Properties of Stable Matchings

by

Michael Jay Szestopalow

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Stable matchings were introduced in 1962 by David Gale and Lloyd Shapley to study the college admissions problem. The seminal work of Gale and Shapley has motivated hundreds of research papers and found applications in many areas of mathematics, computer science, economics, and even medicine. This thesis studies stable matchings in graphs and hypergraphs.

We begin by introducing the work of Gale and Shapley. Their main contribution was the proof that every bipartite graph has a stable matching. Our discussion revolves around the Gale-Shapley algorithm and highlights some of the interesting properties of stable matchings in bipartite graphs. We then progress to non-bipartite graphs. Contrary to bipartite graphs, we may not be able to find a stable matching in a non-bipartite graph. Some of the work of Irving will be surveyed, including his extension of the Gale-Shapley algorithm. Irving's algorithm shows that many of the properties of bipartite stable matchings remain when the general case is examined.

In 1991, Tan showed how to extend the fundamental theorem of Gale and Shapley to non-bipartite graphs. He proved that every graph contains a set of edges that is very similar to a stable matching. In the process, he found a characterization of graphs with stable matchings based on a modification of Irving's algorithm. Aharoni and Fleiner gave a non-constructive proof of Tan's Theorem in 2003. Their proof relies on a powerful topological result, due to Scarf in 1965. In fact, their result extends beyond graphs and shows that every hypergraph has a fractional stable matching. We show how their work provides new and simpler proofs to several of Tan's results.

We then consider fractional stable matchings from a linear programming perspective. Vande Vate obtained the first formulation for complete bipartite graphs in 1989. Further, he showed that the extreme points of the solution set exactly correspond to stable matchings. Roth, Rothblum, and Vande Vate extended Vande Vate's work to arbitrary bipartite graphs. Abeledo and Rothblum further noticed that this new formulation can model fractional stable matchings in non-bipartite graphs in 1994. Remarkably, these formulations yield analogous results to those obtained from Gale-Shapley's and Irving's algorithms. Without the presence of an algorithm, the properties are obtained through clever applications of duality and complementary slackness.

We will also discuss stable matchings in hypergraphs. However, the desirable properties that are present in graphs no longer hold. To rectify this problem, we introduce a new "majority" stable matchings for 3-uniform hypergraphs and show that, under this stronger definition, many properties extend beyond graphs. Once again, the linear programming tools of duality and complementary slackness are invaluable to our analysis. We will conclude with a discussion of two open problems relating to stable matchings in 3-uniform hypergraphs.

Acknowledgements

First and foremost, I wish to thank Penny Haxell. This thesis would not exist without her unwavering encouragement and endless patience.

I would like to thank Nick Harvey and Steve Furino for their time and comments related to this thesis. A very special thanks to Jeanne Youlden for her generous contribution that helped make it financially possible for me to continue my studies. Finally, I would like to thank Cathy Brown for helping to conceal my neglect of the English language.

Dedication

For my parents, Janie and Victor.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Imagine that you travel back through time to your final year of high school. It is an exciting time: The days of walking down the halls towards those "decidedly useless" classes are coming to a triumphant end. In the background, school counsellors can be found preaching the importance of a post-secondary education. Not surprisingly, we will assume that you are one of the best students in your school. Because of this assumption, the school counsellors are particularly interested in seeing you attend a post-secondary school and, eventually, have a profound impact on the rest of the world. Try as you may, you really will not find a better alternative. Wisely, you accept this idea without too much trouble. You will ultimately be faced with three questions.

The first of these questions is arguably the most important: "Which post-secondary institution should I choose to take all of my money?" The answer will not be found easily; factors such as tuition costs, distance from home, and program reputation will undoubtedly affect your eventual choice of school. From the other side, a university has to decide which students it will admit. Naturally, a school would want to admit the best students possible. However, this may not happen since the best students could prefer to attend another school. To help with this problem, imagine the existence of a central admissions committee that assigns students to schools based on the preferences of both. In addition, suppose Rebecca applied for admission to the University of Waterloo. Ideally, this central admissions committee would like to avoid the situation in which Rebecca is denied entry to UW, but would rather admit her than some other student who has already been accepted. Such a scenario would undermine the credibility of the committee since both the student and the school would have a reason to ignore the committee's decision.

The second question you will be faced with is: "Who am I going to be yelling at for spawning a new type of bacteria ten feet from where I sleep?" Once a school has a final list of its new first-year students, it now has the task of assigning roommates for the students living in residence. Similar to the first problem, the school does not want students taking matters into their own hands and reorganizing their living arrangements.

The final question comes after four years of hard work and many hours of lost sleep when you finally graduate with a post-secondary degree. You are now expected to start a life in the real world. Typically, this life will involve a successful career (your recently obtained degree will be useful in this endeavour) and, more relevant for this paper, a beautiful family. The question becomes: "Who am I going to spend the rest of my life with?"

These questions motivate the following three problems:

Problem 1 (College Admissions Problem)**.** A set of n high school students are applying to m universities. Naturally, each student can attend only one university and each university has some maximum quota of students. Students rank universities to which they apply, and each university ranks its applicants. Can we find an assignment of students to schools such that if a student is not attending a particular university, then either the student is attending a school he prefers, or the university has reached its admissions quota with students it prefers?

Problem 2 (Stable Roommates Problem)**.** Suppose there are n people living in a university dormitory. Each person ranks the other students in terms of who they would prefer to have as a roommate. Can we find an assignment of roommates such that if two people are not roommates then at least one of them prefers their current roommate?

Problem 3 (Stable Marriage Problem)**.** A community consists of n men and m women. Each person ranks members of the opposite sex in terms of who they would prefer for a spouse. Can we find a set of couples such that if two people are not married to each other then at least one of them prefers their spouse?

Notice that the stable marriage problem is a special case of the college admissions problem: Imagine that each university is only allowed to admit a single student. This is a ludicrous notion; the tuition fees of those poor students would be astronomical! Fortunately, this is where the marriage metaphor becomes useful. Restricting each university to only one student, while ridiculous in principle, yields the stable marriage problem.

We can view the stable marriage problem as an attempt to prevent affairs among married partners. While this is a noble cause, the recent "transgressions" of a certain professional golfer would suggest that preventing affairs is an impossible task. Needless to say, there are social reasons why the stable marriage problem may not accurately describe successful marriages, but we will not delve into such reasons here. However, the National Resident Matching Program (NRMP) is a real example where these problems find some use.

Similar to post-doctoral fellows, residents are recently graduated medical doctors who practice medicine under the supervision of a full physician until they have the experience to work as physicians themselves. The NRMP began in 1952 to match residents with hospitals based on the preferences of the participating parties. In fact, they try to solve the college admissions problem. Indeed, there is no shortage of examples of these two-sided preference models. Here at the University of Waterloo we can look to the Co-operative Education program: After their interviews, students rank potential employers in terms of work preference and the employers rank their student applicants.

Our three problems were formally introduced in 1962 by David Gale and Lloyd Shapley [13] and fall into the class of *stable matching* problems. The main contribution of Gale and Shapley was their proof that every instance of the college admissions problem, and therefore the stable marriage problem, has a solution. Their proof was based on an algorithm we will see later in Section 2.1.1.

Sadly, Gale and Shapley found the following example to show that an instance of the stable roommates problem may not have a solution: Suppose a dormitory consists of Andrew, Brandon, Christopher, and David. Andrew ranks Brandon first, Brandon ranks Christopher first, Christopher ranks Andrew first, and all three rank David last. David's preferences are immaterial at this point. David's roommate, regardless of who it is, will prefer both of the others, and one of those two will also prefer David's roommate [13]. Thus, there is no stable assignment of roommates.

However, the seminal work of Gale and Shapley has motivated some fascinating mathematics, and has been the starting point for hundreds of research papers. In a series of lectures in 1976, Donald Knuth [33] showed that the set of all stable matchings in a bipartite graph forms a distributive lattice. Although this fact was largely ignored for many years, it eventually led to some remarkable consequences. Knuth also conjectured that stable matchings and lattice theory might be more closely related than originally thought. We will see in Section 2.1.2 that this is indeed the case. As another example, the theory of stable matchings can be used to give a short proof of Galvin's Theorem for list-edge-colourings of bipartite multi-graphs, and hence, a proof of the Dinitz conjecture about partial Latin squares [16].

In the last five years, applications of stable matchings have found their way back to the medical fields. In particular, variants of stable matching problems have been used to model the so-called *kidney transplant problem*. This is essentially the problem of matching patients to donors where the preferences for a patient needing a kidney are based on the suitability of the potential donors [24, 43].

Obviously, the main objective of this thesis is to examine some of the interesting mathematical properties of stable matchings. Consequently, the bulk of our discussion will be an exposition of the works of Gale and Shapley, Knuth, Irving, and many others. Unfortunately, a common trait among stable matching papers is that many are written in the context of computer science. While it is undeniable that there have been many advancements in stable matching research, this language can be unfamiliar to certain aspiring mathematicians. We will try to present this material in the context of graph theory and limit the use of the lists and tables of computer science. Hopefully, this will clean up the presentation of many of the proofs.

Section 1.1 will conclude our introduction with a review of the necessary prerequisite material from graph theory, linear programming, and order theory.

In Chapter 2 we will introduce the stable matching problem. Starting with the work of Gale and Shapley, we highlight some of the interesting properties of stable matchings in bipartite graphs. The chapter will conclude with a look at stable matchings in non-bipartite graphs. We will see that we keep many of the properties of bipartite stable matchings when we jump to the general case.

The purpose of Chapter 3 will be to show that the fundamental theorem of Gale and Shapley from Chapter 2 can be extended to non-bipartite graphs by considering a suitable extension of stable matchings. We will consider a result of Tan, and a subsequent result of Aharoni and Fleiner, which shows that every graph contains a set of edges that is very similar to a stable matching. This special set of edges will allow us to characterize the graphs that have stable matchings. We will also take a vacation from the combinatorial side of stable matchings to visit the linear programming world. Duality and complementary slackness can provide alternate proofs to many of the results of Chapter 2.

Chapter 4 will bring stable matchings to 3-uniform hypergraphs. We will highlight some of the difficulties stable matchings present us when we move from graphs to hypergraphs and discuss some open questions and conjectures. To deal with these difficulties, we propose a new, stronger, definition of stable matching. Using techniques from linear programming, we obtain original results that show the behaviour of these "majority" stable matchings mirrors the behaviour of stable matchings in graphs more closely than the standard stable matching in 3-uniform hypergraphs. We will also construct a very large class of 3-uniform hypergraphs admitting a majority stable matching.

Chapter 5 highlights two open questions about stable matchings in 3-uniform hypergraphs. The first asks if there is always a stable matching when the preferences are structured in a specific way. The second asks if every 3-uniform hypergraph has a fractional stable matching where all the edge values are not too small. For both problems, we prove that small instances have a positive answer.

1.1 Background Check

Possibly excepting some of the order theory, most undergraduates in the Department of Combinatorics and Optimization here at the University of Waterloo will have seen all of this background material by the time they finish the third year of their studies. For this reason, the majority of people who voluntarily read this thesis can skip the remainder of this chapter. However, we include our background section should a reader need a brief review of basic graph theory, linear optimization, or order theory.

1.1.1 Graph Theory

A *graph* is a pair (V, E) where V is a finite set, called the set of *vertices*, and E is a set of two element subsets of V, called the set of *edges*. More generally, a *hypergraph* is a pair (V, E) where V is a finite set and E is a set of subsets of V. If a hypergraph H has the property that every edge has size r then we say that H is *r-uniform*. In particular, a 2-uniform hypergraph is a graph.

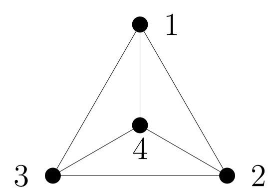

Figure 1.1: Example of a graph

As in Figure 1.1, we will typically represent a graph as a drawing in the plane with points (vertices) and lines (edges) connecting some, or all, of the points. However, it is sometimes useful to represent a graph by a matrix. The *vertex-edge incidence matrix* of a hypergraph $G = (V, E)$ is a $|V| \times |E|$ matrix M where the (v, e)-entry of M is defined as follows:

$$M_{(v,e)} = \begin{cases} 1 & \text{if } v \text{ is an endpoint of } e \\ 0 & \text{otherwise.} \end{cases}$$

for every $v \in V$ and $e \in E$. For the graph in Figure 1.1, we give the following vertex-edge incidence matrix:

$$M = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

In a graph, a vertex v is a *neighbour* of vertex u if $uv \in E$. The *neighbourhood* of a vertex u, denoted $N(u)$, is the set of neighbours of u. Let $v \in V$. The *degree of* v, denoted $deg(v)$, is defined to be the number of edges e such that v is an endpoint of e. If $deg(u) = 0$ for some vertex u then we say that u is *isolated.* Note that in a graph, an isolated vertex has an empty neighbourhood.

A *directed graph*, or *digraph*, is a pair (N, A) where N is a finite set of *nodes* and A is a set of ordered pairs of distinct nodes of N, called *arcs.* The distinction here is that the arcs have a direction associated with them: If $a = (u, v)$ is an arc, we think of the arc as being directed from u to v. Therefore, in Figure 1.2, we see that the arcs $(2, 3)$ and $(3, 2)$ are different. Alternatively, we will say that u is the *tail* of arc a and v is the *head* of a; these will be denoted by $tail(a)$ and $head(a)$, respectively.

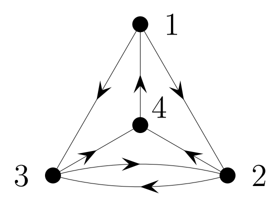

Figure 1.2: Example of a directed graph

Let $G = (V, E)$ be a hypergraph. We say $H = (\bar{V}, \bar{E})$ is a *subhypergraph of* G if $\bar{V} \subseteq V$ and $\bar{E} \subseteq E$. We will often denote $V(H)$ and $E(H)$ to be the vertices and edges of the subhypergraph, respectively.

A *matching* in a hypergraph $G = (V, E)$ is a set of edges, $M \subseteq E$, such that each vertex in V is incident to at most one edge of M. A matching M is *maximal* if $M \cup e$ is not a matching for any $e \in E \backslash M$. In Figure 1.3, the matching is not maximal since we could add the edge 34 to obtain a larger matching. Further, if every vertex is incident to exactly one edge of M, then M is a *perfect* matching.

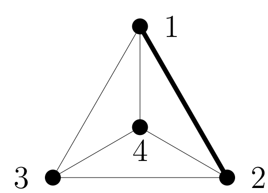

Figure 1.3: Example of a non-maximal matching

A graph $G = (V, E)$ is *bipartite* if there is a partition (A, B) of V such that every edge of E has exactly one endpoint in A and one endpoint in B. Similarly, a 3-uniform hypergraph is *tripartite* if there is a partition (A, B, C) of V such that every edge of E has exactly one endpoint in each of A, B, and C.

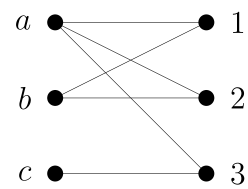

Figure 1.4: Example of a bipartite graph

A *cycle* in a graph is a subgraph C with n distinct vertices $v_0, v_1, \ldots, v_{n-1}$ such that $v_i v_{i+1} \in E(C)$ for all i modulo n, $n \geq 3$. It will be convenient for us to refer to a cycle by its edges: $v_0v_1, v_1v_2, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_0$. If n is an odd number then the cycle is *odd*.

We can also discuss cycles in a directed graph: A *directed cycle* is a subdigraph B with n distinct nodes $u_0, u_1, \ldots, u_{n-1}$ such that $u_i u_{i+1}$ is an arc for all i modulo n, $n \geq 2$. The key here is that the arcs of the directed cycle have a consistent orientation. As examples, $C_1 = 12, 23, 31$ is an odd cycle in Figure 1.1, $C_2 = a1, 1b, b2, 2a$ is an even cycle in Figure 1.4, and $C_3 = 13, 34, 41$ is a directed cycle in Figure 1.2. A well-known result shows us a very close relationship between a bipartite graph and its set of cycles.

Theorem 1.1.1. *A graph is bipartite if and only if it does not have an odd cycle.*

In Chapter 3, we will see a result analogous to Theorem 1.1.1 for stable matchings in graphs. Specifically, we will see how odd cycles can provide a characterization of stable matchings.

A more complete review of graph theory can be found in the excellent book by Bondy and Murty [8].

1.1.2 Linear Programming

For us, *linear programming* is the problem of maximizing a linear function of a finite number of real variables subject to a finite number of linear inequalities. Any linear program can be expressed in the following form:

$$\begin{array}{rrcl} \textbf{max } c^T x & & & \text{(P)} \\ \textbf{subject to: } Ax & \leq & b & \\ x & \geq & 0, & \end{array}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. This is called the *primal* problem. A *feasible solution* of (P) is a vector $x \in \mathbb{R}^n$ such that $Ax \leq b$ and $x \geq 0$. A feasible solution, x^*, is an *optimal solution* of (P) if $c^T x^* \geq c^T x$ for every feasible solution, x, of (P). Associated with (P) is another linear program:

$$\begin{aligned} \textbf{min } b^T y & & \qquad \text{(D)} \\ \textbf{subject to: } A^T y &\geq c \\ y &\geq 0. \end{aligned}$$

This is the *dual* linear program. The feasible solutions of (P) have a special relationship with the feasible solutions of (D).

Lemma 1.1.2 (Weak Duality)**.** *If $\bar{x}$ is a feasible solution to* (P) *and $\bar{y}$ is a feasible solution to* (D)*, then $c^T\bar{x} \leq b^T\bar{y}$.*

Corollary 1.1.3. *If $\bar{x}$ is a feasible solution to* (P)*, $\bar{y}$ is a feasible solution to* (D)*, and $c^T\bar{x} = b^T\bar{y}$, then $\bar{x}$ is optimal for* (P) *and $\bar{y}$ is optimal for* (D)*.*

The ultimate goal of linear programming is to find an optimal solution to (P). Corollary 1.1.3 gives us a simple way to check the optimality of a solution without resorting to an algorithm to solve linear programs. Once we have an optimal solution to our linear program, we would like to be able to deduce some useful properties.

Theorem 1.1.4 (Complementary Slackness)**.** *Let x^* and y^* be feasible solutions to* (P) *and* (D)*. Then x^* and y^* are optimal for* (P) *and* (D) *if and only if*

- $x_j^* = 0$ *or* $(row_j(A^T))y^* = c_j$ *for all* $j \in \{1, \ldots, n\}$*, and*
- $y_i^* = 0$ *or* $(row_i(A))x^* = b_i$ *for all* $i \in \{1, \ldots, m\}$*.*

A vector $z \in \mathbb{R}^n$ is a *convex combination* of x and y if $z = \lambda x + (1 - \lambda)y$ for some λ such that $0 \leq \lambda \leq 1$. In $\mathbb{R}^2$ this is easy to visualize: The set of all convex combinations of x and y is simply the line segment between x and y. A convex combination is *strict* if $0 < \lambda < 1$. A set $C \subseteq \mathbb{R}^n$ is *convex* if for every $x, y \in C$ and every real number λ with $0 \leq \lambda \leq 1$, $\lambda x + (1 - \lambda)y \in C$. In other words, C is convex if for any $x, y \in C$, C contains all convex combinations of x and y.

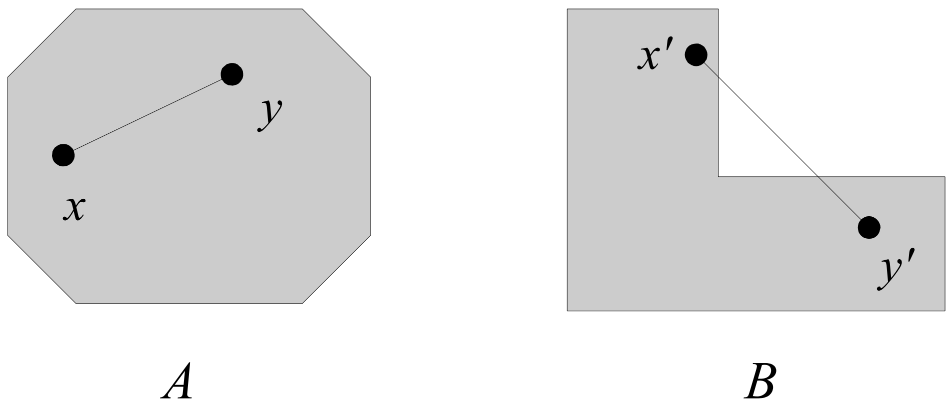

Figure 1.5: Convex and non-convex sets

Figure 1.5 illustrates the difference between convex and non-convex sets. Set A is convex since the line segment between any two points in A is contained in A. Set B is not convex since the line segment between points y and z goes outside of B. The following result is a trivial application of the definition of a convex set.

