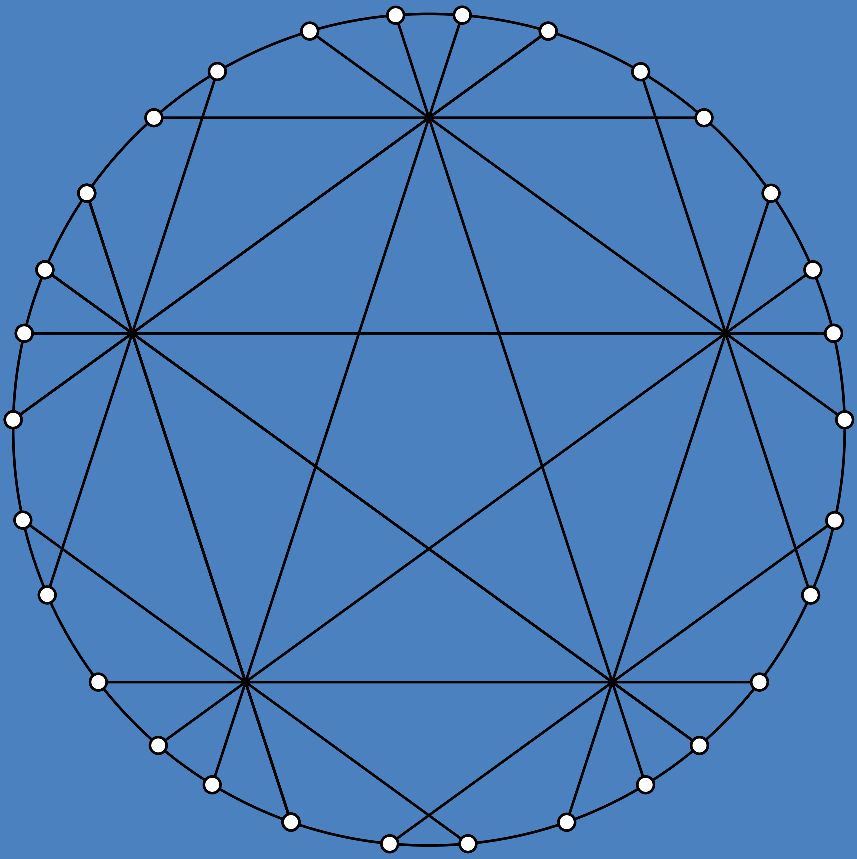
BULLETIN of the INSTITUTE of COMBINATORICS and its APPLICATIONS

Volume 80
May 2017



Boca Raton, Florida

ISSN: 1183-1278

BULLETIN OF THE ICA
Volume **80** (2017), Pages 128–136

Classroom Note. Scheduling an Exciting Set of Rugby Matches

Joy Lind

Department of Mathematics, University of Sioux Falls
joy.lind@usiouxfalls.edu

Abstract

In this paper, we use matchings in graphs to establish a set of "exciting" games between pairs of rugby teams. We then present a series of follow-up exercises appropriate for students in an undergraduate discrete mathematics or graph theory course.

1 Introduction

New Zealand is home to many rugby teams at various levels, and fans attend their games expecting to be treated to a high dose of excitement. Assume that some (even) number of teams will be competing in an upcoming rugby tournament, with winners at each stage progressing to subsequent rounds until an overall winner is determined. Here, we focus on the question of how to establish the set of games comprising the first round of the tournament, where each such game is played by one pair of teams from the original set. The pairings could be made in a variety of ways (e.g. random assignment; pairing the team with the best record, with the team with the worst record, second best with second worst, etc.) In this paper, we apply the following method: we associate with each pair of teams an "excitement factor," a number that quantifies how exciting a game between the two teams is expected to be. (This factor might be derived from historical ticket sales, for example.) We then seek to establish pairings of the teams so as to maximize the lowest excitement factor among the games that comprise the first stage of the tournament. (Note that an academic sabbatical the author spent at

Accepted: 20 April 2017

the University of Auckland in New Zealand motivated the choice of rugby as the focus of this article, but the content is relevant to just about any team sport.)

