An Algorithm for Finding Hamiltonian Cycles in Grid Graphs Without Holes

by
Christopher M. Umans

A Thesis
Submitted in partial fulfillment of
the requirements for the Degree of Bachelor of Arts with Honors
in Computer Science

WILLIAMS COLLEGE
Williamstown, Massachusetts
May 19, 1996

Abstract

The Hamiltonian cycle problem for general grid graphs is $\mathcal{NP}$-complete. However, it is conjectured that an efficient algorithm to solve the Hamiltonian cycle problem exists for a subclass of general grid graphs known as grid graphs without holes. This thesis contains a number of results concerning the structure of this subclass of graphs which are then used to give a correct polynomial time algorithm for finding Hamiltonian cycles in grid graphs without holes.

Contents

Chapter 1

Introduction

1.1 Hamilton Cycles and Traveling Salesmen

The problem of identifying a tour of a number of cities that visits each city once and returns to the starting city is one that confronts airlines and the Post Office daily, and was of concern to a certain "veteran Traveling Salesman" who in 1832 printed a book in Germany entitled, "The Traveling Salesman, how he should be and what he should do to get Commissions and to be Successful in his Business." His admonishment to "cover as many locations as possible without visiting a location twice" is sound advice and also likely the first reference to what is in essence the Traveling Salesman Problem (TSP), one of the most famous combinatorial optimization problems [6].

The TSP is the problem of identifying a minimum total length tour of a list of cities beginning and ending with the same city. It is commonly modeled as a graph, with vertices representing cities, and edges between adjacent cities that are weighted with pairwise distances. The solution is then a circuit in the graph that visits each vertex exactly once and has minimum weight, if one exists.

If the minimum weight requirement is dropped, then the problem becomes simply the identification of *any* cycle that visits each vertex once. Such a cycle is called a Hamiltonian cycle after the 19th century Irish mathematician Sir William Rowan Hamilton, and a graph containing at least one such cycle is termed Hamiltonian. Hamilton's interest in the subject is linked to his invention of the "Icosian Game" which challenged its players to a number of games based on the Hamiltonian cycles in a graph etched on the wooden playing board. The graph was a representation of an icosahedron, and had a mathematical interpretation related to Hamilton's invention in 1856 of a system of noncommutative algebra he called the "Icosian Calculus." [6]

Hamilton was not the first to study Hamiltonian cycles; a special case known as the "knight's tour," in which a knight must visit all 64 squares of a chessboard moving only in the usual way, was studied by Euler and Vandermonde in the middle to late 1700's [6]. Even the general case was likely investigated prior to Hamilton's work by the Reverend T. P. Kirkman, who had the misfortune of studying Steiner triple systems before Steiner as well [7].

But the Hamiltonian cycle problem is far from a mathematical oddity. With the advent of the computer, such graph-theoretic problems have attained practical significance, as many real-world problems phrased in general terms are equivalent to problems in graph theory. Algorithmic solutions to graph problems are thus the basis for computer programs that accomplish real work, and the Hamiltonian cycle problem is one of the more commonly applicable problems from graph theory. Furthermore, as the next section will discuss, the Hamiltonian cycle problem belongs to a class of problems for which no efficient algorithm is known. The search for an efficient algorithmic solution to the Hamiltonian cycle problem (or, more likely, a proof that one cannot exist) then has theoretical implications as well.

1.2 Computational Complexity

It is very easy to come up with an algorithm to solve the Hamiltonian cycle problem: for a given graph, all of the possible permutations of the list of vertices might be generated, and each checked to see if it represents a Hamiltonian cycle. Unfortunately, this solution is impractical, since the number of possible permutations for even a moderately sized graph of 20 vertices is $20! = 2432902008176640000$, which would require many hundreds of years to generate on even the fastest computer. To make matters worse, the situation deteriorates so rapidly as the size of the input (number of vertices) increases that it overwhelms any foreseeable increase in computing power, so that one can never claim to have a general algorithmic solution in this approach. Clearly a more efficient method must be found. But first, in order to precisely compare the "efficiency" of algorithms, some formal measure of efficiency is required.

1.2.1 $\mathcal{P}$ and $\mathcal{NP}$

The standard measure of the efficiency or speed of an algorithm is its time complexity. To obtain this measure, a function is identified that can be shown to be an upper bound on the worst case performance of the algorithm, where the "performance" is measured in computational steps as a function of the size of the input. Algorithms with a time complexity that is polynomial in the size of the input are considered efficient, while algorithms with superpolynomial time complexity are considered to be inefficient. If consideration is restricted to "decision problems," or problems whose solution is a yes/no answer (such as determining whether or not a graph is Hamiltonian), then this classification is formalized as follows: problems for which a polynomial time algorithm is known are grouped in the class $\mathcal{P}$, while algorithms for which a solution can be verified to be correct in polynomial time belong to the class $\mathcal{NP}$.[1] Many problems for which the best known algorithms are superpolynomial have the property that a solution can be verified in polynomial time; the enumerative approach to the Hamiltonian cycle problem described at the beginning of this section is an example – given a particular cycle, one can easily check that it is Hamiltonian.

It became apparent during the 1960's that certain problems seemed to be inherently not amenable to efficient algorithmic solutions. In 1971–3, it was shown in two seminal papers by Cook and Karp, that many of these problems are computationally equivalent – that is,

[1] $\mathcal{NP}$ unfortunately is not an abbreviation for "Not Polynomial"; it stands for Non-deterministic Polynomial time. For further explanation, consult any standard text on algorithms or the theory of computation.

if an efficient algorithm to solve one was discovered, then efficient algorithms for all of them would be immediately evident. This equivalence is shown by exhibiting a polynomial time "reduction" from one problem to a second so that the solution to the former may be obtained by solving an instance of the latter. This web of equivalences begins with a problem that is provably the "hardest" problem in $\mathcal{NP}$, giving the class of $\mathcal{NP}$-complete problems.

Beyond being a useful classification system for computational problems, $\mathcal{P}$ and $\mathcal{NP}$ are the source of arguably the most famous and significant open problem in theoretical computer science and mathematics today. The class $\mathcal{P}$ is contained within $\mathcal{NP}$; it is actually possible that $\mathcal{P} = \mathcal{NP}$, and all problems suspected to be "hard" (in the sens e of being in $\mathcal{NP}$) are in fact solvable by an efficient algorithm. If an efficient algorithm to solve an $\mathcal{NP}$-complete problem were discovered, then in addition to resolving the theoretical question, its practical significance would be multiplied many times over due to the large number problems in $\mathcal{NP}$ that are equivalent. However, it is strongly suspected that in fact $\mathcal{P} \neq \mathcal{NP}$, but no proof of this fact has been established to date.

The Hamiltonian cycle problem a well known $\mathcal{NP}$-complete problem, which means that computational complexity is of particular relevance to this thesis.

1.2.2 Complexity of Restricted Cases

It is probable that no efficient algorithm exists for finding Hamiltonian cycles, but that does not prevent the problem from arising in real applications. There are a number of ways to cope with this dilemma. One might be satisfied with an approximation – for example, a cycle that covers most but not all of the vertices of the graph. Or, the particular instance of the problem might be a special case that is solvable efficiently – for example, complete graphs (graphs in which there are edges between every pair of vertices) always have a Hamiltonian cycle, and it is very easy to find. Finally, if an exact solution is required, the inefficient enumerative algorithm (or a variant) might be tried with the hope that its actual performance on this particular instance of the problem does not approach the worst case. This option is not likely to be chosen due to its unpredictable and probably very bad performance.

The search for restricted cases is then of obvious relevance to the second option, and quite possibly a useful starting point for certain approximations if the first option is pursued. A number of restricted classes of graphs have been shown to be either in $\mathcal{P}$ or $\mathcal{NP}$-complete. Figure 1.1 summarizes some of these results and is drawn from David Johnson's "NP-Completeness Column," a recurring feature in the Journal of Algorithms that covers new developments in the theory of $\mathcal{NP}$-completeness [5].

The distinction between $\mathcal{P}$ and $\mathcal{NP}$ is sometimes surprising – it is not always the case that what seem to be the most restricted classes of graphs are in $\mathcal{P}$ while the least restricted ones remain $\mathcal{NP}$-complete. For example, grid graphs can be loosely characterized as graphs that can be drawn on graph paper, with vertices at the intersections of the grid and edges along the lines on the graph paper. This seems to be a fairly strict restriction, yet the Hamiltonian cycle problem for general grid graphs is $\mathcal{NP}$-complete.

This should indicate that intuition about the computational complexity of these problems is hard to come by. Probing the problems that seem to lie on the "border" between $\mathcal{P}$ and $\mathcal{NP}$ is then important in improving an understanding of the inherent characteristics of problems that are responsible for the $\mathcal{P}$ / $\mathcal{NP}$ delineation, if there is one. Hopefully, a level

Graph Class	**Complexity**	**Reference**
Partial k-Trees	$\mathcal{P}$	S. Arnborg and A. Proskurowski
Series Parallel	$\mathcal{P}$	M. Syslo
k-Outerplanar	$\mathcal{P}$	B.S. Baker
Cographs	$\mathcal{P}$	D. G. Corneil et. al.
Interval	$\mathcal{P}$	M. Keil
Circle	$\mathcal{P}$	A. A. Bertossi
Proper Circ. Arc	$\mathcal{P}$	A. A. Bertossi
Grid Graphs	$\mathcal{NP}$-complete	A. Itai et. al.
$K_{3,3}$-Free	$\mathcal{NP}$-complete	M. R. Garey and D. S. Johnson
Thickness-k	$\mathcal{NP}$-complete	M. R. Garey and D. S. Johnson
Genus-k	$\mathcal{NP}$-complete	M. R. Garey and D. S. Johnson
Perfect	$\mathcal{NP}$-complete	T. Akiyama et. al.
Chordal	$\mathcal{NP}$-complete	C. J. Colbourn and L. K. Stewart
Split	$\mathcal{NP}$-complete	C. J. Colbourn and L. K. Stewart
Comparability	$\mathcal{NP}$-complete	T. Akiyama et. al.
Bipartite	$\mathcal{NP}$-complete	T. Akiyama et. al.
Edge (or Line)	$\mathcal{NP}$-complete	A. A. Bertossi
Claw-Free	$\mathcal{NP}$-complete	A. A. Bertossi
Degree-k	$\mathcal{NP}$-complete	M. R. Garey and D. S. Johnson
Planar	$\mathcal{NP}$-complete	M. R. Garey and D. S. Johnson

Figure 1.1: Some classes of graphs for which $\mathcal{P}$ / $\mathcal{NP}$ results have been obtained for the Hamiltonian cycle problem. For definitions of these graph classes, and complete references to the published results, see [5].

of understanding will be reached at which a proof that $\mathcal{P} \neq \mathcal{NP}$ is evident. Or, if $\mathcal{P} = \mathcal{NP}$, then the perceived "border" may be a fruitful place to look for the problem that can be shown to be $\mathcal{NP}$-complete and also solvable in polynomial time.

This thesis attempts to find a polynomial time algorithm for the Hamiltonian cycle problem in a restricted subclass of grid graphs (recall from Figure 1.1 that the problem is $\mathcal{NP}$-complete for general grid graphs). A large part of the motivation for examining such a problem is that it falls into the category of "border" problems, which have the potential to yield theoretical insight.

1.3 Hamiltonian Cycles in Grid Graphs

Formally, grid graphs are graphs that can be drawn so that their vertices lie on the integer lattice, and all edges connect vertices that are one unit apart. These graphs have three important properties that in many cases permit efficient algorithms for a variety of graph problems. The first is that grid graphs are planar, meaning that they can be laid out in the plane so that no edges cross. The second is that they are bipartite, which means that the vertices of the graph can be partitioned into two sets so that all edges in the graph have one endpoint in each set. Finally, the maximum degree of all vertices is four. Unfortunately, for the Hamiltonian cycle problem, these features are not likely to simplify the problem enough to permit an efficient algorithm.

1.3.1 General Grid Graphs

In 1982, Itai, Papadimitriou, and Szwarcfiter showed that the Hamiltonian cycle problem for general grid graphs remains $\mathcal{NP}$-complete [4]. Itai *et al.* also showed that the Hamiltonian cycle problem for rectangular grid graphs can be solved in polynomial time. In 1986, Everett used this result to give polynomial algorithms for the related problem of Hamiltonian paths in certain non-rectangular grid graphs [3]. Her strategy involves decomposing the graph into a series of rectangular subgraphs, solving the problem for these smaller problems, and then "gluing" the paths into one large path. This strategy does not work in general; the sub-paths may not line up properly, or the decomposition may not be simple enough to permit the gluing of all paths, but certain classes that are amenable to this solution were identified: evenly decomposable grid graphs of size four, grid graphs of size six with a near perfect or perfect matching, pyramid grid graphs of size six without holes, and L-shaped grid graphs without holes. Briggs applied a similar strategy to the Hamiltonian cycle problem for certain grid graphs that are decomposable into two or three rectangular grid graphs [2].

1.3.2 Grid Graphs Without Holes

One restriction on grid graphs that was used by Everett in two of the cases she solved was the restriction to certain grid graphs without holes. A grid graph without holes has the property that no internal face has area greater than one; intuitively, a grid graph without holes has a single outside border with every integer coordinate within that border contained in the vertex set and all edges between adjacent vertices present in the edge set. The Hamiltonian cycle problem for grid graphs without holes remains an open problem; an algorithm has been proposed by Bridgeman, but its putative correctness and polynomial complexity remain conjectural [1]. This algorithm attempts to repeatedly "improve" a 2-factor by reducing its number of components through two cycle merging operations: sliding and gluing. A sequence of slides rearranges a single cycle to include a parallel edge necessary for a glue.

Bridgeman's approach to proving the correctness of the algorithm was to first show that sliding is sufficient to transform any Hamiltonian cycle on some subgraph into any other Hamiltonian cycle on the same vertices. She established that if the algorithm terminated with multiple cycles, then these cycles met along boundaries classified as zipper, stairstep, or combination. To prove the correctness of the algorithm, it is necessary to show that no Hamiltonian cycle in the whole graph can exist if the cycles meet in these configurations. Bridgeman completed this proof for the stairstep case, and made progress on the zipper case as well as the sufficiency of sliding proof upon which that case depends. So despite indications that the algorithm proposed by Bridgeman may place the Hamiltonian cycle problem for grid graphs without holes in $\mathcal{P}$, that classification still awaits proof.

1.4 Outline

In this thesis, the underlying structure of grid graphs without holes is examined in detail, with the ultimate goal of deriving a polynomial time algorithm for finding Hamiltonian cycles in these graphs. Chapter 2 introduces some formal terminology and properties of grid graphs and 2-factors, and discusses in detail the problem of finding a 2-factor in a grid graph without holes. Chapters 3 through 7 constitute the majority of the original work presented. In them, a number of properties of 2-factors in grid graphs without holes are described, and structures are characterized that may be exploited by an efficient algorithm to find Hamiltonian cycles. This culminates in the identification of a general improving transformation that is provably present in any Hamiltonian grid graph without holes. Chapter 8 uses these results to give a polynomial time algorithm. Chapter 9 briefly summarizes the major results of this thesis, and discusses directions for future work. Appendix A discusses some of the implications of the results in this thesis on prior work by Bridgeman in [1].

Chapter 2

Preliminaries

In this chapter, formal definitions of the objects treated in this thesis are introduced, along with some elementary properties of these objects. A familiarity with some basic graph theory is assumed.

2.1 Grid Graphs and Hamiltonian Cycles

Definition 2.1 *A* **grid graph** *is a graph whose vertices lie on integer coordinates and whose edges connect all pairs of vertices that are unit distance apart.*

Definition 2.2 *A* **grid graph without holes** *is grid graph for which all interior faces have unit area.*

Figure 2.1 illustrates a general grid graph (with holes) and a grid graph without holes. Grid graphs have several basic graph properties that will be used heavily for proofs in later chapters.

Definition 2.3 *A graph is* **bipartite** *if its vertices can be partitioned into two sets so that all edges have one endpoint in each set.*

Grid graphs are bipartite; this property can be seen by coloring vertices according to their integer coordinates. If vertices whose integer coordinates sum to an even number are colored red and vertices whose coordinates sum to an odd number are colored green, it is easy to see that every edge must connect a red and green vertex. Since each edge connects vertices that are unit distance apart, either the x or y coordinate but not both must change by one between endpoints. The sum of the coordinates at one endpoint must then be odd and the sum at the other endpoint must be even, so the endpoint must be different colors. The set of red vertices and the set of green vertices then form the partition required by the definition of a bipartite graph.

Definition 2.4 *A graph is* **planar** *if it can be drawn in the plane with no edge crossings.*

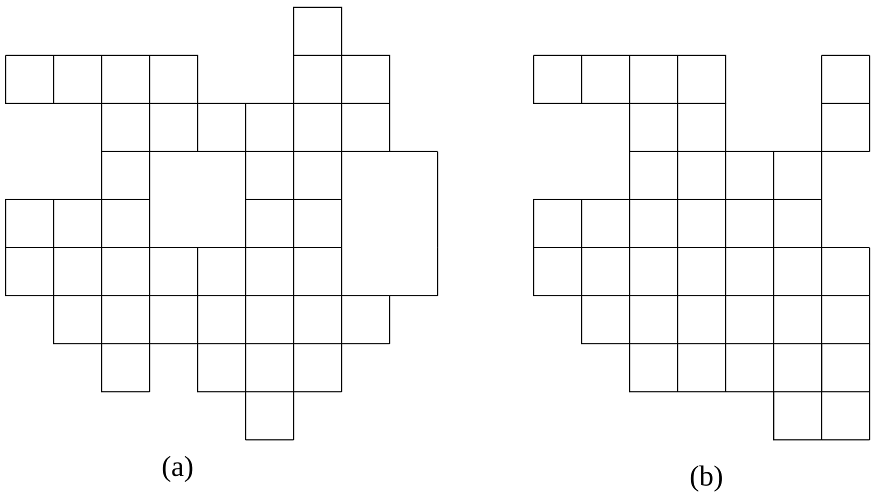


Figure 2.1: A grid graph with holes (a) and a grid graph without holes (b).

As mentioned in the introduction, grid graphs are planar, since their definition requires a layout in the plane in which no edges cross. Finally, grid graphs have maximum degree four; this should be apparent from the definition and the figure.

The following property of bipartite graphs will be important:

Lemma 2.1 *Every cycle in a bipartite graph has even length.*

Proof Since every edge is between vertices of different colors, the colors of the vertices must alternate along the cycle. There must then be an equal number of vertices of each color along the cycle, so the number of vertices in the cycles must be even. □

Definition 2.5 *A* **Hamiltonian cycle** *in a graph is a cycle that visits every vertex exactly once.*

Definition 2.6 *A* **Hamiltonian graph** *is a graph that contains at least one Hamiltonian cycle.*

Notice that while all vertices are visited, not all of the edges need be used; in fact this is rarely the case. It is precisely the fact that there are often many edges *not* used in a Hamiltonian cycle that makes the problem of designing an algorithm to identify one difficult – there are too many "wrong" choices to make.

2.2 2-Factors

The 2-factor is the starting point for both the algorithm presented in Chapter 8 and the algorithm proposed by Bridgeman in [1] for finding Hamiltonian cycles in grid graphs without holes. In this section, 2-factors are defined and two methods for finding 2-factors are presented, one in detail.

Definition 2.7 *A* **2-factor** *of graph G is a spanning subgraph of G for which all vertices have degree two.*

A 2-factor can also be thought of as a set of vertex-disjoint cycles that completely covers the vertex set of G, where "covers" means that the cycles collectively visit every vertex. A Hamiltonian cycle is just a special 2-factor that consists of only one cycle, so a reasonable approach to finding one is to start by finding an initial 2-factor and then to attempt to transform it into 2-factors with successively fewer cycles, until a Hamiltonian cycle is found if one exists.

Polynomial algorithms exist to find 2-factors in arbitrary graphs, although the specific algorithm presented here is suited only for bipartite graphs.

2.2.1 Linear Programming Method

This method is used by Bridgeman, and it works by reducing the 2-factor problem to a much more general problem called linear programming.

In a linear programming problem, the goal is to maximize an objective function of a set of variables $x_1, x_2, \cdots, x_n$, subject to a set of linear inequality constraints on the variables. Identifying a 2-factor may be thought of as choosing a subset of edges in a graph G so that every vertex is incident to exactly two of the edges. In linear programming terms, the goal is to maximize the number of edges subject to the degree constraint.

To express the 2-factor problem as a linear program, each edge in G is assigned a variable from $x_1, x_2, \cdots, x_n$, where $n = |E(G)|$, with the constraints $0 \leq x_i \leq 1$ for all $1 \leq i \leq n$. A value of 0 for any variable is interpreted as the absence of its corresponding edge in the subset of edges, while a value of 1 indicates the presence of the corresponding edge, so the objective function is simply the sum of the x_i's. To introduce the degree requirement it is necessary to add constraints for each vertex v. Letting $E(v)$ be the set of x_i's that correspond to the edges incident to v, each constraint is expressed as:

$$\sum_{x_i \in E(v)} x_i \leq 2$$

The variables in a linear programming problem are real-valued, so for this method to work for finding 2-factors, it must be shown that the objective function's maximum is at integer values for all of the variables. Fortunately, such a property holds for this particular linear programming problem; for a proof and more detailed discussion of this method of finding 2-factors, see [1].

