# Timetabling Round Robin Tournaments with the Consideration of Rest Durations

**Tasbih Tuffaha · Burak Çavdaroğlu · Tankut Atan**

**Abstract** Rest durations of opposing teams have recently emerged as a new fairness criterion for the timetabling of sports leagues. A rest difference is the difference between the rest durations of the opposing teams of each game. A problem, so-called rest difference problem, simultaneously schedules the games to the rounds and assigns the games of each round to the matchdays in order to minimize the total rest difference throughout a round robin tournament. In this study, we provide a mixed integer programming (MIP) formulation of the problem and propose a heuristic method which outperforms the results of the MIP on several problem instances.

**Keywords** sports timetabling · tournament fairness · rest differences · circle method · Vizing method

## 1 Introduction

Timetabling of round robin tournaments with respect to various fairness criteria is one of the most popular research topics in sports scheduling. The related literature has mainly focused on fairness issues such as balancing the carry-over effect (e.g. Russell (1980), Anderson (1997), Guedes and Ribeiro (2011))

T. Tuffaha
Department of Industrial Engineering, Kadir Has University, Cibali, İstanbul, Turkey, 34083
Tel.: +90-212-5336532
Fax: +90-212-5336515
E-mail: 20161107007@stu.khas.edu.tr

B. Çavdaroğlu
Department of Industrial Engineering, Kadir Has University, Cibali, İstanbul, Turkey, 34083
E-mail: burak.cavdaroglu@khas.edu.tr

T. Atan
Department of Industrial Engineering, Bahçeşehir University, Beşiktaş, İstanbul, Turkey, 34353 E-mail: sabritankut.atan@eng.bau.edu.tr

or the opponent's strength in a series of consecutive matches (e.g. Briskorn (2009), Briskorn and Knust (2010), Zeng and Mizuno (2013)), and minimizing the total number of breaks (e.g. de Werra (1981), Elf et al. (2003), Miyashiro and Matsui (2005), van't Hof et al. (2010)). However, fairness criteria regarding the rest durations between the consecutive games have not received much attention by researchers. In this study, we consider such a fairness criterion which aims to minimize the total number of rest differences between the opposing teams of each game in a compact round robin tournament. A *round* is composed of a set of games in which every team plays at most one game. In *compact round robin tournaments*, games are scheduled in a minimum number of rounds necessary for finishing all the games so that each team plays exactly one game in each round. Since each round usually consists of several days in practice, tournament organizers need to determine the *matchday* of each individual game in a compact round robin tournament. The problem, so-called the *rest difference problem (RDP)*, constructs a *timetable* which determines both the round and matchday of each game such that the total rest difference (the difference between the rest durations of two opposing teams in a game) throughout the tournament is minimized.

We now provide an illustrative example to describe the rest difference problem. In the example, we consider a single round robin (SSR) tournament in which each of $n = 10$ teams plays against each other exactly once. The tournament is composed of 9 rounds and each team plays exactly one game in each round (since the SSR tournament is assumed to be *compact*). The total number of games is $n(n-1)/2 = 45$. Assuming that each round is spread into 3 consecutive matchdays, the next round immediately starts the next day after the last matchday of the previous round. As a result, the tournament lasts for 27 days. Table 1 provides a feasible timetable for this example. The number of games distributed to first, second and third matchdays are selected to be 2, 2 and 1, respectively. Table 2 illustrates the rest durations of opposing teams in each game. One can observe that the opposing teams do not rest equal number of days in most of the games of this timetable. For example, before the game 13 (G13) between team 6 and team 2 in round 3, team 6 and team 2 play their games of previous round in the third and first matchdays, respectively. Therefore, team 6 has two days before its game in round 3, while team 2 has four days, which is 2 days more than the rest duration of team 6. The rest difference of 2 days between the teams in this game is considered as an unfairness that weighs against team 6 (or favors team 2). When we sum the rest differences in all games of Table 2 , the total rest difference in this tournament is found to be 38.

