Round-robin tournament scheduling considering fairness under break constraints

September　２０２４

Fei　Xue

Round-robin tournament scheduling considering fairness under break constraints

Graduate School of Systems and Information Engineering
University of Tsukuba

September　２０２４

Fei　Xue

Abstract

Sports scheduling is the research region that concerns making a reasonable game schedule, especially a round-robin tournament, for sports competitions. One of the important factors in game scheduling is fairness. Based on the fairness of the schedule, the elements of break, carry-over effect and travel distance are considered in this study. This thesis investigates the creation of a fair schedule based on the rules of round-robin tournaments. In this format, each match involves two teams, with one team playing at home and the other away. Given the general advantage of home games, this study incorporates two conditions to ensure fairness in the scheduling of home and away games for each team: 1)No team is allowed to have more than two consecutive home or away games; 2)The difference between the number of home games and away games for each team is 1 at the end of the tournament.

First and foremost, developing feasible home-away tables for the teams is the core focus of this thesis. This task is divided into two parts: constructing home-away tables with the minimum number of breaks and another with the maximum number of breaks. The space-sequence, combined with the concept of isomorphic home-away tables, is proposed. This approach allows for the rapid classification of home-away tables with few breaks and the identification of infeasible home-away tables. By constructing home-away patterns with the largest and second largest number of breaks, we form feasible home-away tables and determine the maximum number of breaks. This thesis establishes the upper bounds for breaks, demonstrating that these bounds are tight for up to 36 teams. Secondly, by using an integer programming model to calculate the carry-over effect (COE) value, we propose an algorithm called the "successive method," which minimizes both break and COE values. Finally, the relationship between the break and traveling tournament problem(TTP) is analyzed using various distance metrics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Everyone can enjoy sports and participate in sporting events, regardless of age or gender. International sports events foster exchange between nations, bring about peace, allow people worldwide to enjoy sports, and simultaneously improve physical fitness and educational levels. The successful hosting of the Tokyo Olympics exemplifies the importance of sports in human life. The Olympics are a shared heritage of humanity, breaking down political, religious, and racial barriers, and enhancing collective human awareness. Sports events not only promote the development of sports and culture but also stimulate international cooperation, economic growth, and technological advancement. Major sporting events have a significant economic impact, driving advancements in manufacturing technologies and sales strategies, boosting local tourism, and leading to the construction of sports venues, thereby creating numerous job opportunities. Consequently, reasonable and effective planning is crucial for the success of sporting events. Failure to ensure an appropriate schedule may result in significant dissatisfaction among spectators and irreversible pain and injury to participants.

Teams and leagues aim to optimize their investments by developing well-structured schedules that meet various criteria. Effective scheduling is crucial to maximize revenues, enhance the appeal of games, and maintain the interest of both media and fans. Good schedules can significantly impact the financial performance and on-field performance of every team in a tournament, making the task of finding the best schedule complex due to the involvement of multiple decision-makers, constraints, and objectives encompassing logistics, organization, economics, and fairness.

The general problem of scheduling games in a tournament is the most extensively studied area in sports scheduling. It involves determining the dates and venues for each game. Applications of this scheduling are commonly seen in sports such as football, baseball, basketball, cricket, and hockey. However, there are other important scheduling problems in sports, such as assigning referees to games, which also involve multiple objectives. By utilizing mathematical models to solve sports scheduling problems, it is possible to efficiently determine optimal game schedules. These models help shorten travel distances for teams, allocate venues based on spectator numbers, and assist organizers in reasonably condensing schedules during the coronavirus period.

Sports scheduling research can be divided into two main directions: practical application and theoretical research. Practical applications have demonstrated that using mathematical models to solve sports scheduling problems is both rigorous and efficient. This field has garnered increasing attention from researchers in multidisciplinary areas, including operations research, scheduling theory, graph theory, combinatorial optimization, and applied mathematics. Various optimization techniques have been employed to tackle challenges in sports scheduling and management. The complexity of these problems necessitates the use of a range of exact and approximate approaches, including integer programming (IP)([1, 2, 3, 4, 5, 6]), heuristic search and metaheuristics([7, 8, 9, 10, 11]), and hybrid methods([12, 13, 14, 15, 16]).

