

Spotting Graph Theory Problems in Spot It

Dave Fetterman¹ and James Wang²

¹Obviously Unemployed

²Surprisingly Employed

4/16/23

Abstract

The kids' card game Spot It supports a unique mechanic: every card of the 55-card deck has eight different symbols, and shares exactly one symbol of some kind with every other card. Intuitively, this seems impossible. The obvious game play (“spot the match”) works for children as young as two; the intricacies of deck construction astound these children of forty-two. In constructing our own deck, we run across interesting problems in graph theory, number theory, and abstract algebra¹. Some are solved, some unsolved, but they span a wide range of domains not obviously connected.

A deck of n Spot It cards, each featuring s symbols from a catalogue of m total symbol choices can be constructed many ways, but only a few ways if every symbol occurs exactly g times. In that case, the choice of the (s, g) pair both dictates m and n and is strongly restricted.

However, if g is then of the special form $g = p^k + 1, p \in \mathbb{P}, k \in \mathbb{N}$, we can construct a deck with uniform g , and beyond that, we can prove the following four constructions are surprisingly equivalent²:

1. (The Children's Game) A deck of n Spot It Cards with $s = g$ symbol slots, where each of $m = n$ symbols occurs exactly g times,
2. (Graph Theory) An edge partition of the complete graph K_n into complete subgraphs K_g ,
3. (Abstract Algebra) A finite (Galois) field of order $g - 1$, and
4. (Number Theory) A Perfect Difference Set[1] on n elements.

We then examine other configurations of s and g , as well as comment on other reasonable constructions of a Spot It deck.

¹There are also representations in projective geometry, but let's save some domains for everyone else.

²Or, at least, one can derive any from another

1 The Game and the Problem Statement

Introduced to us by Ari Steinberg, Spot It is a children’s game of 55 cards as shown in Figure 1a, featuring eight colorful symbols on each. Though gameplay comes with a few variants in its tiny rulebook, the primary mechanic when presented two cards is *spot the single common symbol first*. The game is simple enough for a two- or three-year old to grasp (and win!), but the question emerges: *just how did they construct such a deck?*³

Naturally, there are trivial constructions: every symbol occurs only twice, or the count of symbols is so varied that a deck can be constructed almost greedily. However, the Spot It game has uniformly 8 symbol “slots” on each card, which each appear 8 times⁴ across the different cards.

This is what we examine in this paper:

The Core Question: For what choices of g and s can we construct a Spot It deck where each card has space for s symbols, and each symbol appears exactly g times throughout the deck?

1.1 Reframing as a graph problem

Noticing that every card has a relationship to every other card (notably, the identity of the single symbol shared between them) as in Fig. 1b, we take our first step by reconstructing this problem as an undirected graph as in Fig. 1c. We can always represent a Spot It deck as a graph.

The Graph Representation: A deck of Spot It Cards each with s symbol slots, where each symbol appears g times, can be represented by a graph G :

1. With n nodes, where $n = s(g - 1) + 1$,
2. With m unique edge colors, where m denotes the number of unique symbols, and
3. Where all edges of any single color form a complete subgraph on g nodes, so $m = \binom{n}{g} / \binom{g}{2}$.

Note: Singletons (groups where $g = 1$) would be rendered as self-edges. These are uninteresting and are generally ignored in this paper

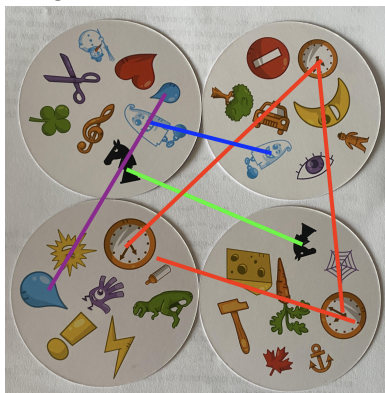
Proof:

³Note: There have been some other investigations into Spot It[2], but for the purposes of enjoyment, everything in this paper was researched without reference to Spot It work.

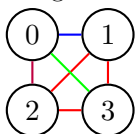
⁴As we will see later, there should be 57 cards for this to be true; it’s likely two cards were removed.



(a) Four cards in the game



(b) Four cards in the game with links



(c) Four cards graph, with cards as nodes and an edge color for each shared symbol

1. As in Fig. 2, node n_0 's adjacencies are exactly s monicolor cliques of size $g - 1$ (excluding n_0 itself). In a complete graph, these adjacencies comprise the total node set, so $n = (g - 1)s + 1$. Using any other node is equivalent.
2. As in Fig. 1b, while card 1 and card 2 have the relationship "clock", node 1 and node

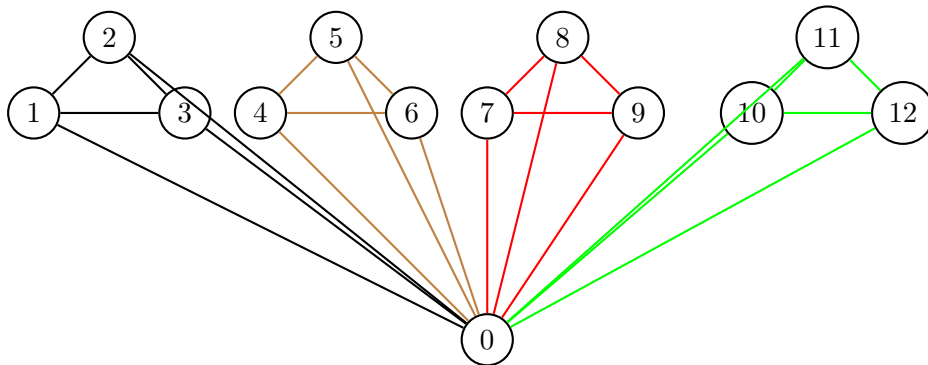


Figure 2: $n = s(g - 1) + 1$. Here, $s = 4, g = 4$

2 share an edge with the color red. This is the same relationship between nodes 2 and 3, and nodes 1 and 3. “Drop”, “knight”, and “ghost” would be colors purple, green, and blue, respectively. This works because every edge has exactly one color in this formulation (corresponding 1:1 with a symbol), and every card pair shares exactly one symbol.