Lemma 1.1.5. *The feasible set of any linear program is a convex set.*

A vector w of a convex set C is an *extreme point* of C if it cannot be written as a strict convex combination of two distinct points in C. In Figure 1.5, any "corner" of set A is an extreme point of A.

Sometimes it is useful, and possibly necessary, to consider solutions of (P) where all the components are integers. If we restrict all the variables of (P) to take integral values, we obtain an *integer linear program*. Although they are notoriously difficult to solve to optimality [32], integer linear programs are very powerful as a modelling tool. Indeed, many combinatorial problems can be formulated as integer linear programs; the problem of finding a maximum matching in a graph can be expressed as:

$$\begin{array}{rcl} \textbf{max } e_E^T x & & \qquad (\mathrm{P}_{MATCH}) \\ \textbf{subject to: } Mx & \leq & e_V \\ x \in \{0,1\}^n & & \end{array}$$

where M is the vertex-edge incidence matrix of the graph, and e_E and e_V are the vectors of all 1's in $\mathbb{R}^E$ and $\mathbb{R}^V$, respectively.

This will be enough linear programming for this paper. However, we refer the reader to the book of Bertsimas and Tsitsiklis [5] if our brief review is insufficient.

1.1.3 Order Theory

A *partially ordered set* is a pair $(P, \preceq)$ where P is a set and $\preceq$ is a binary relation on the elements of P such that for all $a, b, c \in P$,

- $a \preceq a$,
- if $a \preceq b$ and $b \preceq a$ then $a = b$, and
- if $a \preceq b$ and $b \preceq c$ then $a \preceq c$.

In other words, '$\preceq$' is a binary relation on P that is reflexive, transitive, and antisymmetric. Further, we will use $a \prec b$ to represent that $a \preceq b$ and $a \neq b$. For brevity we will say $(P, \preceq)$ is a *poset* or *partial order*. For example, let $a, b \in \mathbb{N}$ and define $a \preceq b$ to mean that a divides b. Then $(\mathbb{N}, \preceq)$ is a poset.

Let $(P, \preceq)$ be a poset. If $a \not\preceq b$ and $b \not\preceq a$ then a and b are *incomparable*. In $(\mathbb{N}, \preceq)$, 2 and 3 are incomparable. The *converse* of a poset $(P, \preceq)$ is the poset $(P, \preceq^c)$ where $a \preceq^c b$ if and only if $b \preceq a$.

We can visually represent a poset by its *Hasse diagram*. We can think of a Hasse diagram as a graph: Construct a vertex for each element of P and add an edge from a up to b if $a \prec b$ and there does not exist a $c \in P$ such that $a \prec c \prec b$. Notice that, unlike a graph, the relative positions of vertices can be important. Informally, we mean "big elements go above small elements".

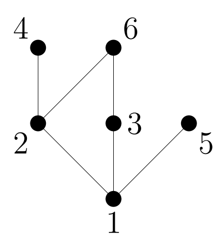

Figure 1.6: Hasse diagram

Figure 1.6 shows the Hasse diagram of $(\{1, 2, 3, 4, 5, 6\}, \preceq)$ where $\preceq$ again represents divisibility. The Hasse diagram of the converse of $(P, \preceq)$ is obtained by simply turning the Hasse diagram of $(P, \preceq)$ upside down.

A partial order $(P, \preceq)$ is *totally ordered* if P does not have any incomparable elements. The set of natural numbers with the canonical meaning of $\leq$, $(\mathbb{N}, \leq)$, is a totally ordered set. A *total extension* of partial order $(P, \preceq)$ is a totally ordered set $(P, \leq)$ such that if $a \preceq b$ then $a \leq b$. Notice that $(\mathbb{N}, \leq)$ is a total extension of $(\mathbb{N}, \preceq)$.

Let $(P, \preceq)$ be a poset, and let a and b be two elements of P. An element c of P is the *join* (least upper bound) of a and b, if the following two conditions are satisfied:

- $a \leq$ c and $b \leq c$, and
- for any w in P, such that $a \leq w$ and $b \leq w$, we have $c \leq w$.

We will denote such a c by $a \vee b$. Analogously, we can define the *meet* of a and b, $a \wedge b$ (greatest lower bound of a and b). If $(P, \preceq)$ is a poset and for every $a, b \in P$ there exist elements $a \vee b$ and $a \wedge b$ in P then we say that $(P, \preceq)$ is a *lattice*.

A lattice is *distributive* if the following identities hold for every $a, b, c \in P$:

- $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, and
- $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.

These are often called the distributivity laws.

For our purposes, this is a sufficient introduction to order theory. However, additional material can be found in Grätzer's book [17]. We are now ready to discuss stable matchings.

Chapter 2

Stable Matchings in Graphs

Our examination of stable matchings begins with graphs. As we will see, our problems from Chapter 1 can be modelled as matching problems in appropriate graphs. However, our definition of a "graph" is incomplete for the purpose of studying stable matchings. Notice that a major component of Problems 1, 2, and 3 is the set of preferences of the parties involved. Intuitively, the set of stable matchings should depend on the set preferences. Indeed, this is the case. Therefore, we will be concerned with graphs that have additional structure.

Let $G = (V, E)$ be a graph. For a vertex v, a *preference list of* v, L_v, is a totally ordered list of the edges that contain v. If every vertex of G has a preference list we will say that $G = (V, E, L)$ is a *graph with preferences* where L is the set of vertex preference lists. We will take advantage of the fact that, in a graph, a vertex could rank either its incident edges or its neighbour vertices; there is a one-to-one correspondence between the two sets. We will take the convention that $a >_v b$ means vertex v prefers edge a to edge b. We can also replace the preference list of a vertex with a partially ordered set, called the *preference poset*. In this case G is a *graph with poset preferences*. Suppose P_v is the preference poset for vertex v. If a and b are incomparable elements of P_v, then v considers a and b to be equally as good or tied.

Let $G = (V, E, L)$ be a graph with preferences and let M be a matching of G. If there is an edge xy such that x and y prefer each other to their partners in the matching M, then we will call it a *blocking edge* for M. It is certainly possible that an endpoint of a blocking edge is unmatched in a matching. For our purposes, it will be convenient to imagine such a vertex as being matched to itself. This will serve the noble purpose of reducing the number of cases in many of our proofs. We will also say that the *v-rank* of an edge e is the position of e in v's preference list (i.e. v's favourite edge has a v-rank of 1).

A matching M is a *stable matching* if for every edge $xy \notin M$ either x prefers its partner in M to y or y prefers its partner in M to x. Equivalently, we could define a stable matching to be a matching with no blocking edges. Notice that a stable matching is necessarily maximal; otherwise, the edge that we could add to obtain a bigger matching would be a blocking edge. Figure 2.1 gives us an example of a stable matching. A vertex y is a *stable partner of* x if xy is an edge of some stable matching; we will call xy a *stable edge*.

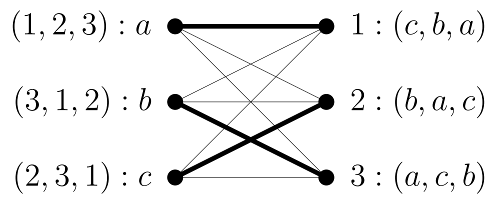

Figure 2.1: Example of a stable matching

Naturally, we will first consider stable matchings in the bipartite setting. It will turn out that we will always be able to find a stable matching, regardless of the bipartite graph with preferences. Afterwards, we will move to the non-bipartite case. The difference here will be that we can find examples of graphs with preferences without stable matchings. However, there are many properties that bipartite and non-bipartite stable matchings still share; we will try to feature the most interesting of these properties.

2.1 Bipartite Graphs

The stable marriage problem can be formulated as the problem of finding a stable matching in a bipartite graph with preferences: Let the vertex classes be $A := \{men\}$ and $B := \{women\}$. We add the edge ab if man a and woman b both consider each other to be possible partners. The preferences for a vertex will simply be the preferences of the person it is representing.

Similarly, we can formulate the college admissions problem. However, since each university will likely admit more than one student, the college admissions problem is not precisely a matching problem. We can circumvent this complication by constructing clone vertices for each university. Specifically, if university a is willing to admit at most k students, then we should have k copies of vertex a, where each clone vertex is incident to exactly the same set of student vertices. Gale and Shapley gave an algorithm that would solve both the stable marriage and college admissions problems and, in the process, prove the following seminal result.

Theorem 2.1.1 (The Fundamental Theorem of Stable Matchings [13]). *Every bipartite graph with preferences has a stable matching.*

The ultimate goal of the next section will be to highlight this result and the Gale-Shapley algorithm. Later on we will also see that for a bipartite graph with preferences, the set of stable matchings forms a distributive lattice.

2.1.1 The Gale-Shapley Algorithm

The stable matching algorithm proposed by Gale and Shapley is very simple. Let $G = (A \cup B, E, L)$ be a bipartite graph with preferences. In turn, each unmatched vertex in A proposes to its favourite neighbour vertex to which it has not yet proposed. If vertex b is unmatched and receives a proposal from a, b accepts the proposal. If vertex b is matched and likes the proposal from a better than its current proposal, b accepts a's proposal and rejects the old proposal; if b likes the current proposal better, then b rejects a. If a vertex is rejected, then it proposes to its next most preferred neighbour. We stop once every $a \in A$ is either matched or has proposed to all of its neighbours.

Algorithm 1 Gale-Shapley Algorithm

Input: a bipartite graph with preferences, $G = (A \cup B, E, L)$
Output: a stable matching M

set $K(x) := L_x \ \forall\ x \in A \cup B$
set $M := \emptyset$
while $\exists\ a \in A$ unmatched in M and $K(a) \neq \emptyset$ **do**
    let b be the most preferred vertex of $K(a)$
    if b is not covered by M **then**
        $M := M \cup \{ab\}$
    else if $zb \in M$ for some $z \neq a$ and $a >_b z$ **then**
        $M := M \backslash \{zb\} \cup \{ab\}$
    end if
    $K(a) := K(a) \backslash \{b\}$
    $K(b) := K(b) \backslash \{u | a >_b u\}$
end while
output M, a stable matching

An amazing feature of this procedure is that it is very natural. It is the obvious thing to do: Start with your favourite possible partner and work your way down the list until you are matched with a partner or you run out of possible partners. Even more amazing is that such a natural procedure is correct!

Theorem 2.1.2 (Gale and Shapley [13]). *For any bipartite graph with preferences, the Gale-Shapley Algorithm terminates with a stable matching.*

Proof: Let $G = (A \cup B, E, L)$ be a bipartite graph with preferences and let M^* be the set of edges returned by the algorithm. Lines 3 through 9 ensure that M^* is indeed a matching. Suppose, for a contradiction, that M^* is not stable. Then there exists a blocking edge $ab \in E$. Specifically, there exists $a \in A$ and $b \in B$ such that $ab \notin M^*$ and both a and b prefer each other to their partners (if they have one) in M^*.

Since $ab \notin M^*$ then either there exists $z \in A$ who proposed to b during the algorithm such that $z >_b a$ or b never received a proposal throughout the algorithm. Notice that if b receives a proposal, then b is guaranteed to be matched at the end of the algorithm and will only accept a new proposal if it is better than the current one. Therefore, if b receives a proposal, b is matched to a vertex w such that $w \geq_b z >_b a$, contradicting our assumption that ab is a blocking edge for M^*. If b did not receive a proposal then every vertex in A is matched to a vertex in B that it prefers to b, again contradicting our assumption. □

Observe that in order for the algorithm to always output a stable matching, any bipartite graph with preferences must have a stable matching. In this way, Theorem 2.1.1 follows directly from the algorithm.

Returning to our University of Waterloo Career Services example from Chapter 1, a reasonable conjecture is that the algorithm used to match students to employers is a variant of the above algorithm, with $A = \{employers\}$ and $B = \{students\}$. Unfortunately, a quick search through the Career Services website shows that this conjecture is false. The algorithm used by the Co-operative Education program is "greedy" and does not guarantee a stable matching [50]. However, it is noteworthy that, ten years before the the Gale-Shapley paper was published, the NRMP began using the Gale-Shapley algorithm to match newly graduated doctors with hospitals [41]. The NRMP has proven so successful that the program continues to use a version of the algorithm and thousands of hospitals and doctors participate in the program [38].

We will refer the reader to the books of Knuth [33], or Gusfield and Irving [19], for a simple analysis of this algorithm. However, if the reader is not interested, we will just mention that the analysis essentially boils down to the fact that each vertex will propose to each of its neighbours at most once. If there are n vertices then each vertex has at most $n-1$ neighbours; this gives us $O(n^2)$ as the running time.

The Gale-Shapley algorithm leads to some interesting theoretical consequences. We will mention two of these here. However, we will save the proofs of these results until Section 2.3.

Theorem 2.1.3 (Gale and Sotomayor [14], McVitie and Wilson [35])**.** *Let G be a bipartite graph with preferences. If vertex v is unmatched in a particular stable matching of G, then v is unmatched in all stable matchings of G.*

Corollary 2.1.4. *Let G be a bipartite graph with preferences. All stable matchings of G have the same size.*

In anticipation of the underlying structure of stable matchings, we will call a stable matching A-*optimal* if every vertex in A is matched to the most preferred partner it can have in any stable matching. Notice that in a bipartite graph with preferences, the A-optimal stable matching, should it exist, is unique. However, the existence of such a stable matching seems unlikely (the fact that it is a matching is surprising). Remarkably, Gale and Shapley showed that their algorithm could do more than just guarantee the existence of a stable matching.

Lemma 2.1.5 (Gale and Shapley [13]). *For any bipartite graph with preferences, the Gale-Shapley Algorithm terminates with the A-optimal stable matching.*

Proof: Let M^* be the stable matching given by the algorithm. Suppose, for a contradiction, that M^* is not the A-optimal stable matching. Then there exists another stable matching, M, and vertices $a \in A$ and $x, y \in B$ such that $ax \in M^*$, $ay \in M$, and $y >_a x$. Since $ay \notin M^*$, y must have rejected a during the algorithm. This rejection must have happened because y received a proposal from a vertex $z \in A$ such that $z >_y a$. Since proposals occur one at a time, we may assume that this is the first time during the algorithm that a vertex in B rejected a proposal from a stable partner.

Now z cannot have a better stable partner than y since, at the time of z's proposal to y, y was first on z's preference list and ay was the first stable edge that was rejected during the algorithm. Thus, since $zy \notin M$, z must prefer y to its partner in M. But then zy is a blocking edge for M, contradicting that M is a stable matching. □

We now know that the A-optimal stable matching exists. But, we actually know more than that! The uniqueness of the A-optimal stable matching shows that the Gale-Shapley algorithm will always terminate with the same stable matching. However, throughout our discussion of Gale-Shapley, we did not specify a proposal order for the vertices in A. Surprisingly, this is not a mistake. Such an order is not necessary! In the proof of Lemma 2.1.5 there were no special assumptions about the execution of the algorithm that produced M^*. Hence the proposal order is irrelevant: ANY execution of the algorithm on a bipartite graph with preferences will yield the A-optimal stable matching. We will see this non-determinism again when we look at Irving's algorithm.

Many aspects of our world, especially athletics and politics, share a common trait: The triumph of one group of people usually comes at the expense of another group. This is also true with stable matchings in bipartite graphs with preferences.

Corollary 2.1.6 (McVitie and Wilson [36]). *In the A-optimal stable matching, each vertex in B is matched with the least preferred vertex it can have in any stable matching.*

Proof: Let M^* be the A-optimal stable matching and let M be another stable matching. Let $a, w \in A$, $x, y \in B$ such that $ay \in M^*$, $wy \in M$, and $ax \in M$. Suppose, for a contradiction, that $a >_y w$. Since M^* is the A-optimal stable matching and $ay \in M^*$, we must have $y >_a x$. Then ay is a blocking edge for M. But this contradicts that M is a stable matching, since $ay \notin M$. $\square$

This result justifies using the alternate term *B-pessimal* in place of A-optimal. By the noted anti-symmetry of the stable matching problem, we obtain the B-optimal/A-pessimal stable matching by simply running the algorithm with the women as the proposers. Further, the previous two results will give us maximal and minimal elements of the promised lattice.

2.1.2 The Lattice of Stable Matchings

In general, a bipartite graph with preferences can have more than one stable matching. The example in Figure 2.2 has ten stable matchings. Can you find them all? In point of fact, it is known there exist graphs with preferences with exponentially many stable matchings [19] and the problem of counting stable matchings is #P-complete [28].

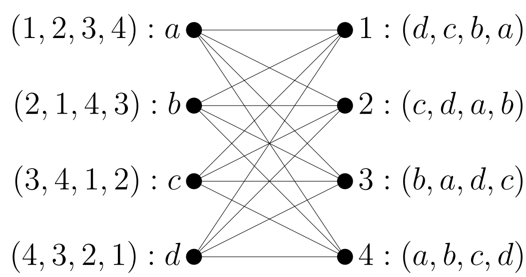

Figure 2.2: Bipartite graph with 10 stable matchings

We have been advertising that the set of stable matchings forms a distributive lattice. As a first step towards this fact, we claim that the set of stable matchings is a partially ordered set. Let M_1 and M_2 be stable matchings of $G = (A \cup B, E, L)$. We say that $M_1 \preceq M_2$ if every vertex in A has at least as good a partner in M_2 as it does in M_1. It is easy to see that $\preceq$ is reflexive, antisymmetric, and transitive. This work is originally due to Donald Knuth and John Conway [33].

Theorem 2.1.7. *Let $G = (A \cup B, E)$ be a bipartite graph with preferences, and let M_1 and M_2 be stable matchings of G. If each vertex in A is paired with its most preferred neighbour from M_1 and M_2, then the result, $M_1 \vee M_2$, is a stable matching.*

Proof: If $M_1 = M_2$, then the result is trivial. We will first show that $M_1 \vee M_2$ is a matching. Suppose, for a contradiction, that $M_1 \vee M_2$ is not a matching. Then there exists $a, x \in A$ and $b, y, z \in B$ such that $ab, xy \in M_1$, $az, xb \in M_2$, with both $b >_a z$ and $b >_x y$.

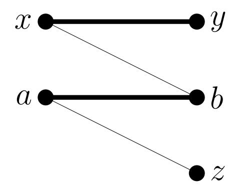

Figure 2.3: Stable matchings M_1 (thicker edges) and M_2

Since $b >_a z$, we must have $x >_b a$; otherwise M_2 would not be stable. But this contradicts that M_1 is stable since, by above, $b >_x y$ making xb a blocking edge for M_1. Thus, $M_1 \vee M_2$ is indeed a matching.

To show that $M_1 \vee M_2$ is indeed stable, we again suppose the contrary. Then there exist vertices u and v such that uv is a blocking edge for $M_1 \vee M_2$. We may assume that $u \in A$. Since u prefers v to its partner in $M_1 \vee M_2$, the definition of $M_1 \vee M_2$ tells us that u prefers v to its partners in both M_1 and M_2. Now v's partner in $M_1 \vee M_2$ will either be its partner in M_1 or M_2. Suppose v's partner in $M_1 \vee M_2$ is its partner from M_1. But since uv is a blocking edge for $M_1 \vee M_2$, v prefers u to its partner in M_1. This means that uv is also a blocking edge for M_1, contradicting that M_1 is a stable matching. Hence, $M_1 \vee M_2$ is indeed a stable matching. $\square$

Corollary 2.1.8. *Let $G = (A \cup B, E)$ be a bipartite graph with preferences, and let M_1 and M_2 be stable matchings of G. Then $M_1 \vee M_2$ is the least upper bound of M_1 and M_2.*

Proof: From the definition of $\preceq$, we can see that $M_1 \preceq M_1 \vee M_2$ and $M_2 \preceq M_1 \vee M_2$. Let $\bar{M}$ be a stable matching satisfying $M_1 \preceq \bar{M}$ and $M_2 \preceq \bar{M}$. Then every vertex in A must have a partner in $\bar{M}$ that it prefers at least as much as its partners in M_1 and M_2. Therefore $M_1 \vee M_2 \preceq \bar{M}$, and $M_1 \vee M_2$ is the least upper bound of M_1 and M_2. $\square$

In a similar way, we can define $M_1 \wedge M_2$ to be the stable matching where each vertex in A is paired with its least preferred partner from M_1 and M_2. Thus, for any two stable matchings, M_1 and M_2, we can find a least upper bound, $M_1 \vee M_2$, and a greatest lower bound, $M_1 \wedge M_2$. Hence, $(\mathscr{M}, \preceq)$ is a lattice. Notice that the converse of this lattice is obtained by simply switching perspective to the B-vertices. The Hasse diagram for the stable matching example in Figure 2.2 can be found in Figure 2.4. The theorem also shows that the A-optimal stable matching is indeed the maximal element of our lattice. In point of fact, Donald Knuth [33] noted that this lattice has much more structure.