There exist a few studies regarding the relative rest durations of the opposing teams. Suksompong (2016) investigates three different fairness criteria, guaranteed rest time, games played difference index, and rest difference index, in asychronous round robin tournaments. Asychronous tournament is a special case of round robin tournaments in which each game is played at a distinct consecutive time (e.g. matchday). In particular, rest difference index bears a resemblance to the objective function of rest difference problem. The rest dif-

**Table 1** A feasible timetable of RDP(10,3)

| | First Matchday | | | | Second Matchday | | | | Third Matchday | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Round 1** | **G1:** | 1 vs 10 | **G2:** | 2 vs 9 | **G3:** | 3 vs 8 | **G4:** | 4 vs 7 | **G5:** | 5 vs 6 |
| **Round 2** | **G6:** | 10 vs 2 | **G7:** | 7 vs 5 | **G8:** | 8 vs 4 | **G9:** | 9 vs 3 | **G10:** | 1 vs 6 |
| **Round 3** | **G11:** | 1 vs 4 | **G12:** | 5 vs 3 | **G13:** | 6 vs 2 | **G14:** | 7 vs 10 | **G15:** | 8 vs 9 |
| **Round 4** | **G16:** | 2 vs 3 | **G17:** | 8 vs 6 | **G81:** | 9 vs 5 | **G19:** | 10 vs 4 | **G20:** | 1 vs 7 |
| **Round 5** | **G21:** | 4 vs 5 | **G22:** | 10 vs 8 | **G23:** | 2 vs 7 | **G24:** | 3 vs 6 | **G25:** | 1 vs 9 |
| **Round 6** | **G26:** | 1 vs 5 | **G27:** | 6 vs 4 | **G28:** | 7 vs 3 | **G29:** | 8 vs 2 | **G30:** | 9 vs 10 |
| **Round 7** | **G31:** | 1 vs 8 | **G32:** | 9 vs 7 | **G33:** | 10 vs 6 | **G34:** | 2 vs 5 | **G35:** | 3 vs 4 |
| **Round 8** | **G36:** | 4 vs 2 | **G37:** | 1 vs 3 | **G38:** | 5 vs 10 | **G39:** | 6 vs 9 | **G40:** | 7 vs 8 |
| **Round 9** | **G41:** | 1 vs 2 | **G42:** | 3 vs 10 | **G43:** | 4 vs 9 | **G44:** | 5 vs 8 | **G45:** | 6 vs 7 |

**Table 2** Rest durations (in days) for the timetable given in Table 2

| | First Matchday | | | | Second Matchday | | | | Third Matchday | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Round 1** | **G1:** | – | **G2:** | – | **G3:** | – | **G4:** | – | **G5:** | – |
| **Round 2** | **G6:** | 3 vs 3 | **G7:** | 2 vs 1 | **G8:** | 3 vs 3 | **G9:** | 4 vs 3 | **G10:** | 5 vs 3 |
| **Round 3** | **G11:** | 1 vs 2 | **G12:** | 3 vs 2 | **G13:** | 2 vs 4 | **G14:** | 4 vs 4 | **G15:** | 4 vs 4 |
| **Round 4** | **G16:** | 2 vs 3 | **G17:** | 1 vs 2 | **G18:** | 2 vs 4 | **G19:** | 3 vs 4 | **G20:** | 5 vs 4 |
| **Round 5** | **G21:** | 2 vs 2 | **G22:** | 2 vs 3 | **G23:** | 4 vs 2 | **G24:** | 4 vs 4 | **G25:** | 3 vs 4 |
| **Round 6** | **G26:** | 1 vs 3 | **G27:** | 2 vs 3 | **G28:** | 3 vs 3 | **G29:** | 4 vs 3 | **G30:** | 3 vs 5 |
| **Round 7** | **G31:** | 3 vs 2 | **G32:** | 1 vs 2 | **G33:** | 2 vs 4 | **G34:** | 3 vs 4 | **G35:** | 4 vs 5 |
| **Round 8** | **G36:** | 1 vs 2 | **G37:** | 3 vs 1 | **G38:** | 3 vs 3 | **G39:** | 3 vs 4 | **G40:** | 5 vs 5 |
| **Round 9** | **G41:** | 3 vs 3 | **G42:** | 3 vs 2 | **G43:** | 4 vs 3 | **G44:** | 3 vs 2 | **G45:** | 4 vs 3 |

ference index defined in the aforementioned study is equal to the maximum difference in rest durations of opposing teams among all games of a timetable, while the objective function of rest difference problem is the sum of rest differences in all games of a timetable. The study also shows that the lower bound for the rest difference index is 1, and a timetable with $n \geq 6$ teams constructed by the circle method always has the rest difference index value of 2.

