B. A. Reed

Graph Minors I:

Rooted Routing

January 11, 2013

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Contents

1 An Introduction to Routing ... 1

2 An Introduction to Graph Minors ... 15

3 K_5 Minors and the 2-DRP Problem ... 27

4 Planar Routing: The Algorithm ... 51

5 Planar Graphs Revisited: Basics and Base Cases ... 63

6 Planar Routing: The Key Lemma ... 83

7 Irrelevant Vertices ... 93

8 A Connectivity Measure ... 103

9 Introducing Tree Decompositions ... 115

10 A Duality Theorem ... 129

11 Tangles, Separations, and Canonical Tree Decompositions ... 133

12 Excluding Walls ... 151

13 Using Bounded Width Tree Decompositions ... 169

14 From Bounded Width Tree Decompositions To Minimum Width Tree Decompositions ... 181

15 Finding Bounded Width Tree Decompositions ... 199

16 Staring at The Walls ... 209

17 The Algorithm 229

18 Producing Portfolios 245

19 Isolated Vertices and Homogeneous Walls 253

20 Unique Linkages 255

21 A Sketch Proof 273

22 Wagner's Conjecture 275

23 A Structure Theorem 277

References 279

1

An Introduction to Routing

1.1 Some Classical Results

Finding ways of getting from one place to another has occupied humanity since the dawn of civilization. The Greeks drew straight lines, the Romans built roads, Moses parted the waters, and Dorothy clicked her heels (cf. *The Wizard of Oz*). Mathematicians settle for studying graphs.

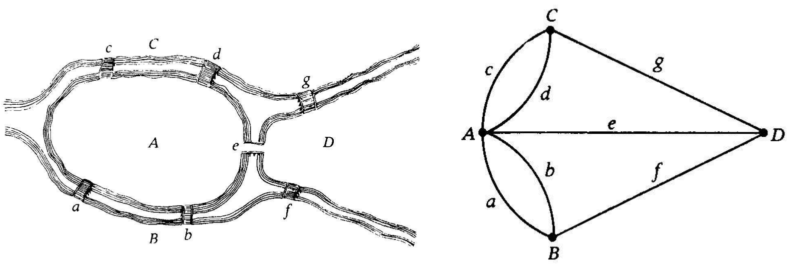
Recall that a *graph* is a set of vertices and a set of edges, each of which links a pair of vertices. Thus, graphs may abstractly represent the highways (edges) linking a set of cities (vertices), the bridges linking the islands of an archipelago, or the wires of a telephone network. We insist that graphs have at most one edge between each pair of vertices. If we want to allow multiple edges we make this explicit by saying we are considering a *multigraph*.

Since graphs are such a natural model of physical networks, it is not surprising that the problem commonly referred to as the oldest in graph theory concerns routing. In particular, it deals with the existence of a route through the Prussian town of Königsberg which traverses each bridge exactly once. The problem was solved by Euler in 1736, for the bridges that were standing at that time. In fact, he gave a general procedure for resolving problems of this type.

We reproduce below the opening portion of Euler's article in which he formulates the problem. This passage is taken from an excellent book on the history of graph theory, written by Biggs, Lloyd and Wilson. Readers should consult [9] for a translation of the complete article, and a fuller discussion of the history of this problem and the solution Euler obtained.

> 1. In addition to that branch of geometry which is concerned with magnitudes and which has always received the greatest attention, there is another branch, previously almost unknown, which Leibniz first mentioned, calling it the *geometry of position*. This branch is concerned only with the determination of position and its properties; it does not involve measurements, nor calculations made with them. It has not yet

been satisfactorily determined what kind of problems are relevant to this geometry of position, or what methods should be used in solving them. Hence, when a problem was recently mentioned, which seemed geometrical but was so constructed that it did not require the measurement of distances, nor did calculation help at all, I had no doubt that it was concerned with the geometry of position — especially as its solution involved only position, and no calculation was of any use. I have therefore decided to give here the method which I have found for solving this kind of problem, as an example of the geometry of position.
2. The problem, which I am told is widely known, is as follows: In Königsberg in Prussia, there is an island A, called the *Kneiphof*; the river which surrounds it is divided into two branches, as can be seen in Fig. 1.1(a) [actually the figure was labelled differently in the original article], and these branches are crossed by seven bridges, a, b, c, d, e, f and g. Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he would cross each bridge once and only once. I was told that some people had asserted that this was impossible, while others were in doubt; but nobody would actually assert that it could be done. From this, I have formulated the general problem: whatever be the arrangement and division of the river into branches, and however many bridges there be, can one find out whether or not it is possible to cross each bridge exactly once?



(a) (b)

Fig. 1.1. The Seven Bridges of Königsberg

3. As far as the problem of the seven bridges of Königsberg is concerned, it can be solved by making an exhaustive list of all possible routes, and then finding whether or not any route satisfies the conditions of the problem. Because of the number of possibilities, this method of solution would be too difficult and laborious, and in other problems with more bridges it would be impossible. Moreover, if this method is followed to its conclusion, many irrelevant routes will be

found, which is the reason for the difficulty of this method. Hence I rejected it, and looked for another method concerned only with the problem of whether or not the specified route could be found; I considered that such a method would be much simpler.
4. My whole method relies on the particularly convenient way in which the crossing of a bridge can be represented. For this I use the capital letters A, B, C, D, for each of the land areas separated by the river. If the traveller goes from A to B over bridge a or b, I write this as AB -where the first letter refers to the area the traveller is leaving, and the second refers to the area he arrives at after crossing the bridge. Thus if the traveller leaves B and crosses into D over bridge f, this crossing is denoted by BD, and the two crossings AB and BD combined I shall denote by the three letters ABD, where the middle letter B refers both to the area which is entered in the first crossing and to the one which is left in the second crossing.
5. Similarly, if the traveller goes on from D to C over the bridge g, I shall represent these three successive crossings by the letters $ABDC$, which should be taken to mean that the traveller, starting in A, crosses to B, goes on to D, and finally arrives in C. Since each land area is separated from every other by a branch of the river, the traveller must have crossed three bridges. Similarly, the successive crossing of four bridges would be represented by five letters, and in general, however many bridges the traveller crosses, his journey is denoted by a number of letters one greater than the number of bridges. Thus the crossing of seven bridges requires eight letters to represent it.
6. In this method of representation, I take no account of the bridges by which the crossing is made, but if the crossing from one area to another can be made by several bridges, then any bridge can be used, so long as the required area is reached. It follows that a journey across the seven bridges of Fig. 1.1(a) can be arranged in such a way that each bridge is crossed once, but none twice, then the route can be represented by eight letters which are arranged so that the letters A and B are next to each other twice, since there are two bridges, a and b, connecting the areas A and B; similarly, A and C must be adjacent twice in the series of eight letters, and the pairs A and D, B and D, and C and D must occur together once each.
7. The problem is therefore reduced to finding a sequence of eight letters, formed from the four letters A, B, C, D, in which the various pairs of letters occur the required number of times.

We remark that although Euler referred to this problem as one in Leibniz's "geometry of position" (which is the earliest name for topology), we can see from the formulation in 4. and the modern equivalent Fig. 1.1(b), of his Fig. 1.1(a) that it is more specifically a problem in graph theory (the geometry of connection?). Moreover, Euler recognizes the importance of the graph theoretic formulation, stating in 4. that his whole method relies on it. We remark further that in 4. and 5., Euler essentially defines a path in a graph (this being arguably the first definition in graph theory), and then in 6. remarks that although a path is a sequence of vertices and edges, he will denote it just by the sequence of vertices. To the author's mind this indicates a healthy disrespect for notation. We will adopt the same specific policy, and the same disrespect for the formal niceties of notation. Finally, we note that the comments in 3., concerning the computational complexity of the problem solved, sound remarkably modern for a paper written in Latin in 1736.

The Königsberg Bridge Problem was not the only problem concerning routing in graphs which was studied before the term graph was coined. T.P. Hamilton, a mathematician in Victorian England, developed and sold a toy known as the Icosian Game in which players were expected to find routes passing through the graph in Fig. 1.2 below (the incidence graph of the Dodecahedron) which visited each vertex exactly once, returned to the starting point, and satisfied various other criteria.

Fig. 1.2. Hamilton's Icosian Game

Nowadays, a *Hamilton cycle* in a graph is a cycle which passes through all of the graph's vertices exactly once (actually Hamilton cycles were first studied by the same T.P. Kirkman who studied Steiner triple systems before Steiner, see [9] for more of the history of this topic).

