Flows in Graphs	Instructor: Padraic Bartlett
Lecture 3: Scheduling Tournaments with Flows	
Week 2	*Mathcamp 2011*

Today's lecture is focused on a more complicated problem in combinatorial designs, which (unlike the results we discussed yesterday) has yet to reveal a solution that doesn't relate back to flows:

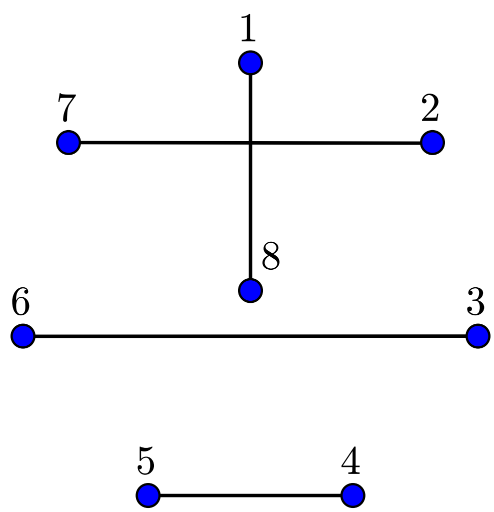
1 Tournaments and 1-Factorizations

Consider the following question:

Question 1 *Suppose that we've decided to create a Rock-Scissors[1] tournament for eight people. Specifically, in each round, we want to divide them up into four pairs and play each other. How many rounds do we need in order to have each player face each other player exactly once?*

Solution. Trivially, we will need at least seven rounds; this is because each team is playing exactly one other team in each round, and there are eight total teams.

As it turns out, this bound is achievable! To see why, consider the following picture:



This is a schedule for the first round; to acquire schedules for later rounds, simply take this picture and rotate it by appropriate multiples of $2\pi/7$.

This observation can in fact be generalized: if we want to create a tournament for $2n$ teams that requires $2n-1$ rounds for everyone to play each other, we can do this by:

- letting G be an abelian group of order $2n-1$,

[1] Rock-Scissors is played like Paper-Rock-Scissors, except you cannot play paper.

- for each $g \in G$, letting $M_g = \{\{g, \infty\} \cup \{a, b\} : a + b = 2g, a \neq b\}$, and
- identifying our teams with the set $G \cup \{\infty\}$, and each of our $2n - 1$ rounds with the $2n - 1$ distinct M_g's.

(If you're unconvinced, prove this on the HW!)

This is all well and good for 2-player games. But what if we want to consider more complicated games, like (say) Looping Louie? Or Telephone Stomp? Or n-player chess[2]?

In other words: suppose we have a game that requires k teams to play, and we want to make a tournament with n teams. How many rounds do we need to insure that all possible $\binom{n}{k}$ combinations of teams are played?

Well: for the moment, assume that k divides n, for convenience's sake. In the best case, then, we'll need to have $\binom{n}{k} \cdot \frac{k}{n} = \binom{n-1}{k-1}$. Can we achieve this for every such k, n where $k|n$?

For $k = 2$, this was not too hard; for $k = 3, 4$, it's **far** harder, and for any value of $k \geq 5$ there are no similarly constructive proofs known. Thus, the following result of Baranyai, that settles the case for all k via the language of flows, was **really** surprising when it first came out! – with far less effort than any case other than $k = 1, 2$, we can settle all of these problems at once!

Theorem 2 *(Baranyai) If k divides n, then the set of all $\binom{n}{k}$ k-subsets of the set $\{1, \ldots n\}$ can be divided into disjoint parallel classes*[3]

Proof. Let n, k be given, let $m = n/k$, and $M = \binom{n-1}{k-1}$. First, make the following definition:

Definition. A m**-partition** of a set X is a collection A of m pairwise disjoint subsets of X, some of which can be empty, so that their union is X.

We will prove a (seemingly) stronger version of our claim: for any integer $0 \leq l \leq n$, there is a collection

$$A_1, \ldots A_M$$

of m-partitions of $\{1, \ldots l\}$, with the property that each subset $S \subseteq \{1, \ldots l\}$ occurs in precisely $\binom{n-l}{k-|S|}$ many of the partitions A_i.