3. All cards with a given symbol (say, “clock”) must link via a single color (here, red) to all other nodes whose card has that symbol; this is a complete subgraph. A complete graph K_n has $\binom{n}{2}$ edges. A monochrome clique of size g is a complete graph as well, with $\binom{g}{2}$ edges. K_n 's edges are exactly these equal-sized cliques, so there are therefore $m = \frac{\binom{n}{2}}{\binom{g}{2}}$ of them, corresponding to “colors”.

And since every edge in our complete graph K_n is in exactly one monochrome clique of size K_g , this becomes a crisp graph theory problem:

The Core Question in Graph Terms: Given s and g as before, can we construct an edge partition (colloquially here, “coloring”) of a complete graph on n nodes K_n into a set of complete subgraphs of size g (denoted K_g)?

Note: Our idiosyncratic term “coloring”, which really just serves to better visualize partitioning a graph into disjoint edge sets, should not be confused with the traditional term “edge coloring”, where no edges of the same color can meet at a node. Also, we sometimes sloppily use “clique”, “group”, and “complete subgraph” interchangeably to mean “the entire set of nodes connected by one color of edge”.

Though exhaustive research wasn’t done, this graph problem does not appear to have a clear analytical solution out there.⁵

⁵Or people who care about publishing it within the reach of lazy hobbyists, anyway!

Since m and n are determined from s and g , we'll start by looking at possible candidate configurations of s and g .

2 The Candidate Theorem: $g|s(s-1), g \leq s$

Suppose that every card has s symbols, and any symbol has exactly g cards containing it⁶. Then

1. $g|s(s-1)$.
2. If $s > 1$ and $g > 1$ then $g \leq s$.
3. If $s > 1$ and $g > 1$, then $m = (\frac{s}{g})n$ and therefore $m \geq n$.
4. All nontrivial candidate configurations of g, s are then $g \leq s, g|s(s-1)$.

Proof:

1.

$$\binom{g}{2} \Big| \binom{n}{2} \Rightarrow \frac{n(n-1)}{g(g-1)} \in \mathbb{N} \Rightarrow g(g-1)|n(n-1) \quad (1)$$

$$n = (g-1)s + 1 \Rightarrow g(g-1)|(sg-s+1)(sg-s) = (sg-s+1)s(g-1) \quad (2)$$

$$\Rightarrow g|s^2g - s^2 + s \Rightarrow g|(1-s)s \Rightarrow g|s(s-1) \quad (3)$$

2. Any node n_i is adjacent to s monochlor cliques of size g . These cliques $C_1 \dots C_s$ (containing non- n_i nodes if $g > 1$) comprise all nodes, and any other cliques can contain no more than one of each C_i . This means that clique of size g greater than s cannot be formed, since the only place to find nodes are these $C_1 \dots C_s$. The other trivial case, $s = 1$, means there is only one color in the whole graph.

This means we need not consider configurations like $g = 6, s = 3$ even though $6|3(3-1)$.

⁶for example, all cards contain $s = 4$ symbols in Fig. 4, which each appear on $g = 3$ cards

3. Another corollary here is that $\boxed{m \geq n}$, since:

$$n = (sg - s + 1) \quad (4)$$

$$m = \frac{\binom{(sg-s+1)(sg-s)}{2}}{\binom{g}{2}} = \frac{(sg-s+1)(sg-s)}{g(g-1)} = \frac{(sg-s+1)s}{g} \quad (5)$$

$$\Rightarrow m = \left(\frac{s}{g}\right)n \quad (6)$$

$$s \geq g \Rightarrow m \geq n \quad (7)$$

4. This is just a combination of (1) and (2). But for example, a tiling of triangles ($g = 3$) means that either $s \equiv 0 \pmod 3$ (see Fig. 3) or $s \equiv 1 \pmod 3$ (see Fig. 4).

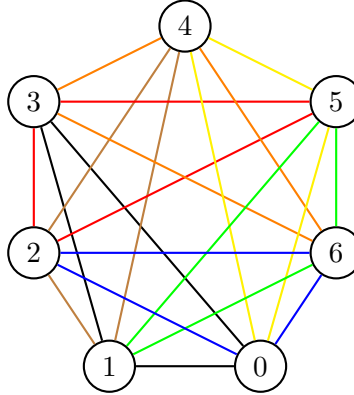


Figure 3: $s=3, g=3, n=7, m=7$. Cliques of form $(n_i, n_{i+1}, n_{i+3}), \forall i$

3 Constructing $g = s - 1$ over a field

Though we presented a few legitimate examples of complete graphs tiled by uniformly sized complete subgraphs in Fig. 3 and Fig. 4, these are not easy to find by hand once s becomes much larger. The whole problem of graph partitioning admits many algorithms, most approximations[3], though usually referring to separating actual *nodes* into partitions, rather than edges, and usually over an arbitrary graph instead of a relatively simple complete graph.

We can, however, systematically construct an edge partition if g is a prime power.

g=s-1 construction: If g is a prime power p^k , we can explicitly construct a graph that satisfies our game with $g = s - 1$.