A polynomial time algorithm for solving linear programming problems exists but is slow in practice; other algorithms, such as the simplex method, are faster but are not guaranteed to complete in polynomial time. The linear programming model is general enough that additional information concerning the structure of the graph might be incorporated into this method of finding 2-factors, to potentially improve this initial phase of the algorithm for finding Hamiltonian cycles. However, absent such an improvement, the method described in the next section is far simpler and probably faster in practice than the linear programming method.

2.2.2 Reduction to Matching

This section describes a reduction of the 2-factor problem to the problem of finding a maximum matching, suggested by an exercise[1] in [8]. A relatively simple algorithm is described that solves the maximum matching problem for bipartite graphs.

Definition 2.8 *A* **matching** *of a graph G is the subgraph induced by a subset of the edges of G that contains no adjacent edges.*

Definition 2.9 *A* **perfect matching** *of a graph G is a matching that covers all of the vertices of G.*

The following definition gives the transformation that reduces the problem of finding a 2-factor in one graph into the problem of finding a perfect matching in a related graph:

Definition 2.10 *Given a graph G, G^* is the graph obtained by the following two transformations:*

1. *Replace every edge (x, y) with the vertices x' and y' and the edges (x, x'), (y, y') and (x', y').*
2. *For every vertex x belonging to the original graph G, add the vertex x'' with edges (x'', y) for all y such that (x, y) is an edge after step 1.*

Figure 2.2 gives an example of this transformation.

Lemma 2.2 *If G is bipartite then G^* is also bipartite.*

Proof In step one of the definition, vertices x and y are different colors in the bipartite coloring of G. Coloring x' the same color as y, and y' the same color as x preserves the bipartite coloring. Likewise, in step two, coloring x'' the same color as x preserves the bipartite coloring since the only edges added are edges between x'' and vertices that were connected to x in the original bipartite graph. So G^* is bipartite if G is. □

Intuitively, a 2-factor F of G can be transformed into a perfect matching in G^* in two steps corresponding to the two steps that transform G into G^*. In the first step, each edge (x, y) in F is "split" into two edges, (x, x') and (y', y). In the next step, degree two vertices in F are "pulled apart" so that the two edges incident to a vertex x are split into one edge incident to x and one edge incident to x''.

The inverse process of "joining" vertices x and x'' and then "merging" the edges that end at x' and y' into a single edge transforms any perfect matching in G^* into a 2-factor in G. These procedures are formalized in the following lemma:

Lemma 2.3 *A perfect matching exists in G^* if and only if a 2-factor exists in G.*

Proof In this proof, the labeling of vertices is the same as in the definition of G^*, so the "doubled" vertices from step two are the pair marked with no prime and with a double

[1] The exercise actually suggests a generalization of the reduction described here that reduces the f-factor problem to matching.

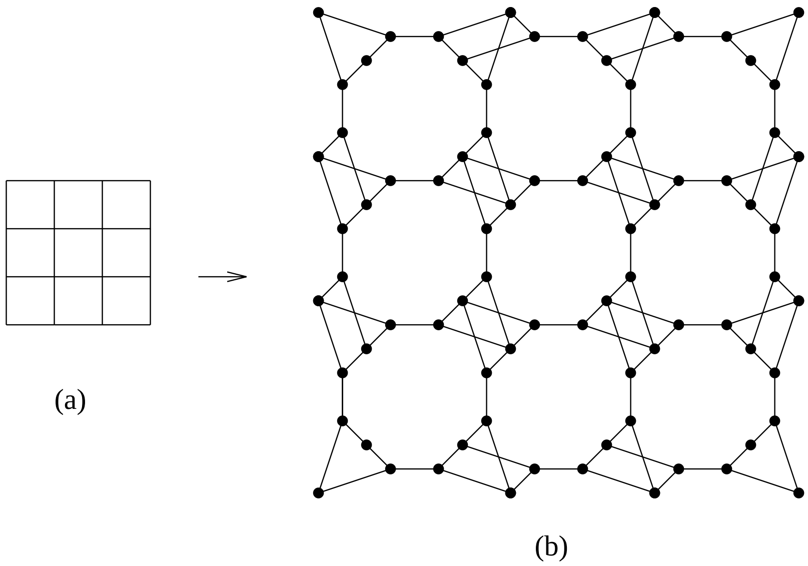


Figure 2.2: A simple grid graph (a), and the transformed graph for the matching problem (b).

prime (e.g. x and x''), and the edges inserted in step one are between two vertices each marked with a single prime, (e.g. x' and y').

Let F be a 2-factor in G. Then a perfect matching in G^* exists by the following construction: Pick a cycle C in F; it has even length since G is bipartite. For every other edge (x, y) along C, add the edges (x, x') and (y', y) to the matching in G^*. For the remaining edges (x, y) along C, add the edges (x'', x') and (y', y'') to the matching in G^*. Repeat this procedure for each cycle in F. Finally, for every edge (x, y) of G not in F, add the edge (x', y') to the matching in G^*.

Since every vertex in F has degree two (and so must be on a cycle), all of the vertices x and x'' are the endpoints of some edge in the matching by the first step in the construction. Each pair of vertices x' and y' are in the matching either by the first step, when a pair of edges is added for each corresponding edge in the 2-factor, or by the second step. Since every vertex is covered by the matching, it is a perfect matching. Figure 2.3 shows a 2-factor and the corresponding perfect matching.

Now, let M be a perfect matching in G^*. Then there exists a 2-factor F in G by the following construction: For every edge (x', y') *not* in the M, add the edge (x, y) to F. A consideration of Figure 2.3 should reveal that this is effectively the inverse of the process used to obtain a perfect matching from a 2-factor.

Consider a pair of "doubled" vertices in G^*, x and x'' and the (up to) four surrounding vertices and edges as pictured in Figure 2.4. Since there is no edge between x and x'', exactly two of the vertices a', b', c', and d' must be the endpoints of two edges incident to x and x''. Without loss of generality, assume that these edges are (x, a') and (x'', b'). Then note that edge (a', e') and (b', f') cannot be in the matching, so the two corresponding edges

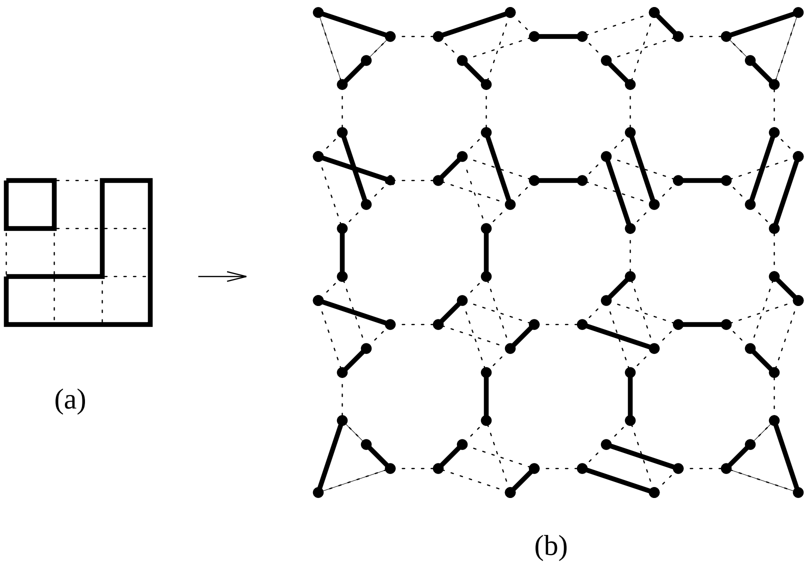


Figure 2.3: A 2-factor (a), and the corresponding perfect matching (b).

incident to vertex x in G must be in F. Further, in order to cover vertices c' and d' (if they exist), edges (c', g') and (d', h') must be in M, so the corresponding edges which are also incident to x in G must *not* be in F. So every vertex in G has degree two in F, and so F is a 2-factor. $\square$

Since G^* is constructed to contain a perfect matching if G contains a 2-factor, an algorithm that finds a *maximum* matching will suffice. If the maximum matching is not perfect, then a 2-factor does not exist.

An algorithm due to Edmonds[2] finds maximum matchings in arbitrary graphs, so in principle the reduction to matching method can be used to find 2-factors in general graphs. However the algorithm is quite complicated and will not be presented here. In contrast, there are many relatively simple and fast algorithms to find maximum matchings in bipartite graphs. For the purpose of finding Hamiltonian cycles in grid graphs without holes these will be sufficient since every grid graph G is bipartite, and by Lemma 2.2 G^* is also bipartite.

Two of the most common algorithms are the maximum flow and alternating path algorithms. The alternating path method will be described here as it introduces a major theme (alternating structures) that recurs throughout this thesis.

Alternating Path Method

The description of this algorithm and the relevant terminology are based heavily on the presentation in [9]

[2]See [9] for a description of Edmonds' algorithm.

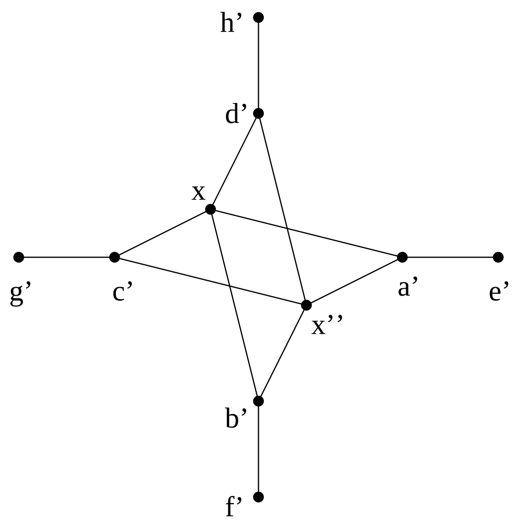


Figure 2.4: The edges and vertices surrounding a doubled vertex in G^*.

Definition 2.11 *A* **free vertex** *with respect to a matching M of G is a vertex in G with no incident edge in M.*

Definition 2.12 *An* **alternating path** *with respect to a matching M is a path in G whose edges alternate between being in and out of M.*

Definition 2.13 *An* **augmenting path** *with respect to a matching M is an alternating path that begins and ends at free vertices.*

Notice that "applying" an augmenting path by adding all of the edges of the path that are not in M to the matching, and removing all of the edges of the path that were originally in M from the matching, yields a matching with one more edge than the original matching.

Definition 2.14 *An* **alternating search tree** *with respect to a matching M is a tree rooted at a free vertex for which every path from the root to a leaf is an alternating path with respect to M.*

Definition 2.15 *An alternating search tree is* **blocked** *if it is maximal and does not reach a free vertex at a leaf.*

The following theorems form the basis of the algorithm. They are stated here without proof; consult [9] for a more thorough treatment.

Theorem 2.1 *A matching M in G is maximal if and only if there exists no augmenting path with respect to M.*

Theorem 2.2 *An alternating search tree rooted at r and built by repeatedly adding pairs of alternating edges will find an augmenting path starting at r if one exists.*

```
algorithm Maximum_Matching

var G : Graph
    Root : Vertex
    T : Tree

while Exists_Free_Vertex(G) do
        Root = Next_Free_Vertex(G)
        T = Find_Alternating_Tree(G, Root)
        if T not blocked then
           Apply_Augmenting_Path(T, Root)
        else Remove_Tree(G, T)
end.
```

Figure 2.5: The alternating path algorithm for finding a maximum matching in a bipartite graph.

Theorem 2.3 *A blocked alternating search tree may be ignored in subsequent searches.*

The algorithm is then quite simple, and is given in Figure 2.5. Exists_Free_Vertex returns a boolean; Next_Free_Vertex returns a free vertex to serve as the root of an alternating tree. Find_Alternating_Tree builds a maximal alternating tree rooted at the free vertex, and if the resulting tree is not blocked, an augmenting path starting at the root is applied, increasing the size of the matching. If the tree is blocked, then its vertices and edges are removed from G and the next free vertex is tried.

The performance of this algorithm on a graph $G(V, E)$ is shown in [9] to be $O(|V||E|)$. It should be clear from the definition of G^* and the construction in the proof of Lemma 2.3 that the transformation of G to G^* and the extraction of the 2-factor in G from a perfect matching in G^* are linear time operations. So the alternating path algorithm gives an efficient method for quickly constructing a 2-factor in a grid graph without holes, to serve as a starting point for working toward a Hamiltonian cycle.

Of course, if the alternating path algorithm returns a maximum matching that is not a perfect matching, then no perfect matching exists in G^*, and by Lemma 2.3, no 2-factor exists in G. In this case, G is not Hamiltonian since a Hamiltonian cycle *is* a 2-factor in G, so no further work is needed.

The case in which G does not contain a 2-factor is one of two cases in which it is easily seen that G is not Hamiltonian. The second case arises if G has multiple components. For the remainder of this thesis, it will be assumed that G is a single component, and until the presentation of the algorithm in Chapter 8, it will also be assumed that G contains a 2-factor.

Chapter 3

Alternating Cycles

In this chapter, the transformations that turn one 2-factor into another are investigated. In a simplistic view, this consists simply of a series of edge parity flips, where each edge parity flip changes the edge from being in the original 2-factor to being out, or from being out to being in. As is suggested by the correspondence between the 2-factor problem and the matching problem that is solved by identifying alternating paths, these edge parity flips collectively form alternating cycles. The proof of this fact and some important properties of the cycles are presented in this chapter.

3.1 Terminology and Conventions

This section briefly introduces some terminology common to all of the chapters.

Definition 3.1 *Let F be a 2-factor of a grid graph without holes G. Edges of G are either "in" F or "out" of F; this terminology defines the* **parity** *of the edge.*

In figures, "in" parity is represented by dark lines, "out" parity by light dotted lines. Unless otherwise specified, G refers to a grid graph without holes, F to a 2-factor of G, and G_F is defined as follows:

Definition 3.2 *The graph with respect to F, denoted G_F, is G with edge parities assigned with respect to F.*

Definition 3.3 *The* **symmetric difference** *of two subgraphs, G_1 and G_2, of a graph G is denoted $G_1 \oplus G_2$, and consists of all edges in exactly one of the subgraphs.*

The following properties of the symmetric difference are important and follow from the definition[1]:

1. $G_1 \oplus G_2 = G_2 \oplus G_1$ (commutative law)
2. $(G_1 \oplus G_2) \oplus G_3 = G_1 \oplus (G_2 \oplus G_3)$ (associative law)

[1] In fact, the set of all subgraphs of G form an abelian group under $\oplus$.

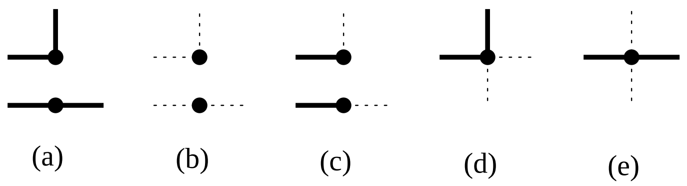


Figure 3.1: The possible configurations of the edges surrounding a vertex in $F_1 \oplus F_2$. Dark edges belong to F_1; light edges belong to F_2. For the purposes of the proof of Lemma 3.1, each pair of configurations in (a-c) is identical.

3. $G_1 \oplus G_1 = \emptyset$ (every subgraph is its own inverse)

4. $G_1 \oplus \emptyset = G_1$ ($\emptyset$ is the identity with respect to $\oplus$)

Often, G_2 will be thought of as a "transformation" that changes G_1 into a new subgraph $G_1' = G_1 \oplus G_2$. More complex transformations are built from a sequence of simpler transformations, so the following shorthand is adopted:

Definition 3.4 *If $A = (a_1, a_2, \cdots, a_n)$ is a sequence of subgraphs of G, and G_1 is a subgraph of G, then $G_1 \oplus A = G_1 \oplus a_1 \oplus a_2 \oplus \cdots \oplus a_n$.*

Definition 3.5 *If $A = (a_1, a_2, \cdots, a_n)$ is a sequence and $1 \leq j,\ k \leq n$ then $A_{j,k} = (a_j, \cdots, a_k)$.*

3.2 Edge-Disjoint and Alternating Properties

Definition 3.6 *An* **alternating cycle** *is a cycle in G_F whose edges alternate parity.*

Lemma 3.1 *Let F_1 and F_2 be 2-factors of G, and let $S = F_1 \oplus F_2$. Then S contains an alternating cycle.*

Proof Each 2-factor is a set of edge-disjoint cycles, and it is a standard proof from graph theory that the symmetric difference of two sets of edge-disjoint cycles is a set of edge-disjoint cycles.

To show that S contains an alternating cycle, the possible configurations of the edges incident to a vertex in S are examined. Since S is a set of edge-disjoint cycles, all vertices have even degree, and since S is a subgraph of G, it has maximum degree four. The possible configurations around each vertex are pictured in Figure 3.1.

The configurations shown in (a) and (b) are not possible, for if they were, then either the shared vertex is "missed" in F_2 (a) or F_1 (b), or the other two edges incident to the shared vertex are not in the symmetric difference because they are edges in both F_1 and F_2, which in either case implies a vertex of degree greater than two in F_1 or F_2.

Each vertex must then have the configuration shown in (c), (d), or (e), so it is always possible to "walk" along alternating edges of S until returning to a previously visited vertex, at which point an alternating cycle is identified. As a subgraph of G, S is bipartite, so all cycles have even length, and so the parities of the starting and ending edges in this cycle alternate. □

Lemma 3.2 *Let C be an alternating cycle in G_F. Then $F \oplus C$ is a 2-factor of G.*

Proof Cycle C alternates with respect to F, so in the symmetric difference, every vertex on the cycle has one incident edge removed, and another added, keeping its degree equal to two. No other vertices are affected, so the result is another 2-factor. □

Lemma 3.3 *Let F_1 and F_2 be 2-factors of G, and let $S = F_1 \oplus F_2$. Then S can be partitioned into edge-disjoint alternating cycles.*

Proof The proof is by induction on $|S|$. By Lemma 3.1, S contains an alternating cycle C and by Lemma 3.2, $F_1 \oplus C = F_1'$ is a 2-factor of G. Also, $F_1' \oplus F_2 = (F_1 \oplus C) \oplus F_2 = F_1 \oplus F_2 \oplus C = S \oplus C = S'$, and since $C \subseteq S$, $|S'| < |S|$.

So by induction, S' can be partitioned into edge-disjoint alternating cycles. Adding C gives a partition of S into edge-disjoint alternating cycles, since C is alternating and S' contains no edges in C. As a base case, if S contains only a single cycle, it must be alternating by Lemma 3.1. □

3.3 Non-Intersection Property

A stronger version of Lemma 3.3 is possible – in fact the edges in $F_1 \oplus F_2$ can be partitioned into edge-disjoint *non-intersecting* alternating cycles. The proof of this fact requires some additional definitions and lemmas:

Definition 3.7 *A* **cell** *in G_F is a cycle with exactly four edges. If a cell is an alternating cycle, then it is an* **alternating cell.**

Definition 3.8 *The* **area** *of a cycle C in a grid graph is the number of interior cells of C.*

Definition 3.9 *Two edge-disjoint cycles are* **nested** *if all of the interior cells of one are interior cells of the other.*

Definition 3.10 *Two edge-disjoint cycles* **intersect** *if they share at least one interior cell and are not nested.*

Lemma 3.4 *Let F_1 and F_2 be 2-factors of G, and let $S = F_1 \oplus F_2$. Then an alternating cycle in S of minimal area intersects no other cycle in S.*

Proof First, recall that by Lemma 3.1, S contains an alternating cycle. Let C be an alternating cycle of minimal area in S. If C contains no degree four vertices with incident edges of $S - C$ that point "in" to the interior of C, then C satisfies the lemma, since no other cycle crosses its boundary.

Suppose C contains a degree four vertex with at least one incident edge of $S - C$ on the interior of C; it must be in one of the three configurations shown in Figure 3.2. As

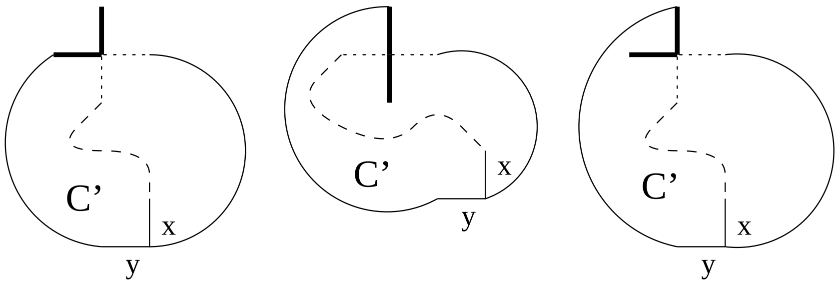


Figure 3.2: The configurations of a degree four vertex with at least one edge inside the cycle, for the proof of Lemma 3.4.

in the proof of Lemma 3.1, it is possible to "walk" from one of the free edges along an alternating path inside C until intersecting C at a degree four vertex. In the figure, this path is represented by a dashed line.

