Designs for Forests with Seven Edges

Danny Banegas

University of Minnesota: Duluth

September 20, 2024

Graph Fundamentals

Graph Fundamentals

- A **graph** $G = (V, E)$ consists of a set of vertices V and a set of edges E.

Graph Fundamentals

- A **graph** $G = (V, E)$ consists of a set of vertices V and a set of edges E.
- An edge $e \in E$ is a pair of vertices $e = \{u, v\}$ where $u, v \in V$.

Graph Fundamentals

- A **graph** $G = (V, E)$ consists of a set of vertices V and a set of edges E.
- An edge $e \in E$ is a pair of vertices $e = \{u, v\}$ where $u, v \in V$.
- A **subgraph** H of G is a graph whose vertices and edges are subsets of V and E, respectively.

Graph Fundamentals

- A **graph** $G = (V, E)$ consists of a set of vertices V and a set of edges E.
- An edge $e \in E$ is a pair of vertices $e = \{u, v\}$ where $u, v \in V$.
- A **subgraph** H of G is a graph whose vertices and edges are subsets of V and E, respectively.
- A **simple graph** is one where two vertices may only share one edge and where no vertex shares an edge with itself.

Graph Fundamentals

Here is an example of a disconnected simple graph. $G = (V, E)$ where:

Graph Fundamentals

Here is an example of a disconnected simple graph. $G = (V, E)$ where:

- $V = \{a, b, c, d, e, f, g, h, i\}$

Graph Fundamentals

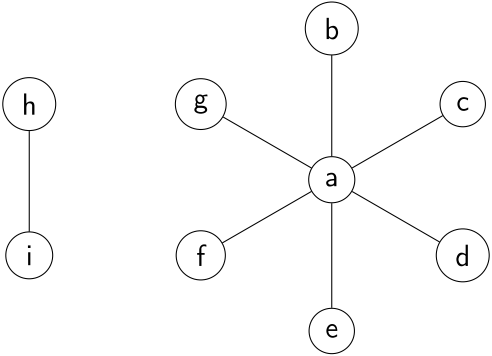
Here is an example of a disconnected simple graph. $G = (V, E)$ where:

- $V = \{a, b, c, d, e, f, g, h, i\}$
- $E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{a, f\}, \{a, g\}, \{h, i\}\}$

Graph Fundamentals

Here is an example of a disconnected simple graph. $G = (V, E)$ where:

- $V = \{a, b, c, d, e, f, g, h, i\}$
- $E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{a, f\}, \{a, g\}, \{h, i\}\}$



Cycle Graphs

Cycle Graphs

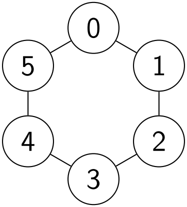
- A **cycle graph** C_n is a graph that consists of a single **cycle**. A cycle is an internally non-repeating sequence of vertices which starts and ends at the same vertex.

Cycle Graphs

- A **cycle graph** C_n is a graph that consists of a single **cycle**. A cycle is an internally non-repeating sequence of vertices which starts and ends at the same vertex.
- It has n vertices and n edges. pause
- We denote can these cycles using cycle notation like in a symmetric group, $(012345) = (501234) = (450123) = \cdots = (123450)$. The previous cycle is the only one in C_6 which is shown below.

Cycle Graphs

- A **cycle graph** C_n is a graph that consists of a single **cycle**. A cycle is an internally non-repeating sequence of vertices which starts and ends at the same vertex.
- It has n vertices and n edges. pause
- We denote can these cycles using cycle notation like in a symmetric group, $(012345) = (501234) = (450123) = \cdots = (123450)$. The previous cycle is the only one in C_6 which is shown below.



The Cycle graph C_6

Tree Graphs

Tree Graphs

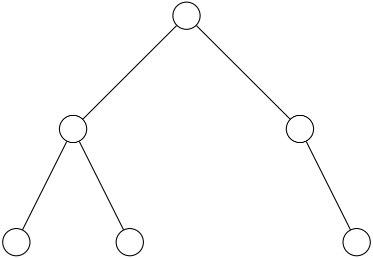
- A **tree graph** is an **acyclic** connected graph. An acyclic graph is one which contains no cycles.

