Localized Structure in Graph Decompositions

by

Flora Caroline Bowditch
B.Sc., University of Victoria, 2017

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Mathematics and Statistics

Localized Structure in Graph Decompositions

by

Flora Caroline Bowditch
B.Sc., University of Victoria, 2017

Supervisory Committee

Dr. Peter Dukes, Supervisor
(Department of Mathematics and Statistics)

Dr. Gary MacGillivray, Department Member
(Department of Mathematics and Statistics)

Supervisory Committee

Dr. Peter Dukes, Supervisor
(Department of Mathematics and Statistics)

Dr. Gary MacGillivray, Department Member
(Department of Mathematics and Statistics)

ABSTRACT

Let $v \in \mathbb{Z}^+$ and G be a simple graph. A G-*decomposition* of K_v is a collection $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$ of subgraphs of K_v such that every edge of K_v occurs in exactly one of the subgraphs and every graph $F_i \in \mathcal{F}$ is isomorphic to G. A G-decomposition of K_v is called *balanced* if each vertex of K_v occurs in the same number of copies of G. In 2011, Dukes and Malloch provided an existence theory for balanced G-decompositions of K_v. Shortly afterwards, Bonisoli, Bonvicini, and Rinaldi introduced *degree-* and *orbit-balanced* G-decompositions. Similar to balanced decompositions, these two types of G-decompositions impose a local structure on the vertices of K_v.

In this thesis, we will present an existence theory for degree- and orbit-balanced G-decompositions of K_v. To do this, we will first develop a theory for decomposing K_v into copies of G when G contains coloured loops. This will be followed by a brief discussion about the applications of such decompositions. Finally, we will explore an extension of this problem where K_v is decomposed into a family of graphs. We will examine the complications that arise with families of graphs and provide results for a few special cases.

Table of Contents

List of Figures

Acknowledgements

To my supervisor, Peter, I'm so glad I decided to do my master's degree with you. I couldn't have asked for a better collaborator and mentor.

To the entire math department, thank you for providing me with such a wonderful work environment. It's been amazing getting to know and work with so many of you.

To all of my parents and grandparents, thank you for always supporting my goals. I wouldn't be here without your constant love and encouragement.

To the entire Wheeler family, thank you for your endless support over the last five years. I appreciate it more than I can express.

To Michael, Hermione, Madison, India, Jeremy, Matti, Lucy, David, Lauren, Chloe, Dan, and Jane, thank you for keeping me sane along this journey. You are my closest friends and greatest cheerleaders.

Finally, I'd like to thank Mackenzie. My love, I'm so glad we got to experience all of this together. Thank you for your never-ending patience (which, let's face it, I didn't deserve a lot of the time.) It's been so fun working through two degrees with you. I can't wait to see what's next!

Dedication

For Fred —I hope you always pursue your passions (even if the rest of the world thinks you're crazy!)

Chapter 1

Introduction

A *graph* is an ordered pair $G = (V, E)$, where V is a set of points, called *vertices*, and E is a set of 2-element subsets of V, called *edges*. When referring to a particular graph G, we may write $V(G)$ and $E(G)$ to represent its vertex set and edge set, respectively. Each edge $e = \{u, v\}$ of G will be denoted by uv or vu. For vertices u and v, we say that u is *adjacent* to v if $uv \in E$. In this case we will also say that e is *incident* with both u and v.

The type of graph we have just described is known as a *simple graph*. There are many generalizations of the definition given above, and we will encounter several of them in this thesis. A *directed graph* or *digraph* has an edge set consisting of ordered pairs. These directed edges are commonly referred to as *arcs*, and we will write $G = (V, A)$ instead, where A is the arc set. A *multigraph* has a multiset of edges, allowing multiple edges between pairs of vertices. Finally, a *pseudograph* is a multigraph that can have edges from a vertex back to itself, which are known as *loops*. In Figure 1.1, we give an example of each of these graphs. Throughout this thesis, we will make it clear what type of graph is being dealt with at any given point.

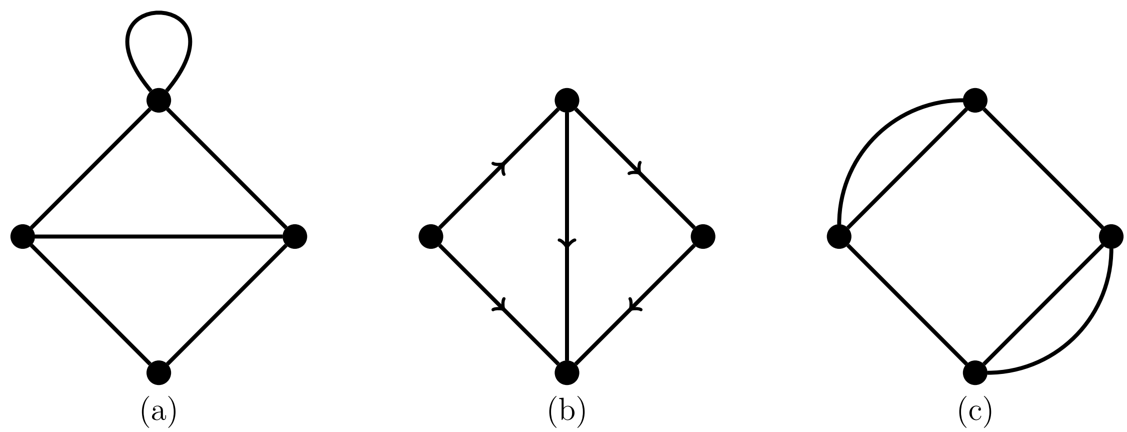

Figure 1.1: (a) a pseudograph; (b) a directed graph; (c) a multigraph.

In this thesis, we will be examining decompositions of graphs. Let $\mathcal{G}$ be a family of undirected graphs. A *graph decomposition* of a graph H is a collection $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$ of subgraphs of H such that every edge of H occurs in exactly one of the subgraphs. Given a family of graphs $\mathcal{G}$, if every graph $F_i \in \mathcal{F}$ is isomorphic to some graph in $\mathcal{G}$, then we call $\mathcal{F}$ a *$\mathcal{G}$-decomposition* of H. If $\mathcal{G} = \{G\}$ for some fixed graph G, then we say that $\mathcal{F}$ is a *G-decomposition* of H. In this case, we say that H is *G-decomposable*. We illustrate these definitions in Example 1.1. Certain graph decompositions are closely related to block designs, which will be discussed later on. For this reason, the copies of G in a G-decomposition may sometimes be referred to as *blocks*.

For positive integers v and λ, we will define K_v^λ to be the multigraph on v vertices with exactly λ edges between every pair of vertices. In the case $\lambda = 1$, we have the *complete graph on v vertices*, denoted K_v. In this thesis, we will be focusing exclusively on decompositions of K_v^λ. For our main result, we will look at decomposing K_v^λ into copies of a single graph G. In some of the later chapters, we will consider decomposing K_v^λ into a family of graphs $\mathcal{G}$ and explore some specific cases there.

Example 1.1. Suppose we want to decompose K_7 into copies of K_3. Such a decomposition is illustrated in Figure 1.2, where each copy of K_3 has been given a unique colour. It is easy to see that every edge of K_7 is contained in exactly one copy of K_3.

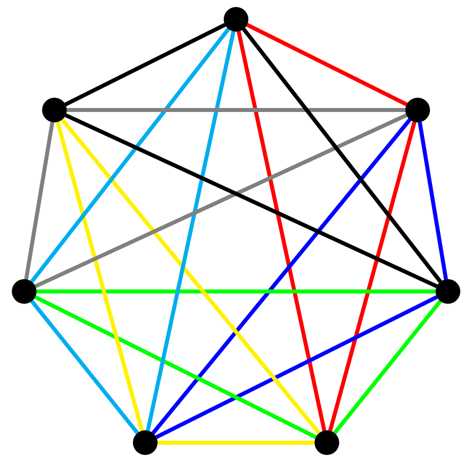

Figure 1.2: A K_3-decomposition of K_7.

1.1 History and Motivation

A specific type of graph decomposition known as a balanced graph decomposition was first introduced by Hell and Rosa in [17]. Given a simple graph G, a G-decomposition of K_v is called *balanced* if every vertex of K_v appears in an equal number of copies of G. In [10] and [21], Dukes and Malloch showed that for a fixed graph G, there is a balanced G-decomposition of K_v^λ for all sufficiently large integers v satisfying certain necessary conditions. They first developed a theory for decomposing K_v^λ into copies of G when G contains loops. Designating these loops in a specific way, they were able to obtain the result.

Recently, Bonisoli, Bonvicini, and Rinaldi introduced two slightly more restrictive types of balanced graph decompositions in [3]: degree-balanced and orbit-balanced.

For a given graph G, let $D(G)$ be the set of all degrees of the vertices of G. For each $d \in D(G)$, the subset of all vertices of degree d is the *degree-class* defined by d. These degree classes form a set of equivalence classes which partition $V(G)$. For every $d \in D(G)$ and every $u \in V(K_v)$, let $r_d(u)$ denote the number of blocks in a G-decomposition of K_v containing u as a vertex of degree d. A G-decomposition of K_v is called *degree-balanced* if for each $d \in D(G)$, $r_d(u)$ is independent of u. That is, there exists a constant positive integer r_d such that $r_d(u) = r_d$ for all $u \in V(K_v)$.

For a given graph G, an *automorphism* of G is a permutation σ of the vertex set $V(G)$, such that for any $u, v \in V(G)$, $uv \in E(G)$ if and only if $\sigma(u)\sigma(v) \in E(G)$. Equivalently, an automorphism is a graph isomorphism from G to itself. The composition of two automorphisms is another automorphism, and the set of automorphisms of G, under the composition operation, forms a group known as the *automorphism group* of G. The *orbit* of a vertex $u \in V(G)$ is the set of all vertices $\sigma(u)$ such that σ is an automorphism of G. That is, the orbit of u is the set of all vertices that u can be mapped to under some automorphism. The equivalence classes of $V(G)$ under the action of the automorphisms are called *orbit-classes*.

Let $A(G)$ be the set of orbit-classes of G under its automorphism group. For each $a \in A(G)$ and every $u \in V(K_v)$, let $r_a(u)$ denote the number of blocks in a G-decomposition of K_v containing u as a vertex in orbit a. A G-decomposition of K_v is called *orbit-balanced* if for each $a \in A(G)$, $r_a(u)$ is independent of u. That is, there exists a constant positive integer r_a such that $r_a(u) = r_a$ for all $u \in V(K_v)$. Since each orbit-class contains vertices of a common degree, it is clear that orbit-balanced graph decompositions are also degree-balanced. It is also easy to see that both degree- and orbit-balanced graph decompositions are balanced. However, the converse of each of these statements is not true. Bonvicini gives many counterexamples to these in [5], two of which are given below.

Example 1.2. To see that not all balanced graph decompositions are degree-balanced, take K_5 and consider decomposing it into copies of the graph G shown in Figure 1.3. Here, each copy of G in the decomposition has been given a unique colour. Every vertex of K_5 appears in exactly two copies of G, so the decomposition is certainly balanced. However, vertex d appears twice as a vertex of degree two in the copies of G, and all the other vertices appear once as a vertex of degree one and once as a vertex of degree three. Thus, the decomposition is not degree-balanced.

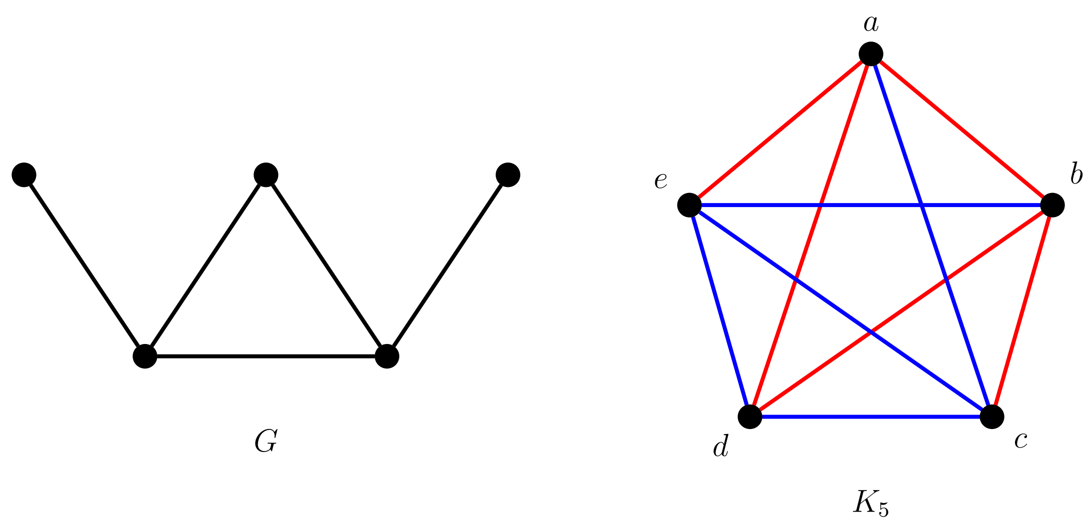

Figure 1.3: A balanced G-decomposition of K_5 which is not degree-balanced.

Example 1.3. To see that not all degree-balanced graph decompositions are orbit-balanced, take K_7 and consider decomposing it into copies of the graph $G = P_3 \cup P_2$ shown in Figure 1.4. Again, each copy of G in the decomposition has been given a unique colour. Every vertex of K_7 appears as a vertex of degree two exactly once, and as a vertex of degree one exactly four times. Thus, this decomposition is degree-balanced. Observe that G has three orbit-classes: $a_1 = \{x_1, x_2\}$, $a_2 = \{y_1\}$, and $a_3 = \{z_1, z_2\}$. In this decomposition, vertices 0, 1, 2, 3, and 6 appear twice as a vertex in a_1, once as a vertex in a_2, and twice as a vertex in a_3. However, vertex 4 appears once as a vertex in a_1, once as a vertex in a_2, and three times as a vertex in a_3; vertex 5 appears three times as a vertex in a_1, once as a vertex in a_2, and once as a vertex in a_3. Thus, the decomposition is not orbit-balanced.

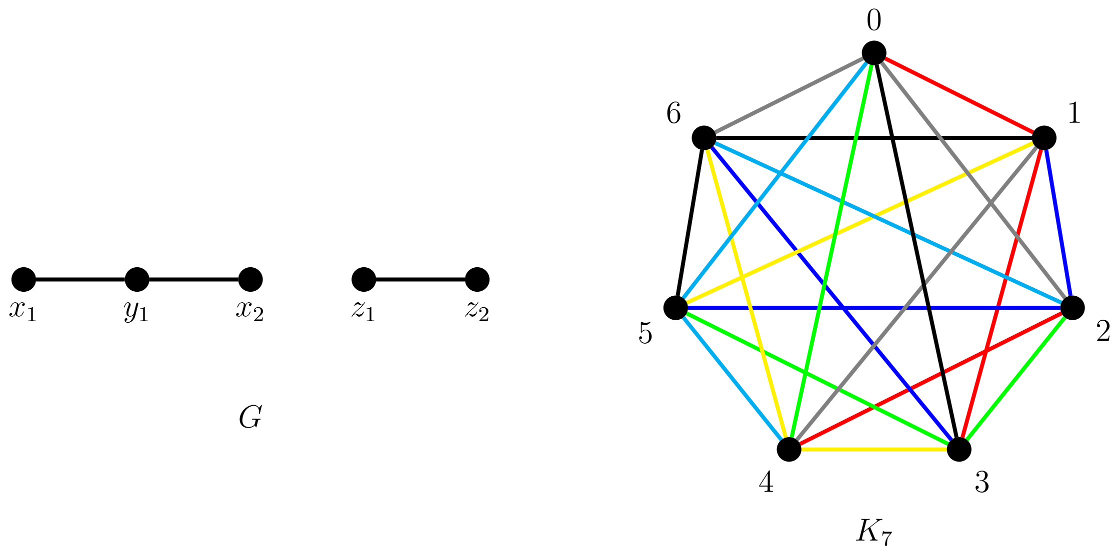

Figure 1.4: A degree-balanced G-decomposition of K_7 which is not orbit-balanced.

In [3], Bonisoli, Bonvicini, and Rinaldi determined the necessary conditions for both degree- and orbit-balanced graph decompositions of K_v. For a degree-balanced graph decomposition, it is necessary that the following two equations are satisfied for every $u \in V(K_v)$:

$$r(u) = \sum_{d \in D(G)} r_d(u) = \sum_{d \in D(G)} r_d \quad \text{and} \quad \sum_{d \in D(G)} d r_d(u) = \sum_{d \in D(G)} d r_d = v - 1,$$

where $r(u)$ denotes the number of copies of G in which u appears. Similarly, for an orbit-balanced graph decomposition, it is necessary that the following two equations are satisfied for every $u \in V(K_v)$:

$$r(u) = \sum_{a \in A(G)} r_a(u) = \sum_{a \in A(G)} r_a \quad \text{and} \quad \sum_{a \in A(G)} d(a) r_a(u) = \sum_{a \in A(G)} d(a) r_a = v - 1,$$

where $d(a)$ denotes the degree of the vertices in orbit a.

In [5], Bonvicini determined for which values of v the graph K_v admits a balanced, degree-balanced, or orbit-balanced G-decomposition for each graph G on five vertices. They also analyzed these decompositions to determine which values of v give a balanced decomposition which is not degree-balanced, and a degree-balanced decomposition which is not orbit-balanced. Similar to what Dukes and Malloch did in [10] and [21], we will develop a theory for decomposing K_v^λ into copies of a graph G when G contains *coloured* loops. Using this theory, we will obtain the existence of both degree- and orbit-balanced decompositions for all large admissible integers v.

1.2 Necessary Conditions

As mentioned before, our main result deals with decomposing K_v^λ into copies of some graph G, where G contains coloured loops. Since we will have various loop colours involved, our necessary conditions will be a bit more complicated than those given in [10] and [21]. Let G be an undirected graph with n vertices, m edges, and with c different loop colours. For any vertex $u \in V(G)$, let d_u denote the number of standard edges incident with u, and let $e_{u,i}$ denote the number of loops of colour i at u for $i = 1, \ldots, c$. Let $\ell_i = \sum_{u \in V(G)} e_{u,i}$ denote the total number of loops of colour i in G. We want to determine the necessary conditions for the existence of a G-decomposition of K_v^λ. As in [21] and [26], we have the following (global) necessary condition:

$$\lambda v(v-1) \equiv 0 \pmod{2m} \tag{1.1}$$

as the number of standard edges in G must divide the number of standard edges in K_v^λ.