The initial theoretical advancements in the field of sports scheduling can be traced back to the 1980s, attributed to the work of De Werra[17]. Notably, De Werra utilized concepts from graph theory to demonstrate specific results concerning the minimum number of breaks (consecutive matches with the same home-away situation) in round-robin tournaments. In the same year, Russel[18] also showed a theoretical study from a different perspective.

A large number of theoretical studies for sports scheduling consider a round-robin tournament, i.e., each pair of two teams plays a match exactly once or twice. A round-robin tournament where each team needs to play one match against other teams is known as a single round robin, and a tournament where each team plays two matches against other teams is known as a double round robin. In sports events that require many matches to be held per season, the double round-robin mode is very common. Usually, a double round-robin tournament repeats two periods, where one period consists of a single round-robin schedule and the games scheduled in the second period follow exactly the same as those in the first period. Such a double round-robin tournament schedule is called a mirrored schedule.

According to Nemhauser and Trick[19], round-robin schedules can be categorized into two types: temporally constrained and temporally relaxed. Temporally constrained schedules, also referred to as compact schedules, have a number of available game rounds that equals the number of games each team must play, plus any necessary byes for leagues with an odd number of teams. This format is commonly adopted by many professional football leagues in Europe and Latin America. Conversely, temporally relaxed schedules offer more rounds than the minimum required, allowing each team to have multiple byes. This scheduling approach is prevalent in professional leagues in North America, such as the National Basketball Association (NBA) ([20, 21]) and the National Hockey League (NHL) ([22, 23]). It is also utilized by numerous amateur sports leagues[24] and cricket leagues in Australia, England, and New Zealand ([25, 26, 27]

Studies also show that it is very effective to utilize mathematical optimization in large competitions for a great variety of real-world cases (Kendall et al.[28]; Ribeiro[29]; Durán[30]). Recently, van Bulck et al.[31] classified a wide variety of sports scheduling problems as the conditions to be considered increased.

A single round-robin tournament of high quality is helpful for such a double round-robin tournament. The thesis focuses on the theoretical research on single round-robin tournament games. The remaining part of this chapter introduces the basic rules of round-robin tournaments and the fairness criteria primarily considered in this thesis when creating the tournament schedule.

1.1 The rules of the tournament games

Research on sports scheduling began in the 1970s, with researchers from fields such as mathematical programming, graph theory, and metaheuristics getting involved. The primary focus of sports scheduling research has been on round-robin tournaments, with the most prominent being the home-and-away format. In a home-and-away format, each team has a home venue, and each match is held at the home venue of one of the teams. Games held at a team's home venue are referred to as home games, while those held at the opponent's home venue are called away games.

Figure 1.1 provides an example of a home-and-away round-robin tournament schedule. In this table, the opponent for team i on round s is indicated by the i-th row and s-th column, with "@" representing away games. Suppose that there are n teams competing with each other on a single round-robin tournament, i.e., each pair of two teams plays a match exactly once. It is assumed that n is even and $n \geq 6$. participates in one game at every round. Thus, each round has exactly $n/2$ games, and a game schedule has $n-1$ rounds. It is also assumed that each match is played at either home of the opposing team.

The quality of scheduling can be evaluated from various perspectives, including the fairness of the match schedule, the burden on the players, the number of spectators, TV broadcasting, and the distances teams have to travel. Creating high-quality schedules involves addressing several challenges, each aimed at producing schedules of good "quality" from these different perspectives.

Team	Round 1	Round 2	Round 3	Round 4	Round 5
1	@6	@4	2	5	3
2	5	@3	@1	@4	6
3	@4	2	5	@6	@1
4	3	1	6	2	@5
5	@2	6	@3	@1	4
6	1	@5	@4	3	@2

Figure 1.1: An example of a match schedule for a team in a home-and-away format round-robin tournament.

This thesis aims to study fair round-robin tournament schedules based on theories of sports scheduling rather than actual games.

