Scheduling a Tournament

Dalibor Froncek
University of Minnesota Duluth

February 9, 2010

Abstract
We present several constructions for scheduling round robin tournaments. We explain how to schedule a tournament with alternate home and away games and point out some other interesting properties of the schedules. Finally, we show what it means that two schedules are "different" and mention the number of different schedules for the cases where the number is known.

1. Some small tournaments

Suppose we have four teams named $1, 2, 3, 4$ and we want to schedule a three-day round robin tournament with each team playing one game on each day. All we need to do is to choose an opponent for team 1 on the first day and another opponent for team 1 on the second day. All other games are determined by these choices.

Say we choose the game $1-2$ for Friday and $1-3$ for Saturday. Of course, the remaining teams must meet on both days—teams 3 and 4 on Friday and teams 2 and 4 on Saturday. Then we have just one choice for Sunday, namely games $1 - 4$ and $2 - 3$.

	Friday	**Saturday**	**Sunday**
Game 1	$1-2$	$1-3$	$1-4$
Game 2	$3-4$	$2-4$	$2-3$

Table 1: Four team tournament

Now let us try to schedule a five-day round robin tournament for six teams. More simply, we may just say "round robin tournament of six teams" as we always assume that each team plays one game each day. Because the number of opponents for each team is always one less than the total number of teams, the number of days (or rounds) is at least one less than the number of teams—we actually need to show that it is possible to find a five-day schedule. We can start as in the previous example and schedule for Round 1 games $1-2, 3-4$, and adding the game of the two new teams $5-6$. Then for Round 2 we schedule games $2-3, 4-5, 6-1$ and for Round 3 games $1-4, 2-5$, and $3-6$.

For Round 4 we have two choices for an opponent of team 1, either 3 or 5. Say we choose the game $1-3$. But then team 5 cannot play a game, since the only opponents they did not play are 1 and 3, which are scheduled to play each other! Similarly, if we choose the game $1-5$, we have the same problem with team 3—they could only play either 1 or 5, but these teams are already scheduled to play another game.

	Round 1	**Round 2**	**Round 3**	**Round 4**	**Round 5**
Game 1	$1-2$	$2-3$	$1-4$	$1-3$	
Game 2	$3-4$	$4-5$	$2-5$	$5-?$	
Game 3	$5-6$	$6-1$	$3-6$		

Table 2: Incomplete six team tournament

With some effort, using the method of trial and error, or, more precisely, an exhaustive (and exhausting!) search, we would find a schedule. An example is in Table 3.

	Round 1	**Round 2**	**Round 3**	**Round 4**	**Round 5**
Game 1	$1-6$	$2-6$	$3-6$	$4-6$	$5-6$
Game 2	$2-5$	$3-1$	$4-2$	$5-3$	$1-4$
Game 3	$3-4$	$4-5$	$5-1$	$1-2$	$2-3$

Table 3: Six team tournament

However, scheduling a round robin tournament of eight teams using an exhaustive search could take several hours. Just try to count how many possibilities we have. The first round can be selected at random. Say we select games $1-2, 3-4, 5-6$, and $7-8$.

	Round 1	**Round 2**	**Round 3**	**......**	**......**
Game 1	$1-2$	$1-\{\mathbf{3},4,\dots,8\}$			
Game 2	$3-4$	$2-\{\mathbf{4},5,\dots,8\}$			
Game 3	$5-6$	$6-\{7,8\}$			
Game 4	$7-8$	forced			

Table 4: Games $1-3$ and $2-4$ selected in Round 2

	Round 1	**Round 2**	**Round 3**	**......**	**......**
Game 1	$1-2$	$1-\{\mathbf{3},4,\dots,8\}$			
Game 2	$3-4$	$2-\{4,\mathbf{5},\dots,8\}$			
Game 3	$5-6$	$6-\{4,7,8\}$			
Game 4	$7-8$	forced			

Table 5: Games $1-3$ and $2-5$ selected in Round 2

But then just for Round 2 we have six choices for team 1 (since one out of seven possible opponents was already selected in Round 1). For Game 2 we can choose team 2, which can play one out of five opponents-if we have for instance scheduled Game 1 as $1-3$, then 2 can play any of $4, 5, 6, 7$, or 8. If we choose $2-4$, then 6 can select from two opponents, 7 and 8, and the last game is then left for the two remaining teams. This gives $6 \cdot 1 \cdot 2 = 12$ choices. If we choose $2-5$, say, then 6 can be matched with one of $4, 7$, and 8. The same would apply if we choose $2-7$ or $2-8$. This gives $6 \cdot 3 \cdot 3 = 54$ choices. Therefore, we have 66 choices just for Round 2. For each choice, there are many different choices for Rounds 3 to 6, and only Round 7 is fully determined by the previous rounds.

