

## Locally recoverable codes from planar graphs

Research Article

Kathryn Haymaker, Justin O’Pella

**Abstract:** In this paper we apply Kadhe and Calderbank’s definition of LRCs from convex polyhedra and planar graphs [4] to analyze the codes resulting from 3-connected regular and almost regular planar graphs. The resulting edge codes are locally recoverable with availability two. We prove that the minimum distance of planar graph LRCs is equal to the girth of the graph, and we also establish a new bound on the rate of planar graph edge codes. Constructions of regular and almost regular planar graphs are given, and their associated code parameters are determined. In certain cases, the code families meet the rate bound.

**2010 MSC:** 94B05, 94B25, 94B65

**Keywords:** Error-correction, Local recovery, Planar graphs, Availability, Rate bound

## 1. Introduction

In the classical view of error-correcting codes (ECCs), the central question involves recovering an original transmitted message from an entire received word, even in the presence of errors or erasures. The recent rise of distributed storage applications has inspired research on the *local* erasure-correcting capabilities of ECCs. A code  $C$  is a *locally recoverable code (LRC)* with *locality*  $r$  if for all codewords  $\mathbf{c} \in C$ , any erased symbol of  $\mathbf{c}$  can be recovered by accessing at most  $r$  other symbols from  $\mathbf{c}$ . In [2], Gopalan, et al. describe the need for efficient erasure coding with small locality to address node failure in distributed storage networks.

A code  $C$  has availability  $t$  if for any codeword position  $i$ , there are  $t$  disjoint recovery sets of sizes  $r_1, r_2, \dots, r_t$ , respectively for position  $i$ . Kadhe and Calderbank present a construction of LRCs with availability two from convex polyhedra, specifically demonstrating the construction with the five examples of the platonic solids. The authors prove that LRCs from convex polyhedra have length and dimension given by the number of edges and the number of faces minus one, respectively, of the polyhedra, and they provide the weight distribution of the codes from the platonic solids [4]. The pre-print [5] is an

---

*Kathryn Haymaker (Corresponding Author); Department of Mathematics and Statistics, Villanova University, United States (email: [kathryn.haymaker@villanova.edu](mailto:kathryn.haymaker@villanova.edu)).*

*Justin O’Pella; Thomas Jefferson University, United States (email: [justin.opella@jefferson.edu](mailto:justin.opella@jefferson.edu)).*

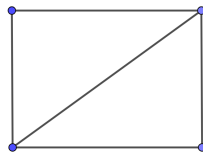
extended version of the conference paper [4] and also contains results on the rate of optimal binary LRCs with small availability and locality. We direct the reader to [5] and the references therein for additional background on binary LRCs.

The goal of this paper is to build upon the results in [4] on LRCs generated by convex polyhedra to classify and construct infinite families of LRCs generated by simple, 3-connected planar graphs. We prove a relationship between the girth of planar graphs and the minimum distance  $d$  of the code. We also prove a bound on the parameters  $n$ ,  $k$ , and  $d$  of an LRC generated by a planar graph using classical facts about planar graphs. In many cases, the bound is tighter than previously discovered bounds for LRCs with availability two. We discuss how constructions of  $j$ -regular planar graphs ( $j = 3, 4, 5$ ) yield planar graph LRCs with recovery set size  $j - 1$ . Finally, we present families of almost regular planar graphs. A graph is a  $(j, j + 1)$  *almost regular graph* if its degree sequence contains only  $j$  and  $j + 1$ , for some positive integer  $j$ . We determine the code parameters from these graphs and classify the types of graphs that yield codes meeting the rate bound.

## 2. Preliminaries

We begin with the necessary definitions and notation. A *planar graph* is a graph that can be drawn in the plane with no edges crossing. Let  $V_i$  represent the number of vertices with degree  $i$ . The maximum degree of a graph is denoted by  $\Delta$ .

A *wheel graph* is a graph that contains a cycle where every vertex on the cycle is connected to a universal vertex (see the black edges of Figure 8). The *girth* of a graph is the minimum length of a cycle contained in the graph. The *degree of a face  $f$*  of a planar graph is the number of edges bordering  $f$ . Figure 1 shows an almost regular  $(2, 3)$  graph, with three faces. The outer or infinite face is degree four. The other two faces are each of degree three. The girth of the graph is three and  $\Delta = 3$ . A *3-connected graph* is a graph in which the removal of any collection of two or fewer vertices does not disconnect the graph.



**Figure 1.** A planar type  $(2, 3)$  graph with  $V_2 = 2$ ,  $V_3 = 2$ . This graph is not 3-connected, since the deletion of the two vertices of degree 3 disconnects the graph.

The following bound on the rate of LRCs with availability two is proven in [8] for sequential recovery codes and in [4] for non-sequential recovery codes.

The rate  $R$  of a locally recoverable code with availability two and recovery set sizes at most  $r$  satisfies:

$$R \leq \frac{r}{r + 2}. \tag{1}$$

In [4], Kadhe and Calderbank introduced codes generated by convex polyhedra.

**Definition 2.1** (Def. 2, [4]). Consider a convex polyhedron  $\Gamma$  with  $v$  vertices,  $e$  edges, and  $f$  faces. Fix an arbitrary labeling of its edges 1 through  $e$ . Let  $\mathbb{C}$  be a subset of  $\mathbb{F}_2^e$  such that for a vector  $\mathbf{c} \in \mathbb{C}$ , every collection of entries corresponding to edges that meet at a vertex sum to zero over  $\mathbb{F}_2$ . We say that the code is generated by  $\Gamma$ , and denote it as  $\mathbb{C}(\Gamma)$ .

Kadhe and Calderbank use properties of convex polyhedra to prove that the dimension of a code  $\mathbb{C}(\Gamma)$  is  $f - 1$ .

**Lemma 2.2** (Kadhe and Calderbank, [4]). *For a convex polyhedron  $\Gamma$  with  $v$  vertices,  $e$  edges and  $f$  faces, the code  $\mathbb{C}(\Gamma)$  generated by  $\Gamma$  is an  $[e, f - 1]$  code.*

In this paper we restrict our work to 3-connected planar graphs because there is a correspondence between the embeddings of convex polyhedra and 3-connected planar graphs [10]. Moreover, it is important to guarantee that every edge is involved in a cycle that encloses a face so that we can unambiguously quantify the degree of a face. Throughout, we will use the notion of an LRC generated by a planar graph introduced in [4], with the additional understanding that all planar graphs considered in this paper are 3-connected.

**Definition 2.3** (Def. 7, [4]). *Consider a planar graph  $\Gamma$  with  $v$  vertices and  $e$  edges. Fix an arbitrary labeling of its edges from 1 through  $e$ . Let  $\mathbb{C}$  be a subset of  $\mathbb{F}_2^e$  such that for every vector  $\mathbf{c} \in \mathbb{C}$ , the entries of  $\mathbf{c}$  corresponding to edges that meet at a vertex sum to zero over  $\mathbb{F}_2$ . We say that the code  $\mathbb{C}$  is generated by  $\Gamma$ , and denote it as  $\mathbb{C}(\Gamma)$ .*

We call a code generated by a planar graph in this way an *edge code*.