1.2 Important indicators and main problems

In order to create a fair competition schedule, many factors need to be considered. The remainder of the introduction focuses on the indicators that influence the fairness of the round-robin tournament schedule and the main problems.

1.2.1 The breaks and break number maximization or minimization problem

When a team has two consecutive home games or two consecutive away games, it is called a "break." Taking Figure 1.1 as an example, Team A plays away games in both Round 1 and Round 2, so we refer to this two rounds as a break.

The balance of breaks is one important thing for making a fair tournament and the demand for break numbers varies depending on different sports events. Due to the characteristics of the round-robin tournament, it is not easy to increase or decrease the participating teams once the game starts, and the game schedule will be arranged in advance. After all competitions, the ranking will be calculated according to the results of all the matches. In addition to the results of the two teams involved, each match of the cyclical match may also affect the ranking of the other teams. Therefore, fairness is particularly important when arranging the game schedule. The fairness of sports events can be cut from many angles. For example, it is generally speaking that home games are advantageous to away games. Thus, the number of home and away games played by each team at the end of the season should be as balanced as possible. The imbalance of resting time between the games of the tournament, which is concentrated in a short period of time, is sometimes regarded as bringing unfairness. Therefore, making a game schedule that is well-balanced and fair among teams in terms of various factors has been a topic of study. High-quality scheduling can be evaluated from various perspectives, such as fairness of match schedules, player workload, attendance, TV broadcasting, and team travel distance.

Numerous studies have been conducted to address these problems, which are classified into two approaches: first-schedule then-break and first-break then-schedule. The former finds good home and away patterns according to a given match schedule. To minimize the break number, Regin [9] addressed this problem by adopting constraint programming. Ensuring that games at each home field are evenly distributed, as well as minimizing the discrepancy in consecutive home games and away games for each team after equalizing them, are considered in Easton et al. [32]. Rasmussen and Trick [12] addressed the problem by algorithm and integer programming. Miyashiro and Matsui [33] presented a polynomial time for deciding whether home and away patterns with minimum breaks exist according to the match schedule. These approaches offer advantages such as maintaining fairness among teams, leading to extensive research in this area. On the other hand, the first-break then-schedule problem finds a match schedule corresponding to a given set of home and away patterns. In this case, we need to decide whether a given set of home and away patterns is feasible, which is also known as the home-away acceptability judgment problem. Bulck and Goossens [34] showed how Benders' decomposition regulates the home-away status of games in combination with variable neighborhood search regulating the order of opponents and operates on actual instances.

Meanwhile, with the theorem on feasible home-away table of Miyashiro and Matsui [33], the break maximization problem and the break minimization problem have been theoretically proven to be equivalent problems, but the study of the break maximization problem has not been as in-depth as that of the break minimization problem. Therefore, the problem of finding a set of home and away patterns that maximizes the total number of breaks under break constraints is also relevant to travel distance minimization.

1.2.2 The carry-over effect value and carry-over effect value minimization problem

The concept of the carry-over effect (COE) is critical in understanding the fairness of sports tournament scheduling. The COE describes the potential disadvantage a team faces when it plays against a strong opponent and then immediately plays against another team in the subsequent round. The idea is that a team may be significantly fatigued after facing a tough opponent, which could negatively impact their performance in the next game, thereby giving an unintended advantage to their next opponent.

Figure 1.2 below illustrates this concept. Team C competes against team A in round r and then against team B in round r+1. If Team A is particularly strong, Team C is likely to be exhausted after the match in round r, which can affect its performance in the subsequent round against Team B. Consequently, team B benefits from playing against a potentially weakened Team C.

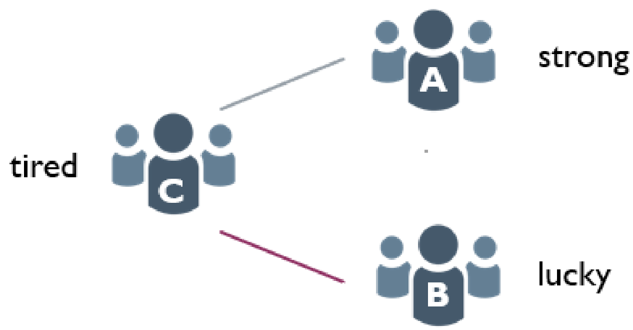


Figure 1.2: Team relations in two consecutive rounds.