A new cycle C' is thus formed with smaller area than C, and it must be an alternating cycle. If it were not, then the edges labeled x and y in the figure would have the same parity, since the dashed path and the original cycle alternate. But then C' would have odd length, which is not possible in a bipartite graph. So C' is an alternating cycle with smaller area than C, which contradicts the choice of C, so C must not contain any degree four vertex with an incident edge of $S - C$ on the interior of C. Therefore no cycles intersect it, and it satisfies the lemma. □

Lemma 3.5 *Let F_1 and F_2 be 2-factors of G, and let $S = F_1 \oplus F_2$. Then S can be partitioned into edge-disjoint non-intersecting alternating cycles.*

Proof The proof is by induction on $|S|$. By Lemma 3.4, it is possible to select an alternating cycle, C, from S so that C intersects no other cycles in S. If this is the only cycle in S, the lemma follows trivially.

If not, then by Lemma 3.2, $F_1 \oplus C = F_1'$, a 2-factor of G. As in the proof of Lemma 3.3, $F_1' \oplus F_2 = (F_1 \oplus C) \oplus F_2 = F_1 \oplus F_2 \oplus C = S \oplus C = S'$, and since $C \subseteq S$, $|S'| < |S|$. By induction S' can be partitioned into edge-disjoint non-intersecting alternating cycles. Cycle C encloses no portion of any other cycle, so adding C to this partition does not create any intersecting cycles.

Thus, adding C gives a partition of S into edge-disjoint non-intersecting alternating cycles, since C is alternating and S' contains no edges in C. □

The previous lemma implies that any 2-factor of G can be transformed into any other 2-factor by flipping the edge parities of a set of edge-disjoint non-intersecting alternating cycles. Of particular relevance are those 2-factors that are also Hamiltonian cycles; hence the following corollary:

Corollary 3.1 *Let F be a 2-factor of a grid graph without holes G, and let H be a Hamiltonian cycle in G. Then $F \oplus A = H$, where A is a sequence of edge-disjoint non-intersecting alternating cycles in G_F.*

Proof Simply let $F_1 = F$ and $F_2 = H$ in the previous lemma, and let A be a partition of $S = F \oplus H$ into edge-disjoint non-intersecting alternating cycles in G_F. By the properties of the symmetric difference, $F \oplus A = F \oplus S = F \oplus (F \oplus H) = \emptyset \oplus H = H$. $\square$

Chapter 4

Alternating Cells

The previous chapter establishes that any transformation between 2-factors of a grid graph without holes can be viewed as flipping the parities of the edges of a set of edge-disjoint non-intersecting alternating cycles. In this chapter, it is shown that such a transformation can be accomplished by simply flipping the edge parities of a sequence of individual cells, each of which is an alternating cell at the time it is flipped. The existence of these "alternating cell sequences" is shown with the help of a dependency graph defined below.

4.1 The Dependency Graph

Definition 4.1 *The* **dual of** G *is the graph whose vertex set consists of a vertex for each cell of* G *and whose edge set consists of an edge between two vertices if the corresponding two cells in* G *share an edge.*

Definition 4.2 *The* **dependency graph for** G_F*, denoted* G_F^**, is constructed from the dual of* G *by adding to it an edge that crosses each of the edges of* G *that border the outer face, and the necessary endpoint vertices, and assigning edge parities according to the following rules (for a fixed orientation of* G*):*

1. *edges of* G_F^* *that cross a vertical edge in* F *are dark,*
2. *edges of* G_F^* *that cross a vertical edge not in* F *are light,*
3. *edges of* G_F^* *that cross a horizontal edge in* F *are light, and*
4. *edges of* G_F^* *that cross a horizontal edge not in* F *are dark.*

This definition corresponds to superimposing over each cell of G_F an alternating cell, and assigning one parity to edges of G_F^* that cross an edge of G_F whose parity matches the superimposed edge, and the other parity to edges of G_F^* that cross an edge of G_F whose parity does not match the superimposed edge. Thus a cell in G_F is alternating if and only if the parities of all four of the edges in G_F^* that cross its edges are the same.

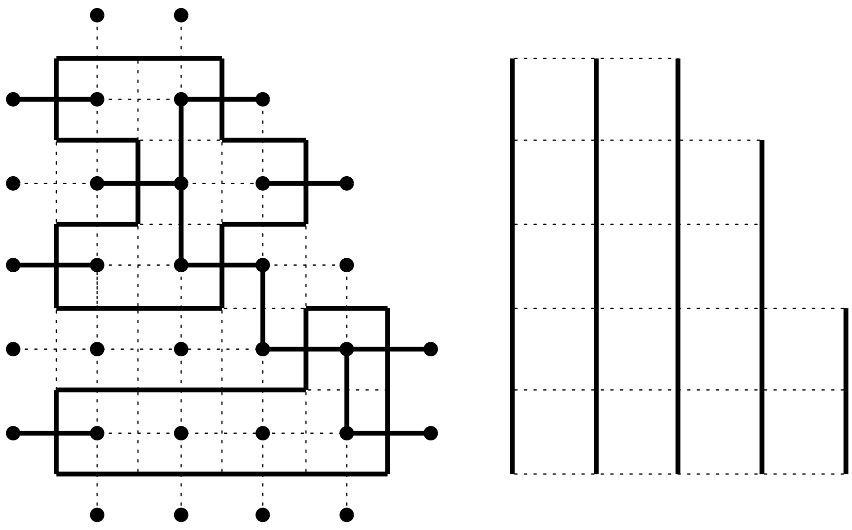


Figure 4.1: An example of a dependency graph (a), and the superimposed grid of alternating cells (b). The vertices of the dependency graph are marked with heavy dots, to distinguish dependency graph edges from those in the 2-factor.

The purpose of the dependency graph is to describe the sequence of cells that must be flipped for a particular cell to become alternating. To encode this information, directions are associated with the edges of G_F^* so that all edges are directed from cells that must be flipped earlier in an alternating cell sequence to cells that must be flipped later in the sequence. In general there are two ways to assign directions, since there are two (inverse) configurations of an alternating cell.

4.2 The Oriented Dependency Graph

Definition 4.3 *The dependency graph G_F^* is* **oriented** *if all of its edges are directed so that edges (x, y) and (y, z) have the same direction with respect to y if they have the same parity, and opposite directions with respect to y if they have opposite parities, for all vertices x, y, and z.*

It is not obvious that an arbitrary dependency graph can be oriented, or even that any dependency graph can be oriented. However the following lemmas show that all dependency graphs can be oriented with consistent edge direction assignments:

Lemma 4.1 *If directions are assigned to successive edges along a path of even length in G_F^* satisfying the edge direction rules of an oriented dependency graph, then either:*

1. *the directions of the edges at the end of the path are the same (with respect to a fixed orientation of the path) and the edges at the ends have different parities, or*

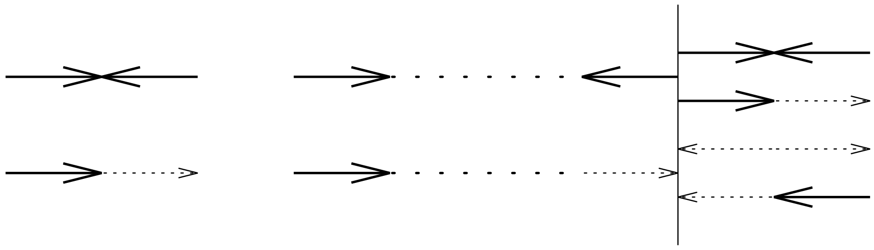


Figure 4.2: The base case (a) and induction step (b) for Lemma 4.1.

2. *the directions of the edges at the end of the path are different (with respect to a fixed orientation of the path) and the edges at the ends have the same parity.*

Proof The proof is by induction on the length of the path. Figure 4.2(a) shows the two possible configurations for paths of length two, which both satisfy the lemma. For a path of length $2n$, the path excluding the last two edges, of length $2(n-1)$, satisfies the lemma by induction, and so must be in one of the configurations shown in (b), to the left of the vertical line. The possible parities and directions for the next two edges are shown to the right of the vertical line, and each results in a path of length $2n$ that satisfies the lemma. □

Lemma 4.2 *Every dependency graph G_F^* can be oriented.*

Proof If a consistent assignment of edge directions is not possible, then there must be a cycle in G_F^* with the property that assigning directions to successive edges along the cycle results in the starting edge and ending edge violating the rules of edge direction in an oriented dependency graph. That is, if the starting edge is (x, y) and the ending edge is (z, x), then either the two edges have the same parity and are directed in opposite directions with respect to x, or they have opposite parities and are directed in the same direction with respect to x. However, all cycles in a bipartite graph have even length, so by Lemma 4.1, neither of these two combinations are possible. □

It should be noted that reversing all of the directions of an oriented dependency graph gives the other valid orientation of the same dependency graph. The actual edge directions, however, are immaterial to the remainder of this chapter; the information that they encode is extracted from their *relative* directions. Therefore, "the oriented dependency graph" shall refer to whichever of the two orientations is most convenient for the statement of the lemma or proof.

Definition 4.4 *A* **dependency arc** *is a directed edge in the oriented dependency graph.*

Lemma 4.3 *The oriented dependency graph G_F^* contains no directed cycle.*

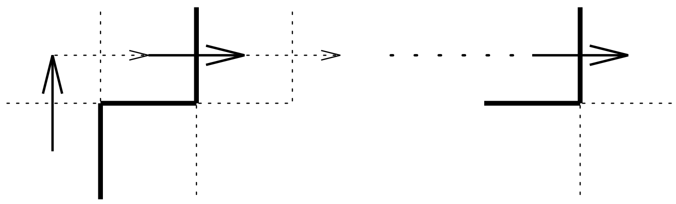


Figure 4.3: The topmost row of a directed cycle D in G_F^*.

Proof Every directed cycle must have edges that alternate parity, since any repeated parity implies oppositely oriented edges along the cycle. Suppose D is a directed cycle in G_F^*, and consider the topmost row of D and the corresponding edges of G_F. Without loss of generality, these edges have the parities pictured in Figure 4.3. The edge entering the topmost row from below (on the left) and the next edge along the cycle imply that the parities of the two edges of G_F leftmost in the figure are light. The other two edges incident to the shared vertex of these two light edges must be dark due to the degree constraint of the 2-factor. The next edge along D must be dark, implying that the edge of G_F that it crosses must also be dark. This pattern repeats all along the topmost row, with the parity of each successive edge of D the same parity as the edge of G_F that it crosses. At the end of the topmost row of D the cycle turns "down", so an edge of D must cross the rightmost pictured (light) edge of G_F. By the construction of G_F^*, this dependency graph edge must be dark and therefore directed "up" in the figure, but then D doesn't alternate, and thus cannot be a directed cycle, a contradiction. So there can be no directed cycle in the oriented dependency graph. $\square$

4.2.1 The Oriented Dependency Graph and Alternating Cycles

With the next lemma, the connections between the oriented dependency graph and the alternating cycles of the previous section become apparent.

Lemma 4.4 *Let C be a cycle in G_F, and let G_F^* be the oriented dependency graph for G_F. Then C is alternating if and only if every edge of C is crossed by an edge of G_F^* oriented in the same direction with respect to the interior of the cycle.*

Proof Suppose C is alternating. Starting with some edge e on C, it is shown that the dependency arc crossing e and the dependency arc crossing e's neighboring edge must have the same direction with respect to the interior of C. Figure 4.4 illustrates the possible configurations of the relevant edges of G_F along with the directed edges of G_F^* that cross them. The consistent direction of the dependency arcs with respect to the interior of C must be true all the way around C, and the first half of the lemma follows.

Now suppose that C is any cycle such that the dependency arcs that cross C are all oriented in the same direction with respect to the interior of C. The possible orientations for the arcs crossing adjacent edges of C are exactly those shown in the figure (up to reversing all of the directions). In each case, successive edges of C are forced to alternate

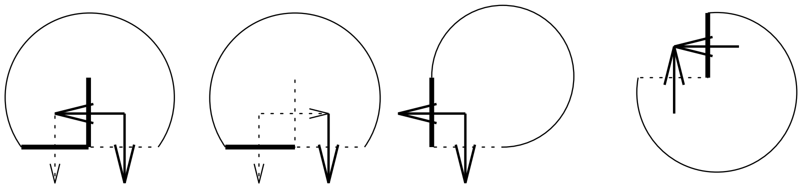


Figure 4.4: The possible configurations of the oriented dependency graph near two adjacent edges on an alternating cycle.

by the parities of the dependency arcs that cross them. So C must alternate, proving the second half of the lemma. □

4.2.2 The Oriented Dependency Graph and Regions

Definition 4.5 *A* **region** *of a grid graph without holes G is a subgraph of G with a connected dual graph.*

The **border** of a region is the set of edges in the region that belong to exactly one interior cell.

Definition 4.6 *If every arc of the oriented dependency graph G_F^* that crosses a border edge of the region R in G_F is oriented in the same direction with respect to the interior of R, then R is an* **alternating cell region**.

The justification for this name is in the next section.

Lemma 4.5 *Let R be an alternating cell region in G_F. An alternating path in G_F^* can cross no more than one border edge of R.*

Proof Recall that in an oriented dependency graph all of the edges along an alternating path have the same direction with respect to the path. Now suppose some alternating path p crosses more than one border edge of R. The section of p between the closest two crossings is either entirely inside, or entirely outside R. In either case, because the edges of p all have the same orientation, these two crossings must have opposite orientations with respect to the interior of R, contradicting the definition of an alternating cell region. So p cannot cross more than one border edge of R. □

Notice that by Lemma 4.4, all of the dependency arcs that cross an alternating cycle in G_F are oriented in the same direction with respect to the interior of the cycle. Thus the region enclosed by an alternating cycle is just a special case of an alternating cell region.

4.3 Alternating Cell Sequences

In this section, the oriented dependency graph is used to prove the existence of alternating cell sequences.

Definition 4.7 *An* **alternating cell sequence** *in G_F is a sequence of cells, defined recursively as follows:*

1. *an alternating cell in G_F is an alternating cell sequence, and*
2. *if c is an alternating cell in G_F and $A = (a_1, a_2, \cdots, a_n)$ is an alternating cell sequence in $G_{(F \oplus c)}$, then $(c, a_1, a_2, \cdots, a_n)$ is an alternating cell sequence in G_F.*

Lemma 4.6 *If $A = (a_1, a_2, a_3 \cdots, a_n)$ is an alternating cell sequence in G_F and $B = (b_1, b_2, \cdots, b_n)$ is an alternating cell sequence in $G_{(F \oplus A)}$, then*

$$(a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_n)$$

is an alternating cell sequence in G_F.

Proof The proof is by induction on the length of A. If A has length one, then $G_{(F \oplus A)} = G_{(F \oplus a_1)}$ and so by the definition of alternating cell sequences, $(a_1, b_1, b_2, \cdots b_n)$ is an alternating cell sequence.

Otherwise, by the definition, $B' = (a_n, b_1, b_2, \cdots, b_n)$ is an alternating cell sequence in $G_{(F \oplus A_{1,n-1})}$, and so by induction, $A_{1,n-1}$ and B' can be concatenated to give the required alternating cell sequence in G_F. □

Definition 4.8 *In a directed graph, a* **source** *is a vertex with in-degree zero; a* **sink** *is a vertex with out-degree zero.*

Lemma 4.7 *Let R be an alternating cell region in G_F. Then there exists an alternating cell sequence consisting of exactly the interior cells of R with no repeated cells.*

Proof For convenience, assume that the orientation of G_F^* is chosen so that all dependency arcs that cross the border of R are directed away from the interior of R. (The opposite orientation simply requires the substitution of the word "sink" for "source" in the proof.)

The proof is by induction on the number of interior cells of R. If R has only one cell, its dual vertex must be a source in G_F^*, and so it must be an alternating cell. The alternating cell sequence thus consists of only that cell. Otherwise, R must contain a dual vertex that is a source in G_F^*. This can be located by starting at a dependency arc that crosses the border of R and following it backwards as far as possible. Since the oriented dependency graph contains no directed cycles (Lemma 4.3), this path cannot repeat vertices. Also, since all dependency arcs crossing border edges of R are directed away from the interior, the path must remain within the interior of R, and so must terminate, necessarily at a source.

By the construction of the dependency graph, this source corresponds to an alternating cell c in G_F, and in $G_{(F \oplus c)}^*$ the dependency arcs that cross c have the opposite parity and direction as they did in G_F^*. Figure 4.5 illustrates the effect of flipping the edge parities of c. The remainder of R is a smaller region (or possibly two smaller regions, if c was a bridge) that are alternating cell regions in $G' = G_{(F \oplus c)}$. By induction an alternating cell sequence A exists in G' that consists of exactly the interior cells in these regions. In these smaller regions, the cell c is absent, so c does not appear in A. Then by the definition of alternating cell sequences, appending c to A gives an alternating cell sequence in G_F with no repeated cells. □

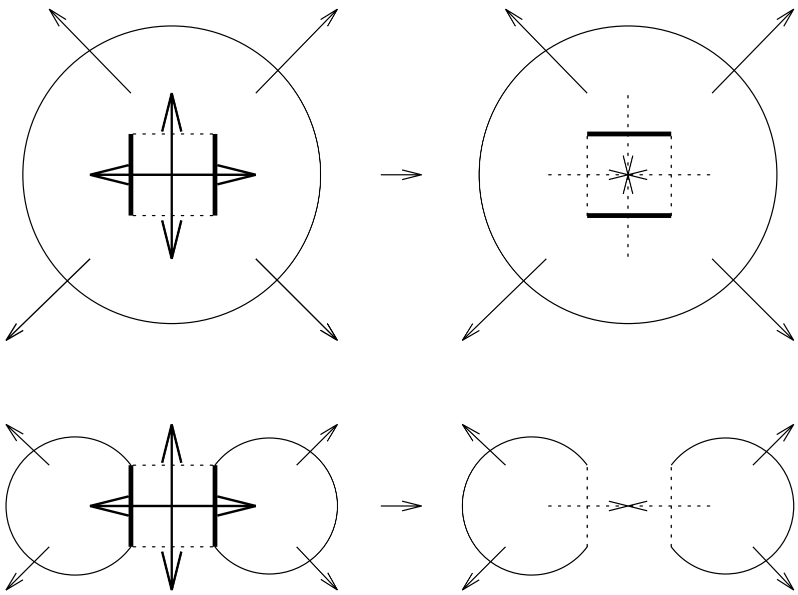


Figure 4.5: The smaller alternating cell region(s) resulting from flipping the edge parities of an alternating cell in the larger region.

Lemma 4.8 *Let $c_1, c_2, \cdots, c_n$ be the interior cells of a region R in G_F, and let S be the set of border edges of R. Then $F \oplus S = F \oplus c_1 \oplus c_2 \oplus \cdots \oplus c_n$.*

Proof All edges not in S that belong to any cells that are flipped belong to exactly two interior cells, so their parity is flipped once and flipped again, back to its original state. All edges in S belong to only one interior cell by definition, and so their parities are flipped exactly once. So flipping the parity of the border edges of R is equivalent to flipping the parity of the edges of each of the interior cells of R in any order. □

Theorem 4.1 *Let F_1 and F_2 be 2-factors of G. Then $F_1 \oplus A = F_2$, where A is an alternating cell sequence in G_{F_1}.*

Proof Let $S = F_1 \oplus F_2$. The proof is by induction on $|S|$.

By Lemma 3.1, there exists an alternating cycle C in S, which as noted, is an alternating cell region. By Lemma 4.7 there exists an alternating cell sequence B consisting of exactly the interior cells of C.

By Lemma 4.8, $F_1 \oplus B = F_1 \oplus C$, and by Lemma 3.2, $F_1 \oplus C = F_1'$, a 2-factor in G. Now, $F_1' \oplus F_2 = (F_1 \oplus C) \oplus F_2 = F_1 \oplus F_2 \oplus C = S \oplus C = S'$, and since $C \subseteq S$, $|S'| < |S|$.

By induction, there exists an alternating cell sequence B' in $G_{F_1'}$ such that $F_1' \oplus B' = F_2$, and by Lemma 4.6, B can be concatenated with B' to give an alternating cell sequence A in G_{F_1}. Finally, $F_2 = F_1' \oplus B' = (F_1 \oplus B) \oplus B' = F_1 \oplus (B \oplus B') = F_1 \oplus A$, as required by the lemma. □

The next observation formalizes a fact that follows from the proof of the theorem:

Observation 4.1 *Let F_1 and F_2 be 2-factors of G, and let S be a partition of $F_1 \oplus F_2$ into edge-disjoint non-intersecting alternating cycles. Then $F_1 \oplus A = F_2$, where A is an alternating cell sequence in G_{F_1} of length no greater than the total area of the cycles in S.*

Proof At each step in the inductive proof of Theorem 4.1, an alternating cell sequence consisting of exactly the interior cells of some cycle in S is appended to the total sequence. Thus the length of the final alternating cell sequence is exactly the total area of the cycles in S. $\square$

The fact that the total number of alternating cell flips required for a transformation is no greater than the total area of the cycles is encouraging since it suggests that an algorithm that flips a sequence of alternating cells might be polynomially bounded. In fact, the number of alternating cell flips is significantly less than this bound in many cases, as will be shown in the next two chapters.

Chapter 5

Alternating Strips

In this chapter it is shown that any 2-factor of a Hamiltonian grid graph without holes must contain one of two types of "alternating strips," which are structures that make up the transformations introduced in the next chapter that reduce the number of components in a 2-factor.