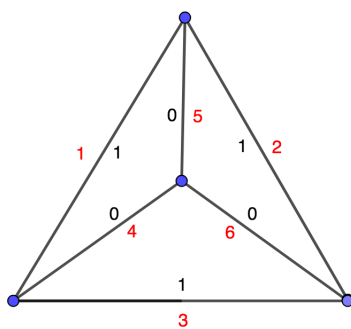
**Remark 2.4.** *Edge codes have availability  $t = 2$  since every edge is incident with two vertices that provide parity checks on the edge. The sizes of the recovery sets for edge  $\{v_i, v_j\}$  are  $\deg(v_i) - 1$  and  $\deg(v_j) - 1$ , respectively. That is,  $r \leq \Delta - 1$ .*

Throughout the paper we make use of Euler’s formula for planar graphs: a planar graph with  $v$  vertices,  $e$  edges, and  $f$  faces satisfies  $v - e + f = 2$ .

In the proof of Lemma 2.2, the fact that  $\Gamma$  is a convex polyhedron is used to invoke Euler’s formula. Since Euler’s formula also applies to 3-connected planar graphs, the proof of Lemma 2.2 implies the following corollary.

**Corollary 2.5.** *For a 3-connected planar graph  $\Gamma$  with  $v$  vertices,  $e$  edges and  $f$  faces, the code  $\mathbb{C}(\Gamma)$  generated by  $\Gamma$  is an  $[e, f - 1]$  code.*

**Example 2.6.** *A tetrahedron embedded into the plane contains 4 vertices of degree 3, 6 edges, and 4 faces. The code generated by the tetrahedron is a  $[6, 3]$  code with locality  $r = 2$  and availability  $t = 2$ . The tetrahedron embedded in the plane and one example codeword are shown in Figure 2. The examples constructed in [4] include LRCs from the Platonic solids. Parameters of these codes are summarized in Table 1.*



**Figure 2.** Tetrahedron with example codeword bits in black, and index positions in red. The codeword shown is  $(1, 1, 1, 0, 0, 0)$ .

**Table 1.** Parameters of codes associated with the Platonic solids [4].

Platonic Solid	$(n, k)$	$r$	$t$
Tetrahedron	(6, 3)	2	2
Cube	(12, 5)	2	2
Octahedron	(12, 7)	3	2
Dodecahedron	(30, 11)	2	2
Icosahedron	(30, 19)	4	2

### 3. Parameters of edge codes

We begin by proving that the minimum distance of an edge code is equal to the girth of the planar graph, which is the same as the smallest degree of a face.

**Proposition 3.1.** *The minimum distance of an edge code generated by a planar graph is equal to the smallest degree of a face.*

**Proof.** Let  $C$  represent the set of edges in a planar graph  $G$  corresponding to a smallest face of  $G$ . Then the indicator vector of  $C$ ,  $\mathbf{x}_C$ , is a codeword, since every vertex in the cycle  $C$  has exactly two incident edges labeled 1, while every vertex outside of the cycle has all incident edges labeled 0.

Next we show that a minimum-weight codeword must contain a cycle from  $G$ . Suppose that  $\mathbf{x}$  is a nonzero minimum-weight codeword in a code generated by the edges of a 3-connected planar graph  $G$ . Let  $G_{\mathbf{x}}$  be the subgraph induced by the edges that correspond to the support of  $\mathbf{x}$ . Consider an edge  $e$  corresponding to a position in the support of the codeword  $\mathbf{x}$ .

Seeking a contradiction, suppose that the connected component of  $G_{\mathbf{x}}$  containing  $e$  is a tree. Then there must be at least two leaf vertices, each of which has exactly one neighboring edge labeled 1, and therefore these leaf vertices represent unsatisfied parity checks. This contradicts the assumption that  $\mathbf{x}$  is a codeword.  $\square$

Next we use facts about planar graphs to prove the following rate bound.

**Theorem 3.2.** *An  $[n, k, d]$  edge code generated by a 3-connected planar graph  $G$  with  $v \geq 3$  satisfies the following bound:*

$$\frac{k}{n} \leq \frac{2}{d} - \frac{1}{n}. \tag{2}$$

**Proof.** Note that for any simple planar graph, the sum of the degrees of all faces is equal to twice the number of edges. Let  $d$  be the smallest degree of a face in  $G$ . More formally,  $\sum_i f_i = 2e$ , and therefore:

$$3f \leq \sum_i f_i = 2e.$$

Since  $d$  is the smallest degree of a face in  $G$ ,  $df$  can replace  $3f$  in the inequality as follows:

$$df \leq 2e.$$

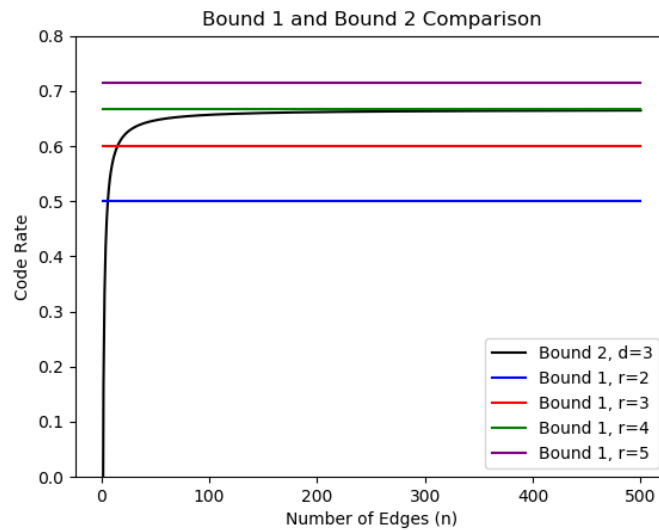
Therefore we obtain

$$\frac{f-1}{e} \leq \frac{2}{d} - \frac{1}{e}.$$

Applying Corollary 2.5, we have established that  $R \leq \frac{2}{d} - \frac{1}{n}$ .  $\square$

**Remark 3.3.** A graph in which every face has degree  $d$  would result in equalities in the steps of the proof, producing a code whose rate meets the bound. In Section 5.1, we use this fact to classify almost regular planar graphs whose corresponding codes meet the rate bound.

Notice that the smallest degree of a face in  $G$  is also the girth of the graph, so Bound 2 can also be stated in terms of the girth. The cases in this paper include many codes with  $d = 3$ . Figure 3 shows a comparison of Bound 1 and Bound 2 for different recovery set sizes and  $d = 3$ . For cases where  $r \geq 4$ , Bound 2 is tighter than Bound 1. When  $r = 3$ , Bound 2 is tighter than Bound 1 for  $n < 15$ . In the case where  $r = 2$ , Bound 2 is tighter than Bound 1 for  $n < 6$ . For codes generated by planar graphs with  $d > 3$ , Bound 2 is tighter than Bound 1.



**Figure 3.** Bound comparison for edge codes: Bound 1 and Bound 2, for  $d = 3$ .

We will repeatedly make use of the following well-known fact about planar graphs, so we include a brief proof.

**Fact 1.** The average degree of a planar graph is strictly less than 6.

**Proof.** Considering that  $3f \leq 2e$  for a planar graph and substituting this into Euler’s formula gives:

$$\sum_i d_i \leq 6v - 12.$$

Therefore the average vertex degree is at most  $6 - \frac{12}{v}$ . □

#### 4. $j$ -regular planar graphs

Regular planar graphs of degree 3, 4, or 5 can yield LRCs with availability two, where every recovery set has the same size,  $\Delta - 1$ . Relaxing the regularity condition can still yield codes that meet Bound 2, so we also explore the expanded class of almost regular planar graph constructions in Section 5.