It is obvious that we need a better method than trial and error. To find one, we turn to the branch of mathematics called graph theory. This relatively new field (dating back to 1930s) is often used for modeling many types of applications, such as communication networks, traffic flows, task assignments, timetable scheduling, etc. Graph theory has nothing in common with graphs of functions. For us, a graph consists of a set of points, called *vertices* (singular form *vertex*), and a set of lines, called *edges*. Each edge joins two vertices. Some examples are given in Figure 1.

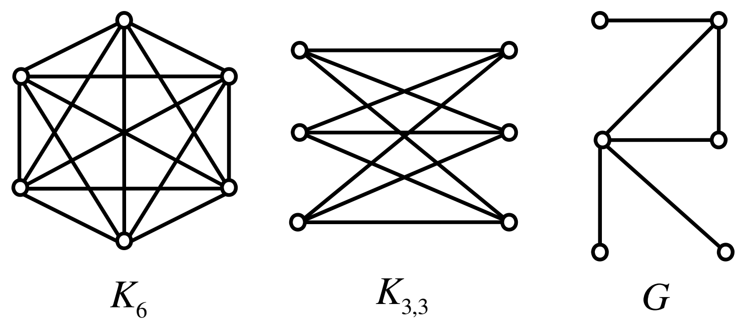

Figure 1: Complete graph K_6, complete bipartite graph $K_{3,3}$, graph G

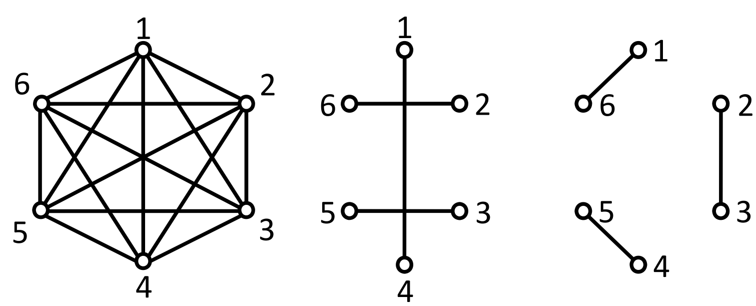

Figure 2: Complete graph K_6 and two different one-factors

A graph in which every vertex is joined by an edge to every other vertex is called a *complete graph*. A complete graph can be viewed as a model of a round robin tournament in a natural way. Each vertex represents one team, and an edge joining two vertices, i and j, represents a game between the corresponding teams, $i - j$. If we assume that the number of teams is even, say $2n$ where n is a natural number, then a round consists of a collection of n edges such that no two edges share a vertex. Such a collection is called a *one-factor* of the complete graph. An example for six teams is shown in Figure 2. (If two edges in the same collection share a vertex j, then team j would be scheduled to play two games in the same round, which we want to avoid.)

Now we are ready to introduce a method that is widely used in tournament scheduling. It was originally discovered in 1846 by Reverend T. P. Kirkman [6], although he did not use it for tournament scheduling, and it is probably the most popular method for tournament scheduling. When it was first used for this purpose is not known. We call a tournament using it a *Kirkman tournament*.

Construction 1 We label the vertices of the complete graph by the team numbers. Place integers $1 - 7$ in order on a circle at a uniform distance and put the vertex 8 in the center of the circle. For Round 1 we select the edge joining 8 to 1 and all other edges perpendicular to the edge $8 - 1$. They are $2-7, 3-6$, and $4-5$. To select Round 2, we rotate them clockwise by $2\pi/7$. That is, we select the edge $8 - 2$ and the perpendicular edges $3 - 1, 4 - 7$, and $5 - 6$. In general, in Round k we select the edge $8 - k$ and all edges perpendicular to it.