It was Hamilton who discovered the existence of non-commutative algebras. His interest in routing in the Dodecahedron was prompted by his discovery of a non-commutative algebra defined on the paths through it. We can only speculate as to the impetus for the posing of the Königsberg Bridge problem: the probing questions of a young child? a long walk on a lazy Sunday afternoon? The motivation for the study of many routing problems is more immediate. Often their solution may mean profits made, time gained, or wars won.

A consultant trying to solve such a problem may well have a client who wants the solution yesterday. Alternatively, he may have a more far-seeing client who wants to be furnished with an efficient procedure for solving similar problems which may arise in the future. Fortunately, for many applied problems, such algorithms exist. In particular, there are polynomial time algorithms for solving the two fundamental problems stated below.

Problem 1: *Given a graph G, integer k, and two vertices s and t, determine if there are k paths from s to t sharing no vertices except their endpoints (such paths are called internally vertex disjoint).*

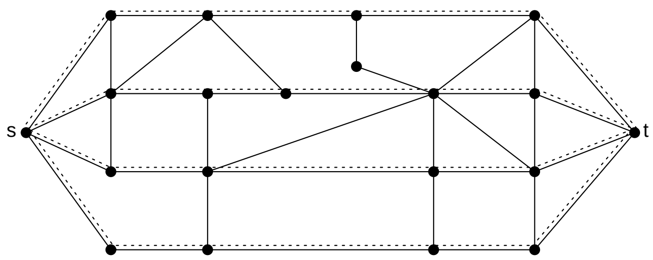
Problem 2: *Given a graph G, integer k, and two sets of vertices S and T, determine if there are k vertex disjoint paths from S to T in G.*

Menger[46] studied these questions in 1927 (for a discussion of how his work relates to earlier work of König and Frobenius, see [41], the introduction and first two chapters). He proved:

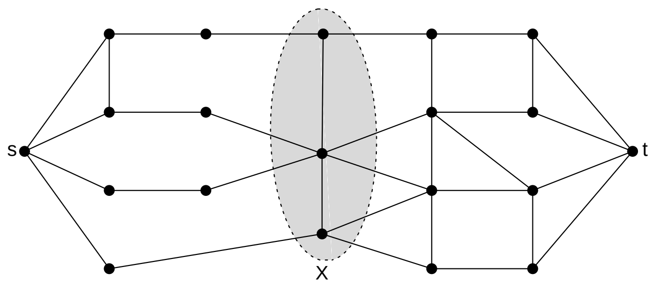
Theorem 1.1. (Menger's Theorem) *Let G be a graph, k be an integer and s, t be two vertices of G. There are k internally vertex disjoint paths of G from s to t if and only if there is no set X of fewer than k vertices disjoint from $\{s, t\}$, such that there are no paths from s to t in $G - X$ (see Figure 1.3).*

Remark. *It is clear that if such a set X exists then each path from s to t must use a vertex of X and hence the desired k internally vertex disjoint paths cannot exist. The difficult part of Menger's Theorem is to show that if k internally vertex disjoint paths do not exist then we can certify this fact using a small set X of vertices separating s from t.*

Corollary 1.2. *Let G be a graph, k be an integer, and S and T be two sets of vertices of G. Then there are k vertex disjoint paths of G from S to T if and only if there is no set X of fewer than k vertices such that there are no paths from S to T in $G - X$.*



(a) four paths



(b) $|X| = 3$

Fig. 1.3. Menger's Theorem

Proof. We simply add a vertex s adjacent to all the vertices of S and a vertex t adjacent to all the vertices of T and apply Menger's Theorem. ■

Now, efficient algorithms for Problem 1 were developed by Ford and Fulkerson in 1956 ([25],[26]). Actually, they presented an algorithm for solving more general network flow problems, in which more than one path can pass through a vertex and so to each vertex there is associated an integer capacity indicating how many paths run through it. (Work in a similar spirit is due to König and Egerváry, again see [41] page xix.)

This algorithm is very versatile. It can of course be used to solve routing problems, however it also has applications to scheduling, resource allocation, plant location, tax avoidance, and many other optimization problems. We refer the reader to [8] for some sample applications, descriptions of more recent faster algorithms for the same problem, and lists of references. We will use only the vanilla flavour of the algorithm which solves instances of Problems 1 and 2 in $O(k|E(G)|)$ time.

So, not only do we have a beautiful min-max theorem characterizing which instances of Problem 2 have a solution, we also have a simple efficient algorithm for solving it which is widely applicable. We could not really hope for more.

1.2 Rooted Routing

In this book, we study a slightly different problem.

Disjoint Rooted Paths: *Given a graph G, integer k, and two sets of vertices $S = \{s_1, ..., s_k\}$ and $T = \{t_1, ..., t_k\}$, determine if there are k vertex disjoint paths $P_1, ..., P_k$ from S to T in G such that P_i has endpoints s_i and t_i.*

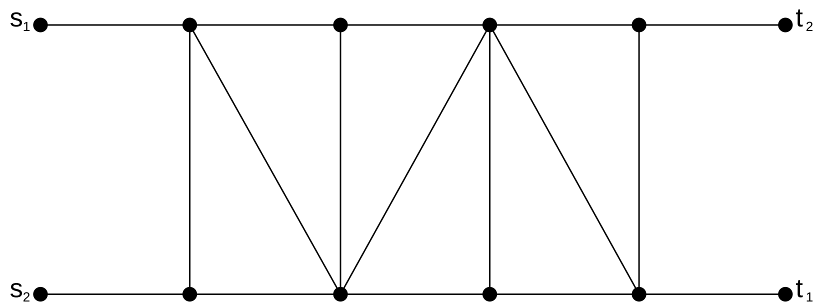


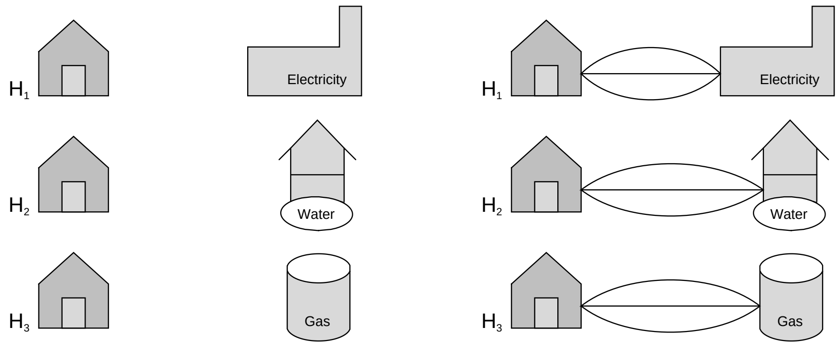
Fig. 1.4. An instance of DRP

Figure 1.4 illustrates the difference between this problem and Problem 2 as two disjoint S-T paths exist, but the desired disjoint P_1 and P_2 do not. To drive home the distinction, we consider a few sample problems with the same flavour.

1) A well known children's puzzle concerns the diagram in Fig. 1.5(a) The child is told that each of the three houses must be connected to each of the three utilities. He is asked to determine if there is a way of drawing these connections so that no two of them cross. As the reader is invited to verify, this is impossible. However, if the problem called only for three connections between each house and the utility stations, then the diagram in Fig. 1.5(b) would be a feasible solution. The problem is difficult because for each of the nine required connections we have specified which house must be linked with which utility.

2) Often, in the production of computer chips, thousands of pairs of "pins" must be wired together. (Actually usually more than two pins must be joined by many of the "nets" of wires.) This yields a Rooted Routing problem, sets of paths linking the wrong pairs of pins are useless. (Actually, we often want to find linkings which satisfy certain other criteria: e.g. minimizing the total amount of wire used.)

3) An airline company discovers that one of its planes has mechanical difficulties and hence a flight must be cancelled. Passengers who intended to travel on this flight need to be rerouted using available seats on other flights. Each passenger corresponds to a source city and a destination city which must be linked via a sequence of flights. This can be modelled as a rooted routing problem in a *directed graph* in which the vertices correspond to cities and each flight links its origin to its destination. Again, the correct linking of the endpoints is crucial to the solution of the problem. Obviously, if there is a passenger in Australia who wants to travel to New York and another in Mexico City who wants to travel to Paris, then sending the passenger in Australia to Paris and the passenger in Mexico City to New York will satisfy neither.