Using the website [7], with information from [6], we determine the parameters of LRCs from small 3-regular planar graphs of girth at least 5 in Table 2. The graph with  $v = 20$  is the dodecahedron edge code that was presented in [4, 5].

**Table 2.** 3-regular planar graphs of girth at least 5 and their edge code parameters. All codes have minimum distance at least 5. Bound 3 assumes  $d = 5$ . Rate and Bound 3 rounded to 4 decimal places. There are three non-isomorphic graphs with 28 vertices.

$v$	$e$	$f$	Code parameters	Rate	Bound 3
20	30	12	[30, 11]	.3667	.3667
24	36	14	[36, 13]	.3611	.3722
26	39	15	[39, 14]	.3590	.3744
28	42	16	[42, 15]	.3591	.3762

We now summarize some constructions of infinite families of 3, 4, and 5-regular planar graphs and give their corresponding edge code parameters. There are no  $j$ -regular planar graphs for  $j > 5$  by Fact 1. We concentrate on girth 3, 4, and 5 to obtain LRCs with minimum distance at most 5. Bound 2 shows that larger minimum distance results in low code rates ( $R < \frac{1}{3}$ ), and therefore we do not consider  $d > 5$  in this paper.

The proof of Bound 2 shows that a planar graph with all faces of degree  $g$  for a planar graph of girth  $g$  yields an LRC with rate meeting Bound 2. The graphs in which every face has the same degree are called triangulations ( $g = 3$ ), quadrangulations ( $g = 4$ ), and pentangulations ( $g = 5$ ), and the  $j$ -regular versions of these are precisely the platonic solids (see [4]).

**Case 1: 3-regular planar graphs**

Define the operation of *splitting a face* of a planar graph as adding a vertex to each of two distinct edges of the face and joining the two new vertices by an edge. Steinetz (edited by Rademacher) showed that 3-regular planar graphs can be generated by “adding edges” to the tetrahedron [10], including by splitting faces. This process is also called *adding handles* or *adding ears*. The face splitting operation increases the number of vertices in the graph by two, increases the number of edges by three, and increases the number of faces by one. The degree of all vertices remains three. Therefore, starting with the tetrahedron with  $v = 4, e = 6, f = 4$ , the splitting process at iteration  $i$  results in a graph with  $v = 4 + 2i, e = 6 + 3i, f = 4 + i$ . The resulting family of edge codes has parameters:  $[6 + 3i, 3 + i, 3]$ , with rate  $R_i = \frac{3+i}{6+3i}$ , which approaches  $\frac{1}{3}$  from above as  $i$  increases. The rate can be improved by adding edges between existing vertices, but this process destroys the 3-regularity of the graph, which would also negatively impact the small locality of the resulting LRC. Face splitting is the only edge-addition operation that preserves 3-regularity, and the cost is that every operation that splits a face adds at least one additional non-triangular face to the graph.

**Case 2: 4-regular planar graphs**

Broersma, et al. detailed a process of generating all 3-connected 4-regular planar graphs from the octahedron [1]. Inserting a new triangle into an existing triangular face of the octahedron is one method that results in an infinite family of 4-regular planar graphs. See the yellow edges being inserted into the red triangle in Figure 15 for an example. On iteration  $i$  of this process, the resulting graph has  $v = 6 + 3i, e = 12 + 6i, f = 8 + 3i$ . The resulting family of edge codes has parameters  $[12 + 6i, 7 + 3i, 3]$ , with rate  $R_i = \frac{7+3i}{12+6i}$ , which approaches  $\frac{1}{2}$  from above as  $i$  increases. As in Case 1, adding edges could increase the rate at the cost of the graph regularity and the small locality of the edge code.

**Case 3: 5-regular planar graphs**

Families of 5-regular simple planar graphs are generated in [3]. An infinite family  $D_1, D_2, \dots$  is given in [3] where  $D_1$  is the icosahedron planar embedding.  $D_2$  is formed by splitting an edge of  $D_1$ , duplicating the graph and gluing the split edges back together. See [3], page 420 for the first few examples in this infinite family. The graph  $D_i$  has  $v = 12i, e = 30i, f = 18i + 2$ . The resulting edge code family has parameters  $[30i, 18i + 1, 3]$  and rate  $R_i = \frac{18i+1}{30i}$ .

Section 5.3 contains another approach to constructing an infinite family of 5-regular planar graphs.

Dropping the strict regularity condition and considering almost regular graphs in the next section in some cases yields improved code rates and more flexible constructions, without a large difference in the local erasure correction capabilities of the codes.

## 5. Almost regular type $(j, j + 1)$ planar graphs

Almost regular type  $(j, j + 1)$  planar graphs will be considered for  $j \in \{3, 4, 5\}$ . Values of  $j \in \{1, 2\}$  are not considered due to the resulting small locality. Since the average degree of a planar graph is strictly less than 6 (Fact 1),  $j$  cannot be greater than 5.

A type  $(j, j + 1)$  graph has  $V_j + V_{j+1}$  vertices. The number of edges  $e$  can be calculated as follows:

$$e = \frac{jV_j + (j + 1)V_{j+1}}{2}. \tag{3}$$

We use Euler’s formula to find the number of faces  $f$  and the code rate for all codes in this section.

### 5.1. Almost regular graphs attaining the bound

First we consider graphs that yield codes that meet Bound 2, with minimum distance  $d = 3$ . That is, we assume the graphs are triangulations since every face must have degree 3 to meet Bound 2 with  $d = 3$ .

**Theorem 5.1.** *There are no infinite families of planar graphs with vertices of degree 3, 4, or 5 that are triangulations. The finite list of such graphs is given in Table 3.*

**Proof.** The number of vertices  $V$  in an almost regular graph with only vertices of degree 3, 4, and 5 can be restated as  $V = V_3 + V_4 + V_5$ . The number of edges is  $e = \frac{3}{2}V_3 + 2V_4 + \frac{5}{2}V_5$ . Note a graph that generates a code with  $d = 3$  will attain Bound 3 when every face of the graph is degree 3. More formally, the bound is attained if  $3f = 2e$ . Substituting this information into Euler’s Formula allows for an algebraic solution.

$$\begin{aligned} v - e + f &= 2 \\ 3v - 3e + 3f &= 6 \\ 3v - e &= 6 \\ 3V_3 + 3V_4 + 3V_5 - \frac{3}{2}V_3 - 2V_4 - \frac{5}{2}V_5 &= 6 \\ \frac{3}{2}V_3 + V_4 + \frac{1}{2}V_5 &= 6. \end{aligned}$$

There are a finite number of combinations of non-negative integers that satisfy the equation above. In [9], Schmeichel and Hakimi determine which almost regular degree sequences are planar graphical. Table 3 contains the list of 11 combinations of  $V_3, V_4, V_5$  that are planar graphical. □

Notice that some of the cases listed in Table 3 are regular graphs, some are almost regular graphs, and some contain other combinations of vertex degrees (such as only degree 3 and degree 5 vertices, for example).