Since our graph G contains loops, we must append loops to the vertices of K_v^λ if we are going to decompose it. Let $K_v^{[\mu_1,\mu_2,\ldots,\mu_c;\lambda]}$ denote the multigraph on v vertices with λ edges between every pair of vertices and μ_i loops of colour i at every vertex. For the sake of notation, we may sometimes write $K_v^{[\mu_1,\mu_2,\ldots,\mu_c;\lambda]}$ as $K_v^{[\boldsymbol{\mu};\lambda]}$ where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_c)$. Counting as in [21], we see that we require

$$\mu_i = \frac{\lambda \ell_i (v-1)}{2m} \tag{1.2}$$

loops of colour i at each vertex in $K_v^{[\boldsymbol{\mu};\lambda]}$ for $1 \le i \le c$.

Locally, we require simultaneous decomposition of the edges and loops at each vertex of $K_v^{[\boldsymbol{\mu};\lambda]}$. That is, we need an integral solution $\{s_u\}$ to

$$\sum_{u \in V(G)} s_u d_u = \lambda(v-1) \quad \text{and} \tag{1.3}$$

$$\sum_{u \in V(G)} s_u e_{u,i} = \mu_i \tag{1.4}$$

for $1 \le i \le c$. In [26], Wilson showed that the condition in (1.3) can be reduced to

$$\lambda(v-1) \equiv 0 \pmod{g} \tag{1.5}$$

where $g = \gcd\{d_u : u \in V(G)\}$.

Now (1.2), (1.3), and (1.4), together with the fact that we require $\mu_i \in \mathbb{Z}$, gives us

$$\lambda(v-1)\begin{bmatrix}1\\ \frac{\ell_1}{2m}\\ \vdots\\ \frac{\ell_c}{2m}\end{bmatrix} \in \operatorname{span}_{\mathbb{Z}}\left\{\begin{bmatrix}d_u\\ e_{u,1}\\ \vdots\\ e_{u,c}\end{bmatrix}\right\}, \tag{1.6}$$

where u ranges over all vertices of G. Putting (1.1) and (1.6) together, we obtain our necessary conditions:

$$\begin{aligned}\lambda v(v-1) &\equiv 0 \pmod{2m}\\ \lambda(v-1) &\equiv 0 \pmod{\alpha}\end{aligned} \tag{1.7}$$

where α is the least positive integer such that

$$\alpha\begin{bmatrix}1\\ \frac{\ell_1}{2m}\\ \vdots\\ \frac{\ell_c}{2m}\end{bmatrix} \in \operatorname{span}_{\mathbb{Z}}\left\{\begin{bmatrix}d_u\\ e_{u,1}\\ \vdots\\ e_{u,c}\end{bmatrix}\right\}. \tag{1.8}$$

For a given graph G, we will say that the integers λ and v are *admissible* for G if they satisfy (1.7). It is important to note that these necessary conditions will give different admissible values of λ and v depending on where loops are placed on the underlying graph G.

Example 1.4. This example will demonstrate the importance of how coloured loops are distributed across vertices in the given graph. Figure 1.5 shows us two graphs, G and H, which are isomorphic in their standard edges, and contain the same number of red and blue loops. Their parameters are $n = 3$, $m = 2$, $\ell_1 = 2$, and $\ell_2 = 2$, where ℓ_1 is the number of red loops and ℓ_2 is the number of blue loops.

Figure 1.5: Two distinct graphs with the same set of parameters.

For G, (1.8) tells us that α is the least positive integer satisfying

$$\alpha \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \in \operatorname{span}_{\mathbb{Z}} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

which gives us $2 \mid \alpha$. It turns out that $\alpha = 2$, as

$$\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = 0 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

So our necessary conditions for G are

$$\lambda v(v-1) \equiv 0 \pmod 4 \quad \text{and}$$
$$\lambda (v-1) \equiv 0 \pmod 2.$$

For H, we have

$$\alpha \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \in \operatorname{span}_{\mathbb{Z}} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \right\}$$

which again gives us $2 \mid \alpha$. However, $\alpha = 2$ does not give us integer coefficients for our set of vectors. It turns out that $\alpha = 4$ as

$$\begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

Here we notice that $\alpha = 4 = 2m$. From (1.8), it is clear that $2m$ is always a solution as

$$\begin{bmatrix} 2m \\ \ell_1 \\ \vdots \\ \ell_c \end{bmatrix} = \sum_{u \in V(G)} \begin{bmatrix} d_u \\ e_{u,1} \\ \vdots \\ e_{u,c} \end{bmatrix}.$$

Thus, $\alpha \mid 2m$ for any graph with m edges. So our necessary conditions for H are

$$\lambda v(v-1) \equiv 0 \pmod 4 \quad \text{and}$$
$$\lambda(v-1) \equiv 0 \pmod 4.$$

Now the parameters $\lambda = 2$ and $v = 4$ satisfy the necessary conditions for G, but they do not satisfy the necessary conditions for H. Solving for μ_1 and μ_2 with $\lambda = 2$ and $v = 4$, we obtain $\mu_1 = \mu_2 = 3$. Thus, there is no H-decomposition of $K_4^{[3,3;2]}$, but there is in fact a G-decomposition of $K_4^{[3,3;2]}$. The graph of $K_4^{[3,3;2]}$ is given in Figure 1.6 with its vertices labeled 0 through 3. With $V(G) = \{u_1, u_2, u_3\}$ as in Figure 1.5, we can take the following as our copies of G:

$$V(G_1) = \{0, 1, 3\} \quad V(G_2) = \{1, 2, 0\} \quad V(G_3) = \{2, 3, 1\}$$
$$V(G_4) = \{3, 0, 2\} \quad V(G_5) = \{0, 3, 2\} \quad V(G_6) = \{2, 1, 0\}.$$

Every vertex of $K_4^{[3,3;2]}$ appears exactly three times as vertices u_1 or u_2 (and hence has exactly three red and three blue loops) and every edge is covered exactly twice.

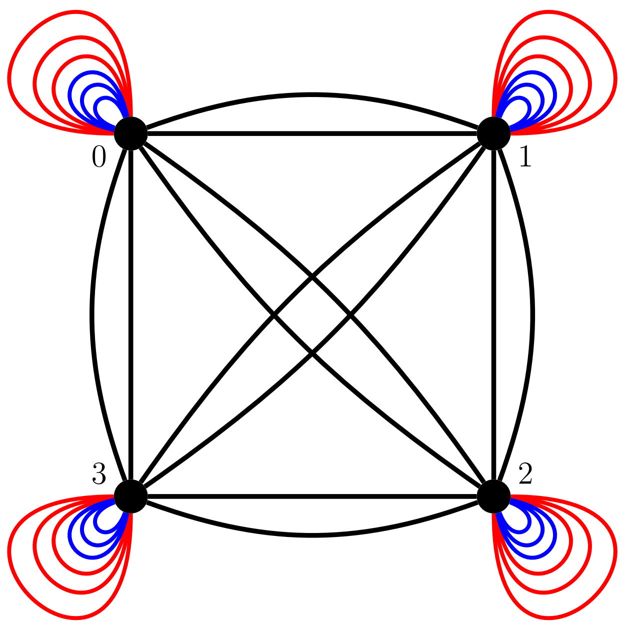

Figure 1.6: The graph of $K_4^{[3,3;2]}$.

1.3 Main Result

Now that we have determined the necessary conditions for our problem, we are ready to state our main result.

Theorem 1.5. *Let $\lambda \in \mathbb{Z}, \lambda \geq 0$. Suppose G is an undirected graph with n vertices, m edges, and ℓ_i loops of colour i for $1 \leq i \leq c$. Then there exists a G-decomposition of $K_v^{[\boldsymbol{\mu};\lambda]}$ for all sufficiently large integers v satisfying the necessary conditions given in (1.7).*

Chapter 3 is devoted to the proof of Theorem 1.5. In [21], Malloch's theory for graph decompositions with loops gave the asymptotic existence of balanced graph decompositions of K_v^λ. This was achieved by placing exactly one loop on every vertex of the given simple graph. By using coloured loops, we are able to extend our result to the more restrictive degree-balanced and orbit-balanced graph decompositions. This gives us the following two corollaries to Theorem 1.5.

Corollary 1.6. *Let $\lambda \geq 0$. Suppose G is a simple graph with n vertices, m edges, and degree set $D(G)$. Then there exists a degree-balanced G-decomposition of K_v^λ for all sufficiently large v satisfying (1.7) with $e_{u,d} = 1$ if deg$(u) = d$, and 0 otherwise.*

Corollary 1.7. *Let $\lambda \geq 0$. Suppose G is a simple graph with n vertices, m edges, and orbit-class set $A(G)$. Then there exists an orbit-balanced G-decomposition of K_v^λ for all sufficiently large v satisfying (1.7) with $e_{u,a} = 1$ if orb$(u) = a$, and 0 otherwise.*

The same proof can be used for both corollaries by simply interchanging degree-class and orbit-class.

Proof. Let $\lambda \geq 0$. Suppose G is a simple graph with n vertices, m edges, and degree set $D(G)$. For each $d \in D(G)$, let $e_{u,d}$ denote the number of loops of colour d at every vertex $u \in V(G)$. For each $u \in V(G)$, set $e_{u,d} = 1$ if *deg*$(u) = d$, and 0 otherwise. That is, designate a unique colour to each degree-class of G and give each vertex in that class exactly one loop of that colour. With these loops appended to G, Theorem 1.5 then tells us that there is a G-decomposition of $K_v^{[\boldsymbol{\mu};\lambda]}$ for all sufficiently large integers v satisfying (1.7). In such a decomposition, (1.2) tells us that for each $d \in D(G)$, we require

$$\mu_d = \frac{\lambda \ell_d (v-1)}{2m}$$

loops of colour d at every vertex in K_v^λ. For a given degree $d \in D(G)$, every vertex in that degree-class has exactly one loop of colour d. Hence, every vertex in K_v^λ must see the same number of vertices in that degree-class. Removing the loops from the resulting decomposition then yields a degree-balanced G-decomposition of K_v^λ. $\square$

Chapter 2

Background

2.1 Block Designs

The existence of block designs is closely related to the existence of certain graph decompositions. This relationship will be extremely useful in proving many of our results. Pairwise balanced designs will play an important role in Chapter 3 when we prove Theorem 1.5. Balanced incomplete block designs and θ-resolvable designs will be used in Chapter 5 when we examine decompositions involving families of graphs. In Chapter 6, we will look at how uniformly resolvable designs and cyclic designs might be used to obtain more graph decomposition results.

A *balanced incomplete block design* (BIBD) with parameters (v, k, λ) is an ordered pair $(V, \mathcal{B})$ where

- V is a set of v objects called *points*;
- $\mathcal{B}$ is a collection of (not necessarily distinct) k-subsets of V called *blocks*; and
- every pair of distinct points are contained together in exactly λ blocks.

We will usually refer to such a design as a BIBD(v, k, λ). A BIBD(v, k, λ) can be thought of as a partition of the edge set of K_v^λ into copies of K_k. This means that for given v, k, and λ, the existence of a BIBD(v, k, λ) also guarantees the existence of a K_k-decomposition of K_v^λ.

Example 2.1. Let $V = \mathbb{Z}_{11}$ and let $\mathcal{B} = \{x + \{1, 3, 4, 5, 9\} : x \in V\}$, where addition is distributed into the set and is modulo 11. This gives us a BIBD$(11, 5, 2)$. In terms of graph decompositions, we can think about taking the graph K_{11}^2 and labelling its vertices as the integers modulo 11. The set of blocks $\mathcal{B}$ generated by this block design tells us on which vertices of K_{11}^2 to place copies of K_5, producing a K_5-decomposition of K_{11}^2.

We have two necessary conditions for the existence of a BIBD(v, k, λ). There are, of course, many examples that show that these conditions are not sufficient.

Proposition 2.2. [7] *If there exists a BIBD*(v, k, λ)*, then*

$$\begin{aligned} \lambda v(v-1) &\equiv 0 \pmod{k(k-1)} \text{ and} \\ \lambda(v-1) &\equiv 0 \pmod{k-1}. \end{aligned} \tag{2.1}$$

This follows from the fact that the number of blocks is $\frac{\lambda v(v-1)}{k(k-1)}$, and every point belongs to exactly $\frac{\lambda(v-1)}{k-1}$ blocks. Another necessary condition for the existence of a BIBD(v, k, λ) was given by Fisher in 1940, and is commonly referred to as "Fisher's Inequality". For certain parameters satisfying the conditions in Proposition 2.2, Fisher's inequality can be used to rule out the existence of a BIBD(v, k, λ).

Theorem 2.3 (Fisher's Inequality)**.** [7] *In any BIBD*(v, k, λ) *with* $1 < k < v$*, it is necessary that* $b \geq v$*.*

A *pairwise balanced design* (PBD) is an ordered pair $(V, \mathcal{B})$ where $|V| = v$ and $|B| \in K$ for each $B \in \mathcal{B}$. Here, K is a set of sizes that the blocks may take on, and every pair of points belongs to exactly one block. Note that not all block sizes in K need to be used. Based on this definition, if a PBD(v, K) exists with blocks $\mathcal{B} = \{B_1, B_2, \ldots, B_t\}$, then K_v can be decomposed into subgraphs, each of which is a clique K_{k_i} with vertex set $B_i \in \mathcal{B}$ for some $k_i \in K$.

2.1.1 Resolvable Designs

A *parallel class* in a design is a set of blocks that partition the point set. A *resolvable* balanced incomplete block design is a BIBD(v, k, λ) whose blocks can be partitioned into parallel classes. In this case, we will write RBIBD(v, k, λ). Similarly, we can define a resolvable pairwise balanced design, which we denote as RPBD(v, K).

Example 2.4. We obtain an RBIBD$(9, 3, 1)$ by taking $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and

$$\begin{aligned}\mathcal{B} = \Big\{&\{1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{1,4,7\}, \{2,5,8\}, \{3,6,9\},\\ &\{1,5,9\}, \{2,6,7\}, \{3,4,8\}, \{1,6,8\}, \{2,4,9\}, \{3,5,7\}\Big\}.\end{aligned}$$

We have two necessary conditions for the existence of an RBIBD(v, k, λ).

Proposition 2.5. [7] *If there exists an RBIBD(v, k, λ), then*

$$\begin{aligned} v &\equiv 0 \pmod{k} \text{ and} \\ \lambda(v-1) &\equiv 0 \pmod{(k-1)}. \end{aligned} \tag{2.2}$$

This follows from the fact that the number of parallel classes is $\frac{\lambda(v-1)}{k-1}$, and the number of blocks in each parallel class is $\frac{v}{k}$. A parallel class in a PBD(v, K) is *uniform* if all blocks in the parallel class have the same size. A *uniformly resolvable design*, denoted URD(v, K, R), is an RPBD(v, K) such that all of the parallel classes are uniform. Here R is a list of integers (r_k) indexed by $k \in K$ such that there are exactly r_k parallel classes of size k.

A θ-*parallel class* in a design is a subcollection of blocks $\mathcal{A}$ such that every point in V belongs to exactly θ of the blocks in $\mathcal{A}$. A θ-*resolvable* balanced incomplete block design is a BIBD(v, k, λ) whose blocks can be partitioned into θ-parallel classes. In this case, we will write θ-RBIBD(v, k, λ). Similarly, we can define a θ-resolvable pairwise balanced design, which we denote as θ-RPBD(v, K). We have the following two necessary conditions for the existence of a θ-RBIBD(v, k, λ).

Proposition 2.6. [9] *If there exists a θ-RBIBD(v, k, λ), then*

$$\begin{aligned} \theta v &\equiv 0 \pmod{k} \text{ and} \\ \lambda(v-1) &\equiv 0 \pmod{\theta(k-1)}. \end{aligned} \tag{2.3}$$

This follows from the fact that the number of θ-parallel classes is $\frac{\lambda(v-1)}{\theta(k-1)}$, and the number of blocks in each θ-parallel class is $\frac{\theta v}{k}$. We then have the following result due to Dukes, Ling, and Malloch.

Theorem 2.7. [9] *Let $k \geq 2$, $\theta \geq 1$, and $\lambda \geq 0$ be integers. For sufficiently large v, there exists a θ-RBIBD(v, k, λ) if and only if (2.3) holds.*

2.1.2 Cyclic Designs

A (v, k, λ)-*difference set* is a k-subset $D \subseteq \mathbb{Z}_v$ such that each nonzero element of $\mathbb{Z}_v$ occurs exactly λ times among the $k(k-1)$ differences of elements in D.

Example 2.8. Consider the set $\{0, 1, 3\}$. This is a $(7, 3, 1)$-difference set, as taking the differences (modulo 7) we obtain

$$\begin{aligned} 0-1 &\equiv 6, \quad 0-3 \equiv 4, \quad 1-0 \equiv 1, \\ 1-3 &\equiv 5, \quad 3-0 \equiv 3, \quad 3-1 \equiv 2. \end{aligned}$$

It isn't hard to see that every translate of a (v, k, λ)-difference set modulo v gives another (v, k, λ)-difference set. The collection of translates of a (v, k, λ)-difference set yields a BIBD(v, k, λ). We can simply think of this as the cyclic development of a base block modulo v. If $D_1, D_2, \ldots D_t$ are (not necessarily distinct) k-subsets of $\mathbb{Z}_v$ such that the $tk(k-1)$ numbers arising from taking all possible differences of elements in each set contains every nonzero element of $\mathbb{Z}_v$ exactly λ times, then $\mathcal{D} = \{D_1, D_2, \ldots D_t\}$ is a (v, k, λ)-*difference system*.

Example 2.9. Consider the $(7, 3, 1)$-difference set given in Example 2.8. Taking all translates modulo 7 yields the 3-subsets

$$\{0,1,3\}, \quad \{1,2,4\}, \quad \{2,3,5\}, \quad \{3,4,6\}, \quad \{4,5,0\}, \quad \{5,6,1\}, \quad \{6,0,2\}$$

which form a BIBD$(7, 3, 1)$. Taking all possible differences of elements in each of these sets, we obtain every nonzero element of $\mathbb{Z}_7$ exactly seven times. So this is also a $(7, 3, 7)$-difference system.