Atan and Çavdaroğlu (2018) is another study concerning the rest durations of the opposing teams comparatively. In this study, they first define a fairness criterion called *rest mismatch* as the occurrence of a difference between the rest durations of two opposing teams in a game. It should be noted that a rest mismatch does not consider the magnitude of the difference in the rest durations of opposing teams. Next, they construct a timetable with both round and matchday assignments that aims to minimize the total number of rest mismatches in the tournament. The heuristic proposed in the study finds optimal results but only works for rest mismatch problems where the number of matchdays is restricted to 2 and the number of teams is a multiple of 8 (It finds near optimal results if the number of teams is a multiple of 4 but not 8).

Last but not least, Çavdaroğlu and Atan (2020) investigates the rest difference problem for given *opponent schedules*, i.e. schedules in which games have already been assigned to rounds. The study shows that the rest difference problem of a given schedule is decomposable into optimizing the rounds separately, and that each decomposed problem is an instance of the quadratic assignment problem. It also provides a polynomial-time exact algorithm for opponent schedules constructed by the circle method.

The organization of the rest of the paper is as follows. Section 2 first formally describes the problem of minimizing the sum of rest differences of teams by determining the round and matchday of each game. In Section 3, a heuristic method that decides the round and matchday of each game is described. The experiments and the comparison of the performance of the heuristic method with that of mixed integer programming formulation are also given in this section. Section 4 concludes the paper.

## 2 A Formal Description of the Problem

Suppose that there is a single round robin tournament (SRRT) for teams $i \in T = \{1, \ldots, n\}$ where $n$ is even. In each round $r \in R = \{1, \ldots, n-1\}$, the games of that round have to be assigned to consecutive matchdays $d \in D = \{1, \ldots, p\}$. The *Rest Difference Problem* with $n$ teams and $p$ matchdays is prescribed as RDP$(n, p)$. In RDP$(n, p)$, we assume that games $g \in G = \{1, \ldots, n(n-1)/2\}$ are allocated to $p$ matchdays as evenly as possible in each round. If an allocation with equal number of games in each matchday is not possible, then the numbers of games in the matchdays are assumed to be in descending order. For example, if $p = 3$ in an SRRT with $n = 10$ teams, the number of games in the matchdays should be $(2, 2, 1)$. Let $play_{g,i}$ get the value of 1 if team $i$ plays in game $g$, 0 otherwise. The number of games to be played in matchday $d$ of each round is denoted by $nGames_d$. We let $M$ be a large positive number at least equal to $p - 1$, the maximum possible difference in rest periods between two teams. The binary variable $x_{g,r,d}$ decides if game $g$ is played in matchday $d$ of round $r$ or not. The binary variable $y_{g,r}$ represents the decision of whether game $g$ is assigned to round $r$ or not. The rest difference variable $p_g^1$ ($p_g^2$) denotes the number of days the first (second) team in game $g$ had less rest than its opponent. Unless the opposing teams of game $g$ rest for equal amount of time after their games in the previous round, a difference in rest durations occurs and either $p_g^1$ or $p_g^2$ gets a positive value. The following mixed integer program (`MIP1`) formulates RDP$(n, p)$.