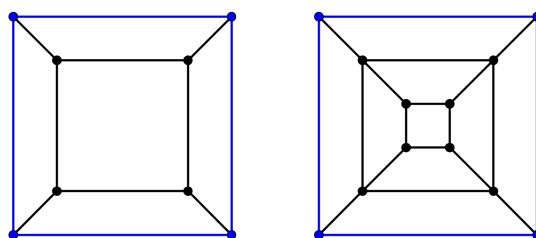
Figure 6 shows all type  $(4, 5)$  planar graphs whose corresponding codes are rate optimal according to Bound 2.

Next we consider the cases  $d = 4$  and  $d = 5$ . We first present two iterative constructions of almost regular  $(3, 4)$  planar graphs whose codes meet Bound 2. Both constructions begin with one of the platonic solids.

**Table 3.** Graphs that generate codes attaining Bound 3. Code rate rounded to 4 decimal places. Codes possess minimum distance  $d = 3$ .

$V_3$	$V_4$	$V_5$	Code Rate
0	0	12	.6333
0	2	8	.6250
0	3	6	.6190
0	4	4	.6111
0	5	2	.6000
0	6	0	.5833
1	3	3	.6000
2	0	6	.6111
2	2	2	.5833
2	3	0	.5556
4	0	0	.5000

**Construction 1.** The starting point of this construction is the Cube graph—a 3-regular planar graph with faces of degree 4, shown on the left in Figure 4. The first iteration in the construction is to insert a copy of the black edges and vertices into any face in the graph that has all vertices of degree 3. In Figure 4, the new vertices and edges are inserted into the center face. There will be at least one face incident with all degree-3 vertices on every iteration because the inner-most face being inserted always has this property. On iteration  $i$ , the graph has  $V_3 = 8, V_4 = 4i$ , with  $6 + 4i$  faces, and all faces of degree 4. The code parameters at the  $i^{\text{th}}$  iteration are  $[12 + 8i, 5 + 4i, 4]$ , and the rate meets Bound 2.



**Figure 4.** The Cube and the first iteration of Construction 1.

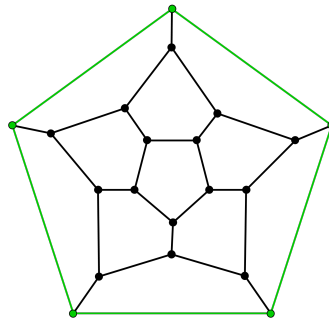
**Construction 2.** This construction begins with the planar embedding of the Dodecahedron—a 3-regular planar graph with faces of degree 5, shown in Figure 5. The first iteration of the construction takes a copy of the edges and vertices in black in the figure, and inserts them into any face that is incident with only vertices of degree 3. Like in Construction 1, there will always be a face incident with vertices of degree 3, since there are several such faces being inserted at each iteration.

At iteration  $i$ , the graph has  $V_3 = 20 + 10i, V_4 = 5i$ , with  $12 + 10i$  faces, each of degree 5. The code at iteration  $i$  has parameters  $[30 + 25i, 11 + 10i, 5]$ , and the rate meets Bound 2.

The same argument as Theorem 5.1 can be applied with  $4f = 2e$  and  $5f = 2e$  to gain insight into the possible parameters of rate optimal codes with  $d = 4$  and  $d = 5$ , respectively. The resulting equations are  $V_3 - V_5 = 8$  for  $d = 4$  and  $\frac{V_3}{2} - V_4 - \frac{5V_5}{2} = 10$  for  $d = 5$ .

For  $d = 4$ , Construction 1 gives an infinite family of  $(3, 4)$  almost regular planar graphs in which every face has degree 4 and whose resulting code rates meet Bound 2. There are no type  $(4, 5)$  almost regular graphs satisfying the  $d = 4$  equation. Furthermore, the resulting equation gives a necessary condition for a rate optimal type  $(3, 5)$  graph to exist.





**Figure 5.** The Dodecahedron with black edges and vertices denoting the portion of the graph that should be copied and inserted into a face bordered by vertices of degree 3 for each iteration of Construction 2.

The results for  $d = 5$  do not provide a finite list of cases that meet the bound but, rather, show necessary conditions for a rate optimal code to exist. These observations are in Table 4.

**Table 4.** Necessary existence conditions for number of vertices needed for a rate optimal code to exist with  $d = 5$ .

Graph Degrees (Type)	Necessary Conditions
(3, 4)	$V_3 \geq 22, V_4 \geq 1$
(4, 5)	Does not exist
(3, 5)	$V_3 \geq 30, V_5 \geq 2$
(3, 4, 5)	$\frac{V_3}{2} - V_4 - \frac{5V_5}{2} = 10$

### 5.2. Type (3, 4) planar graphs with girth 3

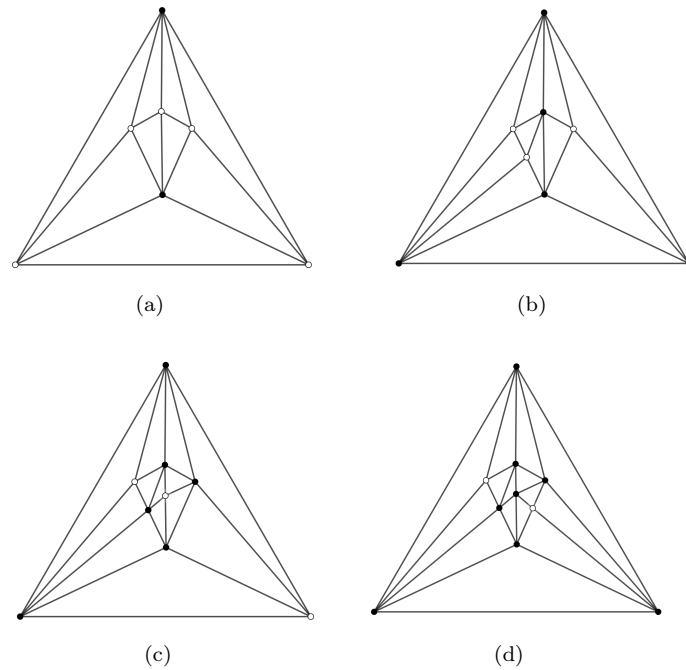
Since Theorem 5.1 establishes that there are no infinite families of (3, 4) almost regular planar graphs that are triangulations, in this section we present alternative graph constructions which contain some faces of degree four.

**Algorithm 5.2.** This algorithm subdivides the edges of a triangle embedded into a plane with new vertices to form a planar graph with  $V_2 = 2, V_3 = 2, V_4 = k,$  and  $f = k + 3$  for any non-negative integer  $k$ . See Figure 7 for an example.

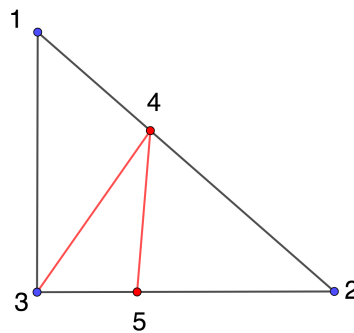
Begin with a planar 2-regular graph with 3 vertices labeled 1, 2, and 3.

1. Subdivide edge  $v_1v_2$  by inserting vertex  $i = 4$ .
2. Insert edge  $v_3v_4$ .
3. If this is the desired number of  $V_2, V_3$  and  $V_4$ , stop. Else, continue to step 4.
4. Set  $i = i + 1$ .
5. Subdivide edge  $v_{i-2}v_2$  by inserting vertex  $i$ .
6. Insert edge  $v_{i-1}v_i$ . Return to step 3.