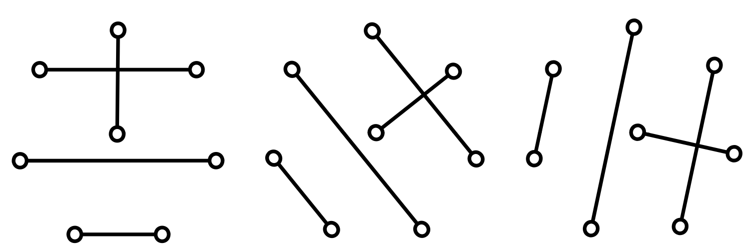

Figure 3: First three rounds in Construction 1

Obviously, each team plays exactly one game in each round. To see that this construction yields a round robin tournament, we only need to convince ourselves that every team plays every other team exactly once. First we count the total number of games. In each of the seven rounds there are four games, which makes a total of 28 games. This is exactly the number of games that we need to schedule—there are eight teams and each of them needs to play seven opponents. This makes 56, but we must divide it by 2 since each game was counted twice (a game $k - j$ was counted once when we counted team ks games and again when we counted team js games). We observed that every team plays a game in each round. Therefore, either every team plays every opponent exactly once, or some team misses one or more teams and hence plays another one at least twice. So we only need to show that each team plays every opponent at most once. Obviously, team 8 plays each opponent exactly once, namely team j in Round j. Now if team k plays another team, i, in round x, then the edge $k - i$ is perpendicular to the edge $8 - x$. Similarly, if team k plays i in (another) round y, then the edge $k - i$ is perpendicular to the edge $8 - y$. Because $k - i$ is perpendicular to both $8 - x$ and $8 - y$, it follows that $8 - x$ and $8 - y$ must be parallel or identical. But no two edges of the form $8 - z$ are parallel in our construction, hence $x = y$ and team k plays i in just one round, namely Round x.

Construction 1 can be easily generalized for any even number $2n$ of teams. This construction can be also described algebraically rather than geometrically. We describe two views of the algebraic construction.

Construction 2 For this construction, we define for each edge its length. Suppose we have an edge $k - j$ and both k and j are less than 8. The length of this edge is the "circular distance" between the vertices k and j. The number of steps we need to take around the circle to get from k to j using the shorter of the two paths between them. Algebraically, we define the length of the edge $k - j$ as the minimum of $|k - j|$ and $7 - |k - j|$. If we suppose, without loss of generality, that $k > j$, then the minimum is the smaller of $k - j$ and $7 - (k - j)$. We also rename team 8 to ∞ (they are pretty similar anyway) and we define the length of any edge $\infty - k$ as ∞. This definition of the edge length is consistent with the definition of edge lengths of the other edges, since $\infty - k = \infty$ for any finite k.

Now we observe that in Round 1 the edge $2 - 7$ has length $7 - (7 - 2) = 2$, edge $3 - 6$ has length $6 - 3 = 3$, and $4 - 5$ has length $5 - 4 = 1$. Hence we have three different lengths, $1, 2$, and 3. The remaining edge $\infty - 1$ is of length ∞ and all four lengths are different. We also observe that no two vertices on the circle can be at distance more than 4, because one of the two paths between them is always shorter than four steps. Next we can construct Round 2 from

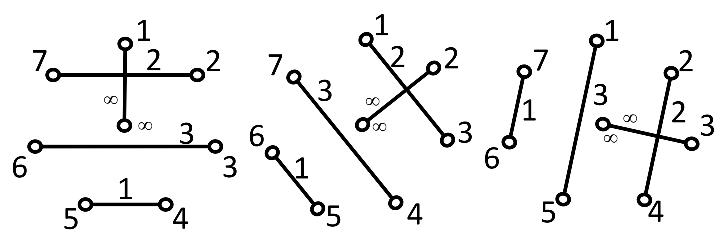

Figure 4: First three rounds in Kirkman tournament

Round 1 by taking each game (i.e., edge) and adding 1 to each team number except team ∞. (In fact, we can add 1 to ∞ as well, assuming naturally that $\infty + 1 = \infty$.) For doing this we need to use "wrap around arithmetic." If the finite number we receive exceeds 7, we subtract 7 to get back within our range. Game $2-7$ then becomes game $3-1$ instead of $3-8$ and the length of the corresponding edge is $3-1=2$, which is the same length as the length of the edge $2-7$. Game (edge) $3-6$ becomes $4-7$ and the length is $7-4=7$, the same as of $3-6$. Game $4-5$ becomes $5-6$ of length 1, same as for $4-5$. Finally, the edge $\infty - 1$ becomes $\infty - 2$, but the length here is defined as ∞. Hence, we again have four different lengths, $1, 2, 3, \infty$. In general, in Round $1+i$ we add i to each vertex (team number). It is easy to observe that an edge $k-j$ of Round 1 and an edge $(k+i)-(j+i)$ of Round $1+i$ have the same length, because $(k+i)-(j+i) = k+i-j-i = k-j$.

