Scrutiny on Winners and Tournaments

Allysa Tan Li Ying
Cao Jialin, Marianna

1 Introduction

In almost every sport and game, the best team or player is determined through tournaments. In the NBA season, all thirty teams play with each other more than once (amounting to eighty two games for each team), following the top eight teams in each conference play in the elimination playoffs to determine the champion team.

However, there are a few issues in this kind of tournament format. First, teams played unequal number of games against each other, which is unfair due to the difference in strength of the teams. Second, the champion team may win in the finals despite not establishing a 'winning' relationship against the most number of teams in the season.

Offhandedly, the ideal situation seems to be letting every team play with each other the same number of times and determine the best team by selecting the one with the most number of wins. This would then encounter the issue about time as the round-robin format takes a long time, which would make holding tournaments under time constraint or tournaments with a large number of teams difficult. A possible issue that may arise is that more than one team can have the same most number of wins.

In schools and national games championship, we most frequently use a tournament format where random teams are chosen to play against each other and only the winners are able to proceed to the next round. Although it is unfair, it is the fastest and simplest way for the schools to hold a tournament. We hope to find a fair yet efficient tournament format, specifically targeting small-scale tournaments held under time constraint.

2 Objectives

In this project, we aim to find an optimum way to plan tournaments so as to

- ensure a fair system where every team play the same number of games
- reduce the time involved
- decide an effective definition of best team

3 Tournaments and Kings

To investigate the optimum way, we first explore the concept of tournament graphs and kings. We will use the following definitions found in [1].

A *tournament* graph is a graph with a non-empty finite set of vertices in which every two vertices are joined by one and only one arrow (such an arrow is also called an arc or directed edge).

Let T be a tournament and x, y be two vertices in T. If there is an arrow from x to y, we say that x *dominates* y or y is *dominated* by x (symbolically, $x \to y$).

For any two vertices x, y in T, the *distance* from x to y, denoted by $d(x, y)$, is the minimum number of arrows one has to follow in order to travel from x to y. Clearly, $d(x, y) = 1$ if x dominates y.

Let T be a tournament with $n \geq 2$ vertices. A vertex x in T is called the *emperor* if $d(x, y) = 1$ for any other vertex y in T; a vertex x in T is called a *king* if $d(x, y) \leq 2$ for any other vertex y in T.

Ideally, an emperor x is the true winner of the tournament. But in real life, a tournament that is dominated by one team is very rare. It would be more probable for kings to occur in a real life tournament. The problem is that there may exists more than one king in a tournament.

In addition, a tournament is a complete graph, which is not time efficient when it comes to smaller scale competitions.

4 A family of graphs *K*

From the idea of king in a tournament graph, we define a family of graphs K to solve our problems.

A graph in K satisfies the following properties:

(a) It is a regular graph.

(b) For any two vertices u, v in K, $d^*(u,v) \leq 2$, where $d^*(x,y)$ is the minimum number of edges one has to follow in order to travel from x to y for any two vertices x and y in the graph.

In order to reduce the amount of time tournaments span over, we switched out the concept of completeness for regularity. This reduces the total number of edges in the graph, thus reducing the time required for the tournament.

4.1 Obtaining a graph in *K*

We want to find an algorithm to obtain a graph of n vertices in K.

First, we arrange all n vertices in a circle. Without loss of generality, we choose a random vertex to be labelled "0" and go on to label every other vertex from 1 to $n-1$ in an anticlockwise manner. From here, we denote the vertex labelled "p" as vertex "p", where $0 \leq p \leq n$.

We define "skip x cycle", where x is the number of vertices skipped, as follows:

- Starting from vertex "0", we intend to form a cycle by connecting to vertex "$x+1$", followed by vertex "$2x+2$", with an increment of $x+1$ each time, until it connects back to vertex "0".
- If n is divisible by $x+1$, then a total of $\frac{n}{x+1}$ cycles has to be constructed by starting from vertex "1", vertex "2", …, until the number of cycles is achieved.

For example, "skip 0 cycle" implies connecting adjacent vertices.