A BIBD(v, k, λ) is called *cyclic* if its blocks can be partitioned into sets of blocks, each of which has been obtained by cyclic development. In Example 2.9, we built a cyclic BIBD$(7, 3, 1)$ by simply developing the block $\{0, 1, 3\}$ modulo 7. Thus, we can obtain a cyclic BIBD(v, k, λ) from a (v, k, λ)-difference set.

2.2 Network Flows

In Chapter 5, we will look at decomposing K_v^λ into a family $\mathcal{G}$ of looped graphs. For one of the specific cases we will examine, network flows are used to prove the existence of the decompositions. The idea will be to use a flow to distribute loops. A *network*, N, is a directed graph D with two special vertices s and t, called the *source* and the *sink*, respectively, with a nonnegative real-valued function c on the arcs of D. D is called the *underlying digraph* of N, and c is called the *capacity function* of N. For any arc $a = (x, y) \in A(D)$, the value $c(a)$ is the *capacity* of a.

The source can be thought of as the place from which material is shipped and then transported through N until it reaches its destination - the sink. The capacity of any arc in N can then be thought of as the maximum amount of material which can be transported along that edge. When looking at networks, we usually want to maximize the amount of material transported from the source to the sink, without exceeding the capacity of any arc.

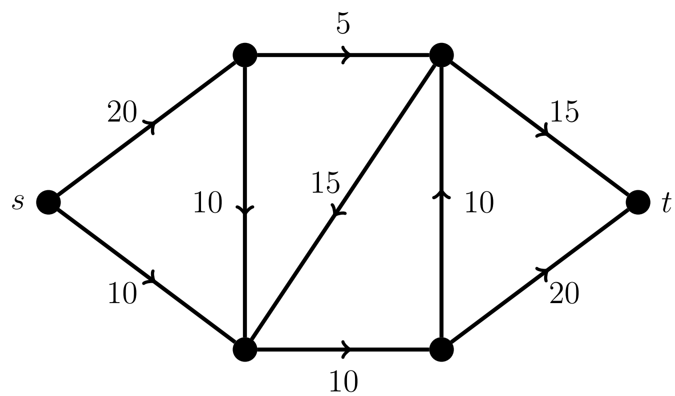

Figure 2.1: A network N with various edge capacities.

For $x \in V(D)$, the *out-neighbourhood*, $N^+(x)$, and the *in-neighbourhood*, $N^-(x)$, of x are defined as $N^+(x) = \{y : (x, y) \in A(D)\}$ and $N^-(x) = \{y : (y, x) \in A(D)\}$. The size of these sets are the *out-degree* and *in-degree* of x, respectively. For a digraph D and a real-valued function g defined on the edges of D, it will be convenient to use the following notation: if X and Y are subsets of $V(D)$, then let

$$[X, Y] = \{(x, y) : x \in X, y \in Y\} \quad \text{and} \quad g(X, Y) = \sum_{(x,y)\in[X,Y]} g(x, y).$$

Here, $g(X, Y) = 0$ if $[X, Y] = \emptyset$. For any vertex $x \in V(D)$, we will write

$$g^+(x) = \sum_{y\in N^+(x)} g(x, y) \quad \text{and} \quad g^-(x) = \sum_{y\in N^-(x)} g(y, x).$$

For any $X \subseteq V(D)$, we will also define

$$g^+(X) = \sum_{x \in X} g^+(x) \quad \text{and} \quad g^-(X) = \sum_{x \in X} g^-(x).$$

A *flow* in a network N with underlying digraph D and capacity function c is a real-valued function f on $A(D)$ satisfying:

i) $0 \leq f(a) \leq c(a)$ for every arc $a \in A(D)$, and

ii) $f^+(x) = f^-(x)$ for every intermediate vertex $x \in V(D)$.

The first of these conditions guarantees that no arc has its capacity exceeded by the flow. The second is referred to as the *conservation equation*, stating that the flow into any vertex must equal the flow out of that vertex. For any arc $a \in A(D)$, the flow along the arc, $f(a)$, can be thought of as the rate at which material is being transported along a under f. For any arc $a \in A(D)$, if $f(a) = c(a)$, then the arc is said to be *saturated* by f. Otherwise, the arc is *unsaturated*. Using the definition of a flow, we obtain the following theorem.

Theorem 2.10. [6] *Let s and t be the source and sink, respectively, of a network N with underlying digraph D, and let f be a flow defined on N. Then the net flow out of s equals the net flow into t. That is, $f^+(s) - f^-(s) = f^-(t) - f^+(t)$.*

The *value* of a flow f in a network N, denoted by $val(f)$, is the net flow out of the source of N (or equivalently, into the sink). A flow in a network N whose value is maximum among all possible flows on N is called a *maximum flow*. That is, f is a maximum flow on N if $val(f) \geq val(f_0)$ for any other flow f_0 on N. Given a network, we usually want to determine its maximum flow value, which will be discussed in Section 2.2.1.

Based on the definition above, a flow f may take the value 0 along certain arcs of the network. However, this may not always be desirable in a flow. For example, in the context of material being shipped, there may be some minimum demand that needs to be met at various points along the way. In this case, we define another nonnegative real-valued function d on the arcs of D, called the *demand function* of N. For any arc $a = (x, y) \in A(D)$, the value $d(a)$ is the *demand* of a, with $0 \leq d(a) \leq c(a)$. If $d(a) > 0$ for some arc a, we need to determine if there is a *feasible* flow on N. That is, a flow f that satisfies $d(a) \leq f(a) \leq c(a)$ for every arc a in our network. To do this, we define a new network, N', called the *transformed network*. To build this transformed network, we take our network N and add a new source s' and sink t'. For every vertex $u \in V(D)$, we add to our network the arcs $s'u$ and ut'. The capacity function c' of our transformed network will take the following values:

(i) $c'(s', u) = \sum_{w \in V(D)} d(w, u)$ and $c'(u, t') = \sum_{w \in V(D)} d(u, w)$ for every $u \in V(D)$;

(ii) $c'(u, w) = c(u, w) - d(u, w)$ for each arc $(u, w) \in A(D)$; and

(iii) $c'(t, s) = \infty$.

Less formally, we construct N' by replacing every arc $(u, w) \in A(D)$ with three arcs: (u, w) with capacity $c(u, w) - d(u, w)$; (s', w) with capacity $d(u, w)$; and (u, t') with capacity $d(u, w)$. We also add the arc (t, s) with unbounded capacity. If this construction produces multiple arcs from s' to the same vertex w (or to t' from the same vertex u), we can merge them into a single arc with the same total capacity. Arcs with zero capacity can be removed from N' altogether. It should be clear that the total capacity out of s' and the total capacity into t' is $T = \sum_{(u,w) \in A(D)} d(u, w)$. A flow of value T in N' would then necessarily saturate all of the arcs containing s' and t'. We can then use the following result to determine if N has a feasible flow.

Theorem 2.11. [1] *A network N contains a feasible flow if and only if its transformed network N' contains a flow that saturates all of the arcs containing s' and t'.*

Example 2.12. Suppose the network given in Figure 2.1 now has a demand on some of its arcs, as shown in Figure 2.2. We denote this in a network by writing $[d(a), c(a)]$ on each arc $a \in A(D)$; this tells us the interval of values a flow is allowed to take along that arc. To determine if there is a feasible flow for this network, we convert it to its transformed network, N'.

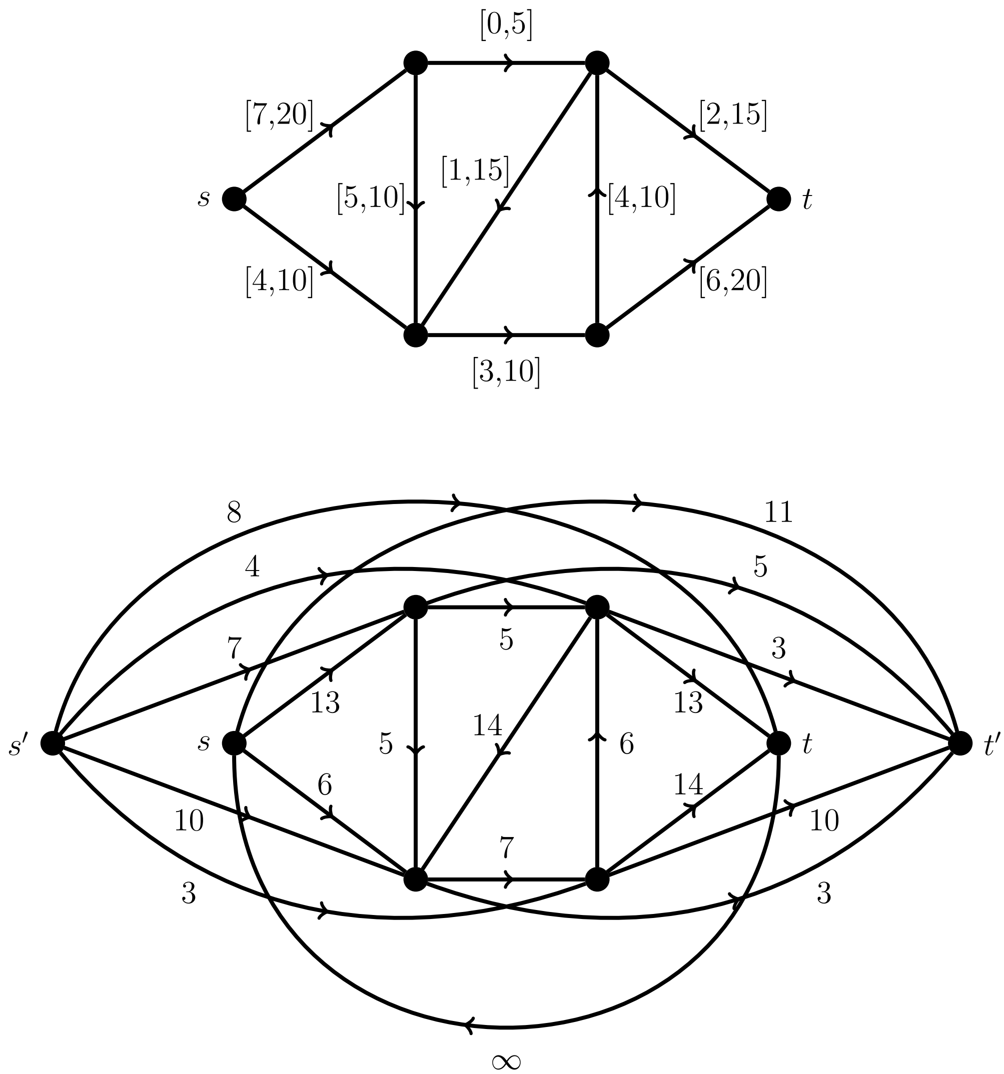

Figure 2.2: A network N with demands and its transformed network N'.

To find a feasible flow for N, Theorem 2.11 tells us that we need only find a flow on N' which saturates the arcs containing s' and those containing t'. In this case, we would need to find a flow on N' with value 32, which is the maximum possible flow value.

2.2.1 The Max-Flow Min-Cut Theorem

For a set $X \subseteq V(D)$, let $\overline{X} = V(D) \setminus X$. A *cut* in N is a set of arcs of the form $[X, \overline{X}]$, where the source s is in X, and the sink t is in $\overline{X}$. If $K = [X, \overline{X}]$ is a cut in N, then the capacity of K is

$$cap(K) = c(X, \overline{X}) = \sum_{(x,y) \in [X, \overline{X}]} c(x, y).$$

It is clear that since $s \in X$ and $t \in \overline{X}$, if all arcs in K were removed from D, then there would be no path from s to t in N. So a cut in D disconnects the source from the sink.

For a set $X \subseteq V(D)$ with $s \in X$ and $t \in \overline{X}$, and a flow f defined on N, the *net flow out of* X is $f^+(X) - f^-(X)$, and the *net flow into* X is $f^-(X) - f^+(X)$. It then follows that $f^+(X) - f^-(X) = f(X, \overline{X}) - f(\overline{X}, X)$. We now show that for a network N and any cut $K = [X, \overline{X}]$ in N, the value of any flow in N is the net flow out of X and that this value never exceeds the capacity of K.

Theorem 2.13. [6] *Let f be a flow in a network N and let $K = [X, \overline{X}]$ be a cut in N. Then $val(f) = f^+(X) - f^-(X) \leq cap(K)$.*

Any cut in N whose capacity is minimum among all cuts in N is called a *minimum cut*. The following two corollaries provide some important information about the relationship between minimum cuts and maximum flows.

Corollary 2.14. [6] *If f is a flow in a network N and K is a cut in N such that $val(f) = cap(K)$, then f is a maximum flow and K is a minimum cut in N.*

Corollary 2.15. [6] *If f is a flow in a network N with capacity function c, and $[X, \overline{X}]$ is a cut in N such that $f(a) = c(a)$ for every arc $a \in [X, \overline{X}]$, and $f(a) = 0$ for every arc $a \in [\overline{X}, X]$, then f is a maximum flow in N and $[X, \overline{X}]$ is a minimum cut.*

Corollary 2.15 suggests how the values of a flow f should be defined on the arcs of a minimum cut in order for f to be a maximum flow. According to Corollary 2.14, if it should ever occur that the value of some flow f in a network N equals the capacity of some cut K in N, then f must be a maximum flow and K is a minimum cut. In 1956, Ford and Fulkerson proved a famous result that is known as the Max-Flow Min-Cut Theorem [14]. Independently, and also in 1956, Elias, Feinstein, and Shannon discovered and proved the very same result [13].

Theorem 2.16 (The Max-Flow Min-Cut Theorem)**.** *In any network, the value of a maximum flow equals the capacity of a minimum cut.*

The proof of Theorem 2.16 provides the basis of an algorithm for finding a maximum flow in a network (known as the Ford-Fulkerson Algorithm). A slight refinement of the Ford-Fulkerson Algorithm, provided by Dinic, was first published in 1970 [11] and published independently by Edmonds and Karp in 1972 (known as the Edmonds-Karp Algorithm) [12].

Chapter 3

Proof of the Main Theorem

Now that we have all the necessary background information, we are ready to prove Theorem 1.5. The proof will be carried out in several steps. First, we will construct infinitely many examples for which $K_v^{[\boldsymbol{\mu};\lambda]}$ can be G-decomposed. We will then use pairwise balanced designs to obtain asymptotically complete residue classes modulo some period n_G. There are only finitely many residue classes satisfying the necessary conditions in (1.7), so our next step will be to obtain an example in each class. To do this, we first show that although the necessary conditions are not sufficient for a G-decomposition in general, they are sufficient for the existence of a "signed" G-decomposition. We will then pad this decomposition by all copies of G to obtain a G-decomposition with large $\lambda' \gg \lambda$. Finally, we will use a construction of Wilson's to take this decomposition and stretch it to a G-decomposition with the desired λ on a larger number of vertices lying in the same residue class. This will give us an example in each admissible residue class, as desired.

3.1 Construction for Large Prime Powers

Before we develop our full theory for G-decompositions of $K_v^{[\boldsymbol{\mu};\lambda]}$, we will use cyclotomy in finite fields to obtain infinitely many examples. This construction is used in

Theorem 3.1 of [21] and works the same way when we introduce colours to our loops. We will give an example demonstrating the construction before stating the general theorem.

Example 3.1. Consider the graph G given in Example 1.4, which has $n = 3$, $m = 2$, $\ell_1 = \ell_2 = 2$ and $\alpha = 2$. Let's choose $v = 13$ and $\lambda = 1$. Notice that these values satisfy the necessary conditions given in (1.7) as $v \equiv 1 \pmod{2m}$ and $\alpha \mid 2m$. From (1.2), we have that $\mu_1 = \mu_2 = 6$. Thus, we want to decompose $K_{13}^{[6,6;1]}$ into copies of G.

Now we have $\mathbb{Z}_{13}^* = \langle 2 \rangle = \{1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7\}$, which has multiplicative cosets of index two: $C_0 = \{1, 4, 3, 12, 9, 10\}$ and $C_1 = \{2, 8, 6, 11, 5, 7\}$. We also see that C_0 can be expressed as $C_0 = \{1, 4, 3, -1, -4, -3\}$, and hence forms a subgroup of $\mathbb{Z}_{13}^*$. Now we can label the vertices of $K_{13}^{[6,6;1]}$ with the integers modulo 13 and place $V(G) = \{u_1, u_2, u_3\}$ onto the vertices of $K_{13}^{[6,6,1]}$ in such a way that the $m = 2$ differences $j - i$ (for $ij \in E(G)$ and $j > i$) lie in different cosets. Suppose we choose $u_1 = 0$, $u_2 = 1$, and $u_3 = 3$. Then the differences formed by the edges in G are $1 - 0 \equiv 1 \pmod{13}$ and $3 - 1 \equiv 2 \pmod{13}$, which lie in C_0 and C_1, respectively.

Consider multiplying $V(G)$ by each of the elements in $\{1, 4, 3\}$, which represents the "positive half" of C_0. This gives us the three sets $\{0, 1, 3\}, \{0, 4, 12\}$, and $\{0, 3, 9\}$. We think of these as our "base blocks" or initial copies of G to be placed on $K_{13}^{[6,6;1]}$. We can then additively develop each of these blocks modulo 13 to obtain a total of 39 copies of G to be placed on $K_{13}^{[6,6;1]}$:

$\{0, 1, 3\}$	$\{0, 4, 12\}$	$\{0, 3, 9\}$
$\{1, 2, 4\}$	$\{1, 5, 0\}$	$\{1, 4, 10\}$
$\{2, 3, 5\}$	$\{2, 6, 1\}$	$\{2, 5, 11\}$
$\{3, 4, 6\}$	$\{3, 7, 2\}$	$\{3, 6, 12\}$
$\{4, 5, 7\}$	$\{4, 8, 3\}$	$\{4, 7, 0\}$

$\{5,6,8\}$	$\{5,9,4\}$	$\{5,8,1\}$
$\{6,7,9\}$	$\{6,10,5\}$	$\{6,9,2\}$
$\{7,8,10\}$	$\{7,11,6\}$	$\{7,10,3\}$
$\{8,9,11\}$	$\{8,12,7\}$	$\{8,11,4\}$
$\{9,10,12\}$	$\{9,0,8\}$	$\{9,12,5\}$
$\{10,11,0\}$	$\{10,1,9\}$	$\{10,0,6\}$
$\{11,12,1\}$	$\{11,2,10\}$	$\{11,1,7\}$
$\{12,0,2\}$	$\{12,3,11\}$	$\{12,2,8\}$

Observe that by doing this, each vertex of $K_{13}^{[6,6;1]}$ appears exactly once as each vertex in G for each base block that we develop. Thus, each vertex of $K_{13}^{[6,6;1]}$ appears exactly three times as each vertex in G, and so it contains a red loop exactly six times, as well as a blue loop exactly six times. It is easy to check that these blocks partition the edge set, and so they give us a G-decomposition of $K_{13}^{[6,6;1]}$.

