) + \sum_{w_r \in B_j} s(w_r) &\le (2nS - S) + (2nS+1) + (S-1) \\ &= B. \end{aligned}$$

For any other bin B_k $(1 \le k \ne j \le n)$ we have

$$\begin{aligned} s(y_k) + s(z) + \sum_{w_r \in B_k} s(w_r) + \sum_{w_r \in B_{n+1}} s(w_r) &\le (2nS+1) + S + nS \\ &\le B. \end{aligned}$$

Hence bin B_{n+1} can be redistributed over bins B_j and B_k, a contradiction. Thus each bin B_j $(1 \le j \le n)$ must have 'small' elements of weight totalling exactly S. Let $i_1, i_2, \ldots, i_{3n}$ be a permutation of $1, 2, \ldots, 3n$ such that $\{w_{i_{3r+1}}, w_{i_{3r+2}}, w_{i_{3r+3}}\} \in B_{r+1}$ for $0 \le r \le n-1$. Then

$$\{\{a_{i_{3r+1}}, a_{i_{3r+2}}, a_{i_{3r+3}}\} : 0 \le r \le n-1\}$$

is a 3-partition of the a_i. ∎

Note that MAXIMUM $\prec_b$-MINIMAL BIN PACKING DECISION is not known to be in NP (see Theorem 9.3.2). Also, the transformation given in Theorem 7.4.2 is pseudo-polynomial. Hence, since 3-PARTITION is strongly NP-complete [92, problem SP15], then MAXIMUM $\prec_b$-MINIMAL BIN PACKING is strongly NP-complete.

7.5 Compression and Representation

In this section, we consider source optimisation problems concerned with finding longest and shortest length common subsequences, supersequences, substrings and superstrings of a given set of strings.

7.5.1 Common subsequence

Source problem: LONGEST COMMON SUBSEQUENCE (LCS) = $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle S, \Sigma \rangle : |S| < \infty \ \wedge \ \forall s \in S \bullet s \in \Sigma^*\}$
- $\mathcal{U}(x) = \{t \in \Sigma^* : |t| \leq \min\{|s| : s \in S\}\}$[1]
- $\pi(x, t) \Leftrightarrow t \lll S$
- $m(x, t) = |t|$
- $\text{OPT} = \max$.

Complexity of source problem: LCS DECISION is NP-complete, even for $|\Sigma| = 2$ [161]. For $|S| = 2$, a variety of polynomial-time algorithms have been proposed for LCS, and are surveyed by Fraser [80, p.10]. Polynomial-time algorithms also exist for the case that $|S|$ is fixed, with $|S| > 2$, and are also surveyed by Fraser [80, p.11]. For example, Irving and Fraser [130] provide a polynomial-time algorithm for the case $|S| = 3$, extendible to any fixed $|S| > 2$.

Partial order: subsequence

Minimaximal problem name: SHORTEST MAXIMAL COMMON SUBSEQUENCE (SMCS)

Complexity of minimaximal problem: Fraser, Irving and Middendorf [81] show that SMCS DECISION is NP-complete in general. For $|S| = 2$, the same authors show SMCS to be polynomial-time solvable, and remark that this result is extendible to any fixed $|S| > 2$.

7.5.2 Common supersequence

Source problem: SHORTEST COMMON SUPERSEQUENCE (SCS)= $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle S, \Sigma \rangle : |S| < \infty \ \wedge \ \forall s \in S \bullet s \in \Sigma^*\}$
- $\mathcal{U}(x) = \{t \in \Sigma^* : |t| \leq \sum\{|s| : s \in S\}\}$
- $\pi(x, t) \Leftrightarrow S \lll t$
- $m(x, t) = |t|$

[1] We cannot simply set $\mathcal{U}(x) = \Sigma^*$ here, since then $\mathcal{U}(x)$ would not be finite, a criterion of Definition 2.2.1. The same comment applies in the case of the SHORTEST COMMON SUPERSEQUENCE, LONGEST COMMON SUBSTRING and SHORTEST COMMON SUPERSTRING problems.

- OPT = min .

Complexity of source problem: SCS DECISION is NP-complete [161], even for $|\Sigma| = 2$ [188]. Timkovskii [205] considers the (m, n)-SCS problem, which is the original SCS problem with given constraints m and n on the input, where

$$m = \max\{|s| : s \in S\} \tag{7.4}$$

and

$$n = \max\{|O_c| : c \in \Sigma\}, \tag{7.5}$$

where O_c is the *orbit* of c, i.e. the total number of occurrences of c among the strings of S. (2,2)-SCS is found to be polynomial-time solvable, while the decision versions of (2,3)-SCS and (3,2)-SCS are reported NP-complete. Garey and Johnson [92, p.228] assert that SCS is polynomial-time solvable for $m = 2$; however Timkovskii's result for (2,3)-SCS disproves this.

For $|S| = 2$ the SCS and LCS problems are dual. However, for $|S| > 2$ (fixed) there is no obvious duality between the two problems. Even so, polynomial-time algorithms still exist in this case. Fraser [80, p.14] surveys existing such algorithms and presents two new ones [80, Chapter 3].

Partial order: subsequence

Maximinimal problem name: LONGEST MINIMAL COMMON SUPERSEQUENCE (LMCS).

Complexity of maximinimal problem: Fraser, Irving and Middendorf [81] show that LMCS DECISION is NP-complete. The same authors also demonstrate that LMCS is polynomial-time solvable for the case $|S| = 2$, and also that this result holds for any fixed $|S| > 2$. In the case $m = 2$ (where m is defined by Equation 7.4), a linear-time algorithm is given for LMCS.

Thus the SHORTEST COMMON SUPERSEQUENCE and LONGEST MINIMAL COMMON SUPERSEQUENCE problems, each when restricted to the case when all input strings are of length two, form an example of an NP-hard source optimisation problem Π, together with a polynomial-time solvable maximinimal optimisation problem derived from Π using a partial order.

7.5.3 Common substring

Source problem: LONGEST COMMON SUBSTRING (LCSt) $= \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, where

- $\mathcal{I} = \{\langle S, \Sigma\rangle : |S| < \infty \ \wedge \ \forall s \in S \bullet s \in \Sigma^*\}$
- $\mathcal{U}(x) = \{t \in \Sigma^* : |t| \leq \min\{|s| : s \in S\}\}$
- $\pi(x, t) \Leftrightarrow t \lesssim S$
- $m(x, t) = |t|$

- OPT = max.

Complexity of source problem: LCSt is polynomial-time solvable. Given a set S containing k strings, where n denotes the *total* length of all strings in S, Gusfield [103, §7.6] describes an $O(kn)$ method to generate a longest common substring of S using suffix trees. (Gusfield describes the suffix tree data structure in Chapter 5 of [103].) However, an even more sophisticated approach, also using suffix trees, yields an $O(n)$ algorithm to solve the problem [103, §9.7].

Partial order: substring

Minimaximal problem name: SHORTEST MAXIMAL COMMON SUBSTRING (SMCSt).

Complexity of minimaximal problem: As in the case of LCSt, the SMCSt problem is polynomial-time solvable using suffix trees. Before demonstrating this, we establish some definitions relating to suffix trees (the terminology follows that of Gusfield [103, §5.2]).

Let T be a suffix tree for a string s. The *label of a path* from the root of T to a node v is the concatenation, in order, of the substrings labelling the edges of that path. The *path label* of a node v is the label of the path from the root of T to v. For any node v, the *string depth* of v is the number of characters in the path label of v.

Theorem 7.5.1 *Let S be a set of k strings over an alphabet Σ, and let n denote the total length of all strings in S. Then we may find a shortest maximal common substring of S in $O(|\Sigma|kn)$ time.*

Proof: Let s_i $(1 \leq i \leq k)$ be the strings in S. To each string s_i $(1 \leq i \leq k)$, we append a unique termination symbol α_i not occurring in Σ; let s_i' be the resultant string. Denote by s the concatenation of the strings s_i' $(1 \leq i \leq k)$. Now suppose that x is a suffix of s, beginning at position i of s. This position of s corresponds to a unique string s_j' for some j $(1 \leq j \leq k)$. We call j the *string identifier* for suffix x. Define the *suffix predecessor of x in s* to be the $(i-1)$th character of s if $i > 1$, or α_0 (a symbol not occuring in Σ) if $i = 1$.

We build the suffix tree T for the string s, storing two pieces of information at each leaf node. Recall that each leaf node v of T corresponds to a unique suffix x of s. Store the string identifier for suffix x, together with the suffix predecessor of x in s, at v. (For the purposes of this algorithm, it is not necessary to store at v the *suffix number* of x in s, i.e., the position of s at which x begins.) Denote by $i(v)$ the string identifier of the suffix x represented by v, and denote by $p(v)$ the suffix predecessor of the suffix x represented by v in s. Two more pieces of information will be stored at each node of T, and these will be described later. It is clear that the construction of this paragraph may be carried out in $O(n)$ time, which is the time required to build a suffix tree [209].

For a node v of T, let T_v denote the subtree of T with root v. Let $C(v)$ denote the number of distinct string identifiers that appear at the leaves of T_v. Define a *matching node* of T to be an internal node v of T such that $C(v) = k$. It follows that a string p is a common substring of S if and only if p is, or is a prefix of, the path label of some matching node v of T. Thus a string p is a maximal common substring of S implies that p is the path label of some matching node v of T. Computing $C(v)$ for each node v may be carried out in $O(kn)$ time overall [103, §7.6.1].

We now consider *R-tight matching nodes.* Such a node v is a matching node such that no child of v in T is a matching node.

Claim. Let v be a matching node of T and let p be the path label of v. Then v is an R-tight matching node if and only if $p +\!\!+ \langle\sigma\rangle \not\mathrel{\underline{\ll}} S$, for each $\sigma \in \Sigma$.

Proof of claim: Suppose that v is an R-tight matching node and $q = p +\!\!+ \langle\sigma\rangle \mathrel{\underline{\ll}} S$, for some $\sigma \in \Sigma$. Then $q \mathrel{\underline{\ll}} s_j$ for each j $(1 \leq j \leq k)$, which implies that there are k suffixes x_j $(1 \leq j \leq k)$ of s such that, for each j $(1 \leq j \leq k)$, x_j has string identifier j, and q is a prefix of x_j. Thus, by definition of T, there is a matching node w in T_v, where w is a child of v (w has path label r, such that either $q = r$, or q is a prefix of r). Thus v has a child that is a matching node, a contradiction.

Conversely, suppose that $p +\!\!+ \langle\sigma\rangle \not\mathrel{\underline{\ll}} S$ for each $\sigma \in \Sigma$, and v is not an R-tight matching node. Then v has a child w that is a matching node; let q be the path label of w. Then by definition of T, there is some $\sigma \in \Sigma$ such that $p +\!\!+ \langle\sigma\rangle$ is a prefix of q (possibly $p +\!\!+ \langle\sigma\rangle = q$). Thus we reach a contradiction, since $q \mathrel{\underline{\ll}} S$ implies that $p +\!\!+ \langle\sigma\rangle \mathrel{\underline{\ll}} S$.

It is clear that the R-tight matching nodes may be determined by a straightforward traversal of T, in $O(n)$ time, once the $C(v)$ values have been computed.

By the above claim, the path label of an R-tight matching node is a common substring of S that cannot be extended to the right to give another common substring of S. Next, we show how to locate common substrings that cannot be extended to the left, in addition to being non-extendable to the right.

For any node v of T, for any $\sigma \in \Sigma$ and for any j $(1 \leq j \leq k)$, let $D_v(\sigma, j)$ have value true if T_v has a leaf node with suffix predecessor σ and string identifier j; $D_v(\sigma, j) = \mathsf{false}$ otherwise. Define an *LR-tight matching node* v to be an R-tight matching node v such that, for all $\sigma \in \Sigma$,

$$\bigwedge_{j=1}^{k} D_v(\sigma, j) = \mathsf{false}.$$

Claim. Let v be an R-tight matching node of T and let p be the path label of v. Then v is an LR-tight matching node if and only if $\langle\sigma\rangle +\!\!+ p \not\mathrel{\underline{\ll}} S$, for each $\sigma \in \Sigma$.

Proof of claim: Suppose that v is an LR-tight matching node and $q = \langle\sigma\rangle +\!\!+ p \mathrel{\underline{\ll}} S$, for some $\sigma \in \Sigma$. Then $q \mathrel{\underline{\ll}} s_j$ for each j $(1 \leq j \leq k)$, which implies that there are k suffixes x_j $(1 \leq j \leq k)$ of s such that, for each j $(1 \leq j \leq k)$, p is a prefix of x_j, x_j has string identifier j, and σ is the suffix predecessor of x_j in s. Thus, by construction of T, we have that T_v has a leaf node with suffix predecessor σ and string identifier j, for each j $(1 \leq j \leq k)$. Hence $D_v(\sigma, j) = \mathsf{true}$ for each j $(1 \leq j \leq k)$, a contradiction.

Conversely, suppose that $\langle\sigma\rangle +\!\!+ p \not\mathrel{\underline{\ll}} S$ for each $\sigma \in \Sigma$, and v is not an LR-tight matching node. Then there is some $\sigma \in \Sigma$ such that T_v has a leaf node with suffix predecessor σ and string identifier j for each j $(1 \leq j \leq k)$. Hence $q = \langle\sigma\rangle +\!\!+ p$ satisfies

$q \lll s_j$, for each j $(1 \leq j \leq k)$, so that $q \lll S$, a contradiction.

By the above claim, a string p is a maximal common substring of S if and only if p is the path label of an LR-tight matching node v of T. A straightforward traversal of T will determine the R-tight matching nodes that are also LR-tight matching nodes, in $O(kn)$ time, once the D_v matrices have been computed. Thus it remains to establish the D_v matrices, for each node v in T. In order to do this, we use the $i(v)$ and $p(v)$ values, stored at leaf nodes v.

The D_v matrices may be computed as follows. If v is a leaf node, then for any $\sigma \in \Sigma$ and for any j $(1 \leq j \leq k)$,

$$D_v(\sigma, j) = \begin{cases} \textsf{true}, & \text{if } p(v) = \sigma \text{ and } i(v) = j \\ \textsf{false}, & \text{otherwise.} \end{cases}$$

If v is an internal node, then for any $\sigma \in \Sigma$ and for any j $(1 \leq j \leq k)$,

$$D_v(\sigma, j) = \bigvee \{D_w(\sigma, j) : w \text{ is a child of } v\}.$$

It is clear that the computation of the D_v matrices, for every node v of T, requires $O(|\Sigma|kn)$ time in total.

Once the LR-tight matching nodes have been located, a final traversal of the tree will establish an LR-tight matching node v of smallest string depth, in $O(n)$ time. By construction, the path label p of v corresponds to a shortest maximal common substring of S. Thus the overall time complexity of this algorithm is $O(|\Sigma|kn)$. ∎

7.5.4 Common superstring

Source problem: SHORTEST COMMON SUPERSTRING (SCSt)= $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle S, \Sigma \rangle : |S| < \infty \ \wedge \ \forall s \in S \bullet s \in \Sigma^*\}$
- $\mathcal{U}(x) = \{t \in \Sigma^* : |t| \leq \sum\{|s| : s \in S\}\}$
- $\pi(x, t) \Leftrightarrow S \lll t$
- $m(x, t) = |t|$
- $\text{OPT} = \min$.

Complexity of source problem: The decision problem of SCSt is NP-complete [88], even for $|\Sigma| = 2$. When all strings in S have length at most two, SCSt is solvable in linear time [88].

Partial order: substring

Maximinimal problem name: LONGEST MINIMAL COMMON SUPERSTRING.

Complexity of maximinimal problem: The complexity of LONGEST MINIMAL COMMON SUPERSTRING is open.

7.6 Mathematical Programming

7.6.1 Knapsack

The following source optimisation problem is based on the well-known KNAPSACK problem [92, problem MP9] (defined informally in Section 1.4.4).

Source problem: MAXIMUM KNAPSACK= $\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle w_1, \ldots, w_n, v_1, \ldots, v_n, W \rangle : n, W \in \mathbb{Z}^+ \ \wedge \ \forall 1 \leq i \leq n \bullet w_i, v_i \in \mathbb{Z}^+\}$
- $\mathcal{U}(x) = \mathbb{P}(\{1, 2, \ldots, n\})$
- $\pi(x, S) \Leftrightarrow \sum_{i \in S} w_i \leq W$
- $m(x, S) = \sum_{i \in S} v_i$
- $\text{OPT} = \max$.

Complexity of source problem: MAXIMUM KNAPSACK DECISION is NP-complete [92, problem MP9].

Partial order: set inclusion

Minimaximal problem name: MINIMUM MAXIMAL KNAPSACK.

Intuitively, a knapsack packing is maximal if we cannot put any more objects into the knapsack without exceeding the capacity. Whereas the objective in MAXIMUM KNAPSACK is to maximise the total value of the objects selected, the objective in MINIMUM MAXIMAL KNAPSACK is to minimise the total value of the selected objects, such that this maximality criterion is satisfied. Burglary is usually cited as an application of MAXIMUM KNAPSACK; one might envisage spring-cleaning as being an application of MINIMUM MAXIMAL KNAPSACK!

Complexity of minimaximal problem: The complexity of MINIMUM MAXIMAL KNAPSACK DECISION is resolved by the following theorem. The proof involves a transformation from the NP-complete problem SUBSET SUM [92, problem SP13], which may be defined as follows:

> *Name:* SUBSET SUM.
> *Instance:* Collection of n elements, each with weight $a_i \in \mathbb{Z}^+$ $(1 \leq i \leq n)$, and target value $B \in \mathbb{Z}^+$.
> *Question:* Is there a subset S of $\{1, 2, \ldots, n\}$ such that $\sum_{i \in S} a_i = B$?

Theorem 7.6.1 MINIMUM MAXIMAL KNAPSACK DECISION *is NP-complete.*

Proof: It is clear that MINIMUM MAXIMAL KNAPSACK DECISION is in NP. To show NP-hardness, we give a transformation from SUBSET SUM, as defined above. Suppose we have an instance of SUBSET SUM: a collection of n elements, each with weight $a_i \in \mathbb{Z}^+$ $(1 \leq i \leq n)$, and target value B. Construct the following instance of MINIMUM MAXIMAL KNAPSACK DECISION: weight bound $W = B$, value bound $V = n$, and $n+1$ objects, where $w_i = a_i$ for $1 \leq i \leq n$, $v_i = 1$ for $1 \leq i \leq n$, and $w_{n+1} = 1$, $v_{n+1} = n+1$. The transformation is clearly polynomial, and we claim that there is some subset of the a_i which sums to B if and only if there is a maximal knapsack packing of total value at most V.

For, suppose that there is a maximal knapsack packing of total value at most V. Then there exists some $S \subseteq \{1, 2, \ldots, n+1\}$ such that

$$\sum_{i \in S} w_i \leq W = B \quad \text{and} \quad \sum_{i \in S} v_i \leq V = n.$$

The value inequality implies that $S \subseteq \{1, 2, \ldots, n\}$. If strict inequality holds in the weights inequality, then $S \cup \{n+1\}$ is a feasible packing, contradicting the maximality of S. Hence equality holds in the weights inequality, which implies that $\sum_{i \in S} a_i = B$, and the SUBSET SUM instance has a solution.

Conversely, suppose that the SUBSET SUM instance has a solution. Then there exists some $S \subseteq \{1, 2, \ldots, n\}$ such that

$$\sum_{i \in S} w_i = \sum_{i \in S} a_i = B = W. \tag{7.6}$$

Moreover, $\sum_{i \in S} v_i \leq n = V$. Finally, by Equation 7.6, the knapsack is full, so the packing is maximal, as required. ∎

Note that it is open as to whether MINIMUM MAXIMAL KNAPSACK is solvable in pseudo-polynomial time, or strong NP-completeness holds for the decision problem.

7.7 Logic

7.7.1 Definitions relating to logic problems

Before introducing our source optimisation problems concerning logic, we present some related definitions.

Given a set U of *variables*, a *literal* σ over U is either a variable u or its negation $\overline{u}$, where $u \in U$. A *clause* is a set of literals over U, representing the logical disjunction of these literals. A *well-formed formula (w.f.)* is a set of clauses, representing the logical conjunction of these clauses.

Negation extends to a literal σ as follows. If $\sigma = u$ for some $u \in U$, then $\overline{\sigma}$ denotes $\overline{u}$. If $\sigma = \overline{u}$ for some $u \in U$, then $\overline{\sigma}$ denotes u.

Letting F, T respectively stand for the Boolean truth values false, true, a *truth assignment* for U is a function $f : U \longrightarrow \{F, T\}$. The truth assignment f extends to literals, clauses and well-formed formulae as follows:

- Given a variable $u \in U$, the literal $\sigma = u$ satisfies $f(\sigma) = T$ if and only if $f(u) = T$, and the literal $\sigma = \overline{u}$ satisfies $f(\sigma) = T$ if and only if $f(u) = F$.
- Given a clause $C_i = \{\sigma_1, \sigma_2, \ldots \sigma_r\}$, $f(C_i) = T$ if and only if $f(\sigma_j) = T$ for some j $(1 \leq j \leq r)$.
- Given a w.f. $C = \{C_1, C_2, \ldots, C_m\}$, $f(C) = T$ if and only if $f(C_i) = T$ for each i $(1 \leq i \leq m)$.

A variable, literal or clause X is said to be *true*, or *satisfied*, under f if $f(X) = T$, and X is *false*, or *not satisfied*, under f if $f(X) = F$. A w.f. C over a set of variables U is said to be *satisfiable* if there exists a truth assignment f such that $f(C) = T$.

Given a set of variables U, a w.f. $C = \{C_1, C_2, \ldots, C_m\}$ over U and a truth assignment $f : U \longrightarrow \{F, T\}$, we denote by $\mathcal{C}_f$ the clauses of C satisfied by f, i.e.,

$$\mathcal{C}_f = \{C_i \in C : f(C_i) = T\}.$$

Given a clause C_i of C, we denote by $U(C_i)$ the variables in C_i, i.e.,

$$U(C_i) = \{u \in U : u \in C_i \vee \overline{u} \in C_i\}.$$

This definition extends to a set of clauses C', where $C' \subseteq C$, as follows:

$$U(C') = \bigcup_{C_i \in C'} U(C_i).$$

Thoughout this section, we assume that $U(C) = U$, for any w.f. C over a set of variables U, i.e., every variable in U occurs in some clause of C.

Given a pair $x = \langle U, C\rangle$, representing a set of variables U and a w.f. C over U, let $\mathcal{F}(x)$ denote the set of all truth assignments for U. We may define a partial order[2] $\prec_t^x$ on $\mathcal{F}(x)$, called the *partial order on truth assignments*, as follows:

$$\prec_t^x = \left\{(f, g) \in \mathcal{F}(x) \times \mathcal{F}(x) : \mathcal{C}_f \subset \mathcal{C}_g \wedge f\big|_{U(\mathcal{C}_f)} \equiv g\big|_{U(\mathcal{C}_f)}\right\}.$$

Thus, for two truth assignments f, g, we have $f \prec_t^x g$ if g satisfies all of the clauses of C satisfied by f, plus at least one more, and f, g agree on $U(\mathcal{C}_f)$, the variables belonging to the clauses of C satisfied by f. To motivate this definition, observe that a partial order $\prec_{t,2}^x$, of similar definition to $\prec_t^x$, but not requiring the two functions f, g to agree on $U(\mathcal{C}_f)$, preserves much less 'local' information. For, it is possible that f and g could agree on *none* of the variables of U, and yet $\mathcal{C}_f \subset \mathcal{C}_g$. As a simple example, consider the clauses $\{x, y\}, \{z\}$, for three variables x, y, z, and the truth assignment f given by $f(x) = T, f(y) = F, f(z) = F$. Then $f \prec_{t,2}^x g$, where the truth assignment g is given by

[2] The subscript t in $\prec_t^x$ refers to *truth assignment* here.

$g(x) = F, g(y) = T, g(z) = T$. Clearly $\mathcal{C}_f \subset \mathcal{C}_g$, but f, g differ on each of x, y, z. For an example of where we use the property that $f \prec_t^x g$ implies that f, g agree on $U(\mathcal{C}_f)$, see Theorem 8.3.1.

A convenient criterion for a truth assignment to be maximal with respect to $\prec_t^x$ is given by the following proposition. In the remainder of this section on logic, a truth assignment f is said to be *maximal* if f is $\prec_t^x$-maximal.

Proposition 7.7.1 *Let $x = \langle U, C \rangle$, where U is a set of variables and C is a w.f. over U, let $\mathcal{F}(x)$ denote the set of all truth assignments for U, and let $f \in \mathcal{F}(x)$. Then f is maximal if and only if $U(\mathcal{C}_f) = U$.*

Proof: If f is non-maximal, then clearly $U(\mathcal{C}_f) \subset U$. Conversely, suppose that $U(\mathcal{C}_f) \neq U$. Pick any $v \in U \backslash U(\mathcal{C}_f)$. Then v is a variable appearing in a clause C_i not satisfied by f, and not appearing in any clause satisfied by f. Thus, changing the value of $f(v)$ is bound to satisfy at least one more clause, whilst not affecting the satisfaction of the clauses in $\mathcal{C}_f$. More formally, define the truth assignment g as follows: for each $u \in U$,

$$g(u) = \begin{cases} f(u), & u \in U \backslash \{v\} \\ T, & (u = v) \wedge f(u) = F \\ F, & (u = v) \wedge f(u) = T. \end{cases}$$

Then f, g agree on the values of the variables in $\mathcal{C}_f$. In addition, $f(C_i) = F$, whereas $g(C_i) = T$. Hence $f \prec_t^x g$, so that f is non-maximal. ∎

Thus a truth assignment f is maximal if and only if every variable in U appears in some clause satisfied by f.

In each of the remaining sections, we define a source logic-related optimisation problem Π and obtain a minimaximal optimisation problem Π', using the partial order on truth assignments. The problem Π' is then studied from the point of view of algorithmic complexity.

7.7.2 2-satisfiability

In this section, we consider a source optimisation problem[3] in which the objective is to find a truth assignment that simultaneously satisfies the maximum number of clauses of a given w.f., given that each clause has size two.

Source problem: MAXIMUM 2SAT=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{\langle U, C \rangle : C$ is a w.f. over U and $|C_i| = 2$, for all $C_i \in C\}$

[3] There seem to be inconsistencies in the literature regarding terminology for logic-related problems. For example, 3SAT as defined by Garey and Johnson [92, problem LO2] involves clauses of size 3. However, 3SATISFIABILITY as defined by Aho et al. [1, p.384] involves clauses of size *at most* 3. Since the distinction will be of importance to us in this section, and in future sections, we refer to the latter problem as AT MOST 3SAT. These remarks also apply to MAXIMUM 2SAT and ONE-IN-THREE 3SAT (problems LO4 and LO5 of [92] respectively).

- $\mathcal{U}(G) = \{f : f \text{ is a truth assignment for } U\}$
- $\pi(\langle U, C\rangle, f) \Leftrightarrow \textsf{true}$
- $m(\langle U, C\rangle, f) = |\mathcal{C}_f|$
- OPT = max.

Complexity of source problem: MAXIMUM 2SAT DECISION is NP-complete, even if each variable occurs in at most three clauses [189].

Partial order on truth assignments

Minimaximal problem name: MINIMUM MAXIMAL 2SAT.

Complexity of minimaximal problem: The complexity of MINIMUM MAXIMAL 2SAT is resolved by the following theorem. The proof involves a transformation from the problem 3SAT, which is defined as follows:

> *Name:* 3SAT.
> *Instance:* Set U of variables and a set C of clauses over U, where $|C_i| = 3$, for all $C_i \in C$.
> *Question:* Is there a truth assignment f such that $f(C) = T$?

The problem AT MOST 3SAT, which is similar to 3SAT, but distinct in that each clause in the input has size *at most* 3, is shown to be NP-complete by Karp [140]. A simple transformation from AT MOST 3SAT, involving the addition of new variables and clauses corresponding to any clause of size one or two, gives NP-completeness for 3SAT. See, for example, Papadimitriou and Steiglitz [179, p.359].