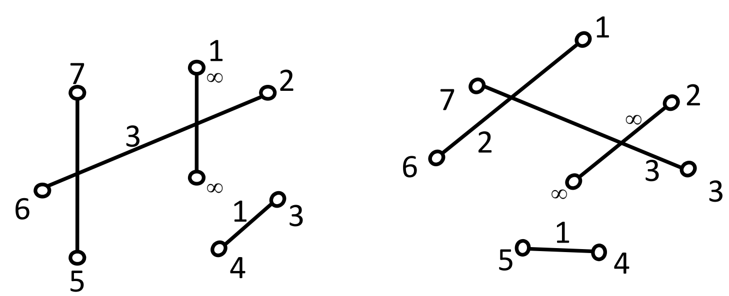

Figure 5: First two rounds in a Steiner tournament

This approach can be used to construct a different schedule. For Round 1 we choose edges $\infty - 1$ of length ∞, $4-3$ of length 1, $7-5$ of length

2, and $2-6$ of length 3. Then for Round i we add $i-1$ to each vertex as in Construction 1. We again have four different lengths and therefore the schedule is covering each game exactly once. A tournament using this schedule is often called a *Steiner tournament*, since it can also be constructed using so-called *Steiner triple system*.

2. Tournaments for any even number of teams

To make sure that this construction gives a good schedule where every team plays every other exactly once, we repeat arguments as in Construction 1. This time we discuss the case for any even number of teams, $2n$. We generalize the edge length used in Construction 2 as follows. For an edge $\infty - i$, the length is defined as ∞. For an edge $k-j$, where $j<k<\infty$, the length is the minimum of $k-j$ and $2n-1-(k-j)$. Obviously, the possible lengths then are $1,2,\ldots,n-1,\infty$.

Theorem 3 *Let K_{2n} be the complete graph on $2n$ vertices $1,2,\ldots,2n-1,\infty$ and F_1 a one-factor of K_{2n} in which every edge length $1,2,\ldots,n-1,\infty$ appears exactly once. For $i=2,3,\ldots,2n-1$ construct a one-factor F_i from F_1 by adding $i-1$ to all vertices $1,2,\ldots,2n-1$. Then the collection $F_1,F_2,\ldots,F_{2n-1}$ contains every edge of K_{2n} precisely once.*

Proof. Each of the $2n-1$ one-factors (rounds) contains n edges, which makes a total of $n(2n-1)$ edges. This is exactly the number of edges in the complete graph with $2n$ vertices. To show that each edge is used exactly once, it is again enough to show that none is used more than once. First we show that in the complete graph with $2n$ vertices there are exactly $2n-1$ edges of each given length l, where l is any of $1,2,\ldots,n-1,\infty$. For each vertex k and each fixed length l (where both $k,l<\infty$) there are two edges of length l incident with k, namely the edges $k-(k+l)$ and $k-(k-l)$, which gives $2(2n-1)$ edges. But every edge was counted twice—the edge $k-(k+l)$ was counted once when we looked at the edges incident with vertex k and again when we counted the edges incident with vertex $(k+l)$. Therefore, there are $2n-1$ edges of length l. For length ∞, there are also precisely $2n-1$ edges, each one joining ∞ to one of the vertices $1,2,\ldots,2n-1$.

We already have shown that the number of edges of any length is the same as the number of one-factors. We still need to show that each length is used in any one-factor exactly once. First we show that each length is used at least once. Suppose that it is not the case and length l is not used in

F_i. By our assumption, there is an edge of length l in F_1, say $(k+l)-k$. Because we obtained F_i by adding $i-1$ to all vertices of F_1, we have the edge $(k+l+i-1)-(k+i-1)$. But the length of this edge also l and hence every length is present. Next we want to show that every length is used at most once. If a length l is used more than once, another length must be missing, which we have just shown cannot happen. Therefore, every length is used in each one-factor exactly once. Because the number of edges of any length is the same as the number of one-factors, it follows that each edge is used precisely once. ∎

A specific version of our theorem producing the Kirkman tournament for $2n$ teams is given here.