Tree Graphs

- A **tree graph** is an **acyclic** connected graph. An acyclic graph is one which contains no cycles.
- It has n vertices and $n-1$ edges.

Tree Graphs

- A **tree graph** is an **acyclic** connected graph. An acyclic graph is one which contains no cycles.
- It has n vertices and $n-1$ edges.



A tree graph on 6 edges

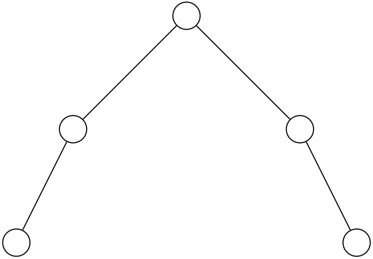
Path Graphs

Path Graphs

- A **path graph** is a special type of tree graph. It is simply a graph that can be described completely by a sequence of vertices.

Path Graphs

- A **path graph** is a special type of tree graph. It is simply a graph that can be described completely by a sequence of vertices.
- It has n vertices and $n-1$ edges.



A path graph on 6 edges

Forest Graphs

Forest Graphs

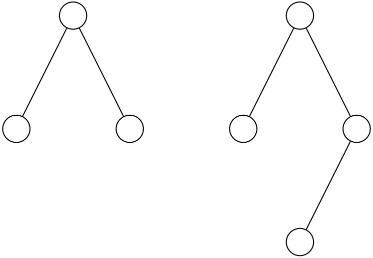
- A **forest** is a disjoint union of trees.

Forest Graphs

- A **forest** is a disjoint union of trees.
- It consists of multiple components, each being a tree.

Forest Graphs

- A **forest** is a disjoint union of trees.
- It consists of multiple components, each being a tree.



A Forest graph composed of trees with 3 and 4 vertices, respectively

Complete Graphs

Complete Graphs

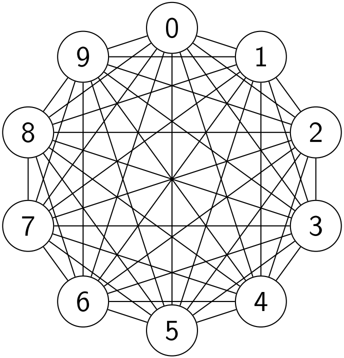
- The **complete graph** K_n is one where every pair of distinct vertices is connected by a unique edge.

Complete Graphs

- The **complete graph** K_n is one where every pair of distinct vertices is connected by a unique edge.
- It has n vertices and how many edges?

Complete Graphs

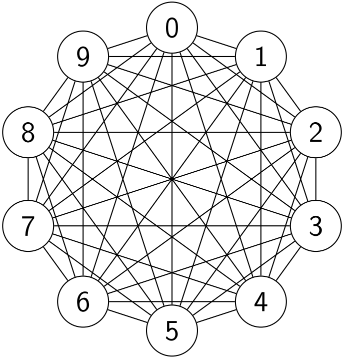
- The **complete graph** K_n is one where every pair of distinct vertices is connected by a unique edge.
- It has n vertices and how many edges?



The Complete graph K_{10}

Complete Graphs

- The **complete graph** K_n is one where every pair of distinct vertices is connected by a unique edge.
- It has n vertices and $\binom{n}{2}$ edges.



The Complete graph K_{10}

Graph Decompositions

Graph Decompositions

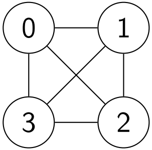
- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.

Graph Decompositions

- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.

Graph Decompositions

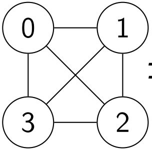
- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.



Decomposition of K_4 into a C_4 and two P_2's

Graph Decompositions

- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.