Theorem 7.7.2 MINIMUM MAXIMAL 2SAT DECISION *is NP-complete.*

Proof: By Proposition 7.7.1, MINIMUM MAXIMAL 2SAT DECISION is in NP. To show NP-hardness, we give a transformation from 3SAT, defined above. Suppose that U (a set of variables) and $C = \{C_1, C_2, \ldots, C_m\}$ (a set of clauses), where $|C_i| = 3$ $(1 \le i \le m)$, is an instance of 3SAT. Assume that $C_i = \{a_i, b_i, c_i\}$ $(1 \le i \le m)$. Construct the following instance of MINIMUM MAXIMAL 2SAT DECISION. Let $U' = U \cup \{d_i, e_i, f_i : 1 \le i \le m\}$ be a set of variables, where the d_i, e_i and f_i are new variable names $(1 \le i \le m)$, and let C' be a set of $10m$ clauses, where

$$C' = \bigcup_{i=1}^{m} \left\{ \begin{array}{l} \{\overline{a_i}, d_i\}, \{\overline{b_i}, d_i\}, \{\overline{c_i}, d_i\}, \\ \{d_i, e_i\}, \{d_i, \overline{e_i}\}, \{\overline{d_i}, e_i\}, \{\overline{d_i}, \overline{e_i}\}, \\ \{f_i, a_i\}, \{f_i, b_i\}, \{f_i, c_i\} \end{array} \right\}.$$

Set $K = 6m$, and denote by C_i' the set of ten clauses of C' involving literals with subscript i $(1 \le i \le m)$. We claim that there is a truth assignment satisfying C if and only if there exists a maximal truth assignment satisfying at most K clauses of C' simultaneously.

For, let f be a truth assignment such that $f(C) = T$. Then, for each i $(1 \le i \le m)$, f satisfies at least one literal from C_i. Define a truth assignment g on U' by setting

$g(u) = f(u)$, for all $u \in U$, and by setting $g(d_i) = g(e_i) = g(f_i) = F$, for every i $(1 \le i \le m)$. By the symmetry of the construction of C', for each i $(1 \le i \le m)$, we may consider the following cases:

1. $g(a_i) = F, g(b_i) = F, g(c_i) = T$.
2. $g(a_i) = F, g(b_i) = T, g(c_i) = T$.
3. $g(a_i) = T, g(b_i) = T, g(c_i) = T$.

In each case, it may be verified that g satisfies exactly six clauses from each C'_i, and each of the variables $a_i, b_i, c_i, d_i, e_i, f_i$ appears in at least one of these clauses. Hence $U'(\mathcal{C}'_g) = U'$, so that g is maximal by Proposition 7.7.1. Also, g satisfies exactly K clauses of C' simultaneously.

Conversely, suppose that g is a maximal truth assignment satisfying at most K clauses of C' simultaneously. Suppose that i $(1 \le i \le m)$ is given. If g satisfies one of cases 1,2,3 above, then it may be verified that in each case, g satisfies at least six clauses from C'_i. By symmetry, the only additional case we need consider is that $g(a_i) = g(b_i) = g(c_i) = F$. In this case, g must satisfy nine clauses from C'_i, i.e., $g(f_i) = T$, or else g is not maximal. Since this would satisfy too many clauses simultaneously, we conclude that this case cannot occur. Thus, we may define a truth assignment f on U by setting $f(u) = g(u)$, for all $u \in U$. Since g satisfies at least one of a_i, b_i, c_i for each i $(1 \le i \le m)$, then $f(C) = T$ as required. ∎

7.7.3 One-in-three satisfiability

In this section, we consider a source optimisation problem based on the following decision problem:

> *Name:* ONE-IN-AT-MOST-THREE SAT.
> *Instance:* Set U of variables and a set C of clauses over U, such that, for each $C_i \in C$, $|C_i| \le 3$ and C_i contains no negated variable of U.
> *Question:* Is there a truth assignment f such that f satisfies exactly one variable from each $C_i \in C$?

In the context of ONE-IN-AT-MOST-THREE SAT, variables and literals mean one and the same thing, since no clause contains a negated variable. Thus, for the remainder of this section, we use the term 'variable' when 'literal' may otherwise have been used.

Schaefer [192] studies a problem that is similar to ONE-IN-AT-MOST-THREE SAT, but is distinct in that a clause is a multiset in his definition, so that a variable can occur more than once within a clause, where each clause has size three. In the context of Schaefer's definition, multiple occurrences of a variable within a clause are significant, when considering satisfaction by a truth assignment. However, it turns out that the problem under his definition is reducible to the problem within our logical framework, which may be seen as follows.

Lemma 7.7.3 ONE-IN-AT-MOST-THREE SAT *under Schaefer's definition [192] is reducible to* ONE-IN-AT-MOST-THREE SAT *under our definition.*

Proof: Suppose that U (a set of variables) and $C = \{C_1, C_2, \ldots, C_m\}$ (a set of clauses over U) is an instance of ONE-IN-AT-MOST-THREE SAT under Schaefer's definition, where a clause of C is a multiset of size three, containing no negated variable of U. We construct a set C' of clauses (where a clause of C' is a set of members of U', of size at most three) as follows.

We may assume that no variable $u \in U$ occurs three times in any clause of C, for then the given instance $\langle U, C\rangle$ has a 'no' answer. Corresponding to any clause $C_i \in C$ which contains two occurrences of some variable u, and one occurrence of some variable v, we add the two clauses $C'_{i,1}, C'_{i,2}$ to C', where $C'_{i,1} = \{u, v\}$ and $C'_{i,2} = \{v\}$. Corresponding to any clause C_i which has no multiple occurrences of any variable, we let $C'_i = C_i$, and add C'_i to C'. Our constructed instance of ONE-IN-AT-MOST-THREE SAT therefore consists of the variables $U' = U$, and the clauses C'.

Now a truth assignment f satisfies exactly one variable occurrence from every clause in C if and only if f satisfies exactly one variable from every clause in C'. For, suppose that f satisfies exactly one variable occurrence from each clause in C. Let i $(1 \le i \le m)$ be given. If C_i contains no multiple occurrences of any variable, then clearly f satisfies exactly one variable from C'_i. Now suppose that some variable u appears twice in C_i, and some variable v appears once in C_i. Then $f(u) = F$ and $f(v) = T$. Hence f satisfies exactly one variable from each of $C'_{i,1}, C'_{i,2}$. Thus f satisfies exactly one variable from every clause in C'.

Conversely, suppose that f satisfies exactly one variable from each clause in C'. Let i $(1 \le i \le m)$ be given. If C_i contains no multiple occurrences of any variable, then clearly f satisfies exactly one variable occurrence from C_i. Now suppose that some variable u appears twice in C_i, and some variable v appears once in C_i. As f satisfies exactly one variable from $C'_{i,2}$, then $f(v) = T$, which implies that $f(u) = F$, since f satisfies exactly one variable from $C'_{i,1}$. Thus f satisfies exactly one variable occurrence from C_i. ∎

Results from [192], together with Lemma 7.7.3, prove the NP-completeness of ONE-IN-AT-MOST-THREE SAT. Now consider a restricted version of this problem, called ONE-IN-THREE 3SAT, in which each clause has size exactly three. It turns out that the restricted problem is also NP-complete, as the following lemma demonstrates. This result will be utilised later in this section, when we consider the complexity of a minimaximal optimisation problem, to be defined.

Lemma 7.7.4 ONE-IN-THREE 3SAT *is NP-complete.*

Proof: Clearly, ONE-IN-THREE 3SAT is in NP. To show NP-hardness, we give a simple transformation from ONE-IN-AT-MOST-THREE SAT. Suppose that U (a set of variables) and $C = \{C_1, C_2, \ldots, C_m\}$ (a set of clauses), where $|C_i| \le 3$ and C_i contains no negated variable of U $(1 \le i \le m)$, is an instance of ONE-IN-AT-MOST-THREE SAT. We construct a set of clauses C' as follows. Corresponding to any clause C_i of size one, where $C_i = \{a_i\}$, we add the three clauses of C_i' to C', where

$$C_i' = \{\{a_i, p_i, q_i\}, \{a_i, p_i, r_i\}, \{a_i, q_i, r_i\}\}.$$

Corresponding to any clause C_i of size two, where $C_i = \{a_i, b_i\}$, we add the three clauses of C_i' to C', where

$$C_i' = \{\{a_i, b_i, x_i\}, \{a_i, b_i, y_i\}, \{x_i, y_i, z_i\}\}.$$

In both cases, the $p_i, q_i, r_i, x_i, y_i, z_i$ are new variable names. Finally, for any clause C_i of size three, where $C_i = \{a_i, b_i, c_i\}$, we add C_i to C' (let C_i' contain the clause C_i in this case). Our constructed instance of ONE-IN-THREE 3SAT therefore consists of U', which contains the variables in U plus the new variable names introduced, and the clauses C'.

There is a truth assignment f defined on U, satisfying exactly one variable from every clause in C, if and only if there is a truth assignment g defined on U', satisfying exactly one variable from every clause in C'. For, suppose that f satisfies exactly one variable from each clause in C. We construct a truth assignment g defined on U'. Let $g(u) = f(u)$ for all $u \in U$. Set $g(p_i) = g(q_i) = g(r_i) = F$ for all i such that $|C_i| = 1$. Set $g(x_i) = g(y_i) = F$ and $g(z_i) = T$ for all i such that $|C_i| = 2$. Now let i $(1 \leq i \leq m)$ be given. It may be verified that in each of the three cases $|C_i| = 1, 2, 3$, g satisfies exactly one variable from each clause in C_i'. Hence g satisfies exactly one variable from every clause in C'.

Conversely, suppose that g satisfies exactly one variable from each clause in C'. We construct a truth assignment f defined on U, by letting $f(u) = g(u)$ for all $u \in U$. Now let i $(1 \leq i \leq m)$ be given. Suppose that $C_i = \{a_i\}$ for some a_i, and that $g(a_i) = F$. Then $\{a_i, p_i, q_i\} \in C'$ implies that, either (i) $g(p_i) = T$ and $g(q_i) = F$, or (ii) $g(p_i) = F$ and $g(q_i) = T$. In case (i), $\{a_i, q_i, r_i\} \in C'$ implies that $g(r_i) = T$. But we reach a contradiction, since $\{a_i, p_i, r_i\} \in C'$. In case (ii), $\{a_i, p_i, r_i\} \in C'$ implies that $g(r_i) = T$. But we again reach a contradiction, since $\{a_i, q_i, r_i\} \in C'$. Thus $f(a_i) = T$. Now suppose that $C_i = \{a_i, b_i\}$ for some a_i, b_i, and $g(a_i) = g(b_i) = F$. Then $\{a_i, b_i, x_i\} \in C'$ implies that $g(x_i) = T$, and $\{a_i, b_i, y_i\} \in C'$ implies that $g(y_i) = T$. But again we reach a contradiction, since $\{x_i, y_i, z_i\} \in C'$. Hence exactly one of $f(a_i) = T, f(b_i) = T$ holds. Finally, if $C_i = \{a_i, b_i, c_i\}$ for some a_i, b_i, c_i, then clearly f satisfies exactly one variable from C_i. Hence f satisfies exactly one variable from every clause in C. ∎

We now consider a source optimisation problem Π that is based on the decision problem ONE-IN-THREE 3SAT. Both Π, and the minimaximal optimisation problem derived from Π using the partial order on truth assignments, feature in the proof of Theorem 8.3.1.

Given a set of variables U and a set of m clauses C over U, consider the problem of maximising k, such that there exists a truth assignment f which simultaneously satisfies exactly one variable from k clauses of C. Given such a truth assignment f, there are a variety of possibilities for specifying how many variables f should satisfy from the remaining $m - k$ clauses of C. We choose to demand that f should satisfy *none* of the variables from the remaining $m - k$ clauses. This resolution is reflected in the following source optimisation problem definition.

Source problem: MAXIMUM ONE-IN-THREE 3SAT=$\langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, where

- $\mathcal{I} = \left\{ \langle U, C \rangle : \begin{array}{l} C \text{ is a w.f. over } U, |C_i| = 3, \text{ and } C_i \text{ contains} \\ \text{no negated variable of } U, \text{ for all } C_i \in C \end{array} \right\}$
- $\mathcal{U}(G) = \{f : f \text{ is a truth assignment for } U\}$
- $\pi(\langle U, C \rangle, f) \Leftrightarrow \forall\, C_i \in C \bullet |\{v \in C_i : f(v) = T\}| \leq 1$
- $m(\langle U, C \rangle, f) = |\mathcal{C}_f|$
- OPT $=$ max.

Complexity of source problem: MAXIMUM ONE-IN-THREE 3SAT DECISION is NP-complete, for if we restrict the target value to satisfy $K = |C|$, then we obtain ONE-IN-THREE 3SAT.

Partial order on truth assignments

Minimaximal problem name: MINIMUM MAXIMAL ONE-IN-THREE 3SAT.

Complexity of minimaximal problem: The complexity of MINIMUM MAXIMAL ONE-IN-THREE 3SAT is resolved by the following theorem.

Theorem 7.7.5 MINIMUM MAXIMAL ONE-IN-THREE 3SAT DECISION *is NP-complete.*

Proof: By Proposition 7.7.1, MINIMUM MAXIMAL ONE-IN-THREE 3SAT DECISION is in NP. To show NP-hardness, we give a transformation from ONE-IN-THREE 3SAT, defined above. Suppose that U (a set of variables) and $C = \{C_1, C_2, \ldots, C_m\}$ (a set of clauses), where $C_i = \{a_i, b_i, c_i\}$ and C_i contains no negated variable of U $(1 \leq i \leq m)$, is an arbitrary instance of ONE-IN-THREE 3SAT. Construct the following instance of MINIMUM MAXIMAL ONE-IN-THREE 3SAT DECISION. Let $U' = U \cup \{d_1, d_2, \ldots, d_m\}$ be a set of variables, where the d_i are new variable names $(1 \leq i \leq m)$, and let C' be a set of $3m$ clauses, where

$$C' = \bigcup_{i=1}^{m} \{\{a_i, b_i, d_i\}, \{a_i, c_i, d_i\}, \{b_i, c_i, d_i\}\}.$$

Set $K = 2m$, and denote by C'_i the set of three clauses of C' involving literals with subscript i $(1 \leq i \leq m)$. Clearly $|C'_i| = 3$ and C'_i contains no negated variable of U' $(1 \leq i \leq m)$. We claim that there is a truth assignment satisfying exactly one literal from each clause of C if and only if there exists a maximal truth assignment simultaneously satisfying exactly one literal from k clauses of C', and no literals from the other $3m - k$ clauses of C', where $k \leq K$.

For, let f be a truth assignment such that f satisfies exactly one literal from every clause of C. Then for each i $(1 \leq i \leq m)$, exactly one of the cases $f(a_i) = T$, $f(b_i) = T$, $f(c_i) = T$ holds. Define a truth assignment g on U' by setting $g(u) = f(u)$, for all $u \in U$, and by setting $g(d_i) = F$, for every i $(1 \leq i \leq m)$. Then it may be verified that for each i $(1 \leq i \leq m)$, g satisfies exactly two clauses from each C'_i, and each one of the variables a_i, b_i, c_i, d_i appears in at least one of these clauses. Hence $U'(\mathcal{C}'_g) = U'$, so that g is maximal by Proposition 7.7.1. Also, g satisfies one literal from exactly $K = 2m$ clauses of C', and no literals from the remaining $3m - K = m$ clauses.

Conversely, suppose that g is a maximal truth assignment satisfying exactly one literal from k clauses of C', and no literals from the other $3m - k$ clauses of C', where $k \leq K$. Suppose that i $(1 \leq i \leq m)$ is given. Clearly, at most one of $g(a_i) = T, g(b_i) = T, g(c_i) = T$ holds, or else two literals from some clause are true. Moreover, if exactly one of these cases does hold, then g satisfies at least two clauses from C_i. The only additional case we need consider is that $g(a_i) = g(b_i) = g(c_i) = F$. In this case, g must satisfy all three clauses from C_i', i.e., $g(d_i) = T$, or else g is not maximal. Since this would satisfy too many clauses simultaneously, we conclude that this case cannot occur. Thus, we may define a truth assignment f on U by setting $f(u) = g(u)$, for all $u \in U$. Then g satisfies exactly one of a_i, b_i, c_i, for each i $(1 \leq i \leq m)$, as required. ■

7.8 Summary of complexity results for non graph-theoretic optimisation problems considered in this chapter

Table 7.1 summarises the complexity results of the optimisation problems considered in this chapter. In a table entry, entry, 'N' denotes NP-completeness for the decision version of the relevant optimisation problem (NP-hardness in the case of MAXIMUM $\prec_b$-MINIMAL BIN PACKING), and 'P' denotes polynomial-time solvabilty for the optimisation problem. Appropriate references are indicated. An asterisk indicates that the result is new and the proof is presented here for the first time, and a question mark indicates that the corresponding problem is open.

The results presented in this chapter demonstrate that many interesting examples of minimaximal and maximinimal non graph-theoretic optimisation problems may be formulated using the framework of Definition 2.3.5. In addition to the open problems indicated by Table 7.1, there remains much scope for the further study of non graph-theoretic minimaximal and maximinimal optimisation problems, formulated from non graph-theoretic source optimisation problems and partial orders not considered in this chapter.

Some general conclusions, which refer to certain problems studied in this chapter, are drawn in Section 9.6.

Source problem name	Source problem complexity	Partial order	Minimaximal/ maximinimal problem complexity
LONGEST PATH	N[92]	$\overline{\ll}$	P(*)
LONGEST PATH	N[92]	$\ll$	N(*)
UNCONSTRAINED LONGEST PATH	N[92]	$\overline{\ll}$	N(*)
UNCONSTRAINED LONGEST PATH	N[92]	$\ll$	N(*)
MAXIMUM 3D-MATCHING	N[92]	$\subset$	N(*)
MINIMUM TEST SET	N[92]	$\subset$	N(*)
MINIMUM BIN PACKING	N[92]	$\prec_a$	N(*)
MINIMUM BIN PACKING	N[92]	$\prec_b$	N(*)
LONGEST COMMON SUBSEQUENCE	N[92]	$\ll$	N[81]
LONGEST COMMON SUBSEQUENCE $(\lvert\Sigma\rvert = 2)$	N[161]	$\ll$	?
SHORTEST COMMON SUPERSEQUENCE	N[92]	$\ll$	N[81]
SHORTEST COMMON SUPERSEQUENCE $(\lvert\Sigma\rvert = 2)$	N[188]	$\ll$	?
SHORTEST COMMON SUPERSEQUENCE $(m = 2)$	N[205]	$\ll$	P[81]
LONGEST COMMON SUBSTRING	P[103, §7.6]	$\overline{\ll}$	P(*)
SHORTEST COMMON SUPERSTRING	N[92]	$\overline{\ll}$	?
MAXIMUM KNAPSACK	N[92]	$\subset$	N(*)
MAXIMUM 2SAT	N[93]	$\prec_t$	N(*)
MAXIMUM ONE-IN-THREE 3SAT	N[192, (*)]	$\prec_t$	N(*)

Table 7.1: Summary of complexity results for non graph-theoretic optimisation problems considered in this chapter.

Chapter 8

Minimaximal and maximinimal reductions

8.1 Introduction

This chapter is concerned with a restricted form of Turing reduction relating to optimisation problems. Given a source optimisation problem Π, the reduction yields complexity results for both Π and minimaximal or maximinimal optimisation problems that may be derived from Π using appropriate partial orders.

A fundamental concept for reasoning about Turing reduction from an optimisation problem, $\Pi_1 = \langle \mathcal{I}_1, \mathcal{U}_1, \pi_1, m_1, \text{OPT}_1 \rangle$, to another, $\Pi_2 = \langle \mathcal{I}_2, \mathcal{U}_2, \pi_2, m_2, \text{OPT}_2 \rangle$, is the notion of a *hypothetical subroutine* S that solves Π_2, given an instance $x' \in \mathcal{I}_2$. The reduction uses a polynomial number of calls to S, in order to solve a given instance x of Π_1. Thus if S is a polynomial-time algorithm, then there is a polynomial-time algorithm to solve Π_1. As a consequence, Π_1 is NP-hard implies that Π_2 is NP-hard, and Π_2 is in P implies that Π_1 is in P.

In this chapter, we consider a restricted form of Turing reduction that uses only *one* call to this hypothetical subroutine S. Given an instance x of Π_1, suppose that $f(x)$ is an instance of Π_2, and $S(f(x))$ returns $y \in \mathcal{F}_2^*(f(x))$, a globally optimal solution to Π_2. Suppose further that $g(f(x), \cdot)$ is a function (whose domain will be defined fully in due course) that maps y to a globally optimal solution to Π_1, i.e., $g(f(x), y) \in \mathcal{F}_1^*(x)$. Then if f, g can be computed in polynomial time, they constitute a Turing reduction from Π_1 to Π_2.

Placing extra constraints on f and g allows us to construct a reduction that is relevant to our study of minimaximal and maximinimal optimisation problems. For $i = 1, 2$, let $\prec_i^{x_i}$ be a partial order defined on the feasible solutions $\mathcal{F}_i(x_i)$ for a given instance x_i of Π_i. Let Π_i' be the minimaximal or maximinimal optimisation problem obtained from Π_i and $\prec_i$, using Definition 2.3.5. Suppose that, in addition to satisfying the properties of the previous paragraph, the function g maps a $\prec_2^{f(x)}$-optimal feasible solution of Π_2 to a $\prec_1^x$-optimal feasible solution for a given instance x of Π_1. Suppose further that g preserves the measures of these feasible solutions in such a way that a globally optimal solution of Π_2' maps to a globally optimal solution of Π_1', i.e., $y \in (\mathcal{F}_2')^*(f(x))$ implies that $g(f(x), y) \in (\mathcal{F}_1')^*(x)$. Then f, g constitutes a Turing reduction from Π_1' to Π_2'.

We call a polynomial reduction satisying the properties in the two previous paragraphs an *MM-reduction* (to be defined formally in due course), standing for *minimaximal / maximinimal reduction*.

For example, the transformation described in Section 4.2.6, from MINIMUM VERTEX COVER DECISION to MAXIMUM INDEPENDENT SET DECISION gives rise to an MM-reduction. This transformation is based on the result of Proposition 4.2.8, namely that, given a graph $G = (V, E)$ and a subset V' of V, V' is a vertex cover for G if and only if $V \backslash V'$ is an independent set for G'. When we consider the partial orders of set inclusion, defined on the feasible solutions of both problems, it is a consequence of Lemma 4.2.9 that this transformation also constitutes a Turing reduction from MAXIMUM MINIMAL VERTEX COVER to MINIMUM MAXIMAL INDEPENDENT SET[1]. For this result states that, given a graph $G = (V, E)$ and a subset V' of V, V' is a *minimal* vertex cover for G if and only if $V \backslash V'$ is a *maximal* independent set for G'.

This chapter is devoted to the study of MM-reductions and is organised as follows. In Section 8.2, we define formally an MM-reduction, and prove that the reduction does what is required. In Section 8.3, we provide further examples of MM-reductions, some of which correspond to polynomial reductions that have already appeared in the literature. Finally, in Section 8.4, we present some closing remarks.

8.2 Definitions and general results for MM-reductions

In this section, we define formally an MM-reduction, and present some results that are consequences of this definition. Recall that a minimaximal or maximinimal optimisation problem is derived from a source optimisation problem Π, using a partial order $\prec^x$ defined on the feasible solutions of Π, given an instance x of Π. Thus, the MM-reduction will be defined as a transformation from one pair, $\langle \Pi_1, \prec_1 \rangle$, to another pair, $\langle \Pi_2, \prec_2 \rangle$, where $\Pi_i, \prec_i^{x_i}$ are as defined in the previous sentence ($i = 1, 2$ and x_i is an instance of Π_i). As in the case of PLS-reductions [136], the MM-reduction is concerned with preserving the local optimality of feasible solutions. However, in order to be a Turing reduction from a source optimisation problem, Π_1, to another, Π_2, and a Turing reduction from a minimaximal or maximinimal version of Π_1 to a minimaximal or maximinimal version of Π_2, it is clear that an MM-reduction has to preserve more structure than that maintained by a PLS-reduction.

Definition 8.2.1 Let $\Pi_1 = \langle \mathcal{I}_1, \mathcal{U}_1, \pi_1, m_1, \text{OPT}_1 \rangle$ and $\Pi_2 = \langle \mathcal{I}_2, \mathcal{U}_2, \pi_2, m_2, \text{OPT}_2 \rangle$ be two optimisation problems. Suppose that, for $i = 1, 2$, and for any $x_i \in \mathcal{I}_i$, $\prec_i^{x_i}$ is a partial order defined on $\mathcal{F}_i(x_i)$, satisfying POMM with respect to Π_i. Let $\Pi_1' = \langle \mathcal{I}_1', \mathcal{U}_1', \pi_1', m_1', \text{OPT}_1' \rangle$

[1] Of course, the implicit reduction also gives rise to a Turing reduction from MAXIMUM INDEPENDENT SET to MINIMUM VERTEX COVER, and from MINIMUM MAXIMAL INDEPENDENT SET to MAXIMUM MINIMAL VERTEX COVER, so that both pairs of problems are NP-equivalent.

and $\Pi'_2 = \langle \mathcal{I}'_2, \mathcal{U}'_2, \pi'_2, m'_2, \text{OPT}'_2 \rangle$ be the minimaximal or maximinimal optimisation problems obtained from $\Pi_1, \prec_1$, and from $\Pi_2, \prec_2$ respectively, using Definition 2.3.5. Then $\langle \Pi_1, \prec_1 \rangle$ is *MM-reducible* to $\langle \Pi_2, \prec_2 \rangle$, written $\langle \Pi_1, \prec_1 \rangle \alpha_{MM} \langle \Pi_2, \prec_2 \rangle$, if there exist two functions f, g such that:

1. For any $x \in \mathcal{I}_1$, $f(x) \in \mathcal{I}_2$ is computable in polynomial time.
2. For any $x \in \mathcal{I}_1$ and for any $y \in \mathcal{F}_2(f(x))$, $g(f(x), y) \in \mathcal{F}_1(x)$ is computable in polynomial time.
3. For any $x \in \mathcal{I}_1$ and for any $y \in \mathcal{F}_2(f(x))$,
$$y \in \mathcal{F}_2^*(f(x)) \Rightarrow g(f(x), y) \in \mathcal{F}_1^*(x).$$
4. For any $x \in \mathcal{I}_1$ and for any $y \in \mathcal{F}_2(f(x))$,
$$y \in (\mathcal{F}'_2)^*(f(x)) \Rightarrow g(f(x), y) \in (\mathcal{F}'_1)^*(x).$$

The pair of functions $\langle f, g \rangle$ is said to be an *MM-reduction* from $\langle \Pi_1, \prec_1 \rangle$ to $\langle \Pi_2, \prec_2 \rangle$. ∎

Properties 1, 2 and 3 of Definition 8.2.1 imply that $\langle f, g \rangle$ constitutes a restricted form of Turing reduction from Π_1 to Π_2, as is demonstrated by the next result.

Theorem 8.2.2 *Let* $\Pi_1 = \langle \mathcal{I}_1, \mathcal{U}_1, \pi_1, m_1, \text{OPT}_1 \rangle$ *and* $\Pi_2 = \langle \mathcal{I}_2, \mathcal{U}_2, \pi_2, m_2, \text{OPT}_2 \rangle$ *be two optimisation problems. Suppose that, for* $i = 1, 2$, *and for any* $x_i \in \mathcal{I}_i$, $\prec_i^{x_i}$ *is a partial order defined on* $\mathcal{F}_i(x_i)$, *satisfying POMM with respect to* Π_i. *Suppose that* $\langle \Pi_1, \prec_1 \rangle \alpha_{MM} \langle \Pi_2, \prec_2 \rangle$. *Then* $\Pi_1 \alpha_T \Pi_2$.

Proof: Let $\langle f, g \rangle$ be an MM-reduction from $\langle \Pi_1, \prec_1 \rangle$ to $\langle \Pi_2, \prec_2 \rangle$, and let x be an instance of Π_1. Suppose that, for any instance x' of Π_2, $S(x')$ is a hypothetical subroutine that finds some $y' \in \mathcal{F}_2^*(x')$ in polynomial time. Consider the instance $f(x)$ of Π_2, and let $y \in \mathcal{F}_2^*(f(x))$ be the feasible solution returned by $S(f(x))$. Then $g(f(x), y) \in \mathcal{F}_1^*(x)$. Since $g(f(x), y)$ has been computed in polynomial time, the result follows. ∎

Corollary 8.2.3 *Let* Π_1, Π_2 *and* $\prec_1, \prec_2$ *be defined as in Theorem 8.2.2, and suppose that* $\langle \Pi_1, \prec_1 \rangle \alpha_{MM} \langle \Pi_2, \prec_2 \rangle$. *Then*

1. Π_1 *is NP-hard implies that* Π_2 *is NP-hard.*
2. Π_2 *is in P implies that* Π_1 *is in P.*

Properties 1, 2 and 4 of Definition 8.2.1 also imply that $\langle f, g \rangle$ constitutes a restricted form of Turing reduction from Π'_1 to Π'_2, as we now show.