Proposition 3.2. *Let G be an undirected graph with n vertices, m edges (where m is even), and ℓ_i loops of colour i for $1 \le i \le c$. Then $K_q^{[\boldsymbol{\mu};\lambda]}$ can be decomposed into copies of G when q is a prime power with $q \equiv 1 \pmod{2m}$ and $q > m^{n^2}$.*

Proof. The presence of multiple loop colours has no impact on the construction given in [21], which can be referenced for details. Let G' denote the graph G with loops removed. Following as in [21], a base block in $\mathbb{F}_q$ is constructed by distributing the vertices of G' so that the m edge differences lie in distinct cosets of a subgroup $C_0 \subset \mathbb{F}_q^\times$ of index m. The construction then develops this base block

- multiplicatively by a transversal of $\{\pm 1\}$ in C_0;
- additively in $\mathbb{F}_q$; and
- taking λ copies of the resulting set of blocks.

The result is a set of G'-blocks which decompose K_q^λ. Since the additive group $\mathbb{F}_q^+$ is cyclic, the set of G'-blocks produced by this construction admits a transitive automorphism. It then follows that every vertex u of G' appears in a block as an element of $\mathbb{F}_q$ exactly $\frac{\lambda(q-1)}{2m}$ times. Now consider applying the same construction to G instead. Summing over all the vertices in G, every element of $\mathbb{F}_q$ meets exactly $\frac{\lambda\ell_i(q-1)}{2m} = \mu_i$ loops of colour i in the G-blocks, as there are a total of ℓ_i loops of colour i in G, for $1 \leq i \leq c$. Thus, we obtain a G-decomposition of $K_q^{[\boldsymbol{\mu};\lambda]}$. □

Remark. In Example 3.1, we found a G-decomposition of $K_q^{[\boldsymbol{\mu};\lambda]}$ when $q = 13$, but Proposition 3.2 only guaranteed it for $q > 2^9$.

Now that we have the existence of these decompositions for prime powers, we'd like to extend the result to all sufficiently large values of v which satisfy our necessary conditions.

3.2 PBD Closure

As mentioned before, our next step is to use pairwise balanced designs to cover residue classes that satisfy the necessary conditions laid out in (1.7). This is also discussed in Proposition 3.5 of [21] and works in much the same way.

Proposition 3.3. *Let G be an undirected graph with n vertices, m edges, and ℓ_i loops of colour i for $1 \leq i \leq c$. There exists a positive integer n_G (divisible by m) such that if $K_v^{[\boldsymbol{\mu};\lambda]}$ is G-decomposable for some positive integer v, then $K_{v'}^{[\boldsymbol{\mu}';\lambda]}$ is G-decomposable for all sufficiently large integers $v' \equiv v \pmod{n_G}$.*

Proof. Let $S_G = \{v_j \in \mathbb{Z} : K_{v_j}^{[\boldsymbol{\mu_j};\lambda]}$ is G-decomposable$\}$, where for each $v_j \in S_G$ we have

$$\boldsymbol{\mu_j} = \Big(\frac{\lambda\ell_1(v_j-1)}{2m}, \frac{\lambda\ell_2(v_j-1)}{2m}, \ldots, \frac{\lambda\ell_c(v_j-1)}{2m}\Big).$$

The first thing we will show is that this set is PBD-closed. That is, $v \in S_G$ whenever there exists a PBD(v, S_G). From our discussion of pairwise balanced designs earlier, if a PBD(v, S_G) exists with blocks $\mathcal{B} = \{B_1, B_2, \ldots, B_t\}$, then K_v can be decomposed into subgraphs, each of which is a clique K_{v_j} with vertex set $B_j \in \mathcal{B}$ for some $v_j \in S_G$. Similarly, by taking λ copies of the subgraphs in the decomposition of K_v, we obtain a decomposition of K_v^λ into the subgraphs $K_{v_j}^\lambda$. Now since each $v_j \in S_G$, we know that $K_{v_j}^{[\boldsymbol{\mu_j};\lambda]}$ is G-decomposable, so we can attach $\frac{\lambda \ell_i (v_j - 1)}{2m}$ loops of colour i to each vertex of B_j. Putting these decompositions together, we will obtain a G-decomposition of $K_v^{[\boldsymbol{\mu};\lambda]}$ as long as for each vertex u in K_v^λ we have

$$\sum_{\substack{u \in B_j \\ |B_j| = v_j}} \frac{\lambda \ell_i (v_j - 1)}{2m} = \mu_i$$

for each $1 \leq i \leq c$. That is, the total number of loops of colour i at u, over all copies of G, sums to μ_i. From the way we have constructed this decomposition, we see that the sum of the degrees of the normal edges in the subgraphs $K_{v_j}^\lambda$ which contain u is equal to the degree of u in K_v^λ. That is, we have

$$\sum_{\substack{u \in B_j \\ |B_j| = v_j}} \lambda (v_j - 1) = \lambda (v - 1).$$

Multiplying each side by $\frac{\ell_i}{2m}$, we obtain

$$\sum_{\substack{u \in B_j \\ |B_j| = v_j}} \frac{\lambda \ell_i (v_j - 1)}{2m} = \frac{\lambda \ell_i (v - 1)}{2m},$$

as desired. Thus, we have a G-decomposition of $K_v^{[\boldsymbol{\mu};\lambda]}$. That is, $v \in S_G$. Therefore, S_G is PBD-closed. We can now apply a theorem, due to Wilson, which is stated below as Theorem 3.4.

Now Proposition 3.2 along with Dirichlet's Theorem on the existence of primes in arithmetic progression tell us that S_G is nonempty and contains integers greater than 1. Thus, we know that $\beta(S_G) > 0$. Suppose $K_v^{[\boldsymbol{\mu};\lambda]}$ is G-decomposable. Taking $n_G = \beta(S_G) = \gcd\{v(v-1) : v \in S_G\}$, we see that $K_{v'}^{[\boldsymbol{\mu}';\lambda]}$ is G-decomposable for all sufficiently large integers $v' \equiv v \pmod{n_G}$. □

Theorem 3.4. [27] *Every PBD-closed set K is eventually periodic with period $\beta(K) = \gcd\{k(k-1) : k \in K\}$. That is, there exists a constant C such that, for every $k \in K$, $\{v : v \geq C, v \equiv k \pmod{\beta(K)}\} \subseteq K$.*

3.3 Integral Solutions

In general, the necessary conditions laid out in (1.7) are not sufficient. However, they do guarantee the existence of a G-decomposition where we are allowed to take negative copies of G.

Proposition 3.5. *Let G be an undirected graph with n vertices, m edges, and ℓ_i loops of colour i for $1 \leq i \leq c$. Let $\mathcal{D}_v$ be the set of subraphs of K_v which are isomorphic to G. If $v \geq n+2$, and v satisfies the congruences in (1.7), then there exist integers x_H for each $H \in \mathcal{D}_v$ such that*

$$\sum_{\{H : ij \in E(H)\}} x_H = \lambda \tag{3.1}$$

for every edge $ij \in E(K_v)$ and

$$\sum_{\{H : j \in V(H)\}} e_{j,i} x_H = \mu_i \tag{3.2}$$

for every vertex $j \in V(K_v)$ and every colour $1 \leq i \leq c$.

This proposition can be proved using the following well-known lemma.

Lemma 3.6. [22] *Given an $m \times n$ rational matrix M and some $\mathbf{f} \in \mathbb{Q}^m$, the equation $M\mathbf{x} = \mathbf{f}$ has an integral solution $\mathbf{x}$ if and only if $\mathbf{y}^\top \mathbf{f}$ is integral whenever $\mathbf{y} \in \mathbb{Q}^m$ is such that $\mathbf{y}^\top M$ is integral.*

In the context of Lemma 3.6, we want

$$\mathbf{f}^\top = (\lambda, \lambda, \dots, \lambda, \mu_\mathbf{1}, \mu_\mathbf{1}, \dots, \mu_\mathbf{1}, \mu_\mathbf{2}, \mu_\mathbf{2}, \dots, \mu_\mathbf{2}, \dots, \mu_\mathbf{c}, \mu_\mathbf{c}, \dots, \mu_\mathbf{c}),$$

where we have $\binom{v}{2}$ copies of λ and v copies of μ_i for $1 \leq i \leq c$. Our matrix M is going to tell us, for each graph $H \in \mathcal{D}_v$, which edges of K_v are in H, and how many loops of colour i are at each vertex of H, for $1 \leq i \leq c$. The columns of M are indexed by $\mathcal{D}_v$, and the rows of M are indexed by $E(K_v)$, followed by c copies of $V(K_v)$. From this, we see that Proposition 3.5 claims we can find an integral solution $\mathbf{x}$ to $M\mathbf{x} = \mathbf{f}$, with entries x_H for each $H \in \mathcal{D}_v$. To prove this claim, we can exploit Lemma 3.6 to show that if $\mathbf{y}$ makes $\mathbf{y}^\top M$ integral, then $\mathbf{y}^\top \mathbf{f}$ is also integral. This same technique is used to prove the analagous theorems in [10] and [21] (stated as Lemma 2.3 and Proposition 4.1, respectively). We will highlight the differences that occur in the set up, although they will not affect the proof.

Proof. We want to show that whenever we assign integers β_{ij} to the edges $ij \in E(K_v)$, and integers β_j^i (for $1 \leq i \leq c$) to the vertices $j \in V(K_v)$ such that for each subgraph H the sum

$$\sigma_H = \sum_{ij \in E(H)} \beta_{ij} + \sum_{i=1}^{c} \sum_{j \in V(H)} e_{j,i} \beta_j^i$$

is divisible by some integer d, then the sum

$$\sigma = \lambda \sum_{ij \in E(K_v)} \beta_{ij} + \sum_{i=1}^{c} \mu_i \sum_{j \in V(K_v)} \beta_j^i$$

is also divisible by d. In the context of Lemma 3.6, these β_{ij}'s and β_j^i's are the entries of $\mathbf{y}$. Following as in [21], we see that having multiple loop colours has no effect in showing that this divisibility condition is satisfied. We refer the reader to that paper for details. □

3.4 Wilson's Construction

From Proposition 3.3, we know that if $K_v^{[\boldsymbol{\mu};\lambda]}$ is G-decomposable for some positive integer v, then $K_{v'}^{[\boldsymbol{\mu}';\lambda]}$ is G-decomposable for all sufficiently large integers $v' \equiv v \pmod{n_G}$. Now we want to show that a G-decomposition can be obtained for each admissible residue class $v \pmod{n_G}$.

The solutions found in Section 3.3 are allowed to take negative copies of G. For v satisfying (1.7), we want to obtain a decomposition using only positive copies of G. From a "signed" decomposition on v vertices, guaranteed by Proposition 3.5, we can add all copies of G sufficiently many times to exceed the weight of any negative blocks. This will leave us with a G-decomposition with large $\lambda' \gg \lambda$. We will then use Wilson's construction as in [21] to take this decomposition and stretch it to a G-decomposition with the desired λ on v' vertices, where $v' > v$, and $v' \equiv v \pmod{n_G}$.

Proposition 3.7. *Let G be an undirected graph with n vertices, m edges, and ℓ_i loops of colour i for $1 \leq i \leq c$. For every integer v satisfying (1.7), there exists an integer $v' \equiv v \pmod{n_G}$ such that $K_{v'}^{[\boldsymbol{\mu};\lambda]}$ can be G-decomposed.*

Proof. Let $\{x_H : H \in \mathcal{D}_v\}$ be an integral solution found from Proposition 3.5. For some integer c, let $x'_H = x_H + c$ for every $H \in \mathcal{D}_v$. Then from (3.1), we have

$$\sum_{\{H:ij\in E(H)\}} x'_H = \lambda + c\lambda_0 = \lambda\Big(1 + c\frac{\lambda_0}{\lambda}\Big)$$

where

$$\lambda_0 = \frac{2m|\mathcal{D}_v|}{v(v-1)}$$

is the number of graphs $H \in \mathcal{D}_v$ containing a given edge. This gives us a multiset $\mathcal{H}$ of G-blocks in $\mathcal{D}_v$ such that each edge $ij \in E(K_v)$ appears in exactly $\lambda(1 + c\frac{\lambda_0}{\lambda})$ blocks.

As in [10] and [21], we may choose c such that the following three conditions are all satisfied:

- $x'_H > 0$ for every $H \in \mathcal{D}_v$,
- $\lambda \mid c$, and
- $1 + c\frac{\lambda_0}{\lambda}$ is a prime congruent to 1 modulo n_G, where n_G is the period from Proposition 3.3.

Let $q = 1 + c\frac{\lambda_0}{\lambda}$. Then we have

$$\lambda' = q\lambda = \lambda + c\lambda_0 \equiv \lambda \pmod{\lambda_0}$$

as the multiplicity of every edge obtained from the G-decomposition. That is, $\mathcal{H}$ is a G-decomposition of $K_v^{[\boldsymbol{\mu}';\lambda']}$. Since padding the signed decomposition amounts to adding an equal number of copies of each $H \in \mathcal{D}_v$, the number of loops of colour i is

$$\mu'_i = \frac{q\lambda\ell_i(v-1)}{2m} = \frac{\lambda'\ell_i(v-1)}{2m}$$

for $1 \leq i \leq c$. Given an admissible residue class $v \pmod{n_G}$, we now need to stretch the G-blocks in $\mathcal{H}$ onto a set of v' points, where $v' > v$, and $v' \equiv v \pmod{n_G}$, and recover the desired values of λ and μ_i for $1 \leq i \leq c$. This is done using the same linear algebraic construction as in [10] and [21], which can be referenced for additional details.

First, we choose $t \geq v^2$ large enough that Proposition 3.2 applies. That is, so that $K_{q^t}^{[\boldsymbol{\mu_t};\lambda]}$ can be G-decomposed. Let M denote the complete multipartite graph with v parts, each of size q^t. Observe that M has $v' = vq^t \equiv v \pmod{n_G}$ vertices. Wilson's construction then produces a set of G-blocks which decompose M, and which are invariant under additive shifts in $\mathbb{F}_{q^t}$ on each part. Just as we observed in the proof of Proposition 3.2, the automorphism guarantees that every vertex u of G appears in a block as an element of $\mathbb{F}_{q^t}$ exactly $\frac{\lambda q^t(v-1)}{2m}$ times. Summing over all the vertices in G, every element of $\mathbb{F}_{q^t}$ meets exactly $\frac{\lambda \ell_i q^t(v-1)}{2m}$ loops of colour i in the G-blocks, as there are ℓ_i loops of colour i in G, for $1 \leq i \leq c$.

Now we are able to apply Proposition 3.2 and include blocks to decompose $K_{q^t}^{[\boldsymbol{\mu_t};\lambda]}$ on each part of M. That is, to fill in the holes of M. Here,

$$\boldsymbol{\mu_t} = \Big(\frac{\lambda\ell_1(q^t-1)}{2m}, \frac{\lambda\ell_2(q^t-1)}{2m}, \ldots, \frac{\lambda\ell_c(q^t-1)}{2m}\Big).$$

This results in a G-decomposition of $K_{v'}^{[\boldsymbol{\mu};\lambda]}$. Adding the loop multiplicites together, we have

$$\mu_i = \frac{\lambda\ell_i q^t(v-1)}{2m} + \frac{\lambda\ell_i(q^t-1)}{2m} = \frac{\lambda\ell_i(v'-1)}{2m}$$

for $1 \leq i \leq c$, as desired. □

Chapter 4

Applications in Design Theory

We have seen that appending coloured loops to a simple graph in a specific way can lead to balanced, degree-balanced, and orbit-balanced graph decompositions. Another type of balanced decomposition that we can explore is a distance-balanced decomposition. This type of decomposition is most appropriate to consider for graphs with a central vertex, such as stars, spiders, or even-length paths. Let G be a graph with central vertex x. For every nonnegative integer k and every $u \in V(K_v)$, let $r_k(u)$ denote the number of blocks in a G-decomposition of K_v containing u as a vertex at distance k from x. Recall that the *distance* between vertices u and w, denoted $d(u, w)$, is the length of a shortest path from u to w in G. A G-decomposition of K_v is called *distance-balanced* if for each $k \in \mathbb{N}$, $r_k(u)$ is independent of u. That is, there exists a constant positive integer r_k such that $r_k(u) = r_k$ for all $u \in V(K_v)$.

Example 4.1. Take K_7 and consider decomposing it into copies of the star S_3 shown in Figure 4.1. Here, each copy of S_3 in the decomposition has been given a unique colour. Every vertex of K_7 appears exactly once as a vertex at distance zero from x, and exactly three times as a vertex at distance one from x. Thus, the decomposition is distance-balanced. This decomposition is also orbit-balanced (and hence degree-balanced), but that is certainly not true in general.