Theorem 2.1.9. *Let $G = (A \cup B, E)$ be a bipartite graph with preferences, and let $\mathscr{M}$ be the set of all stable matchings for G. Then the lattice $(\mathscr{M}, \preceq)$ is distributive.*

Proof: Let M_1, M_2, and M_3 be stable matchings in G. Suppose that $x \in A$ is matched to y_{M_1}, y_{M_2}, $y_{M_3} \in B$, respectively. We note that there are six possible permutations of

y_{M_1}, y_{M_2}, and y_{M_3} in the preference list for x. A simple calculation for each case yields $M_1 \wedge (M_2 \vee M_3) = (M_1 \wedge M_2) \vee (M_1 \wedge M_3)$ and $M_1 \vee (M_2 \wedge M_3) = (M_1 \vee M_2) \wedge (M_1 \vee M_3)$, which are exactly the distributive laws. □

There is much more substantial research into stable matching lattice theory [7, 19, 18, 25, 28]. A particularly interesting result, motivated by a question of Knuth [33], shows how strong the connection is between stable matchings and lattice theory.

Theorem 2.1.10 (Blair [7])**.** *Every finite distributive lattice is the lattice of stable matchings for some bipartite graph with preferences.*

Blair's construction took a lattice on n points and gave a bipartite graph with preferences with $O(2^n)$ vertices. Gusfield, Irving, Leather, and Saks [18] provided an efficient construction of the stable matching instance, while improving the size of the graph to $O(n^2)$.

Our detour through lattice theory may, admittedly, feel a little out of place here. However, these remarkable and, in some sense, fairly simple, facts deserve inclusion in our discussion.

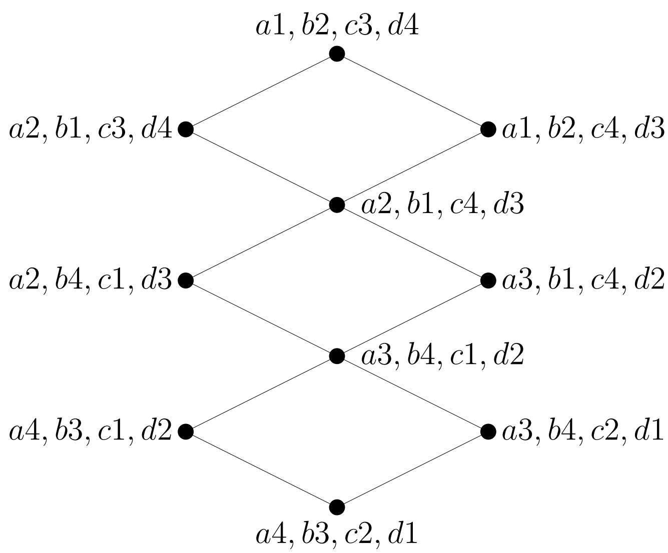

Figure 2.4: Hasse diagram of stable matchings

2.2 Non-Bipartite Graphs

We begin our section on non-bipartite graphs with an example to highlight the main obstacle when considering stable matchings in non-bipartite graphs with preferences. In point of fact, the example in Figure 2.5 is the graph version of the unsolvable instance of the stable roommates problem we saw in Chapter 1.

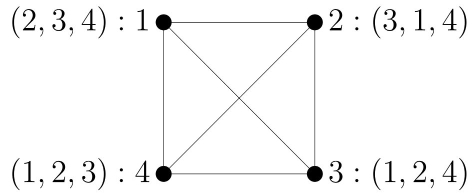

Figure 2.5: Example with no stable matchings

In any maximal matching, one of vertices $\{1, 2, 3\}$ must be matched to vertex 4. Suppose the matching is $M = \{14, 23\}$. Then the edge 13 blocks M since $3 >_1 4$ and $1 >_3 2$. Similarly, edge 12 blocks the matching $\bar{M} = \{24, 13\}$ and edge 23 blocks the matching $\tilde{M} = \{12, 34\}$. Hence, the graph with preferences has no stable matching. Notice that $12, 23, 31$ is an odd cycle. The existence of such an odd cycle leads to a characterization of bipartite graphs.

Theorem 2.2.1 (Abeledo and Isaak [1])**.** *Let $H = (V, E)$ be a graph. Then, H is bipartite if and only if the graph with preferences $G = (V, E, L)$ has a stable matching for any set of preferences L.*

Proof: Let $H = (V, E)$ be a graph. If H is bipartite, then by Theorem 2.1.1, $G = (V, E, L)$ has a stable matching for any set of preferences. Therefore, we can suppose that H is non-bipartite. We need to show that there is a set of preferences, L, such that $G = (V, E, L)$ does not have a stable matching. By Theorem 1.1.1, H has an odd cycle $C = v_0v_1, v_1v_2, \ldots, v_{n-1}v_0$. For each i modulo n, set v_{i+1} and v_{i-1} to have v_i-rank 1 and 2, respectively. The remaining preferences can be filled in arbitrarily.

Suppose, for a contradiction, that $G = (V, E, L)$ has a stable matching M. Since C is an odd cycle, there exists an i such that v_i is not matched to either v_{i-1} or v_{i+1}. Then $v_{i-1}v_i \notin M$, v_i is first on v_{i-1}'s preference list, and v_i is matched to, at best, its third favourite neighbour. This means that $v_{i-1}v_i$ is a blocking edge of M, contradicting that M is a stable matching. Thus, $G = (V, E, L)$ does not have a stable matching. □

Since we now know that there exist graphs with preferences that do not have stable matchings, the best algorithm we could hope for would be one that finds a stable matching if it exists, or tells us that there is no stable matching. In 1985, Robert Irving [26] provided such an algorithm and disproved a conjecture of Knuth, who suggested the problem might be NP-complete [33].

2.2.1 Irving's Algorithm

Irving's algorithm is divided into two phases. The first is very similar to the Gale-Shapley algorithm and removes edges that could not possibly be part of any stable matching. The main difference here is that vertices can both give and receive proposals. The second phase proceeds by deleting edges in a more specialized way until either we are left with a stable matching or we reach a condition that tells us that there is no stable matching.

We will call a vertex *free* if does not have a neighbour holding its proposal. Phase 1 starts with a vertex a proposing to its favourite neighbour, say b. Vertex b accepts the proposal, meaning that a is no longer free, and then deletes any edge it prefers less than edge ab. The algorithm continues with free vertices proposing in the same manner as vertex a. The vertices that receive proposals delete edges in the same manner as vertex b. Notice that these edge deletions prevent a proposal from being rejected immediately, since any such rejection will have already been ruled out. If a vertex is currently holding a proposal and receives a new one, then by the previous edge deletions it would accept the new proposal and delete any less preferable edges still remaining. Hence, the previous proposal edge would be deleted and the rejected vertex would become free again. Phase 1 stops once every vertex is either isolated or no longer free. If the edges form a matching, then it is stable. Otherwise we will continue on to Phase 2.

Algorithm 2 Irving's Algorithm - Phase 1

Input: a graph with preferences $G = (V, E, L)$
Output: the phase 1 subgraph $G^1 = (V, E^1, L^1)$
  set $L^1(x) := L_x \;\forall\; x \in V$ and $E^1 := E$
  while $\exists\; x \in V$ such that x is free and $L^1(x) \neq \emptyset$ **do**
    let y be the most preferred vertex of $L^1(x)$
    if y is holding a proposal from z **then**
      assign z to be free
    end if
    $E^1 := E^1 \backslash \{ay : x >_y a\}$
    $L^1(y) := L^1(y) \backslash \{a : x >_y a\}$
    for all a such that $x >_y a$ **do**
      $L^1(a) := L^1(a) \backslash y$
    end for
  end while
  $V^1 := \{v \in V : deg(v) \geq 1\}$
  output $G^1 = (V^1, E^1, L^1)$, the phase 1 subgraph

For this section, our pet example will be K_6 with preferences, as shown in Figure 2.6. We will use this graph to demonstrate the algorithm.

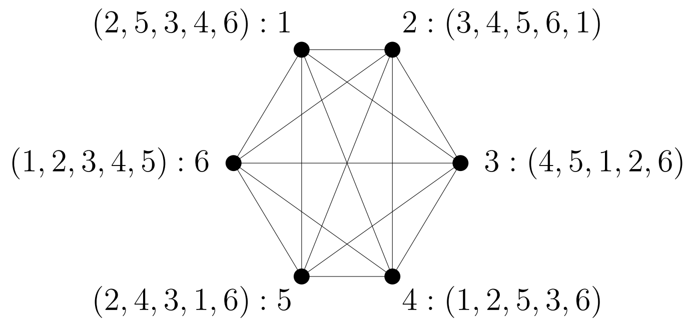

Figure 2.6: K_6 with preferences

Applying Phase 1 to our example we obtain the subgraph Figure 2.7. For similar reasons to the Gale-Shapley algorithm, the order of the proposals is, once again, inconsequential to the outcome of Phase 1.

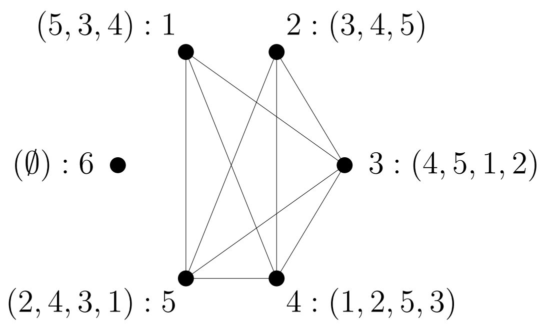

Figure 2.7: Phase 1 subgraph

Let $G^1 = (V^1, E^1, L^1)$ be the subgraph of G at the end of Phase 1. The following Lemma shows, in particular, that we can ignore any vertex that becomes isolated during Phase 1.

Lemma 2.2.2 (Irving [26])**.** *Let $G = (V, E, L)$ be a graph with preferences and let $xy \in E$. If $xy \notin E^1$ then xy is not an edge of any stable matching of G.*

Proof: Suppose, for a contradiction, that xy is an edge of some stable matching M, but $xy \notin E^1$. We may assume that xy was the first such edge deleted during Phase 1. Without loss of generality, xy was deleted when vertex z proposed to x (i.e. x prefers z to y). Now in M, x is matched to y and z is matched to some vertex w. However, we note that z cannot have a better stable partner than x since at the time of z's proposal to x, x was first on the preference list of z and xy was the first stable edge to be deleted. Thus z prefers x to w. So zx blocks M, contradicting the assumption that M was stable. □

If we are lucky and Phase 1 ends with a matching, it would be nonsensical to suggest that we need to do more work to reach a stable matching.

Lemma 2.2.3 (Irving [26])**.** *If every vertex in G^1 is adjacent to at most one other vertex then the set of edges E^1 form a stable matching in G.*

The proof is similar to the proof that the Gale-Shapley algorithm is correct and is therefore omitted. To make our lives easier for the remainder of this section, we will rely heavily on the following two properties.

Property 1. Vertex a is in the current preference list of vertex b if and only if b is in the current preference list of a.

Property 2. Vertex a is first in the current preference list of vertex b if and only if b is last in the current preference list of a.

Property 1 simply means that if we delete an edge ab, we must remember to remove b from a's preference list and remove a from b's preference list. Notice that Property 2 also holds in the Phase 1 subgraph, G^1: Suppose vertex b receives a proposal from a. The fact that a proposed to b tells us that b was first in a's preference list, otherwise a would have proposed to a vertex it preferred more than b. Vertex a becomes last in b's preference list because b deletes all of its neighbours it prefers less than a [26]. We will show that these two properties hold throughout the algorithm, and will be essential to the proof that the algorithm is correct.

We now come to Phase 2. Phase 2 deletes edges that are incident to a special set of vertices. Let H be a subgraph of G^1. Define $f_H(x)$, $s_H(x)$, and $l_H(x)$ to be the first, second, and last vertices of x's preference list in H. A *rotation* in H is a sequence of vertex pairs,

$$R = (x_0, y_0), (x_1, y_1) \dots, (x_{r-1}, y_{r-1}),$$

such that $f_H(x_i) = y_i$ and $s_H(x_i) = y_{i+1}$ for all i modulo r. In Figure 2.7 for example,

$$R = (1, 5), (2, 3), (3, 4)$$

is a rotation. If R is a rotation in a subgraph H, we will use $H \backslash R$ to represent the subgraph obtained by deleting the set of edges

$$E(R) := \{ay_i : x_{i-1} >_{y_i} a \text{ for all } i\},$$

along with the associated entries in the preference lists, from H; we will say that R has been *removed* from H. In our example, $R = (1, 5), (2, 3), (3, 4)$ is a rotation and we delete the following edges:

- 15 since $3 >_5 1$,
- 23 since $1 >_3 2$, and
- 34 and 45 since $2 >_4 5$ and $2 >_4 3$.

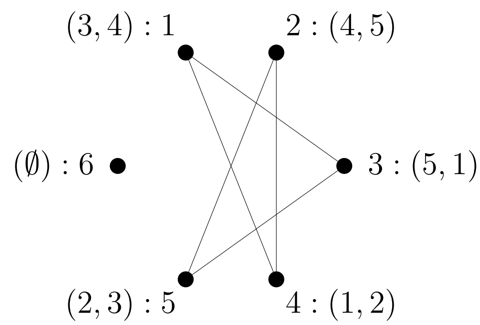

Figure 2.8: Removing a rotation

Figure 2.8 shows the result of removing R. Since we are only deleting edges, the fact that Property 1 holds throughout Phase 2 is obvious. Property 2 also holds in Phase 2: Let H be a subgraph of G^1 in which Properties 1 and 2 hold and let

$$R = (x_0, y_0), (x_1, y_1) \ldots, (x_{r-1}, y_{r-1})$$

be a rotation in H. When we remove R, we delete the edges $E(R)$. In particular, we delete all the edges of the form $x_i y_i$ because x_i is last in y_i's preference list. Notice also that we do NOT delete edges of the form $x_{i-1}y_i$. Therefore, y_i is now first in x_{i-1}'s preference list. By definition of $E(R)$, x_{i-1} is now also last in y_i's preference list. Deleting edges not of the form $x_i y_i$ does not affect Property 2, since they are not first nor last in any preference list of H. So, Property 2 still holds in $H \backslash R$, and rotation removal is well defined.

Lemma 2.2.4 (Irving [26])**.** *Let H be a Phase 2 subgraph of G^1. If H has a vertex with at least two neighbours then H has a rotation.*

Proof: We first note that if vertex a has only one neighbour, say b, then by Properties 1 and 2 the only neighbour of b is a. Let T be the set of vertices with at least two neighbours. Let $v \in T$. By the above observation, every neighbour of v is also in T. So for every $x \in T$, $s_H(x)$ and $l_H(s_H(x))$ exist and are in T. Furthermore, if $l_H(s_H(x)) = x$ then, by Property 2, we would have $f_H(x) = s_H(x)$, which is a contradiction since each neighbour of x can only appear once on x's preference list. Therefore, $l_H(s_H(x)) \neq x$.

Let D be a directed graph with nodes T and for every $x \in T$ an outward arc to $l_H(s_H(x))$. Since every node has out-degree 1, there exists a directed cycle C. Let $C =$

$x_0, x_1, \ldots, x_{r-1}$. Note that $x_{i+1} = l_H(s_H(x_i))$ for all i modulo r. By Property 2 this means that $s_H(x_i) = f(x_{i+1})$. Let $y_i = f(x_i)$ for all i modulo r and we can see that $R = (x_0, y_0), (x_1, y_1) \ldots, (x_{r-1}, y_{r-1})$ is a rotation. $\square$

Phase 2 can be described in the following way: While no vertex of V^1 is isolated and the remaining edges are not a matching, find a rotation, which must exist by Lemma 2.2.4, and remove it. If some some vertex becomes isolated then there is no stable matching. Otherwise the matching we are left with is stable.

Algorithm 3 Irving's Algorithm - Phase 2

Input: a graph with preferences $G = (V, E, L)$ and its phase 1 subgraph $G^1 = (V^1, E^1, L^1)$
Output: a stable matching M, or proof that there is no stable matching in G

```
let V' = {v ∈ V | deg_{G^1}(v) ≥ 1}
let H = G^1
while ∃ v ∈ V(H) such that deg_H(v) ≥ 2 and ∄ u ∈ V(H) such that deg_H(u) = 0 do
    find a rotation R in H
    H := H\R
end while
if ∃ u ∈ V(H) such that deg_H(u) = 0 then
    output that G has no stable matching
else
    output M := E(H), a stable matching
end if
```

The next two results show that if a graph with preferences has a stable matching, then at every iteration there will be at least one stable matching in the current subgraph of G^1. Unlike the Gale-Shapley algorithm, the stable matching found by Irving's algorithm is not unique; different sequences of rotation removals lead to different stable matchings.

Lemma 2.2.5 (Gusfield and Irving [19])**.** *Let H be a Phase 2 subgraph of G^1 and let M be a stable matching of G which is contained in H. If*

$$R = (x_0, y_0), (x_1, y_1) \ldots, (x_{r-1}, y_{r-1})$$

is a rotation in H and $x_i y_i \notin M$ for some i, then M is contained in $H \backslash R$.

Proof: Since rotations are cyclic we may assume $x_0 y_0 \notin M$. By the definition of a rotation, x_0 is therefore matched in M to, at best, y_1. This means that, since M is a stable matching, y_1 must be matched to a vertex it prefers at least as much as x_0; otherwise $x_0 y_1$ would be a blocking edge for M. Notice that by Property 2, x_1 is last in y_1's preference list in H. In particular, y_1 prefers x_0 to x_1. Therefore, we also have $x_1 y_1 \notin M$ since, as we just noticed,

y_1 must be matched to a vertex at least as good as x_0. Repeating this argument, we see that $x_iy_i \notin M$, x_i is matched in M to, at best, y_{i+1}, and y_i is matched in M to, at worst, x_{i-1} for all i modulo r. But when we remove R, we only delete an edge if some y_i prefers that edge less than y_ix_{i-1}. Thus, no edge of M is deleted when we remove R, showing that M is contained in $H\backslash R$. □

Theorem 2.2.6 (Irving [26], Gusfield and Irving [19])**.** *Let H be a subgraph of G^1 in Phase 2 and suppose that H contains a stable matching of G. If*

$$R = (x_0, y_0), (x_1, y_1) \dots, (x_{r-1}, y_{r-1})$$

is a rotation in H, then there is a stable matching contained in $H\backslash R$.

Proof: Let M be a stable matching of G and let $R = (x_0, y_0), (x_1, y_1) \dots, (x_{r-1}, y_{r-1})$ be a rotation in H. If there is an i such that $x_iy_i \notin M$, then Lemma 2.2.5 tells us that M is contained in $H\backslash R$. Thus, we may assume that $x_iy_i \in M$ for all i modulo r.

We first claim that $\{x_0, \dots, x_{r-1}\} \cap \{y_0, \dots, y_{r-1}\} = \emptyset$. Otherwise there are i and j such that $x_i = y_j$. Note that $x_iy_i, x_jy_j \in M$. But this can only happen if $x_j = y_i$ as well. Using Properties 1 and 2, we can say that

$$f_H(x_i) = y_i = x_j = l_H(y_j) = l_H(x_i).$$

Since the first vertex on the preference list of x_i is the same as the last, x_i has only one vertex in its preference list, contradicting our assumption that x_i was in the rotation R.

Now, let M' be the matching obtained from M by replacing the edges x_iy_i with x_iy_{i+1} for all i modulo r. Since R is cyclic and $\{x_0, \dots, x_{r-1}\} \cap \{y_0, \dots, y_{r-1}\} = \emptyset$, M' is indeed a matching. We also note that M' is contained in $H\backslash R$, since $f_{H\backslash R}(x_i) = y_{i+1}$ for all i and only edges containing y_i for some i can possibly be deleted when we remove R.

To complete the proof, we need only show that M' is stable. Suppose, for a contradiction, that the edge uv blocks M'. Since M is a stable matching, uv cannot block M. Notice that only the x_i's and y_i's have different matching partners in M' and only the x_i's prefer their partner in M to their partner in M'. So any blocking edge for M' must involve some x_j. Therefore, we may assume that $u = x_j$ for some j.

Claim: The blocking edge uv is an edge of $H\backslash R$.

Proof of Claim: Suppose, for a contradiction, that uv is not an edge of $H\backslash R$. Then uv was deleted when we removed R. Therefore $v = y_i$ for some i and y_i prefers $x_{i-1} = l_{H\backslash R}(y_i)$ to u. But this contradicts the assumption that uv blocks M' since $x_{i-1}yi \in M$ by assumption.