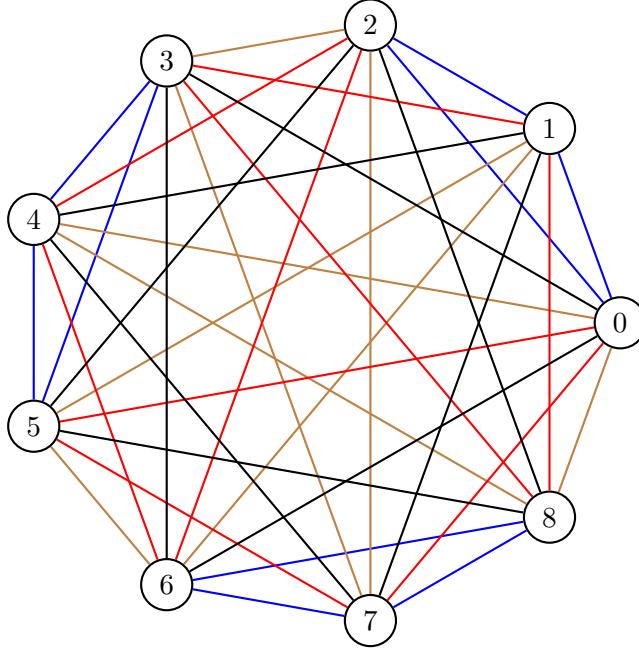


Figure 4: $s=4, g=3, n=9, m=12$.

Cliques: $(n_i, n_{i+3}, n_{i+6}), (n_i, n_{i+1}, n_{i+2}), (n_i, n_{i+4}, n_{i+8}) \forall i, (n_i, n_{i+5}, n_{i+7}), i \equiv 0 \pmod 5$

As the combination of s, g determine the shape of the graph entirely, $g = s-1$ implies:

- The graph has $n = s(g-1) + 1 = (g+1)(g-1) + 1 = g^2$ nodes.
- Those nodes can be grouped into g groups of size g .
- There are $m = \binom{s}{g}n = \binom{s}{g}g^2 = sg = (g+1)g$ colors in the graph.

3.1 General Construction Algorithm

Algorithm Summary: Our algorithm takes every constant c in a finite field \mathcal{F} , and goes through the multiplication table of \mathcal{F} row (y) by row. To a clique $C_{c,y}$ specified by these two, we add an element from G_x for every column x : The entry from G_x indexed by the entry at $z = (x, y)$ in the table, plus the constant c .

Construction Algorithm for $g = s - 1$:

To construct the $m = g(g-1)$ colors (symbol cliques) in the graph:

- Construct a finite field $\mathcal{F} = \mathcal{GF}(g)$, which is of size g .

- Divide the g^2 nodes into g cliques $(G_0, G_1, \dots, G_{g-1})$, where indices $[0 \dots g-1] \in \mathcal{F}$, of g nodes each $((G_{0,0}, G_{0,1} \dots G_{0,g-1}) \dots (G_{g-1,0}, G_{g-1,1} \dots G_{g-1,g-1}))$
- For all $c \in [0, g-1]$:
 - For all $y \in [0, g-1]$:
 - * Create clique $C_{c,y}$, initially the empty set. y is the row (specifying the multiplier) and c is a constant (added to each result).
 - * For all $x \in [0, g-1]$, set $z = G_x \cdot G_y + c$, where $+$ and \cdot refer to addition and multiplication rules for \mathcal{F} , and add $G_{x,z}$ to clique $C_{c,y}$.
- The cliques $(G_0, G_1, \dots, G_{g-1})$, plus the g^2 cliques like $C_{c,y}$ form the $(g+1)g$ cliques or “colors”.

3.2 Constructing with prime g

We can start with the easiest way to see this: $g = p, p \in \mathbb{P}$.

Let’s construct the operation tables for the finite field on $p = 3$, as in Fig. 5, and build an algorithm for constructing the graph from them.

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	1	0

(a) Addition table GF(3)

\cdot	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

(b) Multiplication table GF(3)

Figure 5: Field tables for GF(3)

Refer to Fig. 6.

1. Cliques G_0, G_1, G_2 : these are the triangles in black.
2. Cliques $C_{0,0}, C_{1,0}, C_{2,0}$: these are the triangles in blue. These walk through the 0 row in the multiplication table in Fig. 5b.
3. Cliques $C_{0,1}, C_{1,1}, C_{2,1}$: these are the triangles in brown. These walk through the 1 row in the multiplication table in Fig. 5b.
4. Cliques $C_{0,2}, C_{1,2}, C_{2,2}$: these are the triangles in red. These walk through the 2 row in the multiplication table in Fig. 5b.

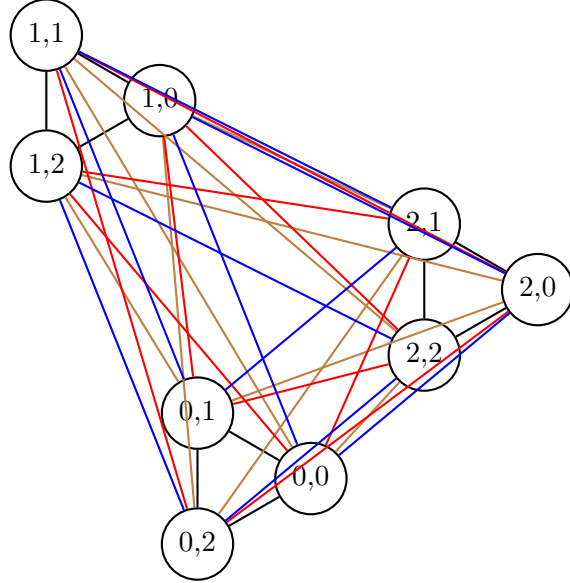


Figure 6: whole K_9 : $s = 4, g = 3, n = 9, m = 12$

Every node is connected to every other node once in Fig. 6, since for nodes x, y and z, w :

- If $x = z$, they're in the same "black" clique G_x .
- If $y = w$, they're in a "blue" clique $C_{c,0}$, where our multiplier y is zero. Every member of $C_{2,0}$, for example, is the 2 element of their G_i : $G_{0,2}, G_{1,2}, G_{2,2}$.
- Else there is some multiplier a where starting from node (x, y) , going $z - x$ multiplicative "hops" away lands us at node w . These are like the brown ($a = 1$) or red ($a = 2 \equiv -1 \pmod{3}$) cliques.