(a) (b)

Fig. 1.5.

Unfortunately, Disjoint Rooted Paths is not as tractable as the Network Flow problems discussed above. For example, it is known to be NP-complete (see [35]) and hence no efficient algorithm for solving it is likely to exist. This suggests that the characterization of which instances of Disjoint Rooted Paths have solutions may be uglier than Menger's theorem. Nevertheless, the study of this problem has led to results which I believe are just as fundamental and beautiful as Menger's Theorem, if not as simple. In this book, we study some of this work. The focus of the book is Robertson and Seymour's polynomial time algorithm to solve Disjoint Rooted Paths [62], and some variants, for fixed k. That is, we will present a proof of the following result:

Theorem 1.3. *For any fixed positive integer k, there are polynomial time algorithms for the following problems:*

Problem: *k-Disjoint Rooted Paths (or k-DRP)*

Instance: *A graph G, and two sets of vertices $S = \{s_1, ..., s_k\}$ and $T = \{t_1, ..., t_k\}$.*
Question: *Are there k vertex disjoint paths $P_1, ..., P_k$ from S to T in G such that P_i has endpoints s_i and t_i?*

Problem: *k-Rooted Routing (or k-RR)*

Instance: *A graph G, and a set X of $2k$ vertices.*
Question: *Solve k-DRP for every instance (G, S, T) such that $S \cup T = X$.*

Problem: *k-Vertex Disjoint Trees (or k-VDT)*

Instance: *A graph G, a set X of at most k vertices of G and a partition $\Delta = \{Y_1, ..., Y_p\}$ of X.*
Question: *Are there vertex disjoint trees $T_1, ..., T_p$ in G such that $Y_i \subseteq T_i$?*

Problem: *k-Realizations*

Instance: *A graph G, and a set X of at most k vertices of G.*
Question: *Solve k-VDT for every partition of X.*

Remark. *A polynomial time algorithm for $2k$-Realizations can be used to solve k-RR, k-DRP and $2k$-VDT, by looking at the appropriate part of the output.*

Definition. *A partition $\Delta = \{Y_1, ..., Y_p\}$ of $X \subseteq V(G)$ is realizable (in G) if there are vertex disjoint trees $T_1, ..., T_p$ of G such that $Y_i \subseteq V(T_i)$. Any such set of trees is called a realization of Δ.*

A CAVEAT: *We prove the above result modulo a lemma whose lengthy proof we only sketch. We prove the lemma for planar graphs in Chapter 6. We explain the structure of the proof for general graphs in Chapter 21. The author believes that the lemma may have a simpler proof. This and other reasons for omitting the proof of the lemma are discussed at the end of the next chapter. In Chapter 20, we present a number of formulations of the lemma which lead the author to believe that it may be resolvable using different techniques.*

If, for a specific instance of k-DRP, G is sufficiently connected, then the desired paths exist. Indeed, improving on results of Mader [44], Thomason [71] and Kostochka [36] independently proved that if there are $\Omega(k\sqrt{\log k})$

internally vertex disjoint paths between every pair of vertices of G, the desired $\{P_1, ..., P_k\}$ exist.

A relatively simple but important piece of Robertson and Seymour's algorithm for k-DRP is a technique for dealing with instances such that G contains a sufficiently highly connected subgraph. To illustrate this technique we sketch how to apply it to an instance (G, S, T) of k-DRP such that G contains a set C of $2k$ vertices, every pair of which are joined by an edge (such a set of vertices is a *clique*; we use K_l to denote the clique with vertex set $1, ..., l$).

If there are $2k$ vertex disjoint paths between $S \cup T$ and C then, as illustrated in Fig. 1.6(a), the desired paths $P_1, .., P_k$ can be obtained by combining these $2k$ paths with k edges of C. Otherwise, by Menger's Theorem, there is a set X of less than $2k$ vertices such that there is no path between $S \cup T$ and C in $G - X$. That is, $S \cup T$ is disjoint from the component U of $G - X$ containing $C - X$. In fact, Menger's Theorem implies that if we choose such an X with $|X|$ minimum then there is a set $\mathcal{R}$ of $|X|$ vertex disjoint paths from X to some subset C' of C (see Fig. 1.6(b)). By taking minimal paths, and choosing C' accordingly, we can ensure they are disjoint from $C - C'$. Now, for any set $\mathcal{P}$ of k vertex disjoint paths linking S to T, the intersection of the paths in $\mathcal{P}$ with $X \cup U$ must be a set of paths with endpoints in X (since there are no edges from U to $G - X - U$). For any such set of paths with endpoints in X, we can clearly obtain a set of paths with the same endpoints using the paths of $\mathcal{R}$ and the edges between vertices of C'. Thus, if a solution to our instance of k-DRP exists, there is a solution whose intersection with U uses only those vertices on some element of $\mathcal{R}$ and in particular uses none of the vertices of $C - C'$. We can therefore delete the vertices in $C - C'$ without affecting the existence of a solution.

So, we obtain:

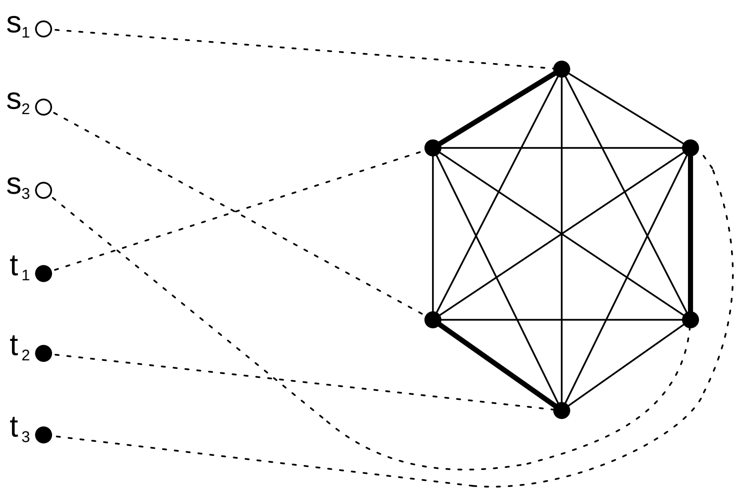
If (G, S, T) is an instance of k-DRP and C is a clique of size at least $2k$ in G then in polynomial time we can either find the desired paths, or find a vertex v such that the desired paths exist in $G - v$ if and only if they exist in G.

It turns out that a similar statement holds if C is a sufficiently large set of disjoint connected subgraphs every pair of which are joined by an edge.

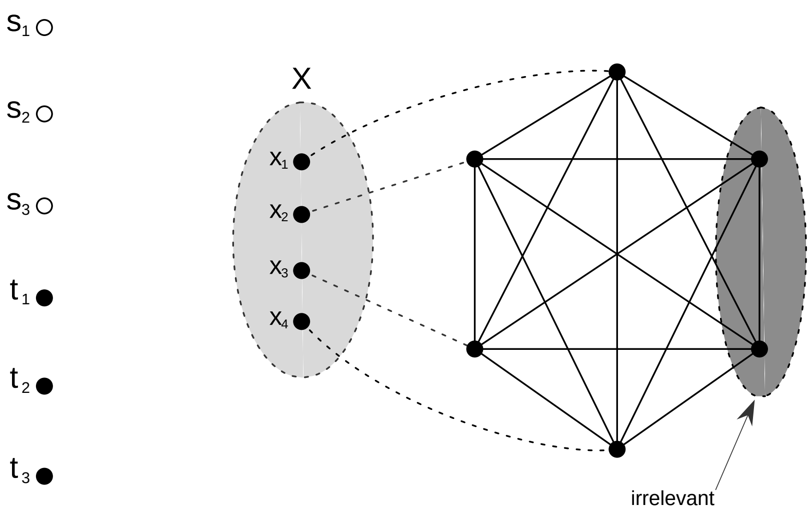
In particular, in Chapter 7 we formally prove (with relative ease):

Lemma 1.4. *If (G, S, T) is an instance of k-DRP, and C is a set of $4k + 1$ disjoint connected subgraphs between every pair of which there is an edge, then in polynomial time, we can either find the desired paths or find a vertex v such that the desired paths exist in G if and only if they exist in $G - v$.*

As we discuss in Chapter 2, graphs which have a set C of l vertex disjoint connected subgraphs between any two of which there is an edge are said to have a *clique minor of order l* (or a K_l minor). Lemma 1.4 suggests our two pronged approach to solving instances of k-DRP . We first attempt to find a



(a)



(b)

Fig. 1.6.

large clique minor in G. If we succeed, we apply Lemma 1.4 which allows us to delete a vertex and recurse. If we fail to find a minor, we will find a structural fault which we can exploit.