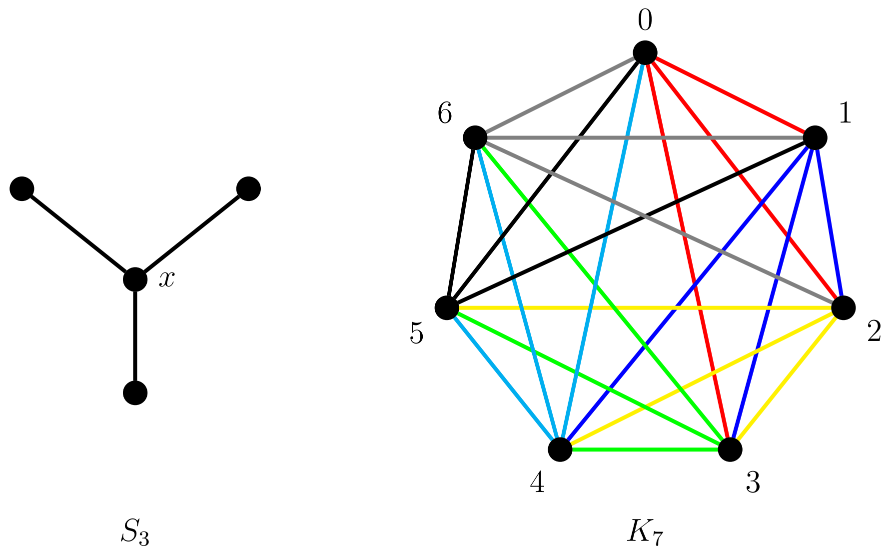

Figure 4.1: S_3 and a distance-balanced S_3-decomposition of K_7.

Similar to our results for degree- and orbit-balanced decompositions, Theorem 1.5 gives us a corollary for distance-balanced decompositions as well.

Corollary 4.2. *Let $\lambda \geq 0$. Suppose G is a simple graph with n vertices, m edges, and central vertex x. Then there exists a distance-balanced G-decomposition of K_v^λ for all sufficiently large v satisfying (1.7) with $e_{u,k} = 1$ if $d(u,x) = k$, and 0 otherwise.*

Example 4.1 was originally given by Bonisoli and Ruini in [4] as a solution to a particular scheduling problem. Their problem involved organizing a seven-student study group. First, each student in the group is assigned a reading on a different subject. For each selected subject, we want to organize a discussion group in which the student who prepared the reading serves as a discussion leader. In order to help students feel easier, each discussion group should be limited to four group members, including the discussion leader. In order to improve acquaintanceship, any two group members should sit once together in a discussion group, with either one as the discussion leader. On the other hand, in order to avoid work-load complaints, each group member should be in the same number of discussion groups. Each discussion group can be modeled by the graph S_3, where the vertex x identifies the discussion leader. Letting the integers modulo 7 denote the group members, Figure 4.1 gives an adequate schedule for the discussion groups. In general, scheduling problems in which participants play some special role may be modeled by some form of balanced graph decomposition.

In each of the corollaries to Theorem 1.5, coloured loops ensured that the local conditions for the graph decompositions were satisfied. Naturally, one might wonder how else coloured loops can be used. In the following sections, we will explore some design theory problems that can be modeled by graphs with coloured loops.

4.1 Ordered Designs

An *ordered design* $\mathrm{OD}_\gamma(t, k, v)$ is a $k \times \gamma \cdot \binom{v}{t} \cdot t!$ array with v entries such that

- each column has k distinct entries, and
- each tuple of t rows contains each column tuple of t distinct entries precisely γ times.

Example 4.3. Here is an $\mathrm{OD}_1(2, 3, 6)$:

1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6	6
2	3	4	5	6	1	3	4	5	6	1	2	4	5	6	1	2	3	5	6	1	2	3	4	6	1	2	3	4	5
3	5	2	6	4	3	1	6	4	5	4	6	5	1	2	2	5	6	3	1	6	4	2	1	3	5	1	4	3	2

Earlier we observed that for given v, k, and λ, the existence of a K_k-decomposition of K_v^λ is equivalent to the existence of a (v, k, λ)-design. We will now look at how ordered designs are related to graph decompositions involving coloured loops. Consider giving each vertex of K_k a single loop with a unique colour, and let G be the resulting graph. We can use ordered designs to obtain G-decompositions of $K_v^{[\boldsymbol{\mu};\lambda]}$. We will first demonstrate the relationship with an example.

Example 4.4. Let G be the graph K_3 with a red loop at vertex v_1, a blue loop at vertex v_2, and a green loop at vertex v_3. In Figure 4.2, we have a decomposition of $K_6^{[5,5,5;6]}$ into copies of G. This was obtained from the $\mathrm{OD}_1(2,3,6)$ given in Example 4.3. The vertices in the copies of G are indexed by the columns of the $\mathrm{OD}_1(2,3,6)$, and the loop colours are indexed by its rows.

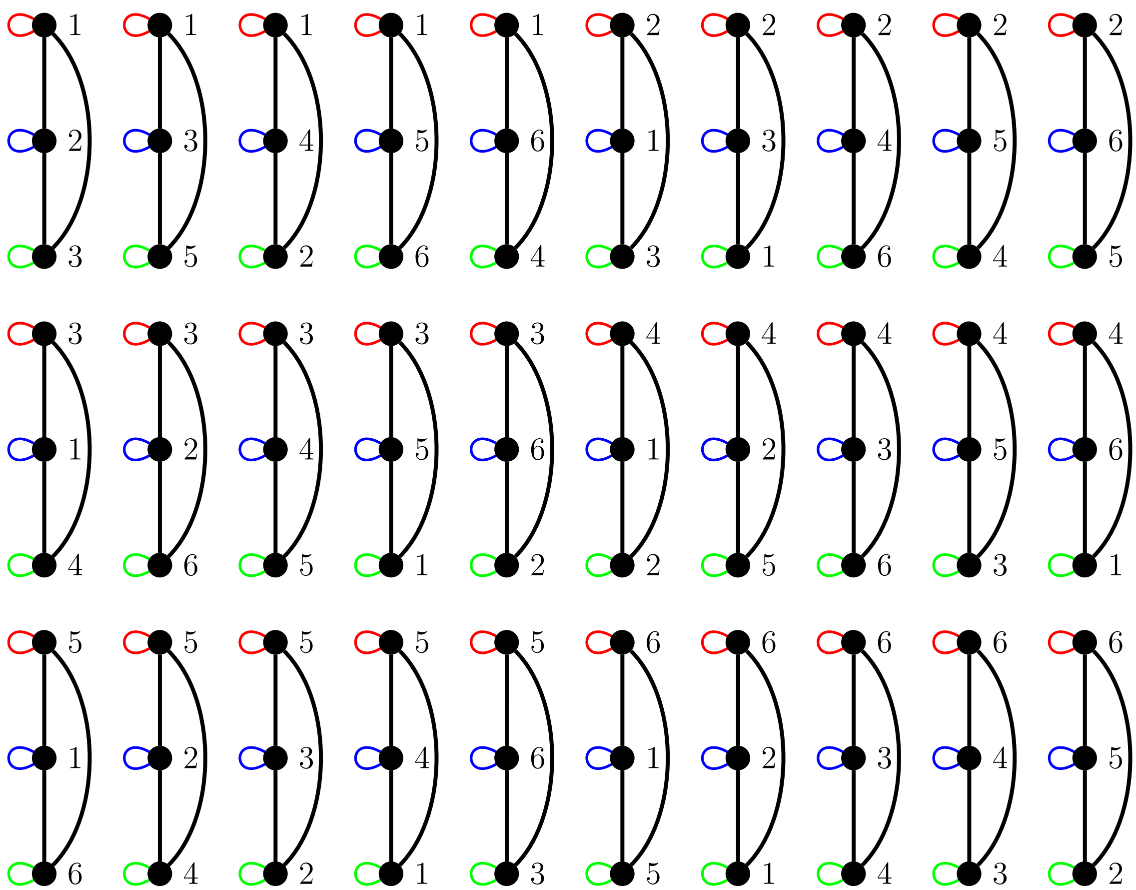

Figure 4.2: A G-decomposition of $K_6^{[5,5,5;6]}$.

Proposition 4.5. *Let $\gamma \in \mathbb{Z}, \gamma \geq 0$. Let G be the complete graph K_k with a loop of colour i at vertex v_i for $1 \leq i \leq k$. If there exists an $OD_\gamma(2, k, v)$, then there exists a G-decomposition of $K_v^{[\mu;\lambda]}$, with $\lambda = \gamma k(k-1)$ and $\mu_i = \gamma(v-1)$ for $1 \leq i \leq k$.*

Proof. The columns of the $\mathrm{OD}_\gamma(2, k, v)$ tell us on which vertices of K_v to place the copies of G. We can designate to each of the k rows a distinct loop colour, as in Example 4.2, to tell us which vertex has a loop of colour i in each copy of G, for $1 \leq i \leq k$. Since this is an $\mathrm{OD}_\gamma(2, k, v)$, each column has k distinct entries, and so a copy of G can certainly be placed on each column. Also, each of the $\binom{k}{2}$ pairs of rows contains each column pair of entries exactly γ times. Thus, every edge of K_v appears exactly $\lambda = 2\gamma\binom{k}{2} = \gamma k(k-1)$ times in the G-decomposition. Some easy counting also tells us that each of the v points appears exactly $\gamma(v-1)$ times in each row of the $\mathrm{OD}_\gamma(2, k, v)$. Hence, each vertex of K_v appears exactly $\gamma(v-1) = \mu_i$ times as a vertex with a loop of colour i for each $1 \leq i \leq k$. □

Proposition 4.5 tells us that ordered designs with $t = 2$ can be modeled by graph decompositions with coloured loops.

4.2 Equitable Block Colourings

Let G be a graph and let $\mathcal{F}$ be a G-decomposition of K_v. An s-*equitable block-colouring* of $\mathcal{F}$ is a colouring $f : \mathcal{F} \to \{1, 2, \ldots, s\}$ of the blocks such that

- the blocks are coloured with exactly s colours, and
- for each vertex u in K_v and for each $\{i, j\} \subset \{1, 2, \ldots, s\}$, we have $|b(f, u, i) - b(f, u, j)| \leq 1$,

where $b(f, u, i)$ is the number of blocks in $\mathcal{F}$ containing u that are coloured i by f. Less formally, it is an assignment of colours to the blocks in $\mathcal{F}$ so that exactly s

colours are used and every vertex appears equally often (or as equally as possible) in blocks of each of the s colours. In [16], M. Gionfriddo and Quattrocchi investigated equitable colourings of 4-cycle systems after seeing much interest in block-colourings where only specified patterns are allowed. Using the asymptotic result in Theorem 1.5, we can also obtain an asymptotic result for certain equitable block-colourings. This idea is illustrated in Example 4.6.

Example 4.6. Consider the K_3-decomposition of K_9 given by the block set of the RBIBD$(9, 3, 1)$ in Example 2.4. We can assign the following colouring f to the blocks:

$$\mathcal{F} = \Big\{\{1,2,3\},\{4,5,6\},\{7,8,9\},\{1,4,7\},\{2,5,8\},\{3,6,9\},$$
$$\{1,5,9\},\{2,6,7\},\{3,4,8\},\{3,5,7\},\{2,4,9\},\{1,6,8\}\Big\}.$$

For each vertex u in K_9, we observe that $b(f, u, black) = 2$, $b(f, u, blue) = 1$, and $b(f, u, purple) = 1$. So f is a 3-equitable block-colouring of $\mathcal{F}$. Let $\mathcal{G} = \{G_1, G_2, G_3\}$, where the G_i's are given in Figure 4.3. Then this 3-equitable block-colouring of $\mathcal{F}$ is equivalent to a $\mathcal{G}$-decomposition of $K_v^{[2,1,1;1]}$, where blocks coloured black, blue, and purple, are replaced by G_1, G_2, and G_3, respectively.

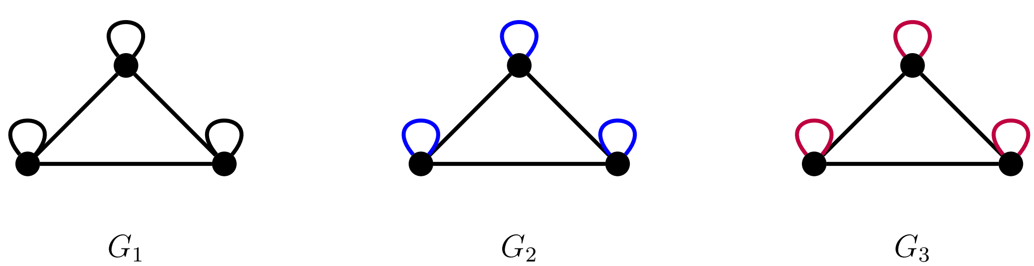

Figure 4.3: The family of graphs $\mathcal{G} = \{G_1, G_2, G_3\}$.

If $G_0 = G_1 \cup G_2 \cup G_3$, then Theorem 1.5 tells us we can obtain a G_0-decomposition of $K_{v'}^{[\boldsymbol{\mu}';1]}$ for all sufficiently large integers v' satisfying the necessary conditions in (1.7). By definition, every vertex of $K_{v'}^{[\boldsymbol{\mu}';1]}$ has exactly μ'_i loops of colour i. Hence, each vertex of $K_{v'}^{[\boldsymbol{\mu}';1]}$ must appear an equal number of times in the copies of G_0 as a vertex with a loop of colour i. Removing the loops, we see that the result gives a G-decomposition of $K_{v'}$ equipped with a 3-equitable block-colouring where every vertex now appears equally often in blocks of each colour.

In [15], L. Gionfriddo, M. Gionfriddo and Ragusa introduced a generalization of these colourings; their work was later extended by Li and Rodger [20]. An (s,p)-*equitable block-colouring* of $\mathcal{F}$ is a colouring $f : \mathcal{F} \to \{1, 2, \ldots, s\}$ of the blocks such that

- the blocks are coloured with exactly s colours,
- for each vertex u in K_v, the blocks containing u are coloured using exactly p colours, and
- for each vertex u in K_v and for each $\{i,j\} \subset C(f,u)$, we have $|b(f,u,i) - b(f,u,j)| \leq 1$,

where $C(f,u) = \{i : f \text{ colours some block containing } u \text{ with colour } i\}$. Less formally, it is an assignment of colours to the blocks in $\mathcal{F}$ so that exactly s colours are used; each vertex is incident with blocks coloured with exactly p colours; and every vertex appears equally often (or as equally as possible) in blocks of each of the p colours. Notice that when $p = s$, we simply have an s-equitable block-colouring.

Just as we observed in Example 4.6, general (s,p)-equitable block-colourings can easily be modelled by a family of graphs containing coloured loops. For a simple graph G and positive integer s, let $\mathcal{G} = \{G_1, G_2, \ldots, G_s\}$, where G_i is the graph G with a loop of colour i at every vertex, for $1 \leq i \leq s$. Suppose $\mathcal{F}$ is a G-decomposition of K_v. Then an (s,p)-equitable block-colouring of $\mathcal{F}$ is equivalent to a $\mathcal{G}$-decomposition of K_v where every graph in $\mathcal{G}$ appears at least once; each vertex in K_v has exactly p

colours appearing amongst its loops; and every vertex in K_v receives an equal number of loops (or as equally many as possible) of each of the p colours.

4.3 Group Divisible Designs

For a positive integer λ, a *group divisible design* (or *GDD*) of index λ is an ordered triple $(V, \mathcal{G}, \mathcal{B})$, where

- V is a finite set of points;
- $\mathcal{G}$ is a partition of V into subsets called *groups*; and
- $\mathcal{B}$ is a collection of subsets of V, called *blocks*, such that a group and a block contain at most one common point, and every pair of points from distinct groups occurs in exactly λ blocks.

The *type* of the GDD is the multiset $\{|G| : G \in \mathcal{G}\}$. This is usually expressed using the following exponential notation: a type $1^i 2^j 3^k \cdots$ denotes i occurrences of 1, j occurrences of 2, and so on. We say that a GDD of index λ is a (K, λ)-GDD if $|B| \in K$ for every block $B \in \mathcal{B}$, where K is a set of positive integers, each of which is at least 2. When $K = \{k\}$, we simply write k instead of K. If $\lambda = 1$, we will replace $(K, 1)$-GDD with K-GDD and $(k, 1)$-GDD with k-GDD.

Example 4.7. Here is one possible 3-GDD of type 2^3:

$$V = \{1,2,3,4,5,6\}, \quad \mathcal{G} = \{\{1,2\},\{3,4\},\{5,6\}\}, \quad \mathcal{B} = \{\{1,3,5\},\{2,4,6\}\}.$$

The definition above can be extended to have two indices: λ_1 and λ_2. In this case, any two points within the same group appear together in exactly λ_1 blocks, and any two points not in the same group appear together in exactly λ_2 blocks. Points occurring together in the same group are called *first associates*, and points occurring in different groups are called *second associates*. Here, we say that a GDD of indices λ_1 and λ_2 is a $(K, \lambda_1, \lambda_2)$-GDD if $|B| \in K$ for every block $B \in \mathcal{B}$, where K is a set of positive integers, each of which is at least 2. Again, when $K = \{k\}$, we simply write k instead of K. When $\lambda_1 = 0$, we have a (K, λ_2)-GDD, as in the original definition.

In Example 4.8, we will demonstrate how coloured loops can be used to encode the information given by a GDD. For a given $(K, \lambda_1, \lambda_2)$-GDD, we will take the blocks and transform them into a family of graphs with coloured edges and coloured loops. Such a graph will tell us exactly which points are covered by its corresponding block.

Example 4.8. Consider the following $(3, 2, 1)$-GDD of type 2^3:

$$V = \{a_1, a_2, b_1, b_2, c_1, c_2\}$$
$$\mathcal{G} = \Big\{A = \{a_1, a_2\}, B = \{b_1, b_2\}, C = \{c_1, c_2\}\Big\}$$
$$\mathcal{B} = \Big\{\{a_1, a_2, b_1\}, \{b_1, b_2, c_1\}, \{c_1, c_2, a_1\}, \{a_1, a_2, b_2\}, \{b_1, b_2, c_2\}, \{c_1, c_2, a_2\}\Big\}$$

This can be thought of as a K_3-decomposition of $3K_2^2 \cup K_{2,2,2}$, where the two graphs have the same vertex set as shown in Figure 4.4.

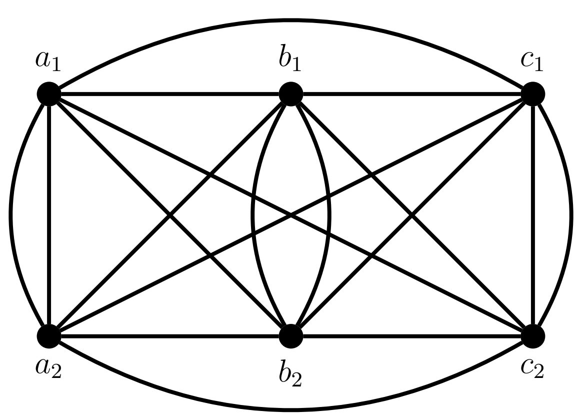

Figure 4.4: The graph $3K_2^2 \cup K_{2,2,2}$.