Compare identical graphs Fig. 6 (generated from the algorithm) and Fig. 4. The mapping between node (i, j) in Fig. 6 and node k in Fig. 4 is $(i, j) \rightarrow k = i + 3j$, with inverse $k \rightarrow (k \pmod{3}, \lfloor k/3 \rfloor)$. In Fig. 4:

- The black cliques are $\{G_0, G_3, G_6\}, \{G_1, G_4, G_7\}, \{G_2, G_5, G_8\}$, grouped by $i \pmod{n}$.
- The blue cliques are $\{G_0, G_1, G_2\}, \{G_3, G_4, G_5\}, \{G_6, G_7, G_8\}$, a walk through the top row of Fig.5b.
- The brown cliques are $\{G_0, G_4, G_8\}, \{G_6, G_1, G_5\}, \{G_3, G_7, G_8\}$, a walk through the second row of Fig.5b.
- The red cliques are $\{G_0, G_7, G_5\}, \{G_3, G_1, G_8\}, \{G_6, G_4, G_2\}$, a walk through the third

row of Fig.5b.

3.3 Constructing for g as prime power

For prime numbers, it's clear that every increment in $[1, p-1]$ traverses a path $a, 2a, 3a \dots (p-1)a$ where the n th member is unique among all other increment paths.

The leap of insight comes in realizing that, in general, these cliques are not additive increments, but journeys through a field's *multiplication table*; for g prime, these look additive, but for more complicated finite fields, we need to defer to the table.

We can do this for prime-power fields like $GF(2^2 = 4)$:

+	0	1	B	D
0	0	1	B	D
1	1	0	D	B
B	B	D	0	1
D	D	B	1	0

·	0	1	B	D
0	0	0	0	0
1	0	1	B	D
B	0	B	D	1
D	0	D	1	B

(a) Addition table $GF(4)$ (b) Multiplication table $GF(4)$

Unlike a prime-order finite field, the addition table is not cyclic, so in a sense, our indices for $G_{0,i}$ say are not $i \in [0, 3]$ but $i \in \{0, 1, B, D\}$! This is why the phrase “where $[0 \dots g-1] \in \mathcal{F}$ ” is important in the algorithm.

c	y	G_0	G_1	G_B	G_D
0	0	0	0	0	0
0	1	0	1	B	D
0	B	0	B	D	1
0	D	0	D	1	B

(a) Cliques $C_{0,i}$

c	y	G_0	G_1	G_B	G_D
1	0	1	1	1	1
1	1	1	0	D	B
1	B	1	D	B	0
1	D	0	B	0	D

(b) Cliques $C_{1,i}$

c	y	G_0	G_1	G_B	G_D
B	0	B	D	0	1
B	1	B	0	1	D
B	B	B	1	D	0
B	D	B	B	B	B

(c) Cliques $C_{B,i}$

c	y	G_0	G_1	G_B	G_D
D	0	D	D	D	D
D	1	D	B	1	0
D	B	D	1	0	B
D	D	D	0	B	1

(d) Cliques $C_{D,i}$

Figure 8: $s=5, g=4$ adjacency tables

Here are the resulting tables for $s = 5, g = 4$. For example, in Fig. 8a, consider the third

row as saying “clique $C_{0,B}$ contains nodes $G_{0,0}, G_{1,B}, G_{B,D}, G_{D,1}$ ”.

(Of course, once the multiplication is defined, feel free to substitute 2 for B and 3 for D , in, say, a program.)

Including the “black” cliques like $G_B = \{G_{0,B}, G_{1,B}, G_{2,B}, G_{3,B}\}$, we see that there are no repeated edges, and all edges are accounted for.

In general, to prove that this table exactly represents the “color” of every edge, we need to show that for every pair of elements, say, $(B, 1)$, and every pair of columns, say 0 and B , (see the blue row in Fig. 8c) that B and 1 appear in the same row in columns 0 and B exactly once.

Proof: To ensure there are no duplicates, consider two columns x_1 and x_2 that have a repeated value pair in some table, once on the table row beginning (c, g_y) , once on (c^*, g_y^*) :

- Assume $g_y g_{x_1} + c = g_y^* g_{x_1} + c^*$, and $g_y g_{x_2} + c = g_y^* g_{x_2} + c^*$ for $g_{x_1} \neq g_{x_2}$
- Subtract the two to get $g_y(g_{x_1} - g_{x_2}) = g_y^*(g_{x_1} - g_{x_2})$
- The field \mathcal{F} requires the nonzero $(g_{x_1} - g_{x_2}) \in \mathcal{F}$ to have an inverse.
- Multiplying both sides by that inverse, we have $g_y = g_y^*$, showing they cannot be distinct. Thus, we have no duplicates.

Similar field-based arguments can be made to ensure that every pair is represented. Since everything we’re dealing with is finite, however, we can also use a pigeonhole approach:

- For any pair of columns, there are g^2 pairs of values to account for.
- Across the tables above, there are g^2 rows (combinations of c and y that generate the cliques).
- From the previous argument, no two rows can be duplicated.
- Therefore, by the pigeonhole principle, every combination is represented, and therefore, between any two G_j cliques, every node has an edge with every other.

4 Constructing $g = s$ from $g = s - 1$

With the previous construction ($g = s - 1$) in hand, we can easily construct a $g = s$ graph (adding one to g , keeping s the same), with the same restrictions on g . See Fig. 9 for a visual on this.

- Create node n_y for $y \in \mathcal{F}$
- Add n_y to all cliques $C_{c,y}$. For a given y , these cliques share no nodes.

- Create node n^* . Add this to every G_i clique.
- Create clique of all $n_y, y \in [0, g]$ plus n^* .

We see then that:

- Every new node n_y gets added to the new “n” clique plus $s - 1$ cliques, all of size s .
- n^* is added to $s - 1$ cliques plus the bottom clique.
- Every G -style clique gets n^* added to it.
- Every $C_{c,y}$ clique adds node n_y .
- So, the new partition is one where $g = s$, built on the previous where $g = s - 1$.

Note that you can create $s = g = 8$ this way, with $n = m = 57$. The Spot It card deck comes with 55 cards with $s = 8$ “slots”, so our suspicion is that two cards were simply dropped from the set,.