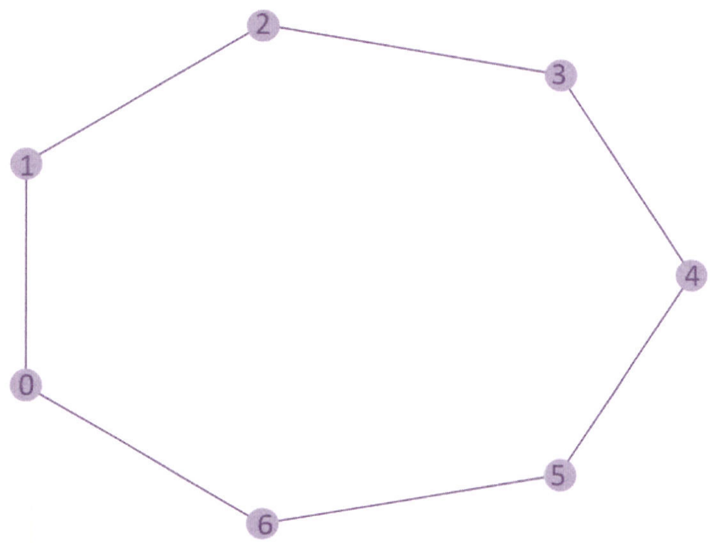

Fig 1: "skip 0 cycle" applied on a graph of 7 vertices

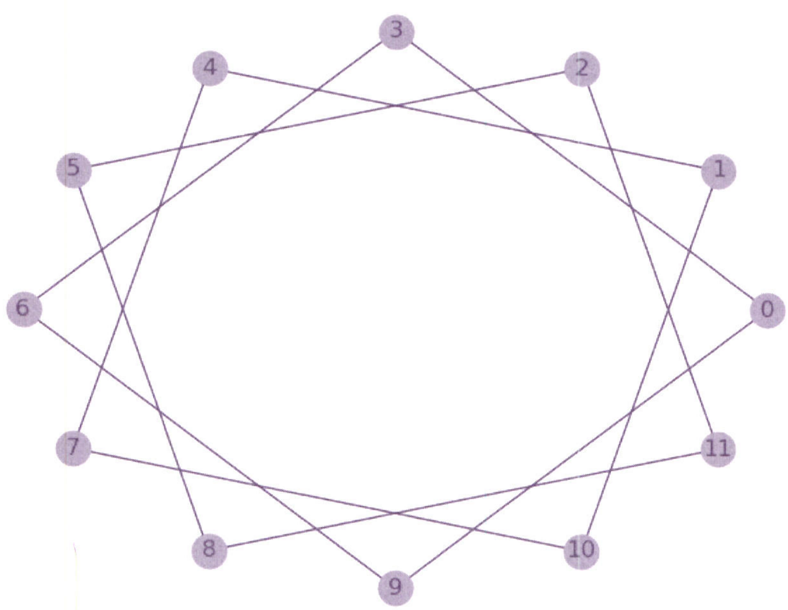

Fig 2: "skip 2 cycle" applied on a graph of 12 vertices

To minimise the number of games played, we tried to limit the number of "skip cycles" to two. The question is now which two "skip x cycle" should be chosen in order for the graph to be in the K family.

We first investigate a graph with seven vertices. Let the two skip cycles be "skip x_1 cycle" and "skip x_2 cycle".

Since the graph would be symmetrical, the vertices can be re-labelled such that it is mirrored according to their distance away from vertex "0", as shown in Fig 3.

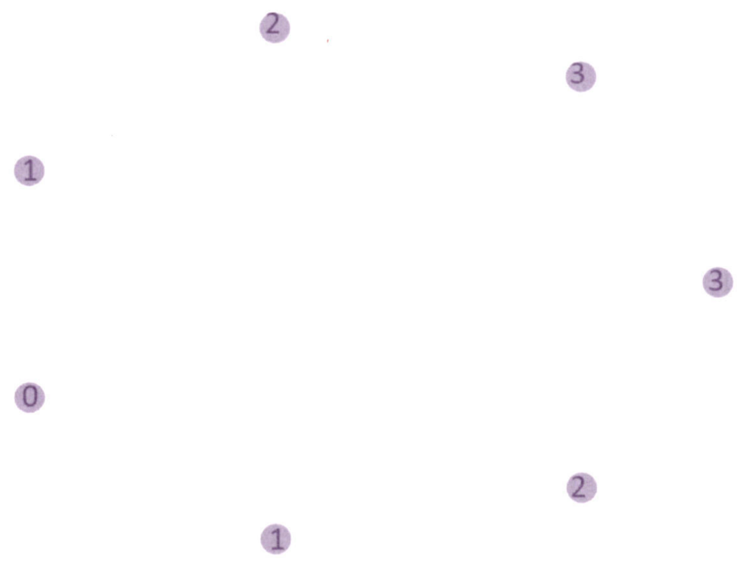

Fig 3: Re-labelling

By trial and error, we observe that if all the numbers from 1 to 3 can be expressed in one of the following forms,

- $x_1 + 1$
- $x_2 + 1$
- a sum or difference of $x_1 + 1$ and $x_2 + 1$

then the condition of $d^*(u, v) \leq 2$ will be fulfilled, thus the graph would be in the K family.

