On the number of cycles and perfect matchings in planar graphs

Xiao, Q.

Award date:
2016

Technische Universiteit Eindhoven University of Technology

Department of Mathematics and Computer Science
Algorithms Group

On the Number of Cycles and Perfect Matchings in Planar Graphs

Master's Thesis

Qi Xiao

Supervisor:

dr. Kevin Buchin

Committee Members:

dr. Anne Driemel

prof. dr. Nikhil Bansal

Eindhoven, July 2016

Abstract

We investigate the number of simple cycles, Hamiltonian cycles and perfect matchings in planar graphs with n vertices.

By analyzing the way simple paths can be extended several steps at a time using a face-coloring technique, we show that there can be at most $O(2.870214^n)$ simple cycles in planar graphs. We look into a result claimed in a previous work that there are at most $O(2.2134^n)$ Hamiltonian cycles in planar graphs, and argue that the proof given there is likely flawed. Using the transfer matrix technique, we show that there is a family of planar graphs with $\Omega(1.535365^n)$ perfect matchings.

Preface

This thesis is the fruit of six months' work at the Eindhoven University of Technology, under the supervision of dr. Kevin Buchin. It is to the best of my knowledge original, except where references to previous work are made.

I owe my gratitude towards my supervisor, dr. Kevin Buchin, for providing me with the fascinating topic in the first place and more importantly, the continuous help throughout the project. Every time I was stuck hypothesizing, proving or calculating, Kevin offered me fresh ideas and pointed out new directions. Your insights and inspirations have been indispensable for this thesis.

I would also like to thank the whole algorithms group for inspiring me to do a thesis in algorithms, in particular prof. dr. Mark de Berg, dr. Herman Haverkort, dr. Kevin Buchin, and dr. Bart M. P. Jansen. The algorithm courses they gave have been some of the most engaging and unforgettable experiences in my life. I would like to thank prof. dr. Jan Friso Groote and prof. dr. Hans Zantema as well, for introducing me to the world of theoretical research. You and your teachings will always inspire me.

Finally, I would like to thank my family, fellow students and friends for their tolerance and encourage towards me. I am lucky to have you in my life.

Quite some pain and joy have been experienced to produce this work. I hope you will find the topic a worthy one and enjoy reading my thesis.

Qi Xiao

Eindhoven, July 2016

Contents

Chapter 1

Introduction

This thesis is primarily concerned with the following question:

How many simple cycles can there be in a planar graph with n vertices?

Simple cycles in planar graphs arise naturally in real-world applications. As an illustrative example, suppose that you want to come up with a good jogging route in your neighborhood [7]. You start from home, go along some roads, and arrive back home. To make the jogging experience more interesting, you do not want to go to the same place or follow the same section of road twice. The roads can be modeled by planar graphs[1], and a jogging route is precisely a simple cycle in the planar graph. Knowing the number of simple cycles in planar graphs tells you how many possible jogging routes there are to choose from. More importantly from the perspective of algorithm design, if you are to come up with an algorithm that finds the best jogging route (according to some criteria) by enumerating all jogging routes, the total number of simple cycles in planar graphs helps you derive a bound on the running time of the algorithm.

Besides the example of jogging routes, simple cycles also arise in problems such as image classification [8][9], routing of electronic circuits and many others. For these problems, the number of simple cycles in planar graphs is useful for similar reasons: solving such problems requires finding cycles matching some specific criteria in a planar graph, the complexity of which is bounded above by the total number of cycles in the graph.

In this thesis, we also consider the number of Hamilotonian cycles and perfect matchings in planar graphs, both closely related to simple cycles and have their own interesting applications. Formally, if $\mathcal{G}_n$ is the set of all planar graphs with n vertices, we define

$$\begin{aligned} C_s(n) &= \max_{G \in \mathcal{G}_n} (\text{number of simple cycles in } G) \\ C_h(n) &= \max_{G \in \mathcal{G}_n} (\text{number of Hamiltonian cycles in } G) \\ C_p(n) &= \max_{G \in \mathcal{G}_n} (\text{number of perfect matchings in } G) \end{aligned}$$

Instead of trying to find precise formulas for C_s, C_h, C_p, which seems out of reach, we study their asymptotic behaviors. More specifically, since all of these functions are known to grow exponentially in n, we try to bound their exponential growth rates, namely to find $c, d \in \mathbb{R}$ such that $c^n \leq C_s(n) \leq d^n$ for large enough n, and analogously for $C_h(n)$ and $C_p(n)$.

These bounds and related problems have attracted considerable research interests in the past. Table 1.1 shows the (lower and upper) bounds obtained for $C_s(n)$, $C_h(n)$ and $C_p(n)$ over the years.

[1] Ignoring tunnels and bridges.

Year	$C_s(n)$	$C_h(n)$	$C_p(n)$	Obtained in
1999	$\Omega(2.259^n)$, $O(3.364^n)$			Alt et al. [1]
2007	$\Omega(2.4262^n)$, $O(2.8927^n)$	$\Omega(2.0845^n)$, $O(2.3404^n)$	$O(1.5651^n)$	Buchin et al. [4]
2012		$O(2.2134^n)$ (questionable)		Biswas et al. [3]
2014			$\Omega(1.535^n)$	Biro [2]
2016	$O(2.870214^n)$		$\Omega(1.535365^n)$	This thesis

Table 1.1: Bounds on $C_s(n)$, $C_h(n)$ and $C_p(n)$

Results in this thesis are closely related to the latest developments in the bounds of $C_s(n)$, $C_h(n)$ and $C_p(n)$.

In 2007, Kevin Buchin et al. obtained the latest results on the number of simple cycles and Hamiltonian cycles, bounding the number of simple cycles by $2.4262^n < C_s(n) < 2.8927^n$ and the number of Hamiltonian cycles by $2.0845^n < C_h(n) < 2.3404^n$ [4]. In the discussion section of their paper, they point out a possibility for improving the upper bound of $C_s(n)$. In Chapter 2, we follow the proposed approach and reach the improved result of $C_s(n) = O(2.870214)^n$.

In 2012, Sudip Biswas et al. aimed to improve the upper bound of the number of Hamiltonian cycles to $C_h(n) < 2.2134^n$ [3]. In Chapter 3, we argue that their reasoning is likely flawed.

In 2014, Michael Biro obtained the latest result on the lower bound of the number of perfect matchings, $1.535^n < C_p(n)$. In Chapter 4, we improve this bound by applying the transfer matrix technique on a twisted cylinder, to $1.535365^n < C_p(n)$. This is an adaption of the technique used in [4] to obtain the lower bound of the number of simple cycles, and its possibility has also been mentioned in the discussion section of the same paper.

The general domain our problem falls into – counting graph-related structures – has attracted a great deal of research interest as well. For instance, given a set of n points, one can ask how many possible simple polygons consist of these points [6]. Given a planar graph, one can ask how many spanning trees it has [5]. These are just two examples of the broad area of "counting" problems.

Chapter 2

Number of Simple Cycles

In this chapter we improve the upper bound on the number of simple cycles in planar graphs. The word "cycle" in this chapter always refers to simple cycles, unless stated otherwise.

In Section 2.1, we introduce the framework of the analysis, an adaption of that employed in [4]. The framework associates the number of cycles to the size of a tree. In Section 2.2, we propose a restriction (the *restricted-child rule*) on a specific class of nodes of the tree, "2-free" nodes. We establish its validity by showing that for a tree rooted at a 2-free nodes, either the rule applies, or the tree is dominated by another tree where the rule applies. Using this rule, we study the shape of the tree in Section 2.3. Finally in Section 2.4, we derive an upper on the number of simple cycles.

2.1 Preliminaries

We start with the following observation[1]:

Observation 2.1. *A cycle in a planar graph always partitions the faces of the graph into two connected regions: an "inside" region R_i and an "outside" region R_o.*

See Figure 2.1a for an example.

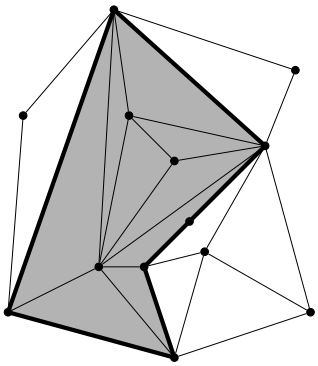
A *cycle-path* is a subpath of a cycle, and by giving the cycle-path an orientation, we obtain an *oriented cycle-path*. In Figure 2.1b, we show an oriented cycle-path of the cycle in Figure 2.1a. All faces on its left side are in one region (in this case R_i), while those on its right side are in another (in this case R_o). A face is on the left of the oriented cycle-path, if it is either incident to an oriented edge in the path and lies on its left, or "between"[2] two faces on the left. The definition for "right" is symmetric. We give the faces on the left a certain color (the "L"-color) and faces on the right another (the "R"-color). If in a coloring of faces, the L-colored faces and R-colored faces do not intersect, we say that the coloring is *legal*. Then the observation can be formulated as follows:

Observation 2.2. *If an oriented path P is an oriented cycle-path, then P induces a legal coloring of faces.*

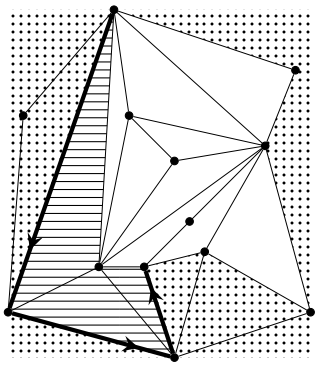
In the drawings in this thesis, we use horizontal lines to represent the L-color and dots to represent the R-color. If an oriented path P induces a legal coloring of faces, we say that it is a *potential oriented cycle-path*.