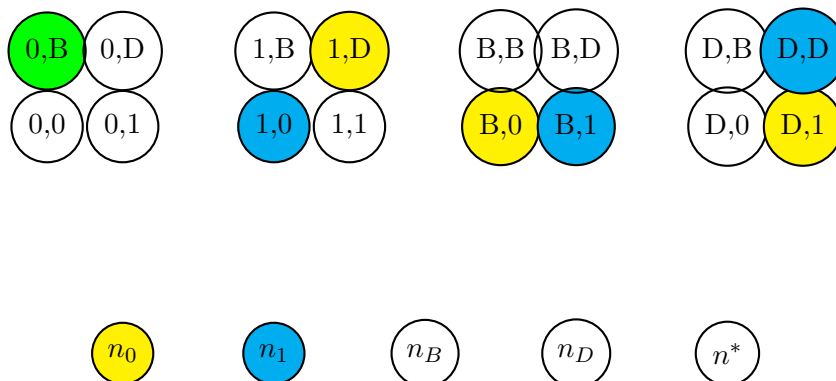


Figure 9: $s=5, g=5$ created from $s=5, g=4$, no edges shown. The blue is the $C_{B,1}$ clique newly adding in n_1 . The yellow is the $C_{0,1}$ clique newly adding in n_0 . Node $0, B$ is in both.

5 Another Approach: Constructing $g = s$ with perfect difference sets

Though it will be shown equivalent to the last construction, we can use another concept to build these graphs when $g = s$: perfect difference sets. Notably, these are proven to exist for $g = p^k - 1$ by Singer[1].

Searching for complete graph partitions by brute force is difficult. Even a graph of size 7 like Fig. 3 requires sorting through putting 21 distinct objects into 7 distinct bins⁷, and

⁷ $(21!)/((3!)^7) \approx 1.8 \times 10^{14}$, though with some symmetries

the numbers get worse from there.

There is some hope that a partition or coloring, should it exist, would exhibit some regularity; after all, every node has a similar configuration of edges and adjacent nodes in a complete graph.

Additionally, when $g = s$, we've seen that $n = m$, so seeking a unique complete subgraph K_g mapped 1:1 to each node which, in total, partition the complete graph, seems like a good strategy.

Looking at Fig. 3, we see that every node at index i can be mapped to a K_3 (color) whose vertices are $i, i + 1$, and $i + 3$, with addition being modulo 7. (Take a look at node 0's black triangle $(0, 1, 3)$). This means that we've created "chords" of offset 1, 2, and 3 (counterclockwise from i) in the graph. If we can make sure node i participates in exactly one chord of each offset in $[1, n - 1]$, then we've ensured our partition represents every edge involving node 0 once.

Node i still needs to find adjacencies with nodes $i + 2, i + 4, i + 5$, and $i + 6$ (or remapped over modulo 7, $i + 2, i - 3, i - 2$, and $i - 1$). However, we can find these latter four by simply repeating this triangle $(j, j + 1, j + 3)$ over the rest of the graph; in particular, the triangles at $(i - 1, i, i + 2)$ (here, blue if $i = 0$) and $(i - 3, i - 2, i)$ (yellow) take care of i 's six required adjacencies. Then, we repeat this recipe for all nodes.

So, if for a given n , we can find a set $a_i \in [1, n - 1]$ of size $g - 1$ such that $\bigcup_{i \neq j} (a_i - a_j \bmod n) = [1, n - 1]$, then this can be repeated for every node to cover the graph.

That such a set exists is not obvious. However, through some basic search algorithms at <https://github.com/fettermania/mathnotes/tree/main/spotit/code>, we can sniff out a few of them:

g,s	n	0 adjacencies
3	7	{1, 3}
4	13	{1, 3, 9}
5	21	{1,4,14,16}
6	31	{1, 3, 8, 12, 18}
8	57	{1, 3, 13, 32, 36, 43, 52}
9	73	{1, 3, 7, 15, 31, 36, 54, 63}
10	91	{1, 3, 9, 27, 49, 56, 61, 77, 81}
12	133	{1, 3, 12, 20, 34, 38, 81, 88, 94, 104, 109}

Figure 10: Perfect Difference Sets up to $g = s = 12$

Fig. 11 is an example of 0's adjacencies for $s = g = 6$:

It turns out that in the 1930s, James Singer found that these configurations, termed *perfect*

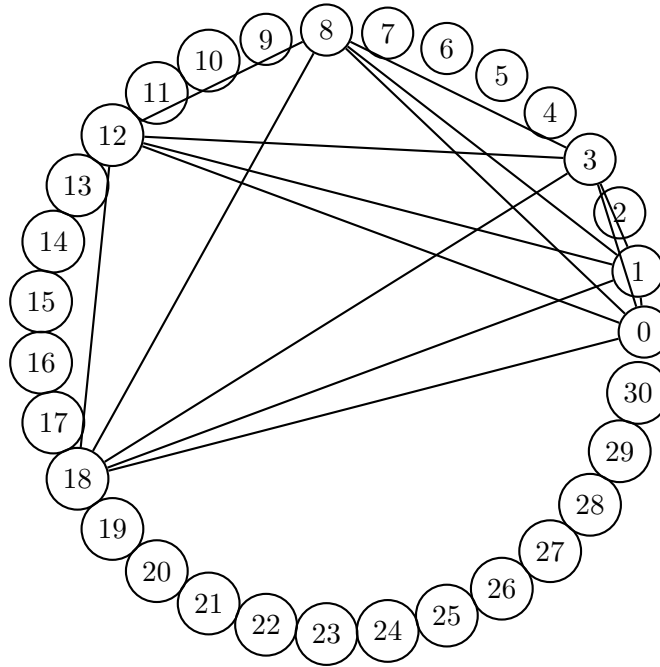


Figure 11: perfect difference set on $g=6$, $s=6$, $n=m=31$

difference sets, exist for $g = p^k, p \in \mathbb{P}, k \in \mathbb{N}$, making $n = g^2 + g + 1$ [1]. This result was not connected to graph edge partitioning in his paper. Though the existence for some of the other types of g have been disproven, this appears to be the main theorem in the area.