From the viewpoint of fairness of the game schedules, the carry-over effect (COE) is one of vital factors to be taken into account. The *carry-over effect value* (COE value), proposed by Russell[18], is one of the measures to evaluate the fairness of the order of games. The strength of each team is not the same. When a team competes with another very strong team, they will inevitably hurt their vitality, which will affect the next game. Therefore, the subsequent team that competes with the hurt team may benefit from it. Such an affect is evaluated by the COE. When team l and team j compete in round $r+1$ after team l and team i compete in round r, team j will obtain a COE from team i. To consider the COE, rounds are considered cyclically, i.e., the first round follows the last round. Let c_{ij} be the number of COE given by team i to team j. Note that $\sum_j c_{ij}$ is constant, i.e., $n-1$. Through this concept, we can calculate the COE matrix as shown in Figure 1.3.

T/T	1	2	...	n
1	0	c_{12}	...	c_{1n}
2	c_{21}	0	...	c_{2n}
...	...	...	...	...
n	c_{n1}	...	...	0

Figure 1.3: A COE matrix for n teams.

The desired schedule is that the numbers of COEs are balanced among all pairs of teams. The degree of balance of the COE is usually measured by the COE value, defined as $\sum_{i,j} c_{ij}^2$. It is clear that the lower bound of the COE value is $n(n-1)$(Figure 1.4 shows the COE matrix which has the minimum COE value when n is 8.), and Russell[18] shows the schedule achieving this minimum COE value when n is a power of two. Unless n is a power of two, the exact minimum COE values are known only for limited small n.

T/T	1	2	3	4	5	6	7	8
1	0	1	1	1	1	1	1	1
2	1	0	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1
4	1	1	1	0	1	1	1	1
5	1	1	1	1	0	1	1	1
6	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	0

Figure 1.4: The balanced COE matrix when n is 8.

Anderson [35] developed another construction method as a starter-based method and showed a balanced schedule when the number of teams is 20 and 22. Beintema et al. [36] show that completely balanced schedules exist for all n which are a power of 2 by modifying Russell's construction method in Russell [18] based on Galois fields. Since then, many approaches have been developed to further improve the solution of the COE value minimization problem, by constraint programming (Trick [37]; Henz et al. [10]), and heuristic approaches (Miyashiro and Matsui [38]; Kidd [39]; Cao et al.[40]).

Meanwhile, Günneç and Demir [41] dealt with both breaks and COEs. They proposed a method minimizing the COE value under some constraints for breaks. The purpose of Chapter 5 is to improve solutions for their COE value minimization problem. In particular, three types of home-away tables with few breaks are evaluated. First, the properties of these home-away tables are investigated. According to operations that do not affect the feasibility and COE value, home-away tables are classified into isomorphic groups, which help to reduce the number of tables to be searched. Then, the COE value minimization problem is solved by finding a feasible schedule that minimizes the COE value for each table. To do this, an integer programming problem is adopted, which is shown in Günneç and Demir[41], together with some valid inequalities, some of which are derived from Miyashiro et al.[42] and Briskorn[2], to reduce the search space. As a result, the exact optimal COE values are found for the small number of teams, no more than 12. In addition, the upper bound of minimizing COE values is obtained for less than or equal to 20 teams.

1.2.3 Distance minimization and the traveling tournament problem

In distance minimization scheduling problems, each pair of teams is associated with a distance (or time or cost), representing the travel distance between their home venues. The objective is to create a schedule that minimizes the total distance traveled by all teams. Typically, additional constraints related to travel are also imposed.

The traveling tournament problem (TTP), introduced in the seminal paper by Easton et al. [32], is arguably the most iconic problem in this field. It is a challenging combinatorial optimization problem in sports scheduling that encapsulates the critical aspects of creating timetables where traveling distances are a significant concern. They propose the TTP as a benchmark problem for two primary reasons:

1. The problem has practical significance in modeling critical aspects of real sports schedules.
2. The combination of feasibility and optimality, along with its relatively recent emergence, makes the problem intriguing to both the operations research and constraint programming communities.