Corollary 4 *Let K_{2n} be the complete graph on $2n$ vertices $1, 2, \ldots, 2n-1, \infty$ and F_1 a one-factor of K_{2n} with edges $\infty - 1$ and $k - (2n+1-k)$ for $k = 2, 3, \ldots, n$. For $i = 2, 3, \ldots, 2n-1$ construct a one-factor F_i from F_1 by adding $i-1$ to all vertices $1, 2, \ldots, 2n-1$. Then the collection $F_1, F_2, \ldots, F_{2n-1}$ contains every edge of K_{2n} precisely once.*

Earlier we promised two different views of the algebraic construction. Here is the other one.

Construction 5 We use the same Kirkman tournament schedule as in Construction 2 and observe that in Round 1 the sum of the team numbers (not counting ∞ and its opponent) playing each other is $9 = 7 + 2 = 7 + 2 \cdot 1$, in Round 2 it is either $4 = 2 \cdot 2$ (remember in Round 2 game $2-7$ becomes $3-1$) or $11 = 7+4$, but $4 = 7+2\cdot 2$. In Round 5 we have games $4-6, 7-3$, and $1-2$ which give the sum equal to $10 = 2 \cdot 5$ or $3 = -7 + 2 \cdot 5$. In general, in Round j the sum becomes $7\delta + 2 \cdot j$ where $\delta = -1, 0$, or 1. This is true for all Kirkman tournaments with $2n$ teams. Round j in such a tournament consists of the game $\infty - j$ and the games whose sum is $(2n-1)\delta + 2j$ with $\delta = -1, 0$, or 1.

3. Some more tournament properties

If every team has its own home field, it is desirable to schedule the tournament in such a way that the home and away games for every team alternate as regularly as possible. We say that a team has a *break* in the schedule when it plays two consecutive home or away games. The most balanced schedule is one with no breaks. However, it is impossible to have a schedule in which three or more teams have a schedule with no breaks. Suppose that there are at least three such teams. Then either at least two of them start their schedules with a home game, or at least two of them start with an away game. Without loss of generality we can assume that two start with a home game. But since both of them have no breaks, they always play home at the same time and therefore can never play each other.

A Kirkman tournament has the nice property that its rounds can be reordered so that two teams have no breaks while all other teams have exactly one break each. We can show this using our example for eight teams. Use the convention that in a game $k - j$ the home team is j and schedule Round 1 as $\infty - 1, 7 - 2, 6 - 3, 5 - 4$. Round 2 is then obtained from Round 1 by adding 4 to each team number (rather than 1 as in Construction 2). In general, Round $(i + 1)$ is obtained from Round i by adding 4 to each team number and the games involving team ∞ alternate ∞ as the away and home team. Then teams ∞ and 4 have no breaks, teams $1, 2,$ and 3 have one home break each and teams $5, 6,$ and 7 have one away break each. If we want to be fair and have one break for each team, we can schedule the game between ∞ and 4 in the Round 7 as $4 - \infty$ rather than $\infty - 4$. This also works in general for $2n$ teams, and Round $i + 1$ is then obtained from Round i by adding n to each team number.

This construction has one interesting property. When we want to schedule a tournament for an odd number of teams, $2n - 1$, we can take any schedule for $2n$ teams and pick a team j to be the *dummy team*. Whatever team is scheduled to play the dummy team in Round i then is said to have a *bye* in that round. The schedule above has the property that when we select team ∞ to be the dummy, then no team has a break in the schedule. Surprisingly, this is the only schedule for odd number of teams with this property, as discovered by Mariusz Meszka and the author [4], who also proved in that for any even number of teams there exists a unique schedule in which every team has one bye and no break.