So $uv \in H\backslash R$ and, by above, $u = x_j$ for some j. By definition of M', u's partner in M' is first on its preference list in $H\backslash R$. Since $uv \in H\backslash R$, u must prefer $f_{H\backslash R}(u)$ to v, contradicting the assumption that uv blocks M'. □

The proof of Theorem 2.2.6 has a corollary. Specifically, this follows from the section of the proof where we show M' is a stable matching.

Corollary 2.2.7. *If, when Irving's algorithm terminates, every vertex of V^1 has degree 1, then the remaining edges are a stable matching of G.*

Let us prove that Irving's algorithm really works (finally!). Let $G = (V, E, L)$ be a graph with preferences. Recall that V^1 is the set of non-isolated vertices after Phase 1. First, suppose that G does not have a stable matching. By Corollary 2.2.7, Phase 2 cannot terminate with a perfect matching of G^1. Also, we continue to remove rotations while there is still a vertex of V^1 of degree at least 2. Therefore, there must exist a vertex of V^1 which became isolated during Phase 2 to cause the algorithm to stop.

Now suppose that G has a stable matching. Lemma 2.2.2 tells us that Phase 1 does not affect any stable matching of G. So, all the stable matchings of G survive to see Phase 2. Now, we keep removing rotations as long as there is a vertex of V^1 with degree at least 2. Since an isolated vertex of V^1 implies no stable matching, Phase 2 must terminate as a perfect matching of G^1. Theorem 2.2.6 tells us that as long as we are removing rotations, there will always be a stable matching in the remaining subgraph. Therefore, the remaining edges must be a stable matching of G.

Thus, Irving's algorithm correctly identifies whether a graph with preferences has a stable matching and, if it does, provides one for us.

Before we move on, we will briefly describe the running time of Irving's algorithm. Phase 1 runs in $O(n^2)$ time for the same reason that the running time of the Gale-Shapley algorithm is $O(n^2)$. In his original paper [26], Irving showed how to implement Phase 2 to run in $O(n^2)$ time. This gives us an overall time complexity of $O(n^2)$.

2.3 Consequences of Irving's Algorithm

In practice, we can use Irving's algorithm to decide if a graph with preferences has a stable matching. It is equally useful as a theoretical tool. We will start with a simple observation.

Lemma 2.3.1. *Let G be a graph with preferences. If, at the end of Phase 1, vertex v has an empty preference list then v is not matched in any stable matching of G.*

This follows directly from Lemma 2.2.2. Vertex v is not incident to any edges in G^1 so no edge incident to v in G can be in any stable matching.

Lemma 2.3.2. *Let G be a graph with preferences. If, at the end of Phase 1, vertex v has a nonempty preference list then v is matched in every stable matching of G.*

Proof: Let v be a vertex of G that has a nonempty preference list at the end of Phase 1. We first note that v must be first in the preference list of some vertex, say u. Otherwise, by the Pigeonhole principle, there is a vertex z such that z is first in the preference lists of both w and y. But, by Property 2, both w and y would then be last in the preference list of z. This is a contradiction since all the preference lists are total orders.

Now if v is unmatched in some stable matching M, then vu blocks M since $vu \notin M$ and u, by Lemma 2.2.2, is matched to a vertex it prefers less than v. Thus, v must be matched in every stable matching. $\square$

The next result follows easily from the previous two lemmas and is the non-bipartite version of Theorem 2.1.3 in Section 2.1.

Theorem 2.3.3 (Gusfield and Irving)**.** *Let G be a graph with preferences. If vertex v is unmatched in a particular stable matching of G, then v is unmatched in all stable matchings of G.*

We now know there exists a partition of the vertices: Vertices that are always matched in a stable matching and vertices that are never matched. Less technically, it does not matter which set of stable marriages we choose, the same set of people will never be married (possibly for the best). Once we have this partition, the consequence is inescapable. Yet, the conclusion still startles many people when they first hear it.

Corollary 2.3.4. *Let G be a graph with preferences. If G has a stable matching, then all stable matchings of G have the same size.*

In the case of graphs with poset preferences, Theorem 2.3.3 and Corollary 2.3.4 no longer hold. Consider Figure 2.9. We have a cycle with three vertices, and each vertex is indifferent to its neighbours.

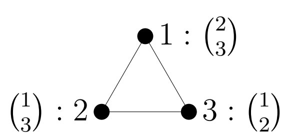

Figure 2.9: Example of a graph with poset preferences

One way to find a stable matching in a graph with poset preferences is to take a total extension of all the preference posets and then find a stable matching in the resulting graph with preferences. However, Figure 2.10 demonstrates a flaw in this approach.

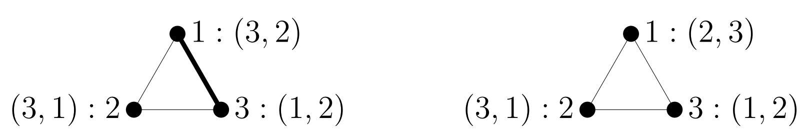

Figure 2.10: A problem with poset preferences

Both graphs have preference lists which are total extensions of their respective preference posets in the original graph. But the graph with preferences on the left has a stable matching, while the one on the right does not! When we take total extensions of the preference posets, a vertex with a partially ordered preference list will have to "break ties". As our example shows, different choices of tie-breaking yield different answers to the question "Does this graph with preferences have a stable matching?" This trait makes finding a stable matching in a non-bipartite graph with poset preferences NP-complete [40]. This is not a problem in a bipartite graph with poset preferences, since the resulting graph with preferences will always have a stable matching by Theorem 2.1.1. However, we still lose the result of Corollary 2.3.4.

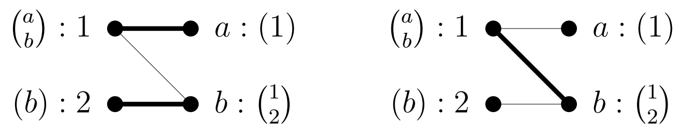

Figure 2.11: Poset preferences and stable matchings of different size

The graph with poset preferences in Figure 2.11 admits exactly two stable matchings, as shown. Notice that one of the stable matchings has size two and the other only has size one. Naturally, "What is the maximum (or minimum) size of a stable matching in a graph with poset preferences?" is the next question. Would it be possible to modify one of our algorithms to find a maximum cardinality stable matching? As with most stable matching problems involving poset preferences, this problem gets caught in the NP-completeness trap [34].

Chapter 3

Fractional Stable Matchings

As we noted in Section 2.2, a non-bipartite graph may not have any stable matchings. A possible way to avoid this unfortunate fact is to relax the "matching" requirement. In this chapter, we will be concerned with graphs with preferences. However, the preliminary definitions can be extended to hypergraphs with preferences and will be discussed in this context later in Chapter 5.

A *hypergraph with preferences*, $H = (V, E, L)$, is a hypergraph where every vertex has a totally ordered list of its incident edges. Let $H = (V, E, L)$ be a hypergraph with preferences and let M be a matching of H. If there is an edge $e \in E$ such that all the vertices of e prefer edge e to their respective edges of M, then we will call it a *blocking edge* for M. A matching, M, is stable if M does not have any blocking edges. In other words, if M is a stable matching for H, then for every $e \notin M$, at least one of the vertices of e prefers its matching edge to e.

A vector $x \in \mathbb{R}_+^E$ is called a *fractional matching* if

$$\sum_{\substack{e \in E: \\ u \in e}} x_e \leq 1 \text{ for every } u \in V.$$

A fractional matching x is called a *fractional stable matching* if every edge e contains a vertex u such that

$$\sum_{e \leq_u j} x_j = 1.$$

We will see that a hypergraph always admits a fractional stable matching regardless of the preferences of the vertices. In some sense we can think of a fractional stable matching as a stable arrangement of timeshared partnerships: Suppose that i and j are the endpoints of edge e. The value of x_e represents the proportion of time that i and j are roommates.

At least one of i and j will always have an additional situation that is preferred at least as much as the assignment of i to j. Admittedly, this is a curious metaphor since we described the original stable marriage problem as a way of preventing multiple partners. Fractional stable matchings in graphs with preferences will eventually lead us to a structure called a "stable partition". Stable partitions very closely resemble stable matchings and can be found using a simple extension of Irving's algorithm. In the process of our investigation we will see necessary and sufficient conditions for the existence of stable matchings in graphs with preferences.

3.1 Scarf's Lemma

Our starting point for fractional stable matchings will be a remarkable result of Scarf [46]. Originally, the result was proved in the context of game theory, but has since found many applications in graph theory (for examples, see [4, 21, 39, 46]). There are several seemingly unrelated topological versions of Scarf's result. However, the version that will be particularly useful for us involves matrices and allows for a connection to linear programming. To set up Scarf's Lemma we need a definition. For a real matrix C, a set of columns S of C is called *dominating* if for every column j of C, there exists a row i such that $c_{ij} \leq c_{ik}$ for every $k \in S$. As an example, consider the following matrix:

$$C = \begin{pmatrix} 2 & 1 & 7 & 4 & 5 & 2 \\ 8 & 2 & 2 & 3 & 6 & 7 \\ 1 & 3 & 2 & 2 & 1 & 4 \\ 4 & 4 & 8 & 1 & 5 & 5 \end{pmatrix}$$

With a bit of effort we can see that the first, third, fifth, and sixth columns form a dominating set of columns.

Theorem 3.1.1 (Scarf's Lemma [46])**.** *Let $m < n$ be positive integers, let $b \in \mathbb{R}^m_+$, and let B and C be $m \times n$ real matrices with the following properties:*

- *the first m columns of B form an identity matrix, and the set $\{x \in \mathbb{R}^n_+ : Bx = b\}$ is bounded,*
- *all entries in each row of C are distinct, and each entry c_{ik} for $k > m$ satisfies $c_{ii} < c_{ik} < c_{ij}$ for each $j \neq i$, $j \leq m$.*

Then there exists $x \in \mathbb{R}^n_+$ such that $Bx = b$ and the set of columns S of C indexed by the support of x, $supp(x) = \{k : x_k > 0\}$, form a dominating set.

The full version of Scarf's Lemma actually claims that the number of solutions to $Bx = b$ corresponding to dominating sets in C is odd. However, the stronger statement requires additional assumptions on the matrices B and C. We will use Scarf's Lemma to show the existence of a fractional stable matching in any hypergraph with preferences. Thus, the above version will be sufficient for our purposes. We refer the reader to the papers of Haxell [21], Rioux [39], or Scarf [46] for a full treatment.

Let $H = (V, E, L)$ be a hypergraph with preferences. The main obstacle that stands in the way of applying Scarf's Lemma is the link between stability and a dominating set of columns. To that end we make the following definitions. Recall the definition of v-rank from the introduction to Chapter 2. Let v be a vertex, let e be an edge such that $v \in e$ and let i be e's rank in the preference list of v. We define the *value of e in the preference list of v* to be $deg(v)-i+1$. For example, v's favourite edge would have value $deg(v)-1+1 = deg(v)$ and v's least favourite edge would have value $deg(v) - deg(v) + 1 = 1$.

We need to encode the preferences of H into a matrix C. The rows of our matrix C will be indexed by an arbitrary, but fixed, ordering of V. The columns will be indexed by $V \cup E$. The first $|V|$ columns will be indexed by the same ordering as the rows. The remaining $|E|$ columns will be indexed by some fixed ordering of E. Let $v \in V$. We first define all (v, e)-entries for $e \in E$: set the (v, e)-entry to be the value of e in the preference list of v if $v \in e$, and otherwise we assign to $\{(v, e) : v \notin e\}$ the values $\{deg(v) + 1, \ldots, |E|\}$ with an arbitrary permutation. Finally for the first $|V|$ entries of row v we let the (v, v)-entry be 0, and all other entries be distinct values strictly larger than $|E|$. We will call C the *value matrix of H*.

Theorem 3.1.2 (Aharoni and Fleiner [4]). *Every hypergraph with preferences has a fractional stable matching.*

Proof: Let $H = (V, E, L)$ be a hypergraph with preferences. Let C be the value matrix of H and let B be the vertex-edge incidence matrix of H with the identity matrix appended to its left. Both B and C have rows indexed by V and columns indexed by $V \cup E$. By construction, C satisfies the conditions of Theorem 3.1.1 and setting b as the all-1's vector allows B to meet the conditions of the theorem. Therefore, there exists $x \in \mathbb{R}_+^{V \cup E}$ satisfying $Bx = b$ and a set of columns S, indexed by $supp(x)$, which form a dominating set in matrix C. Let $\bar{x} \in \mathbb{R}_+^E$ be the entries of x corresponding to the edges of H. We claim that $\bar{x}$ is a fractional stable matching of H.

The definition of B and the fact that $Bx = b$ ensures that $\bar{x}$ is a fractional matching. To check that $\bar{x}$ is fractionally stable, let h be an arbitrary edge. Then in the matrix C there exists a vertex v such that all entries in row v of the columns indexed by $supp(x)$ are at least the (v, h)-entry. Note that the column v is not in $supp(x)$ as the (v, v)-entry is strictly smaller than every other entry in row v. Therefore $\sum_{e \in E : v \in e} \bar{x}_e = 1$. If $v \in h$, then by the way we defined C, we see that every edge e for which $v \in e$ and $\bar{x}_e > 0$ has a higher value in the preference list of v than h. Therefore $\sum_{h \leq_v e} \bar{x}_e = 1$ as required. To complete the proof we must show that the case where $v \notin h$ cannot happen. Suppose, for a contradiction, that $v \notin h$. By the definition of C, the (v,h)-entry is at least $deg(v)+1$. Thus, there is no edge e such that $v \in e$ and $\bar{x}_e > 0$. Since $Bx = b$ we must then have $v \in supp(x)$, contradicting our earlier finding. Therefore $\bar{x}$ is a fractional stable matching as required. □

Interestingly, the existence of a fractional stable matching extends to hypergraphs with poset preferences.

Corollary 3.1.3. *Every hypergraph with poset preferences has a fractional stable matching.*

Proof: Let $H = (V, E, P)$ be a hypergraph with poset preferences. For all $v \in V$ let P_v be the preference poset of v and let L_v be a total extension of P_v. Consider the auxiliary hypergraph with preferences $\bar{H} = (V, E, \bar{L})$ where $\bar{L}$ is the set of all L_v.

By Theorem 3.1.2, $\bar{H}$ has a fractional stable matching x. We claim that x is also a fractional stable matching for H. Clearly x is a fractional matching. Suppose, for a contradiction, that x is not a fractional stable matching for H. Then there exists an edge e such that $\sum_{e \leq_v h} x_h < 1$ for all $v \in e$. In particular, for every $v \in e$ there exists an edge f such that $f <_v e$ and $x_f > 0$. Since $f <_v e$ in P_v, we must have $f <_v e$ in L_v since L_v is a total extension of P_v. As x is a fractional matching, $\sum_{e \leq_v h} x_h < 1$ must hold in $\bar{H}$ as well. This contradicts the assumption that x was a fractional stable matching for $\bar{H}$. □

3.2 Stable Partitions

Stable partitions were introduced by Tan [48] to provide a characterization of graphs with preferences that have stable matchings. This characterization is known as Tan's Theorem. Tan's Theorem is similar in flavour to Tutte's Theorem about perfect matchings and Kuratowski's Theorem about planar graphs: The only impediment to a stable matching is a particular type of stable partition. We will prove this result later after we examine an algorithm to find stable partitions in a graph with preferences. In this section, we will introduce stable partitions and their connection to fractional stable matchings.

A cycle $C = \{v_0v_1, v_1v_2, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_0\}$ of a graph with preferences $G = (V, E, L)$ is a *preference cycle* if $v_{i-1}v_i <_{v_i} v_iv_{i+1}$ for all i modulo n.

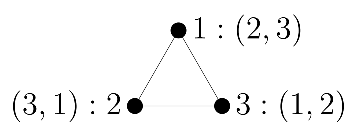

Figure 3.1: Example of a preference cycle

A *stable partition* of $G = (V, E, L)$ is a set of edges, $S \subseteq E$, with the following properties:

- Any component of S is either a cycle, a single edge, or an isolated vertex,
- each cycle component of S is a preference cycle, and
- for any $e \in E \backslash S$, there is a vertex v, incident with an edge of S, such that $v \in e$ and $e <_v f$ for any $f \in S$ with $v \in f$.

Notice that if S does not contain any cycles, then S is actually a stable matching. In the case that S contains a preference cycle component of odd length, we will say that S is an *odd stable partition*.

Theorem 3.2.1 (Tan [48])**.** *Every graph with preferences has a stable partition.*

Proof: Let $G = (V, E, L)$ be a graph with preferences. By Theorem 3.1.2 G has a fractional stable matching $x \in \mathbb{R}^E_+$. Let $S = supp(x)$. We will show that S is a stable partition. Since x is a fractional stable matching, we can orient each edge e towards a vertex v such that $\sum_{e \leq_v j} x_j = 1$.

Claim 1: Each vertex v is the head of, at most, one arc in S.

Proof of Claim 1: Suppose vertex v is the head of arcs $e, h \in S$. Without loss of generality, we may assume that $e <_v h$. Since v is the head of both arcs, we see that $\sum_{e \leq_v j} x_j = 1$ and $\sum_{h \leq_v j} x_j = 1$. But since $e <_v h$, we must have $x_e = 0$, contradicting that $e \in S$.

Claim 2: If e, f is a directed path and $e, f \in S$, then there is a $g \in S$ such that e, f, g is a directed path or a directed 3-cycle.

Proof of Claim 2: Let $w \in V$ be the tail of f (head of e). Since e is oriented towards w we have $e <_w f$. Let $z = head(f)$. Note that $x_f < 1$ as x is a fractional matching. Since x is also a fractional stable matching, there must be an edge $g \in S$ such that $tail(g) = z$ and $f <_z g$. If $head(g) = tail(e)$ then e, f, g is a directed 3-cycle. Otherwise e, f, g is a directed path.

Claim 3: Each vertex v is the tail of, at most, one arc in S.

Proof of Claim 3: Suppose vertex v is the tail of arcs $e, h \in S$. Let $z = head(e)$ and $w = head(h)$. Note that x_e and x_h are both strictly less than 1. By Claim 1 and our choice of edge orientations, there must exist arcs $f, k \in S$ such that $tail(f) = z$ and $tail(k) = w$. Repeatedly applying Claims 1 and 2 shows that there is a directed cycle C_1 containing the path e, f and another directed cycle C_2 containing the path h, k. Note that all the arcs of

C_1 and C_2 are in S. Since C_1 and C_2 share vertex v, there must be a vertex that is the head of two arcs of S, contradicting Claim 1.

Claims 1 and 3 imply that each vertex is incident to, at most, two edges of S. Claim 2 implies that any path of length at least two is actually part of a cycle of edges of S. These two properties show that any component of G induced by S is either an isolated vertex, a single edge or a cycle. Note that the proof of Claim 2 also implies that any cycle component is a preference cycle.

Finally, let $i \in E \backslash S$ and suppose that i is oriented as ab. Note that $x_i = 0$ and $\sum_{i \leq_b j} x_j = 1$. By definition of S and our choice of edge orientation, we must have $i <_b s$ for all s such that $b \in s$ and $s \in S$. Thus S is indeed a stable partition. $\square$

The proof presented here is due to Aharoni and Fleiner [4]. We will give another proof (similar to Tan's proof) of this result in the next section when we examine Tan's Algorithm and show that it will always find a stable partition.

A very nice consequence of the definition of stable partition and Theorem 3.2.1 is that once we have a stable partition, we automatically have a fractional stable matching. As we will see, this fractional stable matching has a very nice property: $x_e \in \{0, \frac{1}{2}, 1\}$ for every $e \in E$. These special fractional stable matchings are called *half-integral stable matchings.*

Theorem 3.2.2 (Tan [48], Aharoni and Fleiner [4])**.** *Every graph with preferences has a half-integral stable matching.*

Proof: Let $G = (V, E, L)$ be a graph with preferences. Theorem 3.2.1 ensures that G has a stable partition S. Recall that the components of S are either cycles or edges. For every $e \in E$ define

$$x_e = \begin{cases} 1 & \text{if } e \text{ is an edge component of } S \\ \frac{1}{2} & \text{if } e \text{ is an edge of a cycle component of } S \\ 0 & \text{otherwise.} \end{cases}$$

We claim that x is a half-integral stable matching. Note that x is certainly a fractional matching, while $\sum_{e \in E : v \in e} x_e = 1$ if v is incident to an edge of S and $\sum_{e \in E : v \in e} x_e = 0$ otherwise. First, let $e \notin S$. Since S is a stable partition, e has an endpoint v such that $e <_v f$ for every $f \in S$ such that $v \in f$. If v is in an edge component of S, then $x_f = 1$, implying that $\sum_{j \in E : e \leq_v j} x_j = 1$. If v is in a cycle component of S, then there are $g, h \in S$ incident with v such that $e <_v g$ and $e <_v h$. Since $x_g = x_h = \frac{1}{2}$, the result is $\sum_{j \in E : e \leq_v j} x_j = 1$.