Theorem 8.2.4 *Let* $\Pi_1 = \langle \mathcal{I}_1, \mathcal{U}_1, \pi_1, m_1, \text{OPT}_1 \rangle$ *and* $\Pi_2 = \langle \mathcal{I}_2, \mathcal{U}_2, \pi_2, m_2, \text{OPT}_2 \rangle$ *be two optimisation problems. Suppose that, for* $i = 1, 2$, *and for any* $x_i \in \mathcal{I}_i$, $\prec_i^{x_i}$ *is a partial order defined on* $\mathcal{F}_i(x_i)$, *satisfying POMM with respect to* Π_i. *Let* $\Pi'_1 = \langle \mathcal{I}'_1, \mathcal{U}'_1, \pi'_1, m'_1, \text{OPT}'_1 \rangle$ *and* $\Pi'_2 = \langle \mathcal{I}'_2, \mathcal{U}'_2, \pi'_2, m'_2, \text{OPT}'_2 \rangle$ *be the minimaximal or maximinimal optimisation problems obtained from* $\Pi_1, \prec_1$, *and from* $\Pi_2, \prec_2$ *respectively, using Definition 2.3.5. Suppose that* $\langle \Pi_1, \prec_1 \rangle \alpha_{MM} \langle \Pi_2, \prec_2 \rangle$. *Then* $\Pi'_1 \alpha_T \Pi'_2$.

Proof: Let $\langle f, g\rangle$ be an MM-reduction from $\langle \Pi_1, \prec_1\rangle$ to $\langle \Pi_2, \prec_2\rangle$, and let x be an instance of Π_1'. Suppose that, for any instance x' of Π_2', $S(x')$ is a hypothetical subroutine that finds some $y' \in (\mathcal{F}_2')^*(x')$ in polynomial time. Consider the instance $f(x)$ of Π_2', and let $y \in (\mathcal{F}_2')^*(f(x))$ be the feasible solution returned by $S(f(x))$. Then $g(f(x), y) \in (\mathcal{F}_1')^*(x)$. Since $g(f(x), y)$ has been computed in polynomial time, the result follows. ∎

Corollary 8.2.5 *Let* $\Pi_1, \Pi_2, \prec_1, \prec_2$ *and* Π_1', Π_2' *be defined as in Theorem 8.2.4, and suppose that* $\langle \Pi_1, \prec_1\rangle \alpha_{MM} \langle \Pi_2, \prec_2\rangle$. *Then*

1. Π_1' *is NP-hard implies that* Π_2' *is NP-hard.*
2. Π_2' *is in P implies that* Π_1' *is in P.*

Note that we do not attempt to introduce a new complexity class, based on a notion of *MM-completeness*. The objective of Definition 8.2.1 is to define sufficient conditions under which a Turing reduction provides us with an extra complexity result. As well as formulating a new reduction that is an MM-reduction, we have found several reductions in the literature that are MM-reductions; doubtless there are many more. Some examples of MM-reductions are given in the following section.

8.3 Examples of MM-reductions

In the following subsections, we give a number of examples of MM reductions from one ⟨optimisation problem, partial order⟩ pair to another. The MM-reductions that we present are summarised in Figure 8.1. If the pair $\langle \Pi_1, \prec_1\rangle$ is the parent of the pair $\langle \Pi_2, \prec_2\rangle$ in the tree shown, then an MM-reduction is given from $\langle \Pi_1, \prec_1\rangle$ to $\langle \Pi_2, \prec_2\rangle$ here. Throughout the section, we find it convenient to use the abbreviations MAX CLIQUE, MAX IND SET and MIN VERTEX COVER for the problems MAXIMUM CLIQUE, MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER, respectively. When reasoning about the local optimality of a feasible solution, the partial order concerned should be clear from the context, since the pairs $\langle \Pi_1, \prec_1\rangle$ and $\langle \Pi_2, \prec_2\rangle$ involved in the MM-reduction will be defined. Thus, in the forthcoming sections, the terms 'maximal' and 'minimal' may not always be prefixed with a partial order symbol.

8.3.1 One-in-three 3SAT to clique

In this section, we formulate a reduction from MAXIMUM ONE-IN-THREE 3SAT to MAX CLIQUE. The components of the former problem were defined in Section 7.7.3, and those of the latter problem were defined in Section 4.3.2.

Theorem 8.3.1 $\langle$MAXIMUM ONE-IN-THREE 3SAT$,\prec_t\rangle \alpha_{MM} \langle$MAX CLIQUE$, \subset\rangle$.

Proof: Suppose that U (a set of variables) and $C = \{C_1, C_2, \ldots, C_m\}$ (a set of clauses over U), where $|C_i| = 3$ and C_i contains no negated variable of U $(1 \le i \le m)$, is an instance x of MAXIMUM ONE-IN-THREE 3SAT. Due to the restriction on the instance, the

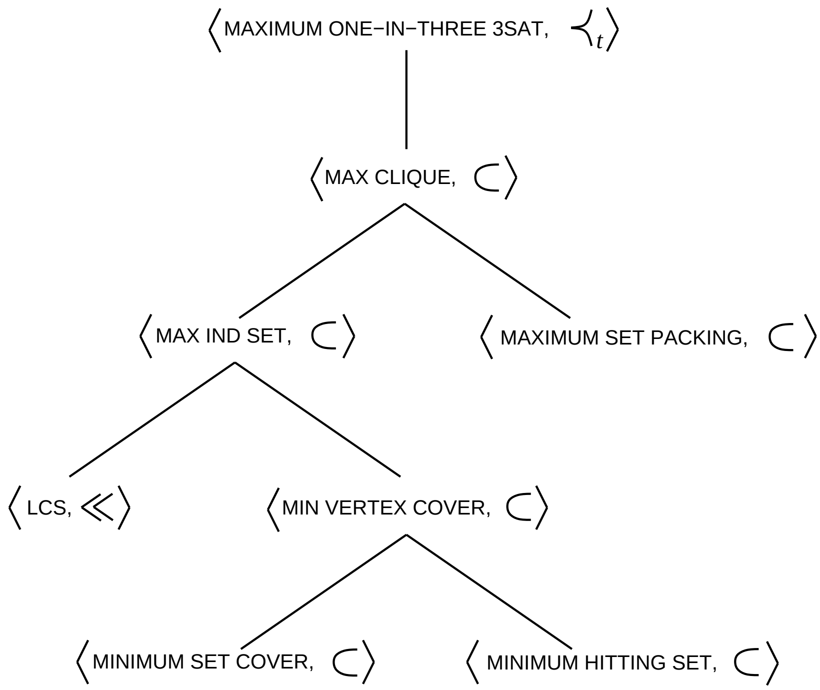

Figure 8.1: Tree structure, showing MM-reductions presented in this chapter.

word 'variable' is used when the term 'literal' would otherwise be used, for the remainder of the proof. Consider the following reduction:

$$
\begin{array}{ll}
f : \langle U, C\rangle \mapsto G, \text{ where} & G = (V, E) \\
& V = \{(u,i) : u \in U \wedge 1 \le i \le m \wedge u \in C_i\} \\
& E = \left\{ \{(u,i),(v,j)\} : \begin{array}{l} (u,i) \in V \wedge (v,j) \in V \wedge i \ne j \wedge \\ (u \ne v \vee \\ \forall 1 \le r \le m \bullet \{u,v\} \not\subseteq C_r) \end{array} \right\} \\
g(G,\cdot) : S' \mapsto \xi, \text{ where} & \xi(u) = T, \ (u,i) \in S \text{ for some } i \ (1 \le i \le m) \\
& \xi(u) = F, \text{ otherwise} \\
& \text{where } S \supseteq S' \text{ and } S \text{ is a maximal clique in } G.
\end{array}
$$

Define also the following function:

$$h(\langle U, C\rangle, \cdot) : \xi \mapsto \{(u,i) \in V : \xi(u) = T\}.$$

To show that $\langle f, g\rangle$ is an MM-reduction, we prove the following:

1. If ξ is a truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m - k$ clauses of C, then $h(\langle U, C\rangle, \xi)$ is a clique in G of size k.

2. If S' is a clique in G of size k', then $g(G, S')$ is a truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m-k$ clauses of C, where $k \ge k'$.

3. If ξ is a maximal truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m-k$ clauses of C, then $h(\langle U, C\rangle, \xi)$ is a maximal clique in G of size k.

4. If S is a maximal clique in G of size k, then $g(G, S)$ is a maximal truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m-k$ clauses of C.

Proof of (1): Suppose that ξ is a truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m-k$ clauses of C. Define

$$S = \{(u, i) \in V : \xi(u) = T\}.$$

If (u, i) and (v, j) are distinct members of S, then $i \neq j$, so that $|S| = k$, by definition of ξ. Now suppose that $u \neq v$ and $\{u, v\} \subseteq C_r$ for some r $(1 \leq r \leq m)$. Then ξ satisfies two variables of C_r, a contradiction. Thus S is a clique in G.

Proof of (3): Suppose that ξ is a maximal truth assignment, satisfying exactly one variable from k clauses of C, and satisfying no variables from the remaining $m-k$ clauses of C. As in the proof of (1) above, we form a clique S of size k. Suppose that S is not maximal. Then there is some $(v, j) \in V\backslash S$ such that $S \cup \{(v, j)\}$ is a clique. Thus $\xi(v) = F$. Also $\xi(C_j) = F$, for otherwise $\xi(w) = T$ for some $w \in C_j\backslash\{v\}$. Hence $(w, j) \in S$, which contradicts the clique property of $S \cup \{(v, j)\}$. Now let $(u, i) \in S$. Then $i \neq j$. Also $u \neq v$ as $\xi(u) = T$ and $\xi(v) = F$. Thus as $S \cup \{(v, j)\}$ is a clique then $v \notin C_i$. Define a truth assignment ξ' as follows:

$$\begin{aligned} \xi'(w) &= \xi(w), \quad && w \in U\backslash\{v\} \\ \xi'(w) &= T, && w = v. \end{aligned}$$

Thus ξ' agrees with ξ on the variables of $\mathcal{C}_\xi$, since $v \notin C_i$, for any $C_i \in \mathcal{C}_\xi$. Also, ξ' satisfies exactly one variable from each clause in $\mathcal{C}_{\xi'}$, and no variables from each clause in $C\backslash\mathcal{C}_{\xi'}$. Since the inclusion $\mathcal{C}_\xi \cup \{C_j\} \subseteq \mathcal{C}_{\xi'}$ holds, $\xi \prec_t^x \xi'$, which contradicts the maximality of ξ. Thus S is maximal.

Proof of (2): Suppose that S' is a clique in G, where $|S| = k'$. Form S, a maximal clique in G, where $S \supseteq S'$, by possibly adding more vertices to S'. Let $k = |S|$; clearly $k \geq k'$. The remainder of the proof of this case is covered by the proof of (4) below.

Proof of (4): Suppose that S is a maximal clique in G, where $|S| = k$. If (u, i) and (v, j) are distinct members of S, then $i \neq j$. Let $(u, i) \in S$, and let $v \in C_i\backslash\{u\}$. Then $(v, j) \notin S$ for any j $(1 \leq j \leq m)$, for otherwise $\{(u, i), (v, j)\} \notin E$ as $\{u, v\} \subseteq C_i$, contradicting the clique property of S.

Now suppose that there is some l $(1 \leq l \leq m)$ such that $(w, l) \in V\backslash S$ for all $w \in C_l$. Suppose further that there is some $(v, l) \in V\backslash S$ such that $(v, j) \in S$ for some j $(1 \leq j \leq m)$. Given any $(u, i) \in S$, if $u \neq v$ then $\{u, v\} \not\subseteq C_r$ for all r $(1 \leq r \leq m)$, for otherwise $\{(u, i), (v, j)\} \notin E$, contradicting the clique property of S. Thus $S \cup \{(v, l)\}$ is a clique, contradicting the maximality of S. Hence no such (v, l) exists.

Thus if we define

$$U' = \{u \in U : (u, i) \in S \text{ for some } i\ (1 \leq i \leq m)\}$$

then we obtain a truth assignment ξ by setting

$$\begin{aligned} \xi(w) &= T, \quad w \in U' \\ \xi(w) &= F, \quad w \in U \backslash U'. \end{aligned}$$

Moreover, ξ satisfies exactly one variable from k clauses of C, and ξ satisfies no variables from the remaining $m - k$ clauses. Now suppose that ξ is not maximal. Then there is some truth assignment ξ' such that $\xi \prec_t^x \xi'$. Let C_j $(1 \leq j \leq m)$ be a clause such that $\xi(C_j) = F$ and $\xi'(C_j) = T$. Then there is some variable $v \in C_j$ such that $\xi(v) = F$ and $\xi'(v) = T$. Hence $(v, j) \notin S$. Now let $(u, i) \in S$. Clearly $i \neq j$, since $\xi(C_i) = T$ and $\xi(C_j) = F$; clearly also $u \neq v$, since $\xi(u) = T$ and $\xi(v) = F$. Also, for any r $(1 \leq r \leq m)$ such that $u \in C_r$, then $C_r \in \mathcal{C}_\xi$. As ξ' agrees with ξ on the variables of $\mathcal{C}_\xi$, then $v \notin C_r$. Thus $S \cup \{(v, j)\}$ is a clique, contradicting the maximality of S. Hence ξ is maximal. ■

8.3.2 Clique to independent set

In this section, we consider a reduction from MAX CLIQUE to MAX IND SET. The components of the latter problem were defined in Section 4.2.6.

Theorem 8.3.2 $\langle$MAX CLIQUE$, \subset\rangle \alpha_{MM} \langle$MAX IND SET$, \subset\rangle$.

Proof: Let $G = (V, E)$ be an instance of MAX CLIQUE. Define the following reduction:

$$\begin{aligned} f &: G \mapsto G^C \\ g(G, \cdot) &: V' \mapsto V'. \end{aligned}$$

The result follows from Proposition 4.3.3 and Lemma 4.3.4. ■

8.3.3 Clique to set packing

In this section, we consider a reduction from MAX CLIQUE to MAXIMUM SET PACKING (whose decision version is problem SP3 of [92]). The latter problem is defined as follows.

Source problem: MAXIMUM SET PACKING$= \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, where

- $\mathcal{I} = \{C : C \text{ is a finite collection of finite sets}\}$
- $\mathcal{U}(C) = \mathbb{P}(C)$
- $\pi(C, C') \Leftrightarrow C'$ is a set packing, i.e., $\forall\, C_i, C_j \in C' \bullet (i \neq j) \Rightarrow C_i \cap C_j = \varnothing$
- $m(C, C') = |C'|$
- OPT $=$ max.

Minimaximal problem name: MINIMUM MAXIMAL SET PACKING.

Theorem 8.3.3 $\langle$MAX CLIQUE, $\subset\rangle \alpha_{MM} \langle$MAXIMUM SET PACKING, $\subset\rangle$.

Proof: Let $G = (V, E)$ be an instance of MAX CLIQUE, where $V = \{v_1, v_2, \ldots, v_n\}$. Define the following reduction (due to Karp [140]):

$$\begin{aligned} f : G \mapsto C, \text{ where } \quad & C = \{C_1, C_2, \ldots, C_n\} \\ & C_i = \{\{v_i, v_j\} : \{v_i, v_j\} \notin E\} \ (1 \leq i \leq n) \\ g(C, \cdot) : \{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\} & \mapsto \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}. \end{aligned}$$

To see that this reduction preserves feasibility and measure, suppose that $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is a clique in G, and $\{v_r, v_s\} \in C_{i_p} \cap C_{i_q}$ for some p, q $(1 \leq p < q \leq k)$ and r, s $(1 \leq r < s \leq n)$. Then without loss of generality $\{v_r, v_s\} \in C_{i_p}$ implies that $r = i_p$, and $\{v_r, v_s\} \in C_{i_q}$ implies that $s = i_q$. Thus $\{v_{i_p}, v_{i_q}\} \notin E$, a contradiction. Hence $\{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\}$ is a set packing.

Conversely, suppose that $\{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\}$ is a set packing and let p, q $(1 \leq p < q \leq k)$ be given. Suppose $\{v_{i_p}, v_{i_q}\} \notin E$. Then $\{v_{i_p}, v_{i_q}\} \in C_{i_p} \cap C_{i_q}$, a contradiction. Thus $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is a clique in G.

By the definitions of f and g, it is easy to see that the reduction also preserves the maximality of feasible solutions. Thus $\langle f, g\rangle$ satisfies Properties 1-4 of Definition 8.2.1, and is therefore an MM-reduction. ■

Corollary 8.3.4 MAXIMUM SET PACKING DECISION *and* MINIMUM MAXIMAL SET PACKING DECISION *are NP-complete, even if* $|C_i| = 3$, *for all* $C_i \in C$.

Proof: Both problems are clearly in NP. From Section 4.2.6, we know that both β_0 and β_0^- are NP-complete for cubic graphs. Thus, from Section 4.3.2, we deduce that both ω and ω^- remain NP-complete for graphs that are regular of degree $n - 3$ (where $n = |V|$). Thus, by setting this restriction on the instance of MAX CLIQUE in Theorem 8.3.3 above, we obtain the stated result. ■

8.3.4 Independent set to LCS

In this section, we consider a reduction from MAX IND SET to LCS. The components of the latter problem were defined in Section 7.5.1.

Theorem 8.3.5 $\langle$MAX IND SET, $\subset\rangle \alpha_{MM} \langle$LCS, $\ll\rangle$.

Proof: Let $G = (V, E)$ be an instance of MAX IND SET, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. The following reduction is due to Fraser et al. [81] (the transformation is similar to one defined by Maier [161], from MIN VERTEX COVER to LCS):

$$\begin{aligned} f : G \mapsto \langle S, \Sigma\rangle, \text{ where } \quad & S = \{s_0, s_1, \ldots, s_m\} \\ & s_0 = \langle v_1 v_2 \ldots v_n\rangle \\ & s_i = \langle v_1 \ldots v_{p-1} v_{p+1} \ldots v_n v_1 \ldots v_{q-1} v_{q+1} \ldots v_n\rangle, \\ & \quad \text{where } e_i = \{v_p, v_q\} \ (1 \leq i \leq m, 1 \leq p < q \leq n) \\ & \Sigma = V \\ g(\langle S, \Sigma\rangle, \cdot) : \langle v_{i_1} v_{i_2} \ldots v_{i_k}\rangle & \mapsto \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}. \end{aligned}$$

To demonstrate that this reduction preserves feasibility and measure, suppose that $I = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is an independent set in G. Assume, without loss of generality, that $i_1 < i_2 < \ldots < i_k$. Now consider the string $s = \langle v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$. Clearly $s \lll s_0$. Now let i $(1 \leq i \leq m)$ be given, and suppose that $e_i = \{v_p, v_q\}$, for some p, q $(1 \leq p < q \leq n)$. If $v_p \notin I$ and $v_q \notin I$, then s is a subsequence of both halves of s_i. If $v_p \in I$ then $v_q \notin I$ by independence, so that s is a subsequence of the second half of s_i. Similarly, if $v_q \in I$ then $v_p \notin I$, so that s is a subsequence of the first half of s_i. Thus $s \lll S$.

Conversely, suppose that $s = \langle v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$ is a common subsequence of S. As $s \lll s_0$, then $i_1 < i_2 < \ldots < i_k$. Consider the set $I = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$. Now let i $(1 \leq i \leq m)$ be given, and suppose that $e_i = \{v_p, v_q\}$, for some p, q $(1 \leq p < q \leq n)$. If $v_p \in I$ and $v_q \in I$ then $v_p v_q \lll s$. But $v_p v_q \not\lll s_i$, so that $s \not\lll s_i$, a contradiction. Thus at most one of v_p, v_q is in I, which implies that I is independent.

The reduction also preserves maximality, which may be seen as follows. Suppose that $I = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is a maximal independent set in G. Assume, without loss of generality, that $i_1 < i_2 < \ldots < i_k$, and let $s = \langle v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$. By the first half of the feasibility argument (using I), $s \lll S$. Suppose that s is not maximal; then there is some s'' such that $s \ll s'' \lll S$. As $s'' \lll s_0$, there is some j $(1 \leq j \leq n)$ such that $\langle v_j \rangle \lll s''$ but $\langle v_j \rangle \not\lll s$. Thus $v_j \notin I$. In addition, there is some string s', such that $s \ll s' \lll s'' \lll S$, where exactly one of the following cases holds:

1. $s' = \langle v_j v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$.

2. $s' = \langle v_{i_1} \ldots v_{i_r} v_j v_{i_{r+1}} \ldots v_{i_k} \rangle$ for some r $(1 \leq i \leq k-1)$.

3. $s' = \langle v_{i_1} v_{i_2} \ldots v_{i_k} v_j \rangle$.

By the second half of the feasibility argument (using s' from the relevant case above), $I \cup \{v_j\}$ is an independent set, which contradicts the maximality of I. Thus s is maximal.

Conversely, suppose that $s = \langle v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$ is a maximal common subsequence of S. As $s \lll s_0$, then $i_1 < i_2 < \ldots < i_k$. By the second half of the feasibility argument (using s), $I = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is independent. Suppose that I is not maximal; then there is some j $(1 \leq j \leq n)$ such that $I \cup \{v_j\}$ is independent. As $j \neq i_p$ for any p $(1 \leq p \leq k)$, exactly one of the following three cases holds:

1. $j < i_1$. Let $s' = \langle v_j v_{i_1} v_{i_2} \ldots v_{i_k} \rangle$.

2. $i_r < j < i_{r+1}$ for some r $(1 \leq r \leq k-1)$. Let $s' = \langle v_{i_1} \ldots v_{i_r} v_j v_{i_{r+1}} \ldots v_{i_k} \rangle$.

3. $j > i_k$. Let $s' = \langle v_{i_1} v_{i_2} \ldots v_{i_k} v_j \rangle$.

Then $s \ll s'$. By the first half of the feasibility argument (using $I \cup \{v_j\}$), $s' \lll S$, which contradicts the maximality of s. Thus I is maximal.

Thus $\langle f, g \rangle$ satisfies Properties 1-4 of Definition 8.2.1, and is therefore an MM-reduction.
■

8.3.5 Independent set to vertex cover

In this section, we consider a reduction from MAX IND SET to MIN VERTEX COVER. The components of the latter problem were defined in Section 4.2.6.

Theorem 8.3.6 $\langle$MAX IND SET$, \subset\rangle \alpha_{MM} \langle$MIN VERTEX COVER$, \subset\rangle$.

Proof: Let $G = (V, E)$ be an instance of MAX IND SET, where $n = |V|$. Define the following reduction:

$$\begin{aligned} &f : G \mapsto G \\ &g(G, \cdot) : V' \mapsto V \backslash V'. \end{aligned}$$

The result follows from Proposition 4.2.8 and Lemma 4.2.9. ∎

8.3.6 Vertex cover to set cover

In this section, we consider a reduction from MIN VERTEX COVER to MINIMUM SET COVER (whose decision version is problem SP5 of [92], referred to there as MINIMUM COVER). The latter problem is defined as follows.

Source problem: MINIMUM SET COVER$= \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT}\rangle$, where

- $\mathcal{I} = \{\langle S, C\rangle : C \subseteq \mathbb{P}(S)\}$
- $\mathcal{U}(\langle S, C\rangle) = \mathbb{P}(C)$
- $\pi(\langle S, C\rangle, C') \Leftrightarrow C'$ is a set cover for S, i.e.,$\forall\, s \in S \bullet \exists\, C_i \in C' \bullet s \in C_i$
- $m(\langle S, C\rangle, C') = |C'|$
- OPT $=$ min.

Maximinimal problem name: MAXIMUM MINIMAL SET COVER.

Theorem 8.3.7 $\langle$MIN VERTEX COVER$, \subset\rangle \alpha_{MM} \langle$MINIMUM SET COVER$, \subset\rangle$.

Proof: Let $G = (V, E)$ be an instance of MIN VERTEX COVER, where $V = \{v_1, v_2, \ldots, v_n\}$. Define the following reduction (due to Karp [140]):

$$\begin{aligned} f : G \mapsto \langle S, C\rangle, \text{ where } \quad & S = E \\ & C = \{C_1, C_2, \ldots, C_n\} \\ & C_i = \{\{v_i, v_j\} : \{v_i, v_j\} \in E\} \ (1 \le i \le n) \\ g(\langle S, C\rangle, \cdot) : \{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\} &\mapsto \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}. \end{aligned}$$

To see that this reduction preserves feasibility and measure, suppose that $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is a vertex cover in G. Let $e \in E$ be given, where $e = \{v_p, v_q\}$ for some p, q $(1 \le p, q \le n)$. Then without loss of generality, $p = i_r$ for some r $(1 \le r \le k)$. Thus $e \in C_{i_r}$, so that $\{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\}$ is a set cover for E.

Conversely, suppose that $\{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\}$ is a set cover for E. Let $e \in E$ be given, where $e = \{v_p, v_q\}$ for some p, q $(1 \le p, q \le n)$. Then $e \in C_{i_r}$ for some r $(1 \le r \le k)$. Without loss of generality, $p = i_r$, so that $\{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ is a vertex cover for G.

By the definitions of f and g, it is easy to see that the reduction also preserves the minimality of feasible solutions. Thus $\langle f, g\rangle$ satisfies Properties 1-4 of Definition 8.2.1, and is therefore an MM-reduction. ■

Corollary 8.3.8 MINIMUM SET COVER DECISION *and* MAXIMUM MINIMAL SET COVER DECISION *are NP-complete, even if* $|C_i| = 3$, *for all* $C_i \in C$.

Proof: From Section 4.2.6, we know that both α_0 and α_0^+ are NP-complete for cubic graphs. Thus, by setting this restriction on the instance of MIN VERTEX COVER in Theorem 8.3.7 above, we obtain the stated result. ■

8.3.7 Vertex cover to hitting set

In this section, we consider a reduction from MIN VERTEX COVER to MINIMUM HITTING SET (whose decision version is problem SP8 of [92]). The latter problem is defined as follows.

Source problem: MINIMUM HITTING SET= $\langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt}\rangle$, where

- $\mathcal{I} = \{\langle S, C\rangle : C \subseteq \mathbb{P}(S)\}$
- $\mathcal{U}(\langle S, C\rangle) = \mathbb{P}(S)$
- $\pi(\langle S, C\rangle, S') \Leftrightarrow S'$ is a hitting set for C, i.e.,$\forall\, C_i \in C \bullet S' \cap C_i \neq \varnothing$
- $m(\langle S, C\rangle, S') = |S'|$
- OPT $=$ min.

Maximinimal problem name: MAXIMUM MINIMAL HITTING SET.

Theorem 8.3.9 $\langle$MIN VERTEX COVER, $\subset\rangle \alpha_{MM} \langle$MINIMUM HITTING SET, $\subset\rangle$.

Proof: Let $G = (V, E)$ be an instance of MIN VERTEX COVER, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. Define the following reduction (due to Garey and Johnson [92, p.64]):

$$\begin{array}{ll} f : G \mapsto \langle S, C\rangle, \text{ where} & S = V \\ & C = \{C_1, C_2, \ldots, C_m\} \\ & C_i = \{v_p, v_q : e_i = \{v_p, v_q\}\}\ (1 \leq i \leq m) \\ g(\langle S, C\rangle, \cdot) : V' \mapsto V'. & \end{array}$$

This reduction preserves feasibility, for it is clear that, given a set of vertices $V' \subseteq V$, V' is a vertex cover for E if and only if V' is a hitting set for C. By the definitions of f and g, it is easy to see that the reduction also preserves the minimality of feasible solutions, and in addition, the measure of feasible solutions. Thus $\langle f, g\rangle$ satisfies Properties 1-4 of Definition 8.2.1, and is therefore an MM-reduction. ■

Corollary 8.3.10 MINIMUM HITTING SET DECISION *and* MAXIMUM MINIMAL HITTING SET DECISION *are NP-complete, even if* $|C_i| = 2$, *for all* $C_i \in C$.