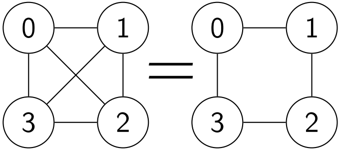


=

Decomposition of K_4 into a C_4 and two P_2's

Graph Decompositions

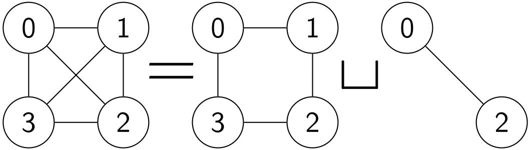
- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.



Decomposition of K_4 into a C_4 and two P_2's

Graph Decompositions

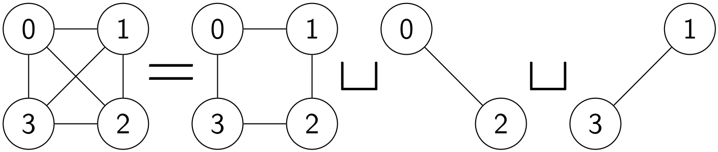
- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.



Decomposition of K_4 into a C_4 and two P_2's

Graph Decompositions

- Let K be a simple graph. A **decomposition** of K is a collection of pairwise edge disjoint subgraphs $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ such that every edge of K belongs to exactly one member of $\mathcal{G}$.
- We call such a collection of subgraphs a decomposition because their union is the entire graph.



Decomposition of K_4 into a C_4 and two P_2's

Designs

Designs

- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.

Designs

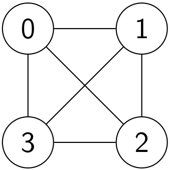
- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.
- If every subgraph in $\mathcal{G}$ is isomorphic to some graph G, then we call $\mathcal{G}$ a $G-$decomposition of K and say that K allows a $G-$decomposition or $(K, G)-$**design**.

Designs

- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.
- If every subgraph in $\mathcal{G}$ is isomorphic to some graph G, then we call $\mathcal{G}$ a $G-$decomposition of K and say that K allows a $G-$decomposition or $(K, G)-$**design**.
- If $K \cong K_n$ then we call $\mathcal{G}$ a $G-$**design** of order n.

Designs

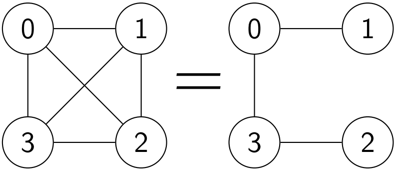
- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.
- If every subgraph in $\mathcal{G}$ is isomorphic to some graph G, then we call $\mathcal{G}$ a $G-$decomposition of K and say that K allows a $G-$decomposition or $(K, G)-$**design**.
- If $K \cong K_n$ then we call $\mathcal{G}$ a $G-$**design** of order n.



A P_3-Design of order 4

Designs

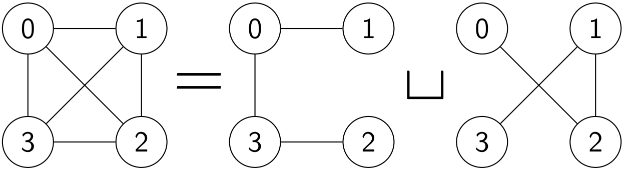
- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.
- If every subgraph in $\mathcal{G}$ is isomorphic to some graph G, then we call $\mathcal{G}$ a $G-$decomposition of K and say that K allows a $G-$decomposition or $(K, G)-$**design**.
- If $K \cong K_n$ then we call $\mathcal{G}$ a $G-$**design** of order n.



A P_3-Design of order 4

Designs

- Let $\mathcal{G} = \{G_0, G_1, \ldots, G_m\}$ be a decomposition of some graph K.
- If every subgraph in $\mathcal{G}$ is isomorphic to some graph G, then we call $\mathcal{G}$ a $G-$decomposition of K and say that K allows a $G-$decomposition or $(K, G)-$**design**.
- If $K \cong K_n$ then we call $\mathcal{G}$ a $G-$**design** of order n.



A P_3-Design of order 4

Necessary condition

- Let G be a graph on m edges. Then there exists a $(K, G)-$design only if m divides $|E(G)|$.