In steps 1-3, an initial graph with  $V_2 = 2, V_3 = 2, V_4 = 0,$  and  $f = 3$  is constructed. For each iteration of steps 3-6, a vertex of degree 4 is added to the degree sequence and another face is bounded.



**Figure 6.** All rate optimal type (4, 5) planar graphs. Degree 5 vertices in black. (a) [5, 5, 4, 4, 4, 4, 4] (b) [5, 5, 5, 5, 4, 4, 4, 4]; (c) [5, 5, 5, 5, 5, 5, 4, 4, 4]; and, (d) [5, 5, 5, 5, 5, 5, 5, 4, 4].

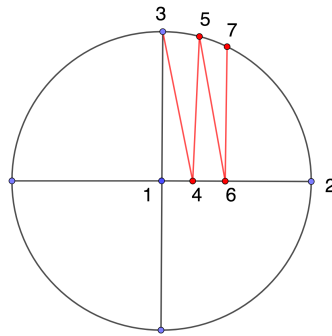


**Figure 7.** A graph with  $V_2 = 2$ ,  $V_3 = 2$ ,  $V_4 = k = 1$ , and  $f = k + 3 = 4$ . Vertices and edges added through the steps of Algorithm 5.2 in red.

Algorithm 5.2 can be generalized to cases where a triangle is present in a larger graph so the initial degree of vertices 1, 2, and 3 depends on the total edges incident with those vertices.

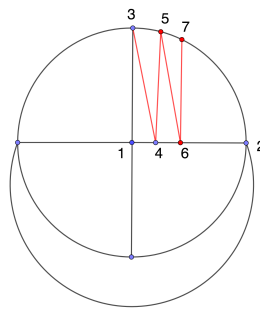
**Construction 3.** This construction (Figure 8) generates an infinite family of type (3, 4) planar graphs with  $V_3 = 4$ ,  $V_4 = k$  and  $f = k + 4$  for all  $k \in \mathbb{Z}^+$ . Begin with a wheel graph with 5 vertices. Select a triangle in the graph and implement Algorithm 5.2. Note that the universal vertex cannot be labeled vertex 3.

The rate of the code in Construction 3 simplifies to  $\frac{1}{2}$  for any value of  $k$ , thus any combination of  $V_3$  and  $V_4$  that can be attained in Construction 3 yields the same code rate.



**Figure 8.** A graph with  $V_3 = 4$ ,  $V_4 = k = 5$ , and  $f = k + 4 = 9$ . Vertices and edges added through the steps of Algorithm 5.2 in red.

**Construction 4.** This construction generates an infinite family of type  $(3, 4)$  planar graphs with  $V_3 = 2$ ,  $V_4 = k$  and  $f = k + 3$  for  $k \in \mathbb{Z}^+$ ,  $k \geq 3$ . Consider the type  $(3, 4)$  graph generated by Construction 3. Insert an edge between vertex 2 and another vertex of degree 3 on the original “wheel” that is not already neighbors with vertex 2 (see Figure 9). Return to step 3 of Algorithm 5.2.



**Figure 9.** A graph with  $V_3 = 2$ ,  $V_4 = k = 7$ , and  $f = k + 3 = 10$ .

As degree 4 vertices are added in Construction 4, the code rate decreases. The degree of the infinite face increases and since 3 is the minimum number of edges needed to form a closed face, it is an expensive face with respect to edges. Similarly, the degree of the face with vertices 1 and 2 that is incident with the face where Algorithm 5.2 takes place increases and becomes an expensive face with respect to edges. As the number of degree 4 vertices becomes larger, the constants in the code rate become less significant and the code rate approaches  $\frac{1}{2}$ .

**Algorithm 5.3.** Algorithm 5.3 subdivides the edges of a rectangle with new vertices to form a new planar graph with  $V_2 = 4$ ,  $V_3 = 2k$ , and  $f = k + 2$  for any  $k \in \mathbb{Z}^+$ . See Figure 10.

Begin with a planar 2-regular graph with 4 vertices. Label the vertices 1, 2, 3 and 4 with edges  $v_1v_2$ ,  $v_1v_3$ ,  $v_2v_4$  and  $v_3v_4$ .

1. Subdivide edge  $v_1v_3$  and  $v_2v_4$  by inserting vertex  $i = 5$  and  $i + 1 = 6$ , respectively.
2. Insert edge  $v_5v_6$ .
3. If this is the desired number of  $V_2$  and  $V_3$ , stop. Else, continue to step 4.
4. Set  $i = i + 2$ .

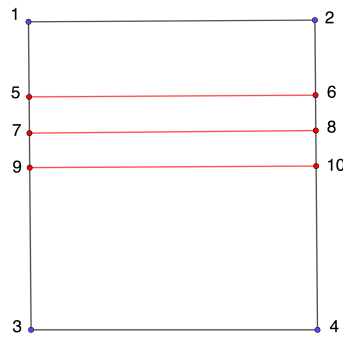


Figure 10. A graph with  $V_2 = 4$ ,  $V_3 = 2k = 2(3) = 6$ , and  $f = 3 + 2 = 5$ .

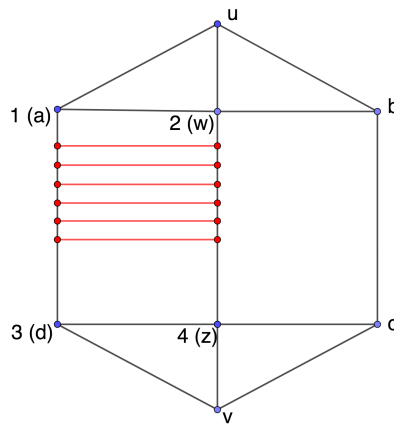


Figure 11. A graph with  $V_3 = 2(9) = 18$ ,  $V_4 = 2$ , and  $f = 9 + 4 = 13$

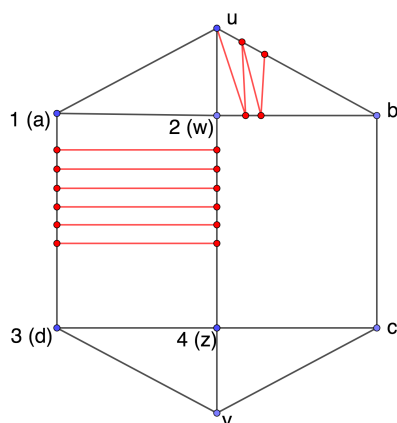
5. Subdivide edges  $v_{i-2}v_3$  and  $v_{i-1}v_4$  with vertex  $i$  and  $i + 1$ , respectively.
6. Insert edge  $v_iv_{i+1}$ . Return to step 3.

Algorithm 5.3 can be applied to cases where a rectangle is present in a larger graph so the initial degree of vertices 1, 2, 3 and 4 depends on the total edges incident with those vertices.

**Construction 5.** This construction generates an infinite family of type (3, 4) planar graph with  $v$  vertices and  $V_3 = 2l, V_4 = 2$  and  $f = l + 4$  for any  $l \in \mathbb{Z}^+, l \geq 3$ . See Figure 11.