This approach works particularly well when applied to instances of 2-DRP. In this special case, by a large clique we mean K_5. A classic result of Wagner [75] says that, provided G is 4-connected (that is, there are four internally vertex disjoint paths between every pair of vertices), it has no K_5 minor precisely if it is planar. Using this result, we can obtain an elegant and efficient algorithm for solving 2-DRP.

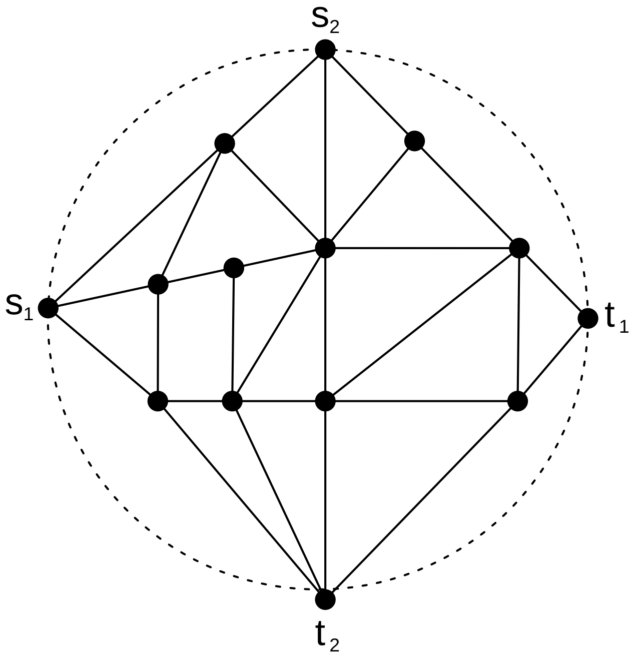
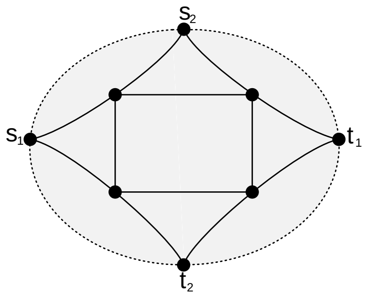
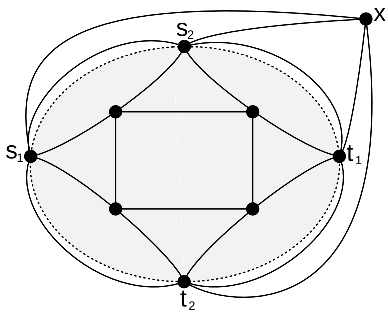


Fig. 1.7.

Figures 1.7 indicates how Wagner's result is used in this algorithm. In Figure 1.7, G is embedded in a disk D so that s_1, s_2, t_1, t_2 appear in the given cyclic order around the boundary. It follows that D does not contain two disjoint arcs A_1 and A_2 such that A_i contains s_i and t_i. This obviously ensures that G does not contain two disjoint paths P_1 and P_2 such that P_i links s_i to t_i. Now, to determine if such a nasty embedding exists, we need only check if an auxiliary graph G' has a planar embedding (see Fig. 1.8). Wagner's result tells us that provided G' is 4-connected, if no such embedding exists



(a) G in a disc



(b) G' in the plane

Fig. 1.8.

then G' contains a K_5 minor. A result analogous to Lemma 1.4 tells us that given such a minor we can find the two desired paths by routing them through the minor.

So, if G' is 4-connected then the desired paths exist if and only if it is non-planar. Thus, to solve 2-DRP in this case, we can apply one of a number of classical planarity testing algorithms to G'. This is the core of our algorithm for 2-DRP as it is straightforward to reduce the general problem to this special case. The details of this algorithm are presented in Chapter 3.

To solve k-DRP for larger values of k with the help of Lemma 1.4, we need to be able to solve it on graphs with no K_{4k+1} minor, as these are precisely the graphs to which we cannot apply the lemma. In particular, we will need to be able to solve it on planar graphs as these have no K_l minor for any $l \geq 5$. An algorithm to solve k-DRP on planar graphs is presented in Chapter 4. Crucial to the algorithm is a lemma, proven in Chapter 6, which states that in any sufficiently robust (in terms of k) instance of k-DRP such that G is a planar graph, we can find, in polynomial time, a vertex v such that the desired paths exist in G if and only if they exist in $G - v$. A similar result holds for graphs with no K_5 minor.

In order to solve k-DRP in general, Robertson and Seymour extended this result to graphs without K_l minors for arbitrary large l. To do so, they needed to develop an understanding of the structure of graphs in which the existence of a K_l minor is excluded.

The above discussion provides a link between Rooted Routing and excluded minor structure theorems. We shall have more to say about the latter in the next chapter.

After having discussed the K_5 minor free case in detail in Chapters 4 through 6 we will have developed enough terminology and machinery to present a fuller discussion of how to generalize to the K_l-free case for larger l. Chapter 7 contains such a discussion.

2

An Introduction to Graph Minors

As the title of this volume suggests, the algorithm for k-Realizations it contains is just one consequence of Robertson and Seymour's research into graph minors. Their results appear in a long series of papers (presently more than 25 papers in the series have been written or are in preparation); the purpose of the present work is to survey this seminal contribution. In this chapter, we content ourselves with making a few definitions, stating the most important results obtained by Robertson and Seymour, and linking these results more closely to Rooted Routing, the ostensible subject of this volume. We will have more to say about these results once we have developed enough of the theory required to make a fuller discussion possible (specifically, in Chapter 13 and Chapter 21). We begin our discussion by defining minors.

We *contract* an edge xy in a graph G to obtain a new graph G_{xy} with vertex set $V(G_{xy}) = V(G) - x - y + (xy^*)$ and edge set $E(G_{xy}) = E(G - y - x) \cup \{(xy^*)z | xz \text{ or } yz \in E(G)\}$ (see Fig 2.1). H is a *minor* of G if H can be obtained from G via a sequence of edge deletions, edge contractions, and vertex deletions (we regard isomorphic graphs as equal).

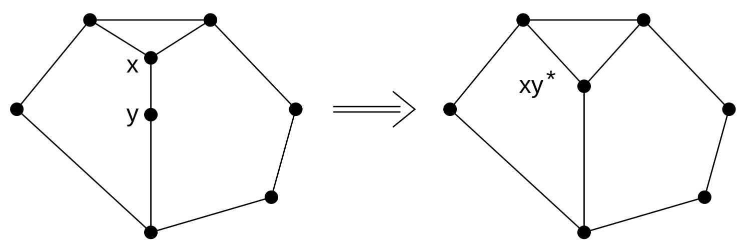
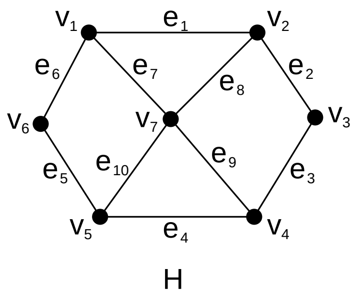
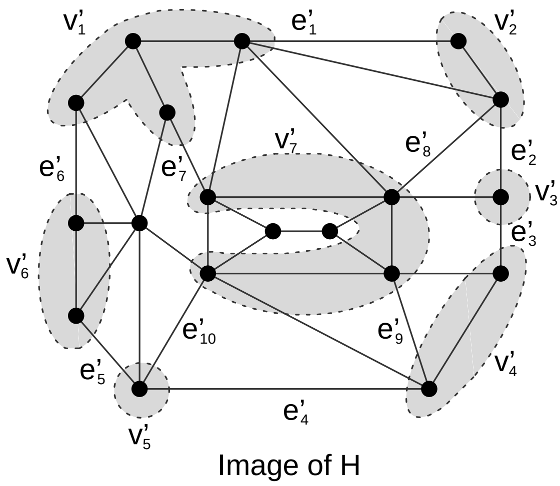


Fig. 2.1. Contracting an edge

We note that if H is a minor of G and F is a minor of H then by concatenating two sequences we have that F is a minor of G. We shall use the rather cumbersome phrase “transitivity of the minor relation” to refer to this fact. A minor H of G is *proper* if it is not G itself.



