What is a graph?

An introduction to graphs. Graphs are mathematical objects comprising of a vertex set V and an edge set E, where the edges connect the vertices.

Aided Decomposition of the Complete Digraph into Orientations of $K_4 - e$ with a Double Edge

Hanson Hao, Claudia Zhu
IMSA

How do graphs decompose?

Subgraphs and decompositions. If F and H are digraphs with D as a subgraph of H (if V(D) is a subset of V(H) and E(D) is a subset of E(H)), then a D-decomposition of H is a partition of the arc set of H into subgraphs that are isomorphic to D, called D-blocks.

Spectrum. The spectrum for a digraph D is the set of all eigenvalues for which a D-decomposition of $K_n$ exists.

Isomorphism. If V($K'_n$) is all of the natural integers, and D is a subgraph of $K_n$ then there exists a $K_n$-decomposition of $K_n - D$.

Cyclic. A D-decomposition of $K_n$ is called cyclic if clicking D preserves the D-blocks of the decomposition.

A Cyclic Decomposition of $K_5$

Examples of Our Decompositions

Computer Algorithm

Graphs and D-blocks become sets of (indegree, outdegree) ordered pairs. For example, $K'_6$ becomes $\{(5,5), (5,5), (5,5), (5,5), (5,5), (5,5)\}$.

General “ Blow-up ” Method

Uses the following theorems:
If $x > 0$, then a $(K_4 - e, K_4 - e)$-decomposition of $K_n$ exists.

- A majority of our cases yielded partial results, as some specific building blocks were not found and so the general blow-up method failed.
- We want to optimize code to reduce runtime and memory usage.
- Investigate specific decomposition attempts to see why they fail.

Examples of Our Decompositions

Cyclic D118- Decomposition of $K'_4$

Cyclic D135- Decomposition of $K'_5$

On a Computer-Aided Decomposition of the Complete Digraph into Orientations of $K_4 - e$ with a Double Edge

Hanson Hao, Claudia Zhu
IMSA

Abstract

Let D be any of the 5 non-symmetric digraphs obtained by orienting the edges of $K_4 - e$ with a double edge (denoted thereafter by $K_4 - e^2$). We obtain some $(K'_n, D)$-designs for small values of n where n ≤ 6 aided by a C++ program. The C++ program was able to verify the nonexistence of results as well as construct some $(K'_n, D)$-designs. It also used a enumeration technique, where previous runs were stored and referenced, in order to reduce runtime. Furthermore, we established necessary and sufficient conditions for the existence of a $(K'_n, D)$-design for some of the general constructions using the aforementioned small cases and a “blow-up” construction. Partial results as well as some nonexistence results are established for the remaining digraphs. Future work on this project may be done by developing more of the partial results and improve the code to reduce both memory usage and runtime, possibly by the use of parallel processing.

Our Research Question

We want to find the spectrum for D such that D is an orientation of $K'_n$ with a double edge $(K'_n, D^2)$. The computer graph file is shown below.

When we orient the graph, one of the two double edges from $v$ to $u$ must be directed towards $u$ and the other must be directed towards $v$.

Digraphs are named using the conventions of directed graphs by Read and Wilson.

Our Design

We eliminated any digraphs that were isomorphic or reverses of each other and ended with the above 5 digraphs of interest from the original 11 possible orientations.

Other Observations

Reversing digraphs. We can reverse the orientation of a digraph D by changing the direction of the arrow on each arc. We denote the reverse orientation as Rev(D).

Observation 1: We observe that D and D are digraphs, then D and D are digraphs.

Observation 2: We observe that D is a digraph, then D decomposes $K'_n$, (if and only if Rev(D) decomposes Rev($K'_n$)).

Conclusion

We obtained some $(K'_n, D)$-designs for small values of n where n ≤ 6. It also used a enumeration technique, where previous runs were stored and referenced, in order to reduce runtime. Furthermore, we established necessary and sufficient conditions for the existence of a $(K'_n, D)$-design for some of the general constructions using the aforementioned small cases and a “blow-up” construction. Partial results as well as some nonexistence results are established for the remaining digraphs. Future work on this project may be done by developing more of the partial results and improve the code to reduce both memory usage and runtime, possibly by the use of parallel processing.