[1] This observation was exploited in [1] to derive the first upper bound on the number of cycles that we know of.

[2] The concept of "between" here should be thought as follows: If we cast a ray into a face from a vertex v on the path and rotates the ray into another face without hitting the path itself, then the faces incident to v and hit by the ray are between these two faces



(a) The shaded region is the "inside" and the unshaded is the "outside".



(b) Faces on the left of an oriented cycle-path are all inside, while those on the right are all outside.

Figure 2.1: A cycle partitions the graph into an two connected regions.

The number of simple cycles in a planar graph G does not decrease if we fully triangulate G. Thus we restrict our attention to fully triangulated planar graphs.

Now let us arrange potential oriented cycle-paths in a tree. The root is a single directed edge. The children of a potential oriented cycle-path are those oriented paths that can be obtained by extending it by one edge and still induce a legal coloring. We identify a potential oriented cycle-path by all its vertices: $u_1 \to \cdots \to u_l$. By abuse of notation, if P is an oriented path $p_1 \to \cdots p_m$, we abbreviate $p_0 \to \cdots p_m \to q_1 \to \cdots q_n$ to $P \to q_1 \to \cdots q_n$. The term "node" in this chapter is used exclusively for referring to nodes of this tree of potential oriented cycle-paths.

At a node $u_1 \to \cdots \to u_l$, the final vertex u_l has at least one incident L-colored face and one incident R-colored face – the two faces incident to $u_{l-1}u_l$. There might be other colored faces incident to u_l. Now, to extend this path, we cannot go between two L-colored faces, since this will put one of these two L-colored faces on the right of the path, making the coloring illegal. Likewise, we cannot go between two R-colored faces. Hence, we color all faces between two L-colored faces L, and all faces between two R-colored faces R. We say that such faces are "lost". After this "coloring completion" stage, the uncolored faces form a continuous range; such uncolored faces are "free". As an example, see Figure 2.2. We denote the number of free faces as $k(u_1 \to \cdots \to u_l)$. If this number is equal to k, we say that the node is k-free. We see that at a k-free node, there are exactly $k+1$ vertices onto which the path may be extended. If we keep track of unused vertices n and uncolored faces f, this observation gives us the following recursive formula for the number of potential oriented cycle-paths:

$$P(n,f) \leq \max_{k\geq 0}(k+1)\cdot P(n-1,f-k)$$
$$P(\cdot,0) = P(0,\cdot) = 1$$

Since we want to maximize $P(n,f)$, we can assume that the value of k for all nodes at the same level in the tree is equal. Call the value of k on the l-th level k_l, and assume that the tree has L levels. Then we have $L \leq n$, and $\sum_{l=1}^{L} k_l \leq f$; we wish to maximize $\prod_{l=1}^{L}(k_l+1)$. The product is maximized when $L = n$ and all k_l's are equal, the maximum value being $(f/n+1)^n$.

Since the number of faces in a planar graph is at most $2n-4$, a bound on the number of simple cycles is then given by $P(n,2n)$, up to a polynomial[3]. In the tree of $P(n,2n)$, all nodes have $k=2$

[3]To be precise, it is $(3n-6)\cdot P(n-2,2n-6)$, $(3n-6)$ being the maximum number of edges in planar graphs. After choosing a starting edge, two vertices and two faces are consumed. But the difference is accounted for in "up to a polynomial"

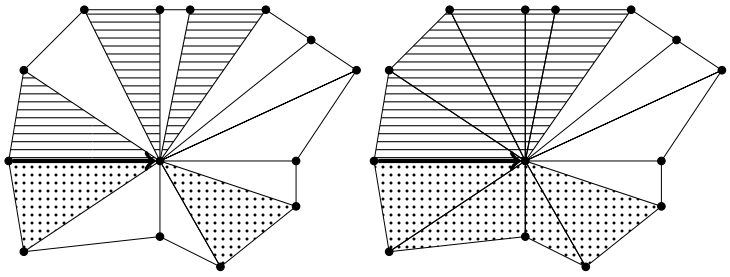


Figure 2.2: Completing the coloring after arriving at a vertex. Two faces are colored L while another two are colored R, adding up to a total loss of four faces. Another four faces remain uncolored and are thus free.

– all nodes are 2-free. If we can show that there has to be some non-2-free nodes in the tree, we will be able to tighten this bound.

We remark that $P(n, 2n) = O(3^n)$ alone is actually insufficient to derive the bound of $O(2.8927^n)$ obtained in [4], and tightening it will not immediately result in an improvement of the result. Instead, we need to consider cycles whose lengths are at most τn ($0 < \tau \leq 1$), whose number is given by $P(\tau n, 2n)$. Equivalently, we can compute $P(n, tn)$ for $t \geq 2$. After doing this, we will be able to combine this bound with another bound for cycles with length at most τn, take the minimum, and derive the ultimate bound.

We will get to the details in Section 2.3 and Section 2.4. For now, let us look into how to reduce the number of 2-free nodes.

2.2 Restricted-Child Rule

The restricted-child rule (RC-rule) is stated as follows:

Restricted-Child Rule. A 2-free node cannot have three 2-free children. In other words, one of the children must be non-2-free ("restricted").

The following lemma states that we can always pretend that the RC-rule is valid, the proof of which constitutes the rest of this section:

Lemma 2.1. *For any potential oriented cycle-path P such that $k(P) = 2$, either the RC-rule applies, or the subtree rooted at P can be dominated by another tree that satisfies the RC-rule.*

Proof. We find all possible violations of the RC-rule, analyze their corresponding trees of potential oriented cycle-paths, and show that they can be dominated by trees that conform to the RC-rule. We do this by a case distinction on the relationship between the grandchildren of the 2-free node. Since all nodes we are going to analyze start with P, we write the last vertex u in place of P when referring to nodes, omitting the preceding vertices. For instance, the node $P \to v_1$ is written as $u \to v_1$.

When drawing trees that result from analysis of actual vertices, we label the nodes with the name of the last vertex. If we also wish to analyze its children, we also list its k value in parentheses. This is to facilitate the computation of the numbers of remaining vertices and faces at leaves. To compute the number of remaining vertices, simply subtract from n the number of levels from the root to the leaf; to compute the number of remaining faces, subtract from f the sum of all k's on the path from the root to a leaf. For instance, in Figure 2.3, if the root has parameters (n, f), then the nodes w_1 and w_2 have parameters $(n-2, f-2)$, and nodes w_3 through w_6 have parameters $(n-2, f-4)$. When drawing trees that dominate the original trees, we only label the nodes with

their k values, since such trees only need to satisfy the constraints of $P(n, f)$ and do not need to have actual graph implementations, so there are no actual vertices to label them with.

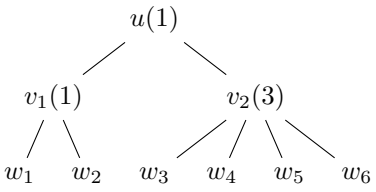


Figure 2.3: An illustrative drawing of a tree of potential oriented cycle-paths.

We now start our analysis. Since $u \to v_1$, $u \to v_2$ and $u \to v_3$ are valid children of u, the faces $\triangle uv_1v_2$ and $\triangle uv_2v_3$ are uncolored. Since the graph is fully triangulated, there is a vertex w_1 adjacent to both v_1 and v_2, and a vertex w_2 adjacent to both v_2 and v_3, such that $w_1, w_2 \neq u$. The vertices w_1 and w_2 may or may not be the same vertex; we distinguish the two cases.

Case 1. $w_1 \neq w_2$. See Figure 2.4. Consider face a. If it is L-colored, $k(u \to v_1) = 0$. If it is R-colored, $k(u \to v_2) = 0$. In both cases, the RC-rule is not violated. The same applies to face b. That leaves only the case where both faces are uncolored.

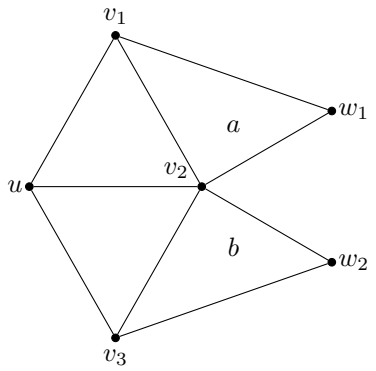


Figure 2.4: $w_1 \neq w_2$.

Now consider faces incident to v_2 that fall in the range $\angle w_1v_2w_2$. Assume that there is a colored face in this range. If the face is L-colored, then at $u \to v_2$ all faces between $\triangle uv_1v_2$ and this face are lost, including at least face a (see Figure 2.5). In this case, the f parameter goes down at least by 3 at $u \to v_2$, but it still only has 3 children. We can then replace the node $u \to v_2$ with a 3-free node, and this new tree dominates the original one (Figure 2.6). The same goes when there is an R-colored face within $\angle w_1v_2w_2$. That leaves only the case where there are no colored faces in $\angle w_1v_2w_2$. Since $w_1 \neq w_2$, there is at least one face in this range. Along with face a and b, there are at least three uncolored faces at $u \to v_2$; so $k(u \to v_2) \geq 3$. This again conforms to the RC-rule.

