Fall 2016

Exploring Tournament Graphs and Their Win Sequences

Sadiki O. Lewis
Bard College, sl1157@bard.edu

Exploring Tournament Graphs and Their Win Sequences

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Sadiki Lewis

Annandale-on-Hudson, New York
December, 2016

Abstract

In this project we will be looking at tournaments on graphs and their win sequences. The main purpose for a tournament is to determine a winner amongst a group of competitors. Usually tournaments are played in an elimination style where the winner of a game advances and the loser is knocked out the tournament. For the purpose of this project I will be focusing on Round Robin Tournaments where all competitors get the opportunity to play against each other once. This style of tournaments gives us a more real life perspective of a fair tournament. We will model these Round Robin Tournaments on tournament graphs which are connected graphs with directed edges. With a value placed on a win and loss, I will explore these conditions and look at what can be said about the win sequences of these tournament graphs. With these conditions, patterns were detected with small sized tournaments and were proved for all tournaments. Other patterns still remain as conjectures.

Contents

Dedication

To all that find math difficult and struggle to grasp, this is for you. To my future students, to the babies in the womb in need of a transformative educator, I dedicate this project to you.

Acknowledgments

Thank you to my advisor Lauren Rose, your help and support throughout this process has been extremely comforting. I was able to explore a topic that I both had no idea about but was still interested in. You went on this journey with me and help made it one of the best academic experience I've had thus far. I want to thank my professors at Bard for what has been an enriching academic feast. You've made me even more hungry for knowledge and the pursuit of it. Major thanks to the Math Department. Every step of the way has been challenging. To my moderation board and professors, Amir Barghi, Jim Belk and Keith O'Hara, thanks for the support and believing that I could do it. To John Cullinan, Real Analysis has been both the most difficult and most rewarding math course I've taken, thank you.

To my BEOP family, Caribbean Students Association, Brothers at Bard and Step Team, thank you for providing a space to just be. For the laughs, the support, the love and fun, it has carried me throughout my years at Bard. I don't know what my time at Bard would've been if I didn't have any of you around to connect with, be reminded of home and to find a new home. To my cousins and best friend from back home, Casia, Camoy, Kibi and Uchenna, those group chats really provided laughter and relief amongst stress and moments of doubt. You truly are the real MVPs. To my bros Elliot, Noah, Jeszack, Dariel and Wailly, thank you. To Sky, Abi, Antionette, Bri, Imani, G and Winta, thank you.

To Jane Duffstein, thank you for your never ending support and being there for me. To Truth Hunter, thank you for always believing in me when I sometimes lost faith in myself and helping me to get into IRT. Mary Ann thank you for allowing me to be and just being you. For reviewing my project, I thank Marley, Tayler and Kim, I don't know where I would've been without your feedback.

Finally, I want to thank my mommy and daddy for constantly believing in me and encouraging me to the best that I can be. Thank you mom for being so involved, you never missed a parent teacher conference even in my last year of high school. Your love and support was crucial for me to be in the position that I am in now. Thanks to my whole family back home in New Jersey and Jamaica. To my siblings, thank you. Knowing I had support from all over help made this possible. To my culture, to Jamaica and all that it is, thank you for providing me with this unspoken confidence in myself.

To everyone that I've ever meet and has impacted my experience and life at Bard, as we say in Jamaica, Big up yuh self. Much love and blessings.

Introduction

This project will explore what it means to model real life tournaments on graphs and what information can be drawn from their win sequences. This project was inspired by Christiane Koffi's senior project on *Exploring a Generalized Partial Borda Count Voting System* [3]. In her conclusion she posed several questions about how to use properties of voting systems in tournaments. Her questions included: "How do the score sequences of a tournament vary as we increase the number of players." and "Is it possible to find a formula that counts all the possible N-player tournaments?" From her questions I started to explore and focus on how real life tournaments, where competitors are competing for a prize, can be modeled on graphs and what can be said about the graphs.

A tournament is a series of contests between a number of competitors who compete for an overall prize. We see tournaments occur all the time in sports such as basketball, chess, and tennis. There is usually a predetermined pairing of opponents who play one another through a process of elimination until there is a winner. Let us look at an example of that kind of tournament. In $Figure$ 0.0.1 we have the Lakewood Boys Basketball Tournament. There are four competitors, the Scorpions, Cyclones, Rockets, and Black Crows. All the teams are competing to win the same prize, which is being crowned champion. As we can see the teams are broken up into predetermined pairings. In bracket 1 we have the Scorpions vs Cyclones and in bracket 2 we

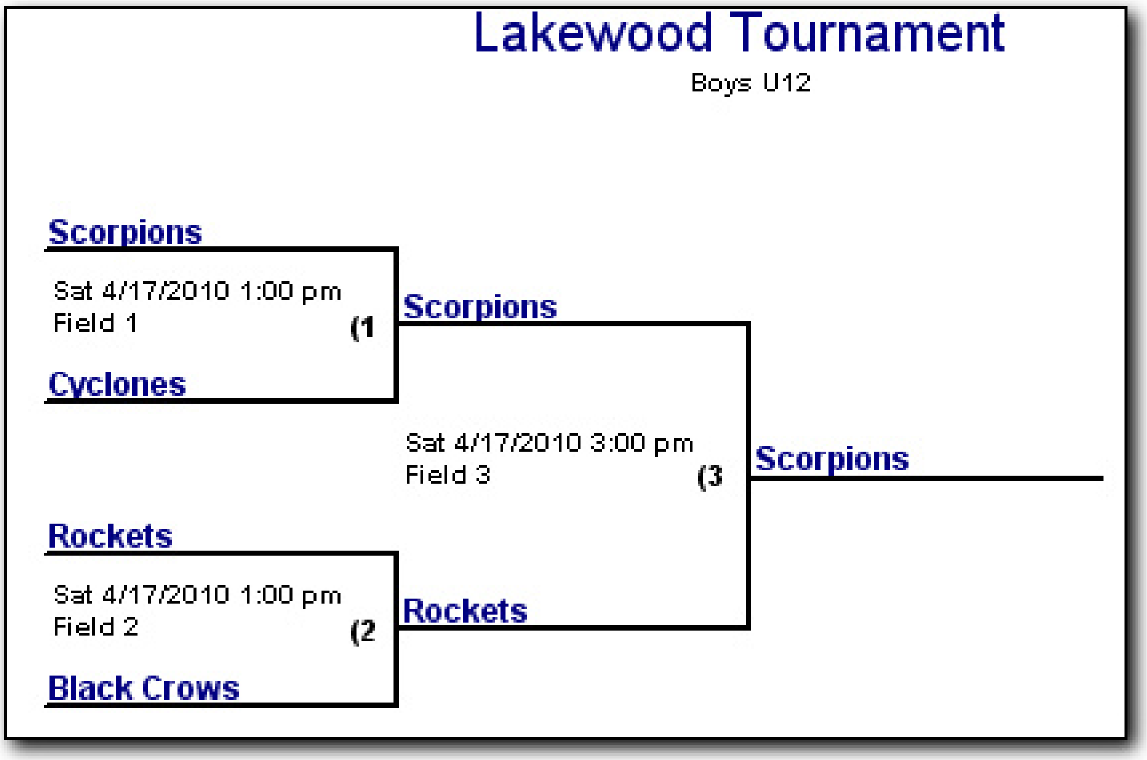

Figure 0.0.1. Basketball Tournament Bracket

have Rockets vs Black Crows. Moving forward into the tournament the Scorpions win and the Cyclones are eliminated. The Rockets also win and the Black Crows are eliminated. The winners proceed forward and face off against each other. Here the Scorpions come out undefeated and take home the championship.

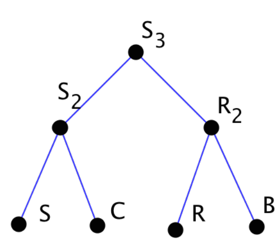

Figure 0.0.2. Basketball Tournament Graph

Now we can begin to model what this specific tournament would look like on a graph. In $Figure$ 0.0.2 we could see how the Lakewood tournament looks like as a graph where the points represent teams and the lines represent the teams advancing to the next round to face one another. For example let us take a look at what that graph would look like. At the bottom of the graph we have our 4 teams and at every step we see which team advances to the next round and then finally at the $3rd$ round we see that Scorpions are the winners. This shows that we can visually represent a tournament on a graph.

For this project I will be modeling Round Robin Tournaments which don't use elimination, which is a competition where all contestants play one another exactly once. This tournament style allows for all competitors to compete against one another and test their skills against all possible opponents.

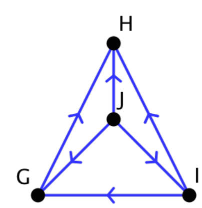

Figure 0.0.3. Tournament Graph of K_4

In $Figure$ 0.0.3 we see how a Round Robin Tournament looks like as a graph. The points on the graph represent teams and the directed line segments represents a game between any 2 teams. Here we have Teams H, I, J, and K. Every team faces off against one another and every team is given a fair chance to face one another. For example, the directed edge with an arrow going from Team I to H indicates a win for Team I. Notice that the arrow is point away from Team I, so that is what a win looks like. The directed edge from Team G to I indicates a loss for Team G. Notice that the arrow is pointing towards Team G, so that is what a loss looks like. We will study and explore these graphs in my project. We will observe the restrictions on forming these graphs and what patterns hold for all tournaments.

In Chapter 1 we will introduce basic graph theory terminology that will serve as a framework for this project. We will look at definitions and lemmas that will help inform us about tournament graphs and their win sequences. In Chapter 2, we will take the graph theory terminology and place it in the context of tournament graphs and their win sequences. In Chapter 3 we will generalize conditions for tournament graphs to exist and explore the cases where a win for a team earns you 1 point and a loss at 0. We will see if these properties and conditions hold for higher orders of teams. Finally in Chapter 4 we will discuss future work that can be done with different conditions set forth in Chapter 3. We will briefly introduce the option for allowing a draw between teams and how the conditions and patterns hold or change. Where a draw is allowed in tournament graphs, we will notice a difference between graphs and win sequences where Win/Lose is an option and where Win/Lose/Draw is allowed.

1
Preliminaries

In this chapter we will define the basic graph theory terminology needed to understand this project. For more details see *Introduction to Graph Theory by Robin Wilson* [1] and *Introduction to Graph Theory by Douglas B. West* [2].

1.1 Basic Graph Theory Definitions

In this section we will define some basic graph theory terminology which includes definitions and theorems that will serve as a framework for this project.

We start by looking at the graphs that we will be using as a framework for this project.