We will assume that points within a group are ordered, as the position of a point within its group will be important. This order corresponds to the order in which vertices appear in the copies of K_2^2 in Figure 4.4. Now we can encode the information given by this set of blocks into a family of graphs $\mathcal{G}$ with coloured edges and loops. For any block, its corresponding graph will have one vertex for each group of the GDD that it intersects. These vertices will be labelled the same as their respective groups. For a given block, we will build its corresponding graph in the following way:

i) if there is a pair of points which lie in the same group, we will put a loop on the corresponding vertex and give it colour (x, y), where x is the position of the first point and y is the position of the second point; and

ii) if there is a pair of points which lie in different groups, we will put an edge between the two corresponding vertices and give it colour (x, y), where x is the position of the point in the first group and y is the position of the point in the second group.

For example, the block $\{a_1, a_2, b_1\}$ only intersects groups A and B, so its corresponding graph lies only on vertices A and B. The pair of points $\{a_1, a_2\}$ both lie in group A, so we will put a loop on vertex A. In this example, there are only two vertices within each group, so we need only use one colour for the loops. The pair of points $\{a_1, b_1\}$ lie in groups A and B, respectively, so we will draw an edge from A to B and colour it $(1, 1)$. Similarly, for $\{a_2, b_1\}$, we will draw an edge from A to B and colour it $(2, 1)$. Doing this for every block, we obtain the collection of graphs given in Figure 4.5.

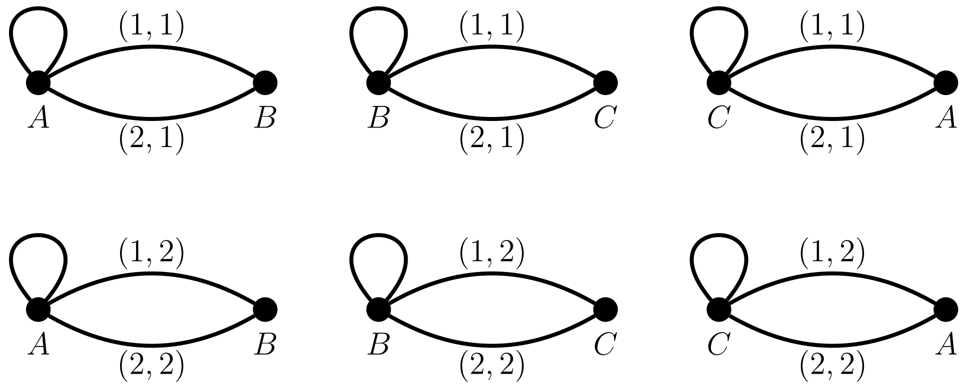

Figure 4.5: Representation of a $(3, 2, 1)$-GDD of type 2^3.

In general, this construction forms a correspondence between group divisible designs and these particular families of graphs. In Example 4.8, we only had one loop colour because the groups were only of size two. For group sizes larger than two, we would observe different loop colours in the representation.

Chapter 5

Extension to $\mathcal{G}$-decompositions

So far we have only considered decompositions into subgraphs isomorphic to a single graph G. We can, however, generalize this further. Recall that if $\mathcal{G}$ is a family of graphs, then a $\mathcal{G}$-decomposition of K_v is a collection $\mathcal{F}$ of subgraphs of K_v such that every edge of K_v occurs in exactly one of the subgraphs, and every graph $F \in \mathcal{F}$ is isomorphic to some graph $G \in \mathcal{G}$. That is, we can find a collection of graphs in $\mathcal{G}$ that partition the edge set of K_v. Since $\mathcal{G}$-decompositions generalize G-decompositions, it seems natural to think that graph decomposition results would be even more useful once a family of graphs is involved.

Example 5.1. Let $\mathcal{G} = \{G_1, G_2\}$ be the family of graphs given in Figure 5.1. These can be used to decompose $K_5^{[2,1;2]}$ by taking the following blocks:

$$\mathcal{B} = \Big\{\{0,1\},\{1,2\},\{2,3\},\{3,4\},\{4,0\},\{0,2\},\{1,3\},\{2,4\},\{3,0\},\{4,1\},$$
$$\{0,1,3\},\{1,2,4\},\{2,3,0\},\{3,4,1\},\{4,0,2\}\Big\}.$$

Here blocks of size two and three correspond copies of G_1 and G_2, respectively, with the sets ordered as the vertices appear in Figure 5.1.

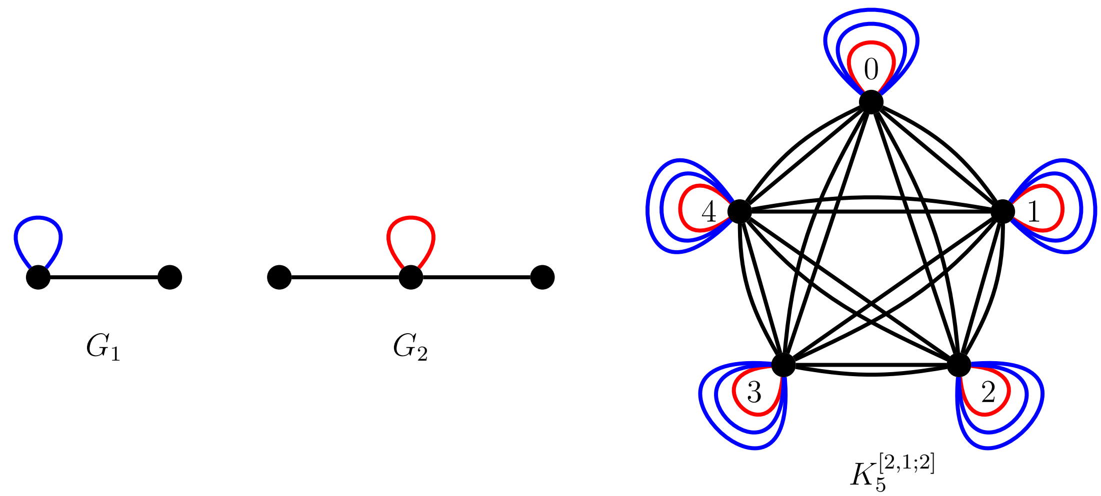

Figure 5.1: The family $\mathcal{G} = \{G_1, G_2\}$ of graphs and $K_5^{[2,1;2]}$.

Let $K_v^{(r)}$ denote the complete digraph on v vertices with exactly one edge of colour i joining any vertex x to any other vertex y for every colour i in a set of r colours. In [19], Lamken and Wilson developed a complete existence theory for $\mathcal{G}$-decompositions of $K_v^{(r)}$, where $\mathcal{G}$ is a family of edge-coloured digraphs satisfying certain necessary conditions. The use of coloured edges gave their result applications to various problems in design theory, such as Mendelsohn triple systems, whist tournaments, Steiner pentagon systems, resolvable designs, and certain group divisible designs. For this reason, it seems natural to consider $\mathcal{G}$-decompositions when $\mathcal{G}$ is a family of graphs with coloured loops and see what applications arise from them.

5.1 Necessary Conditions for $\mathcal{G}$-decompositions

We need to discuss how our necessary conditions will change when we extend to a family of graphs $\mathcal{G}$. For each $G \in \mathcal{G}$, let $m(G)$ be the number of standard edges in G and $\ell_i(G)$ be the number of loops of colour i in G for $1 \leq i \leq c$. Globally, we need to be able to write the total number of edges in K_v^λ as a linear combination of the number of edges in our graphs in $\mathcal{G}$. That is, we need an integral solution $\{s_G\}$ to

$$\sum_{G \in \mathcal{G}} s_G m(G) = \frac{\lambda v(v-1)}{2}. \tag{5.1}$$

Let $m(\mathcal{G}) = \gcd\{m(G) : G \in \mathcal{G}\}$. From [19], we see that (5.1) can be reduced to

$$\lambda v(v-1) \equiv 0 \pmod{2m(\mathcal{G})}. \tag{5.2}$$

Now we need to determine the number of loops of each colour that we require at every vertex of K_v^λ. From the solution above, we see that we need

$$\mu_i v = \sum_{G \in \mathcal{G}} s_G \ell_i(G) \tag{5.3}$$

for $1 \leq i \leq c$. Notice that this implies we are also able to write the total number of loops of colour i in $K_v^{[\boldsymbol{\mu};\lambda]}$ as an integer linear combination of the number of loops of colour i in our graphs in $\mathcal{G}$. Solving for μ_i we obtain

$$\mu_i = \frac{\sum_{G \in \mathcal{G}} s_G \ell_i(G)}{v} = \frac{\lambda(v-1)}{2} \frac{\sum_{G \in \mathcal{G}} s_G \ell_i(G)}{\sum_{G \in \mathcal{G}} s_G m(G)} \tag{5.4}$$

for each $1 \leq i \leq c$. This is much more complicated than in the single graph case, where we simply had $\mu_i = \frac{\lambda \ell_i(v-1)}{2m}$.

For each graph $G \in \mathcal{G}$, let $\hat{\ell}_i(G) = \frac{\ell_i(G)}{m(G)}$ for $1 \leq i \leq c$. This can be thought of as the "density" of loops of colour i which appear in G. From (5.4), we see that each μ_i has a natural upper and lower bound. These are governed, respectively, by the graphs with the greatest and least density of loops of that colour.

For each colour i, let $\hat{\mu}_i = \frac{\mu_i}{\lambda(v-1)}$, which can similarly be thought of as the density of loops of colour i which appear in $K_v^{[\boldsymbol{\mu};\lambda]}$. Since μ_i is a rational multiple of $\lambda(v-1)$, we see that $\hat{\mu}_i$ is independent of v. Now suppose we find a nonnegative rational solution $\{s_G\}$ to (5.1) such that the densities $\hat{\mu}_i$ satisfy

$$\begin{bmatrix} 1 \\ \hat{\mu}_1 \\ \vdots \\ \hat{\mu}_c \end{bmatrix} = \sum_{G \in \mathcal{G}} s_G \begin{bmatrix} m(G) \\ \ell_1(G) \\ \vdots \\ \ell_c(G) \end{bmatrix}. \tag{5.5}$$

We say that a graph G is *useless* in $\mathcal{G}$ if $s_G = 0$ in any such solution. A family of graphs $\mathcal{G}$ is called *admissible* if it contains no useless graphs. That is, there exists a rational linear relation

$$\begin{bmatrix} 1 \\ \hat{\mu}_1 \\ \vdots \\ \hat{\mu}_c \end{bmatrix} = \sum_{G \in \mathcal{G}} s_G \begin{bmatrix} m(G) \\ \ell_1(G) \\ \vdots \\ \ell_c(G) \end{bmatrix} \tag{5.6}$$

with every $s_G > 0$.

Given a family of admissible graphs, let $\beta(\mathcal{G})$ be the least positive integer such that

$$\beta(\mathcal{G})\begin{bmatrix}1\\ \hat{\mu}_1\\ \vdots\\ \hat{\mu}_c\end{bmatrix} \in \mathrm{span}_{\mathbb{Z}}\left\{\begin{bmatrix}m(G)\\ \ell_1(G)\\ \vdots\\ \ell_c(G)\end{bmatrix}\right\}, \tag{5.7}$$

where the span is over all $G \in \mathcal{G}$. Our global necessary condition is then

$$\lambda v(v-1) \equiv 0 \pmod{\beta(\mathcal{G})}. \tag{5.8}$$

Remark. For the global condition, we see that the integral solution $\{s_G\}$ plays two distinct roles. The geometric role involves viewing the $\{s_G\}$ as nonnegative rationals and asking what possible loop densities are achievable, as in (5.6). The arithmetic role is captured by the span definition given in (5.7), together with the congruence in (5.8). However, these conditions are linked via the notion of useless graphs.

Locally, we need to simultaneously be able to decompose the loops and edges incident with each vertex in $K_v^{[\boldsymbol{\mu};\lambda]}$. That is, we need an integral solution $\{t_u\}$ to

$$\sum_{G\in\mathcal{G}} \sum_{u\in V(G)} t_u d_u = \lambda(v-1) \tag{5.9}$$

and

$$\sum_{G\in\mathcal{G}} \sum_{u\in V(G)} t_u e_{u,i} = \mu_i \tag{5.10}$$

for $1 \le i \le c$. Similar to the case where $\mathcal{G} = \{G\}$, (5.9) can be reduced to

$$\lambda(v-1) \equiv 0 \pmod{g(\mathcal{G})} \tag{5.11}$$

where $g(\mathcal{G}) = \gcd\{d_u : u \in V(G), G \in \mathcal{G}\}$. Notice that, for the case of simple graphs without loops, this is the only local condition. Let $\alpha(\mathcal{G})$ be the least positive integer such that

$$\alpha(\mathcal{G}) \begin{bmatrix} 1 \\ \frac{\mu_1}{\lambda(v-1)} \\ \vdots \\ \frac{\mu_c}{\lambda(v-1)} \end{bmatrix} \in \operatorname{span}_{\mathbb{Z}} \left\{ \begin{bmatrix} d_u \\ e_{u,1} \\ \vdots \\ e_{u,c} \end{bmatrix} \right\}, \tag{5.12}$$

where u ranges over all vertices of all graphs in $\mathcal{G}$. Our local necessary condition is then

$$\lambda(v-1) \equiv 0 \pmod{\alpha(\mathcal{G})}. \tag{5.13}$$

Putting (5.8) and (5.13) together, we have our pair of necessary conditions

$$\begin{aligned} \lambda v(v-1) &\equiv 0 \pmod{\beta(\mathcal{G})} \\ \lambda(v-1) &\equiv 0 \pmod{\alpha(\mathcal{G})} \end{aligned} \tag{5.14}$$

for a family $\mathcal{G}$ of admissible graphs, which resemble the familiar conditions for families of graphs without loops [19].

Suppose we have an admissible family of graphs $\mathcal{G}$. The hypothesis that $\mathcal{G}$ is admissible implies that there exists a positive integer t such that

$$\begin{bmatrix} t \\ t\hat{\mu}_1 \\ \vdots \\ t\hat{\mu}_c \end{bmatrix} = \sum_{G\in\mathcal{G}} s_G \begin{bmatrix} m(G) \\ \ell_1(G) \\ \vdots \\ \ell_c(G) \end{bmatrix}$$

with every s_G a positive integer. This means we can form a graph G_0 that is the disjoint union of graphs isomorphic to members of $\mathcal{G}$ and such that G_0 has exactly t edges and $t\hat{\mu}_i$ loops of colour i. By Theorem 1.5, there exists a G_0-decomposition of

$K_v^{[\boldsymbol{\mu};\lambda]}$ for all sufficiently large integers v satisfying

$$\lambda v(v-1) \equiv 0 \pmod{2t}$$
$$\lambda(v-1) \equiv 0 \pmod{\alpha}$$

where α is as in (1.8). Here we have $\mu_i = \frac{\lambda\hat{\mu}_i(v-1)}{2}$ for each $1 \leq i \leq c$. Since G_0 is a disjoint union of graphs in $\mathcal{G}$, this also gives us a $\mathcal{G}$-decomposition of $K_v^{[\boldsymbol{\mu};\lambda]}$. However, since these necessary conditions are different from those in (5.14), we do not obtain the asymptotic sufficiency of those necessary conditions in general.

We will now look at an example which demonstrates some of the complexity behind $\mathcal{G}$-decompositions. This will provide some insight as to why obtaining an asymptotic result for general $\mathcal{G}$-decompositions would be challenging. Suppose a_1, a_2, and a_3 are relatively prime integers. Let $\mathcal{G} = \{G, H_1, H_2, H_3\}$ where $G = K_2$ and H_i is a single vertex with a_i loops. First, observe that G must be used to cover all the edges of K_v^λ for any v and λ. So in any integral solution to (5.1), we must have $s_G = \frac{\lambda v(v-1)}{2}$. Looking at the number of loops μ required at each vertex of K_v^λ, we have

$$v\mu = \sum_{i=1}^{3} s_{H_i} a_i.$$

Let $\hat{\mu} = \frac{\mu}{\lambda(v-1)}$. Now we want to find a nonnegative rational solution $\{s_G\}$ to (5.1) such that $\hat{\mu} = \sum_{i=1}^{3} s_{H_i} a_i$. Since v grows with μ, determining the threshold v_0 for the existence of $\mathcal{G}$-decompositions of $K_v^{[\boldsymbol{\mu};\lambda]}$ for all $v \geq v_0$ is equivalent to finding the Frobenius number of the set $\{a_1, a_2, a_3\}$. Let $\mathcal{A}$ be a set of relatively prime nonnegative integers. Recall that the *Frobenius number* of $\mathcal{A}$ is the largest integer that cannot be expressed as a nonnegative integer combination of elements in $\mathcal{A}$. When $\mathcal{A}$ contains only two integers, there is an explicit formula for the Frobenius number. If the number of integers is three or more, no explicit formula is known. However, for any fixed number of integers, there is an algorithm computing the Frobenius number in polynomial time [18]. No known algorithm is polynomial time in the number of integers, and the general problem, where the number of integers may be as large as desired, is NP-hard [2].

Since the general problem of $\mathcal{G}$-decompositions is quite difficult, we will spend the remainder of this chapter looking at the (already nontrivial) special case when the underlying graph is a clique. In Chapter 6, we will discuss some other techniques that may be used to tackle this problem. We hope that this will provide a good foundation for future work on $\mathcal{G}$-decompositions when the graphs contain loops.

5.2 Decomposing into a Family of Cliques

For the problem we are about to analyze, we will just consider the case where there is only one loop colour. Suppose we have a collection of cliques $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$, where G_i is a copy of K_k with exactly i looped vertices. We would like to know if we are able to decompose the complete graph K_v into copies of graphs from $\mathcal{G}$ so that K_v has an equal number of loops at every vertex. We will see that the number of loops must be a fixed multiple of $v-1$. For this problem, we will use a network flow.