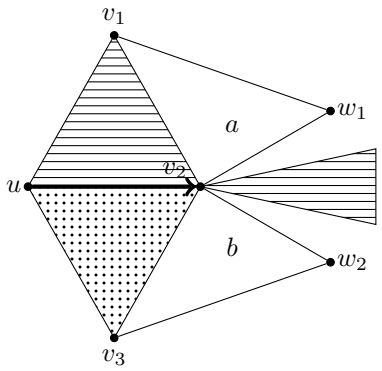
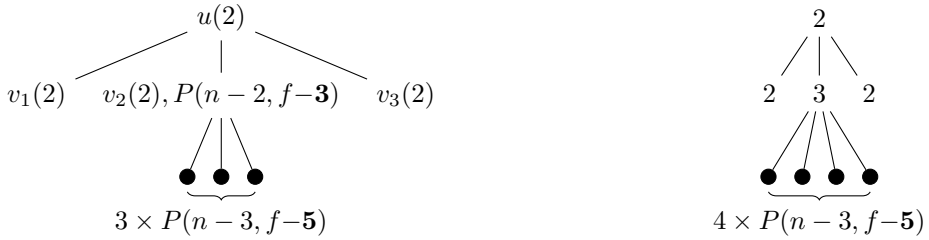


Figure 2.5: Face a is lost when there is an L-colored face in $\angle w_1 v_2 w_2$.



(a) The original tree according to the analysis.

(b) The tree that dominates the original tree.

Figure 2.6: Replacing the 2-free $u \to v_2$ node with a 3-free node when face a is lost.

Case 2. $w_1 = w_2$. Now that w_1 and w_2 are identical, we simply call it w. See the drawing in Figure 2.7. In this case, face a and b must be uncolored; otherwise $k(u \to v_2) < 2$. Face α must be uncolored: if it is L-colored, $k(v_1) = 1$. If it is R-colored, face a is lost at $u \to v_1$, and we can replace the node $u \to v_1$ with a 3-free node, in exactly the same way that we replace $u \to v_2$ in Figure 2.6. By the same argument, face β must be uncolored. To get $k(u \to v_1) = k(u \to v_3) = 2$, there must be an L-colored face opposing face α, and an R-colored face opposing face β. Note that x and y may be the same vertex.

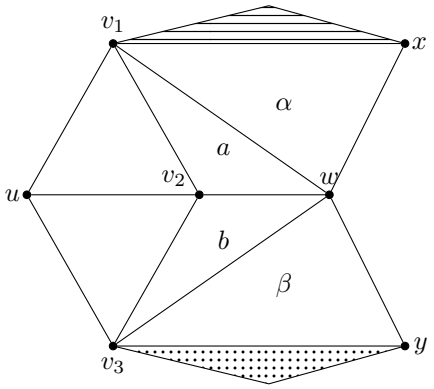
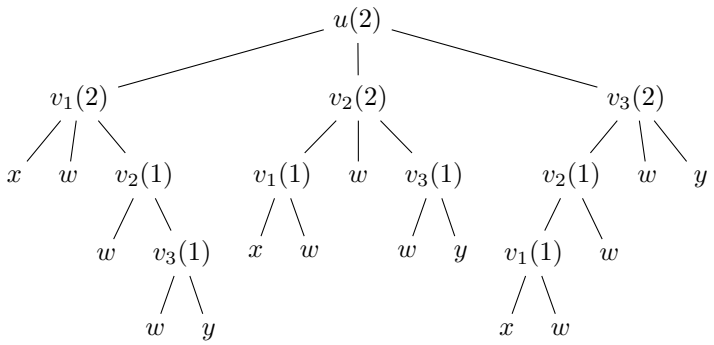


Figure 2.7: $w_1 = w_2$.

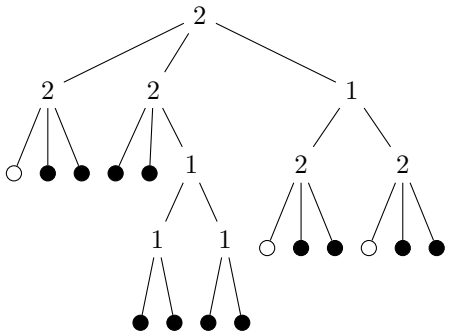
We can now draw the subtree below u (Figure 2.8a). By summing all the leaves of this tree, we can obtain $P(n, f) \leq 5P(n-2, f-4) + 6P(n-3, f-5) + 4P(n-4, f-6)$ in this case. This tree does not conform to the RC-rule, since all three children of the 2-free node u are 2-free.

However, it can be rearranged into the tree shown in Figure 2.8b, whose leaves also sum up to $5P(n-2,f-4)+6P(n-3,f-5)+4P(n-4,f-6)$. Now, if every time we encounter the original tree we replace it with the rearranged one, the final result will not decrease. We will use this "tree replacing" schema many times in the analysis.

The rearranged tree conforms to the RC-rule, if and only if there are at least one non-2-free nodes contributing to $P(n-2,f-4)$ and at least two non-2-free nodes contributing to $P(n-3,f-5)$ – by assigning such nodes to the leaves denoted by a hollow circle, we make the rearranged tree conform to the RC-rule. To know whether such a condition can be met, we distinguish whether $x = y$.



(a) The original tree according to the analysis.



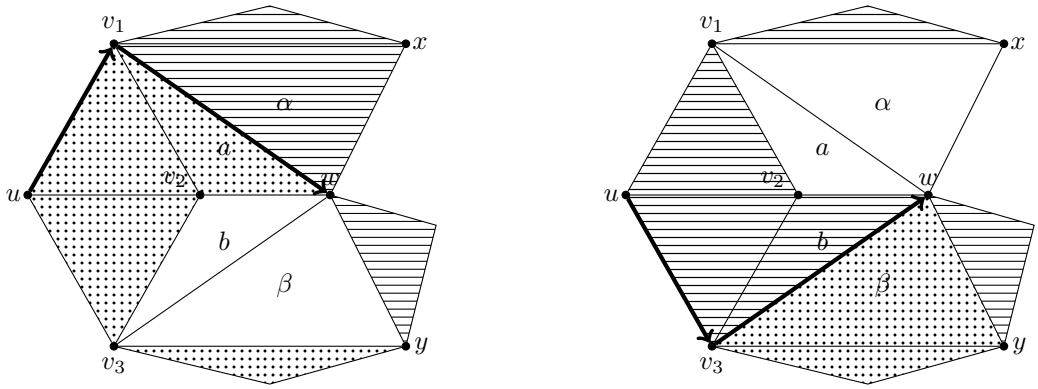
(b) Rearranging the tree, attempting to eliminate violations of the RC-rule.

Figure 2.8: The tree from u when $w_1 = w_2$ and its rearrangement.

Case 2.1. $w_1 = w_2 = w$ and $x \neq y$. We first consider the nodes $u \to v_1 \to w$ and $u \to v_3 \to w$. These two nodes contribute to $P(n-2,f-4)$. We show that it is impossible that both nodes are 2-free. Assume that $k(u \to v_1 \to w) = 2$. At $u \to v_1 \to w$, the faces $\triangle v_3 w v_2$ and $\triangle v_3 w y$ are uncolored; to have $k = 2$ at this point, the face within $\angle xwy$ that is adjacent to β must be L-colored (Figure 2.9a).With this L-colored face, we can see that $k(u \to v_3 \to w) = 0$ (Figure 2.9b). We conclude that there is at least one non-2-free node contributing to $P(n-3,f-5)$.

We now consider the nodes that contribute to $P(n-3,f-5)$. In particular, we consider all the nodes that end with w, namely $u \to v_i \to v_j \to w$ $(1 \leq i,j \leq 3,\ i \neq j)$. They are shown in Figure 2.10.

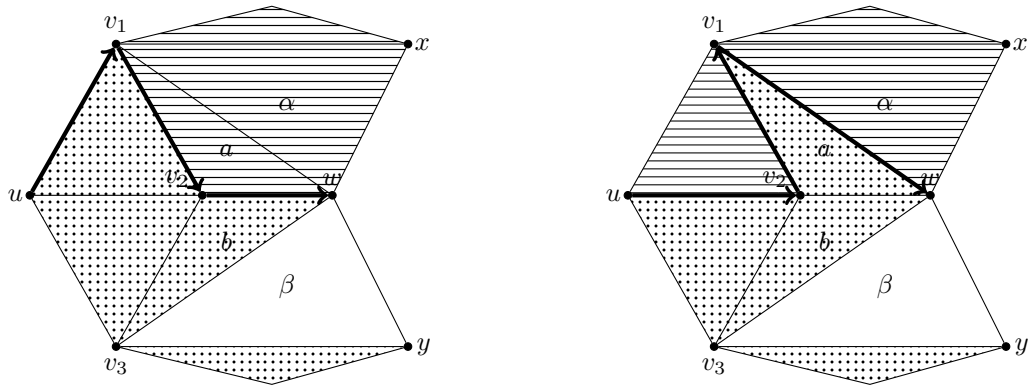
Note that the free faces at $u \to v_1 \to v_2 \to w$ is identical to those at $u \to v_2 \to v_1 \to w$, so $k(u \to v_1 \to v_2 \to w) = k(u \to v_2 \to v_1 \to w)$. Likewise, $k(u \to v_3 \to v_2 \to w) = k(u \to v_2 \to v_3 \to w)$. Hence, if any of these nodes is non-2-free, there is another node that is also non-2-free. By assigning those two nodes to the nodes denoted by hollow circles at Figure 2.8b, the violation of RC-rule is eliminated. That only leaves the case where all these nodes are 2-free, i.e. $k(u \to v_i \to v_j \to w) = 2$ for all $1 \leq i,j \leq 3 \wedge i \neq j$.



(a) The configuration when $k(u \to v_1 \to w) = 2$.