$$min \quad z = \sum_{g \in G} p_g^1 + p_g^2 \tag{1}$$

subject to:

$$\sum_{r \in R} \sum_{d \in D} x_{g,r,d} = 1 \quad \forall g \in G \tag{2}$$

$$\sum_{d \in D} x_{g,r,d} = 1 \quad \forall \{g \in G, i \in T : play_{g,i} = 1\}, \forall r \in R \tag{3}$$

$$\sum_{g \in G} x_{g,r,d} = nGames_d \quad \forall r \in R, \forall d \in D \tag{4}$$

$$\sum_{d \in D} x_{g,r,d} = y_{g,r} \quad \forall g \in G, \forall r \in R \tag{5}$$

$$\sum_{d \in D} d \cdot x_{g',r-1,d} - \sum_{d \in D} d \cdot x_{g'',r-1,d} + p_g^2 - p_g^1 \leq M \cdot (1 - y_{g,r})$$
$$\forall \{i, j \in T, g, g^{'}, g^{''} \in G : play_{g,i} = play_{g,j} = play_{g',i} = play_{g'',j} = 1\}, \forall r \in R \setminus \{1\} \quad (6)$$

$$x_{g,r,d} \in \{0,1\} \quad \forall g \in G, \forall r \in R, \forall p \in P \quad (7)$$

$$y_{g,r} \in \{0,1\} \quad \forall g \in G, \forall r \in R \quad (8)$$

$$p^1_{g,r}, p^2_{g,r} \geq 0 \quad \forall g \in G \quad (9)$$

The problem's objective is to generate a timetable that minimizes the total rest difference among the opposing teams of all $n(n-1)/2$ games in the tournament. Constraint 2 assigns each game to exactly one matchday in the timetable. Constraint 3 makes sure that each team plays exactly once in each round. Constraint 4 sets the number of games to be played in each matchday. The number of games in each matchday $d$ is determined a priori with $nGames_d$. Constraint 5 determines in which round a game was set to be played by the model. Constraint 6 checks for the first and second team of each game $g$ in each round, identifies the matchdays of the games $g'$ and $g''$ played by these two teams in the previous round, and finds the time difference between these matchdays. This difference is equal to the rest difference of the opposing teams in game $g$. If the time difference is in favor of the first (second) team, then $p^2_g$ ($p^1_g$) gets a positive value while $p^1_g$ ($p^2_g$) is forced to be zero. Constraint 7 through Constraint 9 give the types of decision variables.

Solving `MIP1` with commercial solvers cannot return feasible solutions for some RDP$(n,p)$ instances within a reasonable amount of time particularly when $n$ and $p$ values are increased (refer to Section 3 for more details). Even though the computational complexity of RDP has not been proven yet, we conjecture that the problem is NP-hard.

## 3 A Heuristic Method and Experimental Results

In this section, we propose a heuristic method that can be applied to the RDP$(n,p)$ where $n$ is the (even) number of teams, and $p$ is the number of matchdays. The heuristic method is conducted in two steps. First, we generate an initial opponent schedule for an SRRT with $n$ teams using the well-known circle method. The circle methods is popular in sports scheduling particularly because it minimizes the number of breaks (the occurrence of consecutive home or away games) in round robin tournaments. (More details about how the circle method is applied to construct league schedules can be found in Çavdaroğlu and Atan (2020)). In this initial schedule, the games of each round are identified without assigning the games into matchdays. Second, the rounds of the initial schedule are randomized and then the matchdays are determined for each randomized schedule using a mixed integer programming model which minimizes the total rest differences for a given opponent schedule. This MIP model is a modified version of `MIP1` presented in Section 2. The formulation of the modified MIP model (`MIP2`) can be found in Çavdaroğlu and Atan (2020). It is developed to solve the RDP$(n, p)$ in which the opponent schedule (i.e. round assignment of games) is given a priori. Using `MIP2`, matchday assignments are found for each randomly generated schedule and the timetable (i.e. opponent schedule with matchday assignments) having the lowest total rest difference value is selected as the best solution found.

The same heuristic is also applied for a set of opponent schedules generated using a procedure known as *edge coloring* or *Vizing method* (Januario et al. (2016), Januario and Urrutia (2012)). Vizing method presents a framework for the construction of an arbitrary edge coloring of a complete graph $K_n$ with $n - 1$ colors where each one of $n$ teams corresponds to a vertex, each game between teams $i$ and $j$ to an edge $(i, j)$ of $K_n$, and each color to a distinct round. Thus, edges with the same color are the games played during the same round, and each arbitrary edge coloring of a complete graph $K_n$ represents an opponent schedule for SRRT with $n$ teams.