(a)



(b)

Fig. 2.2.

In order to strengthen her grasp on the minor concept, the reader may wish to actually see how a minor H of G is contained in G. The following definition and Figure 2.2 should help. Consider a function im with domain $E(H) \cup V(H)$ such that:

1. $\forall v \in V(H),\ im(v)$ is a tree of G,
2. $\forall e = uv \in E(H),\ im(e)$ is an edge of G with one endpoint in $im(u)$ and the other in $im(v)$,
3. for all distinct $v,v' \in V(H)$, $im(v)$ and $im(v')$ are disjoint,
4. by the last two statements, $\forall e \neq e' \in E(H), v \in V(H) :\ im(e)$ is not an edge of $im(v)$ and $im(e) \neq im(e')$.

By the *image* of a vertex v we mean the tree $im(v)$. By the *image* of an edge e we mean the edge $im(e)$. By the *image* of H we mean the subgraph obtained by taking the union of the image of all the vertices and edges of H. We refer to a function im satisfying the above conditions as a *model* of H in G.

We note that if there is a model of H in G then G has H as a minor, as can be demonstrated by contracting all the edges of all the $im(v)$ and deleting any edge or vertex which is not in the image of H. The converse is also easily seen to be true. Thus, there is a model of H in G if and only if G has H as a minor.

Minors have received a great deal of attention in the literature. One motivation for studying minors is the link between minor theory and the Tutte polynomial of a graph, also defined in terms of deletions and contractions. Another is that many classes of graphs (and more generally matroids) can be characterized in terms of forbidden minors. We shall focus on the latter type of result.

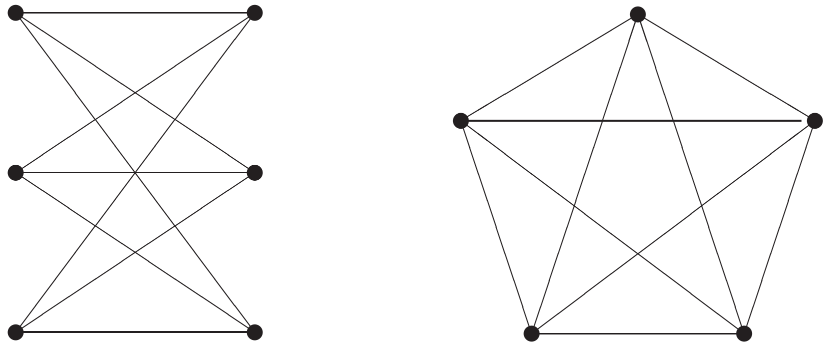
A class of graphs $\mathcal{F}$ is called *minor closed*, if all the minors of each element of $\mathcal{F}$ are also in $\mathcal{F}$. For a minor closed family $\mathcal{F}$ of graphs we define the *obstruction set* for $\mathcal{F}$, denoted $\mathcal{O}_{\mathcal{F}}$, to be $\{H | H \notin \mathcal{F}$ but all proper minors of H are in $\mathcal{F}\}$. The name is justified by the following:

Observation 2.1 *For any minor closed family $\mathcal{F}$, a graph G is in $\mathcal{F}$ if and only if it contains no element of the obstruction set as a minor.*

Proof. The fact that $\mathcal{F}$ is minor closed ensures that no element of $\mathcal{F}$ contains a minor in $\mathcal{O}_{\mathcal{F}}$. The transitivity of the minor relation implies that any graph not in $\mathcal{F}$ contains a member of $\mathcal{O}_{\mathcal{F}}$ as a minor. ■

The archetypical example of such a class of graphs is the planar graphs. A graph is *planar* if it can be drawn in the plane in such a way that no two of its edges cross. Clearly, if G is planar then every subgraph of G is also planar. Furthermore, given a drawing of G in the plane we can obtain a drawing of G_{xy} for some edge xy, by pulling x and y together in the edge. Thus the planar graphs are minor closed. The following well-known result, proved in the 1930s, shows that the obstruction set for this class is very simple.

Theorem 2.1. (Kuratowski's Theorem [38]) *A graph is planar if and only if it has neither $K_{3,3}$ nor K_5 as a minor (see Figure 2.3).*



$K_{3,3}$ K_5

Fig. 2.3. Kuratowski's forbidden minors

Remark. *Actually Kuratowski stated his result in terms of subdivisions (defined below) not minors. However, as we shall see, these two theorems are equivalent.*

In the 60s, Tutte [73] developed a polynomial time algorithm for recognizing planar graphs, thereby giving another "good" characterization of this class.

Robertson and Seymour show that every minor closed class has a simple (i.e. finite) obstruction set and use this result to prove there is a polynomial time algorithm to test membership for every minor closed class. Thus, they show that all minor closed classes are practically as well behaved as the planar graphs. Specifically they prove:

Wagner's Conjecture. *If $G_1, G_2, G_3, \ldots$ is an infinite sequence of graphs then there exist $i \neq j$ such that G_i is a minor of G_j.*

Theorem 2.2. *For any fixed graph H, there is a polynomial-time algorithm which solves the following decision problem:*

Problem: *H-Minor Containment*
Instance: *A graph G.*
Question: *Is H a minor of G?*

They obtain as a corollary:

Corollary 2.3. *If C is a minor closed class of graphs then there is a polynomial-time algorithm for testing membership in C.*

Proof. Let $\mathcal{F}$ be a minor closed family of graphs. Wagner's Conjecture implies that $\mathcal{O}_{\mathcal{F}}$ is finite. Theorem 2.2 implies that for any fixed $H \in \mathcal{O}_{\mathcal{F}}$, we can check for H-Minor Containment in polynomial time. Corollary 2.3 follows. ∎

Remark. *Minor Containment is clearly NP-complete if H is part of the input as the NP-complete problem of determining if a graph has a Hamilton Cycle is a special case (since if we have a cycle C on $|V(G)|$ vertices as a minor of G, we cannot have performed any contractions in obtaining C).*

Remark. *The proof of Wagner's Conjecture is found in [64]. The proofs of Theorems 2.2 and 1.3 are found in [62]. The algorithms of Robertson and Seymour for H-Minor Containment, k-Realizations and testing membership in minor closed classes run in $\mathcal{O}(|V(G)|^3)$ time. Reed developed similar algorithms for the same problems which run in $\mathcal{O}(|V(G)|^2)$ time and are presented for the first time in this volume.*

The results we have just stated have hordes of implications, we mention just a few of them here. To do so, we need to define some minor closed classes of graphs. For any surface Σ, the class C_Σ of graphs embeddable in Σ is clearly closed under the taking of minors (in fact, minors were defined by Tutte [72] who used them extensively whilst studying planar graphs). Now, as we mentioned above, Kuratowski [38] showed that the obstruction set for planar graphs consists of the two graphs $K_{3,3}$ and K_5. Archdeacon and Huneke [3] proved that for any non-orientable surface Σ the obstruction set for C_Σ is finite. Clearly, Wagner's Conjecture implies that for any surface Σ, the obstruction set for C_Σ is finite (this result actually appears in [60]). Furthermore, Corollary 2.3 implies that one can test for any fixed surface Σ whether G is embeddable in Σ in polynomial time and, by our remarks, in $\mathcal{O}(|V(G)|^2)$ time. Polynomial time algorithms for this problem had already been developed [24] but the exponent of their running times depended on Σ.

A graph is called *linklessly embeddable* if it can be embedded in three space so that no pair of disjoint cycles form a link (as do for example two consecutive links in a chain). Clearly, if a graph is linklessly embeddable so are all its minors. Thus, Corollary 2.3 implies that there is an algorithm to determine if G is linklessly embeddable which runs in polynomial time. Previously, we had not even known if any such algorithm existed, i.e. the linkless embeddability decision problem was not known to be decidable. More recently, Robertson, Seymour, and Thomas [65] constructed the obstruction set for the linklessly embeddable graphs.

Other problems to which Corollary 2.3 can be applied include: gate matrix layout, topological bandwidth, disk dimension, and vertex integrity, see [22] and [23] for details.

We remark that Corollary 2.3 only implies the existence of an algorithm to determine if a graph is in a minor closed class $\mathcal{C}$. It does not tell us how to construct such an algorithm, because we may not know how to find the obstructions.