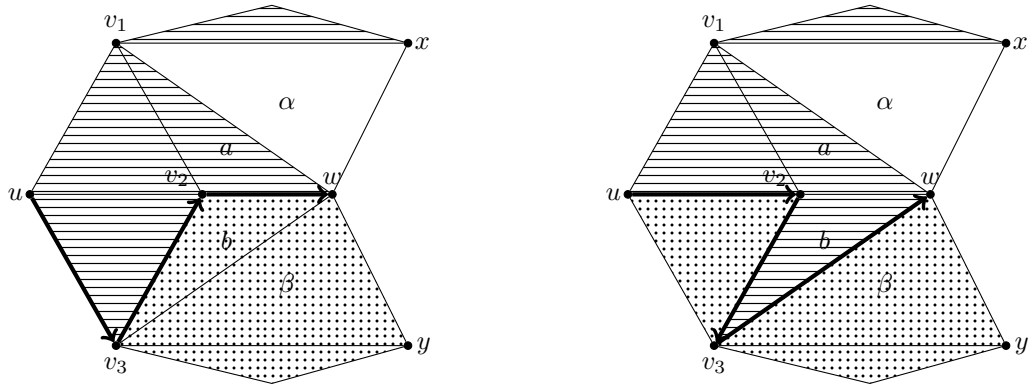
(b) In this configuration, $k(u \to v_3 \to w) = 0$.

Figure 2.9: If $k(u \to v_1 \to w) = 2$, $k(u \to v_3 \to w) = 0$.



(a) $u \to v_1 \to v_2 \to w$

(b) $u \to v_2 \to v_1 \to w$



(c) $u \to v_3 \to v_2 \to w$

(d) $u \to v_2 \to v_3 \to w$

Figure 2.10: $u \to v_i \to v_j \to w$ $(1 \leq i, j \leq 3, i \neq j)$

Now consider the faces in $\angle xwy$. If there is an L-colored face in this range, face α is lost in $u \to v_3 \to v_2 \to w$ and $u \to v_2 \to v_3 \to w$. We can replace those two nodes with 3-free nodes, restoring the RC-rule. With the same argument, there can be no R-colored faces in $\angle xwy$. So all faces in $\angle xwy$ must be uncolored. If there are at least two faces in this range, $k(u \to v_i \to v_j \to w) \geq 3$, again satisfying the RC-rule. That leaves only the case where there is exactly one uncolored face within this region; see Figure 2.11.

Now that we have more information above this neighborhood, we can do a finer analysis of the subtree below u, analyzing the children of all nodes that end with w; see Figure 2.12a. According to the analysis, we have $P(n,f) \leq 2P(n-2,f-4)+2P(n-3,f-5)+6P(n-3,f-7)+2P(n-4,f-6)+12P(n-4,f-7)+10P(n-5,f-7)$. This tree can be rearranged into the tree in Figure 2.12b to conform to the RC-rule. This concludes the analysis of this case.

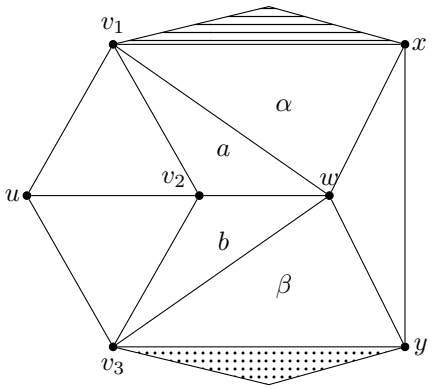
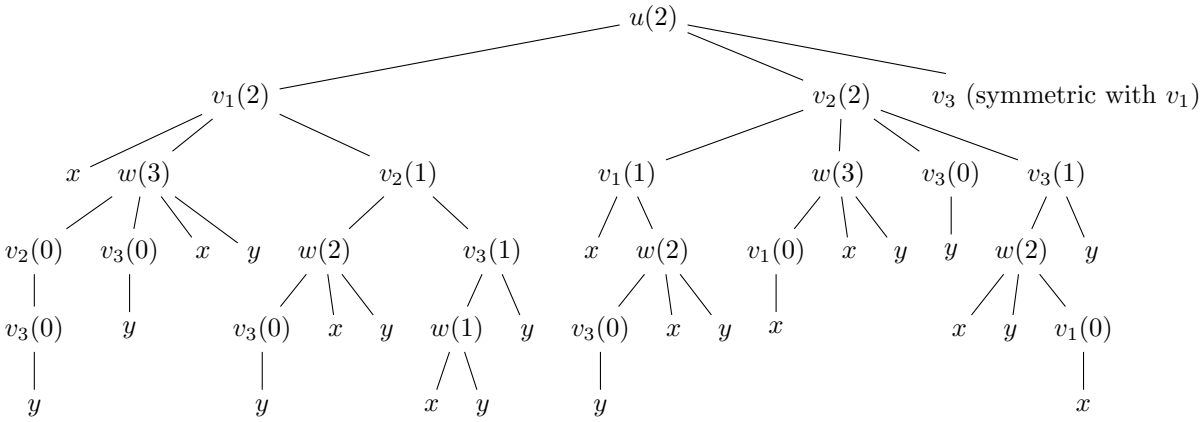
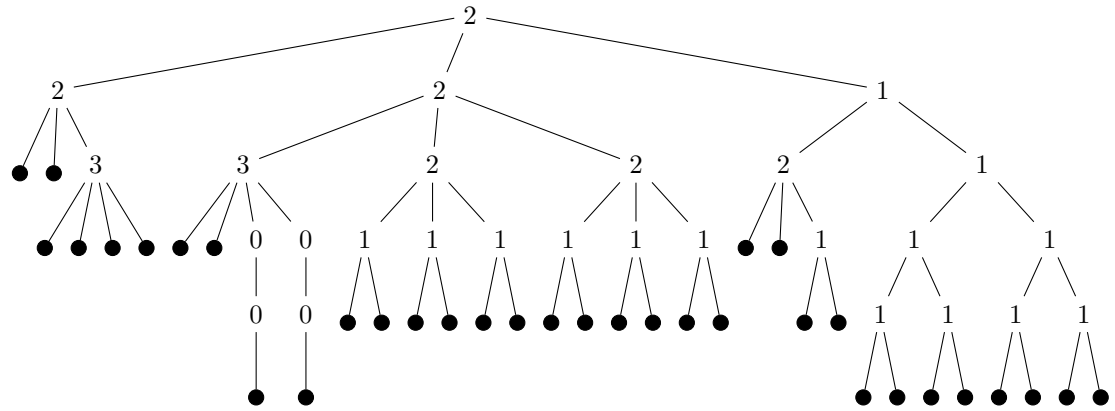


Figure 2.11: $w_1 = w_2$ and there is exactly one uncolored face in $\angle xwy$.



(a) The original tree according to the analysis.



(b) Rearranging the tree, eliminating violations of the RC-rule.

Figure 2.12: The subtree from u when $w_1 = w_2$, $x \neq y$ and there is exactly one uncolored face in $\angle xwy$, and its rearrangement.

Case 2.2. $w_1 = w_2 = w$ and $x = y$. The setting is shown in Figure 2.13. The subtree is analyzed in Figure 2.14a, giving us $P(n, f) = 2P(n-2, f-4) + 3P(n-3, f-6) + 2P(n-3, f-5) + 10P(n-4, f-6) + 8P(n-5, f-6)$. A tree that dominates it is shown in Figure 2.14b. The dominating tree has 11 leaves contributing to $P(n-5, f-6)$ (instead of 8 in the original tree); it strictly dominates the original tree.

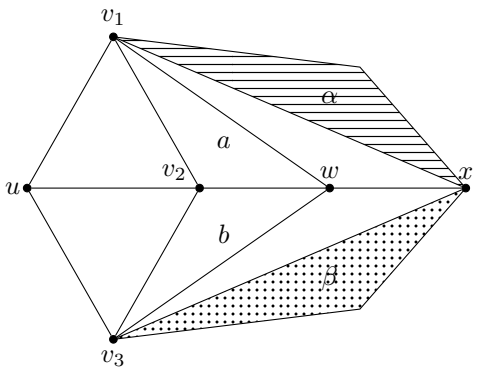
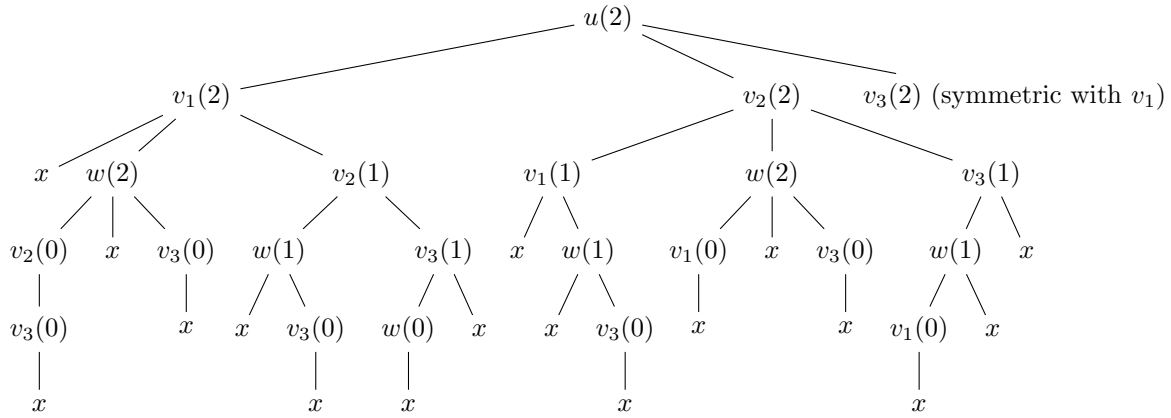
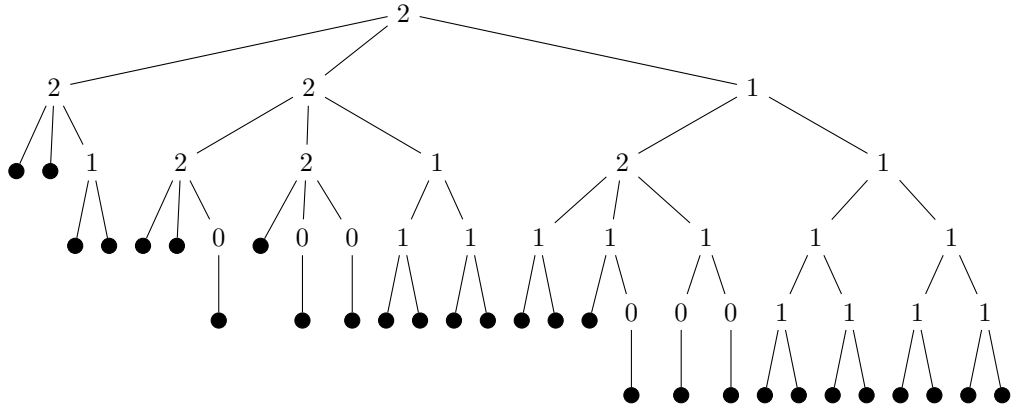


Figure 2.13: $w_1 = w_2$ and $x = y$.