Now, let $e \in S$. If e is in an edge component of S, then since x is a fractional matching and $x_e = 1$, we have $\sum_{j \in E : e \leq_v j} x_j = 1$ for some endpoint, v, of e. If e is in a cycle component, then since the cycle is a preference cycle, there exist a vertex u and an edge

f such that $u \in e$, $u \in f$ and $e <_u f$. Our definition of x ensures that $\sum_{j \in E: e \leq_u j} x_j = 1$. Thus x is, indeed, a half-integral stable matching. $\square$

Notice that if a cycle component C has an even number of edges, we could instead define the edge variables to alternate between 0 and 1 as we go around the cycle. Then we see that the above proof is still valid. Thus, a stable partition with no odd cycles induces a stable matching.

As with Corollary 3.1.3 in the previous section, consider a graph with poset preferences. Once again we can take a total extension of the preference poset for every vertex to obtain a half-integral stable matching in the auxiliary graph. Following the proof of Corollary 3.1.3 yields the analogous result.

Corollary 3.2.3. *Every graph with poset preferences has a half-integral stable matching.*

3.2.1 Tan's Algorithm

Given that a graph with preferences will always have a stable partition, it would be ideal to have an algorithm to find such a set of edges. Tan gave such a procedure in 1991 [48]. The algorithm is essentially the same as Irving's algorithm. In fact, Phase 1 is identical. The only difference will be our treatment of isolated vertices in Phase 2. Instead of stopping when we find an isolated vertex, we will set aside the isolated vertices and continue the algorithm on the non-isolated vertices. As a consequence, we will see that Tan's algorithm provides us with a characterization of graphs with preferences that do not have a stable matching.

We recommend that the reader be familiar with the definitions and results of Section 2.2.1. For convenience, we will give a short review of how the Phase 1 algorithm works. Recall that a vertex x is "free" if there is no vertex z currently holding a proposal from x. Free vertices propose to their favourite remaining neighbour until every vertex is either isolated or not free. If x proposes to vertex v, we delete all the edges incident to v that v prefers less than xv. Note that if v were holding a proposal from z then z would become free after x's proposal to v because v deleted all the edges it preferred less than vz; so, we must have $x >_v z$ in order for x to propose to y. Phase 1 continues as long as there is a free vertex x that is not isolated.

Exactly as in the case of stable matchings, the Phase 1 subgraph contains all stable partitions.

Lemma 3.2.4 (Tan [48])**.** *Let $G = (V, E, L)$ be a graph with preferences and let $xy \in E$. If, at the end of Phase 1, $xy \notin E^1$ then xy is not an edge of any stable partition of G.*

Proof: Suppose, for a contradiction, that xy is an edge of some stable partition S, but $xy \notin E^1$. We may assume that xy was the first such edge deleted during Phase 1. Without loss of generality, xy was deleted when vertex z proposed to x (i.e. $z >_x y$). Now in S, x is incident to y and let w be the most preferred vertex adjacent to z. We note that z cannot be incident to a better vertex in S than x since at the time of z's proposal, x was first on the preference list of z and xy was the first edge of S to be deleted. Thus $x \geq_z w$.

If $x >_z w$, then $xz \notin S$ by definition of w. But this means that xz blocks S since $z >_x y$, contradicting that S is a stable partition. Otherwise, $x = w$. So $deg_S(x) = 2$ as $xy, xz \in S$. By definition of stable partition, the component containing the edges xy and xz is a preference cycle. So there must exist a vertex u, possibly y, such that $zu \in S$. However, $w = x >_z u$ and $z >_x y$, contradicting that the component is a preference cycle and that S was a stable partition. □

Algorithm 4 Tan's Algorithm - Phase 1

Input: a graph with preferences $G = (V, E, L)$
Output: the phase 1 subgraph $G^1 = (V^1, E^1, L^1)$
  set $L^1(x) := L_x \;\forall\; x \in V$ and $E^1 := E$
  while $\exists\; x \in V$ such that x is free and $L^1(x) \neq \emptyset$ **do**
    let y be the most preferred vertex of $L^1(x)$
    if y is holding a proposal from z **then**
      assign z to be free
    end if
    $E^1 := E^1 \backslash \{ay : x >_y a\}$
    $L^1(y) := L^1(y) \backslash \{a : x >_y a\}$
    for all a such that $x >_y a$ **do**
      $L^1(a) := L^1(a) \backslash y$
    end for
  end while
  $V^1 := \{v \in V : deg(v) \geq 1\}$
  output $G^1 = (V^1, E^1, L^1)$, the phase 1 subgraph

Once again we can ignore any isolated vertices at this point. By Lemma 3.2.4 these vertices will be isolated in any stable partition. We now come to Phase 2 where we further delete edges via removal of rotations.

Recall the definition of a "rotation" from Section 2.2.1 and that $E(R)$ is the set of edges that are deleted when we remove the rotation R. The essential ingredient in Tan's Algorithm is the observation that if a vertex becomes isolated after eliminating a rotation R, then $E(R)$ can only be a very special set of edges.

Lemma 3.2.5 (Tan [47])**.** *Let H be a subgraph of G^1 in Phase 2 and let*

$$R = (x_0, y_0), (x_1, y_1) \ldots, (x_{r-1}, y_{r-1})$$

be a rotation in H. If, after removing R from H, some vertex becomes isolated, then $E(R)$ is exactly an odd preference cycle.

To prevent this thesis from being thrown across the room in frustration, we will omit the extremely technical details of this result and refer the interested (and brave) reader to the papers of Tan [47, 48]. Nonetheless, here is a sketch of the proof: By definition of rotation removal we can assume that x_0 is an isolated vertex. We now make the observation that to delete the edge x_0y_0, we must have $y_0 = x_j$ and $x_0 = y_{j+1}$ for some $j \neq 0$. Then, using Properties 1 and 2 from Section 2.2.1, we can show that $deg_H(x_j) = 2$. Finally, we apply induction to show that the same properties hold for all indices i modulo r and that r is indeed odd. Hence, $E(R)$ is an odd cycle. The fact that $E(R)$ is a preference cycle follows since R is a rotation.

As an illustration, consider the graph with preferences in Figure 2.6. Figure 2.8 showed the state of the graph after Phase 1 and the removal of a rotation. Notice that

$$R = (1, 3), (2, 4), (3, 5), (4, 1), (5, 2)$$

is now a rotation and, if we remove it, the graph has no edges left. By inspection $E(R)$ is an odd preference cycle.

During Phase 2 of Irving's Algorithm, we stopped if we found an isolated vertex because there was no stable matching. Tan shows us that the absence of a stable matching implies the presence of an odd preference cycle. This preference cycle will turn out to be an odd component of a stable partition.

We can describe Phase 2 in the following way: While there is an active vertex of degree at least 2, Lemma 2.2.4 tells us that there is a rotation. Therefore, we find a rotation and remove it. If removing a rotation R causes a vertex to become isolated, add $E(R)$ to S and continue Phase 2 on the non-isolated vertices. If all remaining vertices have degree 1, add the remaining edges to S. Then, S will be a stable partition.

We are now ready to prove that Tan's algorithm is indeed correct.

Theorem 3.2.6 (Tan [48])**.** *For any graph with preferences, Tan's algorithm outputs a stable partition.*

Proof: Let S be the set of edges given by Tan's algorithm. Notice that, by Lemmas 3.2.4 and 3.2.5, the edges of S form components which are either an isolated vertex, a single edge, or a preference cycle. Thus, we need only show that S satisfies the third condition of a stable partition. Suppose, for a contradiction, that S is not a stable partition. Then there exists an edge $ab \notin S$ and edges $aw, bz \in S$ such that $b >_a w$ and $a >_b z$.

If ab was deleted during Phase 1, then without loss of generality, there is a vertex u that proposed to vertex a such that $u >_a b$. But this contradicts the fact that edge $aw \in S$, since if such a u existed, $u >_a b >_a w$ which would have caused the edge aw to be deleted as well.

If ab was deleted during Phase 2, then as noted in the proof of Theorem 2.2.6, we may assume that $b = y_i$ for some y_i in the removed rotation

$$R = (x_0, y_0), (x_1, y_1) \dots, (x_{r-1}, y_{r-1}).$$

Since ab was deleted, we must have $x_{i-1} >_b a$. By definition of rotation removal we also see that edge bz was deleted, since $x_{i-1} >_b a >_b z$. Again this contradicts our assumption that $bz \in S$. Thus S must be a stable partition. □

Algorithm 5 Tan's Algorithm - Phase 2

Input: a graph with preferences $G = (V, E, L)$ and its phase 1 subgraph $G^1 = (V^1, E^1, L^1)$
Output: a stable partition S

```
let H := G^1, and S = ∅
while ∃ v ∈ V(H) such that deg_H(v) ≥ 2 do
   find a rotation R in H
   J := H\R
   if ∃ u ∈ V(J) such that deg_J(u) = 0 then
      S := S ∪ E(R)
      V' := V(H)\{v ∈ V(H) : deg_J(v) = 0}
   end if
   E' := E(H)\E(R) and L' := L(H)\L(R)
   H := (V', E', L')
end while
S := S ∪ E(H)
output S, a stable partition
```

This procedure is almost identical to Irving's Algorithm. The only difference comes when we find an isolated vertex in Phase 2. Instead of stopping, we store $E(R)$ and continue removing rotations as if the algorithm did not find anything at all. So, it is not a surprising fact that the running time of Tan's Algorithm remains $O(n^2)$ [48].

3.2.2 A Consequence of Tan's Algorithm

Section 2.3 shows how many interesting properties of stable matchings in bipartite graphs actually extend to the non-bipartite case. This section will look at an implication of Tan's Algorithm and the existence of stable partitions. In particular, we will finally give a proof of Tan's Theorem.

Theorem 3.2.7 (Tan's Theorem [47]). *A graph with preferences has a stable matching if and only if it does not have an odd stable partition.*

Given that all bipartite graphs with preferences have a stable matching, this result should not come as a huge surprise. As we saw in Theorem 1.1.1, any non-bipartite graph necessarily has an odd cycle. Therefore, it is reasonable to suspect that the non-existence of a stable matching in a graph with preferences must have something to do with an odd cycle. Tan's Theorem simply confirms our suspicions.

The reader who is still awake at this point will realize that Tan's characterization of graphs with preferences admitting stable matchings is missing only one piece.

Lemma 3.2.8 (Tan [48]). *If G is a graph with preferences and has an odd stable partition, then G has no stable matching.*

Proof: Suppose, for a contradiction, that G has an odd stable partition, S, and a stable matching M. Let A be the set of vertices who have a partner in M that they prefer more than at least one of their neighbours in S. Let B be the set of vertices who have a partner in M that they prefer less than all of their neighbours in S. Every vertex in A must be matched in M to a vertex in B. Otherwise, if $ab \in M$, $a \in A$ and $b \notin B$ then both a and b would prefer the edge ab to at least one of their partners in S which would contradict that S is a stable partition. So we must have $|A| \leq |B|$.

Now let $xy \in S$. If both x and y are in B, then both x and y prefer the edge xy to their matching partners, contradicting that M is a stable matching. Let C be the vertices of a component of S. Since each component of S is either a single vertex, a single edge or a cycle, and no two adjacent edges are in B, we must have $|A \cap C| \geq |B \cap C|$. If C has an odd number of vertices then the inequality is strict. Let $\mathscr{C}$ be the set of components of S.

$$|A| = \sum_{C \in \mathscr{C}} |A \cap V(C)| > \sum_{C \in \mathscr{C}} |B \cap V(C)| = |B|.$$

The strict inequality comes from the fact that S is an odd stable partition and must therefore have at least one odd component. Thus, we have a contradiction which gives us the result. □

In summary, if $G = (V, E, L)$ does not have a stable matching, then by Phase 2 of Irving/Tan's algorithm and Lemma 3.2.5, G must have an odd stable partition. Conversely, if G has an odd stable partition, then G has no stable matching by Lemma 3.2.8. This is exactly the characterization given by Tan.

3.3 Linear Programming

We now return to fractional stable matchings. An unfortunate consequence of using Scarf's Lemma to prove the existence of a fractional stable matching is that there is no intuition about the set of all fractional stable matchings. Scarf's Lemma simply tells us that there is always a fractional stable matching. In point of fact, the set is not convex! Consider the graph with preferences in Figure 3.2.

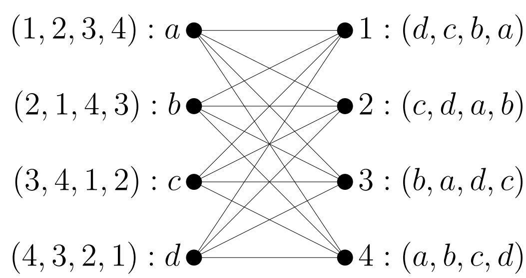

Figure 3.2: Example of non-convex fractional stable matchings

From Figure 2.4, we can check that $M_1 = \{a2, b1, c4, d3\}$ and $M_2 = \{a4, b3, c2, d1\}$ are stable matchings. Any convex combination of these two stable matchings will be of the form:

$$x_e = \begin{cases} \lambda & \text{if } e \in M_1 \\ 1 - \lambda & \text{if } e \in M_2 \\ 0 & \text{otherwise.} \end{cases}$$

where $0 \leq \lambda \leq 1$. We then find that $\sum_{a3 \leq_a j} x_j = \lambda$ and $\sum_{a3 \leq_3 j} x_j = 1 - \lambda$ for the edge $a3$. Both λ and $1 - \lambda$ are strictly less than 1 when $0 < \lambda < 1$. Thus any strict convex combination of M_1 and M_2 is not a fractional stable matching.

We will examine a linear program in which the feasible set will contain all fractional stable matchings. This added structure will allow us to make observations about these feasible solutions and, hence, the set of fractional stable matchings. The proofs of the results in the next section will be left until Chapter 4, where we will consider a generalization to 3-uniform hypergraphs.

3.3.1 Stable Matchings and Linear Programming

The first linear programming formulation for stable matchings was given by Vande Vate in 1989 [51]. However, his formulation was only for stable matching problems where the underlying graph was the complete bipartite graph, $K_{m,n}$. Roth, Rothblum, and Vande Vate [42] extended Vande Vate's work to arbitrary bipartite graphs with preferences in 1993. In 1994, Abeledo and Rothblum [3] showed that many interesting properties of the bipartite formulation easily extend to the non-bipartite case.

Recall that $x \in \mathbb{R}_+^E$ is a fractional stable matching for a graph with preferences if it is a fractional matching and for every edge ij either $\sum_{a \geq_i j} x_{ia} = 1$, or $\sum_{b \geq_j i} x_{jb} = 1$. Observe that any fractional stable matching is a feasible solution to the following linear program:

$$\begin{array}{rrcll} \textbf{max} & \displaystyle\sum_{ij \in E} x_{ij} & & & (\mathrm{P}_{SM}) \\ \textbf{subject to:} & \displaystyle\sum_{j \in N(i)} x_{ij} & \leq & 1 & i \in V \\ & \displaystyle\sum_{a >_i j} x_{ia} + \sum_{b >_j i} x_{jb} + x_{ij} & \geq & 1 & ij \in E \\ & x_{ij} & \geq & 0 & ij \in E. \end{array}$$

Suppose we take another look at the example in Figure 3.2. Notice that, since M_1 and M_2 are stable matchings, the vectors

$$x_e = \begin{cases} 1 & \text{if } e \in M_1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\bar{x}_e = \begin{cases} 1 & \text{if } e \in M_2 \\ 0 & \text{otherwise} \end{cases}$$

are feasible for (P_{SM}). Since the feasible region of any linear program is convex, any convex combination of x and $\bar{x}$ is also a feasible solution to (P_{SM}). But we showed that any strict convex combination of x and $\bar{x}$ is not a fractional stable matching. Therefore, the feasible solutions of (P_{SM}) are not necessarily fractional stable matchings. However, this does not make (P_{SM}) useless! Since we now know that every graph with preferences has a fractional stable matching, Theorem 3.1.2 implies that the feasible set of (P_{SM}) is always non-empty. As we will see, this linear program has some very useful properties which can be transferred directly to the set of fractional stable matchings.

Lemma 3.3.1 (Roth, Rothblum, and Vande Vate [42], Abeledo and Rothblum [3])**.** *Let G be a graph with preferences. The incidence vectors of stable matchings of G are precisely the feasible integer solutions to* (P_{SM})*.*

It is well known that integer programming is a hard problem in general [32]. A natural question to ask is: Can we add any other constraints to (P_{SM}) and obtain the convex hull of the incidence vectors of stable matchings? Or in other words: Is there a linear programming formulation for stable matchings such that the extreme points of the feasible solutions are exactly the stable matchings? Such a formulation would allow us to solve any linear optimization problem over the set of stable matchings using a suitable linear programming algorithm. Sadly, this is not the case.

Theorem 3.3.2 (Feder [11])**.** *Let G be a graph with preferences. It is NP-hard to optimize a linear function over the set of stable matchings of G.*

We should point out that Feder's result does not mean that the extreme point solutions to (P_{SM}) can be arbitrary.

Theorem 3.3.3 (Abeledo and Rothblum [3])**.** *Let G be a graph with preferences and let S be the set of feasible solutions to* (P_{SM})*. Then the extreme points of S are half-integral.*

By the above comments, an ideal formulation would have extreme points with integral coordinates. It turns out (not surprisingly) that by restricting our focus to bipartite graphs with preferences, (P_{SM}) is such a formulation.

Theorem 3.3.4 (Vande Vate [51], Rothblum [45])**.** *Let G be a bipartite graph with preference lists and let S be the set of feasible solutions to* (P_{SM})*. Then the extreme points of S are exactly the incidence vectors of stable matchings of G.*

In the bipartite case, we now have that the feasible set of (P_{SM}) is a non-empty polytope and its extreme points are always integral. This leads us to another proof of The Fundamental Theorem of Stable Matchings from Section 2.1.

As mentioned earlier, duality and complementary slackness play a leading role in the linear programming perspective of stable matchings. Obviously we need a dual linear program to fully utilize these tools. The dual of (P_{SM}) is the following:

$$\begin{array}{rrcll}
\textbf{min} \displaystyle\sum_{i\in V} y_i - \sum_{ij\in E} z_{ij} & & & & (\mathrm{D}_{SM}) \\
\textbf{subject to:} & y_i & \geq & 0 & i \in V \\
& z_{ij} & \geq & 0 & ij \in E \\
\displaystyle y_i + y_j - \sum_{j>_i a} z_{ia} - \sum_{i>_j b} z_{jb} - z_{ij} & & \geq & 1 & ij \in E.
\end{array}$$

Although it is worthwhile to work through the details to obtain (D_{SM}) (and would probably be an excellent assignment question in a linear programming course), we will leave this as an exercise for the reader.

Lemma 3.3.5 (Abeledo and Rothblum [3])**.** *Let $\bar{x}$ be a feasible solution to* (P_{SM}) *and let $\bar{y} \in \mathbb{R}^V$ and $\bar{z} \in \mathbb{R}^E$ be defined as follows:*

$$\begin{aligned} \bar{y}_i &= \sum_{j \in N(i)} \bar{x}_{ij} \quad i \in V, \\ \bar{z}_{ij} &= \bar{x}_{ij} \quad ij \in E. \end{aligned}$$

Then $(\bar{y}, \bar{z})$ is a feasible solution to (D_{SM})*.*

Theorem 3.3.6 (Roth, Rothblum, and Vande Vate [42], Abeledo and Rothblum [3])**.** *Let $\bar{x}$ be a feasible solution to* (P_{SM})*. Then $\bar{x}$ is also an optimal solution to* (P_{SM})*.*

It is also true that the $(\bar{y}, \bar{z})$ defined in Lemma 3.3.5 is optimal for (D_{SM}). This will turn out to be very important when we look at the proofs of these results. The following corollary is now immediate.

Corollary 3.3.7. *Let G be a graph with preferences. If G has a stable matching, then all stable matchings of G have the same size.*

This is exactly the conclusion of Corollary 2.3.4. Additionally, Theorem 2.3.3 is a trivial consequence of the following result.