Singer's Theorem on Perfect Difference Sets: If $g = s$, then $n = g^2 - g + 1$ by the Graph Representation Theorem and $n = m$ by the Candidate Theorem. Thus, we're looking for a PDS over $n = g^2 - g + 1$. This has been proven to exist if $g = p^k, p \in \mathbb{P}, k \in \mathbb{N}$.

Singer's proof relies also on finding a finite field \mathcal{F} of size g . There are some deterministic, non-search-based methods of doing this[4], though they seem quite intricate.

This limitation to prime powers, just like the partitions for $g = s$ found in section 3, explains the row omissions in the brute force search table in Fig. 10. So, if $g = p^k + 1$, a perfect difference set construction for an edge partition exists, and a field-based (section 3) construction also exists.

6 Interlude: Graph Equivalence up to relabeling

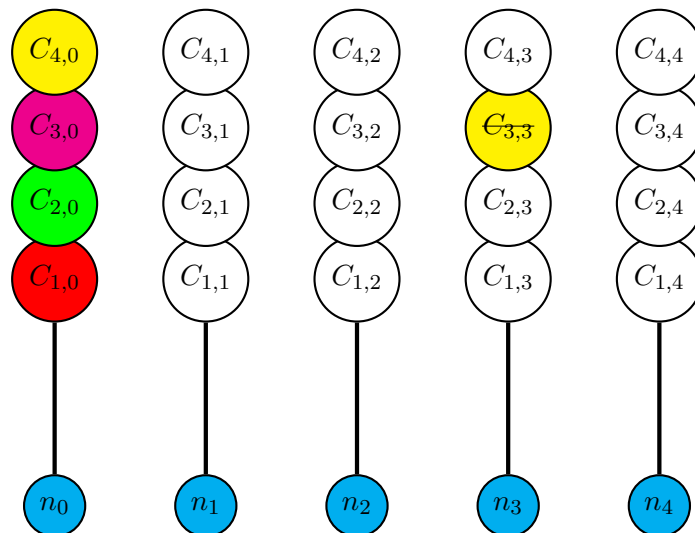


Figure 12: $s=5$, $g=5$ from $s=5$, $g=4$ with no edges. The blue n_j is an arbitrary clique, and $C_{j,*}$ are the $g-1$ cliques (colors) adjacent to each node.

$g=s$ equivalence Theorem: On a complete graph where $g = s$, any edge partition is equivalent up to relabeling.

Proof: If $g = s$, and therefore $m = n = g^2 - g + 1$, consider any clique.

A node like n_0 in this clique (call it the “blue” clique at the bottom of Fig. 12) is adjacent to $g-1$ unique other colors or cliques, represented as the cliques $C_{i,0}$.

No color can be shared between these color adjacency sets (like $C_{4,0}$ and $C_{3,3}$ here). Consider if n_0 and n_3 were both adjacent to yellow, they would have both a yellow and blue edge between them.

Therefore, this blue clique shares a node with all of the other $g(g-1)$ color cliques. Because blue was an arbitrary selection, every color is adjacent to every other color on some single node.

Imagine, for a moment, a new graph G' where colors in our original graph G are nodes with and edges exist between nodes representing colors sharing a node in G . We have just shown that every color is adjacent, and so G' is a complete graph.

Every complete graph of the same size is clearly equivalent up to relabeling, so there is only one way to partition a complete graph into complete subgraphs when $g = s$.

An important consequence: Because $g = s$ in both the cycle generation in section 3 as well as the PDS generation in section 5, these are, up to relabeling, the same coloring.

7 Constructing a $g = s - 1$ partition from a $g = s$ partition

Suppose we have constructed $g = s$ by Perfect Difference Sets or (as we have shown, ultimately identically) by the method shown in Fig. 9. Consider removing the $n^* \cup (\bigcup_i n_i)$ clique and all associated edges. This changes our partition into one in which:

1. We have one fewer clique.
2. Each remaining clique has one member removed (we removed some n_i from each n_i -including clique, and node n^* was removed from each G clique.)
3. We lose g nodes.
4. s remains the same.

This matches the configuration of $g = s - 1$. Though every graph constructed through this method seems like it would be isomorphic (in particular, to graphs constructed using the method in section 4), we can't categorically rule out other constructions.

8 Considering wider $g|s$ and $g|s - 1$: Inception

It seems we have reached the limit of possibilities when $g = s$. In such a graph, when $g - 1 = p^k$:

- We can create a game of Spot It with s symbols, each of which appear g times (section 1)
- We can always construct $g = s - 1$ through a finite field $\mathcal{GF}(\infty)$ (section 3), and augment with another clique to form $g = s$ (section 4).
- We can always find a perfect difference set modulo $n = g^2 + g + 1$ and construct a graph partition where $g = s$ (section 5).
- These partitions, and any where $g = s$, are isomorphic (section 6).
- We can always reduce to a partition where $g = s - 1$ by removing a clique and associated edges. (section 7)

However, these only cover the equality cases of $g = s - 1$ and $g = s$. Section 2 showed that we possibly could accommodate $g|s$ and $g|s - 1$ more generally.

Despite repeated attempts, there don't seem to be many surefire constructions (or reductions) from the above methods for slices like $g^* = \frac{g}{k}, k > 1$. Proving that such constructions

are *impossible* also seems difficult.

However, “inception”⁸ is one very minor method of generating such a partition.

Inception: If K_n can be partitioned into complete subgraphs K_g and K_g can be partitioned into complete subgraphs K_h , then K_n can be partitioned into complete subgraphs of size K_h .

Of course, by the equivalencies above, this means we can make Spot It Games of highly varied g and s , even if well outside practical possibility.