Definition 1.1.1. A **Simple Graph** G consist of a non-empty finite set $V(G)$ of elements called vertices and a finite set $E(G)$ of distinct unordered pairs of distinct elements of $V(G)$ called edges. We call $V(G)$ the **vertex set** and $E(G)$ the **edge set** of G. An edge $\{v, w\}$ is said to join the vertices v and w and is usually abbreviated to vw. $\triangle$

Simple graphs can be visually represented. Let the vertex-set $V(G)$ be $\{M, N, O, P\}$ and edge-set $E(G)$ consists of the edges $\{NM, MO, NO, OP\}$. A visual representation of these sets can be seen in *Figure* 1.1.1. There is a more general definition of a graph. A general graph can

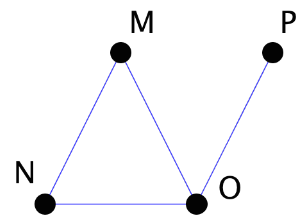

Figure 1.1.1. A Simple Graph

involve multiple edges and loops but for the purpose of this project we will only consider simple graphs.

Notice that the vertex-set $V(G)$ can tell us more about the graph.

Definition 1.1.2. The **order** of a graph G, written $n(G)$ is the number of vertices in G. △

Example 1.1.3. If we take a look at *Figure* 1.1.1, we can see that $n(G) = 4$ since our vertex-set $V(G) = \{m, n, o, p\}$ and consists of 4 vertices. ◊

Looking at the vertex-set from the previous example, notice in *Figure* 1.1.1 that not all vertices have the same number of edges connected to them. We can define what the number of edges are connected to any vertex.

Definition 1.1.4. The **degree** of a vertex v in a graph G, written $d(v)$ is the number of edges incident to v. △

Example 1.1.5. If we take a look at *Figure* 1.1.1, we can see that $d(M) = 2$, $d(O) = 3$, $d(N) = 2$, and $d(P) = 1$. These values follow because of the number of edges incident to each vertex. ◊

With the degrees of all the vertices, notice that we can begin to arrange the degrees in some order.

Definition 1.1.6. The **Degree Sequence** D_G of a graph consists of the list of vertex degrees, usually written in non-increasing order as $(d(v_1) \geq .... \geq d(v_n))$. △

Example 1.1.7. For example, the degree sequence of the graph in *Figure* 1.1.1 is $(1, 2, 2, 3)$. We took the degrees of all the vertices and rearranged them in non-increasing order. ♢

Now that we have defined a simple graph and characteristics of it. Notice that our simple graph in *Figure* 1.1.1 did not have an edge between every vertices. If we allow for an edge between any two vertices, that will create a new kind of simple graph.

Definition 1.1.8. A simple graph on n vertices in which each pair of distinct vertices are adjacent is called a **complete graph**, denoted K_n. △

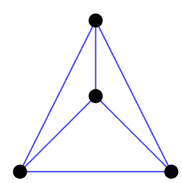

Figure 1.1.2. Complete Graph on 4 vertices

The graph in *Figure* 1.1.2 is a visual representation of what a complete graph looks like on 4 vertices. Notice that there exists an edge between any two vertices.

1.2 Directed Graphs

In this section we consider a variation on the definition of a connected graph, where each edge is given a particular direction, like a one-way street. This is what we will be studying in this project.

Definition 1.2.1. A **directed graph** or **digraph**, D consists of non-empty finite vertex set $V(G)$ of elements called vertices and finite family $E(G)$ or ordered pairs of elements of $V(G)$ called directed edges. △

As with simple graphs, directed graphs can be modeled where the edge ab can be written $a \rightarrow b$, $b \leftarrow a$.

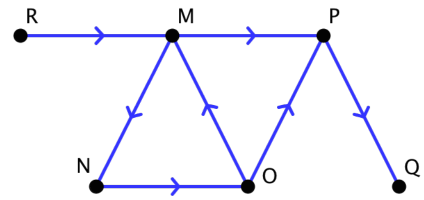

Figure 1.2.1. Directed Graph

A visual representation of a directed graph can be found in *Figure* 1.2.1, with vertex set $V(D) = \{R, M, N, O, P, Q\}$ and edge set $E(D)= \{RM, MN, MP, NO, OM, OP, PQ\}$ with the ordering of the vertices of an edge indicated by an arrow. If we remove the arrows from the graph we would have a simple graph.

Definition 1.2.2. Let v and w be vertices of a digraph D. The **out-degree** $d^+(v)$ is the number of edges of the form vw where there is a directed edge pointing away from v, $v \to w$. The **in-degree** $d^-(v)$ is the number of edges of the form wv, where there is a arrow pointing towards v, $v \leftarrow w$. The **degree** of vertex v can be written as $d(v) = (d^+(v), d^-(v))$. $\triangle$

Example 1.2.3. With the graph in *Figure* 1.2.1 we can figure out the out degree and in degree of all the vertices. First we will focus on the out degree of all the vertices. We have that $d^+(R) = 1$, $d^+(M) = 2$, $d^+(N) = 1$, $d^+(O) = 2$, $d^+(P) = 1$, and $d^+(Q) = 0$. Notice that we get these values from calculating the number of edges coming out the vertex where the arrow is pointing away.

Now we can calculate the in degree of all the vertices. Thus we have $d^-(R) = 0$, $d^-(M) = 2$, $d^-(N) = 1$, $d^-(O) = 1$, $d^-(P) = 2$, and $d^-(Q) = 1$. We get these values from calculating the number of edges coming out of the vertex where the arrow is pointing towards the vertex. $\diamondsuit$

Example 1.2.4. Notice what happens when we add up all the in-degrees and out-degrees of the vertices. If we sum all of the out degrees for graph G we get that $\sum_{v\in V(G)} d^+(v) = 1+2+1+2+1+0 = 7$ and we also get that the sum of the in degrees are $\sum_{v\in V(G)} d^-(v) = 0+2+1+1+2+1 = 7$. Also the number of edges in G, $e(G) = 7$. Notice that the graph in *Figure* 1.2.1 the number of edges e is 7. When sum all the in-degrees of the vertices we get a total of 7. If we sum the out-degrees of all the vertices we get a total of 7. $\Diamond$

Notice that the sum of the out-degrees, in-degrees, and edges equal one another.

Proposition 1.2.5. *In a digraph G, $\sum_{v\in V(G)} d^+(v) = e(G) = \sum_{v\in V(G)} d^-(v)$, where $e(G)$ is the number of edges of G.*

Proof. Notice that on a directed graph, every edge has a direction. Thus each edge contributes exactly 1 in-degree and 1 out-degree to all vertices. Since on every edge there exist an in-degree we can say that $\sum_{v\in V(G)} d^+(v) = e(G)$. Similarly since on every edge there exist an out-degree we can say that $\sum_{v\in V(G)} d^-(v) = e(G)$ Therefore we can conclude that in a digraph G, $\sum_{v\in V(G)} d^+(v) = e(G) = \sum_{v\in V(G)} d^-(v)$ $\square$

Now that we established a foundation of the basic graph theory terminology needed for this project,we will see how this information will inform the rest of this project and our exploration of tournaments.

2
Tournaments

2.1 Tournament Graphs

A tournament is a series of contests where competitors compete for an overall prize. Tournaments occur in all kind of games and sports. This section will gather all our previous terms about Graph Theory together and put them in the context of tournament graphs and their win sequences.

We start by identifying what a tournament graph is in relation to our previous definitions of a complete graph and directed graph.

Definition 2.1.1. A **Tournament Graph** T_G is a directed graph or digraph where any two vertices on a graph are joined by exactly one edge. Note that tournament graphs are complete graphs where each edge is a directed edge. △

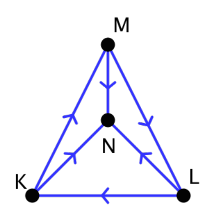

Figure 2.1.1. Tournament Graph on K_4

The graph in *Figure* 2.1.1 is a visual representation of a tournament graph. Between any two vertices there exist exactly one directed edge.

Sometimes tournaments can be modeled differently where all the teams don't play each other. For example, the graph in *Figure* 2.1.2 represents a tournament that isn't a complete graph. There is not a directed edge between any two vertices. Looking at the graph, vertices T and S only have one directed edge. For example notice that there doesn't exist a directed edge between vertices Q and S.

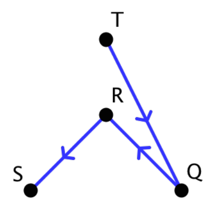

Figure 2.1.2. Unconnected Tournament Graph

In this project we focus on Round Robin Tournaments where every competitor plays each other exactly once. In a tournament graph, vertices are called **teams** and edges represent **games**.

Example 2.1.2. In *Figure* 2.1.1 we have teams K, M, N, and L. Our edge set $E(G)$ which represents games is $\{ML, MN, LN, LK, KN, KM\}$ consists of the edges with the ordering of the vertices in an edge indicated by an arrow. ♢

Similar to a real life tournament, notice that there exist a winner and a loser in any tournament. We can now define what a win and lose are on a tournament graph.

Definition 2.1.3. Each out-degree represents a **win** for team v. △

Example 2.1.4. Looking at the graph in *Figure* 2.1.1 we can count the wins for each team on the tournament graph. Thus we get $d^+(K) = 2$, $d^+(M) = 2$, $d^+(N) = 0$, and $d^+(L) = 2$. ♢

Similarly with the wins, we can also count the losses of teams in a tournament.

Definition 2.1.5. Each in-degree represents a **loss** for a team v. △

Example 2.1.6. Looking at the graph in *Figure* 2.1.1 we can count the losses for each team on the tournament graph. Thus we get $d^-(K) = 1$, $d^-(M) = 1$, $d^-(N) = 3$, and $d^+(L) = 1$. $\Diamond$

Recall from Chapter 1 we were able to collect the degrees of the vertices together into what is called a degree sequence, where the degree sequence is arranged in non-increasing order. Similarly we can collect the in-degrees and out-degrees into a sequence which are the wins and losses in a tournament.

Definition 2.1.7. A **Win Sequence** $S^+ = (s_1^+, s_2^+, ..., s_n^+)$ are the wins of every team on a tournament graph T_G written in non-decreasing order $s_1^+ \geq s_2^+ \geq ... \geq s_n^+$. For a vertex v_i the number of wins $s_i^+ = d^+(v_i)$ $\triangle$

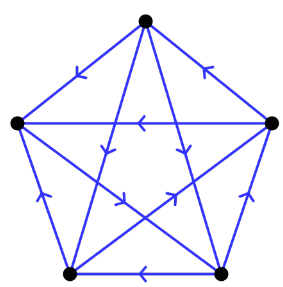

Figure 2.1.3. Tournament Graph on K_5

Example 2.1.8. Looking at the graph in *Figure* 2.1.3 we can get the win sequence S^+ for the tournament graph T_G by looking at the number of wins for each team. Thus we get that $S^+ = (3, 2, 2, 2, 1)$. $\Diamond$

Notice that if we sum the wins in the win sequence of the graph in *Figure* 2.1.3, we get 10. The sum of the wins in the graph equals the number of edges in the graph.