8.4 Concluding remarks relating to MM-reductions

The range of MM-reductions presented in Section 8.3 is obviously far from exhaustive. However, by building up a tree of MM-reductions (shown in Figure 8.1) in the spirit of the one developed by Karp [140] for polynomial reductions, we have at least formulated a variety of MM-reductions pertaining to optimisation problems from several Garey and Johnson subject categories, and involving several partial orders. The MM-reduction from $\langle$MAXIMUM ONE-IN-THREE 3SAT,$\prec_t\rangle$ to $\langle$MAX CLIQUE,$\subset\rangle$ is a new reduction, whereas the other MM-reductions have been constructed from polynomial reductions appearing in the literature. A perhaps desirable addition to Figure 8.1 would be an MM-reduction to the pair $\langle$CHROMATIC NUMBER,$\prec\rangle$, where $\prec^G$ is a suitable partial order defined on the proper colourings of G, such as partition merge or partition redistribution, given an instance G of CHROMATIC NUMBER.

For many of the MM-reductions defined in Section 8.3, showing that the reduction satisfies Properties 3 and 4 of Definition 8.2.1 is immediate, since g is essentially an identity function on feasible solutions. This is the case for the MM-reductions given by Theorems 8.3.2, 8.3.3, 8.3.7 and 8.3.9. In the case of the MM-reduction of Theorem 8.3.6, showing that Properties 3 and 4 of Definition 8.2.1 are satisfied is also a simple matter, using the notion of *complement-related* families of sets [196], which may be defined as follows.

Definition 8.4.1 ([196]) *Let X be some set, and let $\mathcal{F}_1(X), \mathcal{F}_2(X)$ be two families of subsets of X. Then $\mathcal{F}_1(X)$ and $\mathcal{F}_2(X)$ are* complement-related *if, whenever $S \subseteq X$, we have that $S \in \mathcal{F}_1(X)$ if and only if $X\backslash S \in \mathcal{F}_2(X)$.*

In the context of the MM-reduction of Theorem 8.3.6, X is the set of vertices V of a graph $G = (V, E)$, $\mathcal{F}_1(V)$ is the set of all vertex covers of G, and $\mathcal{F}_2(V)$ is the set of all independent sets of G. Thus in this example, $\mathcal{F}_1(V)$ and $\mathcal{F}_2(V)$ are complement-related, which demonstrates that Property 3 of Definition 8.2.1 holds for the MM-reduction of Theorem 8.3.6. The following result indicates that complement-related sets have further implications for our study of MM-reductions.

Proposition 8.4.2 ([196]) *Let X be some set, and let $\mathcal{F}_1(X), \mathcal{F}_2(X)$ be two families of subsets of X. For $i = 1, 2$, let $\mathcal{F}_i^+(X)$ be the set of maximal (with respect to the partial order of set inclusion) elements of $\mathcal{F}_i(X)$, and let $\mathcal{F}_i^-(X)$ be the set of minimal (with respect to the partial order of set inclusion) elements of $\mathcal{F}_i(X)$. Suppose that $\mathcal{F}_1(X)$ and $\mathcal{F}_2(X)$ are complement-related. Then so are $\mathcal{F}_1^+(X)$ and $\mathcal{F}_2^-(X)$, and so are $\mathcal{F}_1^-(X)$ and $\mathcal{F}_2^+(X)$.*

Thus, in the context of the same example, $\mathcal{F}_1^-(V)$ and $\mathcal{F}_2^+(V)$ are complement-related, which demonstrates that Property 4 of Definition 8.2.1 holds for the MM-reduction of Theorem 8.3.6.

In general, suppose that Π_1, Π_2 are two optimisation problems with the same set of instances $\mathcal{I}$, and for a given instance $x \in \mathcal{I}$, suppose that the feasible solutions of Π_1 and Π_2, namely $\mathcal{F}_1(x)$ and $\mathcal{F}_2(x)$ respectively, are complement-related. Let f be the identity function on $\mathcal{I}$, and let g map a feasible solution of $\mathcal{F}_2(x)$ to its complement in x, for any $x \in \mathcal{I}$. Then $\langle f, g \rangle$ constitutes an MM-reduction from $\langle \Pi_1, \subset \rangle$ to $\langle \Pi_2, \subset \rangle$, by Proposition 8.4.2. It ought to be possible to find many other polynomial reductions that are MM-reductions by considering pairs of optimisation problems whose feasible solutions are complement-related families of sets.

Chapter 9

Further issues relating to minimaximal and maximinimal optimisation problems

9.1 Introduction

In the foregoing chapters, our study of minimaximal and maximinimal optimisation problems has focused mainly on the algorithmic complexity of various source optimisation problems, together with their minimaximal or maximinimal counterparts. In this chapter we investigate further general issues that arise from the study of minimaximal and maximinimal optimisation problems.

In Sections 9.2-9.4, we examine the problem of testing a feasible solution of a given optimisation problem Π for local optimality, and finding a locally optimal feasible solution of Π, with respect to a partial order defined on the feasible solutions for a given instance of Π. In particular, we focus on the testing and finding problems where Π is BIN PACKING and CHROMATIC NUMBER, together with certain partial orders.

Recall that a minimaximal or maximinimal optimisation problem may be obtained from a source optimisation problem Π by defining a partial order on $\mathcal{F}(x)$, the set of feasible solutions for a given instance x of Π. In Section 9.5, we consider the effect of defining our partial orders on $\mathcal{U}(x)$, the universal set of possible solutions, rather than on $\mathcal{F}(x)$, for a given instance x of Π. We show that there is a class of partial orders and optimisation problems for which we may define the partial order $\prec^x$ on $\mathcal{U}(x)$, with the result that the $\prec^x$-optimal solutions are the same as those obtained by defining $\prec^x$ on $\mathcal{F}(x)$. We also discuss why the framework of Definition 2.3.5 nevertheless demands that $\prec^x$ should be defined on $\mathcal{F}(x)$, rather than on $\mathcal{U}(x)$, in general.

Finally, in Section 9.6, we present some conclusions and open problems relating to minimaximal and maximinimal optimisation problems in general.

9.2 Testing feasible solutions for local optimality, and finding locally optimal feasible solutions

Any polynomial-time algorithm that solves a minimaximal (respectively maximinimal) optimisation problem must involve *finding*, in polynomial time, a solution that is maximal (respectively minimal) with respect to the partial order concerned. Similarly, any proof that a minimaximal (respectively maximinimal) problem is in NP must involve *verifying*, in polynomial time, that a given feasible solution is maximal (respectively minimal). Thus the issues of testing a feasible solution for maximality or minimality, and finding maximal or minimal feasible solutions, are of paramount importance in our study of the complexity of minimaximal and maximinimal optimisation problems.

For most of the optimisation problems Π and partial orders $\prec^x$ that we have studied in this thesis, the problem of testing a feasible solution of Π for $\prec^x$-optimality is a simple procedure, based on a polynomial-time checkable criterion. For example, in the case of MINIMUM MAXIMAL CLIQUE (defined in Section 4.3.2), given a graph G and a clique S in G, we test for the existence of a vertex $v \in V \backslash S$ adjacent to every w in S. If such a v exists, then $S \cup \{v\}$ is a clique, and if no such v exists, then S is $\subset^G$-maximal.

Suppose that Π and $\prec^x$ are as above, and there is a polynomial-time procedure for testing a feasible solution of Π for $\prec^x$-optimality. (In this chapter, we assume that such a testing algorithm returns 'yes' if the feasible solution is $\prec^x$-optimal, or if not, provides a feasible $\prec^x$-predecessor or $\prec^x$-successor as appropriate.) Suppose further that the range of values that the measure function of Π can take is bounded by a polynomial in $|x|$, for a given instance x of Π. Then it follows by POMM that a series of iterations of the implicit algorithm for testing a feasible solution for $\prec^x$-optimality constitutes a polynomial-time procedure for finding a $\prec^x$-optimal solution. [Note that, in the case of weighted optimisation problems, the 'measure' function to consider for the purposes of this paragraph is the number of elements that a feasible solution contains, rather than the total weight/total measure or total value of a feasible solution. For example, in the case of LONGEST PATH and UNCONSTRAINED LONGEST PATH (defined in Sections 7.2.1 and 7.2.2 respectively), the 'measure' function to be considered here is the number of edges in the path, rather than the total length of a given path in the graph that satisfies the constraints. Similarly, in the case of MAXIMUM KNAPSACK (defined in Section 7.6.1), the 'measure' function to be considered here is the number of elements in the knapsack packing, rather than the sum of the values of the elements in a knapsack packing[1].]

It is not always the case that the problem of testing a feasible solution of Π for $\prec^x$-optimality is polynomial-time solvable. Recall from Section 2.4 the definitions of the partial

[1] In the case of TRAVELLING SALESMAN (defined in Section 1.5.2), we may define (using Theorem 2.5.1) a partial order $\prec^x$ on the feasible solutions of this problem, for a given instance x, based on the *2-opt* neighbourhood (defined in Section 1.5.2). Testing a travelling salesman tour for $\prec^x$-optimality may be achieved in polynomial time by considering pairs of edges in the tour. However, no measure function for TRAVELLING SALESMAN is known that satisfies POMM with respect to $\prec^x$, and whose range of values is bounded by a polynomial in $|x|$. Not surprisingly, TRAVELLING SALESMAN under the *2-opt* neighbourhood is conjectured to be PLS-complete (discussed in Section 1.5.4).

orders of partition merge and partition redistribution, denoted $\prec_a^x$ and $\prec_b^x$ respectively. In Section 9.3, we show that, for a given instance x of MINIMUM BIN PACKING (defined in Section 7.4.1) and a given bin packing P of the objects of x, the problem of testing P for $\prec_b^x$-minimality is NP-hard. However, we show that the problem of finding a $\prec_b^x$-minimal bin packing is polynomial-time solvable. We also show that the problems of testing a bin packing for $\prec_a^x$-minimality and of finding a $\prec_a^x$-minimal bin packing are both polynomial-time solvable.

In Section 9.4, we consider the partial orders $\prec_{a,k}^G$ ($k \geq 2$) and $\prec_{b,k}^G$ ($k \geq 1$) (these partial orders are defined in Section 2.4), defined on the set of all proper colourings of a given graph G. We investigate the problems of testing proper graph colourings for minimality, and of finding minimal proper graph colourings, with respect to these partial orders. For completeness, we also consider the complexity of the associated maximinimal optimisation problems in each case. Our algorithmic results for testing and finding show where the thresholds between polynomial-time solvability and NP-hardness lie, within the hierarchy of problems corresponding to the two partial order families. In particular, we show that the partial order $\prec_{z,k}^G$ (where $k \geq 4$ and z is 'a' or 'b') satisfies the property that both the problems of testing a proper graph colouring for $\prec_{z,k}^G$-minimality and of finding a $\prec_{z,k}^G$-minimal proper graph colouring are NP-hard, for a given graph G.

Note that in Sections 9.3 and 9.4, an NP-hardness result for the problem of testing a given feasible solution s of Π for $\prec^x$-optimality will be demonstrated by proving NP-completeness for the complement of the decision problem 'is s $\prec^x$-optimal?'.

9.3 Testing a bin packing for minimality, and finding minimal bin packings

Consider $\prec_a^x$, the partial order of partition merge, and $\prec_b^x$, the partial order of partition redistribution, defined on the source MINIMUM BIN PACKING problem (whose components were defined in Section 7.4.1), for a given instance x. Testing a bin packing P for $\prec_a^x$-minimality can be achieved in polynomial time. For, we need only consider each pair of bins i, j in P, and check that the two bins cannot be merged without overfilling the bin capacity. If this is the case, then P is $\prec_a^x$-minimal; otherwise we may merge two bins i, j, in order to construct a bin packing P' for which $P' \prec_a^x P$.

On the other hand, the problem of testing a given bin packing for $\prec_b^x$-minimality is NP-hard, as we show in this section. We also show that the problem of finding a $\prec_b^x$-minimal bin packing is polynomial-time solvable – this is perhaps an unexpected result, given the NP-hardness result for the testing problem. The problem of finding a $\prec_a^x$-minimal bin packing is also polynomial-time solvable. This follows either from the algorithm for finding a $\prec_b^x$-minimal bin packing, together with the fact that a $\prec_b^x$-minimal bin packing is $\prec_a^x$-minimal, or follows by an iteration of the implicit algorithm for testing a bin packing for $\prec_a^x$-minimality, as described in the previous paragraph.

We now define two decision problems that will be used in order to prove the NP-hardness result for the problem of testing a bin packing for $\prec_b^x$-minimality.

Name: $\prec_b$-MINIMAL BIN PACKING TEST.
Instance x: Finite set U, each $u \in U$ with associated size $s(u) \in \mathbb{Z}^+$, integer $B \in \mathbb{Z}^+$ and bin packing P of U into k bins $U_1, U_2, \ldots, U_k$, for some $k \in \mathbb{Z}^+$, such that $\forall\, 1 \leq i \leq k \bullet \sum_{u \in U_i} s(u) \leq B$.
Question: Whether P is $\prec_b^x$-minimal.

Name: RESTRICTED MINIMUM BIN PACKING DECISION.
Instance: Finite set U, each $u \in U$ with associated size $s(u) \in \mathbb{Z}^+$ and integers $B, K \in \mathbb{Z}^+$ such that $\sum_{u \in U} s(u) > 2B$.
Question: Whether there is a partition $U_1, U_2, \ldots, U_k$ of U for $k \leq K$ such that $\forall\, 1 \leq i \leq k \bullet \sum_{u \in U_i} s(u) \leq B$.

Let $\prec_b$-MINIMAL BIN PACKING TEST^C denote the complement of the problem $\prec_b$-MINIMAL BIN PACKING TEST. We firstly resolve the complexity of the RESTRICTED MINIMUM BIN PACKING DECISION problem.

Lemma 9.3.1 RESTRICTED MINIMUM BIN PACKING DECISION *is NP-complete.*

Proof: Clearly, RESTRICTED MINIMUM BIN PACKING DECISION is in NP. To show NP-hardness, we give a transformation from MINIMUM BIN PACKING DECISION (defined in Section 7.4.1). Suppose we have an instance of MINIMUM BIN PACKING DECISION: objects $a_1, a_2, \ldots, a_n$, each with size $s(a_i) \in \mathbb{Z}^+$, bin capacity $B \in \mathbb{Z}^+$ and target number of bins $K \in \mathbb{Z}^+$. Construct the following instance of RESTRICTED MINIMUM BIN PACKING DECISION: objects $a'_1, a'_2, \ldots, a'_{n+2}$, bin capacity $B' \in \mathbb{Z}^+$ and target number of bins $K' \in \mathbb{Z}^+$, where

$$\begin{array}{lcll} s(a'_i) & = & s(a_i), & 1 \leq i \leq n, \\ s(a'_i) & = & B, & i = n+1, n+2, \\ B' & = & B & \text{and} \\ K' & = & K+2. & \end{array}$$

Then $\sum_{i=1}^{n+2} s(a'_i) > 2B'$. Clearly, for any $k \leq K$, objects $a_1, a_2, \ldots, a_n$ have a packing into bins $U_1, U_2, \ldots, U_k$ if and only if objects $a'_1, a'_2, \ldots, a'_{n+2}$ have a packing into bins $U_1, U_2, \ldots, U_{k+2}$ (where $k+2 \leq K'$). ∎

We now prove that testing a bin packing for $\prec_b^x$-minimality is NP-hard, for a given bin packing instance x.

Theorem 9.3.2 $\prec_b$-MINIMAL BIN PACKING TEST^C *is NP-complete.*

Proof: $\prec_b$-MINIMAL BIN PACKING TEST^C is in NP, for, given a bin packing and a re-distribution of one bin in this packing amongst the remaining bins, we may verify, in polynomial time, that the resulting packing is legal. To show NP-hardness we give a transformation from RESTRICTED MINIMUM BIN PACKING DECISION. Suppose we have an

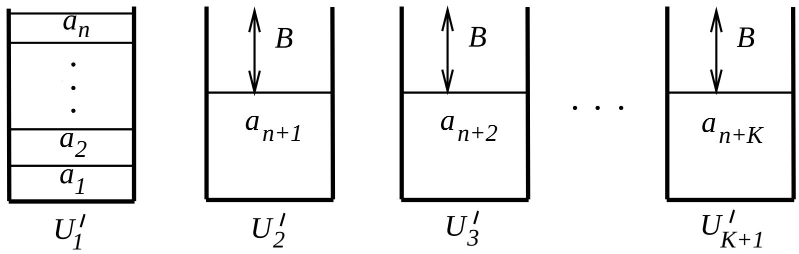

Figure 9.1: Bin packing arrangement in the constructed instance of $\prec_b$-MINIMAL BIN PACKING TEST^C.

instance of RESTRICTED MINIMUM BIN PACKING DECISION: bin capacity $B \in \mathbb{Z}^+$, target number of bins $K \in \mathbb{Z}^+$ and objects $a_1, a_2, \ldots, a_n$, each with size $s(a_i) \in \mathbb{Z}^+$, such that $\sum_{i=1}^{n} s(a_i) > 2B$. Contruct an instance of $\prec_b$-MINIMAL BIN PACKING TEST^C as follows: objects $a'_1, a'_2, \ldots, a'_{n+K}$, bin capacity $B' \in \mathbb{Z}^+$, where

$$\begin{aligned} B' &= \sum_{i=1}^{n} s(a_i), \\ s(a'_i) &= s(a_i), \qquad 1 \le i \le n, \\ s(a'_i) &= B' - B, \qquad n+1 \le i \le n+K. \end{aligned}$$

We also construct a packing of $a'_1, a'_2, \ldots, a'_{n+K}$ into $K+1$ bins $U'_1, U'_2, \ldots, U'_{K+1}$ as follows:

- Into bin U'_1 insert $a'_1, a'_2, \ldots, a'_n$.
- Into bin U'_i insert a'_{n+i-1}, for $2 \le i \le K+1$.

This bin packing is illustrated in Figure 9.1. Clearly, bin U'_1 is full, whilst bin U'_i has space B left, for $2 \le i \le K+1$. The claim is that $a_1, a_2, \ldots, a_n$ has a packing into k bins $U_1, U_2, \ldots, U_k$ of capacity B, for some $k \le K$, if and only if one of the bins U'_i can be redistributed amongst the other K bins $U'_1, \ldots, U'_{i-1}, U'_{i+1}, \ldots, U'_{K+1}$, for some i $(1 \le i \le K+1)$.

For, a packing of $a_1, a_2, \ldots, a_n$ into $U_1, U_2, \ldots, U_k$, for some $k \le K$, corresponds to a redistribution of U'_1 amongst k of the remaining K bins $U'_2, U'_3, \ldots, U'_{K+1}$.

Conversely, if one of the bins U'_i can be redistributed then we must have $i = 1$. For $B' - B > B$ by assumption, so that no U'_i, for $i > 1$, may be redistributed amongst the other K bins. The redistribution of U'_1 among $U'_2, U'_3, \ldots, U'_{K+1}$ corresponds to a packing of $a_1, a_2, \ldots, a_n$ into at most K bins of capacity B. ∎

Perhaps surprisingly, it turns out that, for an arbitrary bin packing instance x, the problem of finding a $\prec_b^x$-minimal bin packing is polynomial-time solvable. In order to demonstrate this, we define the following search problem:

Name: $\prec_b$-MINIMAL BIN PACKING SEARCH.
Instance x: Finite set U, each $u \in U$ with associated size $s(u) \in \mathbb{Z}^+$ and integer $B \in \mathbb{Z}^+$.
Output: A $\prec_b^x$-minimal bin packing.

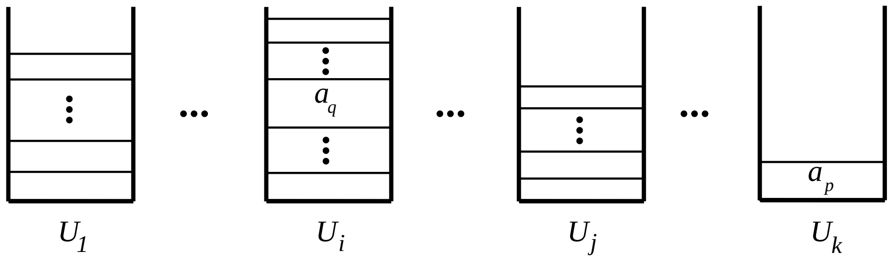

Figure 9.2: Bin packing arrangement produced by AFD after iteration p.

Theorem 9.3.3 $\prec_b$-MINIMAL BIN PACKING SEARCH *is polynomial-time solvable.*

Proof: Let $a_1, a_2, \ldots, a_n$, each with size $s(a_i) \in \mathbb{Z}^+$ $(1 \leq i \leq n)$, and $B \in \mathbb{Z}^+$ (bin capacity) be an instance x of $\prec_b$-MINIMAL BIN PACKING SEARCH. Suppose, without loss of generality, that the objects have indices such that $s(a_1) \geq s(a_2) \geq \ldots \geq s(a_n)$. Consider the following algorithm, which we call *Any Fit Decreasing (AFD)*. During iteration i $(1 \leq i \leq n)$, suppose that objects $a_1, a_2, \ldots, a_{i-1}$ have been packed into bins $U_1, U_2, \ldots, U_r$ (for some $r \geq 0$). The algorithm places a_i into *any* bin U_i $(1 \leq i \leq r)$ into which the object will fit (without exceeding the capacity B), or, if no such bin is available, places a_i into a new bin U_{r+1}. We claim that AFD produces a $\prec_b^x$-minimal bin packing.

For, suppose not. Then there is some element a_p $(1 \leq p \leq n)$ such that, directly after a_p has been packed by AFD, the resulting bin packing arrangement is non-$\prec_b^x$-minimal. Choose p to be the smallest such integer. Let U_i be a bin that can be redistributed among the remaining bins after a_p is packed.

At iteration p of AFD, suppose that object a_p is placed into some bin U_k. We claim that this bin is new. For, suppose not. Then U_k was nonempty before iteration p, so that

$$\begin{array}{rl} & U_i \text{ can be redistributed among remaining bins after iteration } p \text{ of AFD} \\ \Rightarrow & U_i \text{ can be redistributed among remaining bins before iteration } p \text{ of AFD.} \end{array}$$

Since this implication holds even if $i = k$, then we contradict the choice of p.

Now, $k \neq i$, for if U_k (containing only a_p) can be redistributed among the remaining bins, then the new bin U_k would not have been used by AFD. Hence $1 \leq i < k$. Similarly, not all of the contents of bin U_i are placed in bin U_k in the redistribution of U_i, or else a_p would have been placed in bin U_i by AFD. Hence there is some q $(1 \leq q < p)$ such that a_q is placed in a bin U_j $(1 \leq j \neq i < k)$ in the redistribution of U_i. The bin packing construction at this stage is shown in Figure 9.2.

But $q < p$ implies that $s(a_q) \geq s(a_p)$. Hence

$$\begin{array}{rl} & a_q \text{ can be placed in } U_j \text{ in the redistribution of } U_i \\ \Rightarrow & a_p \text{ can be placed in } U_j \text{ in the original packing.} \end{array}$$

Thus, AFD would have placed a_p into bin U_j and not bin U_k. This contradiction shows that AFD does indeed give a $\prec_b^x$-minimal packing. ■

As an addendum to Theorem 9.3.3, we consider the *First Fit Decreasing (FFD)* and *Best Fit Decreasing (BFD)* algorithms [92, p.126].

As with AFD, both algorithms initially reindex the n given objects into nonincreasing order of size, and consider each object in turn, in indicial order (lowest first). During iteration i $(1 \leq i \leq n)$, suppose that objects $a_1, a_2, \ldots, a_{i-1}$ have been packed into bins $U_1, U_2, \ldots, U_r$ (for some $r \geq 0$). FFD places a_i into the lowest indexed bin U_i $(1 \leq i \leq r)$ into which the object will fit (without exceeding the capacity B), or, if no such bin is available, places a_i into a new bin U_{r+1}. BFD places a_i into the bin U_i $(1 \leq i \leq r)$ which has current contents closest to, but not exceeding, $B - s(a_i)$ (choosing the lowest indexed bin in the case of ties), or, if no such bin is available, places a_i into a new bin U_{r+1}.

Since FFD and BFD are special cases of AFD, they may be used as polynomial-time algorithms to solve $\prec_b$-MINIMAL BIN PACKING SEARCH, by Theorem 9.3.3.

For a simple example of where none of AFD, BFD or FFD produces a *maximum* $\prec_b^x$-minimal bin packing, consider the packing of four elements, a_1, a_2, a_3, a_4, of size 1, and four elements, b_1, b_2, b_3, b_4, of size 2, into bins of capacity 4. Each algorithm will pack b_1, b_2 into bin 1, b_3, b_4 into bin 2, and a_1, a_2, a_3, a_4 into bin 3. The maximum $\prec_b^x$-minimal packing places a_i, b_i into bin i, for $1 \leq i \leq 4$.

The result of Theorem 9.3.3 might lead one to consider whether there exists a source optimisation problem Π, together with a partial order $\prec^x$ defined on the feasible solutions for a given instance x of Π, such that the problem of finding a $\prec^x$-optimal solution is NP-hard. In fact, the answer to this question is in the affirmative, and an example may be found by considering the source optimisation problem CHROMATIC NUMBER, and the partial order $\prec_{z,k}^G$ (where $k \geq 4$ and z is 'a' or 'b'), for a given graph G, as is demonstrated in the next section.

9.4 Testing a proper colouring for minimality, and finding minimal proper colourings

9.4.1 Introduction

Consider $\prec_{a,k}^G$ (for $k \geq 2$), the partial order of partition $(k-1, k)$-merge, and $\prec_{b,k}^G$ (for $k \geq 1$), the partial order of partition k-redistribution (given by Definitions 2.4.13 and 2.4.16 respectively), defined on the feasible solutions of CHROMATIC NUMBER (defined in Section 3.1) for a given graph G. In this section, we study three algorithmic problems corresponding to proper graph colourings that are minimal with respect to a given partial order from one of the two families $\prec_{a,k}$ $(k \geq 2)$ and $\prec_{b,k}$ $(k \geq 1)$. These problems relate to the complexity of *testing* a proper colouring for minimality, the complexity of *finding* a minimal proper colouring, and the complexity of *maximising* the number of colours over all minimal proper colourings.

We show that, for each of the two partial order families, there is a threshold lying across the hierarchy of testing problems corresponding to each member of the relevant partial order family, below which the problems are polynomial-time solvable, and above which the problems are NP-hard. There is a similar threshold in the case of the finding problems. Consideration of proper colourings that are minimal with respect to both of the finest partial orders (from each of the two partial order families) such that both of the associated finding problems are polynomial-time solvable, may yield a worthwhile local search strategy for approximating the chromatic number in certain graph classes. In addition to studying the testing and finding problems, we prove complexity results for maximisation problems relating to proper graph colourings that are minimal with respect to partial orders from each of the two partial order families.

We organise the forthcoming sections as follows. In Section 9.4.2, we prove some general results concerning $\prec^G_{a,k}$-minimal ($k \geq 2$) and $\prec^G_{b,k}$-minimal ($k \geq 1$) proper graph colourings. Sections 9.4.2-9.4.9 concentrate on individual members of each family, and the three corresponding algorithmic questions mentioned above. Finally, in Section 9.4.10, we summarise the complexity results appearing in this section, and present some concluding remarks.

9.4.2 (a,k)-minimal $(k \geq 2)$ and (b,k)-minimal $(k \geq 1)$ proper graph colourings

Let $\mathcal{F}(G)$ denote the set of all proper colourings of a given graph G. Recall from Definition 2.4.13 that, intuitively, for two proper colourings c_1 and c_2 in $\mathcal{F}(G)$, $c_1 \sqsubset^G_{a,k} c_2$ if c_1 can be obtained from c_2 by recolouring the vertices of r colours of c_2 $(2 \leq r \leq k)$ by $r-1$ new colours, whilst every other vertex retains its original colour.

Similarly, recall from Definition 2.4.16 that, intuitively, for two proper colourings c_1 and c_2 in $\mathcal{F}(G)$, $c_1 \sqsubset^G_{b,k} c_2$ if c_1 can be obtained from c_2 by distributing the vertices of r colours in c_2 $(1 \leq r \leq k)$ amongst the remaining colours in c_2 plus $r-1$ new colours, whilst every other vertex retains its original colour.

A proper colouring of G that is $\prec^G_{a,k}$-minimal will be called (a,k)-*minimal* (where $k \geq 2$), and a proper colouring of G that is $\prec^G_{b,k}$-minimal will be called (b,k)-*minimal* (where $k \geq 1$). Every graph has at least one proper colouring that is minimal with respect to the partial orders defined above, as we now show.