Given an even number n of teams, distances d_{ij} between the home venues of teams i and j for every $i, j = 1, \ldots, n$ (with $d_{ij} = 0$ if $i = j$), and two integer parameters L and U, and L and U define the tradeoff between distance and pattern considerations. The TTP aims to schedule a double round-robin tournament that minimizes the total distance traveled by the teams while adhering to a set of constraints. The key assumptions include:

- Each team begins the tournament at home and must return home after its last away game;
- No repeaters are allowed, meaning no two teams can play against each other in consecutive rounds;
- Every sequence of consecutive home games played by any team consists of at least L and at most U games;
- Every sequence of consecutive away games played by any team consists of at least L and at most U games;
- The sum of the total traveling distance of each team has to be minimized.

When $L = 1$ and $U = n - 1$, a team's travel may resemble that of a traveling salesman tour. Conversely, for small values of U, teams must return home more frequently, resulting in an increase in the total distance traveled.

Additionally, when a team plays two consecutive away games, it travels directly from the site of the first opponent to that of the second.

Different problems also derive from the TTP, and these descriptions provide clear distinctions between different types of traveling tournament problems, each with its own unique constraints and considerations. The TTP with Predefined Venues (TTPPV) variant introduces the additional constraint of predefined home venues for each game. In this version, the schedule must respect the predetermined home locations, while still aiming to minimize the total travel distance. This makes the problem more complex as it has to accommodate fixed venues along with the original distance minimization and scheduling constraints.

The Mirrored TTP (TTP-mirrored) variant features a mirrored schedule where the second half of the tournament is a mirror image of the first half. This means that the games played in round s are identical to those in round $s + (n - 1)$, providing a symmetric structure to the schedule. This mirroring ensures that the no repeaters constraint is automatically satisfied, as each pair of teams will face each other exactly once in the first half and once in the mirrored second half.

The TTP Non-Round-Robin (TTP-nonRR) variant does not adhere strictly to the round-robin format, allowing for more flexibility in scheduling the games. This version aims to minimize the travel distance while not being bound by the rigid structure of a round-robin tournament. The flexibility in scheduling allows for potentially more optimized travel paths, though it may lead to different competitive dynamics.

The TTP Relaxed (TTP-relaxed) variant relaxes some of the constraints present in the standard TTP. In this version, teams may have more flexibility regarding the number of consecutive home or away games they can play. Additionally, the number of available rounds might exceed the minimum required, permitting more byes and reducing the strictness of the scheduling constraints. This relaxation can lead to more practical and easier-to-manage schedules while still focusing on minimizing travel distances.

Some benchmark instances fall into specific special cases. One notable special case, introduced by Easton et al. [32], is the TTP with the mirrored requirement, where the first half of the schedule mirrors the second half with reversed venues. Another special case, introduced by Nemhauser and Trick [19], assumes a constant travel distance between any pair of cities. The exact solutions obtained so far suggest that these special cases are not as challenging as the general problem. Urrutia and Ribeiro [43] addressed the constant distance instances with the mirrored requirement for 4, 6, 8, 10, 12, and 16 teams. Fujiwara et al. [44] used a constructive method to provide optimal solutions for the constant distance instances for n teams where $n \leq 50$ and $n \equiv 4 \pmod 6$. Additionally, Rasmussen and Trick [12] utilized a Benders approach to solving all constant distance instances for up to 16 teams, as well as constant distance instances with the mirrored requirement for up to 18 teams. Irnich [45] based on a new compact IP formulation, the traveling tournament problem is solved using branch and price. The network structure is explicitly utilized, reducing the column-generation subproblem to a shortest path problem, which is efficiently solved.

The Traveling Tournament Problem is also closely related to the minimum cost problem. Briskorn et al.[46] present a branch-and-price algorithm to find a feasible schedule for a round-robin tournament with a minimum number of breaks and minimum total costs. Computational results are presented for leagues with up to 12 teams.