Theorem 3.3.8 (Roth, Rothblum, and Vande Vate [42], Abeledo and Rothblum [3])**.** *Let $G = (V, E, L)$ be a graph with preferences. There exists a partition (A, B) of V such that for every feasible solution $\bar{x}$ to* (P_{SM})*,*

$$\sum_{j \in N(i)} \bar{x}_{ij} = 1 \quad \text{if } i \in A,$$

$$\sum_{j \in N(i)} \bar{x}_{ij} = 0 \quad \text{if } i \in B.$$

This partition of the vertices can even give us a sufficient condition for the non-existence of a stable matching.

Corollary 3.3.9 (Abeledo and Rothblum [3])**.** *Let G be a graph with preferences. If $|A|$ is odd then G does not have a stable matching.*

The main point here is that linear programming can provide alternate proofs to many of the results in Chapter 2 without relying on the properties of an algorithm. Furthermore, these properties extend to the fractional case since fractional stable matchings are feasible solutions to (P_{SM}). We will see this approach again, along with extensions of many of the proofs, in Section 4.2.

Chapter 4

3-Uniform Hypergraphs

Let us return to our time travelling adventure from Chapter 1. However, against the advice of many science fiction enthusiasts, we are going to allow some small changes to the past.

First, suppose that your university has several options for residence and you decide you would like to live in an apartment. This means that, instead of having a single roommate, you will have several roommates. We will assume that each apartment will have three occupants. The school is faced with essentially the same problem as before: Can it assign students to apartments so that there is no group of three students who would rather live together than with their assigned roommates?

The other change we will make comes when you get married: Your future spouse has agreed to marry you only if you agree to having a dog. Now, instead of only finding a spouse, you have to worry about finding a spouse AND a dog. We will make the assumption that the dogs know what kinds of dog food they will be fed by each possible couple, so that they will be able to rank their possible owners. Once again, the goal is to avoid man-woman-dog triplets that prefer each other to their respective families.

Our motivational problems from Chapter 1 have the following hypergraphic cousins:

Problem 4 (Threesome Roommates Problem)**.** Suppose there are n people living in a university dormitory. Each person ranks the other $\binom{n-1}{2}$ pairs of students in terms of who they would prefer to have as roommates. Can we find an assignment of roommates such that if three students are not roommates then at least one of them prefers their current roommates?

Problem 5 (Stable Family Problem)**.** A community consists of n men, n women, and n dogs. Each man ranks every woman-dog pair in terms of who they would prefer for a family. Similar rankings are made by each woman and dog. Can we find a set of families such that if a man, a woman, and a dog are not a family then at least one of them prefers their current family?

Given that a matching is now a group of three, graphs with preferences are no longer sufficient to model Problems 4 and 5. We turn our attention to stable matchings in hypergraphs. In particular, we will model the threesome roommates and stable family problems with 3-uniform hypergraphs and tripartite 3-uniform hypergraphs, respectively. Recall the definition of a hypergraphic stable matching from Chapter 3.

Unlike a graph, representing a hypergraph can be very messy for all but the smallest of cases. We will display hypergraphs in a table, similar to Table 4.1.

1	123	124	134
2	123	124	234
3	234	134	123
4	134	124	234

Table 4.1: Example of a hypergraph

The first entry of a row represents the name of the vertex. The remaining entries of row v have two purposes: To tell us which edges are incident to v and to display the preference list of v, read from left to right.

Regrettably, the similarities between graphs and hypergraphs end at the definition. In 1991, Hirschberg and Ng showed, through a reduction from *3-dimensional matching*, that the problem of deciding if an instance of the stable family problem has a solution is NP-complete [37]. The situation only gets worse!

4.1 Difficulties with Hypergraph Stable Matchings

As we saw in Chapter 2, stable matchings in a graph with preferences exhibit very strong, and extremely useful, properties:

- All stable matchings have the same size,
- a vertex is matched either in every stable matching or in no stable matchings,
- we have efficient algorithms to find a stable matching or tell us that one does not exist, and
- if there is no stable matching, we can still find a half-integral stable matching.

Is it possible that these properties extend to 3-uniform hypergraphs? We already know that Problem 5 is NP-complete, so efficient algorithms are unlikely. But what about the other three properties? Let us look at some examples.

1	123	156
2	234	123
3	123	234
4	234	
5	156	
6	156	

Table 4.2: Stable matchings of different size

Table 4.2 gives a hypergraph with vertex set $\{1, 2, 3, 4, 5, 6\}$ and edge set $\{123, 156, 234\}$. Notice that $M_1 = \{156, 234\}$ is a stable matching since $123 \notin M_1$ and $234 >_2 123$. Furthermore, $M_2 = \{123\}$ is also a stable matching since $156, 234 \notin M_2$, $123 >_1 156$ and $123 >_3 234$. Clearly, M_1 and M_2 do not have the same size.

Here is another example:

1	123	124	134
2	123	124	234
3	234	134	123
4	134	124	234

Table 4.3: No vertex partition

This hypergraph has vertices $\{1, 2, 3, 4\}$ and edges $\{123, 124, 134, 234\}$. We can also see that $M_3 = \{123\}$ and $M_4 = \{134\}$ are stable matchings. However, vertices 2 and 4 are both matched in only one of M_3 and M_4: We have lost the very nice property that if a vertex is matched in a stable matching, then it is matched in all stable matchings.

Even if we want to talk about fractional stable matchings in hypergraphs we can run into problems:

1	123	124	134
2	234	123	124
3	134	234	123
4	124	134	234

Table 4.4: No half-integral stable matching

We will first show that this hypergraph does not have a stable matching. Consider the matching $M = \{123\}$. Notice that 234 is a blocking edge for M since $234 >_2 123$, $234 >_3 123$, and vertex 4 is unmatched. Hence, M is not stable. A similar argument shows that each of the other three maximal matchings are not stable. Therefore, the example has no stable matchings.

What can we say about its fractional stable matchings? Recall that a fractional matching x is called a *fractional stable matching* if every edge e contains a vertex u such that

$$\sum_{e \leq_u j} x_j = 1.$$

A graph may or may not have a stable matching, but we know that it will always have a fractional stable matching where every edge takes a value in the set $\{0, \frac{1}{2}, 1\}$.

Suppose that this hypergraph has a half-integral stable matching. Since it does not have a stable matching, some edge must have a value of $\frac{1}{2}$. The preference lists have a cyclic structure, so we may assume that $x_{123} = \frac{1}{2}$. The definition of a fractional stable matching tells us that since $x_{123} = \frac{1}{2}$, we must have either $x_{234} = \frac{1}{2}$ or $x_{134} = \frac{1}{2}$. If the former is true, then the same argument shows that either $x_{134} = \frac{1}{2}$ or $x_{124} = \frac{1}{2}$. Now, $x_{134} \neq \frac{1}{2}$ since the sum of the edges at vertex 3 would be $\frac{3}{2}$ and not a matching. But, $x_{124} \neq \frac{1}{2}$ because vertex 2 runs into the same problem. Thus, it is not possible to have $x_{234} = \frac{1}{2}$. The $x_{134} = \frac{1}{2}$ case is very similar and is left to the reader. There is no half-integral stable matching. In fact, the best fractional stable matching can be found by giving each edge a value of $\frac{1}{3}$.

The point of these examples is this: All of our properties can fail to exist when we move to hypergraphs! We must hope there is a special case of stable matchings that will yield these properties.

4.2 Majority Stable Matchings

Let H be a 3-uniform hypergraph with preferences and let M be a matching of H. Recall that M is a stable matching if for every edge $ijk \notin M$, at least one of i,j, or k prefers its matching edge to ijk. If every $ijk \notin M$ has the stronger property that at least TWO of i, j, and k prefer their matching edge to ijk, then we will say that M is a *majority stable matching*. Once more, we can alternatively define a majority stable matching in terms of blocking edges. In this case, a blocking edge for a matching M is an edge $ijk \notin M$ such that at least two of i, j, and k prefer the edge ijk to their respective matching edges; a matching M is a majority stable matching if it does not have any of these blocking edges.

There has been much research into the varying "strength" of a stable matching in a hypergraph with poset preferences [6, 23, 24, 27]. *Weak*, *strong*, *super*, and *ultra* stable matchings form a hierarchy based on the types of blocking edges they exclude [23]. However, all four of these stable matchings coincide with our notion of stable matching when the preference lists are totally ordered. In addition to bringing the desirable properties of stable matchings from graphs to hypergraphs, our definition of majority stable matchings was made to study the strengths of stable matchings when the preference lists are totally ordered.

Consider the hypergraph in Table 4.5. There are two stable matchings: $M_5 = \{123, 456\}$ and $M_6 = \{156, 234\}$. We claim that M_5 is a majority stable matching: Notice that 156 and 234 are not edges of M_5, $123 >_1 156$, $123 >_2 234$, $456 >_4 234$, and $456 >_5 156$. Indeed, majority stable matchings can exist.

1	123	156
2	123	234
3	234	123
4	456	234
5	456	156
6	156	456

Table 4.5: Majority stable matching example

As promised, we will proceed by extending many of the linear programming results of Section 3.3. To do this, we will need a suitable linear program:

$$\begin{aligned} \textbf{max} \sum_{ijk \in E} x_{ijk} & & & \qquad (\mathrm{P}_{MSM}) & (4.1) \\ \textbf{subject to:} \quad \sum_{\substack{uv: \\ iuv \in E}} x_{iuv} & \leq & 1 & \quad i \in V & (4.2) \\ \sum_{ab >_i jk} x_{iab} + \sum_{cd >_j ik} x_{jcd} + \sum_{ef >_k ij} x_{kef} + 2x_{ijk} & \geq & 2 & \quad ijk \in E & (4.3) \\ x_{ijk} & \geq & 0 & \quad ijk \in E. & (4.4) \end{aligned}$$

Lemma 4.2.1. *Let H be a 3-uniform hypergraph with preferences. The incidence vectors of the majority stable matchings of H are precisely the feasible integral solutions to* (P_{MSM}).

Proof: It is not hard to see that an integral vector satisfies (4.2) and (4.4) if and only if it is the incidence vector of a matching. We note that an integral vector violates (4.3) if and only if one of the following happens:

- $x_{ijk} = \sum_{ab >_i jk} x_{iab} = \sum_{cd >_j ik} x_{jcd} = \sum_{ef >_k ij} x_{kef} = 0$, i.e. all of i, j, and k are matched in an edge they prefer less than ijk, or
- $x_{ijk} = 0$ and exactly one of $\sum_{ab >_i jk} x_{iab}$, $\sum_{cd >_j ik} x_{jcd}$, or $\sum_{ef >_k ij} x_{kef}$ equals 1, i.e. only one of i, j, or k is matched in an edge it prefers more than ijk.

Neither case meets the definition of a majority stable matching. Thus, a vector x is the incidence vector of a majority stable matching if and only if it is an integral solution to (4.2)-(4.4). $\square$

Since majority stable matchings are still stable matchings, a 3-uniform hypergraph with preferences may fail to have a majority stable matching. But there is an additional concern here. Consider Table 4.6.

1	124	
2	234	124
3	234	
4	124	234

Table 4.6: Feasible set of (P_{MSM}) can be empty

Both $M = \{123\}$ and $\bar{M} = \{234\}$ are stable matchings. The set of majority stable matchings can be expressed as the integral solutions to the following linear program:

$$\begin{aligned} \textbf{max} \quad & x_{124} + x_{234} & & & (4.5) \\ \textbf{subject to:} \quad & x_{124} & \leq & \ 1 & (4.6) \\ & x_{234} & \leq & \ 1 & (4.7) \\ & x_{124} + x_{234} & \leq & \ 1 & (4.8) \\ & x_{234} + 2x_{124} & \geq & \ 2 & (4.9) \\ & x_{124} + 2x_{234} & \geq & \ 2 & (4.10) \\ & x_{124}, x_{234} & \geq & \ 0. & (4.11) \end{aligned}$$

Notice that, if we add the constraints (4.9) and (4.10), we obtain the valid inequality

$$x_{124} + x_{234} \geq \frac{4}{3},$$

which clearly contradicts constraint (4.8). So, not only does the example in Table 4.6 have no majority stable matchings, but the set of feasible solutions to (P_{MSM}) is empty! Recall that (P_{SM}) is always non-empty, even if the corresponding graph with preferences has no stable matching.

The dual of (P_{MSM}) is once again analogous to (D_{SM}) in Section 3.3. The linear program (D_{MSM}) will allow us to use the powerful ideas of duality and complementary slackness to obtain results that are analogous to the results in Section 3.3:

$$
\begin{array}{rcll}
\multicolumn{4}{r}{\mathbf{min} \ \sum_{i\in V} y_i - 2\sum_{ijk\in E} z_{ijk} \qquad (\mathrm{D}_{MSM})} \\
\mathbf{subject\ to:} \quad y_i & \geq & 0 & i \in V \\
z_{ijk} & \geq & 0 & ijk \in E \\
y_i + y_j + y_k - \sum_{jk>_i ab} z_{iab} - \sum_{ik>_j cd} z_{jcd} - \sum_{ij>_k ef} z_{kef} - 2z_{ijk} & \geq & 1 & ijk \in E.
\end{array}
$$

Lemma 4.2.2. *Let $\bar{x}$ be a feasible solution to* (P_{MSM}) *and let $\bar{y} \in \mathbb{R}^V$ and $\bar{z} \in \mathbb{R}^E$ be defined as follows:*

$$
\begin{array}{rcll}
\bar{y}_i & = & \sum_{\substack{uv: \\ iuv\in E}} \bar{x}_{iuv} & i \in V, \\
\bar{z}_{ijk} & = & \bar{x}_{ijk} & ijk \in E.
\end{array}
$$

Then $(\bar{y}, \bar{z})$ is a feasible solution to (D_{MSM}).

Proof: By construction, we see that $\bar{y}_i \geq 0$ for all $i \in V$ and $\bar{z}_{ijk} \geq 0$ for all $ijk \in E$. To establish the feasibility of $(\bar{y}, \bar{z})$, we consider the remaining constraint of (D_{MSM}). Let $ijk \in E$. For $(\bar{y}, \bar{z})$ the left hand side of the inequality is

$$\bar{y}_i + \bar{y}_j + \bar{y}_k - \sum_{jk>_i ab} \bar{z}_{iab} - \sum_{ik>_j cd} \bar{z}_{jcd} - \sum_{ij>_k ef} \bar{z}_{kef} - 2\bar{z}_{ijk}.$$

Substituting in the values for $\bar{y}$ and $\bar{z}$ we obtain

$$\sum_{\substack{ab: \\ iab\in E}} \bar{x}_{iab} + \sum_{\substack{cd: \\ jcd\in E}} \bar{x}_{jcd} + \sum_{\substack{ef: \\ kef\in E}} \bar{x}_{kef} - \sum_{jk>_i ab} \bar{x}_{iab} - \sum_{ik>_j cd} \bar{x}_{jcd} - \sum_{ij>_k ef} \bar{x}_{kef} - 2\bar{x}_{ijk},$$

which simplifies to

$$\sum_{ab>_i jk} \bar{x}_{iab} + \sum_{cd>_j ik} \bar{x}_{jcd} + \sum_{ef>_k ij} \bar{x}_{kef} + \bar{x}_{ijk} \geq 1.$$

The inequality in the last line follows from the primal feasibility of $\bar{x}$. Thus, $(\bar{y}, \bar{z})$ is feasible for (D_{MSM}). $\square$

In Section 3.3 we saw that every feasible solution to (P_{SM}) was also an optimal solution. The linear program (P_{MSM}) has the same helpful property.

Theorem 4.2.3. *Let $\bar{x}$ be a feasible solution to* (P_{MSM}). *Then $\bar{x}$ is also an optimal solution to* (P_{MSM}).

Proof: Let $\bar{x}$ be a feasible solution to (P_{MSM}). Define $(\bar{y}, \bar{z})$ as in Lemma 4.2.2. Since $(\bar{y}, \bar{z})$ is feasible for (D_{MSM}), we can compute its objective value:

$$\begin{aligned}\sum_{i \in V} \bar{y}_i - 2 \sum_{ijk \in E} \bar{z}_{ijk} &= \sum_{i \in V} \left(\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} \right) - 2 \sum_{ijk \in E} \bar{x}_{ijk} \\ &= 3 \sum_{ijk \in E} \bar{x}_{ijk} - 2 \sum_{ijk \in E} \bar{x}_{ijk} \\ &= \sum_{ijk \in E} \bar{x}_{ijk}.\end{aligned}$$

Since the dual objective value of $(\bar{y}, \bar{z})$ equals the primal objective value of $\bar{x}$, Corollary 1.1.3 implies that $\bar{x}$ is optimal for (P_{MSM}). □

We note that the proof of this result also shows that $(\bar{y}, \bar{z})$, defined in Lemma 4.2.2, is optimal for (D_{MSM}).

Theorem 4.2.4. *Let $H = (V, E, L)$ be a 3-uniform hypergraph with preferences. There is a partition (A, B) of V such that for every feasible solution, x, to* (P_{MSM}),

$$\sum_{\substack{uv: \\ iuv \in E}} x_{iuv} = 1 \quad \textit{if} \;\; i \in A, \;\; \textit{and}$$

$$\sum_{\substack{uv: \\ iuv \in E}} x_{iuv} = 0 \quad \textit{if} \;\; i \in B.$$

Proof: Define the sets A and B as follows:

$$A = \left\{ v \in V : \sum_{\substack{uv: \\ iuv \in E}} x_{iuv} = 1 \text{ for all feasible } x \text{ of } (\mathrm{P}_{MSM}) \right\}$$

and

$$B = \left\{ v \in V : \sum_{\substack{uv: \\ iuv \in E}} x_{iuv} = 0 \text{ for all feasible } x \text{ of } (\mathrm{P}_{MSM}) \right\}.$$

Let $i \in V$ and suppose $i \notin B$. We will show $i \in A$. Since $i \notin B$, there exists a feasible solution $\bar{x}$ to (P_{MSM}) such that

$$\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} > 0.$$

Define $(\bar{y}, \bar{z})$ as in Lemma 4.2.2. The proof of Theorem 4.2.3 shows that $(\bar{y}, \bar{z})$ is optimal for (D_{MSM}). Therefore, $(\bar{x}, (\bar{y}, \bar{z}))$ satisfies the conditions of Theorem 1.1.4. Note that y_i is the dual variable corresponding to the primal constraint

$$\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} \leq 1.$$

Since

$$\bar{y}_i = \sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} > 0,$$

Theorem 1.1.4 implies that

$$\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} = 1. \tag{4.12}$$

Now, Theorem 4.2.3 tells us that every x which is feasible for (P_{MSM}) is also optimal. So, $(x, (\bar{y}, \bar{z}))$ satisfies complementary slackness for all x feasible for (P_{MSM}). In particular, (4.12) holds for all such x. Thus, $i \in A$. $\square$

This is exactly the same partition we saw for graphs with preferences in Sections 2.3 and 3.3. Not surprisingly, this partition of the vertices leads to the same consequences.

Corollary 4.2.5. *Let H be a 3-uniform hypergraph with preferences. For every feasible solution x to (P_{MSM}),*

$$\sum_{ijk \in E} x_{ijk} = \frac{|A|}{3}.$$

Proof: Let x be a feasible solution to (P_{MSM}). Recall from Theorem 4.2.4:

$$\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} = 1 \quad \text{if } \ i \in A, \text{ and}$$

$$\sum_{\substack{uv: \\ iuv \in E}} \bar{x}_{iuv} = 0 \quad \text{if } \ i \in B.$$

This yields the following:

$$
\begin{aligned}
3\sum_{ijk\in E} x_{ijk} &= \sum_{i\in V}\left(\sum_{\substack{uv:\\ iuv\in E}} \bar{x}_{iuv}\right) \\
&= \sum_{i\in A}\left(\sum_{\substack{uv:\\ iuv\in E}} \bar{x}_{iuv}\right) + \sum_{i\in B}\left(\sum_{\substack{uv:\\ iuv\in E}} \bar{x}_{iuv}\right) \\
&= \sum_{i\in A}\left(\sum_{\substack{uv:\\ iuv\in E}} \bar{x}_{iuv}\right) \\
&= \sum_{i\in A} 1 \\
&= |A|.
\end{aligned}
$$

$\square$

Corollary 4.2.6. *Let H be a 3-uniform hypergraph with preferences. If H has a majority stable matching, then all majority stable matchings of H have the same size.*

Corollary 4.2.7. *Let H be a 3-uniform hypergraph with preferences. If $|A|$ is not a positive multiple of 3 then H does not have a majority stable matching.*

Theorem 4.2.4 and its corollaries suggest that the feasible solutions of (P_{MSM}) have very strong structure with respect to the matching constraints (4.2). This structure has been mainly due to some clever complementary slackness arguments. How do these solutions behave with respect to the stability constraints (4.3)?