- Embed a rectangle into the plane with vertices  $a, b, c$ , and  $d$  and edges  $v_av_b, v_av_d, v_bv_c, v_cv_d$ .
- Add two vertices,  $u$  and  $v$ , to the infinite face and edges  $v_av_u, v_bv_u, v_cv_v, v_dv_d$ .
- Subdivide  $v_av_b$  and  $v_cv_d$  with  $w$  and  $z$ , respectively, and add edges  $v_uv_w, v_vv_z$ , and  $v_wv_z$ .
- Set  $v_a = 1, v_w = 2, v_d = 3$ , and  $v_z = 4$  and implement Algorithm 5.3 throughout.

Construction 5 utilizes Algorithm 5.3 to add edges, vertices, and faces to a graph. Each iteration of Algorithm 5.3 adds a face of degree 4. The rate of the code generated by this graph decreases as more degree 3 vertices are added since the degree of each new face is 4. As the number of degree 3 vertices becomes larger, the constants in the code rate become less significant and the code rate approaches  $\frac{1}{3}$ .



**Figure 12.** A graph with  $V_3 = 2(9) = 18$ ,  $V_4 = m = 6$ , and  $f = 9 + 6 + 2 = 17$

**Construction 6.** This construction generates an infinite family of type (3, 4) planar graph with  $V_3 = 2l$  and  $f = l + 4$  for any  $l \in \mathbb{Z}^+, l \geq 3$  can have  $V_4 = m$  for any  $m \in \mathbb{Z}^+, m \geq 2$ . The number of faces can be restated as  $f = l + m + 2$ . See Figure 12.

- Construct a graph using Construction 5.
- Select a triangle and implement Algorithm 5.2. Note: A vertex of degree 4 cannot be labeled 3 in Algorithm 5.2.

The rate of the code generated by the graph in Construction 6 increases as degree 4 vertices are added before reaching a limit that does not attain the bound shown earlier in the paper. While fixing the number of degree 3 vertices, as the number of degree 4 vertices added becomes larger, the code rate approaches  $\frac{1}{2}$ . For larger values of  $l$ , the code rate approaches  $\frac{1}{2}$  at a slower rate since the constants have a larger impact in the calculation.

Taking Constructions 3, 4, 5, and 6 it is possible to construct a type (3, 4) planar graph with  $V_3 = 2a$  and  $V_4 = b$  such that  $V_3 + V_4 \geq 5$  for any  $a, b \in \mathbb{Z}^+$ . Table 5 includes a summary of code rates for these constructions. All of the code families in Table 5 have minimum distance 3.

The code rates for type (3, 4) graphs constructed in Section 5.2 are shown in Figures 13 and 14.

### 5.3. Type (4, 5) planar graphs with girth 3

Next we construct infinite families of type (4, 5) almost regular graphs. The following operations will be used sequentially in the construction and are demonstrated in Figure 15.

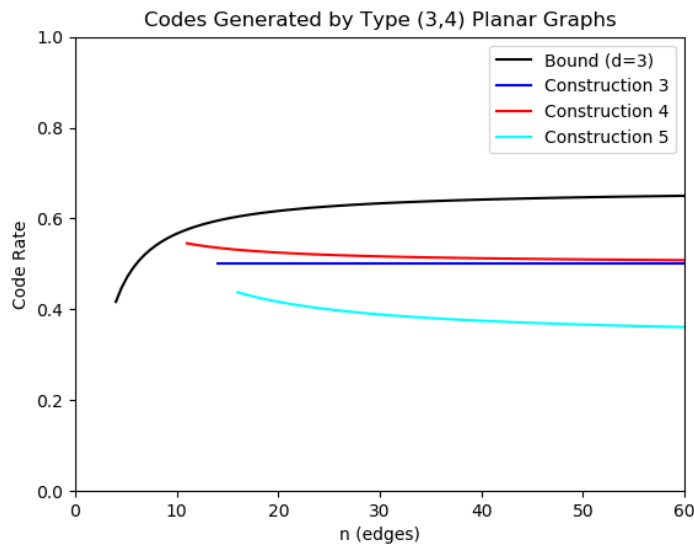
**Definition 5.4 (Operation A).** Let Operation A be defined as an operation performed on a planar graph with a face  $f$ , where the degree of  $f$  is 3 and each  $e \in f$  is incident with another face of degree 3 such that every pair of vertices of  $f$  does not share a common neighbor.

1. Label the vertices of  $f$  as  $v_1, v_2$ , and  $v_3$
2. Subdivide  $v_1v_2, v_1v_3$ , and  $v_2v_3$  with  $v_4, v_5, v_6$ , respectively.
3. Add edges  $v_4v_5, v_4v_6$ , and  $v_5v_6$

**Definition 5.5 (Operation B).** Let Operation B be defined as an operation performed after Operation A has been applied. The labeling of the vertices for Operation B follows from Operation A. Label the vertices adjacent to  $v_1$  and  $v_2, v_1$  and  $v_3$ , and  $v_2$  and  $v_3$  as  $v_7, v_8$ , and  $v_9$ , respectively.

**Table 5.** Performance of Families of Codes Generated by Graph Constructions. The constructions with an (\*) meet rate Bound 2. All codes have availability  $t = 2$ .

Graph Type	Code Rate	Rec. Set Size(s)	Min. dist. $d$
Case 1 (3-reg)	$\frac{3+i}{6+3i}, i \in \mathbb{Z}^+$	2	3
Case 2 (4-reg)	$\frac{7+3i}{12+6i}, i \in \mathbb{Z}^+$	3	3
Case 3 (5-reg)	$\frac{18i+1}{30i}, i \in \mathbb{Z}^+$	4	3
Construction 1* (type (3,4))	$\frac{5+4i}{12+8i}, i \in \mathbb{Z}^+$	{2, 3}	4
Construction 2* (type (3,4))	$\frac{11+10i}{30+25i}, i \in \mathbb{Z}^+$	{2, 3}	5
Construction 3 (type (3,4))	$\frac{k+3}{2(k+3)} = \frac{1}{2}$	{2, 3}	3
Construction 4 (type (3,4))	$\frac{k+2}{2k+3}$ for $k \in \mathbb{Z}^+, k \geq 3$	{2, 3}	3
Construction 5 (type (3,4))	$\frac{l+3}{3l+4}$ for $l \in \mathbb{Z}^+, l \geq 3$	{2, 3}	3
Construction 6 (type (3,4))	$\frac{l+m-1}{3l+2m}$ , for $l, m \in \mathbb{Z}^+, l \geq 3, m \geq 2$	{2, 3}	3
Construction 7 (type (4,5))	$\frac{9p+3m+1}{3(2m+5p)}$ for $m \in \{1, 2\}, p \in \mathbb{Z}^+$	{3, 4}	3
Construction 8* (type (5,6))	$\frac{19+6k}{3(10+3k)}$ for $k \in \mathbb{Z}^+$	{4, 5}	3



**Figure 13.** Code rates generated by type (3, 4) graphs in Constructions 3, 4, and 5 compared to Bound 2.

1. Add edges  $v_4v_7, v_5v_8,$  and  $v_6v_9$ .

**Construction 7.** Consider the graph  $G$  created by embedding the octahedron into the plane so that  $V_4 = 6$  and there are 8 faces.

1. Select any face of degree 3 that is not incident with the infinite face to implement a series of Operations A and B. Perform Operation A followed by Operation B. If this is the desired number of  $V_4$  and  $V_5$ , stop. Else, continue to step 2.
2. Perform Operation A on degree 3 face added in previous iteration of Operation A then perform Operation B.