Proposition 9.4.1 *Let G be a graph, and let $k \geq 2$. Then G has an (a,k)-minimal proper colouring, and G has a $(b,k-1)$-minimal proper colouring.*

Proof: G has a proper colouring using $\chi(G)$ colours. This colouring must be both (a,k)-minimal and $(b,k-1)$-minimal. ■

The following proposition and its corollaries establish limits on the orders of proper colourings that are (a,k)-minimal and (b,k)-minimal $(k \geq 2)$.

Proposition 9.4.2 *Let G be a graph, let $k \geq 1$, and suppose that G is k-colourable. Then G does not have an (a,l)-minimal proper colouring of more than k colours, for any $l \geq k+1$.*

Proof: Suppose G *does* have an (a,l)-minimal proper colouring of more than k colours. Pick any $k+1$ colours in such an (a,l)-minimal proper colouring of G, and consider the subgraph G' of G induced by the vertices belonging to these $k+1$ colours. Then G' is k-colourable, as G is, contradicting the (a, l)-minimality of the colouring of G. ■

Corollary 9.4.3 *Let G be a graph, let $k \geq 1$, and suppose that G is k-colourable. Then G does not have a (b, l)-minimal proper colouring of more than k colours, for any $l \geq k+1$.*

Proof: The result follows from Corollary 2.4.20 and Proposition 9.4.2. ■

Corollary 9.4.4 *Let G be a graph. Then G does not have an (a, l)-minimal proper colouring of more than χ colours, for any $l \geq \chi + 1$, where $\chi = \chi(G)$.*

Corollary 9.4.5 *Let G be a graph. Then G does not have a (b, l)-minimal proper colouring of more than χ colours, for any $l \geq \chi + 1$, where $\chi = \chi(G)$.*

For each partial order belonging to one of the families of Definitions 2.4.13 and 2.4.16, we study three associated algorithmic problems. The problems are concerned with testing a proper colouring for minimality, finding a minimal proper colouring, and maximising the number of colours over all minimal proper colourings, with respect to the partial order concerned. We now define these problems. In the following, assume that 'z' is 'a' or 'b', and that $k \geq 1$ is given ($k \geq 2$ if 'z' is 'a').

Name: (z, k)-MINIMAL GRAPH COLOURING TEST.
Instance: Graph $G = (V, E)$ and a proper colouring c of G.
Question: Is c (z, k)-minimal?

Name: (z, k)-MINIMAL GRAPH COLOURING SEARCH.
Instance: Graph $G = (V, E)$.
Output: (z, k)-minimal proper graph colouring of G[2].

Name: MAXIMUM (z, k)-MINIMAL CHROMATIC NUMBER.
Instance: Graph $G = (V, E)$.
Output: Maximum (z, k)-minimal proper colouring of G.

Clearly, the problems MAXIMUM $(a, 2)$-MINIMAL CHROMATIC NUMBER and MAXIMUM $(b, 1)$-MINIMAL CHROMATIC NUMBER correspond to ACHROMATIC NUMBER (defined in Section 3.2) and B-CHROMATIC NUMBER (defined in Section 3.3), respectively. In the following sections, we consider the three problems defined above, corresponding to the members $\prec^G_{a,k}$ ($k \geq 2$) and $\prec^G_{b,k}$ ($k \geq 1$) of the two partial order families.

9.4.3 $(a, 2)$-minimal graph colourings

Recall from Section 3.2 that a proper colouring c of a graph G is $(a, 2)$-minimal if and only c is achromatic, i.e., c satisfies Property 3.1 on Page 34. Testing a proper colouring c of G for achromaticity is clearly possible in polynomial time. One need only check, for each distinct pair of colours i, j of c, that there is an edge of G that has colours i and j at

[2] Guaranteed to exist by Proposition 9.4.1.

its endpoints. If this is the case, then c is $(a, 2)$-minimal; otherwise every vertex of colour i may be recoloured by colour j in order to construct a proper colouring c' such that $c' \prec^G_{a,2} c$. It follows that an iteration of this implicit algorithm for testing constitutes a polynomial-time procedure for finding an $(a, 2)$-minimal proper colouring. The complexity of ACHROMATIC NUMBER for various graph classes is discussed in Section 3.2.

9.4.4 $(a, 3)$-minimal graph colourings

Testing a proper graph colouring for $(a, 3)$-minimality may be accomplished in polynomial time. For, we consider triples of distinct colours i, j, k and check that the subgraph induced by these three colours contains an odd cycle (i.e., is non-bipartite). If this is the case, then c is $(a, 3)$-minimal; otherwise the subgraph of G induced by three colours i, j, k may be recoloured by two new colours r, s, in order to construct a proper colouring c' such that $c' \prec^G_{a,3} c$. The implicit algorithm for testing clearly gives a polynomial-time strategy for finding an $(a, 3)$-minimal colouring. By Proposition 9.4.2, a bipartite graph cannot have an $(a, 3)$-minimal colouring of three or more colours. For general graphs, the complexity of MAXIMUM $(a, 3)$-MINIMAL CHROMATIC NUMBER is open, and we conjecture that the decision problem is NP-complete.

9.4.5 (a, k)-minimal graph colourings $(k \geq 4)$

Firstly, we consider the problem of testing a given proper colouring for (a, k)-minimality $(k \geq 4)$. We begin by defining a decision problem, and also a series of decision problems, for each fixed $k \geq 3$, and show each to be NP-complete.

Name: PLANAR GRAPH 3-COLOURABILITY.
Instance: Planar graph $G = (V, E)$.
Question: Is G 3-colourable?
Complexity: NP-complete [92, problem GT4].

Name: GRAPH $(k, k + 1)$-COLOURABILITY.
Instance: Graph $G = (V, E)$ and a proper $(k + 1)$-colouring of G.
Question: Is G k-colourable?

Lemma 9.4.6 *For any fixed $k \geq 3$,* GRAPH $(k, k + 1)$-COLOURABILITY *is NP-complete.*

Proof: Let $k \geq 3$ be fixed. Clearly, GRAPH $(k, k + 1)$-COLOURABILITY is in NP. To show NP-hardness, we give a transformation from PLANAR GRAPH 3-COLOURABILITY, defined above; suppose that $G = (V, E)$ is an instance of this problem. Extend G to a graph G' as follows: form a clique on $k - 3$ new vertices $z_1, z_2, \ldots, z_{k-3}$, and join each $v \in V$ to each z_i $(1 \leq i \leq k)$. As G is planar, a proper 4-colouring of G may be constructed in polynomial time [132, p.34]. This colouring c may be extended to a proper $(k + 1)$-colouring c' of G' by setting $c'(v) = c(v)$ for $v \in V$ and $c'(z_i) = 4 + i$ for $1 \leq i \leq k - 3$. Clearly, G is 3-colourable if and only if G' is k-colourable. ■

For $k \geq 3$, let GRAPH $(k, k + 1)$-COLOURABILITY^C denote the complement of the problem GRAPH $(k, k+1)$-COLOURABILITY. The following result demonstrates that testing a proper graph colouring for (a, k)-minimality is NP-hard, for each fixed $k \geq 4$.

Theorem 9.4.7 *For any fixed $k \geq 3$, $(a, k+1)$-MINIMAL GRAPH COLOURING TEST is NP-hard.*

Proof: Let $G = (V, E)$ (a graph) and c (a proper $(k+1)$-colouring of G) be an instance of GRAPH $(k, k+1)$-COLOURABILITY^C. If c is $(a, k+1)$-minimal, then G is not k-colourable, by Proposition 9.4.2. Conversely, if G is not k-colourable, then clearly c is $(a, k+1)$-minimal. ∎

We now turn to the problem of finding an (a, k)-minimal colouring. We begin by defining a search problem, in order to show that finding an (a, k)-minimal proper colouring in a graph G is hard, where $k \geq 4$.

Name: GRAPH 3-COLOURING SEARCH.
Instance: Graph $G = (V, E)$.
Output: Proper 3-colouring for G, if one exists, or "no", otherwise.
Complexity: NP-hard (as PLANAR GRAPH 3-COLOURABILITY is).

Theorem 9.4.8 *For any fixed $k \geq 4$, (a, k)-MINIMAL GRAPH COLOURING SEARCH is NP-hard.*

Proof: Let $k \geq 4$ be fixed. We give a Turing reduction from GRAPH 3-COLOURING SEARCH; let $G = (V, E)$ be any instance of this problem. Suppose that $S(H)$ is a hypothetical subroutine that, in polynomial time, finds an (a, k)-minimal colouring for a graph H. Call S on the given graph G. If S returns an (a, k)-minimal colouring of G with ≤ 3 colours, then we have our desired output to GRAPH 3-COLOURING SEARCH (a proper 3-colouring for G). Otherwise, we have a "no" answer to GRAPH 3-COLOURING SEARCH, by Proposition 9.4.2. ∎

Finally, we note that MAXIMUM (a, k)-MINIMAL CHROMATIC NUMBER is NP-hard, for each fixed $k \geq 4$.

Theorem 9.4.9 *For any fixed $k \geq 3$, MAXIMUM $(a, k+1)$-MINIMAL CHROMATIC NUMBER DECISION is NP-hard.*

Proof: Let $G = (V, E)$ (a graph) and c (a proper $(k+1)$-colouring of G) be an instance of GRAPH $(k, k+1)$-COLOURABILITY^C, and set $T = k+1$ (the target number of colours for MAXIMUM $(a, k+1)$-MINIMAL CHROMATIC NUMBER DECISION). Suppose that G is not k-colourable. Then c is $(a, k+1)$-minimal. Conversely, suppose that G has an $(a, k+1)$-minimal proper colouring of $\geq T$ colours. Then G is not k-colourable, by Proposition 9.4.2. ∎

9.4.6 $(b, 1)$-minimal graph colourings

Recall from Section 3.3 that a proper colouring c of a graph G is $(b, 1)$-minimal if and only c is b-chromatic, i.e., c satisfies Property 3.2 on Page 35. Testing a proper colouring c of G for b-chromaticity is clearly possible in polynomial time. One need only check, for each colour i of c, that there is a vertex v_i in G, coloured i, and adjacent to a vertex of colour j, for each $j \neq i$. If this is the case, then c is $(b,1)$-minimal; otherwise every vertex of colour i may be recoloured by some colour $j \neq i$, in order to construct a proper colouring c' such that $c' \prec^G_{b,1} c$. It follows that an iteration of this implicit algorithm for testing constitutes a polynomial-time procedure for finding an $(b,1)$-minimal proper colouring. The complexity of B-CHROMATIC NUMBER for various graph classes is studied in Sections 3.4-3.6.

9.4.7 $(b,2)$-minimal graph colourings

As mentioned in the previous section, there exists a convenient criterion to check whether a given proper colouring of a graph G is $\prec^G_b$-minimal. However, a similar situation is unlikely to exist in the case of the partial order $\prec^G_{b,2}$, since the problem of testing a proper graph colouring for $(b,2)$-minimality is NP-hard, as we now demonstrate. The result holds even for bipartite graphs. The transformation begins from the following problem, which concerns *precolouring extensions* in graphs.

Name: PRECOLOURING EXTENSION.
Instance: Graph $G = (V, E)$, subset $S \subseteq V$, integer $t \in \mathbb{Z}^+$ and function $f' : S \longrightarrow \{1, 2, \ldots, t\}$.
Question: Can we extend the precolouring f' to a proper t-colouring f of G, i.e., is there a proper colouring $f : V \longrightarrow \{1, 2, \ldots, t\}$ of G such that $f(u) = f'(u)$ for every $u \in S$?

PRECOLOURING EXTENSION is NP-complete, even for bipartite graphs [128]. Let $(b,2)$-MINIMAL GRAPH COLOURING TEST^C denote the complement of the problem $(b,2)$-MINIMAL GRAPH COLOURING TEST. The following result demonstrates that $(b,2)$-MINIMAL GRAPH COLOURING TEST is NP-hard, even for bipartite graphs.

Theorem 9.4.10 $(b,2)$-MINIMAL GRAPH COLOURING TEST^C *is NP-complete, even for bipartite graphs.*

Proof: Clearly, $(b,2)$-MINIMAL GRAPH COLOURING TEST^C is in NP. To show NP-hardness, we give a reduction from PRECOLOURING EXTENSION for bipartite graphs; let $G = (V, E)$ (bipartite graph), $S \subseteq V$, $t \in \mathbb{Z}^+$ and $f' : S \longrightarrow \{1, 2, \ldots, t\}$ be an instance of this problem. It is clear that we lose no generality by assuming that f' is a partial proper colouring of G. We construct an instance of $(b,2)$-MINIMAL GRAPH COLOURING TEST^C. Define a graph $G' = (V', E')$, where initially $V' = V$ and $E' = E$. Assume that $V = \{u_1, u_2, \ldots, u_n\}$. Add the following vertices and edges to G':

- For each i, j $(1 \leq i \leq t, 1 \leq i < j \leq t+2)$, add new vertices $a_{i,j}, b_{i,j}$, together with the edge $\{a_{i,j}, b_{i,j}\}$.
- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq k \leq t)$, add new vertices $p^k_{i,j}, q^k_{i,j}$, together with the edges $\{a_{i,j}, p^k_{i,j}\}$, $\{b_{i,j}, q^k_{i,j}\}$.
- For each i $(1 \leq i \leq n)$, add a new vertex v_i, together with the edge $\{u_i, v_i\}$.
- For each i $(1 \leq i \leq n)$, if $u_i \in S$, then for each j $(1 \leq j \leq t-1)$, add a new vertex x^j_i, together with the edge $\{u_i, x^j_i\}$.

- For each i, j $(1 \leq i \leq n, 1 \leq j \leq t)$, add a new vertex y_i^j, together with the edge $\{v_i, y_i^j\}$.

It may be verified that G' is bipartite. Now define a proper $(t+2)$-colouring c of G' as follows:

- For each i, j $(1 \leq i \leq t, 1 \leq i < j \leq t+2)$ set $c(a_{i,j}) = i$ and set $c(b_{i,j}) = j$.
- For each i, j $(1 \leq i \leq t, 1 \leq i < j \leq t+2)$, colour the $p_{i,j}^k$ vertices $(1 \leq k \leq t)$ such that
$$\{c(p_{i,j}^1), c(p_{i,j}^2), \ldots, c(p_{i,j}^t)\} = \{1, 2, \ldots, t+2\} \backslash \{i, j\}.$$
- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq k \leq t)$, set $c(q_{i,j}^k) = c(p_{i,j}^k)$.
- For each i $(1 \leq i \leq n)$ such that $u_i \in S$, colour the x_i^j vertices $(1 \leq j \leq t-1)$ such that
$$\{c(x_i^1), c(x_i^2), \ldots, c(x_i^{t-1})\} = \{1, 2, \ldots, t\} \backslash \{f'(u_i)\}.$$
- For each i, j $(1 \leq i \leq n, 1 \leq j \leq t)$, set $c(y_i^j) = j$.
- To complete the definition of c, we assign colours to the u_i and v_i vertices $(1 \leq i \leq n)$ from the set $\{t+1, t+2\}$, using the bipartite property of G'.

The claim is that f' can be extended to a proper t-colouring f of G if and only if c is a non-$(b, 2)$-minimal colouring of G'.

For, suppose that there is a proper t-colouring f of G, such that $f(u) = f'(u)$, for all $u \in S$. We define a proper $(t+1)$-colouring c' of G', as follows.

- For each i $(1 \leq i \leq n)$, set $c'(u_i) = f(u_i)$ and set $c'(v_i) = t+3$.
- For each i, j $(1 \leq i \leq t, 1 \leq i < j \leq t+2)$, if $c(b_{i,j}) \in \{t+1, t+2\}$, then set $c'(b_{i,j}) = t+3$.
- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2,\ 1 \leq k \leq t)$, if $c(p_{i,j}^k) \in \{t+1, t+2\}$, then set $c'(p_{i,j}^k) = t+3$.
- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2,\ 1 \leq k \leq t)$, if $c(q_{i,j}^k) \in \{t+1, t+2\}$, then set $c'(q_{i,j}^k) = c(a_{i,j})$.

For every other vertex $u \in V'$, set $c'(u) = c(u)$. It may be verified that c' is a proper colouring of G', since f is an extension of f' in G, and since no vertex among the $a_{i,j}$ $(1 \leq i \leq t, 1 \leq i < j \leq t+2)$ has colour from the set $\{t+1, t+2\}$ in the colouring c. Finally, as colouring c' has been obtained from colouring c by redistributing the vertices of colours $t+1, t+2$ among the remaining colours, plus one new colour, namely $t+3$, then c is non-$(b, 2)$-minimal.

Conversely, suppose that the proper colouring c of G' is non-$(b, 2)$-minimal. Then there is a proper colouring c' of G', obtained from c by redistributing the vertices of two colours, i, j, among the remaining colours, plus one new colour, $t+3$, say, whilst each of the other vertices retains its original colour in the colouring c. The claim is that $\{i, j\} = \{t+1, t+2\}$. For, suppose not. Without loss of generality $i < j$, so that $1 \leq i \leq t$. As $c(a_{i,j}) = i$ and $c(b_{i,j}) = j$, then the two vertices $a_{i,j}, b_{i,j}$ in particular are reassigned colours in the colouring c'. But both $a_{i,j}$ and $b_{i,j}$ are adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t+2\} \backslash \{i, j\}$. Hence $c'(a_{i,j}) = c'(b_{i,j}) = t+3$, a contradiction, since $\{a_{i,j}, b_{i,j}\} \in E'$. Thus the claim is established.

For any i $(1 \leq i \leq n)$, $c(v_i) \in \{t+1, t+2\}$, so v_i is reassigned a colour in the colouring c'. But v_i is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t\}$, which implies that $c'(v_i) = t+3$. Similarly, for any i $(1 \leq i \leq n)$, $c(u_i) \in \{t+1, t+2\}$, so u_i is reassigned a colour in the colouring c'. But $\{u_i, v_i\} \in E'$, so that $c'(u_i) \leq t$. If $u_i \in S$, then u_i is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t\} \backslash \{f'(u_i)\}$. Hence $c'(u_i) = f'(u_i)$, which implies that the function $f : V \longrightarrow \{1, 2, \ldots, t\}$, defined by $f(u) = c'(u)$ for all $u \in V$, is an extension of f', and a proper t-colouring of G. ∎

Clearly, the graph G' constructed in Theorem 9.4.10 is disconnected. However, it is straightforward to verify that the result holds for connected graphs, by making appropriate connections between the $q^k_{i,j}$ vertices and a y^m_l vertex, for example, in the above construction.

The complexity of the problem of finding a $(b, 2)$-minimal colouring is open. However, it is quite possible that the problem is polynomial-time solvable, despite the NP-hardness of $(b, 2)$-MINIMAL GRAPH COLOURING TEST, as is the case for the problems of testing a bin packing for $\prec^x_b$-minimality, and finding a $\prec^x_b$-minimal bin packing (see Section 9.3).

Similarly, the complexity of MAXIMUM $(b, 2)$-MINIMAL CHROMATIC NUMBER is open – we conjecture that this problem is NP-hard.

9.4.8 $(b, 3)$-minimal graph colourings

The construction of Theorem 9.4.10 may be extended in order to show that $(b, 3)$-MINIMAL GRAPH COLOURING TEST is NP-hard. This is demonstrated by the following result. Let $(b, 3)$-MINIMAL GRAPH COLOURING TEST^C denote the complement of the problem $(b, 3)$-MINIMAL GRAPH COLOURING TEST. We now show that $(b, 3)$-MINIMAL GRAPH COLOURING TEST is NP-hard.

Theorem 9.4.11 $(b, 3)$-MINIMAL GRAPH COLOURING TEST^C *is NP-complete.*

Proof: Clearly, $(b, 3)$-MINIMAL GRAPH COLOURING TEST^C is in NP. To show NP-hardness, we give a reduction from PRECOLOURING EXTENSION for bipartite graphs, defined in the previous section; let $G = (V, E)$ (bipartite graph), $S \subseteq V$, $t \in \mathbb{Z}^+$ and $f' : S \longrightarrow \{1, 2, \ldots, t\}$ be an instance of this problem. It is clear that we lose no generality by assuming that f' is a partial proper colouring of G. We construct an instance of $(b, 3)$-MINIMAL GRAPH COLOURING TEST^C. Define a graph $G' = (V', E')$, where initially $V' = V$ and $E' = E$. Assume that $V = \{u_1, u_2, \ldots, u_n\}$. Add the following vertices and edges to G':

- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$, add new vertices $a_{i,j,k}$, $b_{i,j,k}$, $c_{i,j,k}$, together with the edges $\{a_{i,j,k}, b_{i,j,k}\}$, $\{b_{i,j,k}, c_{i,j,k}\}$ and $\{a_{i,j,k}, c_{i,j,k}\}$.

- For each i, j, k, l $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3, 1 \leq l \leq t)$, add new vertices $p^l_{i,j,k}$, $q^l_{i,j,k}$, $r^l_{i,j,k}$, together with the edges $\{a_{i,j,k}, p^l_{i,j,k}\}$, $\{b_{i,j,k}, q^l_{i,j,k}\}$ and $\{c_{i,j,k}, r^l_{i,j,k}\}$.

- For each i $(1 \leq i \leq n)$, add new vertices v_i, w_i, together with the edges $\{u_i, v_i\}$, $\{v_i, w_i\}$ and $\{u_i, w_i\}$.

- For each i $(1 \leq i \leq n)$, if $u_i \in S$, then for each j $(1 \leq j \leq t-1)$, add a new vertex x^j_i, together with the edge $\{u_i, x^j_i\}$.

- For each i, j $(1 \leq i \leq n, 1 \leq j \leq t)$, add new vertices y^j_i, z^j_i, together with the edges $\{v_i, y^j_i\}$ and $\{w_i, z^j_i\}$.

Now define a proper $(t+3)$-colouring c of G' as follows:

- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$ set $c(a_{i,j,k}) = i$, set $c(b_{i,j,k}) = j$ and set $c(c_{i,j,k}) = k$.

- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$, colour the $p^l_{i,j,k}$ vertices $(1 \leq l \leq t)$ such that

$$\{c(p^1_{i,j,k}), c(p^2_{i,j,k}), \ldots, c(p^t_{i,j,k})\} = \{1, 2, \ldots, t+3\} \backslash \{i, j, k\}.$$

- For each i, j, k, l $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3, 1 \leq l \leq t)$, set $c(r^l_{i,j,k}) = c(q^l_{i,j,k}) = c(p^l_{i,j,k})$.

- For each i $(1 \leq i \leq n)$ such that $u_i \in S$, colour the x^j_i vertices $(1 \leq j \leq t-1)$ such that

$$\{c(x^1_i), c(x^2_i), \ldots, c(x^{t-1}_i)\} = \{1, 2, \ldots, t\} \backslash \{f'(u_i)\}.$$

- For each i, j $(1 \leq i \leq n, 1 \leq j \leq t)$, set $c(y^j_i) = c(z^j_i) = j$.

- For each i $(1 \leq i \leq n)$, set $c(w_i) = t+3$.

- To complete the definition of c, we assign colours to the u_i and v_i vertices $(1 \leq i \leq n)$ from the set $\{t+1, t+2\}$, since the subgraph of G' induced by the u_i, v_i vertices $(1 \leq i \leq n)$ is bipartite.

The claim is that f' can be extended to a proper t-colouring f of G if and only if c is a non-$(b, 3)$-minimal colouring of G'.

For, suppose that there is a proper t-colouring f of G, such that $f(u) = f'(u)$, for all $u \in S$. We define a proper $(t+2)$-colouring c' of G', as follows.

- For each i $(1 \leq i \leq n)$, set $c'(u_i) = f(u_i)$, set $c'(v_i) = t+4$ and set $c'(w_i) = t+5$.

- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$, if $c(b_{i,j,k}) \in \{t+1, t+2\}$, then set $c'(b_{i,j,k}) = t+4$.
- For each i, j, k $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$, if $c(c_{i,j,k}) \in \{t+1, t+2, t+3\}$, then set $c'(c_{i,j,k}) = t+5$.
- For each i, j, k, l $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3, 1 \leq l \leq t)$, if $c(p^l_{i,j,k}) \in \{t+1, t+2, t+3\}$, then set $c'(p^l_{i,j,k}) = t+4$.
- For each i, j, k, l $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3, 1 \leq l \leq t)$, if $c(q^l_{i,j,k}) \in \{t+1, t+2, t+3\}$, then set $c'(q^l_{i,j,k}) = t+5$.
- For each i, j, k, l $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3, 1 \leq l \leq t)$, if $c(r^l_{i,j,k}) \in \{t+1, t+2, t+3\}$, then set $c'(r^l_{i,j,k}) = t+4$.

For every other vertex $u \in V'$, set $c'(u) = c(u)$. It may be verified that c' is a proper colouring of G', since f is an extension of f' in G, and since no vertex among the $a_{i,j,k}$ $(1 \leq i \leq t, 1 \leq i < j \leq t+2, 1 \leq i < j < k \leq t+3)$ has colour from the set $\{t+1, t+2, t+3\}$ in the colouring c. Finally, as colouring c' has been obtained from colouring c by redistributing the vertices of colours $t+1, t+2, t+3$ among the remaining colours, plus two new colours, namely $t+4, t+5$, then c is non-$(b, 3)$-minimal.

Conversely, suppose that the proper colouring c of G' is non-$(b, 3)$-minimal. Then there is a proper colouring c' of G', obtained from c by redistributing the vertices of three colours, i, j, k, among the remaining colours, plus two new colours, $t+4, t+5$, say, whilst each of the other vertices retains its original colour in the colouring c. The claim is that $\{i, j, k\} = \{t+1, t+2, t+3\}$. For, suppose not. Without loss of generality $i < j < k$, so that $1 \leq i \leq t$. As $c(a_{i,j,k}) = i$, $c(b_{i,j,k}) = j$ and $c(c_{i,j,k}) = k$, then the three vertices $a_{i,j,k}$, $b_{i,j,k}$, $c_{i,j,k}$ in particular are reassigned colours in the colouring c'. But each of $a_{i,j,k}$, $b_{i,j,k}$ and $c_{i,j,k}$ is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t+3\} \backslash \{i, j, k\}$. Hence, without loss of generality, $c'(a_{i,j,k}) = t+4$ and $c'(b_{i,j,k}) = t+5$. But then there is no available colour for $c_{i,j,k}$ in the colouring c', a contradiction. Thus the claim is established.

For any i $(1 \leq i \leq n)$, $c(w_i) = t+3$, so w_i is reassigned a colour in the colouring c'. But w_i is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t\}$, which implies that $c'(w_i) \geq t+4$. Similarly, for any i $(1 \leq i \leq n)$, $c(v_i) \in \{t+1, t+2\}$, so v_i is reassigned a colour in the colouring c'. But v_i is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t\}$, which implies that $\{c'(v_i), c'(w_i)\} = \{t+4, t+5\}$. Finally, for any i $(1 \leq i \leq n)$, $c(u_i) \in \{t+1, t+2\}$, so u_i is reassigned a colour in the colouring c'. But $\{u_i, v_i\} \in E'$ and $\{u_i, w_i\} \in E'$, so that $c'(u_i) \leq t$. If $u_i \in S$, then u_i is adjacent in the colouring c to vertices of colours $\{1, 2, \ldots, t\} \backslash \{f'(u_i)\}$. Hence $c'(u_i) = f'(u_i)$, which implies that the function $f : V \longrightarrow \{1, 2, \ldots, t\}$, defined by $f(u) = c'(u)$ for all $u \in V$, is an extension of f', and a proper t-colouring of G. ∎

The extension of the construction of Theorem 9.4.10 may be generalised beyond that of Theorem 9.4.11 in order to show NP-hardness for (b, k)-MINIMAL GRAPH COLOURING

TEST, for each $k \geq 4$. However, a simpler transformation exists in the case $k \geq 4$, which will be discussed in the following section.

As for the $(b, 2)$-minimal case, the complexity of $(b, 3)$-MINIMAL GRAPH COLOURING SEARCH is open, as is the complexity of MAXIMUM $(b, 3)$-MINIMAL CHROMATIC NUMBER in general graphs, though we conjecture that the latter problem is NP-hard. However, by Proposition 9.4.3, a bipartite graph cannot have a $(b, 3)$-minimal colouring of three or more colours.