As mentioned in the Introduction, the reduction of movement distance is often related to the study of maximizing the number of breaks. Assuming that the distance between venues is known, Figure 1.5 shows team 1 using two different home-away patterns. The travel process of Team 1 in the left of the figure is $2 \to 1 \to 4 \to 1 \to 6 \to 1$, while the other is $4 \to 2 \to 1 \to 6 \to 1$. In the case on the left, team 1 has no breaks, while in the case on the right, team 1 has two breaks. And it is not difficult to find that in comparison, team 1 on the right has reduced the return trip from team 4's home venue, and moved directly from team 4's home venue to team 2's home venue, and then returned after the games. We believe that the schedule in right hand will reduce the total travel distance when the distance between the home venues of teams 2 and 4 is short.From this, we expect that breaks will indeed affect the travel distance cost.

Under the restriction of prohibiting 3 or more consecutive home or away games, the problem of maximizing breaks was first proposed by Russell and Leung [47] to reduce the travel distance cost of teams in the American League. In the context of the traveling tournament problem, the almost target schedule is the double round-robin tournament, and at most three consecutive series of home and away games are allowed apart from

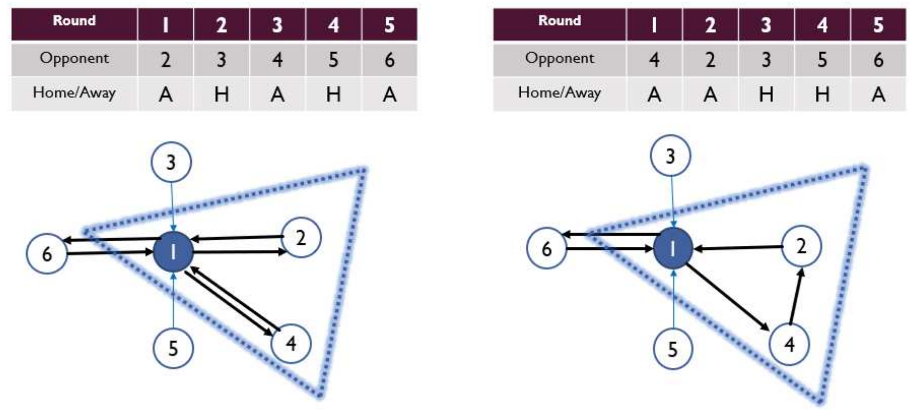


Figure 1.5: The difference between the two types of team1 's travel distance.

the assumption of [47]. Urrutia and Riberio [43] discussed that the number of travels and the number of breaks were related, break maximization and distance minimization were used to derive lower bounds for the traveling tournament problem. Suzuka et al. [48] gave a unified view to the three problems which are break maximization, and break minimization and minimizing travel distance. A detailed review of research results on minimizing travel trips with constant distance and maximizing breaks can be found in Van Bulck and Goossens [49].

1.3 The main research purpose

Chapter 2 provides a foundational understanding of breaks, home-away patterns, and home-away tables. It also introduces key theorems related to feasible home-away tables. As the main focus of this doctoral dissertation, the study of feasible home-away tables is divided into two parts: scenarios with fewer breaks, which are detailed in Chapter 3, and scenarios with the maximum number of breaks, which are discussed in Chapter 4.

Specifying a feasible tournament schedule under the constraint of breaks presents significant challenges. In Chapter 3, for the case of home-away tables with fewer breaks, we introduce the concept of isomorphic home-away tables. Based on this concept, we propose the use of space-sequence. This chapter provides a detailed explanation of how space-sequence can be utilized to eliminate infeasible home-away tables and classify the feasible ones. Compared to existing methods, the space-sequence based approach is more efficient and faster.

In contrast, Chapter 4 addresses the challenge of finding home-away tables with the maximum number of breaks. Initially, we construct home-away patterns that maximize the number of breaks. However, these patterns alone are insufficient to form a feasible home-away table. Therefore, we also construct patterns with the second most breaks. An algorithm is proposed, resulting in an upper bound on the number of breaks for teams, applicable for up to 36 teams.