Lemma 4.2.8. *Let $H = (V, E, L)$ be a 3-uniform hypergraph with preferences and let $ijk \in E$. If there is some $\bar{x}$, feasible for (P_{MSM}), with $\bar{x}_{ijk} > 0$, then for every feasible solution, x, of (P_{MSM}),*

$$\sum_{ab>_i jk} x_{iab} + \sum_{cd>_j ik} x_{jcd} + \sum_{ef>_k ij} x_{kef} + 2x_{ijk} = 2.$$

Proof: Let $\bar{x}$ be a feasible solution to (P_{MSM}) such that $\bar{x}_{ijk} > 0$ and define $(\bar{y}, \bar{z})$ as in Lemma 4.2.2. Theorem 4.2.3 shows us that $(\bar{x}, (\bar{y}, \bar{z}))$ is a primal-dual optimal pair for (P_{MSM}) and (D_{MSM}). But Theorem 4.2.3 also tells us that every x, feasible for (P_{MSM}), is also optimal for (P_{MSM}). So $(x, (\bar{y}, \bar{z}))$ is a primal-dual optimal pair for every feasible x. Note that $\bar{z}_{ijk} = \bar{x}_{ijk}$ is the dual variable corresponding to the primal constraint (4.3).

Since $\bar{z}_{ijk} > 0$, complementary slackness tells us that (4.3) is satisfied with equality for every feasible x. □

Let M and $\bar{M}$ be stable matchings of a hypergraph with preferences. We will say that vertex v *prefers* M to $\bar{M}$ if v prefers its matching edge in M to its matching edge in $\bar{M}$.

Corollary 4.2.9. *Let H be a 3-uniform hypergraph with preferences and let M be a majority stable matching of H. If $ijk \in M$, then there is no majority stable matching, $\bar{M}$, of H such that all of i, j, and k prefer $\bar{M}$ to M.*

Proof: Let x^M be the incidence vector of the majority stable matching M such that $ijk \in M$. Suppose, for a contradiction, that there exists a majority stable matching $\bar{M}$ such that all of i, j, and k prefer $\bar{M}$ to M and let $x^{\bar{M}}$ be the incidence vector of $\bar{M}$. Note that $x^{\bar{M}}_{ijk} = 0$. Since $x^M_{ijk} = 1$, Lemma 4.2.8 shows that

$$\sum_{ab >_i jk} x^{\bar{M}}_{iab} + \sum_{cd >_j ik} x^{\bar{M}}_{jcd} + \sum_{ef >_k ij} x^{\bar{M}}_{kef} = 2. \tag{4.13}$$

However, all of i, j, and k prefer $\bar{M}$ to M. So

$$\begin{aligned} \sum_{ab >_i jk} x^{\bar{M}}_{iab} &= 1, \\ \sum_{cd >_j ik} x^{\bar{M}}_{jcd} &= 1, \text{ and} \\ \sum_{ef >_k ij} x^{\bar{M}}_{kef} &= 1. \end{aligned}$$

This contradicts (4.13) and gives us the result. □

Corollary 4.2.9 is of similar flavour to Corollary 2.1.6 from Section 2.1.1: Improving the outcome of a sufficiently large group of people will almost undoubtedly come at the expense of another group of people.

Sadly, all this exciting work cannot hide the fact that majority stable matchings in hypergraphs with preferences can still pose difficulties. To illustrate this point, consider the following disheartening result.

Theorem 4.2.10. *If H is a 3-uniform hypergraph with preferences and H has a majority stable matching, then H has only one majority stable matching.*

Proof: Let H be a 3-uniform hypergraph with preferences and suppose, for a contradiction, that M and $\bar{M}$ are distinct majority stable matchings of H. Let T be the set of edges which are in M but not in $\bar{M}$ and let U be the set of edges which are in $\bar{M}$ but not M.

Define W to be the set of vertices of H that are matched in M by the edges of T. Notice that the vertex partition from Theorem 4.2.4 shows us that W is also the set of vertices matched in $\bar{M}$ by the edges of U. Furthermore, $|W|$ is a positive multiple of 3.

Now, $x^M_{ijk} = 1 > 0$ for every $ijk \in T$. Therefore, by Lemma 4.2.8,

$$\sum_{ab>_i jk} x_{iab} + \sum_{cd>_j ik} x_{jcd} + \sum_{ef>_k ij} x_{kef} + 2x_{ijk} = 2$$

holds for the incidence vector, x, of any majority stable matching of H. Since $x^{\bar{M}}_{ijk} = 0$ for every $ijk \in T$,

$$\sum_{ab>_i jk} x^{\bar{M}}_{iab} + \sum_{cd>_j ik} x^{\bar{M}}_{jcd} + \sum_{ef>_k ij} x^{\bar{M}}_{kef} = 2.$$

This tells us that for every $ijk \in T$, exactly two of i, j, and k prefer $\bar{M}$ to M. In simple terms, two thirds of the vertices in W prefer $\bar{M}$ to M. However, if we repeat the argument for the edges of U, we will also see that two thirds of the vertices in W prefer M to $\bar{M}$. The result follows from this contradiction. $\square$

Theorem 4.2.10 rules out any further discussion of the set of majority stable matchings for a particular 3-uniform hypergraph. However, if we were given the task of finding a stable matching in a 3-uniform hypergraph with preferences, a majority stable matching would be a desirable choice because of the increased agreement among the vertices. Is there a large class of 3-uniform hypergraphs with preferences that will have a majority stable matching? In point of fact, the class is much larger than the class of graphs with preferences that have a stable matching.

4.2.1 Constructing Majority Stable Matchings

In some sense, majority stable matchings are the ideal stable matchings in 3-uniform hypergraphs with preferences. The problem is that they do not always exist. We would like to construct an infinite class of 3-uniform hypergraphs with preferences that have a majority stable matching.

Our construction starts with a graph with preferences, G, and a fixed stable matching of G, say $\bar{M}$. We will build a 3-uniform hypergraph with preferences, $H(G;\bar{M}) = (V, E, L)$, using the structure of G and $\bar{M}$. Specifically, let $G = (\bar{V}, \bar{E}, \bar{L})$ be a graph with preferences that has at least one stable matching and let $\bar{M}$ be a stable matching of G. The vertices of $H(G;\bar{M})$ are given by

$$V := \bar{V} \cup \{z_{xy} : xy \in \bar{M}\}.$$

That is, the vertices of $H(G;\bar{M})$ are the vertices of G plus a vertex for each edge of $\bar{M}$. To define the edges of $H(G;\bar{M})$, we first choose a subset of edges, $T_{z_{xy}}$, of G for every $xy \in \bar{M}$.

The only rule for choosing these $T_{z_{xy}}$ is that we must have $xy \in T_{z_{xy}}$. We then add z_{xy} to every $ab \in T_{z_{xy}}$. In other words,

$$E := \{xyz_{xy} : xy \in \bar{M}\} \cup \{abz_{xy} : ab \in T_{z_{xy}} \subseteq \bar{E} \text{ for all } xy \in \bar{M}\}.$$

Each vertex will have one of two types of preference lists. If $x \in \bar{V}$, then we construct L_x using the following rules:

- If $a >_x b$ in $\bar{L}_x$, then $xaz_{ij} >_x xbz_{kl}$ for all $ij, kl \in \bar{M}$, and
- if $xy \in \bar{M}$, then $xyz_{xy} \geq_x xyz_{ij}$ for all $ij \in \bar{M}$.

For $xy \in \bar{M}$, $L_{z_{xy}}$ is built in the following way: Let $ab \in \bar{E}$. If $ab \notin \bar{M}$ and both a and b prefer their matching edges of $\bar{M}$ to ab, then $abz_{xy} >_{z_{xy}} xyz_{xy}$. Otherwise, we want $xyz_{xy} >_{z_{xy}} abz_{xy}$. Figure 4.1 and Table 4.7 provide a simple example of our construction.

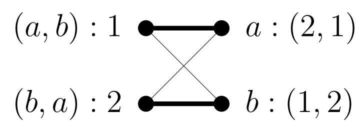

Figure 4.1: Starting the construction of $H(G; \bar{M})$

We can see that $\bar{M} = \{a1, b2\}$ is a stable matching of G. Table 4.7 shows the hypergraph $H(G; \bar{M})$.

1	$a1z_{a1}$	$a1z_{b2}$	$b1z_{a1}$	$b1z_{b2}$
2	$b2z_{b2}$	$b2z_{a1}$	$a2z_{b2}$	$a2z_{a1}$
a	$a2z_{a1}$	$a2z_{b2}$	$a1z_{a1}$	$a1z_{b2}$
b	$b1z_{b2}$	$b1z_{a1}$	$b2z_{b2}$	$b2z_{a1}$
z_{a1}	$a1z_{a1}$	$a2z_{a1}$	$b1z_{a1}$	$b2z_{a1}$
z_{b2}	$b2z_{b2}$	$b1z_{b2}$	$a2z_{b2}$	$a1z_{b2}$

Table 4.7: Construction of $H(G; \bar{M})$

Consider the matching $M = \{a1z_{a1}, b2z_{b2}\}$. Since 1, 2, z_{a1}, and z_{b2} are matched to their favourite edge, M is a majority stable matching. From this example, it seems that most vertices must be matched to their favourite edges. In fact, this construction will provide examples where very few vertices are matched to their favourite edge. However, those hypergraphs are too large to display here.

The example in Figure 4.1 and Table 4.7 illustrates something surprising. A graph with preferences may have many stable matchings and different choices of $\bar{M}$ will lead, in general, to different outcomes for $H(G; \bar{M})$. However, the graph and its preferences in Figure 4.1 is symmetric. If we chose $\bar{M}' = \{a2, b1\}$ (which is also stable) and suitably relabelled the vertices, we would see that $H(G; \bar{M}) = H(G; \bar{M}')$.

Lemma 4.2.11. *Let G be a graph with preferences and let $\bar{M}$ be a stable matching of G. Then $H(G; \bar{M})$ has a majority stable matching.*

Proof: Let G be a graph with preferences and let $\bar{M}$ be a stable matching of G. Define $M := \{xyz_{xy} : xy \in \bar{M}\}$. We will show that M is a majority stable matching of $H(G; \bar{M})$. By construction, M is certainly a matching of $H(G; \bar{M})$. Suppose, for a contradiction, that M is not a majority stable matching. Then, there exists an edge ijz_{xy}, for some $xy \in \bar{M}$, such that at least two of i, j, and z_{xy} prefer ijz_{xy} to their respective matching edges.

Notice that if $ijz_{xy} >_{z_{xy}} xyz_{xy}$, then, by the construction of $L_{z_{xy}}$, $ij \notin \bar{M}$ and both i and j prefer their matching edges of $\bar{M}$ to ij. So, i and j also prefer their matching edges of M to ijz_{xy}. Hence, z_{xy} is the only vertex of i, j, and z_{xy} that prefers ijz_{xy} to its matching edge. This would contradict that ijz_{xy} is a blocking edge of M. So, we may assume that only i and j prefer ijz_{xy}.

Suppose $ikz_{ik} \in M$, so that $ik \in \bar{M}$. By construction of L_i, ikz_{ik} is i's most preferred edge that contains k. So, if i prefers ijz_{xy} to ikz_{ik}, then $j \neq k$ and $j >_i k$ in $\bar{L}_i$. Similarly, j prefers i to its matching partner of $\bar{M}$. Therefore, ij is a blocking edge of $\bar{M}$, contradicting that $\bar{M}$ is a stable matching of G. Thus, M is indeed a majority stable matching of $H(G; \bar{M})$. □

The construction of $H(G; \bar{M})$ takes a graph with preferences and a stable matching to create a 3-uniform hypergraph with preferences and a majority stable matching. Could there be a converse to this statement? In other words, does every 3-uniform hypergraph with a majority stable matching correspond in some natural way to a graph with a stable matching? Currently, we do not know the answer to this question, but we can show a result of this form for a certain special class of hypergraphs.

So far, the graphs we considered have been *simple*: For each pair of vertices, x and y, there is, at most, one edge of the form xy. For the remainder of this section, we are going to relax this requirement and allow pairs of vertices to have multiple edges between them. For a fixed pair of vertices, x and y, the set of edges that have both x and y as endpoints is called the *parallel class* of x and y. The graphs which allow parallel classes bigger than one are called *multigraphs*.

Conveniently, the definition of "stable matching" is the same for graphs and multigraphs. In 2005, Cechlárová and Fleiner showed that finding a stable matching in a multigraph with preferences is equivalent to finding a stable matching in graph with preferences. Specifically, they showed that if H is a multigraph with preferences, then there is a simple graph with preferences, G, such that H has a stable matching if and only if G has a stable matching. Consequently, Irving's algorithm can be used to find a stable matching in a multigraph with preferences by finding a stable matching in a suitable graph with preferences. We will refer the reader to the paper of Cechlárová and Fleiner for more details [9].

Let $H = (V, E, L)$ be a 3-uniform hypergraph with preferences. We will call H *bipartite* if there exists a set $X \subseteq V$ such that every edge of H contains exactly one vertex of X. Notice that, in a bipartite 3-uniform hypergraph, the neighbourhood of X, $N(X)$, induces a multigraph. This multigraph, $H[N(X)]$, has vertex set $N(X)$ and edge set $\{(yz)_x : x \in X \text{ and } xyz \in E\}$.

There is a natural way to define the preference lists of $N(X)$ based on L. Observe that there is a one-to-one correspondence between the edges of H and the edges of $H[N(X)]$. Therefore, if $xyz >_y \bar{x}y\bar{z}$ in H, we will insist that $(yz)_x >_y (y\bar{z})_{\bar{x}}$ in $H[N(X)]$ for all $x \in X$ and $y, z, \bar{z} \in N(X)$. With this requirement, we will say that $H[N(X)]$ has *induced* preferences. Table 4.8 illustrates the close relationship between the preference lists of H and the preference lists of $H[N(X)]$.

H	y	xyz	$\bar{x}yz$	$\bar{x}y\bar{z}$	$xy\bar{z}$
$H[N(X)]$	y	$(yz)_x$	$(yz)_{\bar{x}}$	$(y\bar{z})_{\bar{x}}$	$(y\bar{z})_x$

Table 4.8: Preference lists of $N(X)$ in H and $H[N(X)]$

Lemma 4.2.12. *Let $H = (V, E, L)$ be a bipartite 3-uniform hypergraph with preferences and let $X \subseteq V$ be its distinguished set of vertices. If H has a majority stable matching, then $H[N(X)]$ with induced preferences has a stable matching.*

Proof: Suppose $H = (V, E, L)$ is a 3-uniform hypergraph with preferences and let X be a set of vertices such that every $e \in E$ contains exactly one vertex of X. Let $\bar{M}$ be a majority stable matching of H. Define $M := \{(yz)_x : xyz \in \bar{M} \text{ and } x \in X\}$. We will show that M is a stable matching of $H[N(X)]$. The set M is certainly a matching of $H[N(X)]$. Suppose, for a contradiction, that M is not a stable matching of $H[N(X)]$. Then there exists an edge of $H[N(X)]$, say $(ab)_{\bar{x}}$, that blocks M.

Since $(ab)_{\bar{x}} \notin M$, the construction of M tells us that $\bar{x}ab \notin \bar{M}$. Since $\bar{M}$ is a majority stable matching, at least two of $\bar{x}$, a, and b prefer their respective matching edges of $\bar{M}$ to xab. However, the correspondence between the preference lists of H and $H[N(X)]$ shows that at least one of a or b prefers its matching edge of M to $(ab)_x$, contradicting that $(ab)_x$ was a blocking edge of M. Thus, M is a stable matching of $H[N(X)]$. □

Suppose that M is a stable matching of $H[N(X)]$. We can define a simple graph with preferences, G, by deleting all but one edge from each parallel class, while preserving the edges of M. The preference lists of G are obtained from $H[N(X)]$ by simply deleting the entries corresponding to the edges that were deleted. Then if M is a stable matching of $H[N(X)]$, M remains stable in G because we did not delete any edges of M. As with our construction of $H(G; \bar{M})$, this G is not unique. Different choices of edge deletions will give us, in general, different graphs with preferences.

Another question which we did not answer here is: Can we decide if a 3-uniform hypergraph with preferences has a majority stable matching? At the moment, it is unclear if the feasibility of (P_{MSM}) is sufficient to guarantee a majority stable matching or if (P_{MSM}) can be feasible without an integral solution. In the next chapter, we will discuss some other open problems relating to 3-uniform hypergraphs.

Chapter 5

Concluding Remarks

The practical applications of stable matchings have been evident since their formal introduction in 1962. Indeed, the Gale-Shapley algorithm has been used to solve instances of the college admissions problem for the NRMP and, in some cases, real schools. Today, the medical field has found more complicated applications of hypergraphic stable matchings [24]. Stable matchings have even been applied in many other areas of mathematics [33]. A surprising fact to many people is that rich structure can be found when stable matchings are considered as a theoretical object. This versatility has helped the theory of stable matchings to become a popular and developed research topic.

Before we continue on to some interesting unsolved problems, let us summarize what we have seen in the preceding chapters.

Chapter 2
We began with an introduction to graphs with preferences and the basics of stable matchings. We then saw that the proposal/rejection process used by Gale and Shapley led to some very intriguing properties. We also looked into stable matchings in non-bipartite graphs with preferences. Irving's algorithm showed that many of the interesting stable matching properties found in bipartite graphs remain in stable matchings of non-bipartite graphs.

As an aside, we looked briefly at the intimate connection between stable matchings and lattice theory. Delving further into this association would require much more time than we have here, but is a fascinating subject in its own right.

Chapter 3
Chapter 3 focused on the following question: "Can we generalize Theorem 2.1.1 to non-bipartite graphs with preferences?" In fact, this is achieved by relaxing the integrality of a stable matching to allow edges to take fractional values. But, rather than attacking

non-bipartite graphs directly, we looked to fractional stable matchings in hypergraphs with preferences. Using a matrix version of a powerful result due to Scarf, Theorem 3.1.2 promised that EVERY hypergraph with preferences has a fractional stable matching. We then applied Theorem 3.1.2 to the special case of graphs. We saw that every graph with preferences contains a set of edges that is, at least, very similar to a stable matching. As a consequence of this fact, we can characterize the graphs with preferences that admit stable matchings.

Linear programming provided tools that enabled us to talk about stable matchings and fractional stable matchings at the same time. This allowed us to prove that fractional stable matchings have analogous properties to stable matchings in graphs with preferences.

Chapter 4
Contrary to stable matchings in graphs with preferences, stable matchings in hypergraphs with preferences behave very badly. The remarkable properties evident in graphs with preferences vanish when edges are allowed to have a size bigger than two. The lack of these properties is likely the reason why most hypergraphic stable matching problems are NP-complete.

By considering majority stable matchings, we were able to use linear programming to our advantage again. However, our successes were fleeting. The existence of a majority stable matching in a 3-uniform hypergraph with preferences is such a strong property that there can be, at most, one. However, we saw that it is possible to construct a large class of 3-uniform hypergraphs with preferences which admit majority stable matchings. This construction uncovered a strong relationship between majority stable matchings in 3-uniform hypergraphs and stable matchings in graphs.

5.1 Open Questions

To conclude this thesis, we will look to the future and discuss two open problems about stable matchings in 3-uniform hypergraphs with preferences.

5.1.1 The Cyclic Stable Family Problem

Our first open problem is a slight variation on the stable family problem. It is rumoured to be due to Knuth, but it first appeared in a paper by Ng and Hirschberg after they proved that the stable family problem was NP-complete [37]. Specifically, we are interested in the following:

Problem 6 (Cyclic Stable Family Problem)**.** A community consists of n men, n women, and n dogs. Each man ranks every woman in terms of with whom they would prefer to be in a family. Similarly, each woman ranks all the dogs and each dog ranks all the men. Can we find a set of families such that if a man, a woman, and a dog are not a family then at least one of them prefers their current family?

Once again, we are looking for a stable matching in a tripartite 3-uniform hypergraph with preferences, $H = (A \cup B \cup C, E, L)$. The difference here comes in the structure of the preference lists. Notice that each vertex no longer ranks all its adjacent neighbours: The vertices of A are interested only in the vertices of B, B is interested only in C, and C is interested only in A. For example, if $i \in A$ and $j \in B$, then any edge containing both i and j is the same from i's perspective. In this situation, we will say that H has *cyclic preferences.*

With this special type of preference list, the definition of a blocking edge remains the same: A blocking edge for a matching M of H is an edge $ijk \notin M$ such that all of i, j, and k prefer ijk to their respective matching edges. A matching, M, is stable if it does not have any blocking edges.

For the remainder of this section, we will make two assumptions:

- For every $i \in A$, $j \in B$, and $k \in C$, ijk is an edge of H, and
- $n = |A| = |B| = |C|$.