9.4.9 (b, k)-minimal graph colourings $(k \geq 4)$

It is a straightforward matter to adapt the proofs of Theorems 9.4.7, 9.4.8 and 9.4.9, in order to show that each of the problems of testing a proper colouring for (b, k)-minimality, finding a (b, k) minimal colouring and maximising the number of colours over all (b, k)-minimal colourings are NP-hard, for any fixed $k \geq 4$. For, we simply replace instances of 'by Proposition 9.4.2' with 'by Corollary 9.4.3' in the relevant proofs. We thus have:

Theorem 9.4.12 *For any fixed $k \geq 4$,* (b, k)-MINIMAL GRAPH COLOURING TEST *is NP-hard.*

Theorem 9.4.13 *For any fixed $k \geq 4$,* (b, k)-MINIMAL GRAPH COLOURING SEARCH *is NP-hard.*

Theorem 9.4.14 *For any fixed $k \geq 4$,* MAXIMUM (b, k)-MINIMAL CHROMATIC NUMBER *is NP-hard.*

9.4.10 Conclusion and open problems relating to (a, k)-minimal $(k \geq 2)$ and (b, k)-minimal $(k \geq 1)$ proper graph colourings

Table 9.1 contains a summary of the algorithmic results relating to (a, k)-minimal $(k \geq 2)$ and (b, k)-minimal $(k \geq 1)$ graph colourings that appear in this section. The complexities of the corresponding testing, finding and maximisation problems are indicated in columns 2,3 and 4-6 respectively. Columns 4,5 and 6 indicate the complexity of the maximisation problem in arbitrary graphs, bipartite graphs and trees, respectively. In a table entry, 'P' denotes polynomial-time solvablity for the problem concerned, 'NPC' denotes NP-completeness, 'co-NPC' denotes co-NP-completeness, and 'NPH' denotes NP-hardness. A question mark indicates that the corresponding problem is open.

One is led to ask whether the consideration of proper colourings that are minimal with respect to suitably chosen partial orders from the two partial order families might yield a worthwhile strategy for approximating the chromatic number in certain graph classes. The above table shows that the threshold between polynomial-time solvability and NP-completeness for the testing and finding problems corresponding to (a, k)-minimal colourings occurs between $k = 3$ and $k = 4$. Similarly, the threshold for the testing problems corresponding to (b, k)-minimality occurs between $k = 1$ and $k = 2$, while the finding threshold occurs somewhere between $k = 1$ and $k = 4$. Thus $\prec^G_{a,3}$ and $\prec^G_{b,1}$ are the finest partial orders within each of the two partial order families for which the finding

Criterion for minimality	Complexity of testing for minimality	Complexity of finding a minimal colouring	Complexity of maximum minimal problem		
			Arbitrary	Bipartite	Tree
$(a, 2)$	P	P	NPC	NPC	NPC
$(a, 3)$	P	P	?	P	P
(a, k) $(k \geq 4)$	co-NPC	NPH	NPH	P	P
$(b, 1)$	P	P	NPC	NPC	P
$(b, 2)$	co-NPC	?	?	?	?
$(b, 3)$	co-NPC	?	?	P	P
(b, k) $(k \geq 4)$	co-NPC	NPH	NPH	P	P

Table 9.1: Summary of algorithmic results appearing in Section 9.4.

problem is known to be polynomial-time solvable. It is possible that a local search strategy based on finding proper colourings that are both $(a, 3)$-minimal and $(b, 1)$-minimal might yield improved approximability results for the chromatic number in certain graph classes. Indeed, any such results could be further enhanced if it turns out that the problems of finding a $(b, 2)$-minimal colouring (or indeed a $(b, 3)$-minimal colouring) are polynomial-time solvable.

9.5 Partial orders defined on $\mathcal{U}(x)$

Recall from Definition 2.3.5 that a minimaximal or maximinimal optimisation problem may be obtained from a source optimisation problem Π using a partial order $\prec^x$, defined on the set of all feasible solutions $\mathcal{F}(x)$ for a given instance x of Π, and satisfying POMM with respect to Π. Suppose that we define $\prec^x$ on $\mathcal{U}(x)$, the set of all possible solutions for a given instance x of Π, rather than on $\mathcal{F}(x)$. It is interesting to consider how the $\prec^x$-optimal solutions relate to those that would be obtained by defining $\prec^x$ on $\mathcal{F}(x)$, for a given instance x of Π. For $\prec^x$ defined on $\mathcal{U}(x)$, the definitions of $\prec^x$-maximality and $\prec^x$-minimality given by Definition 2.3.2 still apply. That is, an element $s \in \mathcal{F}(x)$ is $\prec^x$-maximal if there is no $t \in \mathcal{F}(x)$ such that $s \prec^x t$, and an element $t \in \mathcal{F}(x)$ is $\prec^x$-minimal if there is no $s \in \mathcal{F}(x)$ such that $s \prec^x t$. However, the process by which we *test* for these conditions holding may not be immediately obvious.

For example, consider the CHROMATIC NUMBER problem (defined in Section 3.1), with $\prec_b^G$, the partial order of partition redistribution, defined on $\mathcal{F}(G)$, the set of all proper colourings of G, for a given graph G. We test a proper colouring c for $\prec_b^G$-minimality by ensuring that Property 3.2 on Page 35 holds. This shows that there exists no $c' \in \mathcal{F}(G)$ such that $c' \sqsubset_b^G c$. Immediately we deduce that c is $\prec_b^G$-minimal, for if there is some $c' \in \mathcal{F}(G)$ such that $c' \prec_b^G c$, then there is some chain

$$c' = c_1 \sqsubset_b^G c_2 \sqsubset_b^G \ldots \sqsubset_b^G c_{n-1} \sqsubset_b^G c_n = c \tag{9.1}$$

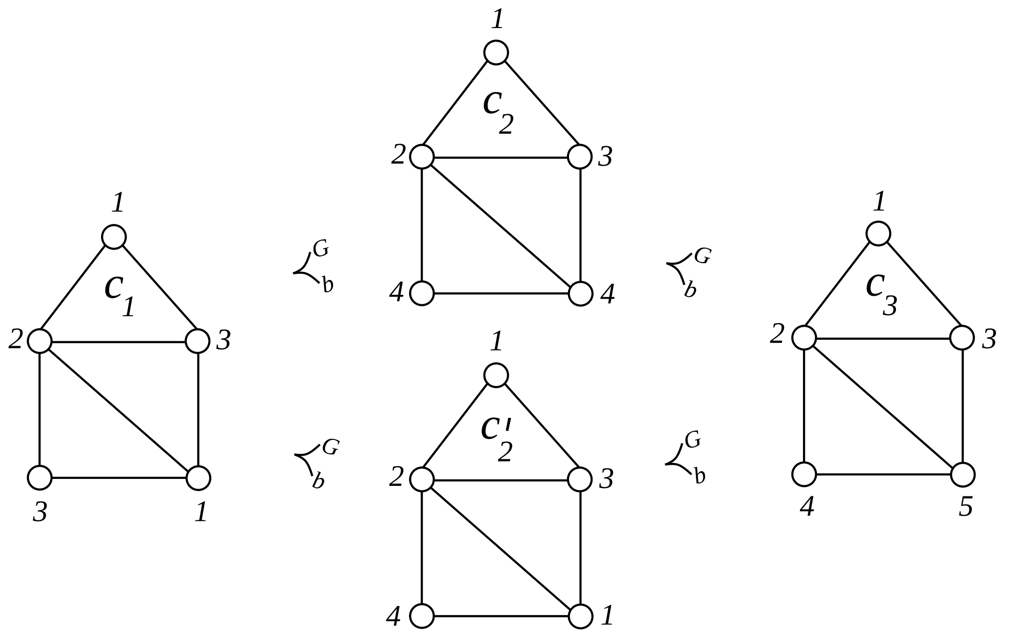

Figure 9.3: Example colourings related by the redefined partial order $\prec_b^G$.

for some $n \geq 2$, such that $c_i \in \mathcal{F}(G)$ for $2 \leq i \leq n-1$. Thus in particular $c_{n-1} \in \mathcal{F}(G)$, a contradiction.

However if $\sqsubset_b^G$, and consequently $\prec_b^G$, is now defined on $\mathcal{U}(G)$, the set of all colourings of G, then it does not follow in general that $c_i \in \mathcal{F}(G)$ $(2 \leq i \leq n-1)$ in (9.1) above. In fact, it is simple to find two proper colourings $c_1, c_3 \in \mathcal{F}(G)$, where $c_1 \prec_b^G c_3$, such that there is a colouring $c_2 \in \mathcal{U}(G) \backslash \mathcal{F}(G)$ with $c_1 \sqsubset_b^G c_2 \sqsubset_b^G c_3$. Such an example is shown in Figure 9.3. However there is a proper colouring c'_2 such that $c_1 \sqsubset_b^G c'_2 \sqsubset_b^G c_3$; this situation is again illustrated in Figure 9.3. Thus, when carrying out two or more redistributions in order to derive a feasible $\prec_b^G$-predecessor c' of a feasible element c, it is possible that the intermediate colourings in the redistribution series are infeasible. Hence the question arises as to how we may test for $\prec_b^G$-minimality in this case.

Later in this section, we show, as in the example of Figure 9.3, that given $c, c' \in \mathcal{F}(G)$ with $c' \prec_b^G c$ ($\prec_b^G$ is still assumed to be defined on $\mathcal{U}(G)$), we can always find a series of elements c_i for $2 \leq i \leq n-1$ such that

$$c' = c_1 \sqsubset_b^G c_2 \sqsubset_b^G \ldots \sqsubset_b^G c_{n-1} \sqsubset_b^G c_n = c$$

for some $n \geq 2$, where $c_i \in \mathcal{F}(G)$ for $2 \leq i \leq n-1$. In fact, we demonstrate that this applies not only to B-CHROMATIC NUMBER, but a whole class of minimaximal and maximinimal optimisation problems whose inherent partial order satisfies a certain additional property. This class includes many of the minimaximal and maximinimal optimisation problems studied in this thesis. Essentially, the minimaximal or maximinimal optimisation problem concerned has the same behaviour as that which would result by defining the inherent partial order on $\mathcal{U}(x)$, for a given instance x. Thus our observations yield a greater insight into the structure of minimaximal and maximinimal optimisation problems.

The remainder of Section 9.5 is organised as follows. In Section 9.5.1, we prove a result of the form discussed in the previous paragraph for a source optimisation problem Π that is *$\prec$-hereditary* or *$\prec$-super-hereditary*, where $\prec^x$ is a partial order defined on $\mathcal{U}(x)$, for a given instance x of Π. The concepts of $\prec$-hereditary and $\prec$-super-hereditary optimisation problem are generalisations of the notions of hereditary and super-hereditary optimisation problems (given by Definition 2.4.5) respectively, for arbitrary partial orders, including the partial order of set inclusion. In Section 9.5.2, we do likewise for a source optimisation problem Π that is *partition-hereditary* or *partition-super-hereditary* – conditions involving the partial orders of partition merge and partition redistribution. We finish by discussing, in Section 9.5.3, why it does not follow for all optimisation problems in general that the partial order should be defined on $\mathcal{U}(x)$.

For the remainder of this Section 9.5, all partial orders are now assumed to be defined on $\mathcal{U}(x)$, rather than on $\mathcal{F}(x)$, for a given instance x of an optimisation problem.

9.5.1 $\prec$-hereditary and $\prec$-super-hereditary optimisation problems

We begin this section by presenting a definition concerning feasibility predicates of optimisation problem that are hereditary or super-hereditary with respect to some partial order $\prec$. As mentioned above, this definition generalises Definition 2.4.5.

Definition 9.5.1 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem, and for $x \in \mathcal{I}$, let $\prec^x$ be a partial order defined on $\mathcal{U}(x)$, satisfying POMM with respect to Π. Property π is *$\prec$-hereditary* if, for any $x \in \mathcal{I}$, whenever $s \in \mathcal{U}(x)$, $t \in \mathcal{F}(x)$ and $s \prec^x t$, then $s \in \mathcal{F}(x)$. Property π is *$\prec$-super-hereditary* if, for any $x \in \mathcal{I}$, whenever $s \in \mathcal{F}(x)$, $t \in \mathcal{U}(x)$ and $s \prec^x t$, then $t \in \mathcal{F}(x)$. ■

Given an optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ and a partial order $\prec^x$, defined on $\mathcal{U}(x)$ and satisfying POMM with respect to Π, for a given instance x of Π, we say that Π is $\prec$-hereditary or $\prec$-super-hereditary if π is $\prec$-hereditary or $\prec$-super-hereditary, respectively. Although we invoke Definition 9.5.1 mainly with the partial order of set inclusion, we also demonstrate that the definition applies to certain optimisation problems together with the subsequence and substring partial orders, and the partial order on truth assignments. (A corresponding definition for partial orders on partitions is given in the next section.)

We now consider the source optimisation problems Π in this thesis for which we have derived minimaximal or maximinimal optimisation problems using an appropriate partial order $\prec^x$ defined on the feasible solutions for a given instance x of Π. We give a series of results that indicate, in each case, whether Π is $\prec$-hereditary or $\prec$-super-hereditary.

Starting with Chapter 3, CHROMATIC NUMBER, with partial orders $\prec_a$ and $\prec_b$, is discussed in Section 9.5.2. We now consider the source optimisation problems and partial orders studied in Chapter 4.

Proposition 9.5.2 *The following source optimisation problems from Chapter 4 are $\subset$-hereditary:*

- MAXIMUM TOTAL MATCHING
- MAXIMUM MATCHING
- MAXIMUM CLIQUE
- MAXIMUM INDEPENDENT SET
- MAXIMUM STRONG STABLE SET
- MAXIMUM IRREDUNDANT SET

The following source optimisation problems from Chapter 4 are $\subset$-super-hereditary:

- MINIMUM TOTAL COVER
- MINIMUM EDGE COVER
- MINIMUM TOTAL DOMINATING SET
- MINIMUM VERTEX COVER
- MINIMUM DOMINATING SET
- MINIMUM EDGE DOMINATING SET

We now consider the source optimisation problems discussed in Chapter 5. As discussed in Section 5.2, both MINIMUM NEARLY PERFECT SET and MAXIMUM NEARLY PERFECT SET are neither $\subset$-hereditary nor $\subset$-super-hereditary. Regarding Section 5.3, we give an example in Section 9.5.3 to show that MAXIMUM INDEPENDENT SET is neither $\subset_2$-hereditary nor $\subset_2$-super-hereditary. Consideration of the source optimisation problems and partial orders discussed in Chapter 7 gives the following result.

Proposition 9.5.3 *The following source optimisation problems from Chapter 7 are $\prec$-hereditary, where $\prec$ is the undernoted partial order:*

- LONGEST PATH, $\overline{\ll}$
- MAXIMUM 3D-MATCHING, $\subset$
- LCSt, $\overline{\ll}$
- MAXIMUM 2SAT, $\prec_t$
- UNCONSTRAINED LONGEST PATH, $\overline{\ll}$
- LCS, $\ll$
- MAXIMUM KNAPSACK, $\subset$
- MAXIMUM ONE-IN-THREE 3SAT, $\prec_t$

The following source optimisation problems from Chapter 7 are $\prec$-super-hereditary, where $\prec$ is the undernoted partial order:

- LONGEST PATH, $\overline{\ll}$
- SCS, $\ll$
- MAXIMUM 2SAT, $\prec_t$
- MINIMUM TEST SET, $\subset$
- SCSt, $\overline{\ll}$

It may be verified that both LONGEST PATH and UNCONSTRAINED LONGEST PATH are neither $\ll$-hereditary nor $\ll$-super-hereditary. The MINIMUM BIN PACKING problem with partial orders $\prec_a$ and $\prec_b$ is discussed in Section 9.5.2. The following result indicates the source optimisation problems from Chapter 8 that are $\subset$-hereditary or $\subset$-super-hereditary.

Proposition 9.5.4 MAXIMUM SET PACKING *is $\subset$-hereditary.* MINIMUM SET COVER *and* MINIMUM HITTING SET *are both $\subset$-super-hereditary.*

We now present the main result of this section. We demonstrate how to test for $\prec^x$-optimality in the case of a source optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$, given a partial order $\prec^x$, defined on $\mathcal{U}(x)$, for a given instance x of Π, where π is $\prec$-hereditary or $\prec$-super-hereditary.

Theorem 9.5.5 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem, and for $x \in \mathcal{I}$, suppose that $\prec^x$ is a partial order defined on $\mathcal{U}(x)$ satisfying POMM. Then:*

1. *If* OPT $=$ min, *and* $c \in \mathcal{F}(x)$ *is such that no immediate* $\prec^x$*-predecessor of* c *is in* $\mathcal{F}(x)$, *and* π *is* $\prec$*-hereditary or* $\prec$*-super-hereditary, then* c *is* $\prec^x$*-minimal.*

2. *If* OPT $=$ max, *and* $c \in \mathcal{F}(x)$ *is such that no immediate* $\prec^x$*-successor of* c *is in* $\mathcal{F}(x)$, *and* π *is* $\prec$*-hereditary or* $\prec$*-super-hereditary, then* c *is* $\prec^x$*-maximal.*

Proof: We prove (1); the proof of (2) is similar. Suppose that c is not $\prec^x$-minimal. Then there exists some $c' \in \mathcal{F}(x)$ such that $c' \prec^x c$. By hypothesis, c' is not an immediate $\prec^x$-predecessor of c, so there exists some $c_i \in \mathcal{U}(x)$, for $1 \le i \le n$, where $n \ge 3$, such that

$$c' = c_1 \prec^x c_2 \prec^x \ldots \prec^x c_{n-1} \prec^x c_n = c$$

and c_i is an immediate $\prec^x$-predecessor of c_{i+1} $(1 \le i \le n-1)$. But then $c_i \in \mathcal{F}(x)$ $(2 \le i \le n-1)$, using either the $\prec$-hereditary or $\prec$-super-hereditary property of π, since $c_1 \in \mathcal{F}(x)$ and $c_n \in \mathcal{F}(x)$. In particular, $c_{n-1} \in \mathcal{F}(x)$, a contradiction. Thus c is $\prec^x$-minimal. ■

A consequence of Theorem 9.5.5 is that the $\prec^x$-optimal solutions are identical to those that result from defining $\prec^x$ on $\mathcal{F}(x)$, for a given instance x, when the $\prec$-hereditary or $\prec$-super-hereditary property is satisfied by the optimisation problem in question. This holds for any Π and $\prec$ chosen from the source optimisation problems and partial orders listed in Propositions 9.5.2, 9.5.3 and 9.5.4.

9.5.2 Partition-hereditary and partition-super-hereditary optimisation problems

In this section, we prove similar results to that of Theorem 9.5.5, for a source optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ and a partial order $\prec^x$, where $\prec^x$ is the partial order of partition merge or partition redistribution, defined on the set $\mathcal{U}(x)$, for an instance x of Π. We begin by defining a property for π, similar to the concept of hereditary-ness, or super-hereditary-ness, when $\prec^x$ is a partial order on partitions.

Definition 9.5.6 Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \text{OPT} \rangle$ be an optimisation problem. For any $x \in \mathcal{I}$, suppose that there is some set S^x, associated with x, such that $\mathcal{U}(x)$ is a set of partitions of S^x. Suppose further that π admits a *secondary predicate* π_s which satisfies the following property:

$$\forall \{U_1, U_2, \ldots, U_k\} \in \mathcal{U}(x) \bullet (\pi(U_1, U_2, \ldots, U_k) \Leftrightarrow (\forall\, 1 \le i \le k \bullet \pi_s(U_i))).$$

Predicate π is *partition-hereditary* if, for any $x \in \mathcal{I}$,

$$\forall\, P \in \mathcal{U}(x) \bullet \forall\, U \in P \bullet ((\pi_s(U) \;\wedge\; U' \subseteq U) \Rightarrow \pi_s(U')).$$

Predicate π is *partition-super-hereditary* if, for any $x \in \mathcal{I}$,

$$\forall\, P \in \mathcal{U}(x) \bullet \forall\, U \in P \bullet ((\pi_s(U) \;\wedge\; U \subseteq U') \Rightarrow \pi_s(U')). \blacksquare$$

Suppose that $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ is an optimisation problem. We say that Π is partition-hereditary or partition-super-hereditary if π is partition-hereditary or partition-super-hereditary, respectively.

Consideration of source optimisation problems in this thesis to which we have applied a partial order on partitions gives rise to the following proposition.

Proposition 9.5.7 CHROMATIC NUMBER *and* MINIMUM BIN PACKING *(defined in Sections 3.1 and 7.4.1 respectively) are both partition-hereditrary.*

We now present the first main result of this section. Suppose that $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ is a partition-hereditary or partition-super-hereditary optimisation problem. Suppose further that $\prec_a^x$, the partial order of partition merge, is defined on $\mathcal{U}(x)$, for a given instance x of Π. The following result shows how to test a feasible solution for $\prec_a^x$-optimality.

Theorem 9.5.8 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem, and for $x \in \mathcal{I}$, let $\prec_a^x$, the partial order of partition merge, be defined $\mathcal{U}(x)$, satisfying POMM with respect to* Π.

1. *Suppose that $P \in \mathcal{F}(x)$ is such that there is no $P' \in \mathcal{F}(x)$ with $P' \sqsubset_a^x P$. Suppose further that π is partition-hereditary. Then P is $\prec_a^x$-minimal.*
2. *Suppose that $P \in \mathcal{F}(x)$ is such that there is no $P' \in \mathcal{F}(x)$ with $P \sqsubset_a^x P'$. Suppose further that π is partition-super-hereditary. Then P is $\prec_a^x$-maximal.*

Proof: We prove (1); the proof of (2) is similar. Suppose that there is some $P' \in \mathcal{F}(x)$ such that $P' \prec_a^x P$. Then there exists some $P_i \in \mathcal{U}(x)$ $(1 \leq i \leq n)$ such that

$$P' = P_1 \sqsubset_a^x P_2 \sqsubset_a^x \ldots \sqsubset_a^x P_{n-1} \sqsubset_a^x P_n = P.$$

By hypothesis, $P_{n-1} \notin \mathcal{F}(x)$, so $n \geq 3$. As $P_{n-1} \notin \mathcal{F}(x)$, then there is some $U_{n-1} \in P_{n-1}$ such that $\pi_s(U_{n-1})$ does not hold, where π_s is the secondary predicate of π. But $P_1 \prec_a^x P_{n-1}$, so there is some $U_1 \in P_1$ such that $U_{n-1} \subseteq U_1$. Thus, by the partition-hereditary property, $\pi_s(U_1)$ does not hold, so that $P_1 \notin \mathcal{F}(x)$, a contradiction. Thus P is $\prec_a^x$-minimal. ∎

A consequence of Theorem 9.5.8 is that the $\prec_a^x$-optimal solutions are identical to those that result from defining $\prec_a^x$ on $\mathcal{F}(x)$, for a given instance x, when the partition-hereditary or partition-super-hereditary property is satisfied by the optimisation problem in question.

The second main result is similar to that of Theorem 9.5.8. Suppose that $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ is a partition-hereditary or partition-super-hereditary optimisation problem. Suppose further that $\prec_b^x$, the partial order of partition redistribution, is defined on $\mathcal{U}(x)$, for a given instance x of Π. The following result shows how to test a feasible solution for $\prec_b^x$-optimality.

Theorem 9.5.9 *Let $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$ be an optimisation problem. For any $x \in \mathcal{I}$, suppose that there is some set S^x, associated with x, such that $\mathcal{U}(x)$ is a set of partitions* *of S^x. For $x \in \mathcal{I}$, let $\prec_b^x$, the partial order of partition redistribution, be defined $\mathcal{U}(x)$, satisfying POMM with respect to Π.*

1. *Suppose that $P \in \mathcal{F}(x)$ is such that there is no $P' \in \mathcal{F}(x)$ with $P' \sqsubset_b^x P$. Suppose further that π is partition-hereditary. Then P is $\prec_b^x$-minimal.*
2. *Suppose that $P \in \mathcal{F}(x)$ is such that there is no $P' \in \mathcal{F}(x)$ with $P \sqsubset_b^x P'$. Suppose further that π is partition-super-hereditary. Then P is $\prec_b^x$-maximal.*

Proof of (1): Suppose that there is some $P' \in \mathcal{F}(x)$ such that $P' \prec_b^x P$. Then there exists some $P_i \in \mathcal{U}(x)$ $(0 \leq i \leq n-1)$ such that

$$P' = P_0 \sqsubset_b^x P_1 \sqsubset_b^x \ldots \sqsubset_b^x P_{n-2} \sqsubset_b^x P_{n-1} = P.$$

By hypothesis, $P_{n-2} \notin \mathcal{F}(x)$, so $n \geq 3$. We prove that there exists a chain

$$P' = P'_0 \sqsubset_b^x P'_1 \sqsubset_b^x \ldots \sqsubset_b^x P'_{n-2} \sqsubset_b^x P'_{n-1} = P$$

such that $P'_i \in \mathcal{F}(x)$ for $0 \leq i \leq n-1$. Then, in particular, $P'_{n-2} \in \mathcal{F}(x)$, a contradiction.

For, suppose that

$$P_i = \{V_1^i, V_2^i, \ldots, V_k^i, V_{k+1}^i, V_{k+2}^i, \ldots, V_{k+i}^i\}$$

for $0 \leq i \leq n-1$ and some $k \in \mathbb{Z}^+$, such that $V_j^i \subseteq S^x$ $(1 \leq j \leq k+i$ and $0 \leq i \leq n-1)$, and $V_j^{i+1} \subseteq V_j^i$ $(1 \leq j \leq k+i$ and $0 \leq i \leq n-2)$. For $0 \leq i \leq n-2$, $P_i \sqsubset_b^x P_{i+1}$ implies that P_i has been obtained from P_{i+1} by redistributing V_{k+i+1}^{i+1} over $V_1^{i+1}, V_2^{i+1}, \ldots, V_{k+i}^{i+1}$. Also, an easy induction establishes that $V_j^{n-1} \subseteq V_j^0$ for $1 \leq j \leq k$.

Define

$$P'_i = \{W_1^i, W_2^i, \ldots, W_k^i, W_{k+1}^i, W_{k+2}^i, \ldots, W_{k+i}^i\}$$

for $0 \leq i \leq n-1$, where:

$$\begin{array}{rcll}
W_j^i & = & V_j^i & \text{for } 1 \leq j \leq k \text{ and } i = 0, n-1 \\
W_j^i & = & W_j^{i+1} \cup \left(V_j^0 \cap V_{k+i+1}^{n-1}\right) & \text{for } 1 \leq j \leq k \text{ and } 1 \leq i \leq n-2 \\
W_j^i & = & V_j^{n-1} & \text{for } k+1 \leq j \leq k+i \text{ and } 1 \leq i \leq n-1.
\end{array}$$

We firstly show that $P'_i \in \mathcal{U}(x)$ for $0 \leq i \leq n-1$. We require to prove that, for each i $(0 \leq i \leq n-1)$, P'_i is a partition of S^x. Clearly $P'_0 = P_0 \in \mathcal{U}(x)$ and $P'_{n-1} = P_{n-1} \in \mathcal{U}(x)$. We argue by induction that $\cup_{j=1}^{k+n-1-i} W_j^{n-1-i} = S^x$ for $0 \leq i \leq n-2$. Clearly the base case $i = 0$ holds. Assume the result is true for some $r < n-2$; we show that it holds for $r+1$ also. Let $s = n-1-r$; then $2 \leq s \leq n-1$. We have

$$\begin{array}{rcl}
\left(\cup_{j=1}^{k+n-1-(r+1)} W_j^{n-1-(r+1)}\right) & = & \left(\cup_{j=1}^{k+s-1} W_j^{s-1}\right) \\
& = & \left(\cup_{j=1}^{k} \left(W_j^s \cup \left(V_j^0 \cap V_{k+s}^{n-1}\right)\right)\right) \cup \left(\cup_{j=k+1}^{k+s-1} W_j^{s-1}\right) \\
& = & \left(\cup_{j=1}^{k} W_j^s\right) \cup \left(\cup_{j=k+1}^{k+s-1} V_j^{n-1}\right) \cup \left(\cup_{j=1}^{k} \left(V_j^0 \cap V_{k+s}^{n-1}\right)\right)
\end{array}$$

$$\begin{aligned}
&= \left(\cup_{j=1}^{k} W_j^s\right) \cup \left(\cup_{j=k+1}^{k+s-1} V_j^{n-1}\right) \cup \left(V_{k+s}^{n-1} \cap \left(\cup_{j=1}^{k} V_j^0\right)\right) \\
&= \left(\cup_{j=1}^{k} W_j^s\right) \cup \left(\cup_{j=k+1}^{k+s-1} V_j^{n-1}\right) \cup V_{k+s}^{n-1} \quad (\text{since } P_0 \in \mathcal{U}(x)) \\
&= \left(\cup_{j=1}^{k+s} W_j^s\right) \\
&= \left(\cup_{j=1}^{k+n-1-r} W_j^{n-1-r}\right) \\
&= S^x \quad (\text{by induction hypothesis})
\end{aligned}$$

and hence the inductive step holds. Thus $P_i' \in \mathcal{U}(x)$ $(1 \leq i \leq n-2)$.