2 Using Graphs to Model the Rugby Problem

The rugby tournament problem can be modeled as a graph, where each team is represented by a vertex, and an (undirected) edge exists between two vertices if the corresponding teams are eligible to play together in the first round of the tournament. If there are no restrictions on eligibility, then the graph is *complete*, in that every pair of vertices is connected by an edge. However, there could be reasons why certain teams cannot play each other in the initial round; for example, if the games in the first round are not played in a central location but instead are geographically dispersed, then teams that are located too far apart cannot be paired. Thus, it could be that some edges are missing from the graph. However, we assume that there is guaranteed to be at least one way of pairing the teams so that each team is included in some pair; that is, any limitations on how teams can be paired are not so restrictive that they would prevent a team from being able to compete in the first round of the tournament. Each edge is assigned a weight, namely an "excitement factor" for the corresponding pair of teams, as explained previously.

The rugby tournament problem can now be expressed in terms of graphs as a *matching problem.* A *matching* on a graph is a collection of edges in the graph such that no two edges share the same vertex. The *cardinality* of a matching is the number of edges in the matching. A matching is *perfect* if every vertex in the graph is on some edge in the matching. Figure 1 shows two different perfect matchings (of cardinality 3) on a particular graph.

There is abundant literature on the topic of maximum cardinality matchings in graphs. In particular, several polynomial-time algorithms exist for finding maximum cardinality matchings. For example, Edmonds' Blossom algorithm can be used for general graphs [5] and has a complexity of $O(n^2m)$ [6] (where n is the number of vertices and m is the number of edges in the graph). The simpler Hungarian algorithm can be used for bipartite graphs and has a complexity of $O(n^3)$ [2]. Note that our previous assumption that there must be at least one way of pairing the teams so that each team is included in some pair, guarantees us a perfect matching. Our objective

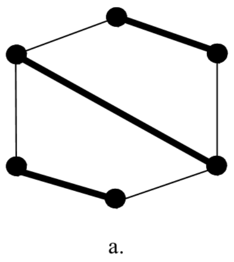


Figure 1: Perfect Matchings on a Graph

then becomes the following: how do we find a perfect matching that maximizes the lowest excitement factor among the games that comprise the first stage of the tournament? To answer this, we present a modified version of a method described in [3].

3 Finding a Perfect Matching

We first address the problem of finding a perfect matching. Here, we apply a variation of the Blossom algorithm described in [3] and [4] for finding a maximum cardinality matching on a graph. Since our assumptions guarantee that a perfect matching exists, finding a maximum cardinality matching is equivalent to finding a perfect matching. The method depends on the notions of *paths*, *alternating paths*, and *augmenting paths* in graphs. A *path* is a sequence of edges between two vertices. An *alternating path* with respect to a matching M is a path whose edges are alternately in M and not in M. An *augmenting path* is an alternating path connecting two unmatched vertices. In Figure 2, there is an augmenting path between unmatched vertices A and B, namely, A - F - C - B.

Note that if an augmenting path exists for a matching M and is such that its endpoints are distinct, then we can find a larger cardinality matching by interchanging the matched and unmatched edges along it; performing these exchanges produces a new matching whose cardinality is one greater than the original matching. Applying this method to the matching in Figure 2, we produce the perfect matching in Figure 1b. In [1], Berge shows that a matching has maximum cardinality if and only if it has no augmenting path. It therefore follows that we can find a maximum cardinality matching by finding any initial matching, then searching for augmenting paths among

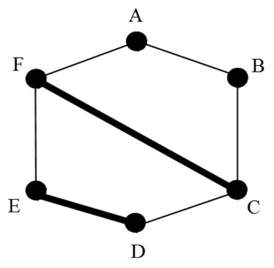


Figure 2: Matching with an Augmenting Path

unmatched vertices. If no augmenting path is found, then the current matching has largest cardinality. Since we are assuming a perfect matching exists, we would need to continue searching for augmenting paths until all vertices have been matched. While this may seem straightforward, there is a complication around graphs with *odd cycles*, where an odd cycle is a path with an odd number of edges that starts and ends at the same vertex. Such cycles may constitute alternating paths with respect to a matching but do not lend themselves to augmentation as described above, since the interchanging of matched and unmatched edges would result in two matched edges meeting at a single vertex. In [3] and [4], the authors explain in depth how to account for odd cycles in graphs. For purposes of simplification, we will assume that our graphs do not have odd cycles (which would be true for bipartite graphs, for example).

4 Finding an Optimal Perfect Matching

In the previous section, we focused on finding a maximum cardinality matching. Such matchings generally are not unique, and so now we turn our attention to maximum cardinality matchings that are "best" in some sense. In our rugby tournament problem, we characterize the quality of a matching by the level of anticipated excitement around the games corresponding to the edges in the matching. We develop that idea in this section.