5.1 Border Cells and Boundaries

Definition 5.1 *A* **border cell** *in G_F is a cell with at least one of its edges between vertices in different components in F.*

Definition 5.2 *A border cell in G_F is classified as* **Type I** *if only one of its four edges is in F,* **Type II** *if exactly two adjacent edges are in F,* **Type III** *if exactly two non-adjacent edges are in F, and* **Type IV** *if none of its four edges are in F.*

This definition completely classifies border cells; Figure 5.1 shows each of the four types of cells.

Definition 5.3 *A* **boundary** *is a set of border cells whose dual graph is either a simple cycle or a path beginning and ending with cells that share an edge with the outer face.*

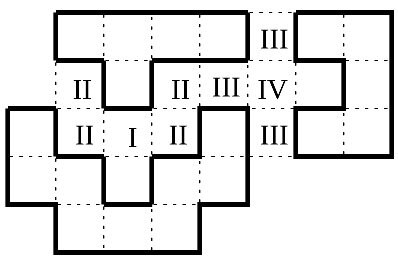


Figure 5.1: A 2-factor showing the four types of border cells.

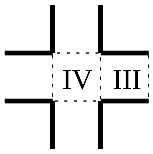


Figure 5.2: A Type IV border cell and the required neighboring Type III border cell.

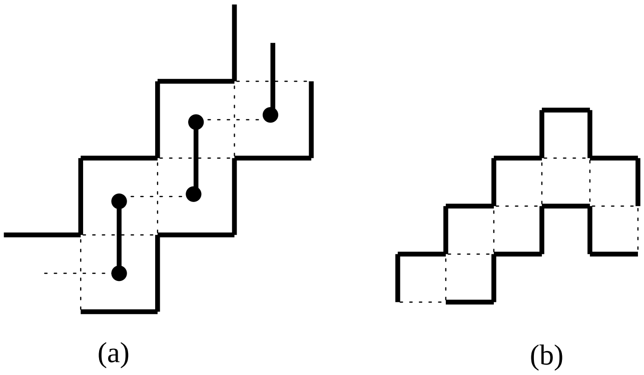


Figure 5.3: The dependency graph edges along a boundary of Type II cells (a), and an illustration of the fact that a boundary of only Type II cells cannot bend (b).

Lemma 5.1 *If a boundary B in G_F contains a Type IV border cell, then B contains a Type III border cell.*

Proof Figure 5.2 shows a Type IV border cell c, and the dark edges surrounding it that are required by the degree constraint of the 2-factor. Without loss of generality, assume that the cell to the right of c is its neighbor on B. The right edge in the cell labeled "III" cannot be a dark edge, or its two dark edges would be in the same component, and it could not be border cell. Thus the cell to the right of c on the same boundary B must be a Type III border cell. □

Lemma 5.2 *If G_F contains a boundary B consisting entirely of Type II cells, then G is not Hamiltonian.*[1]

Proof First note that a boundary consisting entirely of Type II cells cannot have a cyclic dual graph, because that would require a bend in the boundary, which requires a Type I cell as shown in Figure 5.3(b). So the cells in B must have a dual graph that is a path that begins and ends with cells that share an edge with the outer face. Thus the edges of the cells in B must be crossed by dependency graph edges that form an alternating parity path as shown in Figure 5.3(a).

[1]This lemma is originally due to Bridgeman [1] but is restated here with a substantially different proof based on the alternating structures developed in this thesis.

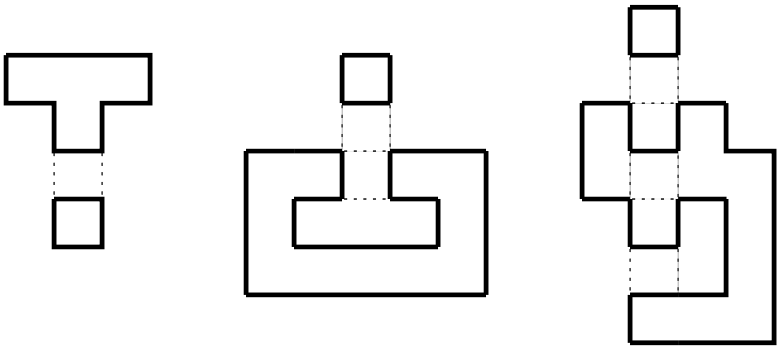


Figure 5.4: Examples of alternating strips.

Now, suppose G is Hamiltonian. Then by Corollary 3.1, F can be transformed into a Hamiltonian cycle by flipping the edge parities of a set of edge-disjoint non-intersecting alternating cycles, S. The parity of one of the edges between the two components separated by B must be flipped by one of the cycles in S, or else $F \oplus S$ would have more than one component, contradicting the choice of S to yield a Hamiltonian cycle. So some cycle in S must cross the alternating parity path in the dependency graph, and since the path crosses the entire graph, the cycle must cross the path again to return to its starting vertex. This contradicts Lemma 4.5, so G must not be Hamiltonian. □

5.2 Alternating Strips

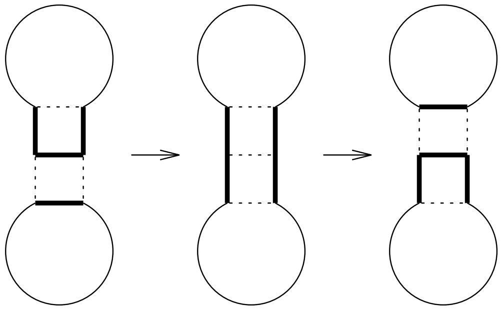
Definition 5.4 *An* **alternating strip** *is an alternating cycle in G_F defined recursively as follows:*

1. *A Type III border cell in G_F is an alternating strip.*
2. *Let c be an alternating cell in G_F and s be an alternating strip in $G_{(F \oplus c)}$. If c shares an edge e with s and either e is in F and s has even area, or e is not in F and s has odd area, then $c \oplus s$ is an alternating strip in G_F.*

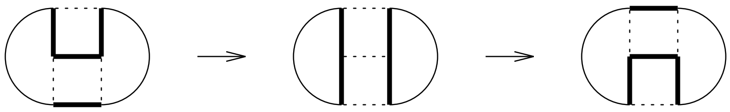
Figure 5.4 shows three examples of alternating strips. An alternating strip either consists of a single Type III border cell or extends in a straight line from a cell d with exactly one edge in the 2-factor (the only type of cell that can be made a Type III border cell by a single adjacent alternating cell flip with the edge parity required). For strips of length at least two, the alternating strip **begins** at cell d and **ends** at the alternating cell at the other end of the strip; for strips of length one, the alternating strip begins at the Type III border cell.

5.2.1 Properties of Alternating Strips

The precise effect of flipping the edge parities of an alternating strip on the number of cycles in the 2-factor is determined in this section.



(a)



(b)

Figure 5.5: The two possible configurations for an alternating cycle with area two as specified in Lemma 5.3. The curved portions of the cycles are purely schematic, and their layout is not meant to preclude the possibility of nested cycles.

Lemma 5.3 *Let C be an alternating cycle with area two in G_F whose interior edge e is in F, and let e_1 and e_2 be edges in $F - C - \{e\}$. Then the following hold:*

1. *the 2-factor $F \oplus C$ contains the same number of components as F does, and*
2. *edges e_1 and e_2 are in the same component in $F \oplus C$ if and only if they are in the same component in F.*

Proof The proof of part one is given first. The parities of the edges of C are completely determined by the specification in the lemma, and the configurations of the affected cycles in F can be broken down into two cases, shown in Figure 5.5. In (a), the alternating cell in C is a (Type III) border cell, so its light edges are between two distinct cycles in F. The three diagrams show the progression of flipping the edge parities of C one alternating cell at a time; the first cell flip merges the two cycles, and the next cell flip breaks the merged cycle, leaving the number of components in $F \oplus C$ the same as the number in F.

In (b), the alternating cell in C is not a border cell, so the dark edges in C are all on one cycle in F. The same progression is shown as in (a); in this case the first alternating cell flip breaks the cycle, and the second rejoins it, again leaving the number of components in $F \oplus C$ the same as the number in F.

For the proof of part two, refer again to Figure 5.5. If either of e_1 or e_2 belong to a cycle unaffected by flipping the edge parities of C (i.e. not shown in the figure), then part two follows trivially.

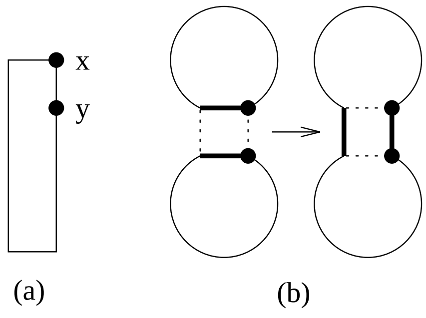


Figure 5.6: An odd alternating strip s (a), with two vertices at the beginning of the strip labeled for the statement of Lemma 5.4, and the possible parity configurations of the relevant edges, if s has area one, in F and $F \oplus s$ (b).

Otherwise, both e_1 and e_2 are on cycles shown in the figure, and there are several cases to consider. If e_1 and e_2 are on the same cycle in F, then they are either both on the curved portion of one or the other of the two cycles in (a), or they are each on the curved portions in (b). In each case, the figure demonstrates that part two of the lemma holds in the backward direction. If e_1 and e_2 are on different cycles in F, then each is on the curved portion of one of the two cycles in (a), and in this case the figure demonstrates that part two holds in the forward direction. □

Definition 5.5 *Alternating strips are classified as* **even** *or* **odd** *according to whether they enclose an even or odd number of cells.*

Lemma 5.4 *Let s be an odd alternating strip in G_F, let e_1 and e_2 be edges in $F - s - \{$the interior edges of $s\}$, and let two vertices belonging to the cell at the beginning of s be labeled as in Figure 5.6(a). Then the following hold:*

1. *the 2-factor $F \oplus s$ contains one fewer component than F does, and*
2. *edges e_1 and e_2 are in the same component in $F \oplus s$ if and only if either they are in the same component in F, or one is in the same component of F as vertex x and the other is in the same component of F as vertex y.*

Proof The proof is by induction on the area of s. If s consists of only one cell, then s is a Type III border cell, and clearly $F \oplus s$ has one fewer component than F does, so part one holds. For part two, refer to Figure 5.6(b), which shows the relevant cycles in F and $F \oplus s$. If one of e_1 and e_2 is in the same component as x and the other is in the same component as y (in F), then they must lie on the two pictured cycles, respectively, and from the figure, it is clear that they are in the same component in $F \oplus s$. If e_1 and e_2 are in the same component in F then they must be on a cycle not pictured in the figure, and so unaffected by the edge parity flips involved. If e_1 and e_2 are in the same component in $F \oplus s$, then they must either be on an unpictured cycle, in which case they are also in the same component in F, or they must be on the pictured cycle in $F \oplus s$, in which case they are either on the same cycle in F or else one must be in the same component as x and the other in the same component as y. So part two of the lemma holds for the base case.

If s has area greater than one, then s has area at least three, and the pair of cells at the end of s constitute an alternating cycle C of the type described in Lemma 5.3. If the alternating cell in C is labeled c_1 and the other cell in C is labeled c_2, then the following are true:

1. $F = F' \oplus c_1 = (F'' \oplus c_2) \oplus c_1 = F'' \oplus (c_2 \oplus c_1) = F'' \oplus C$, and
2. $s = s' \oplus c_1 = (s'' \oplus c_2) \oplus c_1 = s'' \oplus (c_2 \oplus c_1) = s'' \oplus C$, where s'' is by definition an alternating strip in $G_{F''}$.

Since the area of s'' is less than the area of s, by induction $F'' \oplus s''$ has one fewer component than F'' does. By Lemma 5.3, $F'' = F \oplus C$ has the same number of components as F does, so $F'' \oplus s''$ has one fewer component than F does. Now, $F'' \oplus s'' = (F \oplus C) \oplus s'' = F \oplus (C \oplus s'') = F \oplus s$, so $F \oplus s$ has one fewer component than F does, and part one of the lemma holds. Also by induction e_1 and e_2 are in the same component in $F'' \oplus s''$ if exactly the conditions stated in part two hold with respect to F''. As noted, $F'' \oplus s'' = F \oplus s$, and by Lemma 5.3, e_1 and e_2 are in the same component in F'' if and only if they are in the same component in F, so the second part of the lemma holds with respect to $F \oplus s$ and F as required. □

Lemma 5.5 *Let s be an even alternating strip in G_F. If the alternating cell at the end of s is not a Type III border cell, $F \oplus s$ has the same number of components as F does.*

Proof Let c be the alternating cell at the end of s, and let $F' = F \oplus c$. Then by definition $s' = s \oplus c$ is an alternating strip in $G_{F'}$. Further, s' is an odd alternating strip, so $F' \oplus s'$ has one fewer component than F' does, by Lemma 5.4. Since c is not a Type III border cell, it must be that F' has one more component than F does; if F' had fewer components than F, then flipping the edge parities of c would have the effect of merging two cycles in F, implying that c is a Type III border cell, contrary to the statement of the lemma. Since $F' \oplus s$ has one fewer component than F' and F' has one more component than F, $F' \oplus s' = F' \oplus (s \oplus c) = (F' \oplus c) \oplus s = F \oplus s$ has the same number of components as F does. □

Lemma 5.6 *Let s be an even alternating strip in G_F that begins on a border cell and does not end with a Type III border cell. Then the alternating cell at the end of s is a Type I border cell in $G_{(F \oplus s)}$.*

Proof Let c be the alternating cell in question and let $F' = F \oplus c$. The first two diagrams in Figure 5.7(a) show the cycle to which c belongs and the configuration of c in $G_{F'}$, respectively. Notice that in $G_{F'}$, the edges labeled x and y are in different components because c is not a Type III border cell.

By the definition of an alternating strip, $s' = s \oplus c$ is an (odd) alternating strip in $G_{(F \oplus c)}$. Suppose that x and y are in the same component in $F' \oplus s'$. Figure 5.7(b) shows

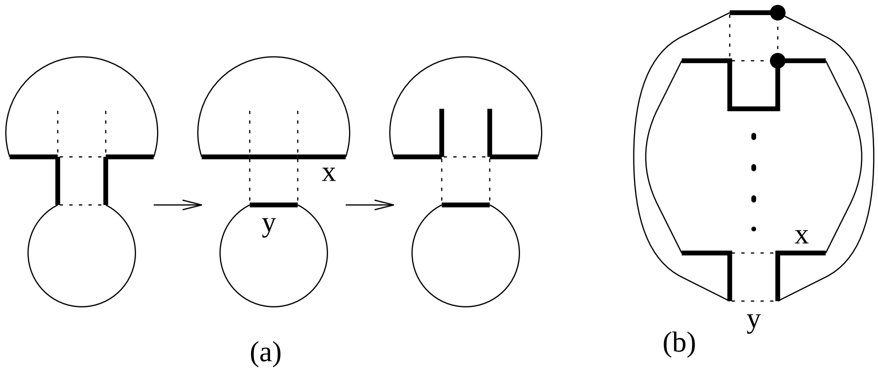


Figure 5.7: The cycles before and after the edge parities of the alternating cell at the end of an alternating strip are flipped, and then after the edge parities of the cell above it are flipped. The curved portions of the cycles are schematic.

the configuration that must arise in this case; this is justified by Lemma 5.4 (part two): since x and y are not in the same component in F', it must be that x is in the same component as one of the dotted vertices in the figure and y is in the same component as the other dotted vertex, all in F'. The figure extrapolates backward to the configuration that must exist in F. Note that the top cell of s is not a border cell, contrary to the statement of the lemma, so x and y must be in different components in $F' \oplus s' = (F \oplus c) \oplus (s \oplus c) = F \oplus s$, and so c is a border cell in $F \oplus s$. Finally, the third diagram in Figure 5.7(a) shows the configuration of c in $F \oplus s$, and it is clear that c is a Type I border cell. $\square$

If G contains no Type III border cells when the edge parities of an even alternating strip s is flipped, then the new Type I border cell lies on a boundary in $G_{(F \oplus s)}$ that did not exist in G_F. This boundary is the boundary **created** by s.

Lemma 5.7 *Let G_F contain no Type III border cells, and let s be an even alternating strip in G_F. If $G_{(F \oplus s)}$ contains a Type III border cell, then it contains a Type III border cell on the boundary created by s.*

Proof First the cells that share at least one edge with s are considered, then alternating cells in F elsewhere in the graph that may potentially become Type III border cells in $F \oplus s$. Figure 5.8 shows an even alternating strip with as many neighboring edges added as can be determined from the requirement that G_F contain no Type III border cells and the degree constraint of the 2-factor. The bottom two dark edges in the figure are not determined in this manner; if either of the two extend to the side, then a Type III border cell is immediately present on the boundary created by s, so this case is simple.

None of the interior cells of s or the cells that share an edge with s are alternating cells in $F \oplus s$ (and so candidates for being Type III border cells) except for the cells marked with an "x" in the figure. However, in $F \oplus s$, the dark edges of these cells are in the same component (since by definition, the cell at the beginning of s is a border cell), so they cannot be Type III border cells.

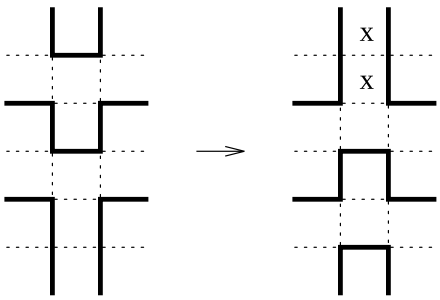


Figure 5.8: An even alternating strip before and after its edge parities are flipped.

The only candidate cells in F for becoming Type III cells in $F \oplus s$ are then the alternating cells in F (and $F \oplus s$) that share no edges with s. Let d be such a cell, and let e_1 and e_2 be its dark edges in G_F. Recall that by the definition of an alternating strip, if c is the alternating cell at the end of s, then $s' = s \oplus c$ is an (odd) alternating strip in $G_{(F \oplus c)}$. By assumption, c is not a Type III border cell in F, so it must be a Type III border cell in $F \oplus c$. Also by assumption, e_1 and e_2 are in the same component in G_F, since G_F contains no Type III cells. So if e_1 and e_2 are in different components in $F \oplus c$, then they must be in two components separated by the boundary that includes c, or equivalently, d must be on the same boundary as c.

By Lemma 5.4, e_1 and e_2 are in different components in $(F \oplus c) \oplus s'$ only if they are in different components in $F \oplus c$ and it is not the case that one is in the same component as vertex x and the other is in the same component as vertex y (in Figure 5.6(a)). As noted, the only way for e_1 and e_2 to be in different components in $F \oplus c$ is if d and c are on the same boundary. In other words, if f_1 and f_2 are the dark edges of cell c in $G_{(F \oplus c)}$, then f_1 and e_1 are in one component and f_2 and e_2 are in the another component. Again by Lemma 5.4, f_1 and e_1 must remain in the same component in $F \oplus c \oplus s'$ and f_2 and e_2 must remain in the same component in $F \oplus c \oplus s'$. Note that $F \oplus c \oplus s' = F \oplus s$, and so if e_1 and e_2 are in different components in $F \oplus s$ (i.e. d is a Type III border cell), then f_1 and f_2 belong to the same two components in $F \oplus s$ respectively, and c is on the new boundary created by s which implies that d is on the boundary created by s. □

5.2.2 Existence

Lemma 5.8 *Let R be an alternating cell region in G_F, and let c be a Type I border cell in R such that the dependency arc that crosses c's dark edge is oriented in the same direction as the dependency arcs that cross R, with respect to the interiors of c and R. Then c begins an alternating strip that is wholly contained within R.*

Proof Figure 5.9(a) shows c and the enclosing region with the dependency arc directions specified in the lemma. The four bent dark edges in (a) are forced by the degree constraint of the 2-factor. If the lower edge of the second cell pictured is a dark edge, then the dependency arc that crosses it is light, so the dependency arcs along the center of the strip

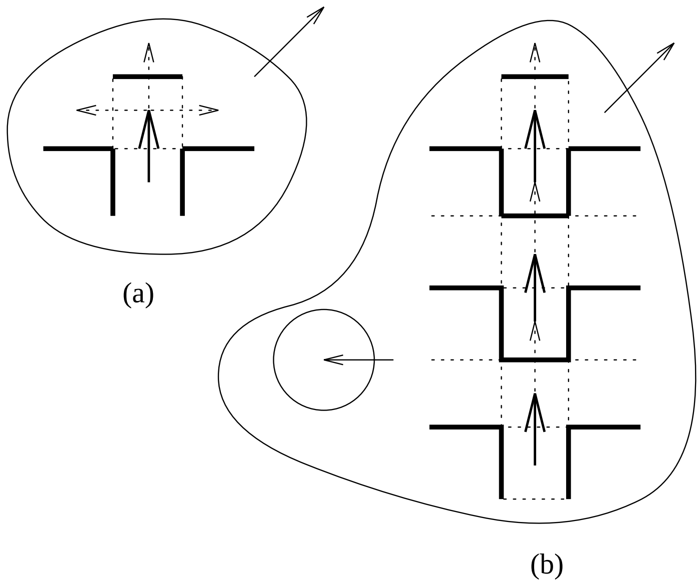


Figure 5.9: The Type I border cell at the beginning of an alternating strip and its enclosing region (a), and an entire alternating strip in an enclosing region (b).

alternate parity for one more edge. In addition, if the bottom edge is dark, then the degree constraint forces the two side edges of the next cell to be light. This pattern repeats as long as the bottom edge of each new cell has the same parity as the side edges. The dependency arcs along the center of the strip form an alternating path, so the pattern must terminate since this alternating path cannot pass outside of the region by the choice of the relative orientations of the dependency arcs that cross the boundary of the region.