Another important aspect of round robin tournaments is the *carry-over effect*. When there are games $k - j$ in Round i and $k - t$ in Round $(i + 1)$, we say that team t receives carry-over effect from team j in Round $(i + 1)$. If we look at a Kirkman tournament as described in Construction 2, we can see that team 1 receives carry-over effect from team 6 in five rounds, namely in Rounds 2 through 6, and once from team ∞ in Round 7 (there is of course no carry-over effect in Round 1). That is, team 1 plays six times

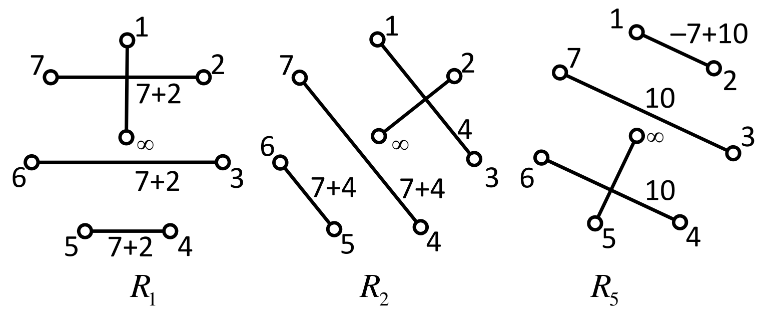

Figure 6: Rounds 1, 2, and 5 in Construction 5 of the Kirkman tournament

during the tournament against the team that played team 6 in the previous round. This may be an advantage or disadvantage, depending on whether team 6 is the best or worst team in the tournament. Similarly, all teams except ∞ receive the carry-over effect from the same team five times, while ∞ receives it in each round from different team. This is due to the rotational structure of Kirkman tournament and the special role of ∞ in it. There are tournaments that have *perfect carry-over effect*, that is, every team receives the carry-over effect from each other team at most once. However, such tournaments are known to exist only when the number of teams is either a power of two (see [7]), 20, or 22 (see [1]). Unfortunately, the tournaments with perfect carry-over effect typically have very bad break structures and balancing both properties is difficult. Some examples of leagues where both properties are well balanced can be found in [2].

Construction 6 *(Another Schedule)* If we want to find a schedule for eight teams different from the two schedules described in Constructions 1 and 5, we may think of a different tournament format. We split the teams into two divisions, East and West. First we play four rounds of interdivisional games between teams from different divisions. After that, we play three more rounds of intradivisional games.

The intradivisional part is now easy, as we already know how to schedule a tournament for four teams. A schedule for the interdivisional rounds is not difficult either. We denote the teams $W1, W2, W3, W4$ and $E1, E2, E3, E4$. Then for Round 1 we choose the games $W1-E1, W2-E2, W3-E3, W4-E4$. For Round 2 we choose $W1-E2, W2-E3, W3-E4, W4-E1$ (note that $E4$ wraps around to $E1$), for Round 3 we choose $W1-E3, W2-E4, W3-$

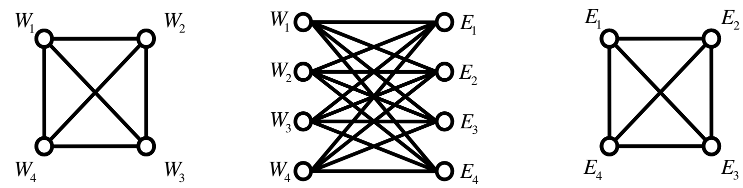

Figure 7: Bipartite interdivisional tournament and two intradivisional tournaments

$E1, W4 - E2$, and finally for Round 4 we have the remaining games $W1 - E4, W2 - E1, W3 - E2, W4 - E3$. This part of our schedule is called a *bipartite tournament* and the graph we use to model it is called the *complete bipartite graph* $K_{4,4}$—its vertex set consists of two disjoint sets of size four, W and E, and there are edges between every two vertices belonging the two different sets, while no edge joins any vertices within the same set.

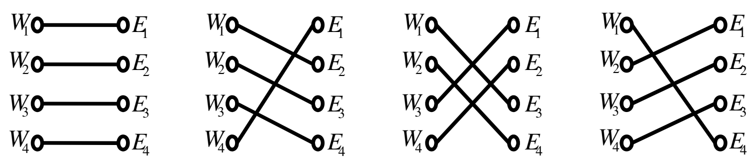

Figure 8: All four rounds of the bipartite tournament

One may now ask a natural question. The one-division 8-team tournament and the two-division 8-team tournament look different, but are they really? Maybe if we rename the teams in our two-divisional tournament, and re-order the rounds, we get the same tournament. If not, then the tournaments are structurally different, or, as we say in graph theory, *nonisomorphic.* In general, to distinguish whether two tournaments are isomorphic is difficult and we are not going to investigate it in full generality. However, we can convince ourselves that the two-divisional tournament is different from the one-division tournaments. It takes some time, but we can show that in the first tournament it is impossible to split the teams into two divisions such that there are four rounds with only interdivisional games and three rounds with just intradivisional games. Even Kirkman and Steiner tournaments are nonisomorphic. We can convince ourselves that any two rounds of a Kirkman tournament when put together as graphs give a cycle of length eight, while in a Steiner tournament the first two rounds give two disjoint cycles of length four.