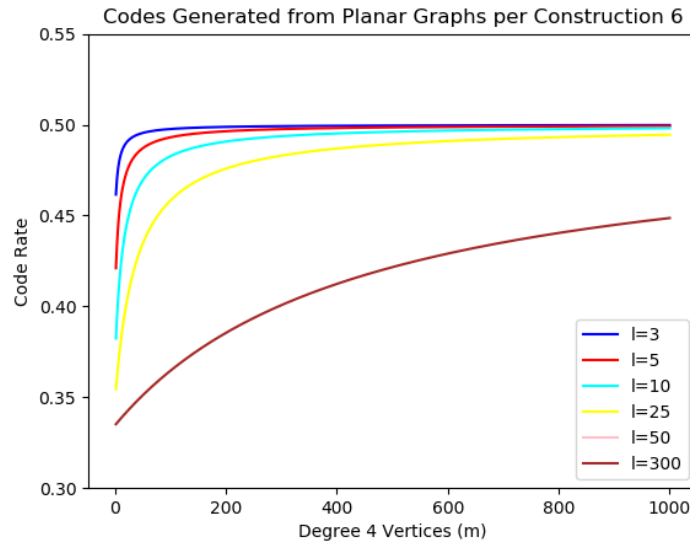


Figure 14. Code Rate Generated by Graph from Construction 6 Compared to Bound 2 for Varying Levels of  $l$ .

3. Perform Operation A on degree 3 face added in previous iteration of Operation A. If this is the desired number of  $V_4$  and  $V_5$ , stop. Else, continue to step 4.
4. Perform Operation A on degree 3 face added in previous iteration of Operation A. If this is the desired number of  $V_4$  and  $V_5$ , stop. Else, continue to step 5.
5. Perform Operation A on degree 3 face added in previous iteration of Operation A then perform Operation B. If this is the desired number of  $V_4$  and  $V_5$ , stop. Else, return to step 2.

The octahedron is a 4-regular graph with 6 vertices. Each application of Operation A increases  $V_4$  by 3. Each application of Operation B decreases  $V_4$  by 6 and increases  $V_5$  by 6. Letting  $a$  and  $b$  represent the number of applications of Operations A and B, respectively,  $V_4 = 6 + 3a - 6b$  and  $V_5 = 6b$ . Applying Operations A and B in construction 5 leads to  $V_4 = 3m$  for  $m \in \{1, 2\}$  and  $V_5 = 6p$  for  $p \in \mathbb{Z}^+$ .

Figure 16 shows the code rate as the number of degree 5 vertices increases. The code rate is highest when  $V_4 = 3$  and  $V_5 = 6$  after step 1 in Construction 7. This is the only instance when every face of the graph has degree 3 and the bound is attained. All future steps involve at least one face of degree 4. While fixing  $m$  at either 1 or 2, as the number of degree 5 vertices are added, the constants in the code rate become less significant and the code rate approaches  $\frac{3}{5}$ .

**Remark 5.6.** It is possible to generate an infinite family of 5-regular planar graphs with  $V_5 = 12i$  for  $i \in \mathbb{Z}^+$  using the steps in Construction 7 by removing the stopping criterion from steps 1, 3-5 and adding an option to stop after step 2. The 5-regular graph generated by this construction produces a code with rate  $\frac{18i+1}{30i}$ , matching the code rate generated by the 5-regular graph is Case 3 of Section 4.

#### 5.4. Type (5, 6) planar graphs with girth 3

The final type of almost regular planar graph to consider is type (5, 6). We construct infinite families of almost regular (5, 6) planar graphs and determine the code parameters of the resulting edge codes. The following construction results in graphs that are triangulations.

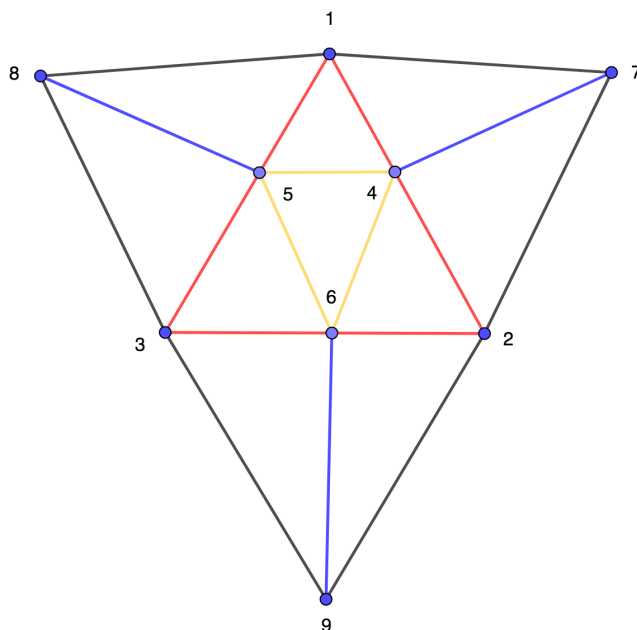


Figure 15. Operations A and B on  $f$ .  $f$  in red. Operation A in gold. Operation B in blue.

**Algorithm 5.7.** For a graph  $G$  with face  $f$  where the degree of  $f$  is 3 and each edge  $e \in f$  is incident with another face of degree 3 such that every pair of vertices in  $f$  do not share a common neighbor, the number of vertices, faces, and edges of a graph  $G$  can be increased by 3, 6, and 9, respectively, for  $i$  iterations, where  $i \in \mathbb{Z}^+$ .

The following steps are shown in Figure 17. Begin with a 2-regular planar graph with 3 vertices  $v_1$ ,  $v_2$ , and  $v_3$ . Add three disconnected vertices to the outside face labeled  $v_4$ ,  $v_5$ , and  $v_6$ . Add the following edges to the graph:  $v_1v_4$ ,  $v_1v_6$ ,  $v_2v_4$ ,  $v_2v_5$ ,  $v_3v_5$ ,  $v_3v_6$ . Note that the graph constructed is in the appropriate form required for the algorithm.

1. Subdivide edges  $v_1v_2$ ,  $v_1v_3$ , and  $v_2v_3$  with new vertices  $v_7$ ,  $v_8$ , and  $v_9$ , respectively, then add edges  $v_7v_8$ ,  $v_7v_9$ , and  $v_8v_9$ .
2. Insert edges  $v_4v_7$ ,  $v_5v_9$ , and  $v_6v_8$ .
3. If this is the desired graph, stop. Else, relabel the vertices as follows: vertices  $v_4$ ,  $v_5$ , and  $v_6$  become unlabeled, and the rest are relabeled  $v_1 \rightarrow v_6$ ,  $v_2 \rightarrow v_4$ ,  $v_3 \rightarrow v_5$ ,  $v_7 \rightarrow v_1$ ,  $v_8 \rightarrow v_3$ ,  $v_9 \rightarrow v_2$ . Return to step 1.

**Construction 8.** This construction generates an infinite family of type  $(5, 6)$  planar graphs with  $V_5 = 12$  and  $V_6 = 3k$  for all  $k \in \mathbb{Z}^+$ . Consider the graph  $G$  created by embedding the icosahedron into the plane so that there are 12 vertices of degree 5 and 20 faces. The structure of  $G$  allows for Algorithm 5.7 to be used. Any face of degree 3 may be chosen to initialize vertices 1, 2, and 3 for Algorithm 5.7. See Figure 18, for example.