Minimizing the carry-over effect (COE) value is another critical topic in sports scheduling, yet there are relatively few studies that simultaneously consider both the COE value and the number of breaks. Chapter 5 is dedicated to achieving the minimum COE value while also minimizing the number of breaks. We modify the original integer programming model to enhance solution speed and propose an algorithm that combines the concepts of space-sequence and isomorphic home-away tables to achieve lower COE values more efficiently.

Chapter 6 explores the Traveling Tournament Problem (TTP) and examines the relationship between the number of breaks and the travel distance. This chapter delves into the intricacies of how scheduling impacts the logistics of team travel. Finally, Chapter 7 presents the conclusions and discusses future research prospects. This chapter summarizes the key findings of the dissertation and suggests potential directions for further investigation in the field of sports scheduling.

Chapter 2

Break, home-away patterns and home-away tables

This chapter focuses on break, *home-away pattern* (HAP) and *home-away table* (HAT). Different home and away situations determine the HAP, assign the HAP to each team, and the HAPs of all teams constitute the HAT. When a HAT can form a schedule, it is called a feasible HAT, and not all HATs are feasible. How to find a feasible HAT is an important topic. The first section discusses the relationship between the break, HAP and HAT. Then details how to find the feasible HAT with the minimum number of breaks and the maximum number of breaks.

2.1 The relationship between home-away pattern and home-away table

Suppose that there are n teams competing with each other in a single round-robin tournament, i.e., each pair of two teams plays a match exactly once. It is assumed that n is even and $n \geq 6$. Each team participates in one game at every round. Thus, each round has exactly $n/2$ games, and a game schedule has $n-1$ rounds. It is also assumed that each match is played at either home of the opposing team.

2.1.1 Home-away patterns and home-away tables

When a team participates in a game, it will either be playing at home or away. If one of the two competing teams is playing at home, the other team must play away, and vice versa. *home-away pattern* (HAP) is an $n-1$ length sequence whose rth element implies that the match of round r is played at home or away by one specified team. Figure 2.1 is a HAP used by a team i when n is 8, where H represents the team playing a home game in the current round, and A represents the team playing an away game in the current round.

A table consisting of a collection of n HAPs is called a *home-away table* (HAT).

A HAT is represented by a matrix with n rows corresponding to teams and $n-1$ columns indexed by rounds, where element (i, r) implies home-away venues for teams i at round r. A HAT is said to be *feasible* if there exists at least one round-robin tournament schedule with which the HAT is consistent, i.e., two teams corresponding to a match in

Team	Round						
	1	2	3	4	5	6	7
i	H	A	H	A	H	A	H

Figure 2.1: A HAP for one team when n is 8.

a round have opposite home-away venues in the HAT. If two teams are assigned to the same HAP, these two teams will not be able to compete with each other. Figure 2.2 shows two teams i and j using the same HAP. When two teams can play, if one team plays at home, the other team must play away. Therefore, the "H" and "A" situations of the two teams in the round where they can play cannot be consistent. If a HAT contains this situation that prevents teams from forming competitions, then the HAT is infeasible. In other words, each HAP can be contained only once in any feasible HAT.

Team	Round						
	1	2	3	4	5	6	7
i	H	A	H	A	H	A	H
j	H	A	H	A	H	A	H

Figure 2.2: Two teams using the same HAP.

Thus, which HAP is assigned to a row of a HAT needs to be considered, so the rows of the HAT are regarded to correspond to HAPs. Figure.2.3 shows a feasible HAT for 8 teams in which each row in the graph represents a HAP assigned to each team, and Figure.2.4 is a schedule for this HAT.

Team	Round						
	1	2	3	4	5	6	7
1	H	A	H	A	H	A	H
2	H	H	A	H	A	H	A
3	H	A	H	H	A	H	A
4	H	A	H	A	A	H	A
5	A	H	A	H	A	H	A
6	A	A	H	A	H	A	H
7	A	H	A	A	H	A	H
8	A	H	A	H	H	A	H

Figure 2.3: A feasible HAT for 8 teams.