Lemma 2.1.9. *Let T_G be a Tournament Graph on n vertices, If S^+ is a win sequence then* $\sum s_i^+ = \binom{n}{2}$.

Example 2.1.10. In *Example* 2.1.8 we have the win sequence $S^+ = (3, 2, 2, 2, 1)$. Thus if we sum the scores of the sequence we get that $\sum s_i^+ = 3 + 2 + 2 + 2 + 1 = 10$. Now we check to see if $\binom{n}{2} = 10$. Since the tournament graph has 5 teams, we get that $n = 5$. Thus $\binom{n}{2} = \frac{n(n-1)}{2} = \frac{5(5-1)}{2} = \frac{5(4)}{2} = \frac{20}{2} = 10 = \sum s_i^+$. $\Diamond$

Proof of Lemma 2.1.9. We know that $\sum s_i^+$ is the sum of all the wins in a tournament graph. Since there is a win arrow associated with each edge, the sum of all the wins equals the total number of arrows in the graph. Since there is a arrow on each edge, this means that the number of arrows equals the number of edges on the tournament graph, where the total amount of edges in the graph is $\binom{n}{2}$. Hence $\sum s_i^+ = \binom{n}{2}$. $\square$

Similar to the win sequence, we can also arrange the loss into some order.

Definition 2.1.11. A **Lose Sequence** $S^- = (s_1^-, s_2^-, ..., s_n^-)$ are the losses of every player on a tournament graph written in non-increasing order where $s_1^- \leq s_2^- \leq ... \leq s_n^-$. Where v is a vertex, the losses are determined by the sum of the in-degree $d^-(v)$. Where $s_i^- = d^-(v_i)$ $\triangle$

Example 2.1.12. If we look at the graph in *Figure* 2.1.3, we can compute the lose sequence S^- for the tournament graph T_G by looking at the losses for each team. Thus we get $S^- = (1, 2, 2, 2, 3)$. $\Diamond$

Similarly to the sum of the wins, we can sum the number of losses.

Lemma 2.1.13. *Let T_G be a Tournament Graph, if S^- is a lose sequence then* $\sum s_i^- = \binom{n}{2}$.

Example 2.1.14. Similar to *Example* 2.1.10 we have the lose sequence of the tournament graph to be $S^- = (1, 2, 2, 2, 3)$. Thus $\sum s_i^- = 1 + 2 + 2 + 2 + 3 = 10$. Notice that $\binom{n}{2} = 10$. Thus it follows that for this tournament graph $\sum s_i^- = \binom{n}{2}$. Now we will prove that works for all n. $\Diamond$

Proof of Lemma 2.1.13. We know that $\sum s_i^-$ is the sum of all the wins in a tournament graph. Since there is a win arrow associated with each edge, the sum of all the wins equals the total number of arrows in the graph. Since there is a arrow on each edge, this means that the number of arrows equals the number of edges on the tournament graph, where the total amount of edges in the graph is $\binom{n}{2}$. Hence $\sum s_i^- = \binom{n}{2}$. $\square$

Notice that both the sum of the in-degrees and the sum of the out-degrees equal one another.

Lemma 2.1.15. $\sum s_i^+ = \sum s_i^-$

Proof. We know that $\sum d^+(v) = \sum d^-(v)$ by *Proposition* 1.2.5. By definition we know that for a vertex v_i, the number of wins $s_i^+ = d^+(v_i)$. Similarly the number of loss $s_i^- = d^-(v_i)$. Then we have $\sum s_i^+ = \sum s_i^-$. Since we have shown that $\sum s_i^+ = \binom{n}{2}$ and $\sum s_i^- = \binom{n}{2}$. It follows that $\sum s_i^+ = \sum s_i^- = \binom{n}{2}$ □

Lemma 2.1.16. $s_i^+ + s_i^- = n - 1$

Proof. We know that $d(v) = (d^+(v), d^-(v))$ and it follows that $d^+(v) + d^-(v) = n - 1$. Since $s_i^+ = d^+(v_i)$ and $s_i^- = d^-(v_i)$. Therefore it follows that $s_i^+ + s_i^- = n - 1$. □

Now that we have laid the ground work for tournament graphs and proved some basic conjectures. We can now start to place conditions on the tournament graphs and see what else can be said for tournament graphs on n vertices.

3

Win/Lose Findings

In this chapter we will gather all the information from Chapter 2 and further explore Tournament graphs and their win sequences. We will discover and prove other properties that hold for all tournament graphs on n vertices.

3.1 Tournament Graphs

In this section we will explore specifically tournament graphs on n vertices and other properties that can be generalized for all tournament graphs. For the purpose of this chapter we value a win at 1 and a loss at 0. That is how we calculate the score of every team on the tournament graph.

There exist additional properties for tournaments on graphs. We know with real life tournaments that a team could either win or lose. When paired with other players there exist a possibility where one team could win all games and one team could lose all games.

Definition 3.1.1. Let T_G be a tournament graph on n vertices. We say v is **sink** when $d^+(v) = 0$. Thus the degree of v, $d(v) = (d^+(v), d^-(v)) = (0, n-1)$ and thus $d^+(v) = n - 1$ $\triangle$

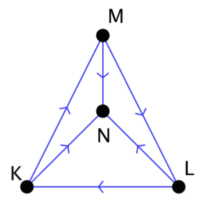

Figure 3.1.1. Tournament Graph on K_4

Looking at the graph in $Figure$ 3.1.1, we can see that there exist a sink in the graph, N. The team located in the middle of the tournament graph has edges with in-degrees incident to that point. Therefore team N loses to all the other teams in the tournament graph.

We now define a concept of winning all games in a tournament graph.

Definition 3.1.2. Let T_G be a tournament graph on n vertices. We say v is a **source** when $d^-(v) = 0$, and thus $d^+(v) = n - 1$. thus the degree of v, $d(v) = (n - 1, 0)$. △

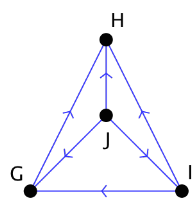

Figure 3.1.2. Tournament Graph on K_4

Looking at the graph $Figure$ 3.1.2 we can see that there exist a source in the graph, J. Team J, located in the middle of the tournament graph, we can see that all the edges incident to J contribute only to the out-degree of J. Therefore team J won every game against all the other teams.

Notice that a tournament graph can have both a source and sink, J and H, as in $Figure$ 3.1.2.

Theorem 3.1.3. *If T_G is a tournament graph on n vertices then T_G has at most 1 source if and only if $s_1^+ = n - 1$ and 1 sink if and only if $s_n^+ = 0$.*

Proof. Case 1: We will prove that a T_G has at most one source if and only if $s_1^+ = n - 1$. Let $v, w \in V$. Suppose v is a source. It follows that $d(v) = ((n - 1), 0)$. This means that v

has an out-degree towards w, $v \to w$. Thus $d^-(w) \geq 1$, hence w is not a source. Now suppose $s_1^+ = n-1$, then v_1 won a game against every other team in the tournament. Thus $v_1^+ = n-1$. Therefore we can conclude that a tournament graph has at most one source.

Case 2: We will prove that a tournament graph T_G has at most one sink if and only if $s_n^+ = 0$. Let $v, w \in V$. Suppose v is a sink. This means that $d(v) = (0, (n-1))$. Then v has an in-degree from w, $v \leftarrow w$ where $d^+(w) \geq 1$, hence w is not a sink. Now suppose $s_n^+ = 0$. This means that vertex v_n lost a games against every team in the tournament. Thus it follows that $v_n^- = n-1$. Therefore we can conclude that a tournament graph has at most one sink. □

We were able to prove that there exists at most 1 sink and 1 source in a tournament graph with n vertices. Having a sink and a source in a tournament graph are not necessarily exclusive. The graph in $Figure$ 3.1.2 shows that there exist both a source and sink in a tournament graph. While the graph in $Figure$ 3.1.1 shows us the possibility of only a sink existing in a tournament graph. We can visually represent a tournament graph with only a source and without both a source and a sink.

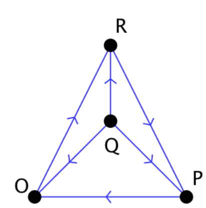

Figure 3.1.3. Tournament Graph on K_4 with a source and no sink

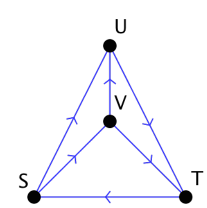

Figure 3.1.4. Tournament Graph on K_4 with no source and sink

The graph in $Figure$ 3.2.8 is an example of a tournament graph where there exist a source with no sink. Team Q will be the source in the tournament graph. The other teams in the graph tie with a score of 1. The graph in $Figure$ 3.1.4 is an example of a tournament graph where neither a sink nor a source exist. Teams V and S tie for a score of 2, while teams T and U tie with a score of 1. Since there is no sink or source, it can be said that this graph represents an example where none of the teams won all games and loss all games. In fact, if neither a source nor a sink exist in a tournament graph then there will be a pair of teams where there is a tie.

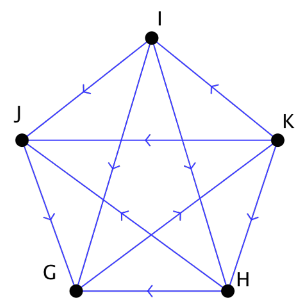

Figure 3.1.5. Tournament Graph on K_5 with no source and sink

The graph in $Figure$ 3.1.5 is an example of a tournament graph where neither a source nor a sink exists and multiple ties occur. Teams I and K both tie for first place with a score of 3. Team H has a score of 2, for second place, while teams G and J tie for last place with a score of 1.

With these examples we can see how various kinds of tournament outcomes can be visually represented on graphs. Next we will look into the conditions that exist for a tournament graph to be formed. To do this we will explore win sequences as well as lose sequences and properties they can generalized for all tournament graphs on n vertices.