Now $W_j^{i+1} \subseteq W_j^i$ for $1 \leq j \leq k+i$ and $1 \leq i \leq n-2$. Hence $P_i' \sqsubset_b^x P_{i+1}'$ for $1 \leq i \leq n-2$. To show that $P_0' \sqsubset_b^x P_1'$, we argue by induction that, for $1 \leq i \leq n-2$ and $1 \leq j \leq k$,

$$W_j^{n-1-i} = V_j^{n-1} \cup \left(V_j^0 \cap \left(\cup_{j=k+n-i}^{k+n-1} V_j^{n-1}\right)\right).$$

Clearly the base case $i = 1$ holds. Assume the result is true for some $r < n-2$; we show that it holds for $r+1$ also. Let $s = n-1-r$; then $2 \leq s \leq n-2$. We have

$$\begin{aligned}
W_j^{n-1-(r+1)} &= W_j^{s-1} \\
&= W_j^s \cup \left(V_j^0 \cap V_{k+s}^{n-1}\right) \\
&= W_j^{n-1-r} \cup \left(V_j^0 \cap V_{k+n-1-r}^{n-1}\right) \\
&= V_j^{n-1} \cup \left(V_j^0 \cap \left(\cup_{j=k+n-r}^{k+n-1} V_j^{n-1}\right)\right) \cup \left(V_j^0 \cap V_{k+n-(r+1)}^{n-1}\right) \\
&\qquad (\text{by induction hypothesis}) \\
&= V_j^{n-1} \cup \left(V_j^0 \cap \left(\cup_{j=k+n-(r+1)}^{k+n-1} V_j^{n-1}\right)\right)
\end{aligned}$$

and hence the inductive step holds. Thus, for $1 \leq i \leq n-2$ and $1 \leq j \leq k$,

$$W_j^i = V_j^{n-1} \cup \left(V_j^0 \cap \left(\cup_{j=k+i+1}^{k+n-1} V_j^{n-1}\right)\right).$$

Hence, for $1 \leq i \leq n-2$ and $1 \leq j \leq k$, $W_j^i \subseteq (V_j^{n-1} \cup V_j^0)$. As $V_j^{n-1} \subseteq V_j^0$ for $1 \leq j \leq k$, then $W_j^i \subseteq V_j^0$ for $1 \leq j \leq k$ and $1 \leq i \leq n-2$. In particular $W_j^1 \subseteq V_j^0$ $(1 \leq j \leq k)$, so that $P_0' \sqsubset_b^x P_1'$.

Finally, we claim that $P_i' \in \mathcal{F}(x)$ for each i $(0 \leq i \leq n-1)$. For, $P' = P_0' \in \mathcal{F}(x)$ and $P = P_{n-1}' \in \mathcal{F}(x)$. For $1 \leq i \leq n-2$ and $1 \leq j \leq k$, $W_j^i \subseteq V_j^0$. But $P' = P_0 \in \mathcal{F}(x)$, so that $\pi_s(V_j^0)$ holds, which implies that $\pi_s(W_j^i)$ holds, by the partition-hereditary property, where π_s is the secondary predicate of π. For $1 \leq i \leq n-2$ and $k+1 \leq j \leq k+i$, $W_j^i = V_j^{n-1}$. But $P = P_{n-1} \in \mathcal{F}(x)$, so that $\pi_s(V_j^{n-1})$ holds, which implies that $\pi_s(W_j^i)$ holds. Hence $P_i' \in \mathcal{F}(x)$ for $1 \leq i \leq n-2$.

This completes the proof that $P_0', P_1', \ldots, P_{n-1}'$ satisfy the required conditions.

Proof of (2): Suppose that there is some $P' \in \mathcal{F}(x)$ such that $P \prec_b^x P'$. Then there exists some $P_i \in \mathcal{U}(x)$ $(0 \leq i \leq n-1)$ such that

$$P = P_0 \sqsubset_b^x P_1 \sqsubset_b^x \ldots \sqsubset_b^x P_{n-2} \sqsubset_b^x P_{n-1} = P'.$$

By hypothesis, $P_1 \notin \mathcal{F}(x)$, so $n \geq 3$. As in the proof of (1), we prove that there exists a chain

$$P = P'_0 \sqsubset^x_b P'_1 \sqsubset^x_b \ldots \sqsubset^x_b P'_{n-2} \sqsubset^x_b P'_{n-1} = P'$$

such that $P'_i \in \mathcal{F}(x)$ for $0 \leq i \leq n-1$. Then, in particular, $P'_1 \in \mathcal{F}(x)$, a contradiction.

For, suppose that

$$P_i = \{ V^i_1, V^i_2, \ldots, V^i_k, V^i_{k+1}, V^i_{k+2}, \ldots, V^i_{k+i}\}$$

for $0 \leq i \leq n-1$ and some $k \in \mathbb{Z}^+$, where the V^i_j are as in the proof of (1) $(1 \leq j \leq k+i)$. Define

$$P'_i = \{ W^i_1, W^i_2, \ldots, W^i_k, W^i_{k+1}, W^i_{k+2}, \ldots, W^i_{k+i}\}$$

for $0 \leq i \leq n-1$, where the W^i_j are as in the proof of (1) $(1 \leq j \leq k+i)$. Using the first induction of the previous proof, we similarly show that $P'_i \in \mathcal{U}(x)$ for $0 \leq i \leq n-1$. It similarly follows that $P'_i \sqsubset^x_b P'_{i+1}$ for $1 \leq i \leq n-2$. Using the second induction of the previous proof, we similarly show that, for $1 \leq i \leq n-2$ and $1 \leq j \leq k$,

$$W^i_j = V^{n-1}_j \cup \left(V^0_j \cap \left(\cup^{k+n-1}_{j=k+i+1} V^{n-1}_j \right) \right).$$

As in the proof of (1), it then follows that $P'_0 \sqsubset^x_b P'_1$.

Finally, we claim that $P'_i \in \mathcal{F}(x)$ for each i $(0 \leq i \leq n-1)$. For, $P = P'_0 \in \mathcal{F}(x)$ and $P' = P'_{n-1} \in \mathcal{F}(x)$. For $1 \leq i \leq n-2$ and $1 \leq j \leq k$, $V^{n-1}_j \subseteq W^i_j$. But $P' = P_{n-1} \in \mathcal{F}(x)$, so that $\pi_s(V^{n-1}_j)$ holds, which implies that $\pi_s(W^i_j)$ holds, by the partition-super-hereditary property. For $1 \leq i \leq n-2$ and $k+1 \leq j \leq k+i$, $W^i_j = V^{n-1}_j$. Again, $P' = P_{n-1} \in \mathcal{F}(x)$, so that $\pi_s(V^{n-1}_j)$ holds, which implies that $\pi_s(W^i_j)$ holds. Hence $P'_i \in \mathcal{F}(x)$ for $1 \leq i \leq n-2$.

This completes the proof that $P'_0, P'_1, \ldots, P'_{n-1}$ satisfy the required conditions. ■

A consequence of Theorem 9.5.9 is that the $\prec^x_b$-optimal solutions are identical to those that result from defining $\prec^x_b$ on $\mathcal{F}(x)$, for a given instance x, when the partition-hereditary or partition-super-hereditary property is satisfied by the optimisation problem in question.

9.5.3 Why we cannot always define the partial order on $\mathcal{U}(x)$

The results of Sections 9.5.1 and 9.5.2 have demonstrated that, for most of the optimisation problems and partial orders considered in this thesis, defining the partial order on $\mathcal{U}(x)$ produces a minimaximal or maximinimal optimisation problem with essentially the same behaviour as that which would result from defining the partial order on $\mathcal{F}(x)$. However this is not always the case, and in this section we demonstrate this by giving two examples.

The first example concerns the source MINIMUM NEARLY PERFECT SET problem (whose components are defined in Section 5.2), together with the partial order $\subset^G_1$, for a given graph G. Consider the graph $G = K_3$, with vertices u, v, w, as an instance of this problem. By defining $\subset^G_1$ on $\mathcal{U}(G)$, we can find vertex sets V_1, V_2, V_3, with $|V_r| = r$ for $1 \leq r \leq 3$, such that $V_1 \sqsubset^G_1 V_2 \sqsubset^G_1 V_3$. For example, we can choose $V_1 = \{u\}$, $V_2 = \{u, v\}$ and $V_3 = \{u, v, w\}$. Now $V_1, V_3 \in \mathcal{F}(G)$ but $V_2 \notin \mathcal{F}(G)$. However there is no $V'_2 \in \mathcal{F}(G)$

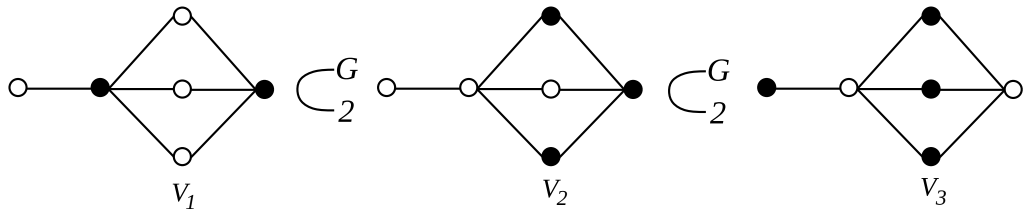

Figure 9.4: Example vertex sets related by the redefined partial order $\subset_2^G$.

such that $V_1 \sqsubset_1^G V_2' \sqsubset_1^G V_3$. Thus V_1 is $\subset_1^G$-maximal when $\subset_1^G$ is defined on $\mathcal{F}(G)$, but not $\subset_1^G$-maximal if $\subset_1^G$ is defined on $\mathcal{U}(G)$.

A second example is the source optimisation problem MAXIMUM INDEPENDENT SET (whose components are defined in Section 4.2.6), together with the partial order $\subset_2^G$, for a given graph G. By defining $\subset_2^G$ on $\mathcal{U}(G)$, we can find vertex sets V_1, V_2, V_3, with $|V_r| = r+1$ $(1 \leq r \leq 3)$, such that $V_1 \sqsubset_2^G V_2 \sqsubset_2^G V_3$. These sets are illustrated in Figure 9.4. It is clear that $V_1, V_3 \in \mathcal{F}(G)$ but $V_2 \notin \mathcal{F}(G)$. However there is no $V_2' \in \mathcal{F}(G)$ such that $V_1 \sqsubset_2^G V_2' \sqsubset_2^G V_3$. Thus V_1 is $\subset_2^G$-maximal when $\subset_2^G$ is defined on $\mathcal{F}(G)$, but not $\subset_2^G$-maximal if $\subset_2^G$ is defined on $\mathcal{U}(G)$.

Both of these examples provide optimisation problems and corresponding partial orders such that no result of the type proved in Sections 9.5.1 and 9.5.2 holds. This fact does not in itself prevent us from defining the partial order $\prec^x$ on $\mathcal{U}(x)$ in each case. However, as we have seen, the $\prec^x$-optimal solutions produced are different to those that are generated by defining $\prec^x$ on $\mathcal{F}(G)$. As a result, the behaviour of the associated minimaximal or maximinimal optimisation problem is inconsistent with that of the corresponding literature definition. This is not a desirable situation, and thus, in the general framework for minimaximal and maximinimal optimisation problems suggested by Definition 2.3.5, the partial order involved is to be defined on the feasible solutions, for a given instance.

9.6 Conclusions and open problems relating to minimaximal and maximinimal optimisation problems in general

Throughout this thesis, we have studied a number of examples of minimaximal and maximinimal optimisation problems from the point of view of algorithmic complexity. In doing so, we have presented an assortment of polynomial-time algorithms and NP-completeness results relating to these problems, though NP-completeness results feature in the main. It is reasonable to conclude from the examples in this thesis that the NP-hardness of a source optimisation problem Π almost certainly implies the NP-hardness of a minimaximal or maximinimal optimisation Π' derived from Π using a partial order. In fact, we have only found four examples of NP-hard source optimisation problems that admit polynomial-time solvable minimaximal or maximinimal versions. These are as follows:

1. MINIMUM DOMINATING SET and MAXIMUM MINIMAL DOMINATING SET in bipartite graphs (see Section 4.3.3).

2. MINIMUM DOMINATING SET and MAXIMUM MINIMAL DOMINATING SET in chordal graphs (see Section 4.3.3).

3. LONGEST PATH and SHORTEST $\overline{\lll}$ -MAXIMAL PATH (see Section 7.2.1).

4. SHORTEST COMMON SUPERSEQUENCE and LONGEST MINIMAL COMMON SUPERSEQUENCE when all inputs strings have length two (see Section 7.5.2).

Jacobson and Peters [131] were aware of Examples 1 and 2 above, and asked: in which other cases can we have a graph parameter ϕ whose value is hard to compute, together with a minimaximal or maximinimal counterpart of ϕ whose value is polynomial-time computable? Whilst we have not found any answers to this question, Examples 3 and 4 show that the more general concept of NP-hard source optimisation problems that admit polynomial-time solvable minimaximal or maximinimal versions prevails outwith the domain of graph theory. (Example 3 is perhaps not as significant as Example 4, since the partial order $\overline{\lll}^{x}$ is empty, when defined on the feasible solutions for a given instance x of LONGEST PATH.)

There are substantially more examples in this thesis of polynomial-time solvable source optimisation problems Π, together with an NP-hard minimaximal or maximinimal version derived from Π using an appropriate partial order. These are as follows:

- CHROMATIC NUMBER and ACHROMATIC NUMBER in trees and bipartite graphs (see Sections 3.1 and 3.2).

- CHROMATIC NUMBER and B-CHROMATIC NUMBER in bipartite graphs (see Section 3.5).

- MINIMUM VERTEX COVER and MAXIMUM MINIMAL VERTEX COVER (or equivalently, MAXIMUM INDEPENDENT SET and MINIMUM INDEPENDENT DOMINATING SET) in bipartite graphs (see Section 4.2.6).

- MINIMUM EDGE COVER and MAXIMUM MINIMAL EDGE COVER in arbitrary graphs (see Section 4.2.7).

- MAXIMUM MATCHING and MINIMUM MAXIMAL MATCHING in arbitrary graphs (see Section 4.2.8).

- MAXIMUM IRREDUNDANT SET and MINIMUM MAXIMAL IRREDUNDANT SET in bipartite and chordal graphs (see Section 4.3.6).

- MAXIMUM NEARLY PERFECT SET and MINIMUM 1-MAXIMAL NEARLY PERFECT SET in arbitrary graphs (see Section 5.2).

- MAXIMUM INDEPENDENT SET and MINIMUM 2-MAXIMAL INDEPENDENT SET in bipartite graphs (see Section 5.3.1).

- MAXIMUM MATCHING and MINIMUM 2-MAXIMAL MATCHING in arbitrary graphs (see Section 5.3.6).

It is likely that there are many other examples of polynomial-time solvable source optimisation problems that admit NP-hard minimaximal or maximinimal versions.

In addition to investigating the computational complexity of minimaximal and maximinimal optimisation problems, there are other issues relating to these questions that are worthy of further investigation.

For example, the question of the approximability of NP-hard minimaximal and maximinimal optimisation problems is of interest. At present, no features of minimaximal and maximinimal optimisation problems have been isolated that allow us to distinguish the question of their approximability from the question of the approximability of optimisation problems in general. However, it is possible that their structure does indeed allow this, and that an approximability-preserving reduction similar in spirit to the MM-reduction may be defined for these problems.

Another direction of research is to investigate instances x of a source optimisation problem Π for which all $\prec^x$-optimal solutions have the same measure, where $\prec^x$ is a partial order defined on the feasible solutions of Π. Graphs G for which all $\subset^G$-maximal independent sets are the same size are called *well-covered graphs*, and have received much attention (see Plummer [186] for a survey). Similarly, *well-dominated*, *well-irredundant* and *totally equimatchable* graphs have received attention [78, 207, 208]. These are graphs G for which all $\subset^G$-minimal dominating sets, all $\subset^G$-maximal irredundant sets, and all $\subset^G$-maximal total matchings are the same size, respectively. By restricting attention to such instances x of Π, we have that a minimaximal or maximinimal version of Π derived using $\prec^x$ is the same optimisation problem as Π.

Furthermore, given an optimisation problem Π and a partial order $\prec^x$ defined on the feasible solutions for a given instance x of Π, the question of whether there are efficient methods for counting and enumerating $\prec^x$-optimal solutions is of interest. For example, the problem of counting the number of maximal cliques in a graph is *#P-complete* [210] and therefore cannot be solved in polynomial time unless P=NP. However, the problem of enumerating all maximal cliques in a graph G is *P-enumerable* [210] and hence all maximal cliques can be listed in time $p(n)N$, where N is the number of maximal cliques in G and $p(n)$ is some polynomial in n, the size of G.

Finally, given an optimisation problem Π and a partial order $\prec^x$ defined on the feasible solutions for a given instance x of Π, there is the question of testing whether two feasible solutions are related by $\prec^x$. Clearly, this problem is trivial if $\prec^x$ is the partial order of set inclusion. However, the question becomes more interesting if, for instance, we consider partition-related optimisation problems, together with the partial order of partition redistribution.

9.7 Afterword

The study of minimaximal and maximinimal optimisation problems was motivated by the fact that both polynomial-time solvable source optimisation problems and NP-hard source optimisation problems can admit NP-hard minimaximal or maximinimal versions. However, as we indicated in the previous section, there are example pairs $\langle \Pi, \prec \rangle$ (where Π

is a source optimisation problem and $\prec$ is a partial order) in all four possible categories for the polynomial-time solvability or NP-hardness of Π and the minimaximal/maximinimal version of Π formulated using $\prec$.

Perhaps there are complexity classes that are dependent on the relationships between the algorithmic complexity of a source optimisation problem Π and a minimaximal or maximinimal version of Π. A notion of completeness in such a class may involve a reduction such as the MM-reduction of Chapter 8. However, any general 'explanation' of why some pairs $\langle \Pi, \prec \rangle$ behave differently from others seems improbable. It is likely that the question 'why does a source optimisation problem admit a hard minimaximal or maximinimal version?' is as difficult to answer as the question 'why is an optimisation problem hard to solve?'.

Glossary of symbols

Some frequently used symbols and abbreviations appearing in this thesis are listed in this glossary, together with a brief explanation of their meaning, and a page number where the symbol is first defined (if applicable).

Symbols relating to optimisation problems

Symbol	Brief explanation of symbol	Page
Π	Optimisation problem $\Pi = \langle \mathcal{I}, \mathcal{U}, \pi, m, \textsc{opt} \rangle$	21
$\mathcal{I}$	Instances of Π	21
$\|x\|$	Size of $x \in \mathcal{I}$	–
$max(x)$	Largest number occurring in $x \in \mathcal{I}$	–
$\mathcal{U}(x)$	Possible solutions of Π for $x \in \mathcal{I}$	21
$\mathcal{F}(x)$	Feasible solutions of Π for $x \in \mathcal{I}$	22
$\pi(x, y)$	Feasibility predicate for Π, where $x \in \mathcal{I}$, $y \in \mathcal{U}(x)$	21
$m(x, y)$	Measure function for Π, where $x \in \mathcal{I}$, $y \in \mathcal{F}(x)$	22
$\textsc{opt}$	Goal of Π, either max or min	22
$m(x, \mathcal{F}(x))$	Range of values that m takes for Π, where $x \in \mathcal{I}$	22
$m^*(x)$	Globally optimal measure function for Π, where $x \in \mathcal{I}$	22
$\mathcal{F}^*(x)$	Globally optimal solutions for Π, where $x \in \mathcal{I}$	22
$N_x(s)$	Neighbourhood relation for Π, where $x \in \mathcal{I}$, $s \in \mathcal{F}(x)$	15
POMM	Partial order measure monotonicity	24
Π_s	Search version of Π	23
Π_e	Evaluation version of Π	23
Π_d	Decision version of Π	23
NPO	Class of NP Optimisation problems	23
PLS	Class of Polynomial Local Search problems	18
$R_A(x)$	Performance ratio of A with respect to x	4
ptas	Polynomial-time approximation scheme	4
$\Pi_1 \alpha_T \Pi_2$	Π_1 is Turing-reducible to Π_2	3
$\langle \Pi_1, \prec_1 \rangle \alpha_{MM} \langle \Pi_2, \prec_2 \rangle$	Pair $\langle \Pi_1, \prec_1 \rangle$ is MM-reducible to pair $\langle \Pi_2, \prec_2 \rangle$	155

Symbols relating to partial orders

Symbol	Brief explanation of symbol	Page
$\prec^x$	Partial order defined on $\mathcal{F}(x)$, satisfying POMM w.r.t. Π, where $x \in \mathcal{I}$	23
$(\sqsubset^x)^*$	Transitive closure of relation $\sqsubset^x$	–
$\subset^x$	Partial order of set inclusion	26
$\subset_k^x = (\sqsubset_k^x)^*$	Partial order of $(k-1,k)$-replacement	26
$\lll^x$	Subsequence partial order	28
$\overline{\lll}^{\,x}$	Substring partial order	28
$\prec_a^x = (\sqsubset_a^x)^*$	Partial order of partition merge	28
$\prec_b^x = (\sqsubset_b^x)^*$	Partial order of partition redistribution	28
$\prec_{a,k}^x = (\sqsubset_{a,k}^x)^*$	Partial order of partition $(k-1,k)$-merge $(k \geq 2)$	29
$\prec_{b,k}^x = (\sqsubset_{b,k}^x)^*$	Partial order of partition k-redistribution $(k \geq 1)$	29
$\prec_f^x$	Partial order on functions	102
$\prec_t^x$	Partial order on truth assignments	144

Symbols relating to graph theory

Symbol	Brief explanation of symbol	Page
G^C	Complementary graph of G	–
$L(G)$	Line graph of G	79
$T(G)$	Total graph of G	57
$V(G)$	Vertices of $G=(V,E)$, i.e., V	–
$E(G)$	Edges of $G=(V,E)$, i.e., E	–
K_n	Complete graph with n vertices	–
P_n	Path with n vertices	–
$d(v)$	Degree of a vertex v in G	–
$\langle S \rangle$	Subgraph of G induced by $S \subseteq V$	–
$d(u,v)$	Distance (number of edges) separating $u, v \in V$ in G	–
$N(v)$	Open neighbourhood of vertex $v \in V$	11
$N[v]$	Closed neighbourhood of vertex $v \in V$	11
$N(S)$	Open neighbourhood of vertices $S \subseteq V$	11
$N[S]$	Closed neighbourhood of vertices $S \subseteq V$	11
$H(v)$	Set of all edges of G incident on $v \in V$	103
$c(v)$	Colour of a vertex $v \in V$ in some colouring of G	–
$m(G)$	m-degree of graph G	38
$p_2(G)$	Minimum maximal 2-packing number of G	108
$\rho(G)$ or $P_2(G)$	Maximum 2-packing number of G	108
$p_f(G)$	Minimum maximal fractional packing number of G	107
$\rho_f(G)$	Maximum fractional packing number of G	104

For other graph parameter symbols used in this thesis, see Table 5.2 on Page 96. Graph parameter symbols with subscript 'f' (apart from p_f and ρ_f) appearing in this thesis denote fractional versions of the graph parameters of Table 5.2, and are defined in Chapter 6.

Symbols relating to strings

Symbol	Brief explanation of symbol	Page
Σ	Alphabet	–
$\langle s_1 \ldots s_r \rangle$	String with symbol $s_i \in \Sigma$ at ith position $(1 \leq i \leq r)$	–
$\lvert s \rvert$	Length of string s	13
$s +\!\!+\, t$	String s concatenated with string t	–
Σ^*	Set of all strings composed of symbols of Σ	13
$s \underline{\ll} t$	String s is a subsequence of string t	13
$s \overline{\underline{\ll}} t$	String s is a substring of string t	13
$s \ll t$	String s is a proper subsequence of string t	13
$s \overline{\ll} t$	String s is a proper substring of string t	13
$s \underline{\ll} S$ $(S \underline{\ll} s)$	String s is a common sub(super)sequence of set of strings S	14
$s \overline{\underline{\ll}} S$ $(S \overline{\underline{\ll}} s)$	String s is a common sub(super)string of set of strings S	14
$\mathbf{seq}_r S$	Set of all strings of length r composed of symbols from S	–

Symbols relating to logic

Symbol	Brief explanation of symbol	Page
false, true	Boolean truth values (sometimes abbreviated F, T)	–
$\wedge$	Logical conjunction operator	–
$\vee$	Logical disjunction operator	–
$\Leftrightarrow$	Logical equivalence operator	–
$\Rightarrow$	Logical implication operator	–
U	Set of variables	–
σ	Literal	143
$\overline{\sigma}$	Negation of literal σ	143
$\mathcal{C}_f$	Set of clauses from $\mathcal{C}$ satisfied by truth assigment f	144
$U(C_i)$	Set of variables in clause C_i	144
$U(C')$	Set of variables in collection of clauses C'	144

Miscellaneous symbols

Symbol	Brief explanation of symbol	Page
$\varnothing$	Empty set	–
$\mathbb{P}(S)$	Power set of S, i.e., set of all subsets of S	–
$\mathbb{Z}$	Set of all integers	–
$\mathbb{N}$	Set of all natural numbers (non-negative integers)	–
$\mathbb{Z}^+$	Set of all positive integers	–
$\mathbb{Q}$	Set of all rational numbers	–
$\mathbb{R}$	Set of all real numbers	–
$[a..b]$	$\{n \in \mathbb{Z} : a \leq n \leq b\}$, for $a, b \in \mathbb{Z}$, $a \leq b$	–
$\mathcal{P}^+$	Maximal elements of collection $\mathcal{P}$	55
$\mathcal{P}^-$	Minimal elements of collection $\mathcal{P}$	55

Bibliography

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] Y. Alavi, M. Behzad, L.M. Lesniak-Foster, and E.A. Nordhaus. Total matchings and total coverings of graphs. *Journal of Graph Theory*, 1:135–140, 1977.

[3] Y. Alavi, M. Behzad, and E.A. Nordhaus. Minimal separating sets of maximum size. *Journal of Combinatorial Theory, Series B*, 28:180–183, 1980.

[4] Y. Alavi, J. Liu, J. Wang, and Z. Zhang. On total covers of graphs. *Discrete Mathematics*, 100:229–233, 1992.

[5] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.

[6] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23:11–24, 1989.

[7] S. Arora and C. Lund. Hardness of approximations. In D.S. Hochbaum, editor, *Approximation Algorithms for NP-hard problems*, chapter 10, pages 399–446. PWS Publishing Company, 1997.

[8] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the Association for Computing Machinery*, 45(3):501–555, 1998.

[9] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the Association for Computing Machinery*, 45(1):70–122, 1998.

[10] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solution of NP optimization problems. *Theoretical Computer Science*, 150:1–55, 1995.

[11] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimisation problems. *Journal of Computer and System Sciences*, 21:136–153, 1980.

[12] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41(1):153–180, 1994.

[13] D.W. Bange, A.E. Barkauskas, and P.J. Slater. Efficient dominating sets in graphs. In R.D. Ringeisen and F.S. Roberts, editors, *Applications of Discrete Mathematics*, pages 189–199. SIAM, Philadelphia, 1988.

[14] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and nonapproximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[15] C. Berge. *The Theory of Graphs and its Applications*. Methuen, 1962.

[16] C. Berge. Balanced matrices. *Mathematical Programming*, 2:19–31, 1972.

[17] C. Berge. *Graphs and Hypergraphs*. North-Holland, 1973.

[18] M.W. Bern, E.L. Lawler, and A.L. Wong. Linear-time computation of optimal subgraphs of decomposable graphs. *Journal of Algorithms*, 8(2):216–235, 1987.

[19] A.A. Bertossi. Dominating sets for split and bipartite graphs. *Information Processing Letters*, 19:37–40, 1984.

[20] T. Beyer, A. Proskurowski, S.T. Hedetniemi, and S. Mitchell. Independent domination in trees. In *Proceedings of the 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 321–328. Utilitas Mathematica, 1977.