When the pattern terminates, the final edge completes an alternating strip that is wholly contained within R. □

Chapter 6

Alternating Strip Sequences

In this chapter alternating strips are assembled into sequences that may be used by an algorithm to work from a 2-factor toward a Hamiltonian cycle. Each sequence is an "improving" transformation; flipping the edge parities of all of its strips results in a 2-factor with one fewer cycle than the original 2-factor.

The alternating strips are linked together so that each new boundary created by an even alternating strip is "removed" by the next alternating strip in the sequence. As long as each additional alternating strip is an even one, the process is repeated. When an odd alternating strip is finally located, the procedure terminates, having identified a complete alternating strip sequence. In addition to proving that the strips can be chosen in this manner in a Hamiltonian grid graph without holes, it must be shown that the process terminates.

6.1 Cell Indices

The proof that the process of identifying successive strips in an alternating strip sequence terminates requires a measure of progress which is based on how "close" a 2-factor is to a given Hamiltonian cycle. To establish such a measure, the "orientation" of an alternating cycle is defined, using the fact from Lemma 4.4 that the dependency arcs crossing an alternating cycle are consistently oriented with respect to the interior of the cycle. To subsume the variation arising from the two different possible orientations of the dependency graph, the definition considers only the *change* in direction of the arcs crossing successive nested cycles.

Definition 6.1 *The* **orientation** *of each cycle in a set S of edge-disjoint non-intersecting alternating cycles in G_F is labeled as either* **positive** *or* **negative***, according to the following procedure:*

1. *Label every cycle that is enclosed by no other cycles in S with a positive orientation, and all cycles nested immediately within these cycles that are crossed by dependency arcs oriented in the opposite direction as the ones crossing the outer cycle with a negative orientation.*

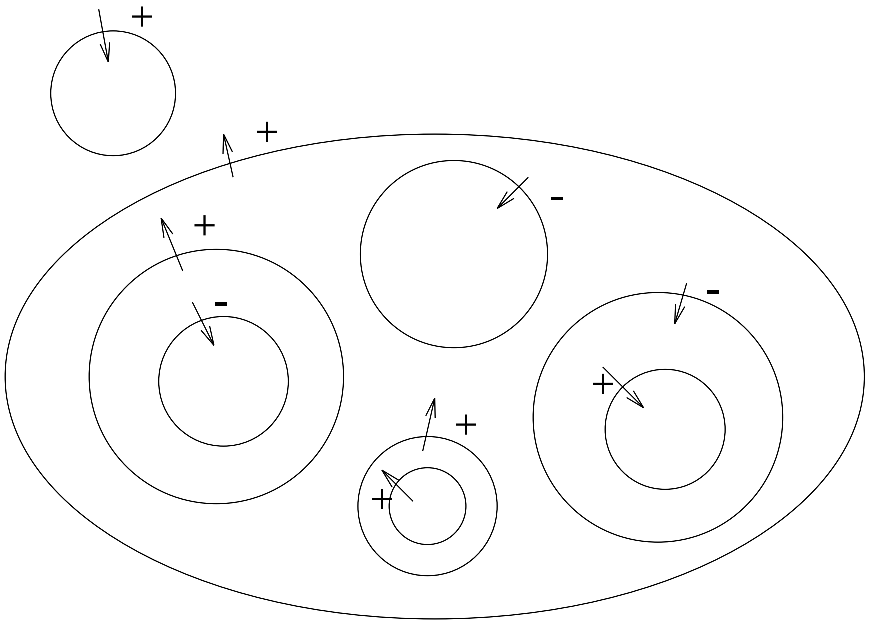


Figure 6.1: The orientations for a set of cycles.

2. *Remove from S all cycles whose orientation was determined in the previous step and repeat step one.*

Figure 6.1 shows the orientations for a sample set of cycles. The motivation for the labeling of these orientations should become apparent with the next definitions. For the remainder of this section, let F_1 and F_2 be 2-factors of G, and let S be a partition of $F_1 \oplus F_2$ into edge-disjoint non-intersecting alternating cycles.

Definition 6.2 *The* **index of cell** c **in** S *is calculated by summing* $+1$ *for positively oriented cycles that enclose* c *and* -1 *for negatively oriented cycles that enclose* c*.*

Definition 6.3 *The* **index of** F_1 **in** S*, denoted* $I(F_1, S)$*, is the sum over all cells in* G *of the indices of the cells in* S*.*

Note that there are potentially different ways to partition $F_1 \oplus F_2$ to get S, and these might yield different index values for the same two 2-factors. However, with the next lemma, the index can be seen as a sufficient measure of progress:

Lemma 6.1 *There exists an alternating cell sequence* A *in* G_{F_1} *of length no greater than the* $I(F_1, S)$ *such that* $F_1 \oplus A = F_2$*.*

Proof The proof is by induction on $I(F_1, S)$. If $I(F_1, S) = 0$, then there can be no cycles in S, and so $F_1 = F_2$, and no alternating cell flips are required to transform F_1 into F_2.

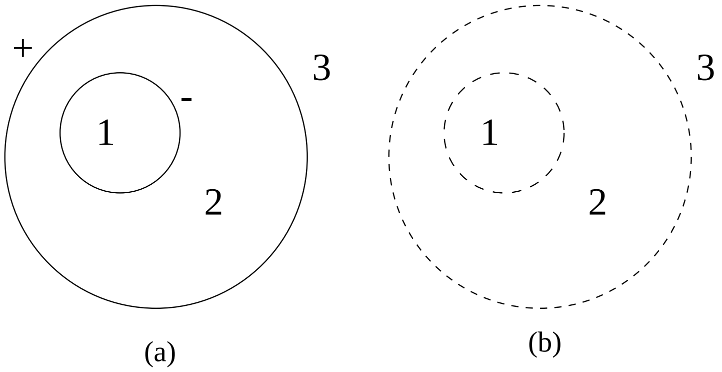


Figure 6.2: The effect of removing an alternating cell region on the indices of cells.

Otherwise, S must contain some cycles, so let T be the set of the outermost cycles and those cycles nested one level inside the outermost level with negative orientations. This set of cycles forms a set of area-disjoint regions in G_F, and in fact the regions are alternating cell regions by the choice of inner cycles to have negative orientation. By Lemma 4.7, alternating cell sequences $A_1, A_2, \cdots, A_n$ exist in G_{F_1} that consist of exactly the interior cells of each region, respectively. By Lemma 4.6, these sequences can be concatenated into a single alternating cell sequence, A'.

Now, $F_1 \oplus A' = F_1'$, a 2-factor, and $F_1' \oplus F_2 = (F_1 \oplus A') \oplus F_2 = F_1 \oplus F_2 \oplus A' = S \oplus A'$. By Lemma 4.8, $S \oplus A' = S \oplus T = S'$, and since $T \subseteq S$, $|S'| < |S|$. In S' the indices of exactly those cells inside the regions of T are reduced by one. This is true because an enclosing cycle has been removed that contributed one to the indices of all interior cells, but an enclosing cycle of those cells not within the regions has also been removed that contributed a negative one to their indices. Furthermore, the indices of no other cells are affected, since every cell not within the regions is either outside the outermost cycles, and thus has index zero, or is inside a cycle with negative orientation in T. The index of these last cells is reduced by -1 for the cycle with negative orientation, and $+1$ for the outermost cycle, for a net change of zero.

Figure 6.2 illustrates this analysis – in (a), two cycles that form a region are pictured; in (b) the cycles are removed. The cells in the area labeled with "1" are outside the region, as are the cells in the area labeled with "3", and it is clear that their indices do not change. The cells in the area labeled with "2" are inside the region, and their indices are reduced by one.

Therefore, $I(F_1', S') < I(F_1, S)$, and so by induction, an alternating cell sequence B exists in $G_{F_1'}$, of length no greater than $I(F_1', S')$, such that $F_1' \oplus B = F_2$. By Lemma 4.6, concatenating A' and B gives an alternating cell sequence A such that $F_1 \oplus A = F_2$. By the above argument, the length of A' is exactly $I(F_1, S) - I(F_1', S')$, so the total length of the alternating cell sequence A in G_{F_1} is no greater than $I(F_1', S') + (I(F_1, S) - I(F_1', S')) = I(F_1, S)$. $\square$

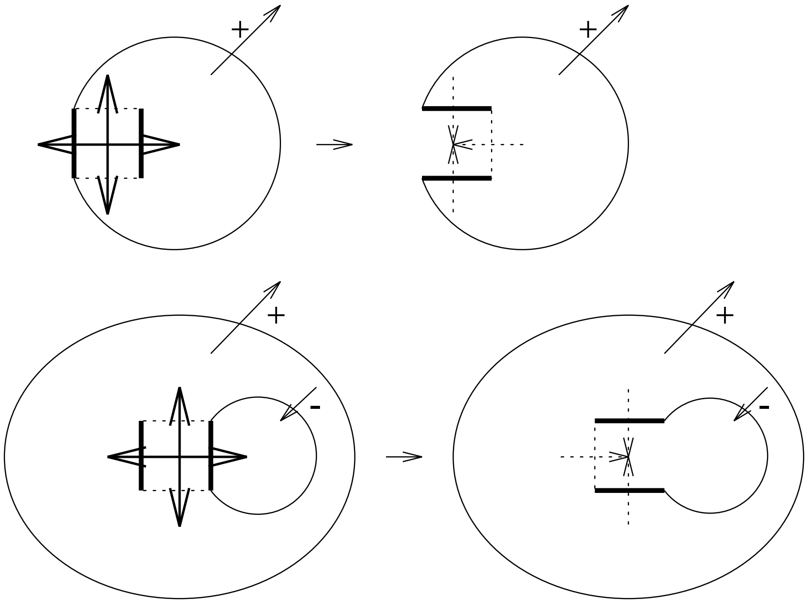


Figure 6.3: The configurations of an improving cell that shares one edge with the cycles in the symmetric difference.

6.2 Improving Cells

Definition 6.4 *A cell c is on the* **immediate interior** *of a cycle C in S if c is an interior cell of C and an interior cell of no other cycle in S that is nested within C.*

Definition 6.5 *An* **improving cell** *is an alternating cell c on the immediate interior of a positively oriented cycle C in S, for which the dependency arcs crossing c are oriented in the same direction as the dependency arcs crossing C, with respect to the interiors of the cycles.*

Lemma 6.2 *Let c be an improving cell in G_{F_1}, and let $F_1' = F_1 \oplus c$. There exists a partition S' of $F_1' \oplus F_2$ into edge-disjoint non-intersecting alternating cycles such that $I(F_1', S') = I(F_1, S) - 1$.*

Proof If c shares no edges with any of the cycles in S, then simply let $S' = S + \{c\}$. In $G_{F_1'}$ the edges of c are crossed by dependency arcs that have the opposite parity and direction as they do in the oriented dependency graph $G^*_{F_1}$. Therefore the dependency arcs that cross cycle c in $G_{F_1'}$ are oriented in the opposite direction as the arcs that cross the enclosing cycle C, so c has negative orientation in S'. The index of the cell c in S' is then one less than it is in S, and no other cells are affected since the added cycle c is an innermost cycle in S'. So $I(F_1', S') = I(F_1, S) - 1$.

If c shares edges with the cycles in S, then a number of cases must be considered. Some of these cases (in which exactly one or two edges of c are shared) are illustrated in Figures

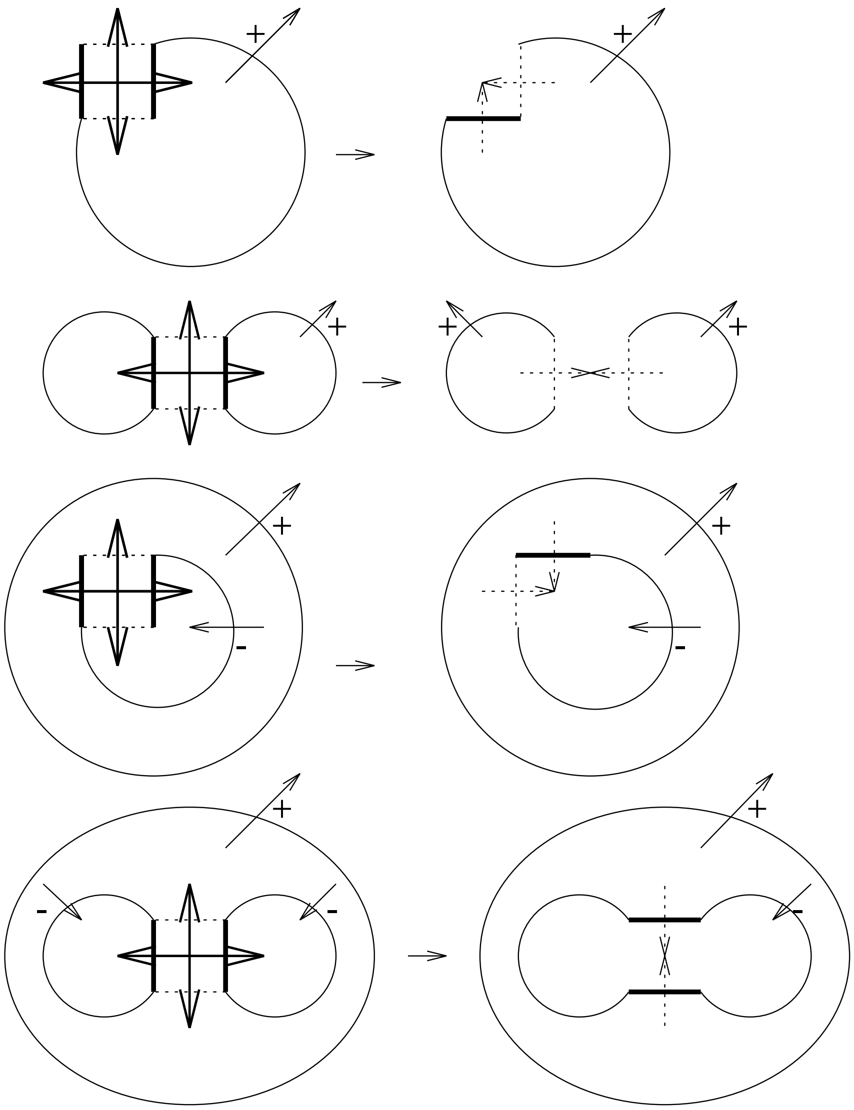


Figure 6.4: Four of the six configurations of an improving cell that shares two edges with the cycles in the symmetric difference.

6.3 through 6.5. In all of these cases, the index of the alternating cell in S' is one less than it is in S, either because in S' one fewer positively oriented cycle encloses c than in S, or because in S' one more negatively oriented cycle encloses c than in S. In the cases in Figure 6.5, in addition, the interior cells of a negatively oriented alternating cycle are removed from the interior of that cycle *and* from the interior of a positively oriented cycle, leaving their index unchanged. In the pictured cases, the new set S' of edge-disjoint non-intersecting

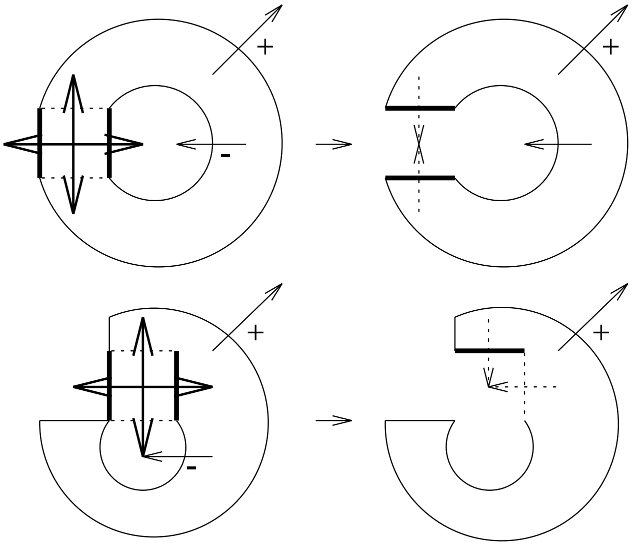


Figure 6.5: The remaining two of six configurations of an improving cell that shares two edges with the cycles in the symmetric difference.

alternating cycles is shown in the figure, as is the new 2-factor F_1', and since the index of c is reduced by one and no other cell indices are changed, $I(F_1', S') = I(F_1, S) - 1$. The remaining cases are similar, and have identical effects on the index of c and the surrounding cells. □

6.3 Identification of Alternating Strips

Lemma 6.3 *Let F_1, F_2, S, and G be defined as above. If S contains an edge e that crosses a boundary in G_{F_1} consisting of entirely Type I and Type II border cells with at least one Type I border cell, then there exists a Type I border cell c on that boundary with the following two properties:*

1. *cell c is on the immediate interior of a positively oriented cycle C in S, and*
2. *the dependency arc that crosses the dark edge of c is oriented in the same direction as the dependency arcs that cross C, with respect to the interiors of c and C.*

Proof The proof describes an alternating "walk" in the oriented dependency graph that begins with the dependency arc that crosses e and ends at c. Let C' be the cycle in S that contains edge e. The walk proceeds in one of the two directions with respect to the interior of C' depending on the orientation of C' in S. If C' is positively oriented, then the walk proceeds toward the interior of C'; otherwise, the walk proceeds away from the interior of C'.

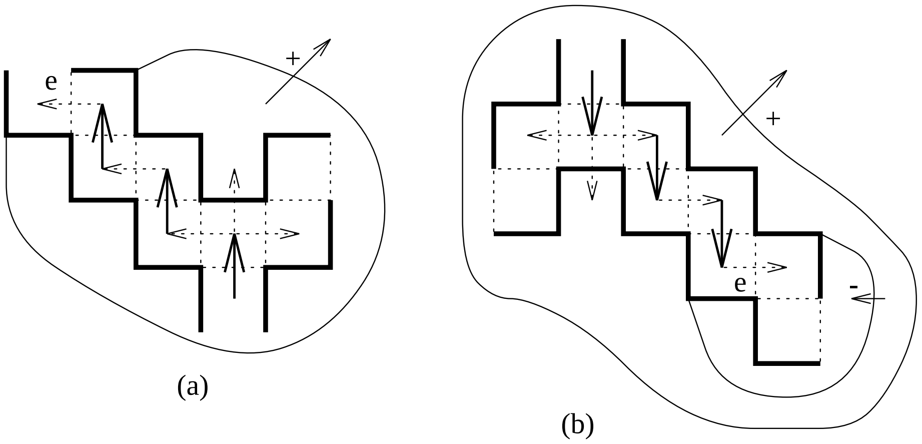


Figure 6.6: The two "walks" along dependency arcs on a boundary to locate the alternating strip specified in Lemma 6.3.

Figure 6.6 illustrates the two possible walks, assuming without loss of generality one of the two possible directions on the dependency arc that crosses e. In (a), e lies on a positively oriented cycle, so the alternating path proceeds inward with respect to this cycle. By Corollary 4.5, this path cannot cross C' again, so C' encloses the entire path. It is possible for the alternating path to cross additional cycles in S that are nested within C', but these cycles must then be crossed by dependency arcs oriented in the same direction as the arcs that cross C', and so must also have positive orientation. So when the path reaches a Type I border cell c, that cell is within a cycle C of S with positive orientation, and the arc that crosses the dark edge of c is oriented in the same direction as the arcs that cross C.

In (b), e lies on a negatively oriented cycle, so the alternating path proceeds outward. Thus the path moves into a positively oriented cycle (since negatively oriented cycles cannot be outermost or immediately nested within each other, by the definition of orientation). Note that this cycle and C' form an alternating cell region, and so by Lemma 4.5, the alternating path cannot cross the region boundary again, so it is wholly contained within the positively oriented cycle. By the same argument as for (a), this path must reach a Type I border cell c, after crossing only into additional positively oriented cycles.

In this case, again, the dependency arc that crosses c is oriented consistently with the dependency arcs that cross the immediately enclosing (positively oriented) cycle. So the lemma holds. □

Lemma 6.4 *Let $F_1, F_2, S,$ and G be as defined in Lemma 6.3 and let c be the Type I border cell identified in Lemma 6.3. Then there exists an alternating strip that begins at c and ends at an improving cell.*

Proof By Lemma 5.8, an alternating strip s must exist that begins at c. From Lemma 6.3, cell c is immediately enclosed in an alternating cycle with positive orientation. Figure 6.7(a) shows such an alternating strip enclosed by a positively oriented alternating cycle.