Suppose $(V, \mathcal{B})$ is a BIBD$(v, k, 1)$. Recall that a BIBD$(v, k, 1)$ is equivalent to a K_k-decomposition of K_v. This is the relationship we will use to analyze our problem. To see how we could distribute loops to the copies of K_k in such a decomposition, we will set up a network, $N(v, k)$. Let s and t denote the source and sink, respectively. The underlying digraph $D(v, k)$ will be constructed as follows:

- $V\Big(D(v,k)\Big) = V \cup \mathcal{B} \cup \{s, t\}$, and
- $A\Big(D(v,k)\Big) = \Big\{(u, B) : u \in B, B \in \mathcal{B}\Big\} \cup \Big\{(s, u) : u \in V\Big\} \cup \Big\{(B, t) : B \in \mathcal{B}\Big\}$.

The flow which will run through the network can be thought of as a designation of loops. Suppose we want to have μ loops at every vertex of K_v. The capacity function c defined on the arcs of $D(v,k)$ will take the following values:

- $c(u,B) = 1$ for all $(u,B) \in [V,\mathcal{B}]$, as each vertex of K_v can have at most one loop in any given block;
- $c(B,t) = k$ for all $B \in \mathcal{B}$, as each block can have at most k loops; and
- $c(s,u) = \mu$ for all $u \in V$, where μ is the desired number of loops at each vertex of K_v.

As for its demand function d, we will have $d(B,t) = 1$ for all $B \in \mathcal{B}$, as we need at least one loop on every block. Every other arc in $D(v,k)$ will have a demand of zero.

Example 5.2. To examine decomposing K_9 into copies of K_3, we can consider the RBIBD$(9,3,1)$ with block set

$$\mathcal{B} = \Big\{\{1,2,3\},\{4,5,6\},\{7,8,9\},\{1,4,7\},\{2,5,8\},\{3,6,9\},$$
$$\{1,5,9\},\{2,6,7\},\{3,4,8\},\{1,6,8\},\{2,4,9\},\{3,5,7\}\Big\}.$$

Suppose each K_3 can have either one, two, or three looped vertices, and we want to have $\mu = 2$ loops at each vertex in K_9. Figure 5.2 illustrates the network $N(9,3)$ for this decomposition, with one possible flow f. Here f takes the values zero, one, and two along the black, red, and blue arcs, respectively. Notice that $f(s,u) = 2$ for every $u \in V$, so every vertex of K_9 receives two loops, as desired. For each $B \in \mathcal{B}$, the value of $f(B,t)$ will tell us how many loops that copy of K_3 has, and the value of $f(u,B)$ will tell us if vertex u has a loop in that copy of K_3.

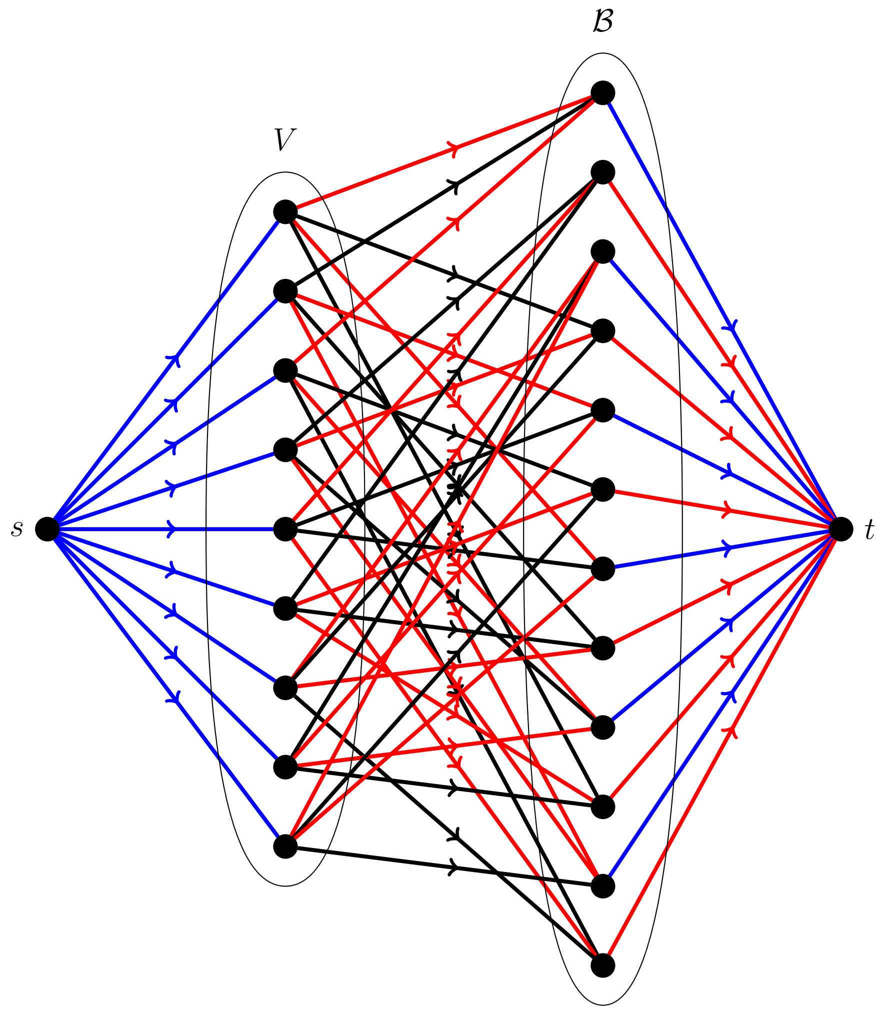

Figure 5.2: $N(9,3)$ with a maximum flow.

In general, we would like to show that the value of a maximum flow on $N(v,k)$ is μv (and hence every vertex in K_v will receive exactly μ loops). From Theorem 2.16, we know that this is equivalent to showing that the capacity of a minimum cut in $N(v,k)$ is μv. Notice that any flow on $N(v,k)$ is restricted by the demands and capacities put on the arcs from $\mathcal{B}$ to t. For every $B \in \mathcal{B}$, we have $1 = d(B,t) < c(B,t) = k$. There are $\frac{v(v-1)}{k(k-1)}$ blocks in a $(v,k,1)$-design, so for any flow f, we must have $\frac{v(v-1)}{k(k-1)} \leq val(f) \leq \frac{v(v-1)}{k-1}$. Since we want μ loops at each of the v vertices in K_v, we must have $\frac{v-1}{k(k-1)} \leq \mu \leq \frac{v-1}{k-1}$.

Theorem 5.3. *Let v and k be positive integers such that a $(v,k,1)$-design exists, and let $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$, where G_i is a copy of K_k with exactly i looped vertices. Then $K_v^{[\mu;1]}$ can be $\mathcal{G}$-decomposed whenever $\frac{v-1}{k(k-1)} \leq \mu \leq \frac{v-1}{k-1}$ and $\mu \in \mathbb{Z}^+$.*

To prove this theorem, we will show that if f is a maximum flow on $N(v,k)$, then $val(f) = \mu v$ for $\frac{v-1}{k(k-1)} \leq \mu \leq \frac{v-1}{k-1}$, $\mu \in \mathbb{Z}^+$. As mentioned before, $d(B,t) = 1$ for all $B \in \mathcal{B}$. Thus, we must first determine if $N(v,k)$ has a feasible flow. That is, a flow which meets the demand along every arc of $N(v,k)$. This will be accomplished in Section 5.2.1. Once we have found a feasible flow, we will remove it from $N(v,k)$. In Section 5.2.2, we then show that the resulting network has the desired remaining maximum flow.

5.2.1 Finding a Feasible Flow

From Theorem 2.11, we know that our network $N(v,k)$ contains a feasible flow if and only if its transformed network $N'(v,k)$ contains a flow which saturates all of the arcs containing s' and t'. From the definition of the transformed network given in Section 2.2, the underlying digraph $D'(v,k)$ will be constructed as follows:

- $V\Big(D'(v,k)\Big) = V\Big(D(v,k)\Big) \cup \{s',t'\}$, and
- $A\Big(D'(v,k)\Big) = A\Big(D(v,k)\Big) \cup \Big\{(s',u) : u \in V\Big(D(v,k)\Big)\Big\} \cup \Big\{(u,t') : u \in V\Big(D(v,k)\Big)\Big\}$.

After some simplification, the capacity function c' on the arcs of $D'(v,k)$ will take the following values:

- $c'(s,u) = c(s,u)$ for all $u \in V$;
- $c'(u,B) = c(u,B)$ for all $(u,B) \in [V,\mathcal{B}]$;
- $c'(s't) = \frac{v(v-1)}{k(k-1)}$;

- $c'(Bt') = 1$ and $c'(Bt) = k - 1$ for all $B \in \mathcal{B}$; and

- $c'(ts) = \infty$.

For any flow f on $N'(v,k)$, we see that $val(f) \leq \frac{v(v-1)}{k(k-1)}$, which is the total possible flow out of s' and the total possible flow into t'. If we can show that $N'(v,k)$ can achieve this maximum flow value, then Theorem 2.11 tells us that $N(v,k)$ has a feasible flow.

Before we proceed, we will simplify our network. First we will send $\frac{v(v-1)}{k(k-1)}$ units of flow from s' to t, as $c'(s't) = \frac{v(v-1)}{k(k-1)}$ and $c'(s'u) = 0$ for all other $u \in V\Big(D'(v,k)\Big)$. Then we will send $\frac{v(v-1)}{k(k-1)}$ units of flow from t to s, as $c'(ts) = \infty$ and $c'(tu) = 0$ for all other $u \in V\Big(D'(v,k)\Big)$. Now we can ignore s', and treat s as our source, with $\frac{v(v-1)}{k(k-1)}$ units of flow which need to leave s. Notice that since $\mu v \geq \frac{v(v-1)}{k(k-1)}$, we will never exceed the total capacity of the arcs leaving s.

To make things easier later on, we will look at achieving a flow value of $\frac{v(v-1)}{k(k-1)}$ when $c'(s,u) = \frac{v-1}{k(k-1)}$ instead of capacity $c'(s,u) = \mu$ for each $u \in V$. Notice that this does not change our maximum possible flow value. To show that we can achieve this flow value we will find a minimum cut K in $N'(v,k)$ and show that $cap(K) \geq \frac{v(v-1)}{k(k-1)}$.

Proposition 5.4. *Let $N(v,k)$ be the network defined in Section 5.2, and let $N'(v,k)$ be its transformed network. If f is a maximum flow on $N'(v,k)$, then $val(f) = \frac{v(v-1)}{k(k-1)}$.*

Proof. Let $X \subseteq V$ and $Y \subseteq \mathcal{B}$, with $|X| = x$ and $|Y| = y$, and let $C = X \cup Y \cup \{s\}$. We will be minimizing the value of the cut $K = [C, \overline{C}]$. For $0 \leq i \leq k$, let $\mathcal{B}_i$ be the set of blocks in $\mathcal{B}$ which intersect X in exactly i points, and let $b_i = |\mathcal{B}_i|$.

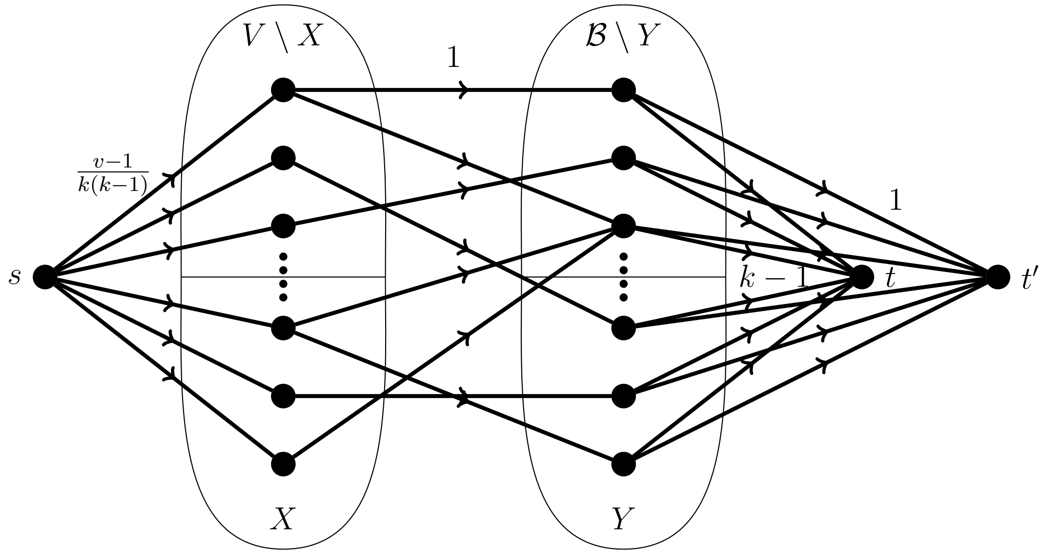

Figure 5.3: $N'(v,k)$ with sets $X \subseteq V$ and $Y \subseteq \mathcal{B}$ for our cut.

Some easy counting gives us the following system of equations:

$$\binom{v}{2} = \binom{k}{2} \sum_{i=0}^{k} b_i. \tag{5.15}$$

$$\binom{x}{2} = \sum_{i=2}^{k} \binom{i}{2} b_i \tag{5.16}$$

$$\binom{v-x}{2} = \sum_{i=0}^{k-2} \binom{k-i}{2} b_i \tag{5.17}$$

$$x(v-x) = \sum_{i=1}^{k-1} i(k-i) b_i \tag{5.18}$$

These equations count the number of pairs of points in V, X, $V \setminus X$, and between X and $V \setminus X$, respectively. In each equation, the left-hand side counts the pairs directly, while the right-hand side counts them using the block set $\mathcal{B}$. These equations will be instrumental in determining the capacity of a minimum cut in $N'(v,k)$. They will also be used once we return to the original network in Section 5.2.2.

Now the arcs contained in our cut will change depending on whether or not t is included in C, so we will consider the two cases separately. First, suppose $t \in C$. The capacity of our cut K is

$$cap(K) = \sum_{u \in V \setminus X} c'(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c'(u,B) + \sum_{B \in Y} c'(B,t').$$

For every $u \in V \setminus X$ (of which there are $v - x$) we have $c'(s,u) = \frac{v-1}{k(k-1)}$. Let B_i be a block intersecting X in i points. If $B_i \in Y$, then we add the arc (B_i, t') to our cut with $c'(B_i, t') = 1$. If $B_i \notin Y$, then for every $u \in X$ that intersects B_i (of which there are i), we add the arc (u, B_i) to our cut with $c'(u, B_i) = 1$. So to minimize $cap(K)$, we should have $Y = \bigcup_{i=1}^{k} \mathcal{B}_i$. Our cut then has capacity

$$\begin{aligned} cap(K) &= \sum_{u \in V \setminus X} c'(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c'(u,B) + \sum_{B \in Y} c'(B,t') \\ &= \frac{(v-x)(v-1)}{k(k-1)} + \sum_{i=1}^{k} b_i \\ &= \frac{(v-x)(v-1)}{k(k-1)} + \frac{v(v-1)}{k(k-1)} - b_0 \\ &\geq \frac{(v-x)(v-1)}{k(k-1)} + \frac{v(v-1)}{k(k-1)} - \frac{(v-x)(v-x-1)}{k(k-1)} \\ &= \frac{v(v-1) + vx - x^2}{k(k-1)} \end{aligned}$$

by using equations (5.15) and (5.17).

Since this is quadratic in x with a negative leading coefficient, the minimum is achieved at one of the endpoints. When $x = 0$ and when $x = v$, we obtain a lower bound on $cap(K)$ of $\frac{v(v-1)}{k(k-1)}$. Hence, the capacity of a minimum cut in $N(v,k)$ is bounded below by $\frac{v(v-1)}{k(k-1)}$. If f is a maximum flow on $N'(v,k)$, Theorem 2.16 tells us that $val(f) = cap(K) \geq \frac{v(v-1)}{k(k-1)}$. However, $val(f) \leq \frac{v(v-1)}{k(k-1)}$, and so we get our desired maximum flow.

Suppose now that $t \notin C$. The capacity of our cut K is

$$cap(K) = \sum_{u \in V \setminus X} c'(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c'(u,B) + \sum_{B \in Y} c'(B,t') + \sum_{B \in Y} c'(B,t).$$

For every $u \in V \setminus X$ (of which there are $v - x$) we have $c'(s,u) = \frac{v-1}{k(k-1)}$. Let B_i be a block intersecting X in i points. If $B_i \in Y$, then we add the arcs (B_i, t') and (B_i, t) to our cut with $c'(B_i, t') = 1$ and $c'(B_i, t) = k - 1$. If $B_i \notin Y$, then for every $u \in X$ that intersects B_i (of which there are i), we add the arc (u, B_i) to our cut with $c'(u, B_i) = 1$. So to minimize $cap(K)$, we should have $Y = \mathcal{B}_k$. Our cut then has capacity

$$\begin{aligned}
cap(K) &= \sum_{u \in V \setminus X} c'(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c'(u,B) + \sum_{B \in Y} c'(B,t') + \sum_{B \in Y} c'(B,t) \\
&= \frac{(v-x)(v-1)}{k(k-1)} + \sum_{i=1}^{k-1} i b_i + k b_k \\
&= \frac{(v-x)(v-1)}{k(k-1)} + \frac{1}{k-1}\left(x(v-x) - \sum_{i=2}^{k-1} i(k-i) b_i\right) + \sum_{i=2}^{k-1} i b_i + k b_k \\
&= \frac{(v-x)(v-1)}{k(k-1)} + \frac{x(v-x)}{k-1} + \frac{2}{k-1} \sum_{i=2}^{k-1} \binom{i}{2} b_i + k b_k \\
&= \frac{(v-x)(v-1)}{k(k-1)} + \frac{x(v-x)}{k-1} + \frac{2}{k-1}\left(\binom{x}{2} - \binom{k}{2} b_k\right) + k b_k \\
&= \frac{(v-x)(v-1)}{k(k-1)} + \frac{x(v-x)}{k-1} + \frac{x(x-1)}{k-1} \\
&= \frac{v(v-1) + (k-1)(v-1)x}{k(k-1)}
\end{aligned}$$

by using equations (5.18) and (5.16).