The code generated by the graph in Construction 8 is rate optimal by Bound 2. All faces of the graph are degree 3 at any iteration of Algorithm 5.7. The code approaches the graph theoretic bound of  $\frac{2}{3}$  (for  $d = 3$ ) as the number of iterations of Algorithm 5.7 go to infinity.



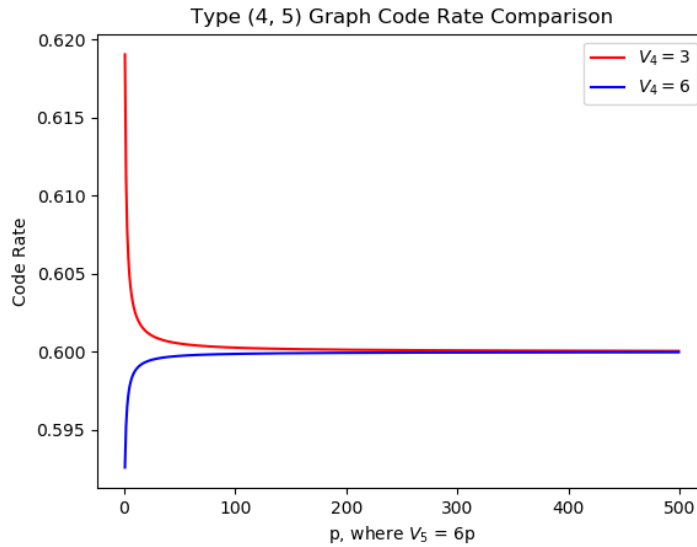


Figure 16. Type (4, 5) Construction 7 Code Rate

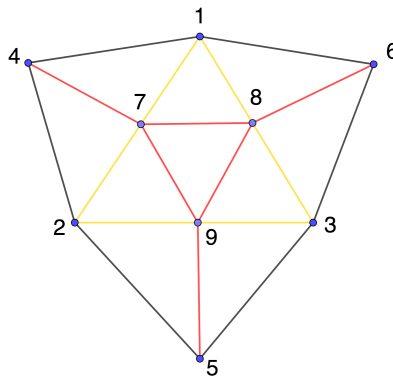
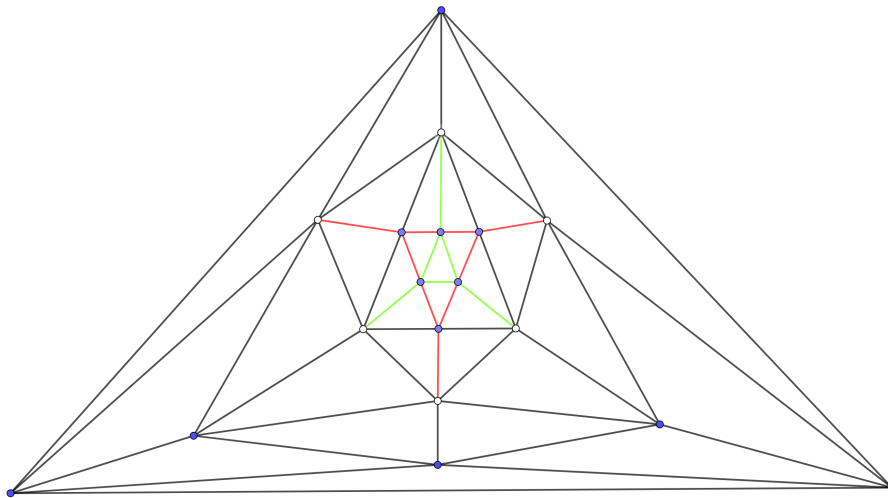


Figure 17. An iteration of Algorithm 5.7. New edges added in red. Face  $f$  in yellow.

## 6. Conclusion

An *edge code* generated by a planar graph has minimum distance  $d$  equal to the smallest degree of a face, and availability  $t = 2$ . The recovery set size for a type  $(j, j + 1)$  planar graph is at most  $r = j$ . In this paper we proved that the minimum distance of an edge code coincides with the girth of the planar graph, and that the code rate satisfies a bound that depends on the length and minimum distance of the code. We applied expansion constructions of  $j$ -regular planar graphs to compute the parameters of infinite families of edge codes. We classified the type  $(3, 4)$  almost regular graphs whose code rates meet Bound 2 and proved that there are no infinite families of this type that achieve the bound. For  $d = 3, 4, 5$  we presented constructions of codes that meet the rate bound (see Table 5). We also presented constructions of type  $(3, 4)$  almost regular planar graphs with  $V_3 = 2a$  and  $V_4 = b$  for any  $a, b \in \mathbb{Z}^+$  such that  $V_3 + V_4 \geq 5$ . Operations A and B were presented to construct infinite families of type  $(4, 5)$  almost regular planar graphs. Finally, type  $(5, 6)$  planar graphs that generate rate-optimal LRCs were presented. The family of  $(5, 6)$  almost regular edge codes approach the graph theoretic bound of  $R = \frac{2}{3}$



**Figure 18.** Algorithm 5.7 implemented on the Icosahedron. Edges added in the first iteration of the algorithm in red. Edges added in the second iteration of the algorithm in green. Vertices of degree 6 are shown in white.

as the number of iterations of operations goes to infinity.

Potential future work includes codes generated by different families of almost regular planar graphs, or of 2-connected planar graphs. Graphs that are “almost” almost regular could also be explored, although a larger difference in recovery set sizes might negatively impact the code rates. Codes generated by graphs embedded on other surfaces may also lead to additional constructions of LRCs from graphs.

**Acknowledgment:** The authors would like to thank the referees for their helpful suggestions.

## References

- [1] H. J. Broersma, A. J. W. Duijvestijn, F. Göbel, Generating all 3-connected 4-regular planar graphs from the octahedron graph, *J. Graph Theor.* 17(5) (1993) 613–620.
- [2] P. Gopalan, C. Huang, H. Simitci, S. Yekhanin, On the locality of codeword symbols, *IEEE Trans. Inform. Theory* 58(11) (2012) 6925–6934.
- [3] M. Hasheminezhad, B. D. McKay, T. Reeves, Recursive generation of simple planar 5-regular graphs and pentangulations, *Journal of Graph Algorithms and Applications* 15(3) (2011) 417–436.
- [4] S. Kadhe, R. Calderbank, Rate optimal binary linear locally repairable codes with small availability, In *2017 IEEE International Symposium on Information Theory (ISIT)* (2017) 166–170.
- [5] S. Kadhe, R. Calderbank, Rate optimal binary linear locally repairable codes with small availability, arXiv preprint, arXiv:1701.02456, 2017.
- [6] M. Meringer, Fast generation of regular graphs and construction of cages, *J. Graph Theor.* 30(2) (1999) 137–146.
- [7] M. Meringer, Regular planar graphs, available online at <http://www.mathe2.uni-bayreuth.de/markus/reggraphs.html>, accessed 2009.
- [8] N. Prakash, V. Lalitha, P. Vijay Kumar, Codes with locality for two erasures, In *2014 IEEE International Symposium on Information Theory (2014)* 1962–1966.
- [9] E. F. Schmeichel, S. L. Hakimi, On planar graphical degree sequences, *SIAM J. Appl. Math.* 32(3)

(1977) 598–609.

- [10] E. Steinitz, Vorlesungen über die Theorie der Polyeder: unter Einschluß der Elemente der Topologie, volume 41, Springer-Verlag, 2013.