As we mentioned in Chapter 1, Rooted Routing and graph minor theory are linked because Robertson and Seymour's algorithm for k-DRP uses a structure theorem for graphs without large clique minors. Actually, the relationship between these two fields is much closer than this one fact suggests.

For example, as we now show, a polynomial time algorithm for k-DRP can be used to prove Theorem 2.2. To do so, we will need to consider subdivisions.

To *subdivide* an edge e in a graph H, we replace it by a path of length two through a new vertex. A *subdivision* of a graph H, consists of a graph obtained from H by repeatedly subdividing edges. That is, a graph in which each edge of H has been replaced by a path with the same endpoints such that these paths can share endpoints but are otherwise disjoint. We say G contains a *subdivision* of H if there is a subgraph of G isomorphic to a subdivision of H. We refer to such a subgraph as an *smodel* of H. We refer to the vertices of this smodel which correspond to the vertices of H as the *centres* of the smodel.

Subdivisions and minors are related by the following result:

Lemma 2.4. *If G contains a subdivision of H then it has H as a minor.*

Proof. Assume G contains a subdivision of H, and let F be an smodel of H in G. For each edge e of H arbitrarily choose some edge of the path of F corresponding to e to be $\mathrm{im}(e)$. Deleting the chosen edges decomposes F into components each containing one centre. For each vertex v of H we let $\mathrm{im}(v)$ be the component containing the centre corresponding to v. This yields a model of H in G and hence H is a minor of G. ■

If G has maximum degree three then the converse of Lemma 2.4 is true and we obtain:

Lemma 2.5. *If H is a graph of maximum degree 3 then G has H as a minor if and only if G contains a subdivision of H.*

Proof. We need only prove that if G has H as a minor then it contains a subdivision of H. Now G contains an H-minor precisely if it contains a model of H. We choose a model im so that the union of the edge sets of the vertex images is as small as possible. This implies that, for every vertex v of H, every leaf of $im(v)$ is incident to the image of an edge of H.

If $im(v)$ has three leaves, then it has a unique vertex of degree three. We let u_v be this vertex and note that the following holds:

(*) There are paths from u_v to the endpoints of all the image edges in $im(v)$ disjoint except at u_v.

Otherwise, $im(v)$ is a path. If there is an internal vertex of this path incident to an edge image, we let u_v be this vertex and note that (*) holds (one of the paths is just u_v). If there is no such internal vertex, we let u_v be the endpoint of $im(v)$ incident to the largest number of edge images and note that (*) holds.

Since we have chosen u_v such that (*) holds for all v, we have an $s$model of H where u_v is the centre corresponding to v. ■

Of course, this lemma does not hold for general H as the graphs in Figure 2.4 show. However, a weaker theorem in the same vein does hold for all graphs. Before proving it, we warm up by proving an observation made earlier.

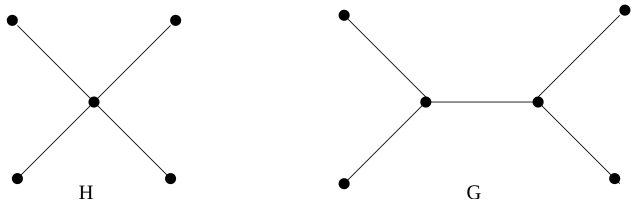


Fig. 2.4.

2.6 (An Aside) *G has one of $K_{3,3}$ or K_5 as a minor if and only if it has one of $K_{3,3}$ or K_5 as a subdivision, thereby proving that our formulation of Kuratowski's theorem is equivalent to his original formulation.*

Proof. Since $K_{3,3}$ is cubic, if G has $K_{3,3}$ as a minor then it also contains a $K_{3,3}$ subdivision and we are done. Suppose then that G has a K_5 minor. As in the proof of Lemma 2.5 above, choose a minimal model of K_5 in G. Now for each $v \in K_5$, $im(v)$ has at most four leaves. If for each v, we can choose u_v in $im(v)$ and four internally vertex disjoint paths from v to the endpoints of the image edges in $im(v)$ then G has a K_5 $s$model with these vertices as centres. So, we can assume that for some v this is not the case.

It follows that the tree formed by $im(v)$ and the image of the four edges of K_5 incident to v, which has exactly four leaves, has no vertex of degree four but rather two vertices of degree 3. We let e be any edge on the path of $im(v)$ between these two vertices and consider the minor H of G obtained from the image of K_5 by contracting all the edges of the vertex images except e. H consists of a 4-clique C and two adjacent vertices joined to disjoint pairs of vertices of C. So H and hence G contain $K_{3,3}$ as a minor and hence has a subdivision of $K_{3,3}$. ■

And now, a more general theorem of the same kind.

Lemma 2.7. *For any graph H, there is a finite set $Z(H)$ of graphs such that G has H as a minor if and only if G contains a subdivision of some element of $Z(H)$.*

Proof. Consider a minimal model im of H in G. By minimality, if $d_H(v) \neq 0$ then every leaf of $im(v)$ is incident to the image of some edge of H. Thus $im(v)$ has at most $max(d_H(v), 1)$ leaves. Since the number of edges in a tree with n nodes is $n-1$, it follows by summing degrees that $im(v)$ has at most $d_H(v)$ nodes of degree exceeding 2. So, the set $\mathcal{S}_v$ of vertices of $im(v)$ which either have degree different than two in $im(v)$ or are incident to the image of an edge of H contains at most $2d_H(v) + 1$ elements. We let T_v be the tree with vertex set $\mathcal{S}_v$ in which there is an edge between x and y if there is a path of $im(v)$ between them which is internally disjoint from $\mathcal{S}_v$. Then $im(v)$ is a subdivision of T_v and $im(H)$ is a subdivision of the graph H'_{im} obtained from

$im(H)$ by replacing each $im(v)$ by T_v. Furthermore, H is a minor of H'_{im}, as can be seen by contracting all of the edges of each T_v. Finally, H'_{im} has at most

$$\sum_{v \in V(H)} 2d_H(v) + 1 \leq 4|E(H)| + |V(H)| \text{ vertices.}$$

Thus, if G has H as a minor, it contains a subdivision of a graph H' which contains H as a minor and has at most $4|E(H)| + |V(H)|$ vertices. The transitivity of the minor relation shows that the converse also holds. ■

(a) H_1 (b) H_2

(c) $Z(H_1)$ (d) $Z(H_2)$

Fig. 2.5. Determining the set $Z(H)$, given H

We remark that we can actually determine the set $Z(H)$ easily given H. We have done so for two example graphs in Figure 2.5. More generally, a graph will be in $Z(H)$ if and only if it can be obtained by choosing for each vertex v of H, a tree T_v with $d_H(v)$ leaves all of whose interior nodes have degree at least three (unless v has degree less than three, in which case T_v is a path with $d(v)$ edges), labelling each pendant edge of T_v by one of the edges out of v and then identifying the edges with the same label so that the interiors of the T_v stay disjoint. We obtain H from such a graph by contracting the edges between the interior vertices of the T_vs.

With Lemma 2.7 in hand, we can now show how to use an algorithm for k-DRP to prove Theorem 2.2. We shall need:

Remark. *An algorithm for k-DRP implies an algorithm for the more general problem where we allow the paths to share endpoints but insist that they are otherwise disjoint (thus S and T may be multi-sets and may intersect). We simply make multiple copies of any vertex which appears more than once in $S \cup T$.*

By Lemma 2.7, if we can test if G has a subdivision of any fixed H in polynomial time then we can test if G has any fixed F as a minor in polynomial time. To test if G has H as a subdivision we need only test, for each of the $\mathcal{O}(n^{|V(H)|})$ injections of $V(H)$ into $V(G)$, whether G has an smodel of H where the given injection specifies the centre corresponding to each vertex. To do this, however, we need only solve (for $k = |E(H)|$) the extension of k-DRP which we have just remarked is no more difficult than k-DRP. Thus, Theorem 1.3 does indeed imply Theorem 2.2.

We note that, in fact, Robertson and Seymour actually proved Theorem 2.2 directly using the same technique they use to prove Theorem 1.3. They thereby obtained an $\mathcal{O}(n^3)$ algorithm for H-Minor Containment. Actually their algorithm solved the following common generalization of H-Minor Containment and k-DRP.