On the other hand, no matter how we schedule a tournament for four teams, the result is always the same. That means we can always rename the teams in one of the tournaments (mathematically speaking, renaming is what we call "finding an isomorphism") and then possibly re-order the rounds in that tournament to obtain the other one. It would take a lot of time but you can show with paper and pencil that there is only one tournament for six teams. That is, all tournaments for six teams can be obtained by re-ordering one particular tournament. Which means that you can schedule this "structurally unique" tournament in many different ways (exactly in $5!3! = 720$ ways, if the order of games played on a particular day matters).

To illustrate how difficult it is to find whether two tournaments are isomorphic, we list the number of nonisomorphic tournaments for small number of teams: There is only one for four and six teams, there are 6 tournaments for eight teams, 396 tournaments for ten teams, and $526,915,620$ tournaments for twelve teams. This number was found with the help of a sophisticated computer program (based on *hill climbing algorithm*) by Jeff Dinitz, David Garnick, and Brendan McKay in 1994 [3]. In 2008, Petteri Kaski and Patric Östergård [5] used another algorithm to show that there are $1,132,835,421,602,062,347$ nonisomorphic tournaments with fourteen teams! Beyond fourteen teams, the numbers are so large that only a rough estimate is known, and it is unlikely that even with the most powerful computers an exact number for sixteen teams would be determined in a near future.

There have been too many papers published about various aspects of round robin tournaments to be listed here. We refer the reader to a survey on tournament scheduling in [2].

References

1. I. Anderson, Balancing carry-over effects in tournaments, ıCombinatorial designs and their applications (Milton Keynes, 1997), Chapman & Hall/CRC Res. Notes Math. 403, Chapman and Hall/CRC, Boca Raton 1–16.
2. J. H. Dinitz, D. Froncek, E. R. Lamken, and W. R. Wallis, Scheduling a tournament, In: C. J. Colbourn and J. H. Dinitz (Editors), *The Handbook of Combinatorial Designs, 2nd edition*, Chapman and Hall/CRC, Boca Raton (2007) 591–606.
3. J. H. Dinitz, D. K. Garnick, and B. D. McKay, There are $526,915,620$ nonisomorphic one-factorizations of K_{12}, *Journal of Combinatorial Designs* 2 (1994) 273–285.
4. D. Froncek and M. Meszka, Round robin tournaments with one bye and no breaks in home-away patterns are unique, in: *Multidisciplinary Scheduling : Theory and Applications (Graham Kendall, Edmund K. Burke, Sanja Petrovic, Michel Gendreau; eds.)* Springer, (2005), 331-340
5. P. Kaski and P. R. J. Östergård, There are $1,132,835,421,602,062,347$ nonisomorphic one-factorizations of K_{14}, *Journal of Combinatorial Designs* 17 (2008) 147–159.
6. T. P. Kirkman, On a problem in combinatorics, *Cambridge and Dublin Mathematical Journal* 2 (1847) 191–204.
7. K. G. Russell, Balancing carry-over effects in round robin tournaments, *Biometrika* 67(1) (1980) 127–131.

Dalibor Froncek received his M.S. and Ph.D. in 1978 and 1992 from Comenius University in Bratislava, Slovakia, and his second Ph.D. in 1994 from McMaster University in Hamilton, Ontario, Canada. After that he returned for six years to Technical University of Ostrava in his home country Czech Republic. In 2000-2001 he was Visiting Assistant Professor at the University of Vermont. Currently he is Professor at the University of Minnesota Duluth. Besides working in graph theory and design theory, he has designed schedules for many sports leagues, including several NCAA basketball and football tournaments and some professional leagues. In 2000 he created the schedule, along with Jeff Dinitz from University of Vermont, for the inaugural (and only) season of XFL. Other professional leagues he has made the schedules for, with Mariusz Meszka from University of Science and Technol-

ogy in Krakow, Poland, include Czech national soccer, hockey, and basketball leagues.