Necessary condition

- Let G be a graph on m edges. Then there exists a $(K, G)-$design only if m divides $|E(G)|$.
- If $K \cong K_n$, then there exists a $(K, G)-$design only if m divides $\binom{n}{2} = \frac{n(n-1)}{2}$.

Necessary condition

- Let G be a graph on m edges. Then there exists a $(K, G)-$design only if m divides $|E(G)|$.
- If $K \cong K_n$, then there exists a $(K, G)-$design only if m divides $\binom{n}{2} = \frac{n(n-1)}{2}$.
- Equivalently, there exists a $G-$design of order n only if n is idempotent modulo $2m$. This is easy to show.

Necessary condition

- Let G be a graph on m edges. Then there exists a $(K, G)-$design only if m divides $|E(G)|$.
- If $K \cong K_n$, then there exists a $(K, G)-$design only if m divides $\binom{n}{2} = \frac{n(n-1)}{2}$.
- Equivalently, there exists a $G-$design of order n only if n is idempotent modulo $2m$. This is easy to show.

Proof

Let $m, n \in \mathbb{N}$. Suppose $m \mid \binom{n}{2}$. Then $\frac{n(n-1)}{2} = mq$ for some $q \in \mathbb{N}$. So then $\frac{n(n-1)}{2m} = q$, and thus $n(n-1) \equiv 0 \pmod{2m}$. Therefore $n \equiv n^2 \pmod{2m}$, and so n is idempotent modulo $2m$. Suppose n is idempotent modulo $2m$, then $n^2 > n$ and therefore $n^2 - n = n(n-1) = 2mp$ for some $p \in \mathbb{N}$. So then $\frac{n(n-1)}{2} = \binom{n}{2} = mp$ and m divides $\binom{n}{2}$. $\square$

Cyclic designs

Cyclic designs

- Let $V(K_n) = \mathbb{Z}_n$

Cyclic designs

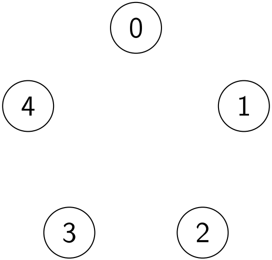
- Let $V(K_n) = \mathbb{Z}_n$
- A G-design is *cyclic* if the permutation $v \mapsto v+1$ on $V(K_n)$ is an automorphism of the design.

Cyclic designs

- Let $V(K_n) = \mathbb{Z}_n$
- A G-design is *cyclic* if the permutation $v \mapsto v + 1$ on $V(K_n)$ is an automorphism of the design.
- We call this act of applying permutations to a labeling *clicking*.