Since

$$1 = 1,$$
$$2 = 2,$$
$$3 = 1 + 2,$$

a graph with 7 vertices formed using "skip 0 cycle" and "skip 1 cycle" would be in the K family.

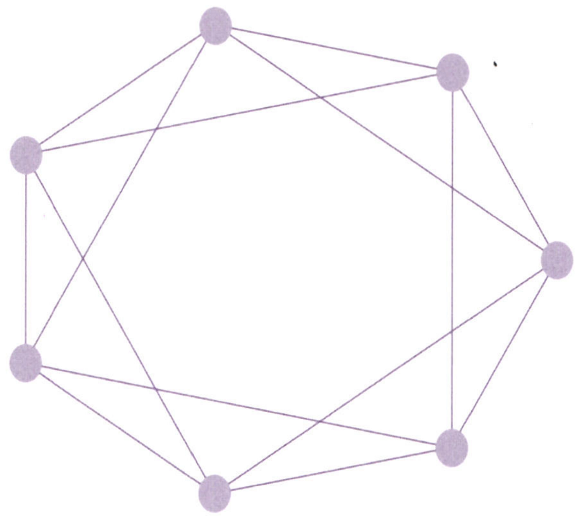

Fig 4: A Graph with 7 vertices in K

To generalise for n vertices, we would have to ensure that all the numbers from 1 to $\left\lfloor \frac{n}{2} \right\rfloor$ can be expressed in one of three forms mentioned above for two chosen integers x_1 and x_2. Because of the symmetrical property, we only have to make sure that every vertex can reach the vertex furthest away (i.e. at the halfway point).

Using this method, we have verified that graphs with 10, 11, 12, 13 vertices can be formed using "skip 1 cycle" and "skip 2 cycle".

When $n \geq 14$, it is no longer possible to limit the number of "skip cycles" to 2.
In the case of $n = 14$, all the numbers from 1 to 7 are to expressed in one of three forms for two chosen integers x_1 and x_2. However,

$7 = 1 + 6$, but 1 and 6 cannot be used to express 3 or 4;
$7 = 2 + 5$, but 2 and 5 cannot be used to express 1 or 6;
$7 = 3 + 4$, but 3 and 4 cannot be used to express 2.

As n increases, it is no longer possible to express all the numbers in the one of three forms.

To be able to obtain all the numbers from 1 to $\left\lfloor \frac{n}{2} \right\rfloor$, we have to increase the number of "skip cycles", thus increasing the number of x values.

Using the above logic, we made the following observations.

For graphs in K with 14, 15, 16, 17 vertices can be formed using "skip 1 cycle", "skip 2 cycle" and "skip 3 cycle".
For graphs in K with 18, 19, 20, 21 vertices can be formed using "skip 2 cycle", "skip 3 cycle" and "skip 4 cycle".
For graphs in K with 22, 23, 24, 25 vertices can be formed using "skip 2 cycle", "skip 3 cycle", "skip 4 cycle" and "skip 5 cycle".
For graphs in K with 26, 27, 28, 29 vertices can be formed using "skip 3 cycle", "skip 4 cycle", "skip 5 cycle" and "skip 6 cycle".
For graphs in K with 30, 31, 32, 33 vertices can be formed using "skip 3 cycle", "skip 4 cycle", "skip 5 cycle", "skip 6 cycle" and "skip 7 cycle".

Following the pattern, we verified that graphs in K with 38 vertices can be formed using "skip 4 cycle", "skip 5 cycle", "skip 6 cycle", "skip 7 cycle", "skip 8 cycle" and "skip 9 cycle".

Starting from 14 vertices, with every increase in the number of vertices by eight, a new "skip cycle" has to be added.

Hence, we have the following results.

1. The graphs in K with n vertices can be formed using "skip $\left\lfloor \frac{n-2}{8} \right\rfloor$ cycle", "skip $\left\lfloor \frac{n-2}{8} \right\rfloor + 1$ cycle", …, and "skip $\left\lfloor \frac{n-2}{4} \right\rfloor$ cycle".
2. The number of "skip x cycles" is $\left\lfloor \frac{n+10}{8} \right\rfloor$.

The formula to determine the number of edges each vertex is connected to can then be easily deduced, noting that every cycle connects a vertex to two edges.

Number of edges each vertex is connected to $= 2\left\lfloor \frac{n+10}{8} \right\rfloor$. This implies that each team will play with $2\left\lfloor \frac{n+10}{8} \right\rfloor$ other teams.