The first assumption simply says that H is a *complete* hypergraph. This problem becomes NP-complete without the completeness restriction [6]. The second assumption is purely aesthetic. If the vertex classes have different sizes, then we can add just enough artificial vertices so that A, B, and C are the same size. Any of these artificial vertices will be appended to the end of any preference lists in an arbitrary order. Then, if the new hypergraph has a stable matching $\bar{M}$, we can exclude any edge of $\bar{M}$ that contains any of the artificial vertices to obtain a stable matching of the original hypergraph.

To some extent, this variant of the stable family problem most closely resembles the original stable marriage problem: Each vertex class has preferences over only one other vertex class. Recall that every instance of the stable marriage problem has a stable matching. The belief is that this fact is also true for the cyclic stable family problem.

Conjecture 5.1.1 (Eriksson, Sjöstrand, and Strimling [10])**.** *Every tripartite* 3*-uniform hypergraph with cyclic preferences has a stable matching.*

How close are we to a proof of this conjecture? Actually, not close at all. At this point in time, we know only that very small instances always admit stable matchings.

Theorem 5.1.2 (Eriksson, Sjöstrand, and Strimling [10])**.** *Let H be a tripartite* 3*-uniform hypergraph with cyclic preferences. If $n \leq 4$, then H has a stable matching.*

The proof of this result is a highly technical case analysis that is difficult to extend to larger values of n. Additionally, Eriksson, Sjöstrand, and Strimling provided substantial computer evidence to suggest that the $n = 5$ case is true [10]. However, a formal proof of this fact remains elusive. We would suspect that successful proofs of the full conjecture would cleverly employ one of two strategies:

- Generalize the Gale-Shapley algorithm to account for the extra vertex class, or
- artificially define a second set of preferences for the vertices of A over the vertices of C and then use a double application of Theorem 2.1.1.

Since we are unable to resolve Conjecture 5.1.1, nor the $n = 5$ case, in this thesis (perhaps at a later date), we will look at some special cases.

Lemma 5.1.3. *Let $H = (A \cup B \cup C, E, L)$ be a 3-uniform, tripartite hypergraph with cyclic preferences. If every vertex in A has the same preference list, then H has a stable matching.*

Proof: We proceed by induction on n. If $n \leq 4$, then H has a stable matching by Theorem 5.1.2. So, we can assume that $n \geq 5$ and that the result holds for all $k < n$. Suppose further that $y \in B$ is the most preferred vertex by the vertices of A, $z \in C$ is the favourite vertex of y, and $x \in A$ is the favourite vertex of z.

Consider the subhypergraph with preferences, T, obtained by removing the vertices x, y, and z from H. By the inductive hypothesis, T has a stable matching $\bar{M}$. Define the matching $M := \bar{M} \cup \{xyz\}$ for H.

Claim: The matching M is a stable matching of H.

Proof of Claim: Suppose, for a contradiction, that M is not a stable matching. Then there exists a blocking edge ijk for M. Notice that any ijk must contain one of the vertices x, y, or z. Otherwise, ijk would also be a blocking edge for $\bar{M}$ in T, contradicting that $\bar{M}$ is a stable matching of T. Now, recall that $xyz \in M$. Therefore, the vertices x, y, and z are already matched to their most preferred vertex. Thus, the blocking edge ijk cannot exist. This contradiction shows that M is a stable matching of T. □

Notice that we could strengthen this result slightly: Instead of restricting the vertices of A to a single preference list, we could just insist that the first $n-4$ entries are identical.

Lemma 5.1.4. *Let $H = (A \cup B \cup C, E, L)$ be a tripartite 3-uniform hypergraph with cyclic preferences. If each vertex of A has a distinct favourite vertex in B, then H has a stable matching.*

Proof: Let M be any matching of H such that each vertex in A is matched in an edge containing its favourite vertex of B.

Claim: The matching M is a stable matching of H.

Proof of Claim: Suppose, for a contradiction, that M is not stable. Then, there exists a blocking edge ijk for M. We may assume that $i \in A$. By definition of M, i is already matched to its favourite vertex, meaning that the vertex j cannot exist. This contradicts that ijk is a blocking edge for M. Thus, M is a stable matching. $\square$

An edge ijk is a *preferable edge* if j is first in i's preference list, k is first in j's preference list, and i is first in k's preference list. In the $n = 5$ case, the existence of a preferable edge is sufficient to guarantee the existence of a stable matching and follows from Lemma 5.1.3.

Lemma 5.1.5. *Let $H = (A \cup B \cup C, E, L)$ be a tripartite 3-uniform hypergraph with cyclic preferences where $n = 5$. If H has a preferable edge, then H has a stable matching.*

As anyone who has been part of a committee will know, the unanimous agreement of Lemma 5.1.3 is an exceedingly special case. Even the complete discord of Lemma 5.1.4 and the ideal triple of Lemma 5.1.5 are very rare. But, these extreme cases make stable matching proofs quite pleasant. To have any hope of proving Conjecture 5.1.1, we will need to fill in this gap. Sadly, this is quite a substantial challenge when we do not know of a way to extend the work of Gale and Shapley. To partially illustrate this fact, let us consider the $n = 5$ case.

Lemma 5.1.6. *Let $H = (A \cup B \cup C, E, L)$ be a tripartite 3-uniform hypergraph with cyclic preferences where $n = 5$. If there are two vertices of B that are first and second in every preference list of A, then H has a stable matching.*

Proof: Let $H = (A \cup B \cup C, E, L)$ be a tripartite 3-uniform hypergraph with cyclic preferences where $n = 5$. Suppose $b, \bar{b} \in B$ are the two vertices described in the statement of the lemma. Let A_b be the subset of vertices of A that rank b first and $\bar{b}$ second. Similarly, let $A_{\bar{b}}$ be the vertices of A that rank $\bar{b}$ first and b second. Notice that, by assumption, $(A_b, A_{\bar{b}})$ is a partition of A.

Let c and $\bar{c}$ in C be the favourite vertices of b and $\bar{b}$, and let a and $\bar{a}$ in A be the favourite vertices of c and $\bar{c}$, respectively.

At this point, we can make the following assumptions:

- $c \neq \bar{c}$,
- $a \neq \bar{a}$, and
- $a \in A_{\bar{b}}$ and $\bar{a} \in A_b$.

If $c = \bar{c}$, then either abc or $a\bar{b}c$ is a preferable edge, depending on whether $a \in A_b$ or $a \in A_{\bar{b}}$. Therefore by Lemma 5.1.5, H has a stable matching. Similarly, we can find preferable edges if $a = \bar{a}$, $a \in A_b$, or $\bar{a} \in A_{\bar{b}}$.

Now, consider the subhypergraph, T, of H obtained by removing the vertices a, $\bar{a}$, b, $\bar{b}$, c, and $\bar{c}$. By Theorem 5.1.2, T has a stable matching, say $\bar{M}$. Define $M := \bar{M} \cup \{abc, \bar{a}\bar{b}\bar{c}\}$, a matching of H.

Claim: The matching M is a stable matching of H.

Proof of Claim: Suppose, for a contradiction, that M is not stable. Then, there exists a blocking edge ijk for M. Since $\bar{M}$ is a stable matching for T and b, $\bar{b}$, c, and $\bar{c}$ are matched in M to their favourite vertices, we can say that the edge ijk contains either a or $\bar{a}$. Suppose $i = a$. Since $a \in A_{\bar{b}}$ and $abc \in M$, the only vertex a prefers to b is $\bar{b}$. But $\bar{b}$ is matched to $\bar{c}$, its favourite vertex of C. Therefore it is not possible for ijk to be a blocking edge, contradicting that M was not a stable matching. This contradiction gives us the result. $\square$

This proof relies on the three assumptions that guarantee a preferable edge. But very few instances of the cyclic stable family problem actually have a preferable edge. This means that we have to consider edges that are not as well liked by their vertices, driving the number of cases out of control very quickly. We hope that a more cunning approach is possible.

5.1.2 Bounded Denominators

In Chapter 3, we saw that every graph with preferences has a stable partition. As a consequence of this, every graph with preferences also has a half-integral stable matching. It is natural to wonder if 3-uniform hypergraphs have a similar fractional stable matching. With a bit of work, or a quick trip back to Table 4.4 in Section 4.1, we see that it is not possible for all hypergraphs with preferences to have a half-integral stable matching. So, what kinds of edge values can we hope for? Are they even rational? More concretely, can we find some absolute constant N such that for every 3-uniform hypergraph with preferences, there is a fractional stable matching where the denominator of every fractional value is at most N?

As an example, we saw that $N = 2$ for graphs. If such an N existed for 3-uniform hypergraphs, Table 4.4 shows that $N = 3$ is the best possible. We are interested in knowing if three is the true value of N for all 3-uniform hypergraphs. Before we continue, we will make this notion precise. A fractional stable matching, x, is a $\frac{1}{3}$*-integral stable matching* if $x_e \in \{0, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1\}$ for every $e \in E$. Based on this definition, we ask the following question:

Question 5.1.7. *Does every 3-uniform hypergraph with preferences have a $\frac{1}{3}$-integral stable matching?*

To clarify, a 3-uniform hypergraph may have many fractional stable matchings; some of which may even have arbitrarily large denominators. We would like to know if at least one of the fractional stable matchings has reasonable values. At this point, we do not know the answer to this problem. However, there is some evidence to suggest it might be true.

Lemma 5.1.8. *Let $H = (V, E, L)$ be a hypergraph with preferences and let x be a fractional stable matching of H. The support of x, $supp(x)$, has at most $|V|$ edges.*

Proof: Let $H = (V, E, L)$ be a 3-uniform hypergraph with preferences. Let x be a fractional stable matching of H and let $S = supp(x)$. Since x is a fractional stable matching, we can orient each edge e towards a vertex v such that $\sum_{e \leq_v j} x_j = 1$. The following claim shows that $|S| \leq |V|$, as required.

Claim: Each vertex v is the head of at most one arc in S.

Proof of Claim: Suppose vertex v is the head of arcs $e, h \in S$. Without loss of generality, we may assume that $e <_v h$. Since v is the head of both e and h, $\sum_{e \leq_v j} x_j = 1$ and $\sum_{h \leq_v j} x_j = 1$. But since $e <_v h$, we must have $x_e = 0$, contradicting that $e \in S$. □

Lemma 5.1.8 tells us that the subhypergraph induced by the support of a fractional stable matching is very sparse. This greatly reduces the number of eccentric special cases for the structure of the support of a fractional stable matching.

Recall from Chapter 3 that we could show another result that is similar to the claim in the above proof: Each vertex v is the tail of at most one arc in S. With this second property, it is possible to deduce the structure of S, and hence, define a half-integral stable matching. Ideally, we would like to be able to do something similar here. As we might expect, however, this is the step that is currently missing. But, for small 3-uniform hypergraphs, it is feasible to attack the problem directly.

Lemma 5.1.9. *Every 3-uniform hypergraph with preferences with four vertices has a $\frac{1}{3}$-integral stable matching.*

Proof: Let $H = (V, E, L)$ be a 3-uniform hypergraph with preferences where $|V| = 4$. Let x be a fractional stable matching of H and let $S = supp(x)$. If $|S| = 1$, then S is a stable matching. So, we may assume that $|S| \geq 2$.

Claim: The set S cannot contain exactly three edges.

Proof of Claim: Suppose, for a contradiction, that $|S| = 3$. Let $a, b, c \in S$. Since H has only four vertices, there must be exactly one vertex, v, that is incident to all of a, b, and c. Since x is a fractional stable matching, we must have

$$x_a + x_b + x_c = 1. \tag{5.1}$$

We may assume that $a >_v b >_v c$. The edge b must contain a vertex $u \neq v$ such that

$$\sum_{b \leq_u j} x_j = 1,$$

as x is a fractional stable matching. Note that $deg(u) = 2$ in S, by the above remarks. However, this contradicts (5.1) since v is the only vertex incident to all of a, b, and c. Thus, it is not possible for $|S| = 3$.

Suppose $|S| = 2$ and let $a, b \in S$. Since H has only four vertices, the edges a and b must share exactly two vertices, say u and v. Further, $x_a < 1$ and $x_b < 1$. Let z be the third vertex of the edge a. Notice that if both $a >_u b$ and $a >_v b$ hold, then we must have

$$\sum_{a \leq_z j} x_j = 1,$$

since x is a fractional stable matching. But, z is only incident to a in S. Therefore,

$$\sum_{a \leq_z j} x_j = x_a < 1,$$

contradicting that x was a fractional stable matching. So, we must have $a >_u b$ and $b >_v a$. Now, we can define a new fractional stable matching, $\bar{x}$, by

$$x_e = \begin{cases} \frac{1}{2} & \text{if } e \in S \\ 0 & \text{otherwise.} \end{cases}$$

The stability of $\bar{x}$ follows since $supp(\bar{x}) = supp(x)$.

If $|S| = 4$, then by the claim in the proof of Lemma 5.1.8, every edge of S must be last in the preference list of some vertex. Simply set $\bar{x}_e = \frac{1}{3}$ for every $e \in E$ to construct a fractional stable matching.

In all cases, H has a $\frac{1}{3}$-integral stable matching. □

It is possible to extend this result to 3-uniform hypergraphs with $|V| \leq 5$. However, it would be rather cruel to force the reader to suffer through the case-by-case verification when the end of this thesis is so close. Hopefully, we will think of a clever way to simplify the proof enough in the near future to make it presentable.

There is some disagreement about the potential answer to Question 5.1.7. In general hypergraphs with preferences, it is known that the denominators in fractional stable matchings can be unbounded. For example, consider the r-uniform analogue of Table 4.4. It is possible to show that the best fractional stable matching assigns $\frac{1}{r}$ to each of the edges. Once again, we will let the reader play with this problem (Hint: Notice the cyclic nature of the preferences). Perhaps, for 3-uniform hypergraphs, it is more realistic to hope that the denominators are simply bounded by a finite positive integer. In either case, we would like to resolve this question.

Bibliography

[1] Hernán Abeledo and Garth Isaak. A characterization of graphs that ensure the existence of stable matchings. *Mathematical Social Sciences*, 22(1):93 – 96, 1991. 21

[2] Hernán G. Abeledo and Yosef Blum. Stable matchings and linear programming. *Linear Algebra and its Applications*, 245:321–333, 1996.

[3] Hernán G. Abeledo and Uriel G. Rothblum. Stable matchings and linear inequalities. *Discrete Appl. Math.*, 54(1):1–27, 1994. 43, 44, 45

[4] Ron Aharoni and Tamás Fleiner. On a lemma of Scarf. *J. Comb. Theory Ser. B*, 87(1):72–80, 2003. 32, 33, 36

[5] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997. 9

[6] Péter Biró and Eric McDermid. Three-sided stable matchings with cyclic preferences. *Algorithmica*, 58(1):5–18, 2009. 50, 65

[7] Charles Blair. Every finite distributive lattice is a set of stable matchings. *J. Comb. Theory, Ser. A*, 37(3):353–356, 1984. 20

[8] J.A. Bondy and U.S.R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008. 7

[9] Katarína Cechlárová and Tamás Fleiner. On a generalization of the stable roommates problem. *ACM Trans. Algorithms*, 1(1):143–156, 2005. 60

[10] Kimmo Eriksson, Jonas Sjöstrand, and Pontus Strimling. Three-dimensional stable matching with cyclic preferences. *Mathematical Social Sciences*, 52(1):77 – 87, 2006. 65, 66

[11] Tomás Feder. A new fixed point approach for stable networks and stable marriages. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 513–522, New York, NY, USA, 1989. ACM. 44

[12] Tamás Fleiner. *Stable and Crossing Structures*. PhD thesis, Technische Universiteit Eindhoven, 2000.

[13] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. 3, 14, 15, 17

[14] David Gale and Marilda Sotomayor. Some remarks on the stable matching problem. *Discrete Applied Mathematics*, 11(3):223 – 232, 1985. 16

[15] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Appl. Math.*, 42(2-3):177–201, 1993.

[16] Fred Galvin. The list chromatic index of a bipartite multigraph. *Journal of Combinatorial Theory, Series B*, 63(1):153 – 158, 1995. 3

[17] George Grätzer. *Lattice Theory: First Concepts and Distributive lattices*. W.H. Freeman and Company, San Francisco, California, 1971. 11

[18] Dan Gusfield, Robert Irving, Paul Leather, and Michael Saks. Every finite distributive lattice is a set of stable matchings for a small stable marriage instance. *Journal of Combinatorial Theory, Series A*, 44(2):304 – 309, 1987. 20

[19] Dan Gusfield and Robert W. Irving. *The stable marriage problem: structure and algorithms*. MIT Press, Cambridge, MA, USA, 1989. 16, 18, 20, 26, 27

[20] Penny Haxell. Stable matchings, October 2008. Lecture Notes, CO 342, University of Waterloo.

[21] Penny Haxell. On forming committees. unpublished manuscript, University of Waterloo, 2010. 32, 33

[22] Penny Haxell and G. T. Wilfong. A fractional model of the border gateway protocol (BGP). In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 193 – 199, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[23] Chien-Chung Huang. Two's company, three's a crowd: Stable family and threesome roommates problems. In *ESA*, pages 558–569, 2007. 50

[24] Chien-Chung Huang. Circular stable matching and 3-way kidney transplant. *Algorithmica*, 58(1):137–150, 2009. 3, 50, 63

[25] J.S. Hwang. The algebra of stable marriages. *Intern. J. Computer Math.*, 20(3-4):227–243, 1986. 20

[26] Robert W. Irving. An efficient algorithm for the "stable roommates" problem. *J. Algorithms*, 6(4):577–595, 1985. 21, 23, 24, 25, 27, 28

[27] Robert W. Irving. Stable marriage and indifference. *Discrete Appl. Math.*, 48(3):261–272, 1994. 50

[28] Robert W Irving and Paul Leather. The complexity of counting stable marriages. *SIAM J. Comput.*, 15(3):655–667, 1986. 18, 20

[29] Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987.

[30] Robert W. Irving and David F. Manlove. The stable roommates problem with ties. *Journal of Algorithms*, 43(1):85 – 105, 2002.

[31] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *ICALP*, pages 443–452, 1999.

[32] Richard Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85 – 103. Plenum Press, 1972. 9, 44

[33] Donald E. Knuth. *Mariages stables et leurs relations avec d'autres problèmes combinatoires*. Les Presses de l'Université de Montréal, Montreal, Quebec, 1976. 3, 16, 18, 19, 20, 21, 63

[34] David F. Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theor. Comput. Sci.*, 276(1-2):261–279, 2002. 30

[35] D. G. McVitie and L. B. Wilson. Stable marriage assignment for unequal sets. *BIT*, 10(3):295–309, 1970. 16

[36] D. G. McVitie and L. B. Wilson. The stable marriage problem. *Commun. ACM*, 14(7):486–490, 1971. 17

[37] Cheng Ng and Daniel S. Hirschberg. Three-dimensional stable matching problems. *SIAM Journal on Discrete Mathematics*, 4:4–245, 1991. 48, 64

[38] National Resident Matching Program. `http://www.nrmp.org/`, 2010. 16

[39] Caroline Rioux. Scarf's theorem and applications in combinatorics. Master's thesis, University of Waterloo, 2006. 32, 33

[40] Eytan Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11(2):285 – 304, 1990. 30

[41] Alvin E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92:991 – 1016, 1984. 16

[42] Alvin E. Roth, Uriel G. Rothblum, and John H. Vande Vate. Stable matchings, optimal assignments, and linear programming. *Math. Oper. Res.*, 18(4):803–828, 1993. 43, 44, 45

[43] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2):151 – 188, 2005. 3

[44] Alvin E. Roth and Marilda Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1990.

[45] Uriel G. Rothblum. Characterization of stable matchings as extreme points of a polytope. *Math. Program.*, 54(1):57–67, 1992. 44

[46] Herbert E. Scarf. The core of an n person game. Cowles Foundation Discussion Papers 182R, Cowles Foundation, Yale University, 1965. 32, 33

[47] Jimmy J. M. Tan. A maximum stable matching for the roommates problem. *BIT*, 30(4):631–640, 1990. 39, 41

[48] Jimmy J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991. 34, 35, 36, 37, 39, 40, 41

[49] Chung-Piaw Teo and Jay Sethuraman. The geometry of fractional stable matchings and its applications. *Math. Oper. Res.*, 23(4):874 – 891, 1998.

[50] Co-operative Education University of Waterloo. Co-op student manual - ranking. `http://www.cecs.uwaterloo.ca/manual/first_cycle/4_11.php`, 2010. 16

[51] John H. Vande Vate. Linear programming brings marital bliss. *Operations Research Letters*, 8(3):147–153, 1989. 43, 44