An example of a cyclic design

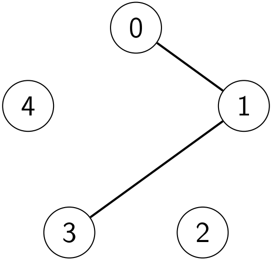
Cyclic P_3-design of order 5



An example of a cyclic design

Cyclic P_3-design of order 5

$\{1, 0, 3\}$

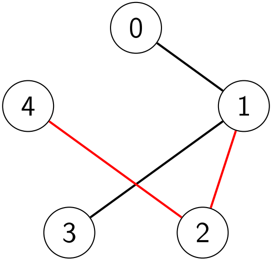


An example of a cyclic design

Cyclic P_3-design of order 5

$\{1, 0, 3\}$

$\{2, 1, 4\}$



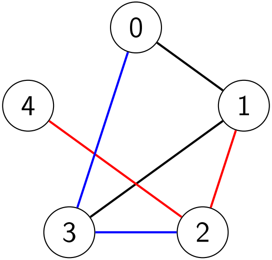
An example of a cyclic design

Cyclic P_3-design of order 5

$\{1, 0, 3\}$

$\{2, 1, 4\}$

$\{3, 2, 0\}$



An example of a cyclic design

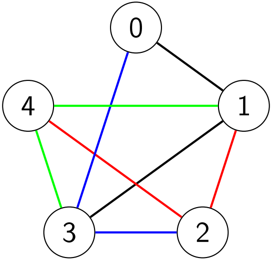
Cyclic P_3-design of order 5

$\{1, 0, 3\}$

$\{2, 1, 4\}$

$\{3, 2, 0\}$

$\{4, 3, 1\}$



An example of a cyclic design

Cyclic P_3-design of order 5

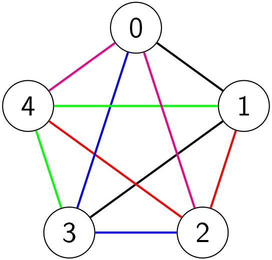
$\{1, 0, 3\}$

$\{2, 1, 4\}$

$\{3, 2, 0\}$

$\{4, 3, 1\}$

$\{0, 4, 2\}$



Cyclic P_3-design of order 5

Edge length

- Let $V(K_n) = \{0, 1, \ldots, n-1\}$

Edge length

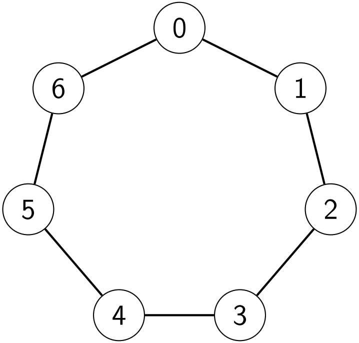
- Let $V(K_n) = \{0, 1, \ldots, n-1\}$
- The *length* of edge $xy \in E(K_n)$ is $\min(|x-y|, n-|x-y|)$

Edge length

- Let $V(K_n) = \{0, 1, \ldots, n-1\}$
- The *length* of edge $xy \in E(K_n)$ is $\min(|x-y|, n-|x-y|)$
- If the length of xy is $n-|x-y|$ or equivalently, if $|x-y| > \lfloor \frac{n}{2} \rfloor$, then we call xy a *wrap-around* edge

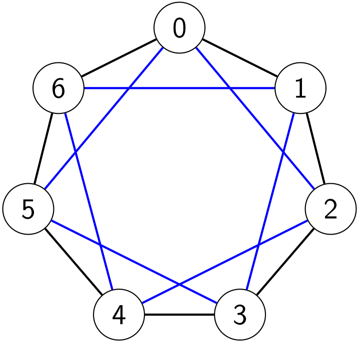
Edge lengths of K_7

- length 1



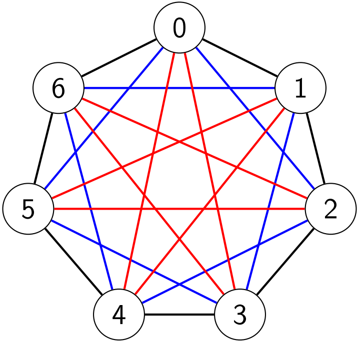
Edge lengths of K_7

- length 1
- length 2



Edge lengths of K_7

- length 1
- length 2
- length 3



Edge Length and Cyclic Decomposition

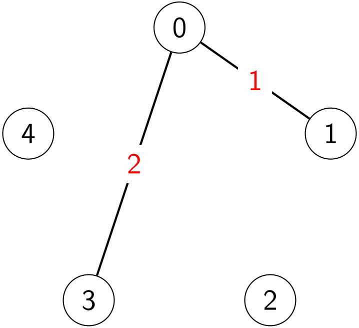
- Notice that edge length is preserved by the permutation $v \mapsto v + 1$ on $V(K_n)$

Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v + 1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)

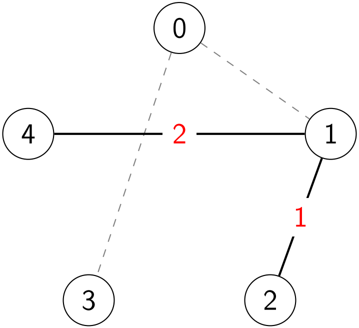
Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v+1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