3.2 Win Sequence with Win/Lose

In this section we will explore win sequences and lose sequences. Recall from Chapter 2 that in a tournament graph there is either a winner or loser. For the purpose of the project we say that a Win = 1 and a Loss = 0. Therefore it follows that $s_i^+ = d^+(v_i)$ for the wins and $s_i^- = d^-(v_i)$ for the loss.

From Definition 2.1.7 we know that a Win Sequence $S^+ = (s_1^+, s_2^+ ... s_n^+)$ is a sequence of wins of every team on a tournament graph written in non-decreasing order, i.e, $(s_1^+ \geq ... \geq s_n^+)$ and $s_1^+, s_2^+, ..., s_n^+ \in \mathbb{N} \cup \{0\}$

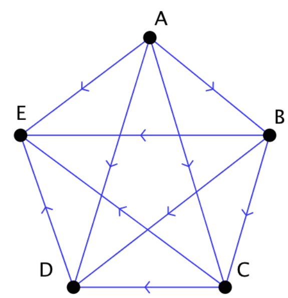

Figure 3.2.1. Tournament Graph on K_5

Example 3.2.1. The graph in $Figure$ 3.2.1 has 5 teams, A, B, C, D, and E. The respective scores for each team are as follows $A = 4$, $B = 3$, $C = 2$, $D = 1$, and $E = 0$. Therefore the win sequence for our tournament graph $ABCDE$ is $(4, 3, 2, 1, 0)$. Notice that given the values for a win and loss that there are restrictions on what scores can occur in a win sequence. Observe that the highest score in our sequence is 4 and the lowest is 0. Recall that every tournament graph has at most 1 source and 1 sink. $\diamondsuit$

Given that there exist at most 1 source and 1 sink in a tournament graph, that places restrictions on the scores of a win sequence. Notice that other scores in the sequence have similar conditions as well.

There exists restrictions on the scores in a sequence so that it can be formed. Observing the previous tournament graphs, we see that there are certain restrictions that allow a win sequence to be formed that yields a tournament graph. We found a number of restrictions on scores in a win sequence. Next we will introduce a theorem that states the restrictions on the scores in a win sequence that produce a tournament graph.

Theorem 3.2.2. *Let* $S^+ = (s_1^+ ... s_n^+)$ *be a win sequence for a tournament graph* T_G*, where* $s_1^+ \geq .... \geq s_n^+$ *then:*

1. $s_1 \leq n-1$
2. $s_2 \leq n-2$
3. $s_1, ..., s_{n-1} > 0$
4. $s_n \leq \lfloor \frac{n-1}{2} \rfloor$
5. $\sum s_i = \binom{n}{2}$

Proof of Theorem 3.2.2. :

(1). Let T_G be a tournament graph with a win sequence $S^+ = (s_1^+, ..., s_n^+)$ where $v, w \in S^+$. Suppose v is a source, so $d(v) = ((n-1), 0)$. Since a tournament graph is a connected graph on n vertices, we know that every team is connected to every other team exactly once. Therefore that team plays $n-1$ teams, so it can win at most $n-1$ games. With the condition that a Win = 1 then we can conclude that $s_1^+ \leq n-1$.

(2). Case 1: There exist a source. Let $v_1, v_2 \in V$ so that $s_2 = d(v_2^+)$ and $s_1 = d(v_1^+)$. Let v_1 be a source in a tournament graph T_G. Then $s_1 = n-1$. Then the $d^-(v_2) \geq 1$. Since there exist a loss for v_2^+, we can say that it will have a maximum score less than s_1^+. Thus $d^+(s_2^+) \leq (n-1) - 1 = n-2$. We can conclude that $s_2^+ \leq n-2$. Case 2: There is no source: If not a source then $s_1 < n-1$. This means that $s_1 \leq n-2$. Since $s_2 \leq s_1$ that means that $s_2 \leq n-2$. We can conclude that $s_2^+ \leq n-2$

(3). Case 1: There exist a sink. Let v^+ be a sink in a tournament graph. Then $d^+(v) = 0 = s_n^+$. We know that if v is a sink then it follows that $d(v) = (0, n-1)$. Let $d(w) = s_{n-1}^+$. Then there exist an in-degree to v, $v \leftarrow w$ for all $w \in S^+$ such that $d^+(w) \geq 1$. Therefore $d^+(w) > 0$, hence $s_{n-1}^+ > 0$. Case 2: There is no sink. If no sink, this means that $s_n > 0$ and it follows that $s_1 \geq s_2 \geq ... \geq s_n > 0$. Thus $s_i > 0$.

(4). We will proceed with proof by contradiction. We know that $\sum s_i^+ = \binom{n}{2} = \frac{n(n-1)}{2}$. Let $s_n > \frac{n-1}{2}$, so $s_n \geq \frac{n}{2}$. Then $s_1, ..., s_n \geq \frac{n}{2}$. If we sum the terms we get $\sum s_i \geq \frac{n}{2} + \frac{n}{2} + ... + \frac{n}{2} = n(\frac{n}{2}) = \frac{n^2}{2} > \frac{n(n-1)}{2}$. Hence we have a contradiction since we started with $\sum s_i^+ = \binom{n}{2} = \frac{n(n-1)}{2}$.

Therefore $s_n^{\leq} \lfloor \frac{n-1}{2} \rfloor$.

(5). Recall the proof of *Lemma* 2.1.9 in Chapter 2. $\square$

Definition 3.2.3. A sequence is **valid** if it satisfies *Theorem* 3.2.2 $\triangle$

Now we will find all valid sequences $n = 2$ to 6and show that they are win sequences of tournament graphs.

If $S=$ $(s_1, ..., s_n)$ where S is arranged in non-decreasing order where $s_i \in \mathbb{N} \cup \{0\}$, we will explore all possible sequences that S could be. We go through this method to construct all valid sequences. Along with the conditions in *Theorem* 3.2.2 we will go through cases to find all valid sequences.

Example 3.2.4. We will compute all valid sequences for $n = 4$. With the conditions from *Theorem* 3.2.2, $n = 4$ becomes:

1.$s_1 \leq 3$

2.$s_2 \leq 2$

3.s_1, s_2, $s_3 \geq 0$

4. $s_4 \leq 1$

5. $s_1 + s_2 + s_3 + s_4 = 6$

By conditions 4, we have two cases to check. Case 1 is when $s_4 = 0$ and case 2 sis when $s_4 = 1$. We proceed with case 1. When $s_4 = 0$ we get a sequence $S= s_1 \geq s_2 \geq s_3 \geq 0$. Therefore we get that s_1 could be either a 2 or 3, s_2 could be 1 or 2, and s_3 could be either 1 or 2. We can get the sequences $(3, 2, 1, 0)$ and $(2, 2, 2, 0)$. Case 2 we get the sequences $(3, 1, 1, 1)$ and $(2, 2, 1, 1)$. The same process we use to construct all valid sequences. $\Diamond$

In fact, these are win sequences with a tournament graph.

Theorem 3.2.5. *All the sequences in Figure* 3.2.2 *are win sequences of tournament graphs.*

Proof. The tournament graphs in *Figure* 3.2.3 and *Figure* 3.2.4 located after the table of win sequences shows us that the wins sequences of $n = 2, 3$, and 4 have tournament graphs. Refer to the appendix for the tournament graphs of $n = 5, 6$. $\square$

Given our restrictions on the scores in a win sequence, we can see that every win sequence can be visually represented by a graph. Therefore we can say that every win sequence in our table has a tournament graph.

The table in $Figure$ 3.2.2 contains all the win sequences up to $n = 6$ that have a tournament graph. Located after the table are the tournament graphs for our win sequences up to $n = 4$. If you refer to the appendix you can find the tournament graphs for $n = 5$ and $n = 6$.

Win Sequences				
n= 2	n = 3	n = 4	n = 5	n = 6
(1,0)	(2,1,0)	(3,2,1,0)	(4,3,2,1,0)	(5,4,3,2,1,0)
	(1,1,1)	(2,2,2,0)	(4,2,2,2,0)	(5,4,2,2,2,0)
		(3,1,1,1)	(3,3,3,1,0)	(5,3,3,3,1,0)
		(2,2,1,1)	(3,3,2,2,0)	(5,3,3,2,2,0)
			(4,3,1,1,1)	(4,4,3,3,1,0)
			(4,2,2,1,1)	(4,4,4,2,1,0)
			(3,3,2,1,1)	(4,4,3,2,2,0)
			(3,2,2,2,1)	(4,3,3,3,2,0)
			(2,2,2,2,2)	(3,3,3,3,3,0)
				(5,4,3,1,1,1)
				(5,4,2,2,1,1)
				(5,3,3,2,1,1)
				(5,3,2,2,2,1)
				(4,4,4,1,1,1)
				(4,4,2,2,2,1)
				(4,3,3,3,1,1)
				(4,4,3,2,1,1)
				(4,3,3,2,2,1)
				(3,3,3,3,2,1)
				(5,2,2,2,2,2)
				(4,3,2,2,2,2)
				(3,3,3,2,2,2)

Figure 3.2.2.

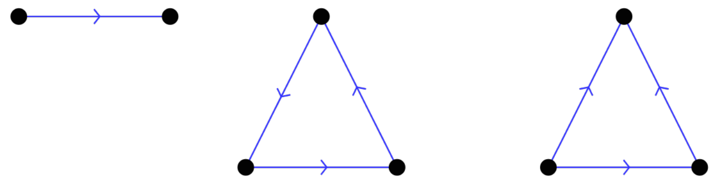

Figure 3.2.3. Tournament Graphs of the win sequences (1,0),(1,1,1) and (2,1,0). (From Left to Right)

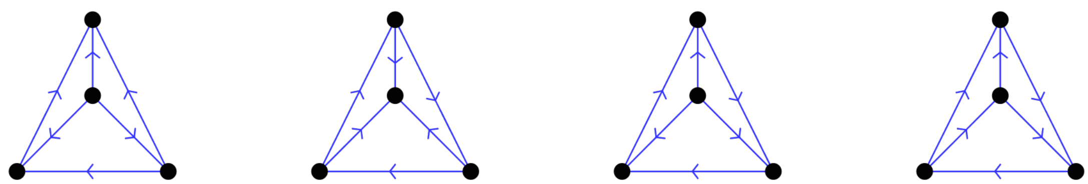

Figure 3.2.4. Tournament Graphs of the win sequences (3,2,1,0), (2,2,2,0), (3,1,1,1), and (2,2,1,1). (From Left to Right)

In the the table in $Figure$ 3.2.2 there exist special cases of win sequences for different n. Given the n, there could be a case where all the teams tie with the same score or all the teams tie with the same score except for 1 team which receives a score of 0.