Assume that in our graph model for the rugby tournament problem, a non-negative integer-valued "excitement factor" has been determined for each team pair and that those factors have been assigned as weights to the graph's edges. There are various methods for comparing matchings relative to these excitement factors as edge weights. One possibility is to measure the overall excitement of (round one of) a tournament by summing the excitement factors along the edges that comprise the tournament an seeking to maximize that sum. Here, we instead choose to find a tournament that maximizes the lowest excitement factor among the games that comprise the tournament; that is, we seek to maximize the minimum edge weight across the edges in a matching, subject to the matching being perfect. Note that different ways of measuring excitement will in general lead to different outcomes. For instance, in a graph consisting of a single four-cycle with excitement factors of 1, 2, 5, 3 (listed cyclically on the graph's edges), the edges with factors of 1 and 5 would be selected if we seek to maximize the sum of excitement factors, whereas the edges with factors of 2 and 3 would be selected if we seek to maximize the minimum excitement factor.

As we seek to maximize the minimum edge weight across the edges in a matching, we note that various methods are described in [3] for solving problems similar to ours. We present a modified version of one of those methods here. In a graph G, let M be any maximum cardinality matching, and let n be the number of matched edges in M. Let $cmin$ be the smallest edge weight (excitement factor) among edges in M. Let $climit = cmin + 1$. We now consider the subgraph of G obtained by removing all edges with weights less than $climit$. For this graph, we find a maximum cardinality matching. If the number of edges found is n, then this new matching is an improvement over M. In this case, the new matching is called M, and the procedure is repeated. If, however, the number of edges in the new matching is less than n, then the previous matching was optimal. It can easily be shown that this method in fact produces an optimal perfect matching:

Proof of optimality: If in a graph G, M is a matching found by the above procedure, let C_M be the lowest edge weight that appears on any of its edges. Assume there is another perfect matching, P, on G whose minimum edge weight is C_P with $C_P > C_M$. Then P does not include any edges with weights less than C_P. Since $C_P > C_M$, this means that P is a perfect matching on G that excludes all edges with weights less than or equal to C_M. However, in that case, our procedure would have found P as an improved perfect matching over M.

An example is presented in the next section.

5 Student Exercises

In this section, we lead students through an example of the above process for optimally pairing rugby teams in an initial round of a tournament. Assume there are ten teams, labeled A through J, that require pairing. Due to eligibility restrictions, some teams are not permitted to play other teams in the initial round. The below table gives the excitement factor (on a scale from 1 to 10) for each pair of teams that are eligible to play each other in the first round. A blank entry indicates that the two teams are not eligible to play each other.

Table 1: Excitement Factors

	A	B	C	D	E	F	G	H	I	J
A		6		9		7		7		7
B	6		10		7		5		5	
C		10		8		8		9		9
D	9		8		7		5		3	
E		7		7		4		6		6
F	7		8		4		5		6	
G		5		5		5		7		6
H	7		9		6		7		7	
I		5		3		6		7		4
J	7		9		6		6		4	

Exercise 1: Draw the graph that models the rugby tournament problem for this set of teams.

Solution 1: Note that the graph is bipartite. To facilitate viewability, edge weights (labels) are not included.

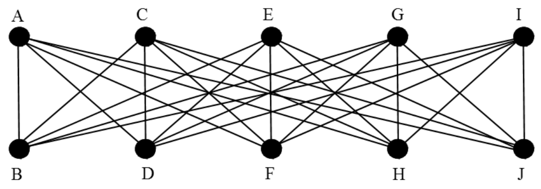


Figure 3: Rugby Graph

Exercise 2: Starting with the matching below (consisting of edges AB, CF, and GH), use the technique of augmenting paths explained in Section 3 to find a perfect matching.