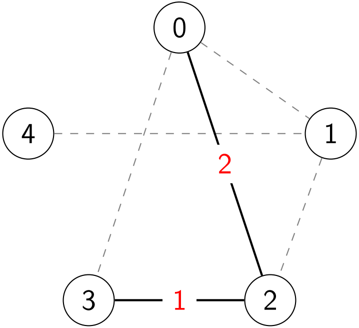
Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v + 1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



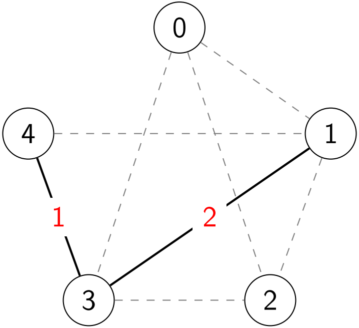
Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v + 1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



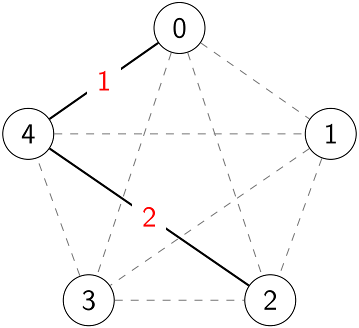
Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v+1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



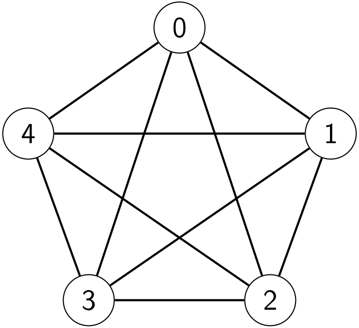
Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v + 1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



Edge Length and Cyclic Decomposition

- Notice that edge length is preserved by the permutation $v \mapsto v+1$ on $V(K_n)$
- Also, when n is odd, edge length partitions $E(K_n)$ into $\frac{n-1}{2}$ (the number of lengths) sets of size n (the number of edges of each length)



σ^{+-}-labelings

- Recall the length of $xy \in E(K_n)$ is $\min(|x-y|, n-|x-y|)$.

σ^{+-}-labelings

- Recall the length of $xy \in E(K_n)$ is $\min(|x-y|, n-|x-y|)$.
- A σ-labeling is a ρ-labeling such that the length of every edge $xy \in E(K_n)$ is $|x-y|$.

σ^{+-}-labelings

- Recall the length of $xy \in E(K_n)$ is $\min(|x-y|, n-|x-y|)$.
- A σ-labeling is a ρ-labeling such that the length of every edge $xy \in E(K_n)$ is $|x-y|$.
- Freyberg and Tran introduced the following restricted σ-labeling in 2020.

Definition

Let G be a bipartite graph with m edges and bipartition $V(G) = A \cup B$. A σ^{+-}-*labeling* of G is a σ-labeling with:

1. $f(a) < f(b)$ for every edge $ab \in E(G)$ with $a \in A$ and $b \in B$
2. $f(a) - f(b) \neq m$ for all $a, b \in V(G)$
3. $f(v) \notin \{2m-1, 2m\}$ for all $v \in V(G)$

σ^{+-}-labelings

Theorem (Freyberg, Tran, 2020)

Let G be a graph with m edges and a σ^{+-}-labeling such that the edge of length m is a pendant edge. Then there exists cyclic G-decompositions of K_{2mt} and K_{2mt+1} for every positive integer t.

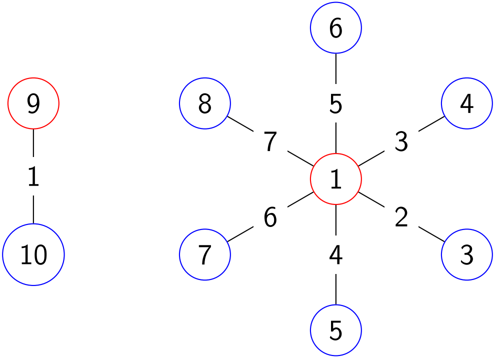
7 edge forest designs

- Recall that if G has m edges, then there exists a G-design of order n only if n is idempotent modulo $2m$.
- So F is a forest on 7 edges, there exists an F-design of order n only if $n \equiv 0, 1, 7,$ or $8 \pmod{14}$, since those are all the idempotents in $\mathbb{Z}_{14}$.
- So by Freyberg and Tran, if there exists a σ^{+-}-labeling of all forests F on 7 edges, then there exists a F-design of order $2mt$ and $2mt + 1$ for all $t > 0$.