(a) The original tree according to the analysis.



(b) A tree that dominates the original tree and satisfied the RC-rule.

Figure 2.14: The subtree from u when $w_1 = w_2$ and $x = y$, and a tree that dominates it.

□

2.3 Shape of the P-tree

The tree generated by $P(n, f)$ does not need to have an implementation of a tree of cycle-paths; to distinguish from an actual tree of potential oriented cycle-paths, we call it the P-tree.

Without the RC-rule, the tree generated by $P(n, f)$ have equal k's at each level; to determine $P(n, f)$ is to determine how to distribute f onto n levels. Let $t = n/f$, then the k's will consist of $\lfloor t \rfloor$ and $\lceil t \rceil$. These layers can be arranged arbitrarily; so which number comes first does not matter. That is to say, both $k = \lfloor t \rfloor$ and $k = \lceil t \rceil$ can maximize $P(n, tn)$.

We now analyze the shape of the P-tree *with* the RC-rule. Formally, $P(n, f)$ is now given by:

$$P(n, f) = \max\{P'(n, f), 2P(n-1, f-2) + P'(n-1, f-2)\}$$
$$P'(n, f) = \max_{k \neq 2, k \leq f} (k+1)P(n-1, f-k)$$

Obviously, $P'(n, f) \leq P(n, f)$.

When $f \leq 1.5$ or $f \geq 2.5$, the RC-rule does not play a role. When $f \leq 1$ or $f \geq 3$, there need not be any $k = 2$ layers at all. When $1 < f \leq 1.5$, there are no more $k = 2$ layers than $k = 1$ layers, so we can always the arrange the layers so that two $k = 2$ layers do not appear consecutively;

the same argument applies when $2.5 \le f < 3$. In particular, when $f = 2.5$, the P-tree consists of alternate layers of 2-free nodes and 3-free nodes; $P(n, 2.5n) = 2^{n/2} \cdot 3^{n/2} = \sqrt{6}^n \approx 2.4495^n$.

We now analyze the shape of the P-tree when $1.5 < f < 2.5$. Let $k(n, f)$ be the optimal choice of k in $P(n, f)$, and $k'(n, f)$ be the optimal choice of k in $P'(n, f)$. We first show that when n is large enough and $2n \le f < 2.5n$, $k(n, f) = 2$ and $k'(n, f) = 3$.

Lemma 2.2. *When $2 \le t < 2.5$, $P(n, tn)$ is maximized when $k = 2$.*

Proof. We prove by induction.

$n = 1$ **(basis)**. $2 \le f < 2.5 \Rightarrow f = 2$. $P(1, 2)$ is indeed maximized when $k = 2$.

$n \ge 2$ **(inductive step)**. We prove by contradiction. Assume that $k(n, f) \ne 2$, the RC-rule does not apply; then $P(n, f) = (k + 1)P(n - 1, f - k)$. Let $t' = (f - k)/(n - 1) = (tn - k)/(n - 1) = t + (t - k)/(n - 1)$.

If $k = k_1 \le 1$, $t' > t \ge 2$.

If $t' < 2.5$, by induction hypothesis, $k_2 = k(n - 1, f - k) = 2$. This gives us $P(n, f) = (k+1)P(n-1, f-k) = (k+1)(2P(n-2, f-k-2)+P'(n-2, f-k-2)) \le 3(k+1)P(n-2, f-k-2)$. Now, if we choose $k(n, f) = 2$ and $k(n - 1, f - 2) = k_1$ (since $k_1 \le 1$, this does not violate the RC-rule) we obtain $P(n, f) = 3(k + 1)P(n - 2, f - k - 2)$. So choosing $k(n, f) = 2$ does not decrease $P(n, f)$.

If $t' \ge 2.5$, we know that $k_2 = k(n - 1, f - k) = [t'] \ge 3$, giving us $P(n, f) = (k_1 + 1)(k_2 + 1)P(n-2, f-k_1-k_2)$. Unless $k_2 = 3$, we can "redistribute" k_1 and k_2 by choosing $k(n, f) = k_1+1$ and $k(n-1, f-2) = (k_2-1)$, which gives us $P(n, f) = (k_1+2)k_2(k_1+k_2-2)P(n-2, f-k_1-k_2) > (k_1 + 1)(k_2 + 1)P(n - 2, f - k_1 - k_2)$. This contradicts with the assumption that $k(n, f) = k_1$. When $k_2 = 3$ and $k_1 = 0$, we can redistribute to 2 and 1, getting $P(n, f) = (2 + 1) \cdot (1 + 1)P(n - 2, f - 3) > (3 + 1) \cdot (0 + 1)P(n - 2, f - 3)$; that leaves only the case of $k_2 = 3$ and $k_1 = 1$. In this case, $f < 2.5n \Rightarrow f - 4 < 2.5(n - 2) + 1$, $f - 1 > 2.5(n - 1) \Rightarrow f - 4 > 2.5(n - 2) - 0.5$; $2.5(n-2)-0.5 < f-4 < 2.5(n-2)+1$. If n is even, $f-4 = 2.5(n-2)$, and $k_3 = f(n-2, f-4) = 2$. We can swap k_1 with k_3 to obtain $k(n, f) = 2$. If n is odd, $f-4 = 2.5(n-2)+0.5$, and the P-tree of $P(n - 2, f - 4)$ consists of alternating layers of 3-free and 2-free nodes. We simply swap any $k = 2$ layer.

If $k = k_1 \ge 3$, $t' < t$. A similar argument applies. If $k_2 = k(n - 1, f - k) = 2$, we swap k_1 with k_2. If $k_2 \le 1$, we first swap k_1 and k_2 and can then redistribute them in the way described above. □

Lemma 2.3. *When $2 \le t < 2.5$, $P'(n, tn)$ is maximized when $k = 3$.*

Proof. We prove by induction.

$n = 2$ (basis). $4 \le f < 5 \Rightarrow f = 4$. $P'(2, 4)$ is indeed maximized when $k = 3$.

$n \ge 3$ (inductive step).

If $k'(n, tn) = k_1 \le 1$, then $t' = (tn-k_1)/(n-1) > 2$. So $k_2 = k(n-1, tn-1) \ge 2$. If $k_2 \ge 3$, we can redistribute the two layers to 3 and $k_2 - 2$. If $k_2 = 2$, then from Lemma 2.2 and the induction hypothesis we know that the tree of $P(n - 1, tn - 1)$ consists entirely of $k = 2$ and $k = 3$ nodes, and along every path to a leaf, there is at least one $k = 3$ node. Now we modify the tree such that the highest $k = 3$ node along each path now has $k = 1$, and change k_1 to 3. This maintains the value of $P(n, tn)$.

If $k_1 \ge 4$, then $k_2 \le 2$. If $k_2 = 2$, redistribute to $k_1 = 3$ and $k_2 = 3$. Otherwise $k_2 \le 1$; we swap k_1 and k_2 and this falls into the case of $k_1 \le 1$.

□

Lemma 2.4. *When $1.5 < t < 2$, $k(n, tn) = 2$ and $k'(n, tn) = 1$.*

Proof. The proof is symmetric to the proofs of Lemma 2.2 and Lemma 2.3. □

2.4 Deriving the Bound

With the shape of the P-tree now clear, we now derive $P(n, tn)$ for $2 \leq t < 2.5$ (the reason we restrict t to this range will be clear later). Instead of a precious formula, we are going to find the function $b(t)$ such that $P(n, tn) = \Theta(b(t)^n)$ for $2 \leq t \leq 2.5$.

Let $P(n, 2n) = p(n)$, $P(n, 2n-1) = p'(n)$. From the previous section, we know that the children of the root of the tree for $P(n, 2n)$ are two 2-free nodes ($P(n-1, 2n-2) = p(n-1)$) and one 3-free nodes, which in turn has four nodes ($P(n-2, 2n-5) = p'(n-2)$) as its children. The children of the root of the tree for $P(n, 2n-1)$ are two 2-free nodes ($P(n-1, 2n-3) = p'(n-1)$) and one 1-free node, which in turn has two nodes ($P(n-2, 2n-4) = p(n-2)$) as its children. Then we have

$$\begin{aligned} p(n) &= 2p(n-1) + 4p'(n-2) \\ p'(n) &= 2p'(n-1) + 2p(n-2) \\ p(1) &= 3, p(2) = 8 \\ p'(1) &= 2, p'(2) = 6 \end{aligned}$$

The solution is $p(n) = \Theta(x^n)$, where $x = 1 + \sqrt{1 + 2\sqrt{2}}$ is the real positive root for $(x^2 - 2x)^2 = 8$. So $b(2) = 1 + \sqrt{1 + 2\sqrt{2}} = \approx 2.956637$.