[21] H.L. Bodlaender. Achromatic number is NP-complete for cographs and interval graphs. *Information Processing Letters*, 31:135–138, 1989.

[22] B. Bollobás, E.J. Cockayne, and C.M. Mynhardt. On generalised minimal domination parameters for paths. *Discrete Mathematics*, 86:89–97, 1990.

[23] K.S. Booth and J.H. Johnson. Dominating sets in chordal graphs. *SIAM Journal on Computing*, 11(1):191–199, 1982.

[24] D.P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall, 1994.

[25] A. Brandstädt, V.D. Chepoi, and F.F. Dragan. The algorithmic use of hypertree structure and maximum neighbourhood orderings. *Discrete Applied Mathematics*, 82(1-3):43–77, 1998.

[26] A. Brandstädt and D. Kratsch. On the restriction of some NP-complete graph problems to permutation graphs. In *Proceedings of the International Conference on Fundamentals of Computation Theory*, volume 199 of *Lecture Notes in Computer Science*, pages 53–62. Springer-Verlag, 1985.

[27] A. Brandstädt and D. Kratsch. On domination problems for permutation and other graphs. *Theoretical Computer Science*, 54:181–198, 1987.

[28] H.J. Broersma, T. Kloks, D. Kratsch, and H. Müller. Independent sets in asteroidal triple-free graphs. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, volume 1256 of *Lecture Notes in Computer Science*, pages 760–770. Springer-Verlag, 1997.

[29] D. Bruschi, D. Joseph, and P. Young. A structural overview of NP optimization problems. In *Proceedings of the 2nd International Symposium on Optimal Algorithms*, volume 401 of *Lecture Notes in Computer Science*, pages 205–231. Springer-Verlag, 1989.

[30] N. Cairnie and K. Edwards. Some results on the achromatic number. *Journal of Graph Theory*, 26:129–136, 1997.

[31] G.J. Chang. Labelling algorithms for domination problems in sun-free chordal graphs. *Discrete Applied Mathematics*, 22:21–34, 1988/89.

[32] G.J. Chang, M. Farber, and Z. Tuza. Algorithmic aspects of neighborhood numbers. *SIAM Journal on Discrete Mathematics*, 6(1):24–29, 1993.

[33] G.J. Chang and G.L. Nemhauser. The k-domination and k-stability problems on sun-free chordal graphs. *SIAM Journal on Algebraic and Discrete Methods*, 5(3):332–345, 1984.

[34] M.S. Chang. Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal on Computing*, 27(6):1671–1694, 1998.

[35] G. Chartrand and L. Lesniak. *Graphs and Digraphs*. Wadsworth, Monterey, CA, 1986.

[36] G. Chartrand and S. Schuster. On the independence numbers of complementary graphs. *Transactions of the New York Academy of Science (2)*, 36:247–251, 1974.

[37] A. Chaudhary and S. Vishwanathan. Approximation algorithms for the achromatic number. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*, pages 557–562. ACM-SIAM, 1997.

[38] G.A. Cheston and G. Fricke. Classes of graphs for which upper fractional domination equals independence, upper domination, and upper irredundance. *Discrete Applied Mathematics*, 55:241–258, 1994.

[39] G.A. Cheston, G. Fricke, S.T. Hedetniemi, and D.P. Jacobs. On the computational complexity of upper fractional domination. *Discrete Applied Mathematics*, 27:195–207, 1990.

[40] C.A. Christen and S.M. Selkow. Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B*, 27:49–59, 1979.

[41] E.J. Cockayne, R.M. Dawes, and S.T. Hedetniemi. Total domination in graphs. *Networks*, 10:211–219, 1980.

[42] E.J. Cockayne, O. Favaron, H. Li, and G. MacGillivray. The product of the independent domination numbers of a graph and its complement. *Discrete Mathematics*, 90:313–317, 1991.

[43] E.J. Cockayne, O. Favaron, and C.M. Mynhardt. Universal maximal packing functions of graphs. *Discrete Mathematics*, 159:57–68, 1996.

[44] E.J. Cockayne, O. Favaron, C. Payan, and A.G. Thomason. Contributions to the theory of domination, independence and irredundance in graphs. *Discrete Mathematics*, 33:249–258, 1981.

[45] E.J. Cockayne, G. Fricke, and C.M. Mynhardt. On a Nordhaus-Gaddum type problem for independent domination. *Discrete Mathematics*, 138:199–205, 1995.

[46] E.J. Cockayne, S.E. Goodman, and S.T. Hedetniemi. A linear algorithm for the domination number of a tree. *Information Processing Letters*, 4:41–44, 1975.

[47] E.J. Cockayne and S.T. Hedetniemi. Independence graphs. In *Proceedings of the 5th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 471–491. Utilitas Mathematica, 1974.

[48] E.J. Cockayne, S.T. Hedetniemi, and R. Laskar. Gallai theorems for graphs, hypergraphs and set systems. *Discrete Mathematics*, 72:35–47, 1988.

[49] E.J. Cockayne, S.T. Hedetniemi, and D.J. Miller. Properties of hereditary hypergraphs and middle graphs. *Canadian Mathematical Bulletin*, 21:461–468, 1978.

[50] E.J. Cockayne, G. MacGillivray, and C.M. Mynhardt. Generalised maximal independence parameters for paths and cycles. *Quaestiones Mathematicae*, 13:123–139, 1990.

[51] E.J. Cockayne and C.M. Mynhardt. On the product of upper irredundance numbers of a graph and its complement. *Discrete Mathematics*, 117-121:76, 1989.

[52] E.J. Cockayne and C.M. Mynhardt. On the product of k-minimal domination numbers of a graph and its complement. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 8:118–122, 1990.

[53] E.J. Cockayne and C.M. Mynhardt. Domination sequences of graphs. *Ars Combinatoria*, 33:257–275, 1992.

[54] E.J. Cockayne and C.M. Mynhardt. The sequence of upper and lower domination, independence and irredundance numbers of a graph. *Discrete Mathematics*, 122:89–102, 1993.

[55] S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, New York, 1971. ACM.

[56] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. McGraw-Hill/MIT, 1990.

[57] D.G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, 1984.

[58] D.G. Corneil and L.K. Stewart. Dominating sets in perfect graphs. *Discrete Mathematics*, 86:145–164, 1990.

[59] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Technical Report SI/RR-95/02, Università di Roma "La Sapienza", Dipartimento di Scienze dell'Informazione, 1995. Lastest version available at `http://www.nada.kth.se/theory/problemlist.html`.

[60] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93:241–262, 1991.

[61] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.

[62] D.E. Daykin and C.P. Ng. Algorithms for generalized stability numbers of tree graphs. *Journal of the Australasian Mathematical Society*, 6:89–100, 1966.

[63] G.S. Domke, S. Hedetniemi, R. Laskar, and R. Allan. Generalized packings and coverings of graphs. *Congressus Numerantium*, 62:259–270, 1988.

[64] G.S. Domke, S.T. Hedetniemi, and R.C. Laskar. Fractional packings, coverings and irredundance in graphs. *Congressus Numerantium*, 66:227–238, 1988.

[65] G.S. Domke, S.T. Hedetniemi, R.C. Laskar, and G. Fricke. Relationships between integer and fractional parameters of graphs. In Y. Alavi, G. Chartrand, O.R. Oellerman, and A.J. Schwenk, editors, *Graph Theory, Combinatorics and Applications*, pages 371–387. John Wiley and Sons, 1991.

[66] Z. Drezner. On minimax optimization problems. *Mathematical Programming*, 22:227–230, 1982.

[67] J. Dunbar, W. Goddard, S. Hedetniemi, A. McRae, and M. Henning. The algorithmic complexity of minus domination in graphs. *Discrete Applied Mathematics*, 68:73–84, 1996.

[68] J.E. Dunbar, F.C. Harris, Jr, S.M. Hedetniemi, S.T. Hedetniemi, A.A. McRae, and R.C. Laskar. Nearly perfect sets in graphs. *Discrete Mathematics*, 138:229–246, 1995.

[69] M.E. Dyer and A.M. Frieze. Planar 3DM is NP-complete. *Journal of Algorithms*, 7:174–184, 1986.

[70] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[71] E.S. Elmallah and L.K. Stewart. Domination in polygon graphs. *Congressus Numerantium*, 77:63–76, 1990.

[72] P. Erdös and A. Meir. On total matching numbers and total covering numbers of complementary graphs. *Discrete Mathematics*, 19:229–233, 1977.

[73] M. Farber. Independent domination in chordal graphs. *Operations Research Letters*, 1:134–138, 1982.

[74] M. Farber. Domination, independent domination and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7(2):115–130, 1984.

[75] M. Farber, G. Hahn, P. Hell, and D. Miller. Concerning the achromatic number of a graph. *Journal of Combinatorial Theory, Series B*, 40:21–39, 1986.

[76] M. Farber and J.M. Keil. Domination in permutation graphs. *Journal of Algorithms*, 6(3):309–321, 1985.

[77] M. Fellows, G. Fricke, S. Hedetniemi, and D. Jacobs. The private neighbor cube. *SIAM Journal on Discrete Mathematics*, 7(1):41–47, 1994.

[78] A. Finbow, B. Hartnell, and R. Nowakowski. Well-dominated graphs: a collection of well-covered ones. *Ars Combinatoria*, 25A:5–10, 1988.

[79] S. Fischer and L. Torenvliet. The malleability of TSP_{2opt}. In *Proceedings of the 21st International Workshop on Graph-Theoretic Concepts in Computer Science, Aachen, Germany*, volume 1017 of *Lecture Notes in Computer Science*, pages 152–166. Springer-Verlag, 1995.

[80] C.B. Fraser. *Subsequences and Supersequences of Strings*. PhD thesis, University of Glasgow, Department of Computing Science, 1995.

[81] C.B. Fraser, R.W. Irving, and M. Middendorf. Maximal common subsequences and minimal common supersequences. *Information and Computation*, 124:145–153, 1996.

[82] G. Fricke. On the equivalence of the upper irredundance and fractional upper irredundance numbers of a graph. Manuscript, 1988.

[83] G. Fricke, E. Hare, D. Jacobs, and A. Majumdar. On integral and fractional total domination. *Congressus Numerantium*, 77:87–95, 1990.

[84] G.H. Fricke, S.T. Hedetniemi, and D.P. Jacobs. Maximal irredundant functions. *Discrete Applied Mathematics*, 68:267–277, 1996.

[85] D.R. Fulkerson, A.J. Hoffman, and R. Oppenheim. On balanced matrices. *Mathematical Programming Study*, 1:120–132, 1974.

[86] M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *Journal of Algorithms*, 17:409–423, 1994.

[87] T. Gallai. Über extreme Punkt-und Kantenmengen. *Ann. Univ. Sci. Budapest, Eötvös Sect. Math.*, 2:133–138, 1959.

[88] J. Gallant, D. Maier, and J.A. Storer. On finding minimal length superstrings. *Journal of Computer and System Sciences*, 20(1):50–58, 1980.

[89] M.R. Garey and D.S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[90] M.R. Garey and D.S. Johnson. Unpublished result, 1978.

[91] M.R. Garey and D.S. Johnson. "Strong" NP-completeness results: motivation, examples, and implications. *Journal of the Association for Computing Machinery*, 25:499–508, 1978.

[92] M.R. Garey and D.S. Johnson. *Computers and Intractability.* Freeman, San Francisco, CA., 1979.

[93] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[94] R.S. Garfinkel and G.L. Nemhauser. *Integer Programming.* John Wiley and Sons, New York, 1972.

[95] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1:180–187, 1972.

[96] A. Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1985.

[97] J. Gimbel, M. Mahéo, and C. Virlouvet. Double total domination of graphs. *Discrete Mathematics*, 165/166:343–351, 1997.

[98] J. Gimbel and P.D. Vestergaard. Inequalities for total matchings of graphs. *Ars Combinatoria*, 39:109–119, 1995.

[99] M.C. Golumbic. *Algorithmic graph theory and perfect graphs.* Academic Press, 1980.

[100] D.L. Grinstead and P.J. Slater. Fractional domination and fractional packing in graphs. *Congressus Numerantium*, 71:153–172, 1990.

[101] D.L. Grinstead and P.J. Slater. A recurrence template for several parameters in series-parallel graphs. *Discrete Applied Mathematics*, 54:151–168, 1994.

[102] R.P. Gupta. Independence and covering numbers of line graphs and total graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 61–62. Academic Press, 1969.

[103] D. Gusfield. *Algorithms on strings, trees and sequences.* Cambridge University Press, 1997.

[104] M.M. Halldórsson. Approximating the minimum maximal independence number. *Information Processing Letters*, 46:169–172, 1993.

[105] M.M. Halldórsson. Approximating discrete collections via local improvements. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco*, pages 160–169. ACM-SIAM, 1995.

[106] F. Harary. *Graph Theory.* Addison-Wesley, 1969.

[107] F. Harary. Maximum versus minimum invariants for graphs. *Journal of Graph Theory*, 7:275–284, 1983.

[108] F. Harary and T.W. Haynes. Nordhaus-Gaddum inequalities for domination in graphs. *Discrete Mathematics*, 155:99–105, 1996.

[109] F. Harary and S. Hedetniemi. The achromatic number of a graph. *Journal of Combinatorial Theory*, 8:154–161, 1970.

[110] F. Harary, S. Hedetniemi, and G. Prins. An interpolation theorem for graphical homomorphisms. *Portugaliae Mathematica*, 26:453–462, 1967.

[111] W.R. Hare, R. Laskar, and K. Peters. Max-min parameters and disconnected colorings. *Congressus Numerantium*, 44:139–150, 1984.

[112] J.H. Hattingh, M.A. Henning, and P.J. Slater. The algorithmic complexity of signed domination in graphs. *Australasian Journal of Combinatorics*, 12:101–112, 1995.

[113] J. Haviland. Independent domination in regular graphs. *Discrete Mathematics*, 143:275–280, 1995.

[114] T. Haynes, S.T. Hedetniemi, and P.J. Slater, editors. *Domination in Graphs: Advanced Topics.* Marcel Dekker, 1998.

[115] T. Haynes, S.T. Hedetniemi, and P.J. Slater. *Fundamentals of Domination in Graphs.* Marcel Dekker, 1998.

[116] S.M Hedetniemi, S.T. Hedetniemi, and D.P. Jacobs. Private domination - theory and algorithms. *Congressus Numerantium*, 79:147–157, 1990.

[117] S.M. Hedetniemi, S.T. Hedetniemi, R. Laskar, A. McRae, and A. Majumdar. Domination, independence and irredundance in total graphs: a brief survey. In Y. Alavi and A. Schwenk, editors, *Graph Theory, Combinatorics and Applications: Proceedings of the Seventh Quadrennial International Conference on the Theory and Applications of Graphs*, volume 2, pages 671–683. John Wiley and Sons, Inc., 1995.

[118] S.M. Hedetniemi, S.T. Hedetniemi, and T. Wimer. Linear time resource allocation algorithms for trees. Technical Report URI-014, Clemson University, Department of Mathematical Sciences, South Carolina, 1987.

[119] S.T. Hedetniemi. A max-min relationship between matchings and domination in graphs. *Congressus Numerantium*, 40:23–34, 1983.

[120] S.T. Hedetniemi. Personal communication, 1997.

[121] S.T. Hedetniemi and R.C. Laskar. Connected domination in graphs. In B. Bollobás, editor, *Graph Theory and Combinatorics*, chapter 18, pages 209–217. Academic Press, 1984.

[122] S.T. Hedetniemi and R.C. Laskar, editors. *Topics on Domination*, volume 86 of *Discrete Mathematics*, 1990.

[123] S.T. Hedetniemi, R.C. Laskar, and J. Pfaff. Irredundance in graphs – a survey. *Congressus Numerantium*, 48:183–193, 1985.

[124] D.S. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[125] A.J. Hoffman and J.B. Kruskal. Integral boundary points of convex polyhedra. In H.W. Kuhn and A.W. Tucker, editors, *Linear inequalities and related systems*, volume 38 of *Annals of Mathematical Studies*, pages 223–246. Princeton University Press, 1956.

[126] J.D. Horton and K. Kilakos. Minimum edge dominating sets. *SIAM Journal on Discrete Mathematics*, 6:375–387, 1993.

[127] F. Hughes and G. MacGillivray. The achromatic number of graphs: a survey and some new results. *Bulletin of the Institute of Combinatorics and its Applications*, 19:27–56, 1997.

[128] M. Hujter and Zs. Tuza. Precoloring extension. II. Graph classes related to bipartite graphs. *Acta Mathematica Universitatis Comenianae*, 62:1–11, 1993.

[129] R.W. Irving. On approximating the minimum independent dominating set. *Information Processing Letters*, 37:197–200, 1991.

[130] R.W. Irving and C.B. Fraser. Two algorithms for the longest common subsequence of three (or more) strings. In *Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching*, volume 644 of *Lecture Notes in Computer Science*, pages 214–229. Springer-Verlag, 1992.

[131] M.S. Jacobson and K. Peters. Chordal graphs and upper irredundance, upper domination and independence. *Discrete Mathematics*, 86:59–69, 1990.

[132] T.R. Jensen and B. Toft. *Graph Colouring Problems*. Wiley, 1995.

[133] D.S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 5:147–160, 1984.

[134] D.S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 6:434–451, 1985.

[135] D.S. Johnson. Local optimization and the traveling salesman problem. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 446–461. Springer-Verlag, 1990.

[136] D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.

[137] P.P Kale and N.V. Deshpande. On line independence, domination, irredundance and neighbourhood numbers of a graph. *Indian Journal of Pure and Applied Mathematics*, 21(8):695–698, 1990.

[138] V. Kann and A. Panconesi. Hardness of approximation. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annoted Bibliographies in Combinatorial Optimization*, chapter 2, pages 13–30. John Wiley and Sons, 1997.

[139] N. Karmarkar. A new polynomial-time algorithm in linear programming. *Combinatorica*, 4(4):373–395, 1984.

[140] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[141] J.M. Keil. The complexity of domination problems in circle graphs. *Discrete Applied Mathematics*, 42(1):51–63, 1993.

[142] B.W. Kernighan and S. Lin. An efficient heuristic for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[143] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[144] T. Kikuno, N. Yoshida, and Y. Kakuda. The NP-completeness of the dominating set problem in cubic planar graphs. *Transactions of the Institute of Electronics and Communication Engineers of Japan*, E63(6):443–444, 1980.

[145] B. Korte and D. Hausmann. An analysis of the greedy heuristic for independence systems. In *Annals of Discrete Mathematics*, volume 2, pages 65–74. North-Holland, 1978.

[146] D. Kratsch. Domination and total domination in asteroidal triple-free graphs. To appear in *Discrete Applied Mathematics*.

[147] D. Kratsch and L. Stewart. Domination on cocomparability graphs. *SIAM Journal on Discrete Mathematics*, 6:400–417, 1993.

[148] D. Kratsch and L. Stewart. Total domination and transformation. *Information Processing Letters*, 63:167–170, 1997.

[149] M.W. Krentel. Structure in locally optimal solutions. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 216–221. IEEE Computer Society, 1989.

[150] V.R. Kulli, S.C. Sigarkanti, and N.D. Soner. Entire domination in graphs. In V.R. Kulli, editor, *Advances in Graph Theory*, pages 237–243. Vishwa International Publications, 1991.

[151] R.C. Laskar and J. Pfaff. Domination and irredundance in split graphs. Technical Report 430, Clemson University, Department of Mathematical Sciences, South Carolina, 1983.

[152] R.C. Laskar, J. Pfaff, S.M. Hedetniemi, and S.T. Hedetniemi. On the algorithmic complexity of total domination. *SIAM Journal on Algebraic and Discrete Methods*, 5(3):420–425, 1984.

[153] L.A. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973 (in Russian). English translation in *Problems of Information Transmission* 9:265-266, 1973.

[154] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(2):2245–2269, 1965.

[155] S. Lin and B. Kernighan. An effective heuristic for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

[156] C.L. Liu. *Introduction to Combinatorial Mathematics*. McGraw-Hill, 1968.

[157] L. Lovász and M.D. Plummer. *Matching Theory*. Number 29 in Annals of Discrete Mathematics. North-Holland, 1986.

[158] H-I Lu and R. Ravi. The power of local optimization: approximation algorithms for maximum-leaf spanning tree. In *Proceedings of the 30th Annual Allerton Conference on Communication, Control and Computing*, pages 533–542. University of Illinois at Urbana-Champaign, October 1992.

[159] H-I Lu and R. Ravi. A near-linear-time approximation algorithm for maximum-leaf spanning tree. Technical Report CS-96-06, Brown University, Department of Computing Science, 1996.

[160] G. Lueker. Manuscript, Princeton University, 1976.

[161] D. Maier. The complexity of some problems on subsequences and supersequences. *Journal of the Association for Computing Machinery*, 25:322–336, 1978.

[162] D. Maier and J.A. Storer. A note on the complexity of the superstring problem. Technical Report 233, Princeton University, Department of Electrical Engineering and Computer Science, Princeton, NJ, October 1977.

[163] A. Majumdar. *Neighborhood hypergraphs: a framework for covering and packing parameters in a graph.* PhD thesis, Clemson University, Department of Mathematical Sciences, South Carolina, 1992.

[164] J.D. McFall and R. Nowakowski. Strong independence in graphs. *Congressus Numerantium*, 29:639–656, 1980.

[165] A.A. McRae. *Generalizing NP-completeness proofs for bipartite and chordal graphs.* PhD thesis, Clemson University, Department of Computer Science, South Carolina, 1994.

[166] A. Meir. On total covering and matching of graphs. *Journal of Combinatorial Theory, Series B*, 24:164–168, 1978.

[167] A. Meir and J.W. Moon. Relations between packing and covering numbers of a tree. *Pacific Journal of Mathematics*, 61(1):225–233, 1975.

[168] S. Mitchell and S.T. Hedetniemi. Edge domination in trees. In *Proceedings of the 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 489–509. Utilitas Mathematica, 1977.

[169] C.M. Mynhardt. Generalised maximal independence and clique numbers of graphs. *Quaestiones Mathematicae*, 11:383–398, 1988.

[170] C.M. Mynhardt and E.J. Cockayne. k-minimal domination numbers of cycles. *Ars Combinatoria*, 23A:195–206, 1987.

[171] J. Nieminen. Two bounds for the domination number of a graph. *Journal of the Institute of Mathematics and its Applications*, 14:183–187, 1974.

[172] E.A. Nordhaus. Generalizations of graphical parameters. In *Proceedings of the International Conference on the Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 420–425, 1976.

[173] E.A. Nordhaus and J.W. Gaddum. On complementary graphs. *American Mathematical Monthly*, 63:175–177, 1956.

[174] R.Z. Norman and M.O. Rabin. An algorithm for a minimum cover of a graph. *Proceedings of the American Mathematical Society*, 10:315–319, 1959.

[175] O. Ore. *Theory of Graphs*, volume 38 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Rhode Island, 1962.

[176] P. Orponen and H. Mannila. On approximation preserving reductions: complete problems and robust measures. Technical Report C-1987-28, University of Helsinki, Department of Computer Science, 1987.

[177] C.H. Papadimitriou. The complexity of the Lin-Kernighan heuristic for the traveling salesman problem. *SIAM Journal on Computing*, 21(3):450–465, 1992.

[178] C.H. Papadimitriou and K. Steiglitz. On the complexity of local search for the traveling salesman problem. *SIAM Journal on Computing*, 6(1):76–83, 1977.

[179] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, New Jersey, 1982.

[180] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[181] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15:251–277, 1981.

[182] U. Peled and F. Sun. Total matchings and total coverings of threshold graphs. *Discrete Applied Mathematics*, 49:325–330, 1994.

[183] K. Peters, R. Laskar, and S. Hedetniemi. Maximal/minimal connectivity in graphs. *Ars Combinatoria*, 21:59–70, 1986.

[184] J. Pfaff, R. Laskar, and S.T. Hedetniemi. Linear algorithms for independent domination and total domination in series-parallel graphs. *Congressus Numerantium*, 45:71–82, 1984.

[185] J. Pfaff, R.C. Laskar, and S.T. Hedetniemi. NP-completeness of total and connected domination, and irredundance for bipartite graphs. Technical Report 428, Clemson University, Department of Mathematical Sciences, South Carolina, 1983.

[186] M.D. Plummer. Well covered graphs: a survey. *Quaestiones Mathematicae*, 16:253–287, 1993.

[187] W.R. Pulleyblank. Fractional matchings and the Edmonds-Gallai theorem. *Discrete Applied Mathematics*, 16:51–58, 1987.

[188] K.-J. Räihä and E. Ukkonen. The shortest common superseqence problem over a binary alphabet is NP-complete. *Theoretical Computer Science*, 16:187–198, 1981.

[189] V. Raman, B. Ravikumar, and S. Srinivasa Rao. A simplified NP-complete MAXSAT problem. *Information Processing Letters*, 65:1–6, 1998.

[190] E. Sampathkumar and P.S. Neeralagi. The neighbourhood number of a graph. *Indian Journal of Pure and Applied Mathematics*, 16(2):126–132, 1985.

[191] E. Sampathkumar and P.S. Neeralagi. The line neighbourhood number of a graph. *Indian Journal of Pure and Applied Mathematics*, 17(2):142–149, 1986.

[192] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 216–226, New York, 1978. ACM.

[193] A. Schäffer and M. Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

[194] E.R. Scheinerman and D.H. Ullman. *Fractional Graph Theory.* John Wiley and Sons, 1997.

[195] D. Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41:579–585, 1994.

[196] P.J. Slater. Enclaveless sets and MK-systems. *Journal of Research of the National Bureau of Standards*, 82(3):197–202, 1977.

[197] P.J. Slater. Generalized graph parameters: Gallai theorems I. *Bulletin of the Institute of Combinatorics and its Applications*, 17:27–37, 1996.

[198] P.J. Slater. LP-duality, complementarity and generality of graphical subset parameters. In T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, editors, *Domination in Graphs - Advanced Topics*, chapter 1, pages 1–29. Marcel-Dekker, 1998.

[199] R. Solis-Oba. 2-approximation algorithm for finding a spanning tree with maximum number of leaves. In *Proceedings of the 6th Annual European Symposium on Algorithms*, volume 1461 of *Lecture Notes in Computer Science*, pages 441–452. Springer-Verlag, 1998.

[200] A. Srinivasan, K. Madhukar, P. Nagavamsi, C.P. Rangan, and M.S. Chang. Edge domination on bipartite permutation graphs and cotriangulated graphs. *Information Processing Letters*, 56(3):165–171, 1995.

[201] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 627–636. IEEE Computer Society, 1996.

[202] G.A. Stephen. *String Searching Algorithms.* World Scientific, 1994.

[203] W.A. Sutherland. *Introduction to metric and topological spaces.* Oxford University Press, 1975.

[204] J.A. Telle and A. Proskurowski. Practical algorithms on partial k-trees with an application to domination-type problems. In *Proceedings of the 3rd Workshop on Algorithms and Data Structures*, volume 709 of *Lecture Notes in Computer Science*, pages 610–621. Springer-Verlag, 1993.

[205] V.G. Timkovskii. Complexity of common subsequence and supersequence problems and related problems. *Kibernetika*, 5:1–13, Sept-Oct 1989 (in Russian). English translation in *Cybernetics*, 25:565-580, 1990.

[206] J. Topp. Graphs with unique minimum edge dominating sets and graphs with unique maximum independent sets of vertices. *Discrete Mathematics*, 121:199–210, 1993.

[207] J. Topp and P.D. Vestergaard. Well-irredundant graphs. *Discrete Applied Mathematics*, 63:267–276, 1995.

[208] J. Topp and P.D. Vestergaard. Totally equimatchable graphs. *Discrete Mathematics*, 164:285–290, 1997.

[209] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.

[210] L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[211] M. Yannakakis. The analysis of local search problems and their heuristics. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science, Rouen, France*, volume 415 of *Lecture Notes in Computer Science*, pages 298–310. Springer-Verlag, 1990.

[212] M. Yannakakis. Computational complexity. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 2, pages 19–56. John Wiley and Sons, 1997.

[213] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 18(1):364–372, 1980.

[214] C.C. Yen and R.C.T. Lee. The weighted perfect domination problem. *Information Processing Letters*, 35(6):295–299, 1990.

[215] J.-X. Zhang, Z.-F. Zhang, and X.-Y. Ma. On the relations of graph parameters. *Chinese Science Bulletin*, 36(5):437–438, 1991.

[216] M. Zito. Personal communication, 1997.