5 Issues in Relationships

We go on to investigate the possible outcomes of the tournament format using a graph in K.

If a vertex v_i is directed related to another vertex v_j (i.e. either $v_i \rightarrow v_j$ or $v_j \rightarrow v_i$), then this relationship will be the conclusive and takes priority over all indirect relationships.

However, for vertices indirectly related, the relationship might not always be conclusive.

5.1 Directed Cycles

Given any directed cycle, it will not be possible to determine the relationship of non-adjacent vertices (which have direct relationships). For example, in a directed C_4 as shown in Fig 5, we are unable to determine the relationship between v_1 and v_3 because $v_1 \rightarrow v_2 \rightarrow v_3$ but $v_3 \rightarrow v_4 \rightarrow v_1$.

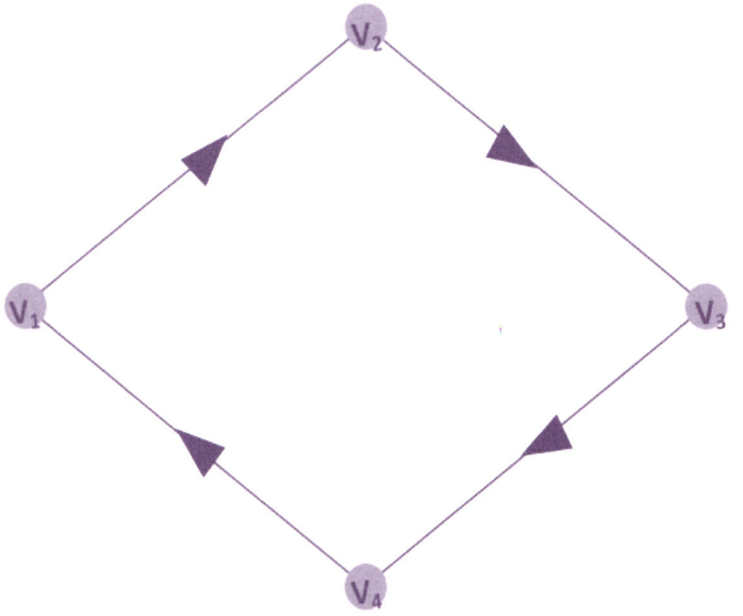

Fig 5: directed C_4

5.2 Inconclusive Relationship

When two disjoint vertices win or lose to a mutual vertex, which connects the two vertices, the relationship between the two vertices would fail to be established. (Examples in Fig 6 and 7)

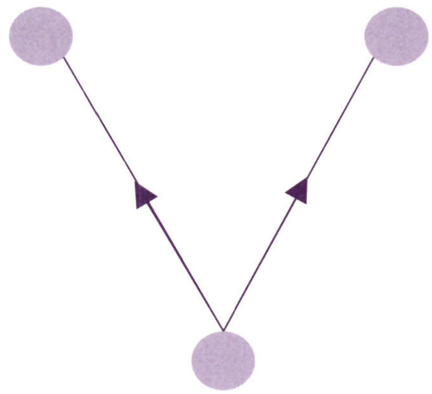

Fig 6: 2 vertices losing to the same vertex

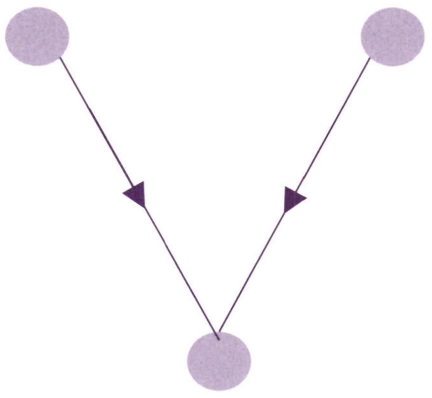

Fig 7: 2 vertices winning the same vertex

We consider them as tie, because from the data gathered, we can only assume that they are of the same ability unless another relationship is established through another path.

5.3 Contradictory results

The number of paths between any two vertices varies from graph to graph, based on how many possible ways to express the vertex label using the x values.

For example, in a graph with 7 vertices, there are three different ways to get to the vertex "3" from vertex "0", i.e. vertex "0" to "1" to "3", vertex "0" to "2" to "3", vertex "0" to "5" to "3". Suppose the paths are $v_0 \rightarrow v_1 \rightarrow v_3$, $v_0 \leftarrow v_1 \leftarrow v_3$ and $v_0 \rightarrow v_5 \rightarrow v_3$, then we will take the majority and conclude the relationship to be $v_0 \rightarrow v_3$ indirectly.