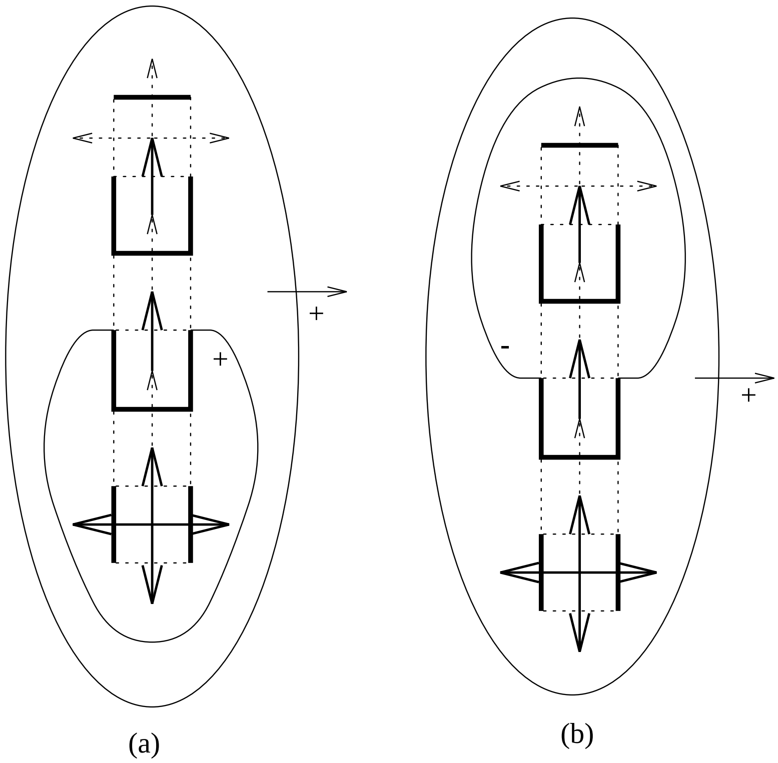


Figure 6.7: Alternating strips and their enclosing positively oriented alternating cycles.

Due to the orientation of the dependency arcs along the center of the strip, the strip can only pass into additional positively oriented cycles, as shown in (a).

Suppose the alternating strip crosses an edge of a negatively oriented cycle, as in (b). Then that cycle must enclose the entire upper portion of the strip by Lemma 4.5, since otherwise the alternating path of dependency arcs along the center of the strip crosses the cycle twice. But c is chosen to be on the *immediate* interior of a positively oriented cycle, which by Lemma 5.9 must enclose all of the strip. The negatively oriented cycle and this positively oriented cycle must then cross, which contradicts the choice of S to be a set of non-intersecting cycles.

The end cell of the strip must then be in a positively oriented cycle, and by the original choice of c such that its dark edge is crossed by dependency arcs oriented in the same direction as the dependency arcs that cross the enclosing alternating cycle, the end cell must also be crossed by dependency arcs directed so as to make it an improving cell. Thus an alternating strip that begins at c exists, and the cell at the end of this strip is an improving cell. □

Lemma 6.5 *Let s be the alternating strip of length n identified in Lemma 6.4, let F_1, F_2, S, and G be defined as in Lemma 6.4, and let $F_1' = F_1 \oplus s$. There exists a partition S' of $F_1' \oplus F_2$*

into edge-disjoint non-intersecting alternating cycles such that $I(S', F_1') = I(S, F_1) - n$.

Proof The proof is by induction on n. By Lemma 6.4, the cell c at the end of s is an improving cell, and so by Lemma 6.2, letting $F_1'' = F_1 \oplus c$, there exists a partition, S'' of $F_1'' \oplus F_2$ such that $I(F_1'', S'') = I(F_1, S) - 1$. If $n = 2$, then $s' = s \oplus c$ is a Type III border cell, and a review of the configurations shown for the proof of Lemma 6.2 shows that s' is an improving cell as well. So by Lemma 6.2, there exists a partition S' of $F_1' = F_1'' \oplus s'$ into edge-disjoint non-intersecting alternating cycles, such that $I(F_1', S') = I(F_1'', S'') - 1$. But $I(F_1'', S'') = I(F_1, S) - 1$, so $I(F_1', S') = I(F_1, S) - 2$ as required by the lemma.

Otherwise, $n > 2$. The cell c does not include the edge e used originally to locate s in Lemma 6.3, since then c would be a Type III border cell contrary to the statement of Lemma 6.3. Thus e is on one of the cycles in S'' and after considering the possible configurations in the figures accompanying the proof of Lemma 6.2, it is clear that the "walk" specified in Lemma 6.3 proceeds in the same direction given S'' as it does given S.

No dependency arc directions are different in $G_{F_1''}$ than they are in G_{F_1} save the ones crossing the edges of C. Thus the same "walk" specified in Lemma 6.3 and beginning at e will identify an alternating strip $s' = s \oplus c$ of length $n - 1$ in $G_{F_1''}$. By induction, there exists a partition of S' of $F_1'' \oplus s' = F_1 \oplus c \oplus s' = F_1 \oplus s = F_1'$ into edge-disjoint non-intersecting alternating cycles such that $I(F_1', S') = I(F_1'', S'') - (n - 1)$. Recall that $I(F_1'', S'') = I(F_1, S) - 1$, so $I(F_1', S') = I(F_1, S) - n$ as required by the lemma. □

6.4 Alternating Strip Sequences

An alternating strip sequence is a transformation of a 2-factor that reduces the number of components in the 2-factor and is built from a sequence of alternating strips. Like individual alternating strips, an alternating strip sequence can be said to **begin** on the boundary that contains the Type III cell or the Type I cell that begins the first alternating strip.

Definition 6.6 *An* **alternating strip sequence** *in* G_F *is defined recursively as follows:*

1. *an odd alternating strip is an alternating strip sequence, and*

2. *if* G_F *contains no Type III border cells,* s *is an even alternating strip in* G_F*, and* $A = (a_1, a_2, \cdots, a_n)$ *is an alternating strip sequence in* $G_{(F \oplus s)}$ *that begins on the boundary created by* s*, then* $(s, a_1, a_2, \cdots, a_n)$ *is an alternating strip sequence in* G_F.

The only alternating strip sequence of length one is a single odd alternating strip, which may be a Type III border cell. (Recall that a Type III border cell is an odd alternating strip). Then by the definition, an alternating strip of length greater than one consists of a sequence of even alternating strips followed by a single odd alternating strip which terminates the sequence.

Lemma 6.6 *Let* A *be an alternating strip sequence in* G_F. *Then* $F \oplus A$ *has one fewer component than* F *does.*

Proof The proof is by induction on n, the number of alternating strips in A. If A consists of a single alternating strip s, then it must be an odd alternating strip. By Lemma 5.4, $F \oplus A = F \oplus s$ has one fewer component that F does, so the lemma holds.

If $n > 1$, then let s be the first alternating strip in A. By the definition of an alternating strip sequence, s must be an even alternating strip, and G_F must contain no Type III border cells. So by Lemma 5.5, $F \oplus s$ has the same number of components as F does. Now, by definition, $A_{2,n}$ is an alternating strip sequence in $G_{(F \oplus s)}$ of length $n-1$, so by induction, $(F \oplus s) \oplus A_{2,n}$ has one fewer component that $F \oplus s$ does. But $(F \oplus s) \oplus A_{2,n} = F \oplus A$, so the lemma holds. □

Lemma 6.7 *Let F be a 2-factor in G with more than one component. If G is Hamiltonian then an alternating strip sequence exists in G_F.*

Proof First, if G_F contains an odd alternating strip, then clearly the lemma holds. Otherwise, it can be assumed that G_F contains no odd alternating strip and in particular, G_F contains no Type III border cells.

Let H be a Hamiltonian cycle in G, and let S be a partition of $F \oplus H$ into edge-disjoint non-intersecting alternating cycles, and let B be some boundary in G_F. By assumption, G_F contains no Type III border cells, and by Lemma 5.1, G_F contains no Type IV border cells either. So B must consist of entirely Type I and Type II border cells, and by Lemma 5.2, B must contain at least one Type I cell since G is Hamiltonian. There must be some edge e in a cycle in S that is between the two components separated by B, or else the 2-factor that results from flipping the edge parities of the cycles in S is not a Hamiltonian cycle. By Lemma 6.3 and 6.4, an alternating strip s can be identified.

If s is an odd alternating strip, then $A = (s)$ is an alternating strip sequence in G_F. Otherwise, if $G_{(F \oplus s)}$ contains any Type III border cells, then by Lemma 5.7, there exists a Type III border cell c on the boundary created by s, so the sequence $A = (s, c)$ constitutes an alternating strip sequence in G_F. These two possibilities are the "terminal" cases.

Otherwise, by Lemma 6.5, there exists a partition S' of $(F \oplus s) \oplus H$ into edge-disjoint non-intersecting alternating cycles such that $I(F \oplus s, S') < I(F, S)$. The graph $G_{(F \oplus s)}$ contains no Type III border cells, so an additional alternating strip that begins on the boundary B' created by s can be identified in the same manner.

This process must terminate since the index of the 2-factor is reduced with each successive strip, and when the index is zero, by Lemma 6.1, the 2-factor is equal to H. This is impossible if every strip identified is an even alternating strip, since by Lemma 5.5 flipping the edge parities of each even alternating strip identified in this process leaves the number of components in the 2-factor unchanged. So a terminal case must be reached, at which point an alternating strip sequence is identified. □

Lemma 6.8 *If $A = (a_1, a_2, \cdots, a_n), n \geq 3$ is an alternating strip sequence in G_F, then $G_i = G_{(F \oplus A_{1,i})}$ contains no Type III border cells, for all $i < n-1$.*

Proof By Lemma 5.7, if a Type III cell exists in G'_i then a Type III cell exists that lies on the boundary created by strip a_i. But by the definition of an alternating strip sequence, this Type III cell should terminate the sequence at $i+1$, but $n > i+1$, so this is a contradiction. □

Theorem 6.1 *Let G be a Hamiltonian grid graph without holes, F a 2-factor in G, and H a Hamiltonian cycle in G. Then $F \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_k = H$, where each A_i is an alternating strip sequence in $G_{(F \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_{i-1})}$ and k is the number of components in F minus one.*

Proof The proof is by induction on the number of components in F. If F has only one cycle then it is already Hamiltonian; this serves as a base case. Otherwise, by Lemma 6.7, G_F contains an alternating strip sequence A, and by Lemma 6.6, $F' = F \oplus A$ has one fewer component than F does. By induction, there exist alternating strip sequences $A_1, A_2, \cdots, A_{k-1}$ such that $F' \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_{k-1} = H$, and each A_i is an alternating strip sequence in $G_{(F' \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_{i-1})}$. Thus $(F \oplus A) \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_{k-1} = H$, where each A_i is an alternating strip sequence in $G_{((F \oplus A) \oplus A_1 \oplus A_2 \oplus \cdots \oplus A_{i-1})}$, and A is an alternating strip sequence in G_F, so the theorem holds. $\Box$

Chapter 7

Static Alternating Strip Sequences

In an alternating strip sequence consisting of more than one strip, the individual alternating strips may overlap and share edges, so that the region of a strip that occurs later in the sequence may not be recognizable as an alternating strip until the edge parities of all of the previous strips have been flipped. An efficient algorithm that identifies alternating strip sequences must be able to identify the *entire* sequence in the graph "statically"; otherwise, a search of candidate sequences might quickly become exponential.

This chapter explains how to eliminate extraneous portions of an alternating strip sequence in order to prove the existence of an alternating strip sequence that is visible "at the top level"–before any edge parity flips have occurred.

7.1 Definition

Definition 7.1 *An alternating strip sequence $A = (a_1, a_2, \cdots, a_n)$ is* **static in** G_F *if the strips in A are area-disjoint and the following holds:*

If e is an edge that is shared by two strips in A, then it is shared by a_i and a_{i+1}, and e is a dark edge on the side of the alternating cell at the end of a_i in $G_{(F \oplus A_{1,i-1})}$ and a light edge on the side of the cell at the beginning of a_{i+1} in $G_{(F \oplus A_{1,i})}$.

With the exception of the shared edges permitted by the definition, a static alternating strip sequence should be thought of as an alternating strip sequence in which no strip's edges or vertices "touch" any other strip's edges or vertices.

Static alternating strip sequences can be derived from ordinary alternating strip sequences in such a way that the area of the derived static alternating strip sequence is always contained in the area of the original one. Figure 7.1 shows an alternating strip sequence in (a) that is not static. The strip that begins at the cell numbered "3" shares edges with the strip that begins at the cell numbered "1". In (b), however, a static alternating strip sequence is shown that is derived from the sequence in (a).

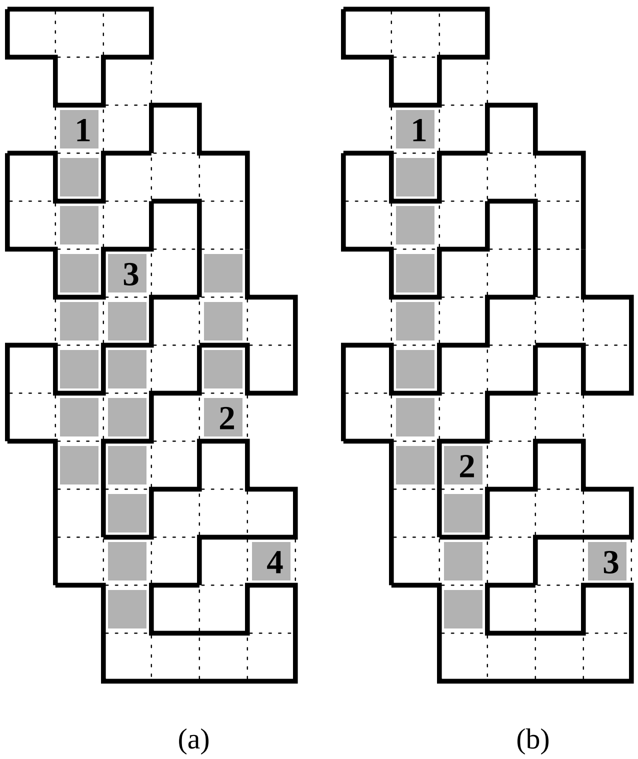


Figure 7.1: An alternating strip sequence (a) that is not static, and the static alternating strip sequence derived from it (b). The numbers label the strips in the order that they appear in the alternating strip sequences.

7.2 Boundaries and the Dual

Definition 7.2 *The* **dual with respect to** F*, denoted* G_F^**, is the dual of* G *(including a vertex for the outer face), with edges that cross an edge of* F *deleted.*

For the remainder of this thesis, G_F^* denotes the dual with respect to F as just defined, and no longer the dependency graph of Chapter 4.

Lemma 7.1 *Let* G_F *contain no Type III border cells, let* s *be an even alternating strip in* G_F*, and let* x *and* y *be two vertices in* G_F^* *that correspond to cells not in* s*. If there exists a path* p *between* x *and* y *in* G_F^* *that does not include the vertex in* G_F^* *that corresponds to the alternating cell at the end of* s*, then there exists a path* p' *between* x *and* y *in* ${G^*}_{(F \oplus s)}$*, with the following properties:*

1. *vertex* v *on* p *corresponds to an alternating cell in* G_F *if and only if* v *is also on* p' *and corresponds to an alternating cell in* $G_{(F \oplus s)}$*, and*

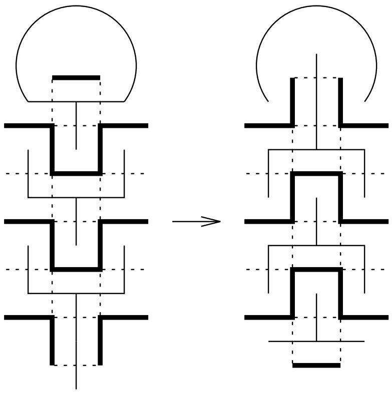


Figure 7.2: The edges of G_F^* (light solid lines) affected by flipping the edge parities of an even alternating strip.

2. *if v is a vertex on p that corresponds to a cell in G_F that is not in s, then v is a vertex on p' in $G^*_{(F\oplus s)}$.*

Proof Figure 7.2(a) shows s and the edges of G_F^* that cross it. Since s begins on a Type I border cell, that cell must be part of a boundary, hence the cycle in G_F^* at the top of the figure. Recall that G_F^* includes a vertex for the outer face, so *all* boundaries are cycles in G_F^*. The only potential sections of a path between x and y in G_F^* that cross edges of s whose parity changes in $G_{(F\oplus s)}$ are the horizontal sections of G_F^* in the figure. The portion that includes the vertical edge through the alternating cell is specifically excluded in the statement of the lemma. Figure 7.2(b) shows the configuration of $G^*_{(F\oplus s)}$ and $G_{(F\oplus s)}$. By replacing the sections of p in G_F^* that pass horizontally through s with the corresponding sections in $G^*_{(F\oplus s)}$, the path p' is obtained.

For the first condition of the lemma, simply note that no cell on the sections of the path that are altered by flipping the edge parities of s is an alternating cell in G_F or $G_{(F\oplus s)}$. The second condition follows from the construction of p'. □

Lemma 7.2 *Let G_F contain no Type III border cells. Let $A = (a_1, a_2, \cdots, a_n)$ be an alternating strip sequence in G_F, let $1 \leq k < i < n$, and let x and y be the vertices in $G_{(F\oplus A_{1,k})}$ that correspond to the cells on either side of the cell at the beginning of a_k. If x and y are not on any of the strips $a_{k+1}, a_{k+2}, \cdots, a_i$, then the alternating cell at the beginning of a_k in $G_{(F\oplus A_{1,i})}$ is not on the boundary created by a_i.*

Proof Let C be the cycle in $G^*_{(F\oplus A_{1,k-1})}$ that corresponds to the boundary on which a_k begins. By Lemma 6.8, since $k < n-1$, this boundary contains no Type III border cells, and by Lemma 7.1 there is a path between x and y in $G_{(F\oplus A_{1,k})}$, as pictured in Figure 7.3(a). No vertices along this path correspond to alternating cells, since no cells on the

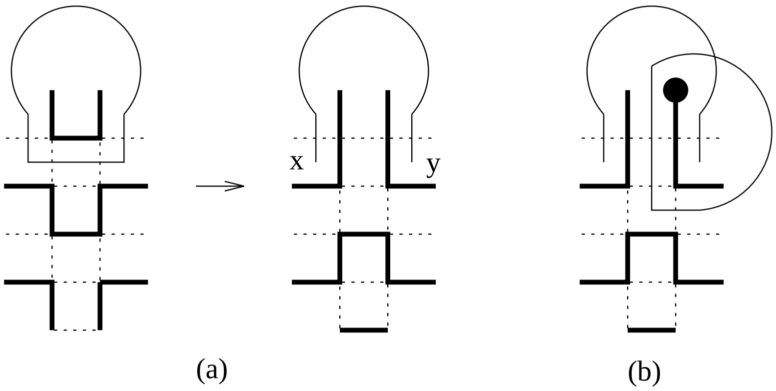


Figure 7.3: The boundary on which a_k begins in $G_{(F \oplus A_{1,k-1})}$ and $G_{(F \oplus A_{1,k})}$ (a), and the intersection with the boundary created by a_i in $G_{(F \oplus A_{1,i})}$ (b).

boundary were Type III border cells. In particular, no vertices along this path correspond to the alternating cells at the end of any of the strips $a_{k+1}, a_{k+2}, \cdots, a_i$, and, as required by the statement of the lemma, none of the strips $a_{k+1}, a_{k+2}, \cdots, a_i$ include either x or y. So by repeated application of Lemma 7.1, there exists a path between x and y in $G^*_{(F \oplus A_{1,i})}$ that includes no vertices that correspond to alternating cells in $G_{(F \oplus A_{1,i})}$.

Let C' be the cycle in $G^*_{(F \oplus A_{1,i})}$ that corresponds to the boundary created by a_i. Suppose C' includes the vertex that corresponds to the alternating cell at the beginning of a_k. The configuration of the dual with respect to $F \oplus A_{1,i}$ must then be as pictured in Figure 7.3(b). But this is impossible, since the vertex in the primal marked with the heavy dot cannot be part of a cycle, contradicting F being a 2-factor. □

7.3 Existence

Lemma 7.3 *Let G_F contain no Type III border cells. If there exists an alternating strip sequence $A = (a_1, a_2, \cdots, a_k)$ in G_F that begins on boundary B, then there exists a static alternating strip sequence A' in G_F that also begins on B with total area no greater than that of A.*

Proof The proof is by induction on the total area of A. If A consists of a single alternating strip, then by definition, it is static, and the lemma holds. Otherwise, by induction, there exists a static alternating strip sequence $S = (s_1, s_2, \cdots, s_n)$ that begins on the boundary created by a_1 in $G_{(F \oplus a_1)}$ with total area no greater than that of $A_{2,n}$.

Let s_i be the last strip in S that shares area or any edges with a_1. If there is no such strip, then $A' = (a_1, s_1, s_2, \cdots, s_n)$ is a static alternating strip sequence in G_F that satisfies the lemma. Otherwise, there are four cases to consider.