Example 3.2.6. Let us look at when n is even where $n = 2, 4,$ and 6. We have the following sequences of the form $(1, 0)$, $(2, 2, 2, 0)$ and $(3, 3, 3, 3, 3, 0)$. Where all the entries are the same or 0. Also notice that when n is odd we have special win sequences. When $n = 3$ and 5 we have the following sequences; $(1, 1, 1)$ and $(2, 2, 2, 2, 2,)$. When n is even there exists a tournament outcome where all the teams tie for first place and one team is in last place. When n is odd, there exist a tournament where all teams tie for first place. $\Diamond$

We can generalize that special case when n is even.

Lemma 3.2.7. *If n is even, then there exist a valid sequence S such that* $S = (\frac{n}{2}, \frac{n}{2}, ..., \frac{n}{2}, 0)$.

Proof. We go through the conditions that make a sequence valid. By conditions 1 and 2 it follows that $\frac{n}{2} \leq n - 1$ and $\frac{n}{2} \leq n - 1$. By condition 3 we know that $\frac{n}{2}, ..., \frac{n}{2} > 0$. Since s_n

$= 0$, then condition 4 holds. We know for a valid sequence $\sum s_i = \binom{n}{2}$. Let n be even and $s_1 = s_2 = ... = s_{n-1} = \frac{n}{2}$. Thus we have $S^+ = (\frac{n}{2} + \frac{n}{2} + ... + \frac{n}{2} + 0) = \frac{n}{2}(n-1) = \frac{n(n-1)}{2} = \binom{n}{2}$. □

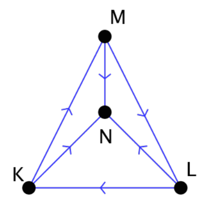

Figure 3.2.5. Tournament Graph on K_4

The graph in *Figure* 3.2.5 is an example of a tournament graph where the scores of the win sequence equals one another and one score is 0. The win sequence of the tournament graph is $(2, 2, 2, 0)$. Where teams M, L, and K tied with a score of 2, while team N has a score of 0.

Similarly we can generalize that special case when n is odd.

Lemma 3.2.8. *If n is odd, then there exist a Win Sequence* $S^+ = (\frac{n-1}{2}, ..., \frac{n-1}{2})$

Proof. Similarly to the proof of *Lemma* 3.2.7 that conditions $1 - 4$ hold for a valid sequence. We know for a valid sequence sequence $\sum s_i = \binom{n}{2}$ Therefore we will sum the terms this odd sequence, $\sum s_i = (\frac{n-1}{2}, \frac{n-1}{2}, ... \frac{n-1}{2}) = \frac{n-1}{2}(n) = \binom{n}{2}$. □

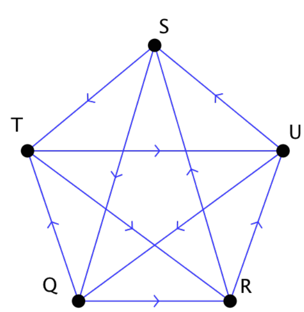

Figure 3.2.6. Tournament Graph on K

The graph in *Figure* 3.2.6 is an example of an odd sequence where all the teams have the same score. The win sequence of the graph is $(2, 2, 2, 2, 2)$.

In any tournament we could see all the outcomes that exist on a tournament graph on n vertices. Now what if we look at the loss of all the teams. Recall *Definition* 2.1.11, which states that a lose sequence $S^- = (s_1^-, s_2^-, ..., s_n^-)$ on a tournament graph are the losses of every player written in non-increasing order $s_1^- \leq s_2^- \leq ... \leq s_n^-$.

Win Sequence	Lose Sequence
(3,2,1,0)	(0,1,2,3)
(3,1,1,1)	(0,2,2,2)
(2,2,1,1)	(1,1,2,2)
(2,2,2,0)	(1,1,1,3)

Figure 3.2.7. Win/Lose Sequence Table for $n = 4$

Example 3.2.9. If we take a look at the table in *Figure* 3.2.7 we were able to compute all the lose sequences for $n = 4$. Notice that for the win sequences, $(3, 2, 1, 0)$ and $(2, 2, 1, 1)$ their lose sequence is the win sequence rearranged in non-increasing order.

The win sequence of $(3, 1, 1, 1)$ has a lose sequence that is different from itself. Taking the losses of the teams in that tournament will produce a different sequence. The lose sequence for $(3, 1, 1, 1)$ is $(0, 2, 2, 2)$. We could see that the lose sequence is another win sequence in our table written in non-increasing.

Now we will show how to go from a win sequence to a lose sequence. Recall that $s_i^+ + s_i^- = n - 1$ then $s_i^- = n - 1 - s_i^+$. Let us start with the win sequence $(3, 1, 1, 1)$ and produce its lose sequence $(s_1^-, s_2^-, s_3^-, s_4^-)$. We know that $n = 4$. Thus it follows that $n - 1 = 4 - 1 = 3$. Therefore $s_i^+ + s_i^- = 3$. For our win sequence $(3, 1, 1, 1)$ we can say that $s_1^+ = 3$, $s_2^+ = 1$, $s_3^+ = 1$, and $s_4^+ = 1$. Since we know what $n - 1$ and s_i^+, we can solve for the lose sequence. Thus it follows that $s_1^- = (n - 1) - s_1^- = 3 - 3 = 0$. If we do this same method for the rest of the scores in the sequence we get the terms of the lose sequence to be $s_2^- = 2$, $s_3^- = 2$, and $s_4^- = 2$. If we perform this method on the other win sequences we get all the lose sequences in the table in $Figure$ 3.2.7 ◊

We can see that computing the lose sequence of a win sequence in the set will produce another win sequence in our set written in non-increasing order.

Similarly we can find the lose sequence from looking at a tournament graph. If we have a tournament graph, if we invert the vertices, i.e, switch the arrows, we can produce the lose sequence on tournament graphs.

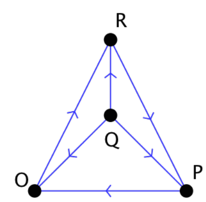

Figure 3.2.8. Tournament Graph on K_4

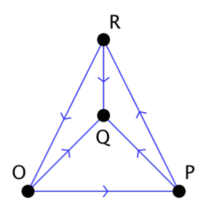

Figure 3.2.9. Tournament Graph on K_4

The graph in $Figure$ 3.2.8 represents a tournament with the win sequence $(3,1,1,1)$. The points of the teams are as follows, $Q = 3$ and $P, R, O = 1$. Now if we want to find the lose sequence of this graph, we will invert the arrows, i.e, flip the direction of the arrow on every edge into the opposite direction. Hence the graph in $Figure$ 3.2.9 show us the lose sequence graph of the win sequence $(3,1,1,1)$. In this graph the points of the teams are now different. Team Q has 0 points. While the scores of teams R, P, O are all 2. Thus we get the lose sequence $(2,2,2,0)$

Definition 3.2.10. Every win sequence is a **Dual** of another sequence when you compute its lose sequence. △

The table in *Figure* 3.2.7 confirms that every win sequence is a dual of another sequence when you compute its lose sequence. Notice that some sequences produce a dual of itself. We can then categorize these sequences.

Definition 3.2.11. A win sequence S^+ is **Self Dual** when the lose sequence is the win sequence rearranged in non-increasing order. △

Example 3.2.12. If we take a look at the sequences $(3, 2, 1, 0)$ and $(2, 2, 1, 1)$ from the table in *Figure* 3.2.7 we know that these win sequences are self dual. Let us take a look at another set of sequences. For example, when $n = 5$ we have the win sequences $(3, 2, 2, 2, 1)$ and $(4, 3, 2, 1, 0)$. Respectively their lose sequences are $(1, 2, 2, 2, 3)$ and $(0, 1, 2, 3, 4)$. Notice that for the win sequences of $n = 5$ to be self dual that a 2 was present in the middle of the sequence. For the case of the win sequences when $n = 4$, for $(3, 2, 1, 0)$ to be self dual, there was a pairing of scores in the sequence where they sum to $n - 1$. So for $(3, 2, 1, 0)$, the scores 3 and 0 will pair since $3 + 0 = 3 = n - 1$. Similarly the scores 2 and 1 will pair since $2 + 1 = 3 = n - 1$. ♢

Thus we can see that there are special conditions to construct self dual sequences when n is odd or even.

Theorem 3.2.13. *Let* $S^+ = (s_1^+, s_2^+, ..., s_k^+)$ *be a win sequence.*

1. *If n is even, where $n = 2k$, then S^+ is self dual when it is of the form* $(s_1^+, s_2^+, .., s_k^+, n - 1 - s_k^+, ..., n - 1 - s_2^+, n - 1 - s_1^+)$.

2. *If n is odd, where $n = 2k+1$, S^+ is self dual when it is of the form* $(s_1^+, ..., s_k^+, \frac{n-1}{2}, n - 1 - s_k^+, ..., n - 1 - s_1^+)$. *Where the middle term in the win sequence* $s_{k+1} = \frac{n-1}{2}$ *and* $S_{k+1}^+ = S_{k+1}^-$

Proof. Case 1:

Suppose n is even where $n = 2k$. Then we have an even win sequence of the form $S^+ = (s_1^+, ..., s_k^+, n - 1 - s_k^+, .., n - 1 - s_1^+)$. We will show that $\sum s_i^+ = \binom{n}{2}$. We get that $\sum s_i^+ = s_1^+ + s_2^+ + s_3^+ + ... + s_k^+ + (2k - 1) - s_k^+ +.. + (2k - 1) - s_3^+ + (2k - 1) - s_2^+ + (2k - 1) - s_1^+ =$

$k(2k-1) = \frac{n(n-1)}{2} = \binom{n}{2}$. Now to confirm that it is self dual we will compute the lose sequence of this wins sequence. Recall that $s_i^+ + s_i^- = n-1$ where $s_i^- = n-1-s_i^+$. Therefore we get the lose sequence $S^- = (n-1-s_1^+, ..., n-1-s_k^+, n-1-(n-1-s_k^+), .., n-1-(n-1-s_1^+))$ $= (n-1-s_1^+, ..., n-1-s_k^+, s_k^+, .., s_1^+)$. Notice that the terms of the sequence are rearranged in non-increasing order, therefore an even win sequence of this form is self dual.

Case 2.

Suppose n is odd where $n = 2k+1$. Then we have an odd win sequence of the form $S^+ = (s_1^+, ..., s_k^+, s_{k+1}^+, n-1-s_k, ..., n-1-s_1^+)$.