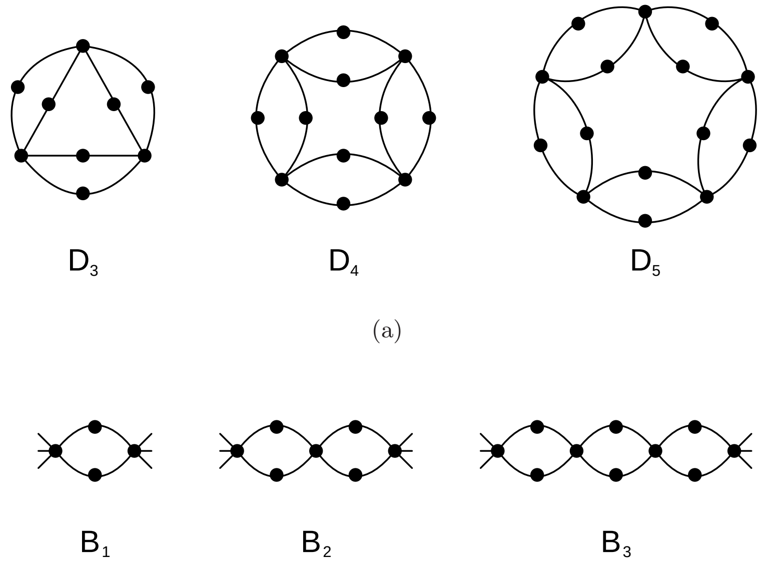
Problem: k-Labelled H-Minor Containment
Instance: A graph G, $X \subseteq V(G)$ with $|X| \leq k$, a function f from X to $V(H)$.
Question: Is there a model im of H in G with $f^{-1}(v) \subseteq im(v)$?

Remark. *An instance of H-Minor Containment is an instance of 0-Labelled H-Minor Containment. An instance of k-DRP can be formulated as an instance of $2k$-Labelled H-Minor containment where H is a stable set with vertices $v_1, ..., v_k$, $X = S \cup T$, and $f(s_i) = f(t_i) = v_i$.*

We shall be content to simply sketch the details of this more general result, secure in the knowledge that our formal presentation of the $\mathcal{O}(n^3)$ algorithm for k-DRP implies a polynomial time algorithm for H-Minor Containment, via a consideration of subdivisions.

Given that we prove Theorem 2.2 by proving the analogous result for subdivisions one might hope to prove Wagner's Conjecture by proving the stronger statement that in any infinite set of graphs one is a subdivision of another. However, this statement is false as the two infinite families in Figure 2.6 show.

Recapitulating, we see that to prove Theorem 2.2, Wagner's Conjecture, and Corollary 2.3, we need only present a polynomial time algorithm for k-DRP (the subject of this volume) and prove Wagner's Conjecture (a proof



(a)

(b)

Fig. 2.6.

sketch appears in Chapter 22). As we discussed in Chapter 1, the algorithm for k-DRP depends heavily on a characterization of the graphs with no large clique minors, which is discussed in Chapter 23. As we show now, so does the proof of Wagner's Conjecture.

To see this, consider a counterexample to Wagner's Conjecture , $G_1, G_2, ...$ Could G_1 possibly be a K_3? We know that for all $j \geq 2$, G_j does not have G_1 as a minor. Thus, if G_1 is a K_3 then each such G_j is a tree (as the reader can easily verify). But Kruskal proved in 1960 [37] that Wagner's Conjecture holds for trees, so we know no counterexample to Wagner's Conjecture has $G_1 = K_3$ or, more generally, has some G_j which is a K_3. Robertson and Seymour prove Wagner's Conjecture, using the same approach. Letting $l = |V(G_1)|$ and applying the transitivity of the minor relation, we see that no $G_j, j \geq 2$ contains K_l as a minor. So, if we had some structure theorem about such graphs, we could apply it to the graphs in the infinite sequence $G_2, G_3, ..,$ to derive a contradiction.

Since it is central to the proof of both Wagner's Conjecture and the routing algorithm, this characterization of graphs with no K_l minor is at the heart of Robertson and Seymour's work. We won't say too much about the characterization here, but we will at least try to give a flavour of the result. If G has no K_5-minor and is 3-connected then one possibility is that G is a planar graph. In fact, a theorem of Wagner[75] tells us that G is a subgraph of a graph obtained by repeatedly gluing together graphs on cliques of size at most 3, beginning with planar graphs (see Fig. 2.7), and one special eight vertex graph (depicted in Fig. 2.8). More generally, if G has no K_l minor then G is a subgraph of a graph which can be built up from graphs which are *nearly embeddable* in a surface in which K_l is not embeddable by repeatedly pasting together graphs on small cliques.

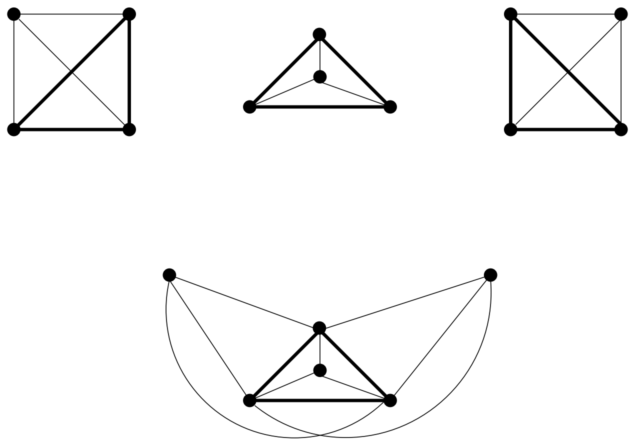


Fig. 2.7. Gluing together a family of planar graphs on triangles

We will not say what nearly embeddable means formally but we will attempt to give a flavour of the notion by considering the special case $l = 6$. Obviously, if G has a K_6 minor then for every vertex v of G, $G - v$ has a K_5 minor. Thus, one family of K_6-minor free graphs are those which can be made planar by the deletion of one vertex. Also, any drawing of K_6 in the plane must have at least three crossings. Thus, any graph which can be drawn in the plane with only two crossings has no K_6 minor. These are the two notions which are used in the definition of near-embeddability. A graph G is nearly embeddable in a surface if deleting a bounded number of vertices of G yields a graph which can be drawn in the surface so that there are only a bounded number of areas at which crossings occur and, furthermore, in each such area the crossings are in some sense bounded.

The most important step in the proof of the excluded minor theorem for cliques is a canonical way of decomposing graphs into their most highly connected pieces. This result is presented in Chapters 8 to 11 of this volume. It allows us to ignore the gluing operation and concentrate on the nearly embeddable pieces of the graph both when proving Wagner's Conjecture and when constructing our routing algorithm.

This near surface embeddability of these pieces is what allows us to extend our lemma about irrelevant vertices for k-DRP in the plane to such graphs (as

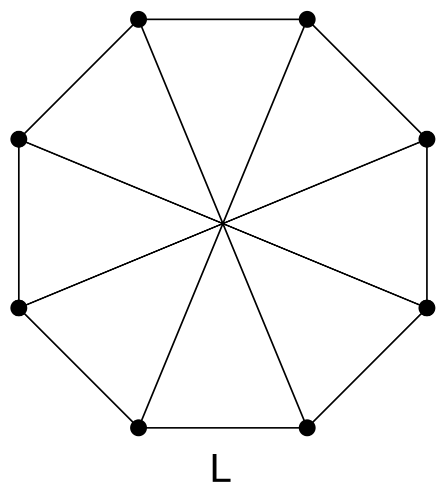


Fig. 2.8. A special eight vertex graph L

mentioned in Chapter 1 and discussed more fully in Chapters 16 and 19). We only sketch the proof of this result. Unfortunately, this requires us to use a lemma about irrelevant vertices in our proof of the correctness of the k-DRP algorithm whose proof is not given in full. Our justifications for omitting the full proof are:

1. The lemma we need seems likely to have a short direct proof and we hope that our presentation encourages researchers to look for such a proof.
2. The discussion in this volume will present all of the key ideas needed in the proof of the excluded minor structure theorem. So, we will be in a position at the end of the volume to present a sketch of this theorem.
3. Our discussion in this volume will not require us to enter into the technicalities of embedding graphs on non-planar surfaces.

This completes our preliminary discussion. Let's roll up our sleeves and get to work.