Case 1:

Figure 7.4(a) shows an even alternating strip G_F and its surrounding edges whose parity

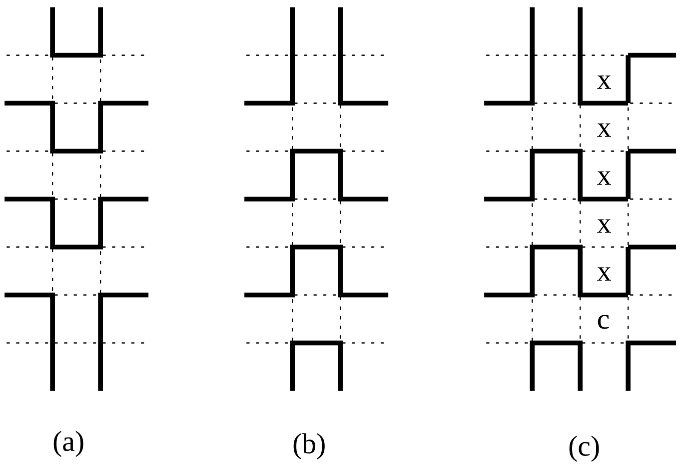


Figure 7.4: The even alternating strip a_1 in G_F (a), the same strip in $G_{(F\oplus a_1)}$ (b), and the required configuration in $G_{(F\oplus a_1)}$ if s_i shares a side edge with a_1 (c).

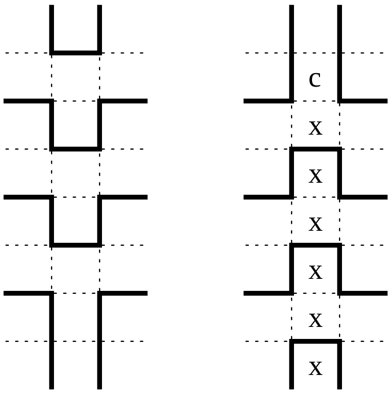
can be deduced from the degree constraints of the 2-factor. The bottommost and topmost vertical dark edges are not implied by the degree constraint – if either of the top ones were instead horizontal, then the border on which the strip begins would contain a Type III border cell; if either of the bottom ones were horizontal, then the border the strip creates would contain a Type III border cell. The alternating strip sequence would then be either (a_1) or (a_1, s_1), respectively, and it is easy to verify that both are static.

Figures 7.4(b) and (c) show the edge parities in $G_{(F\oplus a_1)}$. The marked cells in Figure 7.4(c) are those cells that might belong to s_i if s_i and a_1 share a "side" edge. Notice that if any one of these cells belongs to s_i, then all the marked cells below it must belong to s_i as well, including the bottom-most cell marked c. Cell c must therefore be in s_i, so let s_i' be the portion of s_i including and extending below c, and let s_i'' be the portion of s_i above c.

Since S is a (static) alternating strip sequence, for all $j \geq i$, s_{j+1} begins on the boundary created by s_j in $G_{(F\oplus a_1\oplus S_{1,j})}$. It is now argued that s_{i+1} begins on the boundary created by s_i' in $G_{(F\oplus a_1\oplus s_i')}$ and that for all $j > i$, s_{j+1} begins on the boundary created by s_j in $G_{(F\oplus a_1\oplus s_i'\oplus S_{i+1,j})}$. This will establish that the strips $s_1, s_2, \cdots, s_i, s_i''$ can be discarded from the sequence leaving a valid alternating strip sequence in which each successive strip begins on the boundary created by its predecessor.

Let x_1 and y_1 be the vertices in $G^*_{(F\oplus a_1\oplus S_{1,i})}$ that correspond to the cells on either side of the alternating cell at the end of s_i, and let y be the vertex that corresponds to the cell at the beginning of s_{i+1}.

Notice that in $G_{(F\oplus a_1\oplus S_{1,j})}$, the strips $s_i'', s_{i-1}, s_{i-2}, \cdots, s_1$ are all even alternating strips, and that there is a path in $G^*_{(F\oplus a_1\oplus S_{1,j})}$ between x_1 and x_2 including y, that corresponds to the boundary created by s_i. Also, note that cells that correspond to x_1, x_2, and y are not in any of the strips $s_i'', s_{i-1}, s_{i-2}, \cdots, s_1$, since S is a *static* alternating strip sequence. Further, by Lemma 7.2, the path between x_1 and x_2 does not include the vertex that corresponds to the alternating cell at the beginning of any of the strips $s_i'', s_{i-1}, s_{i-2}, \cdots, s_1$



(a) (b)

Figure 7.5: The even alternating strip a_1 in G_F (a), the same strip in $G_{(F\oplus a_1)}$ (b).

in $G_{(F\oplus a_1\oplus S_{1,j})}$. So by repeated application of Lemma 7.1, there exists a path between x_1 and x_2 that includes y in

$$G^*_{(F\oplus a_1\oplus S_{1,j}\oplus s_i''\oplus s_{i-1}\oplus s_{i-2}\oplus\cdots\oplus s_1)} = G^*_{(F\oplus a_1\oplus s_i'\oplus S_{i+1,j})}.$$

Therefore, $A' = (a_1, s_i', s_{i+1}, s_{i+2}, \cdots, s_n)$ is a static alternating strip sequence in G_F that begins on B and has smaller total area than A. Of course the same argument applies if a_i shares an edge on the left "side" of a_1, by symmetry.

Case 2:

Figure 7.5(a) again shows a_1 in G_F, and (b) shows its configuration in $G_{(F\oplus a_1)}$. The marked cells are those cells that might belong to s_i if s_i and a_1 share an edge at the bottom of a_1, or share area. If any of these cells belong to s_i, then all of the marked cells above it belong to s_i, and s_i ends at the top-most marked cell, c.

It is now argued that s_{i+1} begins on the boundary B in G_F (on which the original sequence A begins), and that s_{j+1} begins on the boundary created by s_j in $G_{(F\oplus S_{1,j})}$ for $j > i$. This will permit the deletion of the strips $a_1, s_1, s_2 \cdots, s_i$ from the sequence in the same manner as in the previous case.

As in the previous case, let x_1 and x_2 be the vertices that correspond to the cells to either side of c, and let y be the vertex that corresponds to the cell at the beginning of s_{i+1}. In the graph $G_{(F\oplus a_1\oplus S_{1,i})}$ there is a path p between x_1 and x_2 that includes y. Now, notice that the cells corresponding to x_1, x_2, and y are not on any of the strips $a_1, s_1, s_2, \cdots, s_i$, since S is static and s_i was chosen to be the *last* strip in S that shares edges or area with a_1. (If, for example, y was on a_1, then s_{i+1} would have been the last such strip.) Further, by Lemma 7.2, none of the vertices on p correspond to the alternating cell at the beginning of any of the strips $s_1, s_2, \cdots, s_i$, and in $G_{(F\oplus a_1\oplus S_{1,i})}$, the alternating cell at the beginning of a_1 in $G_{(F\oplus a_1)}$ is not alternating because s_i overlaps it.

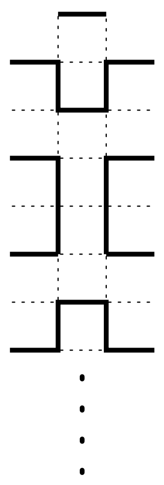


Figure 7.6: The even alternating strip a_1 in $G_{(F \oplus a_1)}$ and s_i above it sharing its top edge.

Then by repeated application of Lemma 7.1, there exists a path between x_1 and x_2 that includes y in

$$G^*_{(F \oplus a_1 \oplus S_{1,i} \oplus s_i \oplus s_{i-1} \oplus \cdots \oplus s_1 \oplus a_1)} = G^*_F.$$

Thus s_{i+1} (which begins at the cell corresponding to y) begins on B in G_F.

Now, let x_1 and x_2 be the vertices that correspond to the cells to either side of the alternating cell at the beginning of s_j, and let y be the vertex that corresponds to the cell at the beginning of s_{j+1}. In the graph $G_{(F \oplus a_1 \oplus S_{1,j})}$ there is a path p between x_1 and x_2 that includes y. As before, the cells corresponding to x_1, x_2, and y are not on any of the strips $a_1, s_1, s_2, \cdots, s_i$, since S is static and s_i was chosen to be the *last* strip in S that shares edges or area with a_1. Also, as before, by Lemma 7.2, none of the vertices on p correspond to the alternating cell at the beginning of any of the strips $s_1, s_2, \cdots, s_i$.

So by repeated application of Lemma 7.1, there exists a path between x_1 and x_2 that includes y in

$$G^*_{(F \oplus a_1 \oplus S_{1,j} \oplus s_i \oplus s_{i-1} \oplus \cdots \oplus s_1 \oplus a_1)} = G^*_{(F \oplus S_{i+1,j})}.$$

Thus s_{j+1} (which begins at the cell corresponding to y) begins on the boundary created by s_j in $G_{(F \oplus S_{i+1,j})}$ for all $j > i$, and so $A' = (s_{i+1}, s_{i+2}, \cdots, s_n)$ is a static alternating strip sequence in G_F.

The remaining cases to consider are those in which s_i includes one of the two alternating cells at the top of Figure 7.5(b) (the topmost one is not marked). First note that neither of these cells can be at the end of an odd alternating strip of length greater than one since the parity configuration does not match. It is possible, however, that s_i is the last strip in the sequence, and that it is in fact a single Type III border cell, or that s_i is an even alternating strip that ends at the topmost (unmarked) alternating cell in the figure. The latter possibility is considered first.

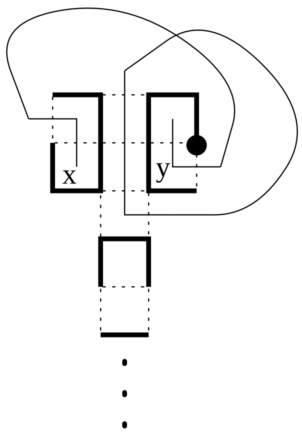


Figure 7.7: The argument for why the top two alternating cells of a_1 cannot become Type III cells in an alternating strip sequence.

Case 3:

Suppose the configuration pictured in Figure 7.6 exists. Labeling vertices in the same manner as in the previous case, there is a path p from x_1 to x_2 that includes y, the vertex corresponding to the cell at the beginning of s_{i+1} in $G^*_{(F\oplus a_1\oplus S_{1,i})}$. As with the previous case, none of the cells corresponding to x_1, x_2 or y are on any of the strips $a_1, s_1, s_2, \cdots s_i$, and p does not include any alternating cells, so by repeated application of Lemma 7.1, there exists a path between x_1 and x_2 that includes y in

$$G^*_{(F\oplus a_1,\oplus S_{1,i}\oplus s_i\oplus s_{i-1},\cdots,s_1,a_1)} = G_F.$$

Thus s_{i+1} begins on the boundary B in G_F, as in the previous case.

Also, as before, if x_1, x_2, and y are the vertices corresponding to the two cells on either side of the cell at the end of s_j and the beginning cell of s_{j+1}, then for $i < j$, in $G^*_{(F\oplus a_1\oplus S_{1,j})}$ there is a path p between x_1 and x_2 that includes y. None of x_1, x_2 and y are on any of the strips $s_i, s_{i-1}, \cdots, s_1, a_1$ by the same argument as in the previous case, and by Lemma 7.2, p includes none of the alternating cells at the beginning of any of the strips $s_i, s_{i-1}, \cdots, s_1$, and a_1 has no alternating cell in $G_{(F\oplus a_1\oplus S_{1,i})}$. Thus by repeated application of Lemma 7.1, there exists a path between x_1 and x_2 including y in

$$G^*_{(F\oplus a_1\oplus S_{1,j}\oplus s_i\oplus s_{i-1}\oplus\cdots\oplus s_1\oplus a_1)} = G^*_{(F\oplus S_{i+1,j})}.$$

So s_{j+1} begins on the boundary created by s_j in $G_{(F\oplus S_{i+1,j})}$. Therefore, $A' = (s_{i+1}, s_{i+2}, \cdots, s_n)$ is a static alternating strip sequence in G_F that begins on boundary B.

Case 4:

For the last case, suppose s_i is one of the single alternating cells in Figure 7.7, (so $i = n$).

Let p be the path in $G^*_{(F \oplus a_1)}$ between the vertices marked x and y in the figure, whose vertices corresponds to the cells on boundary B. This path contains no vertices that correspond to alternating cells, so in particular, it contains no vertices that correspond to the alternating cells at the ends of any of the strips $s_1, s_2 \cdots s_{n-1}$. Also, neither x nor y is on any of those strips, for if either was, then that strip would share an edge with s_i not permitted by the definition of static. So by repeated application of Lemma 7.1, there exists a path between x and y in $G^*_{(F \oplus a_1 \oplus S_{1,n-1})}$. However there also must be a path corresponding to the boundary created by s_{n-1}, and by assumption that boundary includes the vertex corresponding to s_n. The configuration shown in Figure 7.7 must then arise, and this leads to a contradiction, since the dotted primal vertex cannot be in a 2-factor. □

By Theorem 6.1, an alternating cell sequence exists in G_F if G is Hamiltonian, so by this lemma, a static alternating cell sequence exists in G_F if G is Hamiltonian. This permits the development of the algorithm in the next chapter for finding Hamiltonian cycles in grid graphs without holes.

Chapter 8

The Algorithm

In this chapter, the algorithm to find Hamiltonian cycles in grid graphs without holes is presented. The algorithm achieves the main goals of this thesis: it is provably correct and runs in polynomial time. It is not, however, the *most* efficient approach. Several optimizations are immediately evident (such as dynamically updating information about the graph), but are not presented in the interest of simplicity and clarity. The main part of the algorithm is the procedure described in the next section to identify static alternating strip sequences.

8.1 Finding Alternating Strip Sequences

The procedure works by building a graph H from the initial graph G and a 2-factor F of G, in which each path represents a portion of an alternating strip sequence. This graph is then searched for a path representing a complete alternating strip sequence, and it is shown that the minimum length path of this type must be a static alternating strip sequence.

8.1.1 Locating Type III Border Cells

The first step of the procedure is to label the connected components of the 2-factor F, and immediately search the graph for a Type III border cell. (This type of cell is identifiable from its parity configuration and the component labels of its vertices). If a Type III border cell is located, then this constitutes a (static) alternating strip sequence, and the procedure completes.

8.1.2 Locating Alternating Strips

This portion of the procedure identifies all subgraphs of G_F that are *structurally* identical to an alternating strip; that is, they have the required parity configuration. This will result in the addition of vertices to H with the following labels: **BEGIN**, **ODD**, **EVEN**, and **CHAIN**.

Candidate Strips

For each cell c with the parity configuration of a Type I cell (exactly one of four edges in F), a search is made of the cells in a straight line from the edge of c opposite the edge in F to the outside of the graph. Let c' be the first cell in this search that is alternating. The vertices c and c' and the edge (c, c') are then added to H and labeled as follows: c is labeled **BEGIN**, and c' is labeled **ODD** or **EVEN**, depending on its distance (along the search path) from c. The edge (c, c') is weighted with this distance. If no alternating cell is encountered in the search, then no vertices or edges are added to H.

Also at this step, for each alternating cell c, vertices c and c' and an edge (c, c') are added to G labeled and weighted as follows: c is labeled **BEGIN**, c' is labeled **ODD**, and the edge (c, c') is given weight one. These represent the odd alternating strips of length one.

Candidate Chain Strips

Because of the shared edge allowed in the definition of static alternating strip sequences, "chain" strips must also be considered. These are the strips that share the specified edge with the previous strip in the sequence, and so the parity configuration of the cell at the beginning of these strips is not identical to a Type I cell.

For each cell c with the parity configuration of a Type II cell (exactly two adjacent edges in F), that shares one (or both) of its edges in F with a neighboring alternating cell c'', a search is made of the cells in a straight line from the edge of c adjacent to the edge shared with c'' to the outside of the graph. As before, let c' be the first cell in this search that is alternating. The vertices c and c' and the edge (c, c') are then added to H and labeled as follows: c is labeled **CHAIN** and c' is labeled **ODD** or **EVEN**, depending on its distance (along the search path) from c, and c'' is unlabeled. The edge (c, c') is weighted as before with the distance from c to c'. If no alternating cell is encountered in the search, then no vertices or edges are added to H. If both of the edges of c in F are shared with a neighboring cell, then this procedure is repeated with c'' equal to the other neighboring cell.

Next, for each cell c with three edges in F that shares the dark edge that is opposite its light edge with a neighboring alternating cell c'', the vertices c and c' and the edge (c, c') are added to H and labeled as follows: c is labeled **CHAIN** and c' is labeled **ODD**. The edge (c, c') has weight one. This last step adds those candidate odd alternating chain strips of length one.

Linking the Strips Together

The final stage in the construction of H requires an auxiliary graph, D. The graph D is the dual of G with a vertex for the outer face. All edges of D that cross a dark edge in F are discarded, and the all-pairs-shortest-paths are computed for D.

For each vertex c in H labeled **EVEN**, let c_1 and c_2 be the two vertices in D that correspond to the two cells that share dark edges with the cell in G_F that corresponds to c. For every vertex v labeled **BEGIN** in H, if v lies on the shortest path between c_1 and c_2, then the directed edge (c, v) is added to H with weight zero. For every vertex v labeled **CHAIN** in H, if there exists a path between c_1 and c_2, and v corresponds to the same cell in G as either c_1 or c_2, then the directed edge (c, v) is added to H with weight zero.

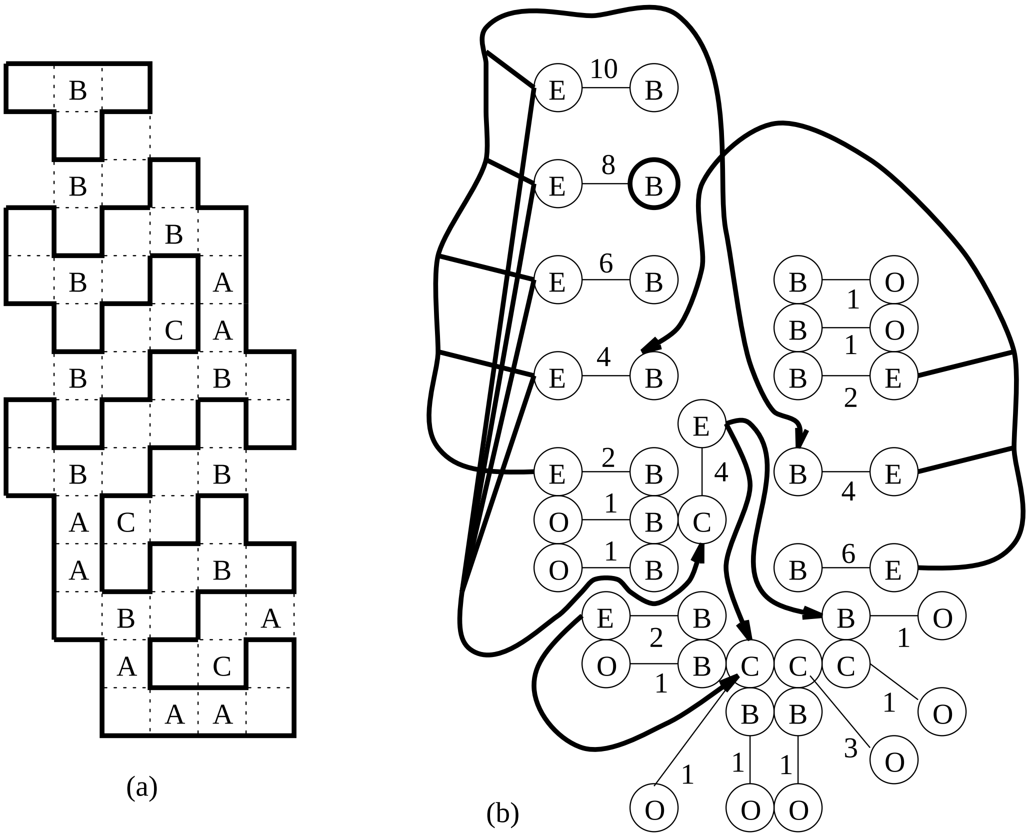


Figure 8.1: The graph H (b) for the 2-factor shown in (a). See the text for a full description of the figure. The "B" and "C" vertices are in the same relative position in (b) as in (a).

The directed edges are required to ensure that multiple consecutive link edges are cannot appear in a path in H. All link edges are directed from the end of one strip to the beginning of another, so no link edges are directed into the end of a strip or out of the the beginning of a strip. They therefore cannot be "chained" together in a manner that might allow a sequence of strips to be located that is not an alternating strip sequence.

Figure 8.1 shows the graph H that has just been described. In (a), the cells of G_F have been marked "B" for **BEGIN**, "C" for **CHAIN**, and "A" for alternating cells. The same convention is used in labeling vertices in H (b), except that the alternating cells have been relabeled "O" for **ODD** and "E" for **EVEN**. The dark edges represent the edges added during the "linking" phase (all have weight zero), while the light edges represent the edges added prior to the linking phase, and their weights are shown.

8.1.3 Shortest Paths in H

Once H is constructed, the shortest path for all pairs of vertices in H is computed. For each vertex b marked **BEGIN** that corresponds to a border cell in F, and each vertex e marked **ODD**, the shortest path between b and e corresponds to a static alternating strip sequence. If no paths exists between any pair b and e, then no alternating strip sequence exists in G_F.

In the example in Figure 8.1(b), the heavily circled "B" vertex is the only potential beginning vertex for the paths described in this section. Notice that there are four paths from that vertex to an "O" vertex, two each with weights 13 and 21. One of the shorter ones corresponds to the static alternating strip sequence shown in Figure 7.1(b).