Now for the case when n is odd we want to check first that $s_{k+1}^+ = s_{k+1}^-$. Given that s_{k+1}^+ is the middle term in our odd sequence, we want to make sure that it equals s_{k+1}^- term in the lose sequence. We know that $s_i^- = n-1-s_i^+$ and $s_{k+1} = \frac{n-1}{2}$. Therefore we can say that :

$s_{k+1}^- = n-1-s_{k+1}^+$

$s_{k+1}^- = n-1-\frac{n-1}{2}$

$s_{k+1}^- = \frac{2(n-1)}{2} - \frac{n-1}{2}$

$s_{k+1}^- = \frac{2n-2-n+1}{2}$

$s_{k+1}^- = \frac{n-1}{2}$

Thus by hypothesis since $s_i^- = n-1-s_i^+$ we showed that $s_k^+ = s_k^-$. We can conclude that the middle term must equal the middle term in the lose sequence for the odd win sequence to be self dual.

Similarly in case 1, we will show that the lose sequence of the odd sequence is its win sequence in written in non-increasing order. Thus we have the win sequence

$S^+ = (s_1^+, ..., s_k^+, s_{k+1}^+, n-1-s_k, ..., n-1-s_1^+) = (s_1^+, ..., s_k^+, \frac{n-1}{2}, n-1-s_k, ..., n-1-s_1^+)$.
Thus we have the lose sequence $S^- = (n-1-s_1^+, ..., n-1-s_k^+, \frac{n-1}{2}, n-1-(n-1-s_k), ..., n-1-(n-1-s_1^+)) = (n-1-s_1^+, ..., n-1-s_k^+, \frac{n-1}{2}, s_k^+, ..., s_1^+)$. Since the lose sequence is the win sequence written in non-increasing order, we can say that an odd sequence of this form is self dual. $\square$

Now we can produce all the self dual win sequences given an n. We will compute all the self dual sequences up to $n = 7$. If we take a look at the table in *Figure* 3.2.10 we can see all the win sequences that are self dual.

Self Dual Win Sequences

n = 2	n = 3	n = 4	n = 5	n = 6	n = 7
(1,0)	(2,1,0)	(3,2,1,0)	(4,3,2,1,0)	(5,4,3,2,1,0)	(6,5,4,3,2,1,0)
	(1,1,1)	(2,2,1,0)	(3,3,2,1,1)	(4,4,3,2,1,1)	(5,5,4,3,2,1,1)
			(4,2,2,2,0)	(5,3,3,2,2,0)	(6,4,4,3,2,2,0)
			(3,2,2,2,1)	(4,3,3,2,2,1)	(5,4,4,3,2,2,2)
			(2,2,2,2,2)	(3,3,3,2,2,2)	(4,4,4,3,2,2,2)
				(4,4,4,1,1,1)	(6,5,3,3,3,1,0)
					(5,5,3,3,3,1,1)
					(6,4,3,3,3,2,0)
					(5,4,3,3,3,2,1)
					(4,4,3,3,3,2,2)
					(6,3,3,3,3,3,0)
					(5,3,3,3,3,3,1)
					(4,3,3,3,3,3,2)
					(3,3,3,3,3,3,3)
					(5,5,5,3,1,1,1)

Figure 3.2.10.

Notice that the self dual win sequences in *Figure* 3.2.10 follow our proven restriction from *Theorem* 3.2.13. With all the win sequences that are self dual, we want to see if we can produce all self dual win sequences with previous win sequences. Now we will run through a couple of examples to see if we can produce all the self dual win sequences from previous self dual win sequences.

We will look at an example from producing self dual win sequences. The first case is going from when n is an even win sequence to when n is an odd win sequence. The other case is going from when n is an odd even win sequence to when n is an even win sequence.

Example 3.2.14. Let us take a look at $n = 2$ going to $n = 3$. The only self dual sequence for $n = 2$ is $(1, 0)$. Proceeding to the self dual sequences of $n = 3$ we go back to the conditions of a win sequence that $s_1 \leq n - 1$. Hence we add another team to the sequence $(1, 0)$ and check all possible scores for the added team. Thus we get $2/1(1, 0)$. Now we check to see if we add a team with a score of 2, will that produce another self dual sequence and if we add a team with score 1, will that produce a self dual sequence. We adding these new scores to the win sequence, we place that score at the front of the score sequence. This to fulfill the conditions of the sequence being self dual for odd we must make sure it has the conditions previously stated. So we adjust the right hand of the sequence to produce the new set of self dual win sequences. Hence from *Theorem* 3.2.13 we know the properties for an odd sequence to exist. Thus we get $(2, 1, 0)$ and $(1, 1, 1)$. Thus we get all the self dual win sequences for $n = 3$ from $n = 2$.

Now we check going from an odd case, $n = 3$ to an even case, $n = 4$. For $n = 3$ we have the win sequences $(2, 1, 0)$ and $(1, 1, 1)$. For both we add team values of either a 3 or 2. Then we check to see if they will produce self dual sequences for $n = 4$. Thus with the sequence $(2, 1, 0)$ with another team added with a value of 3 or 2 we get the sequences $(3, 2, 1, 0)$ and $(2, 2, 1, 1)$. Similarly with the sequence $(1, 1, 1)$ checking for the values of 3 and 3 we also get the win sequences $(3, 2, 1, 0)$ and $(2, 2, 1, 1)$. Thus we are able to produce self dual win sequences from previous self dual win sequences. $\Diamond$

Thus it is only a conjecture that we can produce self dual win sequences from previous self dual win sequences. There was special cases when adjust the right hand side of a win sequences did not suffice to produce all self dual win sequences. To produce the win sequence $(4, 4, 4, 1, 1, 1)$ we had to adjust the right hand side of the win sequence $(3, 3, 2, 1, 1)$ to all for that production of that win sequence. Then we see that the $(4, 4, 4, 1, 1, 1)$ is the only self dual from $n = 6$ that will produce the self dual win sequence $(5, 5, 5, 3, 1, 1, 1)$

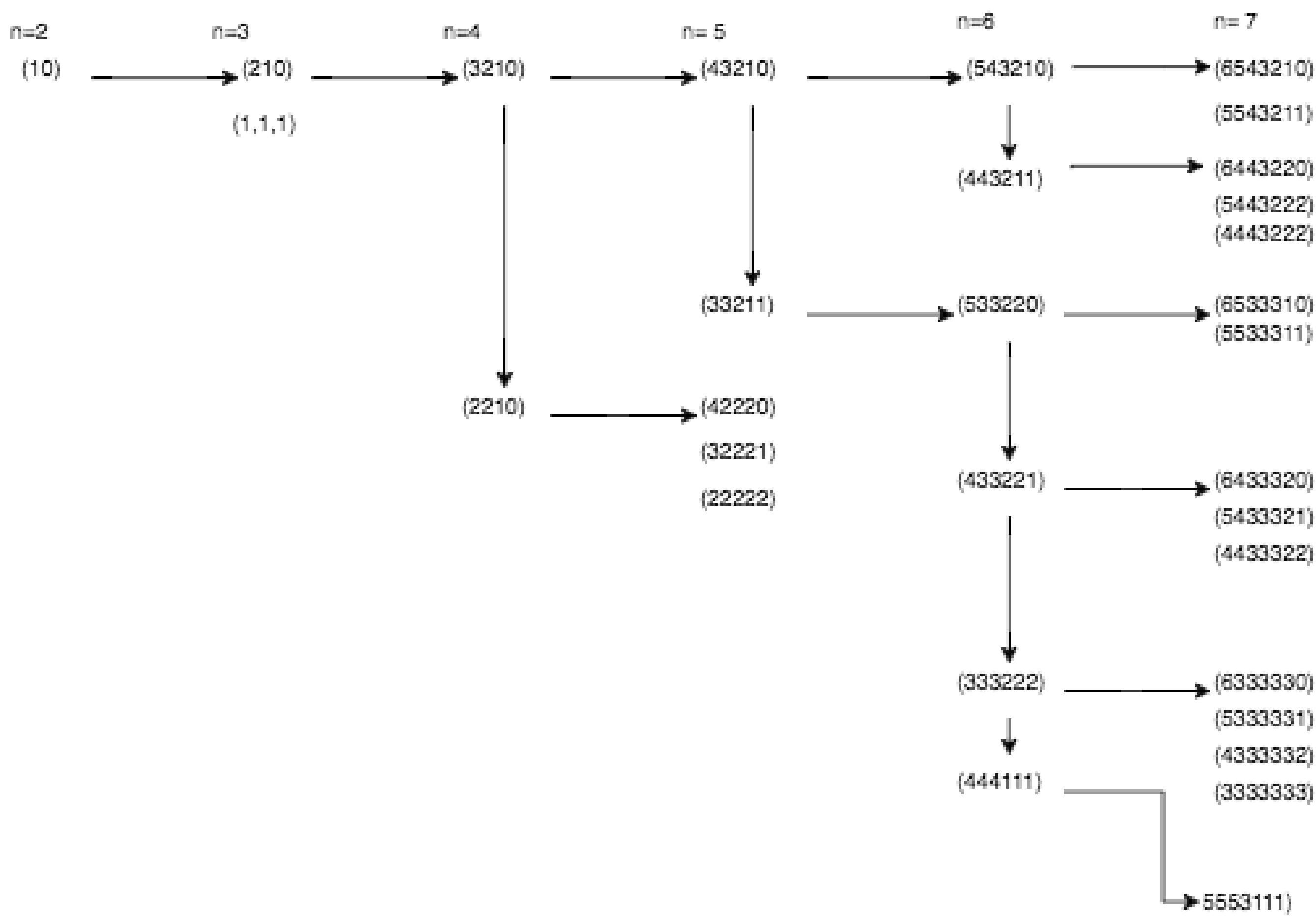

Figure 3.2.11. Production of Self Dual Win Sequences

4
Future Work

4.1 Win/Lose/Draw

This paper examined Tournament Graphs and their win sequences where only win and loss were involved. It was found that every tournament graph has a win sequence. Every win sequence has a lose sequence also that every sequence is a dual of another win sequence in the set. Through a conjecture not yet proven we were able to show that you are able to produce self dual win sequences from previous self dual win sequences. This project focused on tournaments where only win and loss were involved.

Now suppose to we start to model more real life tournament outcomes on graphs. Now suppose we allow for a draw(tie) to be be an option as an outcome between two teams. We notice that the tournament graphs for a win sequence that involves a draw is different then when only win and loss are involve. Here we will introduce draw as an option in the sequence where a $Win = 2$, $Draw = 1$, and $Lose = 0$. Like real world tournaments in sports, it is now favorable for a team to draw then lose to earn a point.