For example: We can partition K_{81} ($s = 10, g = 9, n = 81, m = 90$) into groups of K_9 , and, like in figure Fig. 4, we can split those into ($s^* = 4, g^* = 3, n = 9, m = 12$). This gives us a new partition of configuration ($s = 40, g = 3, n = 81, m = 1080$), in which every node still has a uniform number of attached colors, and every color has the same number of node members. This graph visualization has not been attempted.

Say $s = g$ instead, just for another example. This certainly works for turning $s = 9, g = 9, n = 73, m = 73$ into $s = 36, g = 3, n = 73, m = 876$ as well, or even any other configuration where $g \notin \{s, s - 1\}$ as in section 9.

(Note that an even more trivial example would be turning every edge into a K_2 complete graph.)

9 Considering wider $g|s$: Kirkman’s schoolgirl problem

Apparently some versions of this problem were studied even in the nineteenth century. Kirkman’s schoolgirl problem[5] asks: *Can fifteen girls walk in groups of three to school for seven days, such that each pair walks together exactly once?* With colors representing walking groups, $s = 7, g = 3, n = 15, m = 35$, we have something isomorphic to a Spot It deck construction.

Through ad-hoc tinkering, we can find the graph in Fig. 13, showing the adjacencies for a single node. These are repeated “round the horn”, with the exception that the red triangle only appears five times instead of 15.

These are apparently instances of a *Steiner Triple System* $S(t, k, n)$ [6], where we look for groups of k among n total elements, which cover exactly once every set of size t . If $t = 2$ (edges in the graph, pairs of girls), and $k = 3$ (subgroups K_3 , walking trios) among n , we have an instance of a Kirkman problem⁹ (and Spot It game).

⁸This is really just recursively solving our same partition problem on each complete subgraph K_g . “Inception” is a nod to a movie sort of about recursion but also about running and guns.

⁹In fact, those systems those where $t = 2, k = 3$ are named *Kirkman triple systems*.

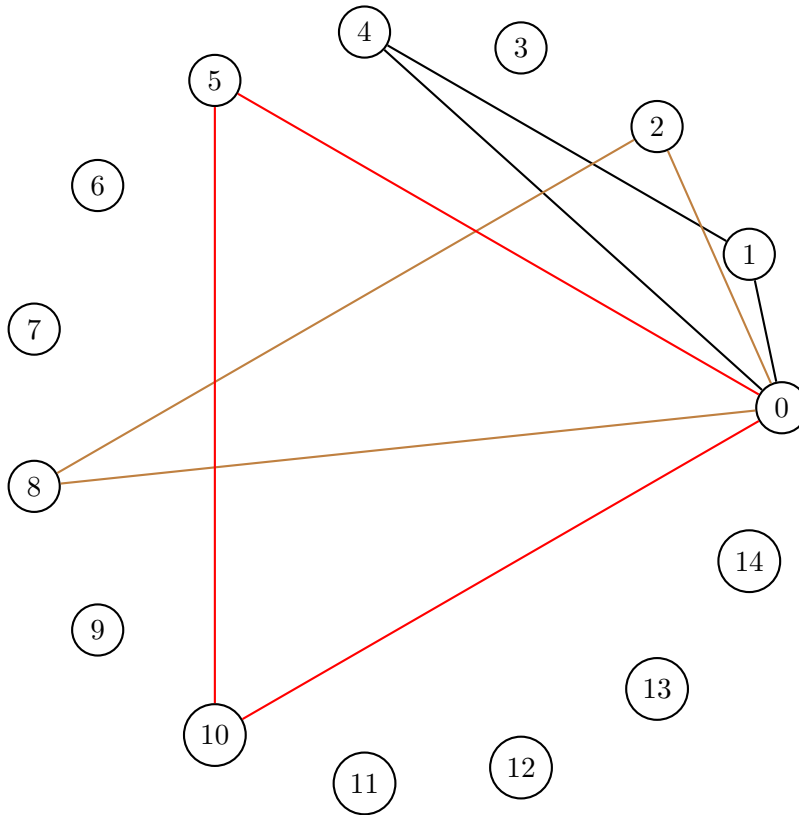


Figure 13: $s=7$, $g=3$, $n=15$, $m=35$, node 0 adjacencies. $i \in [0, 4) : (i, i + 5, i + 10); \forall i : (i, i + 1, i + 4), (i, i + 2, i + 8)$, all mod 15

The followups to this problem are varied (including links to projective and affine geometry) and many unsolved ($t \geq 6$). Much progress has been made on $STS(t = 2, k = 3, n)$ but results seem spotty for $k > 3$.

By some backtracking and brute force searching using our code <https://github.com/fettermania/mathnotes/tree/main/spotit/code>, we can find a few ad-hoc solutions on our own in Fig. 15, even without the problem fully solved.

The adjacencies for node 0 in $s = 10, g = 5$ are shown in Fig. 14, obtained through computational means.

10 Nonuniform g : deletion and partial inception

All of our solutions were concerned with a Spot It deck in which every symbol appears exactly g times. The commercial deck has $s = 8$ on all cards, but contains $n = 55$ cards