8.1.4 Correctness

It should be fairly obvious that the initial phase of this procedure, in which candidate alternating strips (including "chain" strips) are identified, correctly identifies all subgraphs that are structurally identical to alternating strips, and therefore potential elements of an alternating strip sequence. Note that the parities of the cells along the search in a straight line from the beginning cell in this step are determined by the degree constraint, so it is not necessary to verify that they in fact extend the alternating strip.

There are two steps involved in showing that the sequence of alternating strips obtained from the shortest path is in fact a *static* alternating strip sequence. First, it is shown that the strips identified in the shortest path share edges only in the manner allowed by the definition of "static." A sequence of strips for which this is true is said to "satisfy the static property." Then, it is argued that successive strips in the sequence begin on the boundary created by their predecessor at the point in the sequence that their edge parities are flipped.

Static Property

Suppose $A = (a_1, a_2, \cdots, a_n)$ is a sequence of alternating strips obtained using the shortest path method, and suppose it does not satisfy the static property. Let a_i and a_j be strips that violate this property, with $i < j \leq n$.

As in the proof of Lemma 7.3, there are four cases to consider, pictured in Figure 8.2.

In (a), if any of the marked cells except for the bottommost one are in a_j, then the topmost marked cell is in a_j, and that cell corresponds to a **BEGIN** vertex in H that lies on the same path in the dual as the **BEGIN** vertex that corresponds to the cell on which a_i begins. A shorter path can then be obtained by following the link from the end of strip a_{i-1} directly to this **BEGIN** cell. If the bottommost marked cell is in a_j, then a_j might be directed down, in which case it shares only the allowed edge with a_i. Otherwise, a_j is directed up, and it must include the top-most marked cell, so the previous argument applies.

In (b), if any of the cells marked with an "x" are in a_j, then the bottommost cell marked "c" is in a_j, and it corresponds to an **EVEN** vertex in H, as does the same cell of a_i. All **EVEN** vertices in H that coincide in G have identical "link" edges, so a shorter path may be obtained by proceeding directly from the **EVEN** vertex at the end of a_i to the vertex in the path *after* the **EVEN** vertex at the end of a_j.

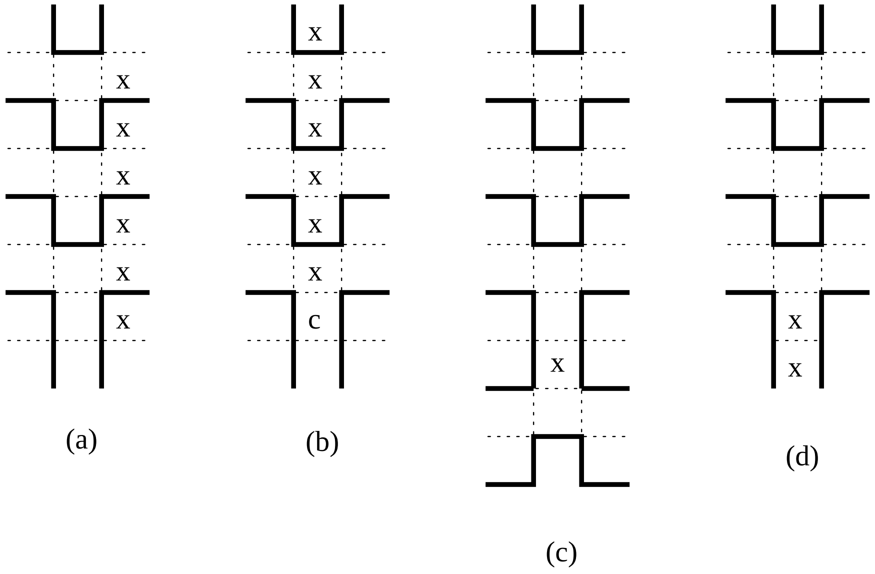


Figure 8.2: The possible ways in which a two strips in the strip sequence identified by the shortest path method might violate the static property.

In (c), a_j is an even alternating strip that shares the bottom edge of a_i. The marked cell in the figure is the alternating cell at the end of a_j. Figure 8.3 shows the possible configurations of the dual edges near this cell. Let p be the shortest path in the dual between the vertices that correspond to the cells on either side of the alternating cell at the end of a_i, and let p' be the shortest path in the dual between the vertices that correspond to the cells on either side of the alternating cell at the end of a_j. Notice that in all of the configurations, both p and p' include the shortest path between the vertices labeled v and w. The **BEGIN** vertex that is linked to the **EVEN** vertex that corresponds to the cell at the end of a_j lies on p', and so must also lie on p. Thus a shorter path can be obtained by following the link from a_i directly to a_{j+1}.

For the last case, suppose either of the marked cells in (d) are in a_j and a_j consists of *only* the single cell (i.e. a_j is a Type III border cell that ends the sequence). This case leads to a contradiction in a manner similar to the proof of Lemma 7.2. The impossibility of this case is stated as a lemma because it is useful for the final step in the proof of correctness.

Lemma 8.1 *Let $A = (a_1, a_2 \cdots, a_n)$ be a sequence of alternating strips obtained via the shortest path method. Let x and y be the two vertices in G_F^* that correspond to the cells on either side of the alternating cell c at the end of a_j. Then the path between x and y does not contain the vertex corresponding to the alternating cell at the end of a_i, for $1 \leq i < j < n$.*

Proof First, note that the alternating cell at the end of a_i must not be a Type III border cell, so its dark edges must belong to the same component. Suppose that the path between x and y in $G^*_{(F \oplus A_{1,j})}$ includes the vertex corresponding to the alternating cell at the end

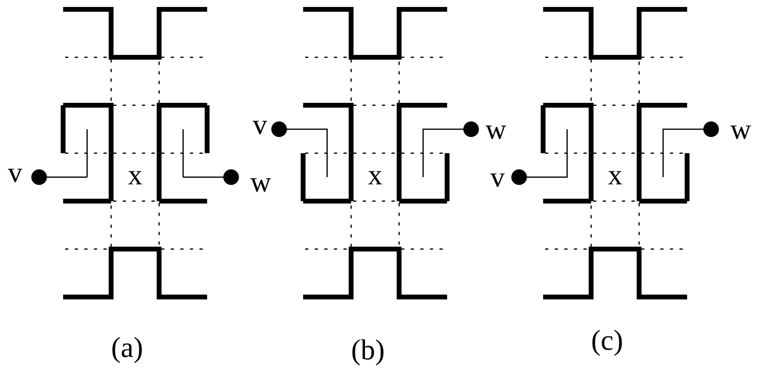


Figure 8.3: The dual edges surrounding the end cells of two even alternating strips that share their bottom edge.

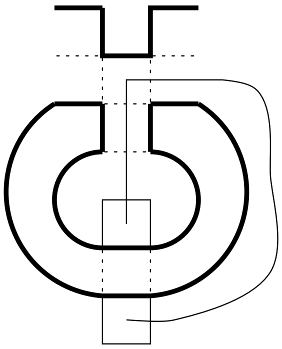


Figure 8.4: The configuration of the path between the neighbor cells of the alternating cell at the end of a_j that passes through the alternating cell at the end of a_i.

of a_i. Since c is an alternating cell between the cells that correspond to x and y, the path must be in the configuration pictured in Figure 8.4. But then c is an alternating cell on the boundary created by a_i, so the sequence should end at a_{i+1}, but $i+1 < n$, which is a contradiction. □

For the case pictured in Figure 8.2(d) to arise, the path between the two vertices that correspond to the cells on either side of the alternating cell at the end of a_{j-1} must include the alternating cell at the end of a_i, which violates Lemma 8.1

Boundaries

Finally, it is necessary to verify that if a_i is followed by a_{i+1} in some shortest path obtained by the algorithm, then a_{i+1} begins on the boundary created by a_i in $G_{(F \oplus A_{1,i})}$.

Lemma 8.2 *Let $A = (a_1, a_2, \cdots, a_n)$ be a sequence of alternating strips obtained via the shortest path method that satisfies the static property, and let x_1 and x_2 be the two vertices in G_F^* that correspond to the two cells in G_F that share a dark edge with the alternating cell at the end of a_i, and let y be the vertex in G_F^* that corresponds to the cell at the beginning of a_{i+1}. If there exists a path p in G_F^* between x_1 and x_2 that includes y, then there exists a path in $G^*_{(F \oplus A_{1,i})}$ between x_1 and x_2 that includes y, for $1 \leq i < n$.*

Proof Note that x_1, y, and x_2 do not correspond to any cells that are in any of the strips $a_1, a_2, \cdots, a_i$, since A satisfies the static property, and the only strip permitted to include one of the cells x_1 or x_2 is a_{i+1}. By Lemma 8.1, p does not include the alternating cell at the end of any of the strips $a_1, a_2, \cdots, a_i$. So by repeated application of Lemma 7.1, there exists a path in $G^*_{(F \oplus a_1 \oplus a_2 \oplus \cdots \oplus a_i)}$ between x_1 and x_2 that includes y. □

The algorithm then correctly identifies a static alternating strip sequence if one exists because:

1. No strips share area or an edge except for the permitted edge, so flipping the edge parities of earlier strips in the sequence does not alter the edge parities of strips that are encountered later in the sequence. If it did, then those later "strips" would not be alternating strips at the point in the sequence that they were flipped.

2. At the point that the edge parities of strip a_i are flipped, it begins on the boundary created by a_{i-1}, since the path between the cells on the sides of the alternating cell at the end of a_{i-1} that passes through the cell at the beginning of a_i is preserved throughout flipping the prior strips in the sequence.

8.2 Finding Hamiltonian Cycles

The algorithm for finding Hamiltonian cycles is listed in Figure 8.5. As noted, the main complexity is in the Find_Static_Alternating_Strip_Sequence procedure just described.

8.3 Time Complexity

This section gives a brief analysis of the running time of the algorithm. It is concerned primarily with showing that the algorithm runs in polynomial time, and not with giving the tightest bound possible. Unless stated otherwise, V and E refer to the vertex and edge sets of the input graph, G.

The standard algorithm for labeling connected components runs in $O(|V| + |E|)$ time, which is equivalent to $O(|V|)$ since the degree of a grid graph is bounded. The initial scan of the cells of the graph to identify possible Type III border cells, **BEGIN** cells, **CHAIN** cells, and alternating cells that may become **ODD** or **EVEN**, examines each cell once, so it takes $O(|V|)$ steps.

For each **BEGIN** and **CHAIN** cell, the search for an end cell (**ODD** or **EVEN**) might involve examining $O(|V|)$ cells, and there can be no more that $O(|V|)$ individual searches, so this phase takes $O(|V|^2)$ time, and it adds $O(|V|)$ vertices to H.

```
algorithm Hamiltonian_Cycle_in_GGWH(G)
Pre: G is a grid graph without holes
Post: F is a Hamiltonian cycle in G, or else G is not Hamiltonian

var F : Graph
    S : Alternating Strip Sequence

F = Find_2-Factor(G)
if F = NULL then
   Report("No Hamiltonian Cycle")
   EXIT
while more than one component in F do
        Label_Connected_Components(F)
        S = Find_Static_Alternating_Strip_Sequence(G, F)
        if S = NULL then
           Report("No Hamiltonian Cycle")
           EXIT
        else
           F = Apply_Sequence(F, S)
end while
```

Figure 8.5: The algorithm for finding a Hamiltonian cycle in a grid graph without holes.

Identifying the dual graph and computing all-pairs-shortest-paths requires $O(|V|^3)$ time using the standard algorithm. Examining every pair of vertices in H for the potential addition of an edge in the linking phase and examining the vertices along the shortest path between takes $O(|V(H)|^3)$ time, and since $|V(H)| \in O(|V|)$, this is equivalent to $O(|V|^3)$. Similarly, identifying the shortest path in H takes $O(|V|^3)$ time, using the standard algorithm.

Thus the entire Find_Static_Alternating_Strip_Sequence procedure is dominated by the shortest path portion, and runs in $O(|V|^3)$ time. There are fewer than $|V|$ components in any 2-factor, so by Theorem 6.1, this procedure is repeated no more than $|V|$ times, for a total running time of the main loop of $O(|V|^4)$. From Section 2.2.2, the initial 2-factor can be identified in $O(|V||E|)$ which is equivalent to $O(|V|^2)$, since the degree of a grid graph is bounded, so the entire algorithm runs in $O(|V|^4)$ time.

Chapter 9

Conclusion

9.1 Summary of Results

The main result of this thesis is the polynomial algorithm presented in the previous chapter. The discovery of this algorithm proves that the Hamiltonian cycle problem for grid graphs without holes is in the class $\mathcal{P}$ of tractable problems. In addition, some of the results concerning the structure of grid graphs without holes are significant in their own right. In particular:

- The alternating cell structure of 2-factors in grid graphs without holes (Theorem 4.1) proved to be a powerful tool in reasoning about transformations of the 2-factor.
- The oriented dependency graph defined in Section 4.2 is a useful method of encoding this structure, and possesses some nice graph-theoretic properties.
- The index of a 2-factor defined in Section 6.1 may be useful as a non-obvious measure of progress for other inductive proofs regarding transformations of a 2-factor.

9.2 Future Work

There are three main directions for future work related to this problem. I list them roughly in order of increasing significance (and difficulty):

- As noted in the previous chapter, the algorithm as presented here can be made much more efficient with some modification. Dynamic updating of relevant information about the graph (i.e. component labels, cell configurations, and dual edge parities) as each alternating strip sequence is identified and applied would improve performance. Eliminating certain "duplicate" vertices (i.e. vertices that represent the same cell in the grid graph) in the derived graph H would also result in an improvement. Finally, if the all-pairs-shortest-path procedures could be modified to take advantage of additional information about the respective graphs on which they operate, it may be that the limiting phase of the entire algorithm is the identification of the initial 2-factor.

- The proofs in Chapters 5 through 7 are, for the most part, not elegant; I am hopeful that a more abstract treatment of these structures, perhaps based on the oriented dependency graph, will result in more concise proofs.
- There are potentially some cases in which this algorithm or a variant could be made to work on grid graphs *with* holes. For example, in cases where the directions on edges in the oriented dependency graph prevent any of the alternating cycles in the symmetric difference of two 2-factors from entering a particular region of the graph, holes might be allowed in that region.

9.3 Acknowledgements

I am grateful to my advisor, Bill Lenhart, for his advice, encouragement, and suggestions throughout this project. I would also like to thank my second reader, Andrea Danyluk, for her thoughtful comments on the draft of this thesis.

Appendix A

Some Notes on Bridgeman's Work

Many of the results of this thesis bear on Bridgeman's work in [1]; a number of the most important implications on her work are gathered in this appendix.

A.1 Sufficiency of Sliding

In the language of this thesis, a slide is an operation that transforms a 2-factor so that it admits a Type III border cell flip ("glue"), which results in a reduction in the number of components in the 2-factor. A slide may also enable another slide, but each sequence of slides is expected to terminate in a glue. A single slide consists of flipping an alternating cell that breaks some cycle in the 2-factor followed immediately by flipping a second alternating cell that glues the two cycles just created. In this way, each slide is a neutral operation.

The question raised in Bridgeman's work is whether the sliding operation is sufficient to transform any Hamiltonian cycle on G into any other Hamiltonian cycle on G. That is, given H_1 and H_2, both Hamiltonian cycles in G, is there some sequence of slides that transforms H_1 into H_2. A positive result would have been crucial for Bridgeman's method of proving the correctness of her algorithm. Unfortunately, the conjecture is false.

Observation A.1 *Sliding is not sufficient to transform any Hamiltonian cycle on G into any other Hamiltonian cycle on G.*

Proof Figure A.1 shows a simple counter-example to the "sufficiency of sliding" conjecture. The diagram in (a) is the initial Hamiltonian cycle, and the one in (b) is the "target" Hamiltonian cycle. The only alternating cells in (a) are the ones marked with X's, and it is easy to see that flipping either of those creates two nested cycles with the property that no single alternating cell flip will merge them. Thus no slides are possible in (a), so clearly no sequence of slides will transform that cycle into the one pictured in (b). □

Despite the fact that sliding is not sufficient to transform one Hamiltonian cycle into another, sliding *is* sufficient to transform a 2-factor in the intended way to set up a glue. This

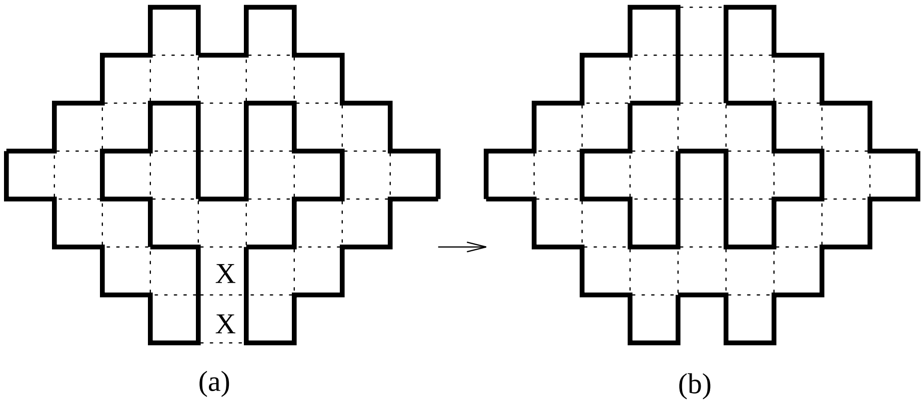


Figure A.1: A counterexample to the sufficiency of sliding conjecture.

is a direct implication of the existence of static alternating strip sequences in Hamiltonian grid graphs without holes. A static alternating strip sequence can be viewed as a sequence of slides culminating in a glue. Along each alternating strip starting at the end of the sequence, the two cells at the end of the strip constitute a slide, as do the pair above them after that, and so on, up to the top of the strip. The cell at the beginning of the strip that ends the static alternating strip sequence and the cell at the end of the previous strip constitute the next slide, and then each pair of cells along that strip are a slide, and so on. The cell at the beginning of the first alternating strip is a glue, so each static alternating strip sequence can be thought of as a sequence of slides followed by a glue, which is the central "move" of Bridgeman's algorithm.

This implies that Bridgeman's approach is correct; however to formalize this would require a verification that the (rather complicated) search strategy she employs for locating each successive slide is also correct, which is beyond the scope of this thesis.

A.2 Zipper and Combination Boundaries

Bridgeman classifies boundaries as "zipper," "stairstep," and "combination" depending on whether they consist of all Type II border cells, all Type I border cells, or a mixture of Type I and Type II border cells, respectively. Her approach to proving the correctness of her algorithm relied on showing that if merging (sliding and gluing) terminates with a 2-factor that contains any of these boundaries, then the graph is not Hamiltonian. Bridgeman resolved the zipper case, showing that any 2-factor that contains a zipper boundary is not Hamiltonian. Lemma 5.2 restates this fact with a different proof. The stairstep and combination cases were left as conjectures, but the work in this thesis resolves these; the proof that these conjectures are true follows from the existence of reduced alternating strip sequences.

In [1], an **alternate** edge is the edge opposite the dark edge in a Type II border cell. Bridgeman conjectured that if a 2-factor F in G contains exactly two components that meet along a zipper or combination boundary, and no cycle that includes an alternate edge exists on either component, then G is not Hamiltonian. If G *is* Hamiltonian, then by Lemma 6.7, an alternating strip sequence A exists that begins on the boundary in question. The first cell of the first alternating strip, c, is a Type III border cell in $G_{(F \oplus A \oplus c)}$ and so a cycle that includes an alternate edge exists. Thus by the contrapositive, the conjecture is true.

A.3 Notes on the I/O Structure

Bridgeman uses a structure called the "odd-parity union", or the "I/O structure" in her investigation of the possible sufficiency of sliding. This structure classifies the parity of dual vertices of G_F as odd or even depending on how many cycles of F contain them. Given two Hamiltonian cycles in G, H_1 and H_2, the dual vertices of G_{H_1} are labeled with an "I" if they become odd parity in the transformation to G_{H_2} and an "O" if they become even parity in the transformation to G_{H_2}. Vertices whose parity does not change are unlabeled. The approach was then to show that the I's and O's could be paired so that each pair constituted a slide. This led to a number of lemmas concerning the structure of the odd-parity union.

By Observation A.1, sliding is not sufficient, but the counterexample itself would appear to contradict a number of the lemmas stated in [1]. This is due to an error in Lemma 4.2 concerning the possible arrangements of I and O vertices around a dual vertex, which propagates to Lemma 4.7, 4.9 and 4.12, the final lemma being the one contradicted by the counterexample.

Bibliography

[1] S. Bridgeman. Finding hamiltonian cycles in grid graphs without holes. Honors thesis, Williams College, 1995.

[2] D. Briggs. Hamiltonian cycles in grid graphs without holes. Honors thesis, Williams College, 1994.

[3] H. Everett. Hamiltonian paths in non-rectangular grid graphs. Master's thesis, University of Saskatchewan, 1986.

[4] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal of Computing*, 11(4):676–686, 1982.

[5] David S. Johnson. The np-completeness column: an ongoing guide. *Journal of Algorithms*, 6:434–451, 1985.

[6] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem.* John Wiley & Sons, 1985.

[7] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem*, chapter 11. John Wiley & Sons, 1985.

[8] László Lovász. *Combinatorial Problems and Excercises*, page 333. North Holland Publishing Company, 1979.

[9] James A. McHugh. *Algorithmic Graph Theory.* Prentice-Hall, Inc., 1990.