Example 4.1.1. We will look at an example of a win/draw sequence where there is multiple tournament graphs for one sequence. Hence let $n = 4$ and we will work with the sequence $(5, 3, 2, 2)$. First let us observe that there are multiple ways to construct the values in the sequence now that we have draw as an option. The only way to earn a score of 5 is if a team had 2 wins and 1 draw. There are two ways for a team to earn a score of 3. A team could have 1 win and 1 draw or 3 draws. For a team to earn a score of 2, they either won once or draw twice. So there are different combinations of forming scores that yield a tournament graph. ◊

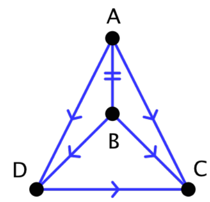

Figure 4.1.1. Win/Lose/Draw Tournament Graph

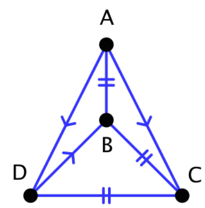

Figure 4.1.2. Win/Lose/Draw Tournament Graph

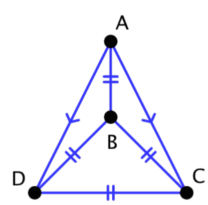

Figure 4.1.3. Win/Lose/Draw Tournament Graph

The graphs in $Figure$ 4.1.1,$Figure$ 4.1.2, and $Figuire$ 4.1.3 represents all the tournament graphs that produce the win sequence $(5, 3, 2, 2)$. Let us observe Team B on the graphs in $Figure$ 4.1.1 and $Figuire$ 4.1.3. Team B both has as a score of 3 in both graphs. In $Figure$ 4.1.1, Team B has a win and a draw to earn 3 points, while on the graph in $Figuire$ 4.1.3, team B

has a draw with all the players in the tournament earning 3 total points. This show that a score can be represented different possible ways on multiple tournament graphs.

Unlike Win/Lose tournament graphs where there is one tournament graph for win sequence, there is the possibility of having more than one tournament graph for a win sequence with draws as an option. Possible areas to further explore:

1. How do the score sequences of the Win/Lose/Draw Tournaments differ from the Win/Lose Tournaments.

2. Is it possible to find a formula to count all possible n-player tournaments for Win/Lose/Draw

3. Is there anyway to produce all, if any self dual win sequences for Win/Lose/Draw

5
Appendix

Listed are all the tournament graphs for $n = 5$ and $n = 6$.

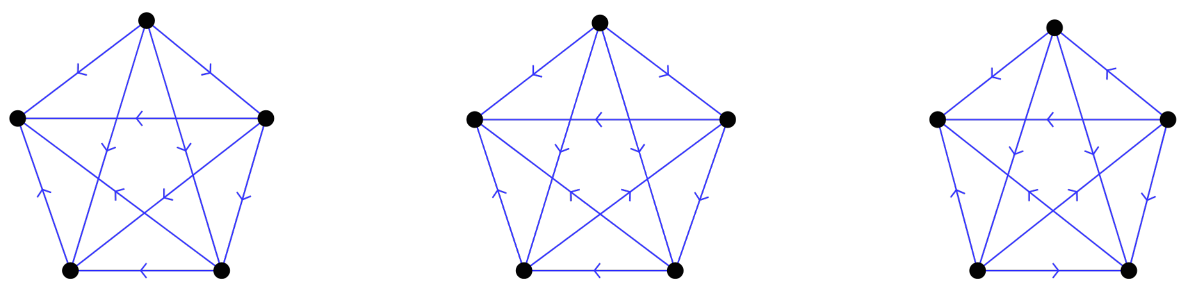

Figure 5.0.1. Tournament Graphs of the win sequences (4,3,2,1,0),(4,2,2,2,0), and (3,3,3,1,0) (From Left to Right)

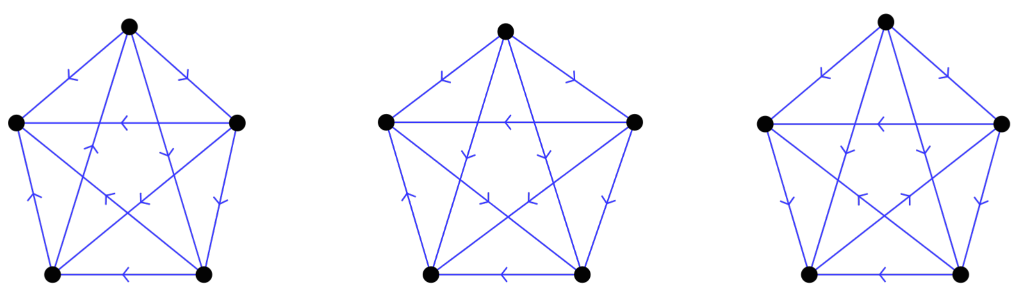

Figure 5.0.2. Tournament Graphs of the win sequences (3,3,2,2,0), (4,3,1,1,1), and (4,2,2,1,1) (From Left to Right)

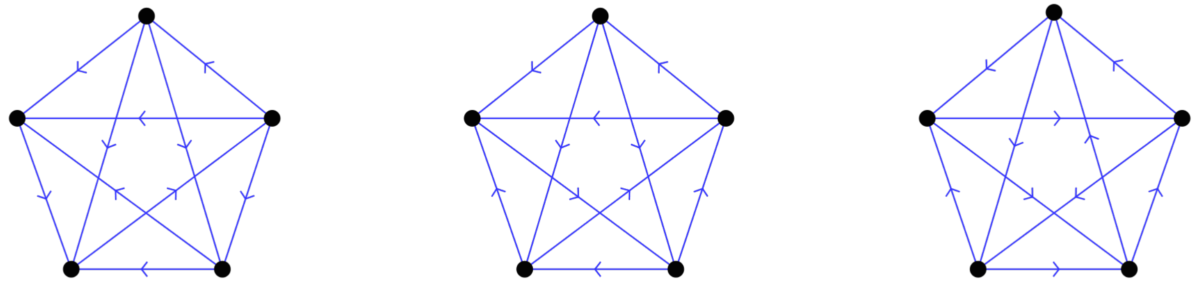

Figure 5.0.3. Tournament Graphs of the win sequences (3,3,2,1,1), (3,2,2,2,1), and (2,2,2,2,2) (From Left to Right)

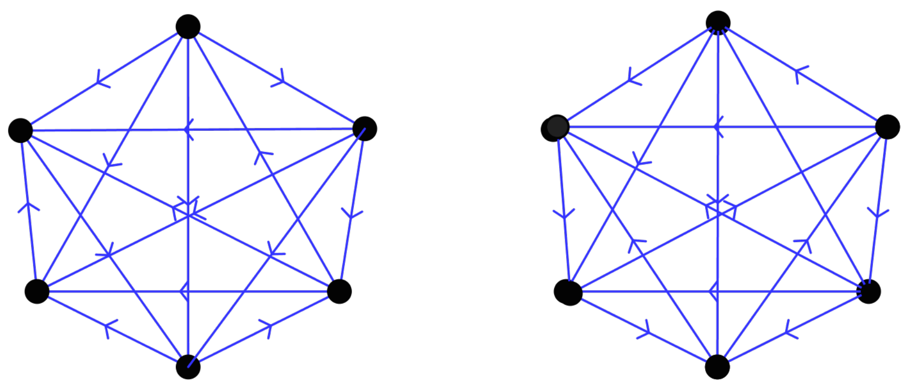

Figure 5.0.4. Tournament Graphs of the win sequences (4,4,3,2,1,1) and (4,3,3,2,2,1) (From Left to Right)

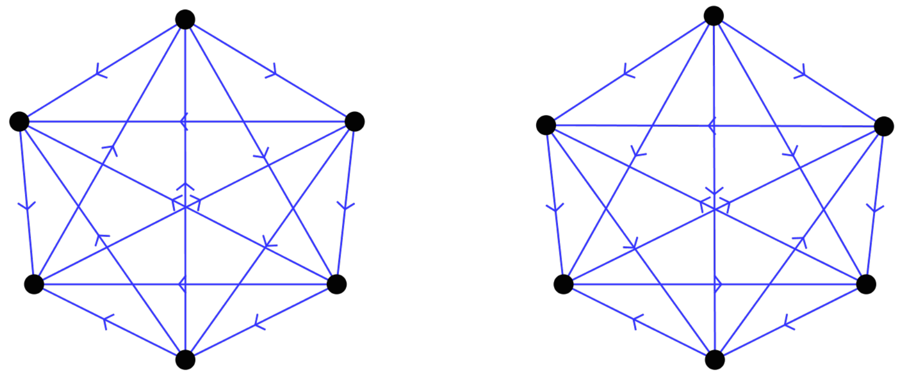

Figure 5.0.5. Tournament Graphs of the win sequences (3,3,3,3,2,1) and (5,2,2,2,2,2) (From Left to Right)

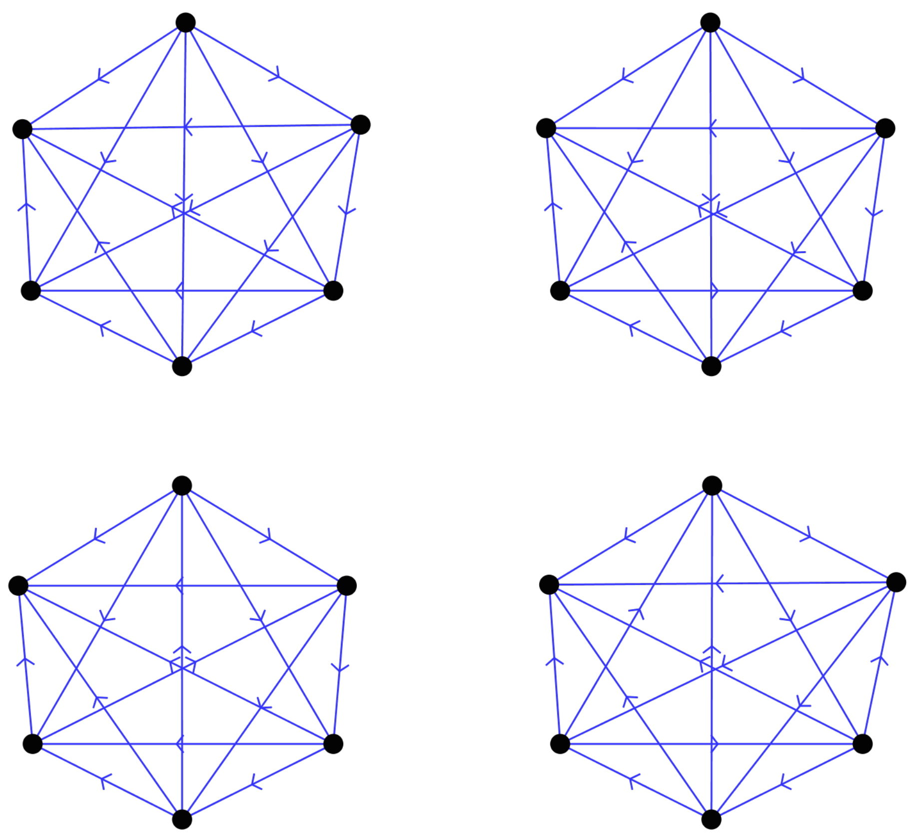

Figure 5.0.6. Tournament Graphs of the win sequences (5,4,4,3,1,0) and (5,4,2,2,2,0) (Top: Left to Right) and (4,3,3,3,2,0) and (3,3,3,3,3,0) (Bottom: Left to Right)