However, there may be two issues. First, the paths are inconclusive as shown in **5.2**, i.e. $v_0 \rightarrow v_1 \leftarrow v_3$. Second, in the event of even number of paths, there may be a tie in the number of wins and losses. In such cases, we would not be able to establish a conclusive relationship.

6 Solution

6.1 Tabulation of Scores

To resolve the issues in **Section 5**, we create a table to express all the relationships between teams using the format in Fig 8. The relationships can be represented by the scores 1, 0 and −1. A win is denoted by 1, a tie or an inconclusive relationship is denoted by 0, and a loss is denoted by −1.

As mentioned in **Section 5**, a direct relationship will take priority over all indirect relationships. Thus, if there exists a direct relationship, then only that relationship needs to be considered in order to derive the score.

The issue of **5.1** will be considered as a score of 0, as there are two paths of opposing results.

The issue of **5.2** will also be considered as a score of 0, as it is an inconclusive relationship.

Each path in the issue of **5.3** will represented by the scores 1, 0 and −1. The final score to be recorded in the table will be taken as the sum of all the scores of the paths.

Tabulation Of Scores

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

Fig 8: Tabulation Format for a 6 team playoff

Since we collate the scores horizontally, the subject is the team in the left column (x) and the object is the team in the top row (y). A cell positioned (x, y) can be read as *x wins/loses to/ties with y*.

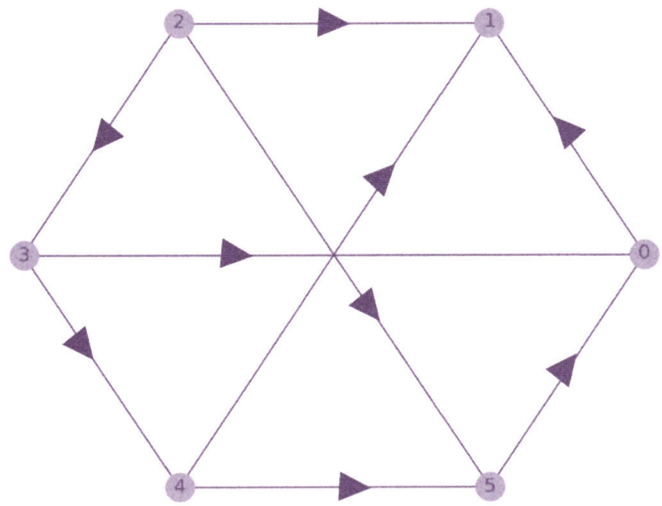

Fig 9: A directed graph in K with 6 vertices

Tabulation Of Scores

	1	2	3	4	5	6	SUM
1		1	1	1	1	1	5
2	-1		1	1	1	1	3
3	-1	-1		1	1	1	1
4	-1	-1	-1		1	1	-1
5	-1	-1	-1	-1		1	-3
6	-1	-1	-1	-1	-1		-5

Fig 10: Tabulation of Scores for Fig 9 graph

For example, the graph in Fig 9 can be translated to the tabulated form in Fig 10. Fig 10 shows all the relationships present including the indirect relationships.

Hence the best team is defined as the team with the highest sum in the tabulation of scores.

6.2 Tie in Sums

In the event where there is a tie in the sums, we will resolve the tie by

- first observing if there exists a direct relationship between the teams involved.
- in the event where the direct relationship does not resolve the tie, then the teams involved will compete in another round of tournament.

7 Conclusion

We have derived a configuration of a tournament format which successfully shortens the time the game spans over, by avoiding playing round-robin and ensures fairness by having all teams play the same number of games. This format caters to any number of teams and adheres to time constraints.

We have highlighted the issues in identifying the relationships between vertices in this format and have finalised a calculation method to tabulate scores and overcome the issues. With this new tournament format and the score tabulation method, winner and loser relationships between teams can be accurately established and faults in current tournament formats and scoring systems are eliminated.

8 Future Work

In future, we aim to find the total number of different paths that connect any two vertices, to easily identify all direct and indirect relationships and eliminate all the inconclusive results. We also aim to find more configurations of graphs in K, which may open up other ways to determining the relationship between vertices. A stimulation program can also be used to exhaust all possible results to see if there are any anomalies and find ways to overcome them.

9 References

[1] Yu, Y., Di, J., & Li, M. (September 2006). Kings in Tournaments. Mathematical Medley, 33(1), 28-32. Retrieved November 17, 2017, from
https://sms.math.nus.edu.sg/smsmedley/Vol-33-1/Kings%20in%20Tournaments%20(Yu%20Yibo,%20Di%20Junwei,%20Lin%20Min).pdf.