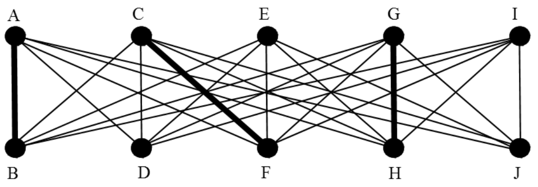


Figure 4: Initial Matching on Rugby Graph

Solution 2: Answers will vary. One possibility is to start with augmenting path D - A - B - C - F - G - H - I that includes three matched and four unmatched edges, then interchange the matched and unmatched edges to arrive at the larger cardinality matching as shown in Figure 5. We can

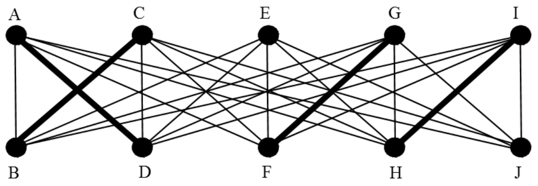


Figure 5: Larger Cardinality Matching on Rugby Graph

then use augmenting path E - F - G - H - I - J to increase the size of our matching by interchanging its matched and unmatched edges. Since the resulting matching covers all vertices, we have found a perfect matching, as shown in Figure 6. Call this matching M_1. We note that the small

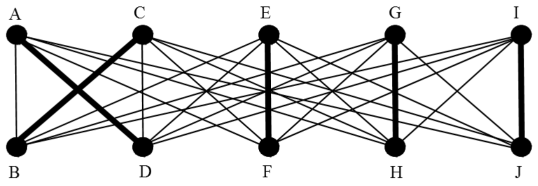


Figure 6: M_1, A Perfect Matching on Rugby Graph

size of this graph enables us to "visually" detect the augmenting path from which we constructed a perfect matching. When the number of vertices is large, we would need to rely on algorithms such as those referenced earlier (Edmonds' Blossom, Hungarian) to find perfect matchings.

Exercise 3: Starting with the final matching in Exercise 2 (M_1 in Figure 6), find a perfect matching that maximizes the minimum excitement factor. Then, use the resulting matching to identify a most exciting set of rugby games to comprise the initial round of the tournament.

Solution 3: Applying the excitement factors from Table 1, we find a minimum excitement factor of 4 within M_1 (edges EF and IJ have factors of 4). We therefore exclude from our original graph all those edges with an excitement factor less than 5 (namely, edges EF, IJ, and DI), and again seek a perfect matching. One possibility is shown below; call this M_2.

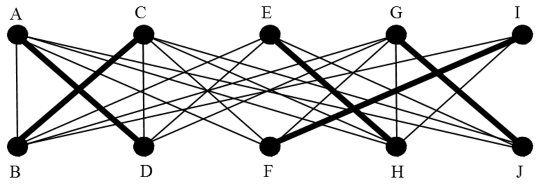


Figure 7: M_2, A More Exciting Perfect Matching on Rugby Graph

In matching M_2, 6 is the minimum excitement factor, so M_2 is an improved matching over M_1. We next exclude from our graph all those edges with an excitement factor less than 7 (namely, edges AB, BG, BI, DG, DI, EF, EH, EJ, FG, FI, GJ, and IJ), and again seek a largest cardinality matching. We find the matching in Figure 8, call it M_3, which has no augmenting path and hence is largest in its respective graph. Since the cardinality of M_3 is less than that of M_2, we conclude that M_2 is optimal. The initial round of the rugby tournament should therefore pair the following teams: A and D; B and C; E and H; G and J; and F and I.

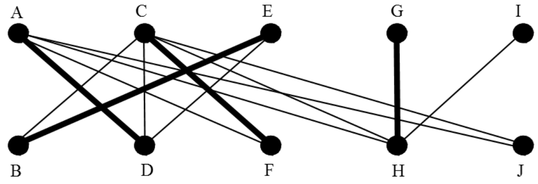


Figure 8: M_3, Smaller Matching on Rugby Graph

References

[1] C. Berge, “Two Theorems in Graph Theory,” Proc. Natl. Acad. Sci. U.S., 43 (1957) pp. 842, 844.

[2] M. Kao, T. Lam, W. Sung, and H. Ting, “A Decomposition Theorem for Maximum Weight Bipartite Matchings,” SIAM Journal on Computing, Vol. 31 Issue 1 (2001), pp. 18-27.

[3] A. Mason and A. Philpott, “Speaker Matching,” A Third Professional Project in Engineering Science (1987).

[4] A. Mason and A. Philpott, “Speaker Pairing using Matching Algorithms,” Asia Pacific Journal of Operations Research, 5 (1988), pp. 101-116.

[5] http://mathworld.wolfram.com/BlossomAlgorithm.html

[6] https://brilliant.org/wiki/blossom-algorithm/♯complexity