In Table 3, the first column provides the problem instances we considered in our experimental analysis. These instances span all RDP$(n, p)$ instances having $n = 16, 18, 20$ teams and $p$ matchdays ranging from 2 to $n/2$. As mentioned earlier in the formal description, we assume the games are allocated to matchdays as evenly as possible. This allocation for each instance is shown in the second column.

The computer runs were executed on an Intel Core i7-7600U CPU 2.9 GHz computer with 8GB of RAM. `MIP1` solutions are given in the third column along with their run times. They were obtained with GAMS using either Gurobi (Gurobi Optimization Inc., 2019) or CPLEX (IBM, 2019) solver. Note that these instances were solved by both solvers but only the best solutions were reported. For `MIP1` solutions, a time limit of 10 hours (36,000 seconds) was used. In all instances, we ran the solver by the end of the time limit, and reported the best solutions found. It can be noted that in some instances `MIP1` could not even find a feasible solution within the time limit. The results also show that `MIP1` performs poorer with increasing values of $n$ and $p$ since both the number of decision variables and the number of constraints are strictly increasing functions of $n$ and $p$. Moreover, in none of the instances did `MIP1` find a lower bound better than 0.

To obtain timetables with total rest difference values better than that of the `MIP1` solutions, we generate $\lambda_{cm} = 1000$ random permutations of rounds of the initial opponent schedule that is constructed using the circle method. We also generate $\lambda_{vm} = 1000$ arbitrary edge colorings using the Vizing method. After solving `MIP2` model for each one of $\lambda_{cm}$ ($\lambda_{vm}$) schedules of RDP$(n, p)$ and selecting the schedule with the lowest total rest difference value, the heuristic using the circle method (the Vizing method) produces the results given in the fourth (fifth) column of Table 3. For each given schedule of RDP$(n, p)$, `MIP2` finds the optimal result in less than 10 seconds.

**Table 3** Results

| RDP $(n, p)$ | # games in each matchday | `MIP1` `(Time)` | Heuristic with circle method | Heuristic with Vizing method |
|---|---|---|---|---|
| (16,2) | 4,4 | 0 (66.75)* | 28 | 16 |
| (16,3) | 3,3,2 | 28 (36000) | 22* | 30 |
| (16,4) | 2,2,2,2 | 44 (36000)* | 72 | 60 |
| (16,5) | 2,2,2,1,1 | 84 (36000)* | 98 | 86 |
| (16,6) | 2,2,1,1,1,1 | 12 8(36000) | 104* | 110 |
| (16,7) | 2,1,1,1,1,1,1 | 152 (36000) | 136* | 138 |
| (16,8) | 1,1,1,1,1,1,1,1 | 180 (36000) | 160* | 166 |
| (18,2) | 5,4 | 16 (36000) | 32 | 14* |
| (18,3) | 3,3,3 | 36 (36000)* | 64 | 44 |
| (18,4) | 3,2,2,2 | 54 (36000)* | 96 | 60 |
| (18,5) | 2,2,2,2,1 | 118 (36000) | 128 | 90* |
| (18,6) | 2,2,2,1,1,1 | 176 (36000) | 160 | 124* |
| (18,8) | 2,1,1,1,1,1,1,1 | No Solution | 224 | 186* |
| (18,9) | 1,1,1,1,1,1,1,1,1 | No Solution | 256 | 218* |
| (20,2) | 5,5 | 14 (36000)* | 36 | 24 |
| (20,3) | 4,3,3 | 70 (36000) | 72 | 44* |
| (20,4) | 3,3,2,2 | 108 (36000) | 108 | 72* |
| (20,5) | 2,2,2,2,2 | 166 (36000) | 144 | 106* |
| (20,6) | 2,2,2,2,1,1 | 168 (36000) | 180 | 148* |
| (20,7) | 2,2,2,1,1,1,1 | 260 (36000) | 216 | 172* |
| (20,8) | 2,2,1,1,1,1,1,1 | No Solution | 252 | 206* |
| (20,9) | 2,1,1,1,1,1,1,1,1 | No Solution | 288 | 244* |
| (20,10) | 1,1,1,1,1,1,1,1,1,1 | No Solution | 324 | 274* |