We proceed by induction on l. Notice that for $l = 0$, this is trivially true by letting all of the A_i's consist of m copies of the empty set, so we've established our base case. Also, notice that for $l = n$, we will have proven our claim, as we'll have made M partitions of $\{1, \ldots n\}$ with the following property: *each subset S of $\{1 \ldots n\}$ shows up in $\binom{0}{k-|S|}$-many collections*! In other words, the only subsets that show up are those with $|S| = n$ (because that's the only value for which $\binom{0}{k-|S|} \neq 0$,) and those all show up precisely once!

Assume that we've demonstrated our claim for some value of $l < n$. Create a directed graph G as follows:

- Let s, t be source and sink vertices, for each A_i, create a vertex A_i, and for each subset $S \subset \{1, \ldots l\}$, create a vertex S.

[2] n-player chess is played like normal chess, except with n players arranged in a cycle, alternating play on the white and black sides. It is recommended to play with an odd number of players, for maximal complexity.

[3] A **disjoint parallel class** A of subsets of $\binom{n}{k}$ is a collection of n/k k-subsets of $\{1 \ldots n\}$ that partition this set.

- Create a directed edge (s, A_i) for every i, and a directed edge (S, t) for every S; as well, create edges from each A_i to each of the subsets that show up in it.
- Define a capacity function on this graph by setting $c(s, A_i) = 1$, $c(A_i, S) = +\infty$, and $c(S, t) = \binom{n-l-1}{k-|S|-1}$.

Consider the following flow f:

- $f(s, A_i) = 1$, for all A_i.
- $f(A_i, S) = \frac{k-|S|}{n-l}$, and
- $f(S, t) = \binom{n-l-1}{k-|S|-1}$.

We claim this is a flow. To see this: note that the sum of elements leaving A_i is

$$\begin{aligned}
\sum_{S \in A_i} \frac{k - |S|}{n - l} &= \frac{1}{n-l} \cdot \sum_{S \in A_i} (k - |S|) \\
&= \frac{1}{n-l} \cdot \left(mk - \sum_{S \in A_i} |S| \right) \\
&= \frac{1}{n-l} \cdot (mk - l) \\
&= \frac{1}{n-l} \cdot (n - l) \\
&= 1
\end{aligned}$$

As well, the sum of values entering a vertex S is

$$\begin{aligned}
\sum_{i : S \in A_i} \frac{k - |S|}{n - l} &= \frac{k - |S|}{n - l} \cdot \binom{n-l}{k-|S|} \\
&= \binom{n-l-1}{k-|S|-1}.
\end{aligned}$$

So it's a flow! Furthermore, it's a maximal flow with value M, because it saturates all of the edges leaving s. So, by max-flow min-cut, because all of the capacity values are integral, there is an integral flow! Let f' be such an integral flow. Because f is 1 on all of the (s, A_i) edges, it assigns each A_i one of its subsets S_i. For each S_i, furthermore, we have that each S is picked out by $\binom{n-l-1}{k-|S|-1}$-many different A_i's.

Finally, turn this collection of $\{A_1, \ldots A_n\}$'s into a collection $\{A'_1, \ldots A'_n\}$, by taking A_i and replacing its assigned subset S_i with $S_i \cup \{l + 1\}$.

How many times does each subset $S \subseteq \{1, 2, \ldots l + 1\}$ show up in our new collection $\{A'_1, \ldots A'_n\}$? First, consider any subset of the form with $S \cup \{l + 1\}$, where $S \subset \{1, \ldots l\}$. By construction, we know that this set shows up in each A'_i-subset that had S as its assigned chosen subset. There were precisely $\binom{n-l-1}{k-|S|-1} = \binom{n-(l+1)}{k-(|S|+1)}$-many such subsets; so our claim holds for all of these kinds of sets!

Now, examine the subsets that don't contain $l+1$: i.e. $S \subset \{1, \ldots l\}$. There were originally $\binom{n-l}{k-|S|}$-many of these subsets, and we took away $\binom{n-l-1}{k-|S|-1}$ of these to form new $l+1$ subsets; this leaves

$$\begin{aligned}
\binom{n-l}{k-|S|} - \binom{n-l-1}{k-|S|-1} &= \frac{n-l}{k-|S|} \cdot \binom{n-l-1}{k-|S|-1} - \binom{n-l-1}{k-|S|-1} \\
&= \frac{n-l-k+|S|}{k-|S|} \cdot \binom{n-l-1}{k-|S|-1} \\
&= \frac{n-l-1}{k-|S|},
\end{aligned}$$

which is again what we claimed. So we have proven our result by induction: setting $l = n$ then gives us our desired collection of disjoint parallel classes.