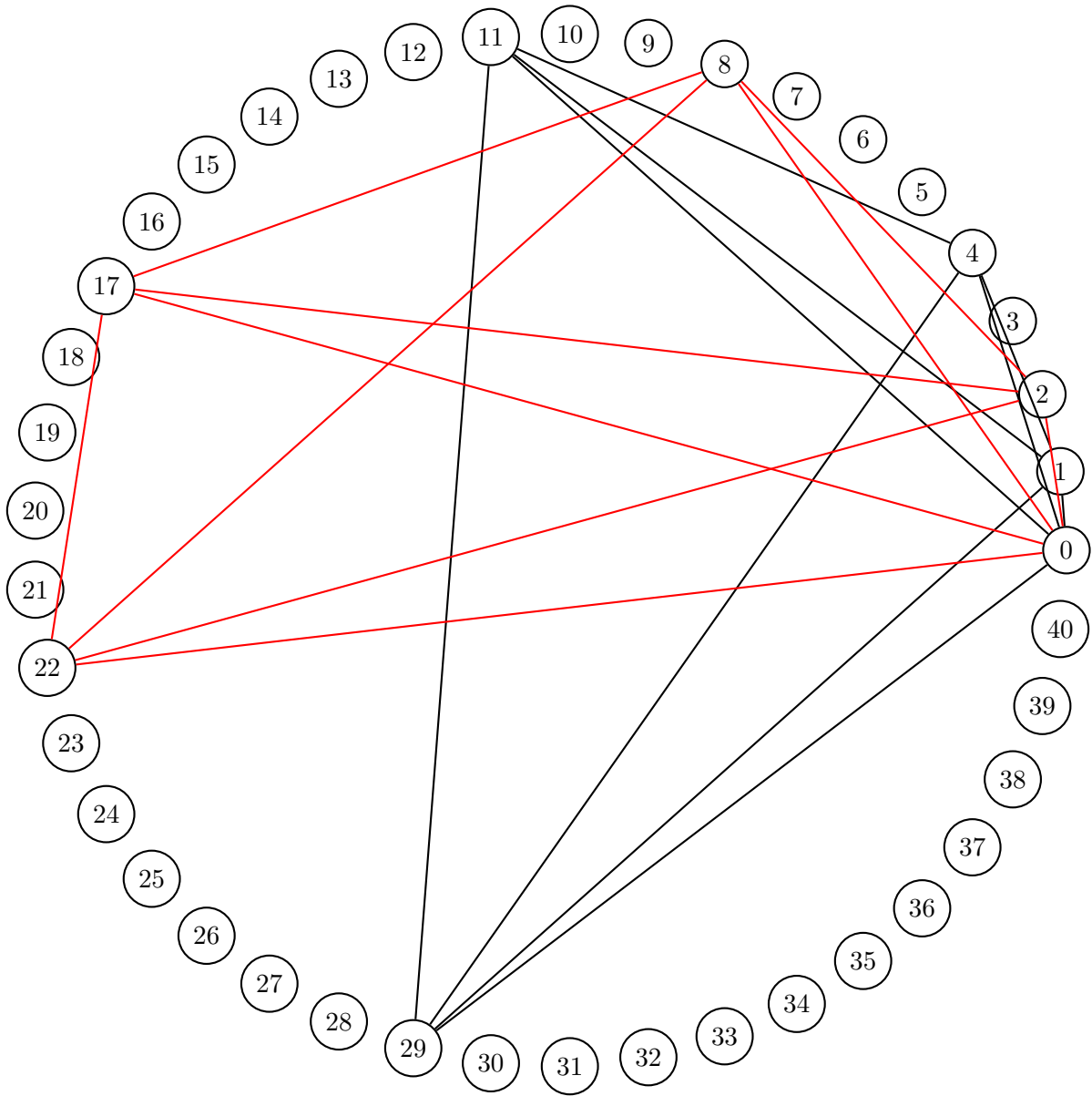


Figure 14: Perfect Difference Set on $s=10$, $g=5$, $n=41$, $m=82$: $(0\ 1\ 4\ 11\ 29)$, $(0\ 2\ 8\ 17\ 22)$

instead of $s(g - 1) + 1 = 8(7) = 57$ cards, assuming $g = 8$. The most likely explanation is that two cards were simply deleted.

Deletion from a deck created with the methods above yields a deck in which s remains constant but g can vary. This could also occur if we replaced *some* of the complete subgraphs

s	g	adjacencies $\forall i$	adjacencies $i \in [0, g - 1]$
6	3	$\{\{0, 1, 4\}, \{0, 2, 7\}\}$	$\{\}$
7	3	$\{\{0, 1, 3\}, \{0, 4, 10\}\}$	$\{\{0, 5, 10\}\}$
9	3	$\{\{0, 1, 6\}, \{0, 2, 10\}, \{0, 3, 7\}\}$	$\{\}$
10	3	$\{\{0, 2, 10\}, \{0, 1, 5\}, \{0, 3, 9\}\}$	$\{\{0, 7, 14\}\}$
10	5	$\{\{0, 1, 4, 11, 29\}, \{0, 2, 8, 17, 22\}\}$	$\{\}$

Figure 15: Ad Hoc Steiner Solutions

K_g with “incepted” graphs (section 8). It’s also likely that we could greedily construct a deck with consistent s if we were unconcerned if g were uniform across colors.

11 Open questions

In addition to the many open questions in Steiner Systems[6], which would include the whether any $g|s$ or $g|s - 1$ has a solution, we have other questions about our systems:

- Can it be true that $g|s(s - 1)$ but not true that $g|s$ or $g|s - 1$ (e.g. $g = 6, s = 9$)?
- Are all constructions where $g = s - 1$ isomorphic?

References

- [1] Singer, James. “A THEOREM IN FINITE PROTECTIVE GEOMETRY AND SOME APPLICATIONS TO NUMBER THEORY”, 1934. <https://www.ams.org/journals/tran/1938-043-03/S0002-9947-1938-1501951-4/S0002-9947-1938-1501951-4.pdf>
- [2] “The Mind-Bending Math Behind Spot It!, the Beloved Family Card Game”, Smithsonian Magazine. <https://www.smithsonianmag.com/science-nature/math-card-game-spot-it-180970873/>
- [3] Wikipedia: https://en.wikipedia.org/wiki/Graph_partition
- [4] Adleman, Leonard M. and Lenstra, Hendrik W. “FINDING IRREDUCIBLE POLYNOMIALS OVER FINITE FIELDS” <https://www.math.leidenuniv.nl/~hwl/PUBLICATIONS/1986a/art.pdf>
- [5] Wikipedia: https://en.m.wikipedia.org/wiki/Kirkman%27s_schoolgirl_problem
- [6] Wikipedia: https://en.wikipedia.org/wiki/Steiner_system
- [7] Pardon the improper formatting of all of the above, including the paper.