7 edge forest designs

- The matching on 7-edges: $\bigsqcup_{i=1}^{7} P_2$ we solved by De Werra in 1970.
- This summer I found a σ^{+-} labeling of all forests on seven edges, up to isomorphism except $\bigsqcup_{i=1}^{7} P_2$. There are 46 total excluding the matching.

7 edge forest designs



A σ^{+-}-labeling of $S_6 \cup P_2$

7 edge forest designs

What about the cases where $n \equiv 7$ or $8 \pmod{14}$?

7 edge forest designs

What about the cases where $n \equiv 7$ or $8 \pmod{14}$?

- This is more complicated. σ^{+-}-labelings work because there are only lengths $\{0, \ldots, 7\}$ in K_{14} and so for for K_{14t} we can simply increase the size of the B partite set on our σ^{+-}-labeling by $7i$ for each $1 \leqslant i \leqslant t$ where and then click these new labelings by 1 to get our G-design of order $14t$.

7 edge forest designs

What about the cases where $n \equiv 7$ or $8 \pmod{14}$?

- This is more complicated. σ^{+-}-labelings work because there are only lengths $\{0, \ldots, 7\}$ in K_{14} and so for for K_{14t} we can simply increase the size of the B partite set on our σ^{+-}-labeling by $7i$ for each $1 \leqslant i \leqslant t$ where and then click these new labelings by 1 to get our G-design of order $14t$.
- This idea also gives G-designs of order $14t + 1$ where the new node is labeled ∞, and the lengths are still $\{0, \ldots, 7\}$ since $\lfloor \frac{15}{2} \rfloor = 7$.

7 edge forest designs

What about the cases where $n \equiv 7$ or $8 \pmod{14}$?

7 edge forest designs

What about the cases where $n \equiv 7$ or $8 \pmod{14}$?

- Lets look at the starting case we want for K_{14t+7} where $t > 0$ or K_n where $n \equiv 7 \pmod{14}$. This is K_{21}.
- $\lfloor \frac{21}{2} \rfloor = 10$, so we have lengths $\{1, \ldots, 10\}$. This means we can't simply use a labeling which we click by 1 because a 7 edge forest can only fit 7 distinct lengths…
- Well, what if we can account for counting edges of some lengths $\{a, b, c\}$ from $\{1, \ldots, 10\}$? Then we can simply click some variation of our σ^{+-}-labelings accounting for lengths $\{1, \ldots, 10\} \setminus \{a, b, c\}$ by 1.

Construction for 7-edge forest designs of order $14t + 7$ and $14t + 8$

- For each forest up to isomorphism: I found three pairwise edge distinct labelings F_1, F_2, F_3 consisting of only lengths $\{1, 2, 3\}$.
- These labelings also had another property, which requires a new edge function: $\ell_7^+ := f(\{u, v\}) = u + v \pmod 7$. This function partitions edges by the sum of their endpoints modulo 7. This induces a another partition on the edges: $\ell \times \ell_7^+$ where ℓ is the standard edge length function defined previously.
- Each partite set $P_{i,j}$ is the set of all edges of length i with endpoint sum j. This gives a new way to count edges of each length, it will allow us to click by 7 to collect all edges of any length.