When $t = 2.5$, the P-tree consists of alternating layers of 2's and 3's; $b(2.5) = \sqrt{(2+1) \cdot (3+1)} = \sqrt{12} \approx 3.464102$.

To find $P(n, tn)$ $(2 < t < 2.5)$, note that as we go down the P-tree, sooner or later t will hit 2 or 2.5. We write $tn = 2n + \alpha = 2.5n - \beta/2$, to keep track of how close tn is to $2n$ or $2.5n$. When $\alpha = 0$, $t = 2$; when $\beta = 0$, $t = 2.5$. $n = 2\alpha + \beta, \alpha = (t-2)n, \beta = (5-2t)n$.

If $k = 2$, α remains unchanged, and β goes down by 1. If $k = 3$, α decreases by 1, and β increase by 1. Write $Q(\alpha, \beta) = P(n, f)$. We have $P(n, f) = 2P(n-1, f-2) + P'(n-1, f-2) = 2P(n-1, f-2) + 4P(n-2, f-5)$, which leads to

$$Q(\alpha, \beta) = 2Q(\alpha, \beta - 1) + 4Q(\alpha - 1, \beta)$$

To derive $Q(\alpha, \beta)$, we sum over all the leaves of the recursion tree – $Q(0, \beta')$ and $Q(\alpha', 0)$. To get to $Q(0, \beta')$, we must take α 4's and $(\beta - \beta')$ 2's. The number of $Q(0, \beta')$ leaves is exactly the number of such sequences: $\binom{\alpha+\beta-\beta'}{\alpha}$. The value of each such leaf is $b^{\beta'}(2)$. Likewise, the number of $Q(\alpha', 0)$ leaves is $\binom{\alpha+\beta-\alpha'}{\beta}$, and the value of each such leaf is $b^{2\alpha'}(2.5)$. Hence:

$$\begin{aligned} & P(n, tn) = Q(\alpha, \beta) \\ =& \sum_{\beta'=1}^{\beta} \binom{\alpha + \beta - \beta'}{\alpha} \cdot 4^{\alpha} \cdot 2^{\beta - \beta'} \cdot b^{\beta'}(2) + \sum_{\alpha'=1}^{\alpha} \binom{\alpha + \beta - \alpha'}{\beta} \cdot 4^{\alpha - \alpha'} \cdot 2^{\beta} \cdot b^{2\alpha'}(2.5) \\ =& \sum_{\beta'=1}^{(5-2t)n} \binom{(3-t)n - \beta'}{(t-2)n} \cdot 4^{\alpha} \cdot 2^{\beta - \beta'} \cdot b^{\beta'}(2) + \sum_{\alpha'=1}^{(t-2)n} \binom{(3-t)n - \alpha'}{(5-2t)n} \cdot 4^{\alpha - \alpha'} \cdot 2^{\beta} \cdot b^{2\alpha'}(2.5) \\ \leq& (5-2t)n \max_{\beta'=0}^{(5-2t)n} \binom{(3-t)n - \beta'}{(t-2)n} \cdot 2^n \cdot (\frac{b(2)}{2})^{\beta'} + (t-2)n \max_{\alpha'=0}^{(t-2)n} \binom{(3-t)n - \alpha'}{(5-2t)n} \cdot 2^n \cdot (\frac{b(2.5)}{2})^{2\alpha'} \end{aligned}$$

To simplify the calculation, we also include the cases of $\beta' = 0$ and $\alpha' = 0$ in the maximum. To find which β' and α' give the maximum, we consider the ratio of consecutive terms. For β', this is:

$$\rho_\beta(\beta') = \frac{\binom{(3-t)n-\beta'-1}{(t-2)n}}{\binom{(3-t)n-\beta'}{(t-2)n}} \cdot \frac{b(2)}{2} = \frac{(3-t)n-\beta'-(t-2)n}{(3-t)n-\beta'} \cdot \frac{b(2)}{2} = \frac{(5-2t)n-\beta'}{(3-t)n-\beta'} \cdot \frac{b(2)}{2}$$

As β' increases from 0 to $(5-2t)n$, $\rho_\beta(\beta')$ decreases from $\frac{5-2t}{3-t} \cdot \frac{b(2)}{2}$ to 0. To know where the maximum value is attained, we need to distinguish whether $\rho_\beta(0) \leq 1$. If that is the case, $t \geq \frac{5b(2)-6}{2b(2)-2} \approx 2.244459$; the maximum is attained at $\beta' = 0$, and is

$$\binom{(3-t)n}{(t-2)n} \cdot 2^n \sim 2^{H(\frac{t-2}{3-t})\cdot(3-t)n} \cdot 2^n = (2^{1+(3-t)\cdot H(\frac{t-2}{3-t})})^n$$

Otherwise, the maximum is attained when $\rho_\beta(\beta') = 1 \Leftrightarrow \beta' = \frac{(5b(2)-6-2b(2)t+2t)n}{b(2)-2} = (\lambda-\mu t)n$, and is

$$\binom{(3-\lambda+\mu t-t)n}{(t-2)n} \cdot 2^n \cdot (\frac{b(2)}{2})^{(\lambda-\mu t)n} \sim (2^{1+(3-\lambda+\mu t-t)\cdot H(\frac{t-2}{3-\lambda+\mu t-t})} \cdot (\frac{b(2)}{2})^{\lambda-\mu t})^n$$

Likewise for α':

$$\rho_\alpha(\alpha') = \frac{\binom{(3-t)n-\alpha'-1}{(5-2t)n}}{\binom{(3-t)n-\alpha'}{(5-2t)n}} \cdot \frac{b^2(2.5)}{4} = \frac{(3-t)n-\alpha'-(5-2t)n}{(3-t)n-\alpha'} \cdot \frac{b^2(2.5)}{4} = 3 \cdot \frac{(t-2)n-\alpha'}{(3-t)n-\alpha'}$$

As α' increases from 0 to $(t-2)n$, $\rho_\alpha(\alpha')$ decreases from $3 \cdot \frac{t-2}{3-t}$ to 0. Again, we distinguish whether $\rho_\alpha(0) \leq 1$. If that is the case, $t \leq \frac{2\cdot 3+3}{3+1} = 2.25$; the maximum is attained at $\alpha' = 0$, and is

$$\binom{(3-t)n}{(5-2t)n} \cdot 2^n \sim (2^{1+(3-t)\cdot H(\frac{5-2t}{3-t})})^n$$

Otherwise, the maximum is attained when $\rho_\alpha(\alpha') = 1 \Leftrightarrow \alpha' = (2t-4.5)n$, and is

$$\binom{(7.5-3t)n}{(5-2t)n} \cdot 2^n \cdot 3^{(2t-4.5)n} \sim (2^{1+(7.5-3t)\cdot H(\frac{5-2t}{7.5-3t})} \cdot 3^{2t-4.5})^n$$

Putting all these together, when $2 < t < 2.5$ we have $b(t) = \max\{b_\beta(t), b_\alpha(t)\}$, where

$$b_\beta(t) = \begin{cases} 2^{1+(3-\lambda+\mu t-t)\cdot H(\frac{t-2}{3-\lambda+\mu t-t})} \cdot (\frac{b(2)}{2})^{\lambda-\mu t} & \text{if } t \leq \frac{5b(2)-6}{2b(2)-2} \approx 2.244459 \\ 2^{1+(3-t)\cdot H(\frac{t-2}{3-t})} & \text{otherwise} \end{cases}$$

$$\lambda = \frac{5b(2)-6}{b(2)-2}, \mu = \frac{2b(2)-2}{b(2)-2}$$

$$b_\alpha(t) = \begin{cases} 2^{1+(3-t)\cdot H(\frac{5-2t}{3-t})} & \text{if } t \leq 2.25 \\ 2^{1+(7.5-3t)\cdot H(\frac{5-2t}{7.5-3t})} \cdot 3^{2t-4.5} & \text{otherwise} \end{cases}$$

$b_\beta(t)$ and $b_\alpha(t)$ are shown in Figure 2.15, along with the "old" base $t+1$ in dashed lines. Interestingly, $b(t)$ is remarkably linear despite of its very complex nature.

Theorem 2.5. *The number of simple cycles in a planar graph with n vertices is bound from above by* $O(2.870214^n)$.