Notation Index

$(T, \mathcal{W})$ - subtree decomposition by node sets, 108
C_Σ - obstruction set for Σ-embeddability, 19
$Crosses_h$, 97
$G[X]$ - subgraph of G induced by X, 99
G_{xy} - contraction of xy in G, 15
K_l - clique of size l, 10
$V_R \bigcup\{W_s,$ is a node of $R\}$ 111
Z_H, 21
$[T, \mathcal{S}]$ - subtree decomposition, 107
$[T, \mathcal{X}]$ - tree decomposition, 107
$[T_{\mathcal{S}_G}, X_{\mathcal{S}_G}]$ - tree decomposition of G, 128
Σ - a punctured plane, 50
β_S - set of big components, 97
$bn(G)$ - bramble number, 97
$\mathcal{L}$ - hexagonal lattice, 93
$\overline{D}$ - the closure of a disc D, 50
$bd(D)$ - the boundary of a disc D, 50
$bd(\Sigma)$ - the union of the boundaries of the discs deleted from the plane to obtain Σ, 50
im - an image, 17
$int(A)$ - the interior of A in a separation (A, B), 123
$ord((A, B))$ - order of a separation (A, B), 123
$pw(G)$ - path width of G, 113
$tw(G)$ - tree width of G, 108
$wl(G)$ - well-linkedness of G, 134
$\mathcal{O}_\mathcal{F}$ - obstruction set for $\mathcal{F}$, 17
$\mathcal{S}_G$ - set of canonical tangle separations, 128
$\mathcal{T}/Z$ - a tangle of order at least ord$(\mathcal{T})$ in $G - Z$, 136

Author Index

Ackerman, 28
Archdeacon, 19
Arnborg, 116, 162

Biggs, 1
Bodlaender, 162–164

Courcelle, 162

Egervary, 6
Euler, 1, 3

Fellows, 19
Filotti, 19
Ford, 6
Frobenius, 5
Fulkerson, 6

Halin, 116
Hamilton, 5
Huneke, 19

Kirkman, 5
Konig, 5, 6
Kostochka, 9
Kruskal, 24
Kuratowski, 17, 19

Lagergren, 162
Langston, 19
Leibniz, 1, 3
Lipton, 102, 103, 143
Lloyd, 1

Mader, 9
McDiarmid, 164
Menger, 5, 93, 135
Miller, 19

Proskurowski, 116, 162

Reed, 19, 49, 164, 184
Reif, 19
Robertson, 8, 9, 13, 15, 18, 19, 23, 24, 49, 85, 101, 116, 123, 125, 127, 144, 146, 184

Schrijver, 49
Seese, 162
Seymour, 8, 9, 13, 15, 18, 19, 23, 24, 49, 85, 101, 116, 120, 123, 125, 127, 144, 146, 184
Steiner, 5

Tarjan, 143
Thomas, 19, 101, 120, 144, 146
Thomason, 9
Tutte, 17–19, 30

Vizing, 164

Wagner, 12, 18, 23–25
Whitney, 31, 86
Wilson, 1

Subject Index

(X, im) cut, 87, 90, 92
(r, s)-expander, 146, 148
(s, r)-concentrator, 146, 148
0-arc, 53
2-Disjoint Rooted Paths, 12, 13
H-Minor Containment, 18, 19, 23
I-arc, 53
S-separator, 143
h-steps, 146
k-Disjoint Rooted Paths, 9, 10, 13, 19, 20, 23, 26, 85
k-Labelled H-Minor Containment, 23
k-Realizations, 9, 19, 49, 50, 85–88, 91, 94
  1-embedded, 67
  c-Embedded, 50, 52–54
  c-embedded, 86
  planar, 49, 85
k-Rooted Routing, 9, 15, 85
k-Vertex Disjoint Trees, 9
k-connected, 93
k-linked, 103
k-linked set, 96
k-tree, 116
  partial, 116
r-step, 147, 148
$s$model, 20
$\mathcal{T}$-linked, 133–135
$\mathcal{T}$-linked set, 144, 146, 148–156
4-connected, 12

Adjacency Lemma, 164
ancestor, 160
arc
  interior, 53
attached, 87, 89, 145
  strongly, 87, 89

bandwidth, 162
bias, 125
  strong, 125, 126
bias of order k, 124
bicycle, 53
big component, 124
bramble, 97–100, 103, 110, 112, 120, 121, 124–126, 131, 133–135, 141, 143, 144, 146, 149
  distinguishable, 126, 133
    maximal, 133
  distinguisher, 126
  in walls, 101
  indistinguishable, 126, 133
  maximal, 126
    indistinguishable, 131
  number, 97
  of order 4, 99, 100
  of order three, 98
bramble number, 93, 94, 98–100, 105, 110, 119, 120, 125, 126
  and 2-cuts, 99
  and clique cutset, 99
branch width, 125
butterfly, 53

Canonical Tree Decomposition Theorem, 127
child, 160
chordal, 116
chordal graph, 105, 106
chromatic index, 164
chromatic number, 162
clique, 10, 96, 97, 106, 116, 162
clique cutset, 105, 106, 113
clique minor, 86, 98
  of order 4, 100
clique model, 85, 87, 88, 90
  near-clique model, 89
Closest Cutter Lemma, 134, 146, 154
Closest cutter lemma, 134, 154, 156
complete rooted cubic tree, 114

concentrators, 146
connectivity, 134
cross-free partition, 142
cuff, 50
cutset, 12, 93
  trivial, 32
cutter, 134
cylinder, 131

descendant, 160
Disjoint Rooted Paths, 7, 8, 162
domination number, 162
duality theorem, 120

edge, 1
edge contraction, 15
embedding, 31
expanders, 146

forest, 93

graph, 1
  linklessly embeddable, 19
graph isomorphism, 162
grid, 109, 142

Hamilton cycle, 4, 19, 162
Helly Property for Trees, 98, 99, 107
hexagonal lattice, 93, 94
hitting set, 97

image
  of a graph, 17
  of a vertex, 17
  of an edge, 17
independent set, 96
internally vertex disjoint paths, 5

Key Lemma, 86, 87
Kuratowski's Theorem, 17, 21

ladder, 101
linear time, 160, 165
linkedness, 97
lollipop, 53

Menger's Theorem, 5, 10, 93, 122, 135
minor, 15
  K_l, 10
  K_l-free, 13
  clique, 10
  closed, 17–19
  containment, 18, 19, 23
  proper, 16
minor containment, 162

NP-complete, 94

obstruction set, 17

parent, 160
partition
  cross-free, 49, 67
path, 3, 4
path scheme, 162
pathwidth, 113, 115
planar graph, 17, 19, 143
polynomial time, 163
preference
  strong, 125–127
preference of order k, 124
proper arc, 52
  length, 52
puncture plane, 50

realizable, 9, 162
realizable in Σ, 66
realization, 162
robust reduction, 48
root, 159
root r, 160
Rooted Routing, 7, 13, 15, 19
rooted subtree decomposition, 161
rooted tree, 160

schism, 53
separation, 123, 125, 127
  canonical, 131
  laminar, 123
  made by an arc of a tree decomposition, 123

order, 123
sides, 123
interior, 123
separator, 102, 103, 112, 119, 133, 134, 143
S-separator, 112
ϵ-separator, 102, 103
epsilon-separator, 103
smodel, 20
centres, 20
snake, 101
subdivision, 20, 21, 23
subgraph
planar, 87
subpath decomposition, 113
subtree intersection representation, 105, 106
surround, 53

tangle, 103, 124–127, 130, 133–135, 142, 144, 146, 148–155
canonical, 127
conformal, 135
distinguishable, 128
distinguisher, 127, 128, 130
indistinguishable, 131
maximal, 126, 127
separation, 128, 131
canonical, 128, 131
minimal, 128
separator
canonical, 130
tangle number, 125, 126
tangles, 123
touch, 97
tractable, 132
transitivity of the minor relation, 16
tree decomposition, 107, 123, 127, 128, 131
standard, 131
subtree decomposition, 107–112, 116, 119–122
cubic, 113
standard, 115
width, 108
tree width, 107–110, 113, 114, 116, 119, 120

vertex, 1
l-isolated, 86, 87
l-protected, 85, 86
$p(k)$-protected, 87
irrelevant, 53, 85–88, 90, 91
isolated, 53–56
universal, 91
vertex disjoint paths, 5

Wagner's Conjecture, 18, 23–25
wall, 95, 100, 101, 110, 141, 143–146, 149, 151, 152
attached to a tangle, 145
corners, 96
elementary, 95
nails, 96
of height h, 96
perimeter, 96
walnut, 53
weight, 160
well-linked, 135
well-linked set, 141, 142, 155, 157
well-linkedness, 134
Whitney's Theorem, 86