Construction for 7-edge forest designs of order $14t+7$ and $14t+8$

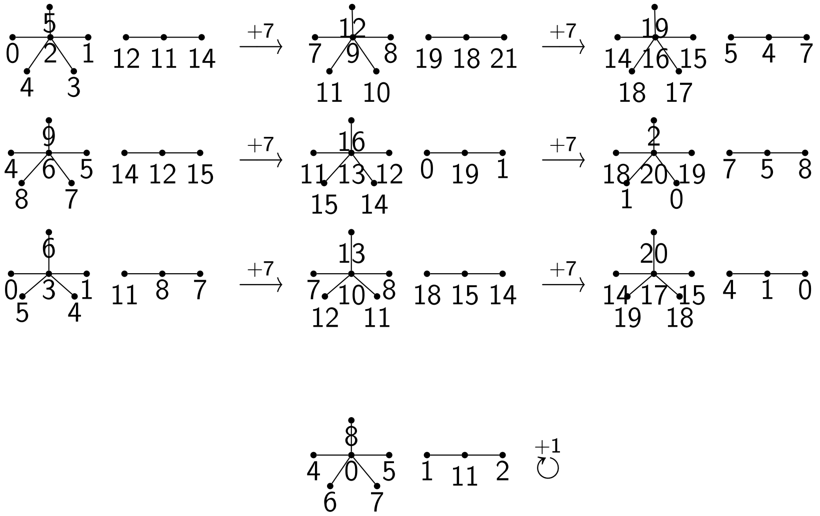
- Let E_i be the set of all edges of length i in K_{21}. Let us now take ℓ_7^+ to be a relation between edges in the natural way. This is clearly an equivalence relation.
- Then, E_i/ℓ_7^+ the set of all equivalence classes of edges of length i with respect to ℓ_7^+; $E_i/\ell_7^+ = \{[e_0],[e_1],\ldots,[e_6]\}$ where $\ell(e_j) = i$ for all $0 \leqslant j \leqslant 6$.

An example:
$[\{1,2\}]_{\ell_7^+} = \{\{1,2\},\{8,9\},\{15,16\}\}$ and if we click $\{1,2\}$ by 7 we get the whole equivalence class. The equivalence class is in fact a partite set of $\ell \times \ell_7^+$ on the edges of K_{21}.

Construction for 7-edge forest designs of order $14t + 7$ and $14t + 8$

- So we see this equivalence relation in fact induces a group action on the edges in E_i. We use use unions of representatives of these equivalence classes to build 'blocks' which will contain all edges of lengths $\{1, 2, 3\}$ of K_{21} between then in order to collect all these edges and complete our design.
- Recall: For each forest up to isomorphism except $S_6 \cup P_2$, I found three pairwise edge distinct labelings F_1, F_2, F_3 consisting of only lengths $\{1, 2, 3\}$, with the added property that they are all pairwise $\ell \times \ell_7^+$-edge disjoint.
- Then, we can see that we will collect all 7 equivalence classes for edges of lengths $1, 2, 3$ and clicking by 7 all edges of lengths $1, 2, 3$. Then we simply adapt a σ^{+-}-labeling to contain edges of lengths $4, 5, 6, 7, 8, 9, 10$, and click that by 1 to get our decompositions.

An example of such a $S_5 \sqcup P_3$-design of order 21



Summary

- Notice that that there are no wrap-around edges in this decomposition. This means we can click the graphs in the first column of the previous figure by 7 a total of $|7|$ in $\mathbb{Z}_{14t+7}$ for any $t > 0$ to get all length $1, 2, 3$ edges in K_{14t+7}
- Then we do the same thing we did for σ^{+-}-labelings with the graph in the fourth row, just make copies of it for the next new edge lengths for each t and click all those by 1
- We simply add one more labeling for K_{14t+8} and a new edge length ∞ that goes with the new ∞ node.
- We had to take a different combinatorial approach for $S_6 \sqcup P_2$ where we broke off edges from 7-star decompositions of K_{21} proven to exist by P. Cain in 1974.

Thank you!

What's left?

- I still have to finish the labelings for K_{22} which give the decomposition for K_{14t+8} where $t > 0$. Going smoothly.
- We have to prove we can break off edges for all graphs in the design constructions by P. Cain to get our $S_6 \sqcup P_2$-designs.
- Once done with the above we can probably extend this strategy to other 7 edge graphs to complete all 7-edge designs of order n. There are similar labelings to σ^{+-} and our approach to the other idempotent n's are not family specific.

Thank you all for coming!