Since this is linear in x with a nonnegative leading coefficient, the minimum is achieved when $x = 0$. With $x = 0$, we obtain $cap(K) = \frac{v(v-1)}{k(k-1)}$, and so the capacity of a minimum cut in $N(v,k)$ is $\frac{v(v-1)}{k(k-1)}$. If f is a maximum flow on $N'(v,k)$, Theorem 2.16 tells us that $val(f) = cap(K) = \frac{v(v-1)}{k(k-1)}$, as desired. □

5.2.2 Finding a Maximum Flow

Returning to our original network $N(v,k)$, we want to make sure that the demands on every arc are met before we find our maximum flow. Hence, we want to remove from $N(v,k)$ a saturating flow found for $N'(v,k)$. Let f_0 be a saturating flow for $N'(v,k)$ found by Proposition 5.4. Define $N_0(v,k)$ to be the network with underlying digraph $D(v,k)$ and capacity function $c_0(a) = c(a) - f_0(a)$ for every arc a in $D(v,k)$. Notice that the demand on every arc is now zero.

Proposition 5.5. *Let $N_0(v,k)$ be defined as above. If f is a maximum flow on $N_0(v,k)$, then $val(f) = \mu v - \frac{v(v-1)}{k(k-1)}$.*

Proof. Looking at the capacities in $N_0(v,k)$, we have

- $c_0(s,u) = \mu - \frac{v-1}{k(k-1)}$ for every $u \in V$;
- $c_0(B,t) = k-1$ for every $B \in \mathcal{B}$; and
- $c_0(u,B) = 1 - f_0(u,B)$ for every $(u,B) \in [V,\mathcal{B}]$.

Let $X \subseteq V$ and $Y \subseteq \mathcal{B}$, with $|X| = x$ and $|Y| = y$, and let $C = X \cup Y \cup \{s\}$. We will be minimizing the value of the cut $K = [C, \overline{C}]$. For $0 \le i \le k$, let $\mathcal{B}_i$ be the set of blocks in $\mathcal{B}$ which intersect X in exactly i points, and let $b_i = |\mathcal{B}_i|$. The capacity of our cut K is

$$cap(K) = \sum_{u \in V \setminus X} c_0(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c_0(u,B) + \sum_{B \in Y} c_0(B,t).$$

For every $u \in V \setminus X$ (of which there are $v - x$) we have $c_0(s,u) = \mu - \frac{(v-1)}{k(k-1)}$. Let B_i be a block intersecting X in i points. If $B_i \in Y$, then we add the arc (B_i, t) to our cut with $c_0(B_i,t) = k-1$. If $B_i \notin Y$, then for every $u \in X$ that intersects B_i (of which there are i), we add the arc (u, B_i) to our cut with $c_0(u,B_i) = 1$ or $c_0(u,B_i) = 0$.

Notice that there can only be at most one $u \in X$ for which $c_0(u, B_i) = 0$. So to minimize $cap(K)$, we should have $Y = \mathcal{B}_k$.

To lower bound $cap(K)$, we observe that for each block $B \in \mathcal{B}$ there is exactly one vertex $u \in V$ such that $c_0(u, B) = 0$. In the most extreme case, every vertex in X is contained in such an arc. However, since arcs between X and Y are not included in our cut, and Y only contains blocks which intersect X in k points, the maximum capacity that could be removed from our cut by the arcs between X and $\mathcal{B}$ is $\frac{x(v-1)}{k(k-1)} - b_k$. Thus, we have

$$\begin{aligned} cap(K) &= \sum_{u \in V \setminus X} c_0(s,u) + \sum_{(u,B) \in [X, \mathcal{B} \setminus Y]} c_0(u,B) + \sum_{B \in Y} c_0(B,t) \\ &\geq \left(\mu - \frac{v-1}{k(k-1)}\right)(v-x) + \sum_{i=1}^{k-1} i b_i - \left(\frac{x(v-1)}{k(k-1)} - b_k\right) + (k-1)b_k \\ &= \left(\mu - \frac{v-1}{k(k-1)}\right)(v-x) - \frac{x(v-1)}{k(k-1)} + \sum_{i=1}^{k-1} i b_i + k b_k \\ &= \left(\mu - \frac{v-1}{k(k-1)}\right)(v-x) - \frac{x(v-1)}{k(k-1)} + \frac{x(v-x)}{k-1} + \frac{x(x-1)}{k-1} \\ &= \frac{\mu v k(k-1) - v(v-1) + k\Big(v-1-\mu(k-1)\Big)x}{k(k-1)} \end{aligned}$$

by using equations (5.18) and (5.16) as in the proof of Proposition 5.4.

By assumption, we have $\mu \leq \frac{v-1}{k-1}$. Looking at the coefficient of x, we see that

$$k\Big(v-1-\mu(k-1)\Big) \geq k\Big(v-1-\frac{v-1}{k-1}(k-1)\Big) = 0.$$

Since this is linear in x with a nonnegative leading coefficient, the minimum is achieved when $x = 0$. With $x = 0$, we obtain a lower bound on $cap(K)$ of $\mu v - \frac{v(v-1)}{k(k-1)}$. Hence, the capacity of a minimum cut in $N_0(v,k)$ is bounded below by $\mu v - \frac{v(v-1)}{k(k-1)}$. If f is a maximum flow on $N_0(v,k)$, Theorem 2.16 tells us that $val(f) = cap(K) \geq \mu v - \frac{v(v-1)}{k(k-1)}$. However, $val(f) \leq \mu v - \frac{v(v-1)}{k(k-1)}$, and so we get our desired maximum flow. $\square$

One thing to note about the proof of Theorem 5.3 is the importance of having consecutive loop values available in the family $\mathcal{G}$. The fact that each copy of K_k can have anywhere between 1 and k looped vertices is crucial for determining the capacity of a minimum cut. In particular, it allows us to easily determine which blocks should be included in a minimum cut. The problem becomes much more complicated if we allow "gaps" between the number of looped vertices. However, we do believe this method has further potential for other families of graphs or other block designs.

5.2.3 Asymptotic Result

Now that we've proven Theorem 5.3, we can use θ-resolvable designs to give us an asymptotic result for this problem. Recall the necessary conditions for the existence of a θ-RBIBD(v, k, λ) given in (2.3). It is not hard to see that, for a given v, the integer $\frac{k}{\gcd(k,v)}$ is the smallest value of θ satisfying (2.3) when $\lambda = 1$. In the case of $\lambda = 1$, if we let $\theta = \frac{k}{\gcd(k,v)}$, we observe that if v and k satisfy (2.1), then they also satisfy (2.3). That is, if the necessary conditions for the existence of a BIBD$(v, k, 1)$ are satisfied, then so are the necessary conditions for the existence of a θ-RBIBD$(v, k, 1)$ with $\theta = \frac{k}{\gcd(k,v)}$.

Let v_0 be such that (2.3) is sufficient for the existence of θ-resolvable designs of order $v \geq v_0$ for every θ which is a divisor of k. There are only finitely many divisors of k, and so v_0 is the maximum of a finite number of integers. From Theorem 2.7, we then obtain $(v, k, 1)$-designs with the property that, for any possible multiple t of $\frac{k}{\gcd(k,v)}$, there exists a subcollection of the blocks which is t-regular. That is, we obtain K_k-decompositions of K_v such that for any possible multiple t of $\frac{k}{\gcd(k,v)}$, there exists a subcollection of the blocks in which every vertex of K_v appears exactly t times. For what follows, let r_θ be the number of θ-parallel classes in a θ-resolvable $(v, k, 1)$-design with $\theta = \frac{k}{\gcd(k,v)}$, and let a and b be positive integers summing to r_θ.

Theorem 5.6. *Let k and k' be positive integers such that a BIBD$(k, k', 1)$ exists, and let $\mathcal{G}$ be defined as in Theorem 5.3. Let $\frac{k-1}{k'(k'-1)} \leq l < h \leq \frac{k-1}{k'-1}$ be integers. Then the complete graph $K_v^{[\mu;1]}$ can be $\mathcal{G}$-decomposed for all $v \geq v_0$ with $\mu = \theta(al + bh)$.*

Proof. As mentioned above, Theorem 2.7 guarantees us K_k-decompositions of K_v such that for any possible multiple t of θ, there exists a subcollection of the blocks in which every vertex of K_v appears exactly t times. For a given θ-parallel class in a θ-RBIBD$(v, k, 1)$, we choose to append either l loops or h loops to every vertex in every block of that θ-parallel class. Every block in the decomposition is then either a $K_k^{[l;1]}$ or $K_k^{[h;1]}$. Letting a and b be the number of θ-parallel classes whose graphs will be copies of $K_k^{[l;1]}$ and $K_k^{[h;1]}$, respectively, we have $r_\theta = a + b$. Since each vertex of K_v appears exactly θ times in each parallel class, each vertex must receive $\mu = \theta(al + bh)$ loops in total. Finally, since $\frac{k-1}{k'(k'-1)} \leq l < h \leq \frac{k-1}{k'-1}$, Theorem 5.3 guarantees that every copy of $K_k^{[l;1]}$ and $K_k^{[h;1]}$ is $\mathcal{G}$-decomposable. $\square$

Chapter 6

Future Work

Theorem 5.3 gave us a result for $\mathcal{G}$-decompositions when $\mathcal{G}$ was a family of looped cliques, all of the same size. We employed several tools to prove Theorem 5.3. A BIBD gave us the initial decomposition and then a network flow was used to distribute the loops. A similar method might be used to obtain more results by allowing other block designs to form the initial decomposition. Network flows may also prove useful if we let vertices have multiple loops within the family of graphs or multiple loop colours.

Of course, there is still much more work to be done on the problem of general $\mathcal{G}$-decompositions, particularly when the underlying graphs are nonisomorphic. Another tool that we used in the proof of Theorem 5.6 was θ-resolvable designs. In the following example, we will see how cyclic designs and resolvable designs could be used to obtain $\mathcal{G}$-decompositions when the underlying graphs are nonisomorphic.

Example 6.1. Let $\mathcal{G} = \{G_1, G_2\}$, where G_1 and G_2 are the two graphs shown in Figure 5.1, except with loops of the same colour. Notice that the underlying graphs of G_1 and G_2 are nonisomorphic. Let $V = \mathbb{Z}_5$ be the vertices of K_5. Consider developing the two blocks $\{0, 1\}$ and $\{0, 2\}$ modulo 5 to obtain a cyclic BIBD$(5, 2, 1)$. That is, the BIBD$(5, 2, 1)$ with block set

$$\mathcal{B} = \{x + \{0,1\} : x \in V\} \cup \{x + \{0,2\} : x \in V\},$$

where addition is distributed into the set and is modulo 5. Observe that we can obtain a G_1-decomposition of $K_5^{[2;1]}$ by using the blocks of this BIBD$(5,2,1)$. We can also develop the block $\{0,1,3\}$ modulo 5 to obtain a cyclic BIBD$(5,3,2)$. Using the blocks of this design, we can obtain a G_2-decomposition of $K_5^{[1;1]}$.

Suppose an RBIBD$(v,5,1)$ exists for some positive integer v, and let r be the number of parallel classes in the design. We can think of the RBIBD$(v,5,1)$ as a K_5-decomposition of K_v where the block set can be partitioned into parallel classes. Recall that a parallel class is a set of blocks that partition the vertex set. For a given parallel class, we choose to append either one loop or two loops to every vertex in every block of that parallel class. Every block in the decomposition is then either a $K_5^{[2;1]}$ or a $K_5^{[1;1]}$, and hence every block can be decomposed into copies of either G_1 or G_2. Since vertices within a single parallel class receive the same number of loops, we see that all of the vertices in K_v receive the same number of loops. Depending on how many times we choose to use one loop or two, the vertices in K_v can all have anywhere between r and $2r$ loops. From this, we can obtain a $\mathcal{G}$-decomposition of every graph in $\{K_v^{[r;1]}, K_v^{[r+1;1]}, \ldots, K_v^{[2r;1]}\}$.

In general, resolvable designs could lead to many other results for $\mathcal{G}$-decompositions using this type of approach. We could have let the graphs in Example 6.1 have different loop colours, leading to decompositions of even more multigraphs. A similar construction to the one in Example 6.1 could also be formed using uniformly resolvable designs. For example, given a family $\mathcal{G}$ of graphs, we could use BIBDs, as in Example 6.1, to decompose a set of looped cliques; these cliques could vary in their size and number of loops. Using a URD, we could form a $\mathcal{G}$-decomposition of a larger multigraph, where we choose how many loops we want in the various parallel classes.

Another possible direction involves specifying the proportion of times each vertex appears in each graph of $\mathcal{G}$ in a $\mathcal{G}$-decomposition. Theorem 6.2 gives a result, due to Wilson, which explores this problem in the case of simple graphs.

Theorem 6.2. [28] *Let $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$ be given, where the graphs G_i are pairwise nonisomorphic and where each has at least one edge. Let*

$$\alpha(\mathcal{G}) = \gcd\{d_u : u \in V(G), G \in \mathcal{G}\} \quad \textit{and} \quad \beta(\mathcal{G}) = \gcd\{2m(G) : G \in \mathcal{G}\}.$$

Let $p_1, p_2, \ldots, p_t$ be nonnegative real numbers that sum to 1, and let $\epsilon > 0$. For every sufficiently large integer v satisfying

$$\begin{aligned} v(v-1) &\equiv 0 \pmod{\beta(\mathcal{G})} \quad \textit{and} \\ v-1 &\equiv 0 \pmod{\alpha(\mathcal{G})}, \end{aligned}$$

there exists a $\mathcal{G}$-decomposition of K_v in which for every point x, the proportion of copies of G_i that appear in the decomposition and that contain x is within ϵ of p_i for all $1 \leq i \leq t$.

If the graphs in $\mathcal{G}$ are adorned with loops in a specific way, then this result might be useful in ensuring that every vertex in K_v receives the same number of loops. There is an error term in Wilson's result, so we may only obtain approximate decompositions. However, we do believe there is potential in taking this type of approach.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows Theory, Algorithms, and Applications*, Prentice Hall, Inc., Upper Saddle River, 1993.

[2] D. Beihoffer, J. Hendry, A. Nijenhuis, and S. Wagon, Faster algorithms for Frobenius numbers, *Electronic Journal of Combinatorics* **12** (2005), R27.

[3] A. Bonisoli, S. Bonvicini, and G. Rinaldi, A hierarchy of balanced graph-designs, *Quaderni di Matematica* **28** (2012), pp. 151-163.

[4] A. Bonisoli and B. Ruini, Balance, partial balance and balanced-type spectra in graph-designs. *J. Combin. Math. and Combin. Comput.* **93** (2015), pp. 3-22.

[5] S. Bonvicini, Degree- and orbit-balanced Γ-designs when Γ has five vertices, *J. Combin. Designs* **21** (2013), pp. 359-389.

[6] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs & Digraphs*, Chapman & Hall/CRC, 2015.

[7] C.J. Colbourn and J.H. Dinitz, *The CRC Handbook of Combinatorial Designs*, 2nd ed., CRC Press, Inc., Boca Raton, 2006.

[8] C. J. Colbourn and V. Rödl, Percentages in pairwise balanced designs, *Discrete Mathematics* **77** (1989), pp. 57-63.

[9] P. Dukes, A. Ling, and A. Malloch, Thickly-resolvable block designs, *Australas. J. Combin.* **64** (2016), pp. 379-391.

[10] P. Dukes and A. Malloch, An existence theory for loopy graph decompositions, *J. Combin. Designs* **19** (2011), pp. 280-289.

[11] E. A. Dinic, An algorithm for the solution of the problem of maximal flow in a network with power estimation, *Soviet Math. Dokl.* **11** (1970), pp. 1277-1280.

[12] J. Edmonds and R. A. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of ACM* **19** (1972), pp. 248-264.

[13] P. Elias, A. Feinstein, and C. E. Shannon, A note on the maximum flow through a network, *IRE Trans. on Inform. Theory* **IT 2** (1956), pp. 117-119.

[14] L. R. Ford, Jr. and D. R. Fulkerson, Maximal flow through a network, *Canad. J. Math* **8** (1956), pp. 399-404.

[15] L. Gionfriddo, M. Gionfriddo, and G. Ragusa, Equitable specialized block colourings for 4-cycle systems —I, *Discrete Math* **310** (2010), pp. 3126-3131.

[16] M. Gionfriddo and G. Quattrocchi, Colouring 4-cycle systems with equitable coloured blocks, *Discrete Math* **284** (2004), pp. 137-148.

[17] P. Hell and A. Rosa, Graphs decompositions, handcuffed prisoners and balanced P-designs, *Discrete Math* **2** (1972) no. 3, pp. 229-252.

[18] R. Kannan, Lattice translates of a polytope and the Frobenius problem, *Combinatorica* **12** (1992) no. 2, pp. 161-177.

[19] E. R. Lamken and R. M. Wilson, Decompositions of edge-colored complete graphs, *J. Combin. Theory*, Series A, **69** (2000), pp. 149-200.

[20] S. Li and C. A. Rodger, Equitable block-colorings of C_4-decompositions of $K_v - F$. *Discrete Math* **339** (2016), pp. 1519-1524.

[21] A. Malloch, "The asymptotic existence of graph decompositions with loops," M.Sc. thesis, Department of Mathematics and Statistics, University of Victoria, 2009.

[22] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.

[23] D. R. Stinson, *Combinatorial Designs: Constructions and Analysis*, Springer, New York, 2004.

[24] L. Teirlinck, On large sets of disjoint ordered designs, *Ars Combin.* **25** (1988), pp. 31-37.

[25] L. Teirlinck, Large sets of disjoint designs and related structures, *Contemporary Design Theory: A Collection of Surveys*, Wiley-Intersci. Ser. Discrete Math. Optim., Wiley-Intersci. Publ., Wiley, New York (1992), pp. 561-592.

[26] R. M. Wilson, Decompositions of complete graphs into subgraphs isomorphic to a given graph, *Proc. of the Fifth British Combinatorial Conference*, Univ. Aberdeen, Aberdeen (1975), pp. 647-659.

[27] R. M. Wilson, An existence theory for pairwise balanced designs, II, *J. Combin. Theory* **13** (1972), pp. 246-273.

[28] R. M. Wilson, The proportion of various graphs in graph designs, *Combinatorics and Graphs: The Twentieth Anniversary Conference of IPM Combinatorics*, American Mathematical Society, Providence, RI (2010), pp. 251-255.