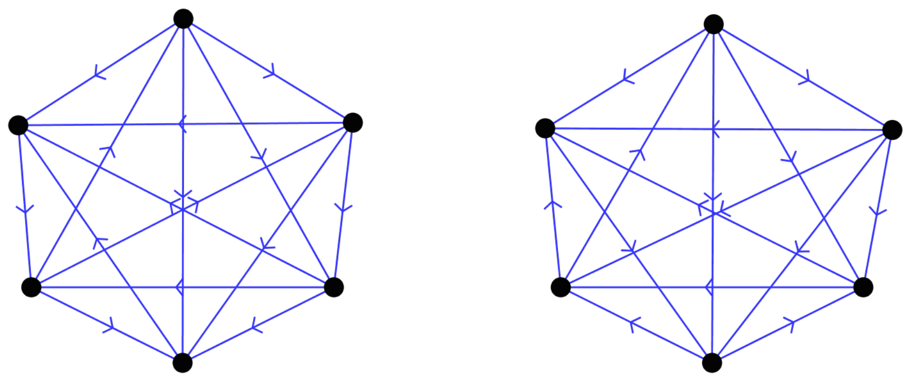

Figure 5.0.7. Tournament Graphs of the win sequences (4,3,3,3,1,1) and (4,4,2,2,2,1) (From Left to Right)

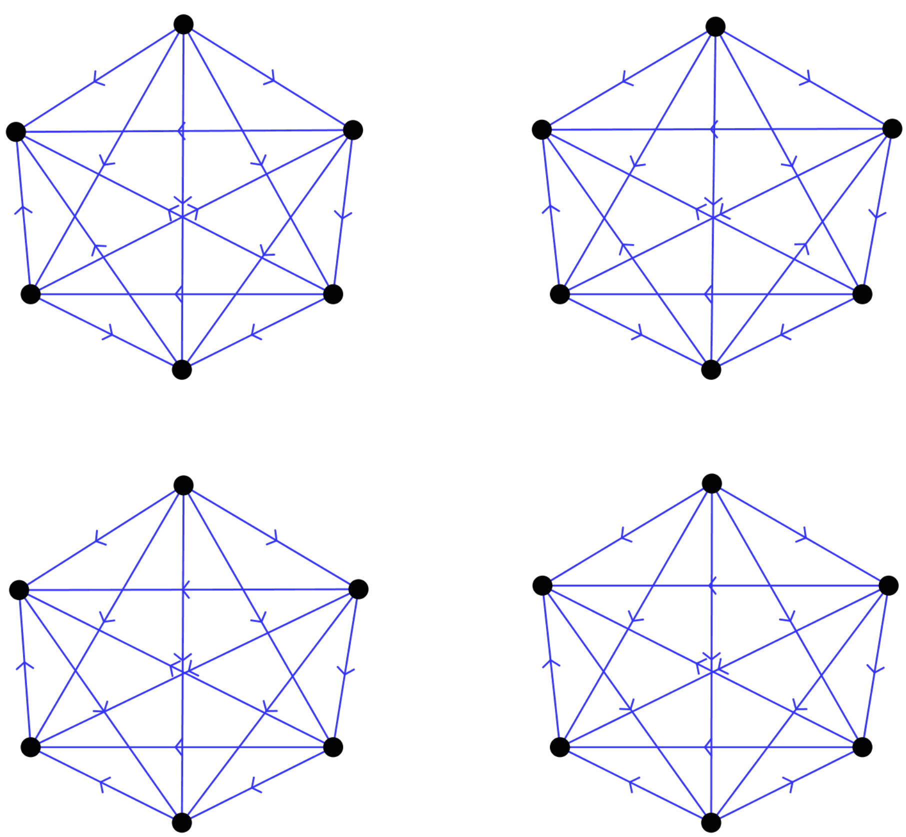

Figure 5.0.8. Tournament Graphs of the win sequences (5,3,3,3,1,0) and (5,3,3,2,2,0) (Top: Left to Right) and (5,4,3,1,1,1) and (5,4,2,2,1,1) (Bottom: Left to Right)

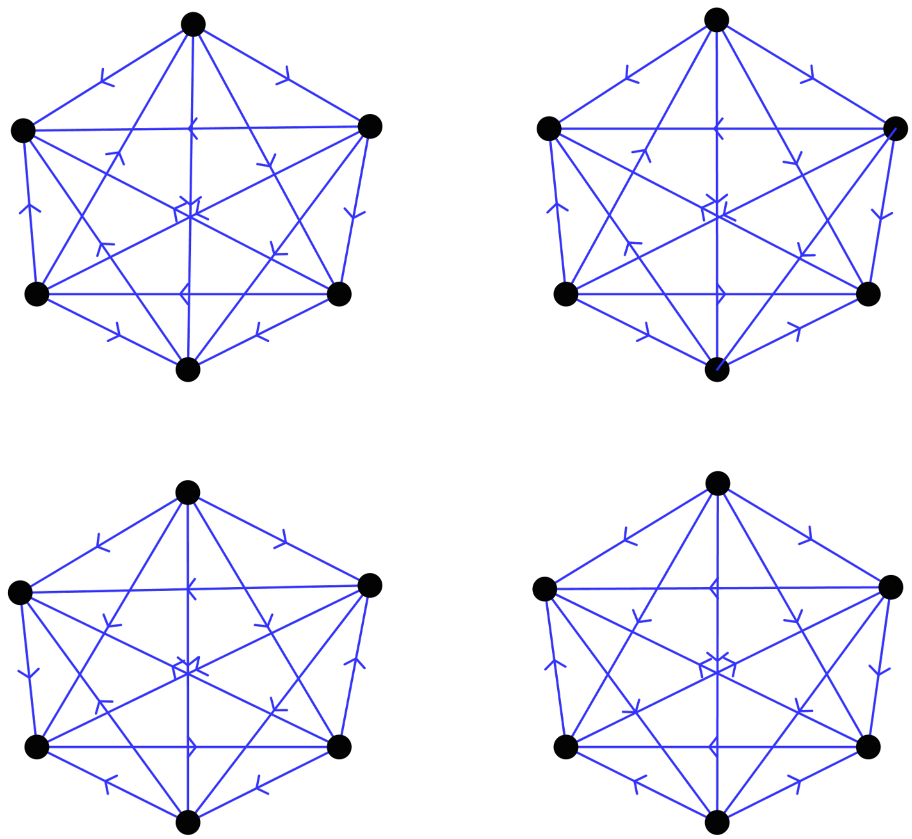

Figure 5.0.9. Tournament Graphs of the win sequences (4,4,3,3,1,0) and (4,4,4,2,1,0) (Top Row: Left to Right) and (5,3,3,2,1,1) and (5,3,2,2,2,1) (Bottom Row: Left to Right)

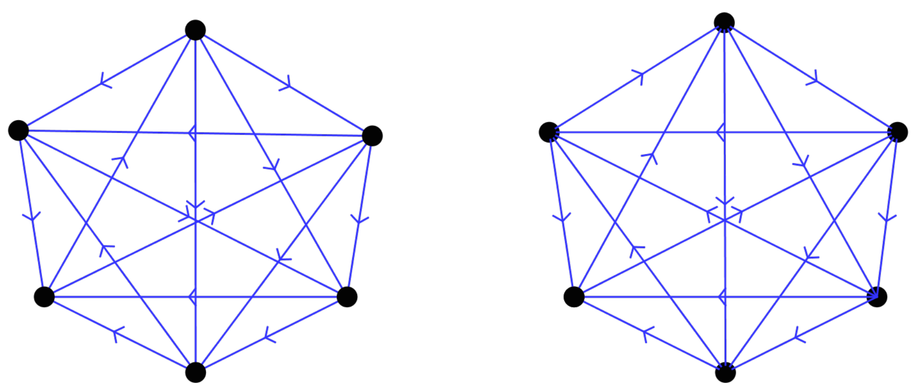

Figure 5.0.10. Tournament Graphs of the win sequences (4,3,2,2,2,2) and (3,3,3,2,2,2) (From Left to Right)

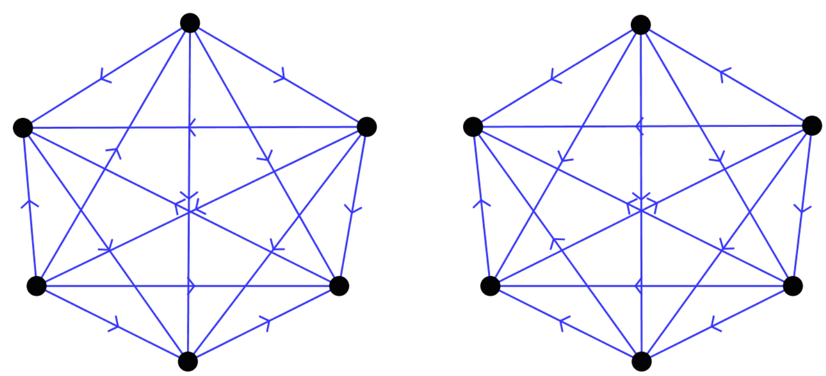

Figure 5.0.11. Tournament Graphs of the win sequences (4,4,4,1,1,1) and (4,4,3,2,2,0) (From Left to Right)

Bibliography

[1] Robin J. Wilson, *Introduction to Graph Theory*, Pearson Eduction Limited, England, 2010.

[2] Douglas B West, *Introduction to Graph Theory*, Prentice Hall, University of Illinois, 2001.

[3] Christiane Koffi, *Exploring a Generalized Partial Borda Count Voting System*, Bard College, United States, 2015.