Proof. We have calculated a function $b(t)$ $(2 \leq t \leq 2.5)$ such that $P(n, tn) = \Theta(b(t)^n p(n))$, where $p(n)$ is a polynomial. We use this to derive a bound on the simple cycles whose length is τn:

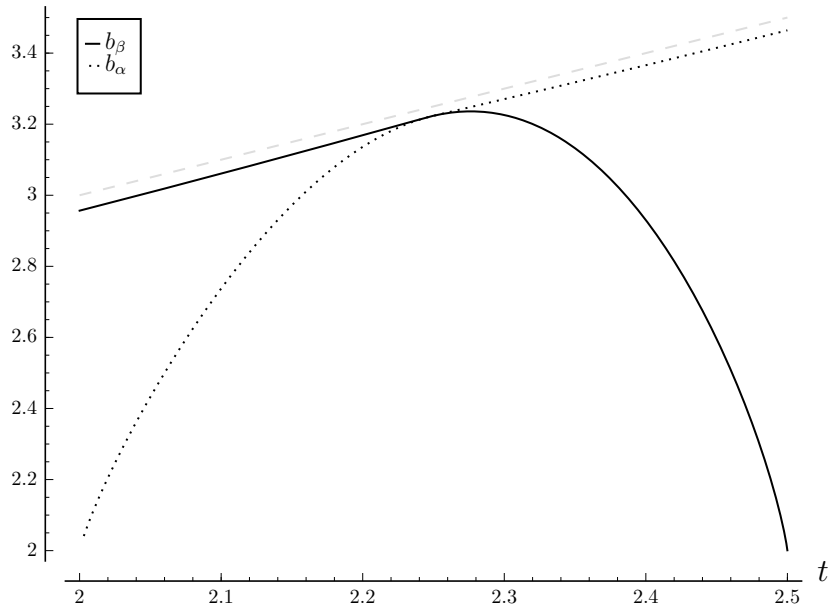


Figure 2.15: $b_\beta(t)$ and $b_\alpha(t)$, for $t \in [2, 2.5]$.

$$C_s^\tau(n) = P(\tau n, 2n) = O((b(\frac{2}{\tau})^\tau)^n)$$

Since we only calculated $b(t)$ for $2 \leq t \leq 2.5$, we only use it when $2 \leq \frac{2}{\tau} \leq 2.5 \Leftrightarrow 0.8 \leq \tau \leq 1$. Elsewhere we use the old value $b(t) = t + 1$.

We combine it with another bound – the Hamiltonian bound – for $C_s^\tau(n)$ given in [4]:

$$C_s^\tau(n) = O((\frac{\sqrt[4]{30}^\tau}{\tau^\tau(1-\tau)^{(1-\tau)}})^n)$$

We plot the bases of the two bounds in Figure 2.16 (note the discontinuity at $\tau = 0.8$). The bound we have just derived is shown in solid lines, while the Hamiltonian bound is shown in dotted lines. Since both are valid upper bounds, the lower envelope of these two functions is a bound on the exponential growth rate of $C_s^\tau(n)$. As we can see from the graph, the intersection point is where the maximum value is attained. According to numerical computation, the two curves intersect at $\tau \approx 0.924109$, where the value is approximately 2.870213. Since

$$C_s(n) \leq \sum_{l=1}^{n} C_s^{l/n}(n) \leq n \cdot \max_{l=1}^{n} C_s^{l/n}(n) \leq n \cdot \max_{0<\tau\leq 1} C_s^\tau(n)$$

We have $C_s(n) = O(n \cdot 2.870213^n) = O(2.870214^n)$.

□

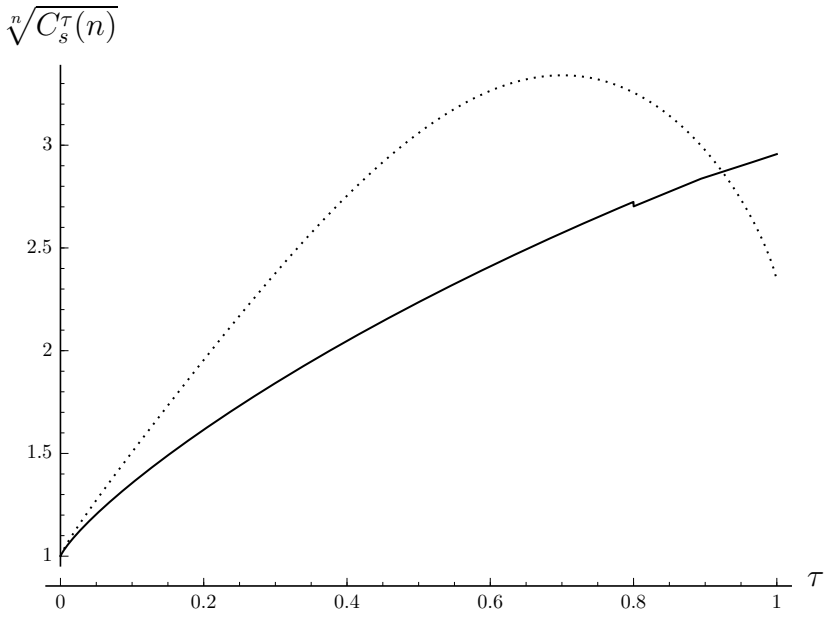


Figure 2.16: Base of $C_s^\tau(n)$ according to two different bounds.

2.5 Discussion

We believe that the multi-level analysis employed here has more potential waiting to be exploited. In our analysis, only the case of $k = 2$ has been treated specially to derive a bound for $P(\tau n, 2n)(0.8 \leq \tau \leq 1)$, since it is one of the two most relevant cases when τ is close to 1, the other being $k = 3$. We believe that a similar analysis can work for the case of $k = 3$, and combining the constraints on $k = 2$ and $k = 3$ cases can, in principle, yield an even tighter bound. However, in our analysis we were able to converge to a relatively simple rule (the restricted-child rule) which then allows us derive the bound. If we add more constraints it will be much more challenging to do so.

We also recognize a significant limitation of our approach. In our analysis, we have taken advantage of lost faces. However, since this effect is non-local in nature – faces can be lost due to decisions made many steps ago – it is very hard to account for in our intrinsically local analysis. Further advancements on the upper bound of $C_s(n)$ might depend on an effective way to analyze this effect.

We conclude with the remark that there exists a less powerful but much simpler way to tighten $b(t)$ and thus tighten $C_s(n)$, without the use of the RC-rule. Knowing that all k's must be integers, when $2 < t < 3$, we cannot make distribute tn evenly onto n elements by having all k_l's equal to t and get $b(t) = t + 1$. Instead, we need to combine $(3 - t)n$ 2's and $(t - 2)n$ 3's to achieve the maximum. This gives us $b(t) = (2+1)^{3-t} \cdot (3+1)^{t-2}$. Using this expression for $b(t)$ and combining it with the Hamiltonian bound, $C_s(n)$ can be improved to $O(2.880050^n)$, which is in the midway of the bound in [4] ($O(2.8927^n)$) and the bound obtained by using the RC-rule ($O(2.870214^n)$).

Chapter 3

Refutation of Biswas et al.'s Bound on Hamiltonian Cycles

In this chapter, we point out some arguments in [3] that are insufficient to justify their conclusions. This does not mean that the conclusion is necessarily wrong, but a proper proof will require more rigorous justifications and most likely very different arguments.

3.1 Doubtful Inequality in Proof of Lemma 3

The concept of *restricted Hamiltonian cycles* is defined in [3] by partitioning the edges at each vertex v into two sets – s_v and s'_v, and require that for each vertex v, the two edges it is incident to in the Hamiltonian cycle be one in s_v and another in s'_v. It then considers the same Hamiltonian cycle with two different orientations, and introduces the following inequality:

$$\sum_{l \leq L} (k_l + k'_l) \leq 2n - 6$$

The justification supposedly being that k_l and k'_{L+1-l} is the value of k at the same vertex v in P and P' respectively, v being the l-th vertex in P and $(L+1-l)$-th vertex in P'. Since this is a restricted setting, summing them will supposedly render the number of free faces in the unrestricted setting.

This is not correct for two reasons. First, since the predecessors of v is different in P and P', the free faces are usually different, so summing them does not necessarily give the number of free faces in the unrestricted setting. Second and more importantly, this argument uses k_l in a wrong way. It assumes that the number of free faces at the l-th vertex of an oriented cycle-path is simply k_l. In [4], k_l is defined as the k that maximizes $P(n-(l-1), f - \sum_{i=1}^{l-1} k_i)$. As such, k_l is meaningful only when analyzing the recursion itself. Indeed, there are many possible choices for oriented cycle-paths of length L starting from a specific edge, and it is not hard to imagine that the number of free faces at the l-th vertex differs from one oriented cycle-path to another. Thus k_l is not well-defined when analyzing individual cycles, and the use of k_l in this inequality is not justified.

3.2 Incorrect Generalization of Lemma 3 to Multigraphs

Lemma 3 claims that the number of restricted Hamiltonian cycles in a planar graph is $O(n \cdot 2^n)$. In the proof of Theorem 4, a perfect matching M is found and then all edges (u, v) are contracted into a single vertex, the edges originally incident to u put in s_v and the edges originally incident

to v put in s'_v. However, after this contraction, the graph becomes a multigraph, since u and v may have a common adjacent vertex. Lemma 3 doesn't deal with multigraphs; so this application of Lemma 3 is erroneous. In particular, in this new multigraph, the number of faces is no longer at most twice the number of vertices, but at most four times the number of vertices, since the number of faces did not change while the number of vertices has halved.

We can generalize Lemma 3 to multi-graphs though, by modifying the conclusion from $O(n \cdot 2^n)$ to $O(n \cdot (\frac{f}{2n} + 1)^n)$. However, in this case, in the multigraph this gives $O(n \cdot 3^n)$, a bound too big to improve the bound on the number of Hamiltonian cycles.

Hence, even if Lemma 3 is correct, the proof of Theorem 4 is flawed.

3.3 Conclusion

Originally we hoped to fix the arguments for the upper bound on Hamiltonian cycles in [3] and in this way provide evidence for the bounds claimed therein. Unfortunately, we currently do not see a way forward.

Chapter 4

Number of Perfect Matchings

4.1 Preliminaries

A *twisted cylinder* of width w and size n, $T_w(n)$, can be defined recursively as follows:

- $T_w(0)$ is an empty graph.
- $T_w(n)$ $(n > 0)$ is constructed on top of $T_w(n-1)$ by adding a vertex n, and connect it to vertex $n-1$ if $n > 1$, and to vertex $n-w$ if $n > w$, and to vertex $n-w-1$ if $n > w+1$.