In Table 3, for each problem instance, the method with the lowest total rest difference value is marked with an asterisk (*). It can be stated that in most problem instances the heuristic approach with either circle method or Vizing method performs better than the commercial solvers running `MIP1` model. RDP(16,2), RDP(16,4), RDP(16,5), RDP(18,3), RDP(18,4), RDP(20,2) are the only cases where `MIP1` performs better than the heuristic approaches. On the otherhand, for the cases where $p \geq 6$, the heuristic approach always outperforms `MIP1`. Thus, one could arguably claim that with increased values of $p$ the heurtic approach is more likely to give better results than the MIP model of the problem. Last but not least, in all instances where either $n = 18$ or $n = 20$, the heuristic with Vizing method outperforms the heuristic with the circle method.

## 4 Conclusion

In this study, we first introduce the rest difference problem which aims to minimize the total rest differences of opposing teams by determining the round and matchday of each game. We then present a mathematical formulation of the problem. The proposed heuristic is capable of finding timetables with better objective values than MIP formulation for most problem instances considered.

We believe that there is still some room for further improvement in the total rest difference value. The schedules obtained by swapping the rounds of a schedule generated by the circle method are isomorphic. Rather than using only a round swap, other neighborhood searches can be applied to the generated schedules potentially leading to more diverse schedules. In the second stage where we solve RDP$(n, p)$ for a given opponent schedule, the MIP model would then be able to find a timetable with even further improved total rest difference values.

On the other hand, in this research, we assume that the games are allocated to matchdays as evenly as possible. For future work, one can consider different allocations of games. Changing allocations of games in the matchdays may lead to an improved or worsened total rest difference value. Furthermore, we provide another direction for future work regarding rest differences. Rather than minimizing the total rest difference in the tournament, one can investigate to balance the rest differences over the teams.

## References

Anderson I (1997) Combinatorial designs and tournaments. Oxford University Press

Atan T, Çavdaroğlu B (2018) Minimization of rest mismatches in round robin tournaments. Computers and Operations Research 99:78–89

Briskorn D (2009) Combinatorial properties of strength groups in round robin tournaments. European Journal of Operational Research 192(3):744–754

Briskorn D, Knust S (2010) Constructing fair sports league schedules with regard to strength groups. Discrete Applied Mathematics 158(2):123–135

Çavdaroğlu B, Atan T (2020) Determining matchdays in sports league schedules to minimize rest differences. Operations Research Letters 28(3):209–216

Elf M, Jünger M, Rinaldi G (2003) Minimizing breaks by maximizing cuts. Operations Research Letters 31(5):343–349

Guedes AC, Ribeiro CC (2011) A heuristic for minimizing weighted carry-over effects in round robin tournaments. Journal of Scheduling 14(6):655–667

Gurobi Optimization Inc (2019) Gurobi Optimizer Reference Manual Version 8.1. URL `http://www.gurobi.com`

van't Hof P, Post G, Briskorn D (2010) Constructing fair round robin tournaments with a minimum number of breaks. Operations Research Letters 38(6):592–596

IBM (2019) Cplex Optimizer User Manual Version 12.8. URL `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.studio.help/pdf/gscplex.pdf`
Januario T, Urrutia S (2012) An edge coloring heuristic based on Vizing's theorem. In: Proceedings of the Brazilian Symposium on Operations Research, pp 3994–4002
Januario T, Urrutia S, Ribeiro CC, De Werra D (2016) Edge coloring: A natural model for sports scheduling. European Journal of Operational Research 254(1):1–8
Miyashiro R, Matsui T (2005) A polynomial-time algorithm to find an equitable home–away assignment. Operations Research Letters 33(3):235–241
Russell K (1980) Balancing carry-over effects in round robin tournaments. Biometrika 67(1):127–131
Suksompong W (2016) Scheduling asynchronous round-robin tournaments. Operations Research Letters 44(1):96–100
de Werra D (1981) Scheduling in sports. In: Hansen P (ed) Studies on Graphs and Discrete Programming, North-Holland, pp 381–395
Zeng L, Mizuno S (2013) Constructing fair single round robin tournaments regarding strength groups with a minimum number of breaks. Operations Research Letters 41(5):506–510