As an example, $T_6(20)$ is shown in Figure 4.1.

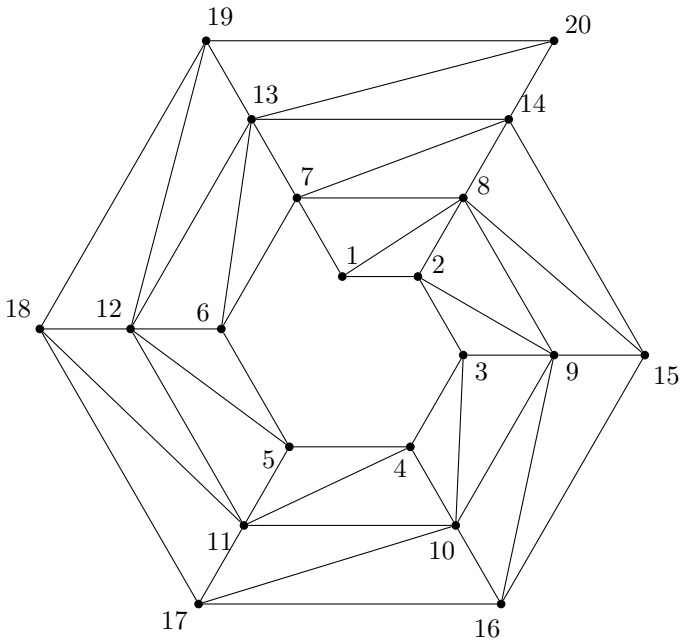


Figure 4.1: A twisted cylinder of width 6 and size 20.

.

The concept of twisted cylinders is used in [4] to prove lower bounds, which constructed twisted cylinders in another (equivalent) way.

If, instead of connecting vertex n to $n-w-1$ when $n \geq w+1$, we connect vertex n to $n-w+1$

when $n \geq w - 1$, we can obtain a variation of twisted cylinders, named a *prism* in [2][1]. A prism of width 6 and size 20, $T'_6(20)$, is shown in Figure 4.2.

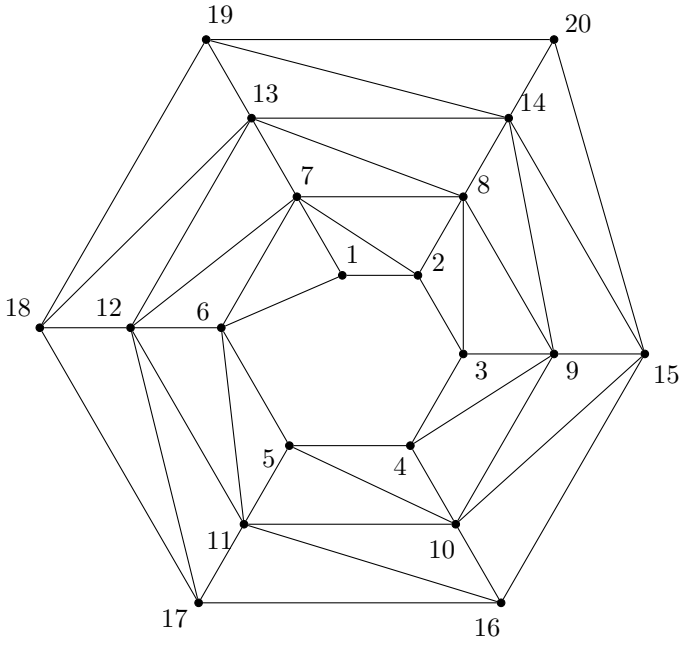


Figure 4.2: A prism of width 6 and size 20.

.

4.2 Transfer Matrix

In [2], the prism graph was used to derive a lower bound on the maximum number of perfect matchings. If we use the twisted cylinder instead, a better bound can be derived. However, in this case, the method in [2] is no longer sufficient. We adopt the transfer matrix method used in [4], where it was used to derive a lower bound on the number of simple cycles using twisted cylinders.

For T_w, the *signature* is a binary number with $w + 1$ bits. These numbers represent whether the last $w + 1$ vertices have been matched. The highest significant bit represents whether the last vertex has been matched, while the lowest significant bit represents whether the last but w vertex has been matched. Vertices before that must be matched.

Given a signature $b = b_1 \cdots b_{w+1}$ in $T_w(n)$, we consider which signatures it can transfer to in $T_w(n + 1)$. If $b_{w+1} = 1$, i.e. the last but w vertex has been matched, we simply drop it, and the next state is $0b_1 \cdots b_w$, or $b \gg 1$. If $b_{w+1} = 0$, we need to match it with either b_w, b_1, or the new vertex that we are going to introduce:

1. If $b_w = 0$, then $0b_1 \cdots b_{w-1}1 = (b|1 \ll 1) \gg 1$ is a valid next state;
2. If $b_1 = 0$, then $01b_2 \cdots b_w = (b|1 \ll w) \gg 1$ is a valid next state;
3. $1b_1 \cdots b_w = (b \gg 1)|(1 \ll w)$ is always a valid next state.

We define the transfer matrix $M \in \{0, 1\}^{2^{w+1} \times 2^{w+1}}$ as follows:

$$M_{i,j} = \begin{cases} 1 & \text{if } j \text{ is a valid next state of } i \\ 0 & \text{otherwise} \end{cases}$$

[1]The paper went on to claim that "prism" and "twisted cylinder" are synonyms, but that is not the case.

Since all the states the signatures represent are reachable, M is primitive, and the eigenvalue with the largest absolute value is real and unique. Let $p_w(n)$ be the number of perfect matchings in $T_w(n)$; then $p_w(n) = O(c^n)$, where c is the largest real eigenvalue of M.

4.3 Result

A computer program, listed in the Appendix, was used to compute c for different values of w. The result is shown in Table 4.1, along with the comparison for the value of c obtained in [2], $c' = \sqrt[6]{7+\sqrt{37}}$. The largest value is seemingly obtained when $w = 3$.

w	c_w	$(c_w - c')/c'$
3	1.535365	0.022%
4	1.535084	0.004%
5	1.535097	0.004%
6	1.535099	0.005%
7	1.535098	0.005%
8	1.535098	0.005%
9	1.535098	0.005%

Table 4.1: c_w for $3 \leq w \leq 9$.

Theorem 4.1. *The maximal number of perfect matchings in planar graphs with n vertices is bound from below by $\Omega(1.535365^n)$.*

4.4 Conclusion

The twisted cylinder is a very useful tool for two reasons. First, it is relatively dense – except for the first few and last few vertices, all vertices have a degree of 6. Second, it is easy to extend to arbitrary number of vertices, and makes a nice fit for the technique of transfer matrix. We expect the it to be useful in other counting problems as well.

Bibliography

[1] Helmut Alt, Ulrich Fuchs, and Klaus Kriegel. On the number of simple cycles in planar graphs. *Combinatorics, Probability and Computing*, 8(05):397–405, 1999. 2, 3

[2] Michael Biro. Planar graphs with many perfect matchings and forests. In *CCCG*, 2014. 2, 22, 23

[3] Sudip Biswas, Stephane Durocher, Debajyoti Mondal, and Rahnuma Islam Nishat. Hamiltonian paths and cycles in planar graphs. In *International Conference on Combinatorial Optimization and Applications*, pages 83–94. Springer, 2012. 2, 19, 20

[4] Kevin Buchin, Christian Knauer, Klaus Kriegel, André Schulz, and Raimund Seidel. On the number of cycles in planar graphs. In *International Computing and Combinatorics Conference*, pages 97–107. Springer, 2007. 2, 3, 5, 16, 17, 19, 21, 22

[5] Kevin Buchin and André Schulz. On the number of spanning trees a planar graph can have. In *European Symposium on Algorithms*, pages 110–121. Springer, 2010. 2

[6] Erik Demaine. Simple polygonizations, 2000. 2

[7] Andreas Gemsa, Thomas Pajor, Dorothea Wagner, and Tobias Zündorf. Efficient computation of jogging routes. In *International Symposium on Experimental Algorithms*, pages 272–283. Springer, 2013. 1

[8] Pablo Sala and Sven Dickinson. Contour grouping and abstraction using simple part models. In *European Conference on Computer Vision*, pages 603–616. Springer, 2010. 1

[9] Pablo Sala, Diego Macrini, and Sven Dickinson. Spatiotemporal contour grouping using abstract part models. In *Asian Conference on Computer Vision*, pages 539–552. Springer, 2010. 1

Appendix A

Program Listing

The following Python program is used in Chapter 4 to compute lower bounds on the number of perfect matchings. Its requires Python 3 and numpy, and is also available at https://github.com/xiaq/masters-thesis.

```
import numpy as np
from numpy.linalg import eigvals, norm

def make_transfer(w):
    nstates = 2 ** (w+1)
    transfer = np.zeros([nstates, nstates], np.int)

    for i in range(nstates):
        if (i & 0b1) == 1:
            # Vertex 0 has been matched. Simply drop it.
            js = {i >> 1}
        else:
            # Match with either vertex 1, w, or w+1.
            js = {((i | (1<<m)) >> 1) for m in [1,w] if not i & (1<<m)}
            js.add((i >> 1) | (1 << w))
        for j in js:
            transfer[i][j] = 1
    return transfer

def main():
    old_c = (7 + 37**(1/2)) ** (1/6)

    for w in range(3, 10):
        transfer = make_transfer(w)
        evs = eigvals(transfer)
        c = np.max(np.abs(evs))
        print(r'{} & {:.6f} & {:.3f}\% \\ \hline'.format(
                w,   c,       (c - old_c) / old_c * 100))

if __name__ == '__main__':
    main()
```