2.1.2 Important theorems about feasible HATs

It is obvious that a feasible HAT needed to have the same numbers of homes and aways for each round. Miyashiro et al.[42] discussed a necessary condition for feasible HATs. For any subset Q of HAPs consisting of a HAT (i.e., a subset of rows in a HAT), the number of homes (resp. aways) assigned in these rows at round r is denoted by $H(Q, r)$ (resp. $A(Q, r)$). Figure.2.5 is a case of a subset Q of three HAPs when $n > 3$. The

Team	Round						
	1	2	3	4	5	6	7
1	6	@7	8	@5	3	@4	2
2	5	3	@4	6	@7	8	@1
3	7	@2	5	4	@1	6	@8
4	8	@5	2	@3	@6	1	@7
5	@2	4	@3	1	@8	7	@6
6	@1	@8	7	@2	4	@3	5
7	@3	1	@6	@8	2	@5	4
8	@4	6	@1	4	5	@2	3

Figure 2.4: A schedule for the HAT in Figure 2.3.

smaller value between $H(Q, r)$ and $A(Q, r)$ is shown in the last row. It is evident that for two teams to play against each other, one team must have a home game while the other must have an away game. Thus, the value "min" in the last row signifies the maximum number of matches among the subset Q that can be scheduled in each round. In this particular scenario, there are three HAPs assigned to three teams, where each team requires a minimum of three matches. However, the total value of "min" is 2 in this case. This indicates that a HAT containing the subset Q is infeasible, as there are at least three teams that cannot play against each other. From this, we can deduce the following Theorem1.

HAP	Round								
	1	2	3	4	5	6	7	8	9
2	A	H	H	A	H	A	H	A	H
3	A	H	A	A	H	A	H	A	H
4	A	H	A	H	H	A	H	A	H
$H(Q, s)$	0	3	1	1	3	0	3	0	3
$A(Q, s)$	3	0	2	2	0	3	0	3	0
min	0	0	1	1	0	0	0	0	0

Figure 2.5: An example for subset Q.

Theorem 1 (Miyashiro et al.[42]) *A feasible HAT satisfies that for any subset Q of HAPs consisting of the HAT,*

$$\sum_{r=1}^{n-1} \min\{H(Q, r), A(Q, r)\} \geq \frac{|Q|(|Q|-1)}{2}. \tag{2.1}$$

The number of matches among teams corresponding to Q at round r is limited by $\min\{H(Q, r), A(Q, r)\}$ since a pair of teams can compete against when their home/away venues are opposite. Therefore, the left side of Eq. (2.1) represented the possible number of matches among teams corresponding to Q, which has not to be less than the requisite number of matches among teams corresponding to Q represented by the right side. Miyashiro et al. [42]'s computational experiments demonstrated that when the number of teams is less than or equal to 26, the proposed necessary condition is also a sufficient condition for feasible 2c-HAT with few breaks, but sufficient conditions for feasible HATs are not known.

2.2 The Significance of Breaks in Sports Scheduling

In sports scheduling, a break refers to a situation where a team has the same home or away status for two consecutive rounds (i.e., two consecutive home games or two consecutive away games). The number of breaks has significant implications for the fairness and logistical costs of a tournament.

2.2.1 The Significance of Breaks

- **Fairness**: Fairness is crucial in a tournament. Too many consecutive home games might give certain teams an unfair advantage, as home games usually come with greater audience support and a familiar playing environment. Similarly, consecutive away games might lead to increased fatigue for teams and add to their travel burden. Therefore, a reasonable distribution of breaks ensures that all teams have a similar home-away distribution throughout the season, enhancing the overall fairness of the competition.

- **Fitness Management**: Frequent matches and continuous away travels can increase player fatigue, affecting their performance and health. By controlling the number of breaks, the physical condition of athletes can be better managed, reducing the risk of injuries. For example, reducing the number of consecutive away games can help teams recover their energy more effectively.

- **Logistics and Costs**: Managing the travel distance and frequency for teams is also an important logistical consideration. The number of breaks is closely related to the number of moves the team makes, and therefore also affects the travel cost. This is particularly crucial in leagues that require long-distance travel.

2.2.2 Why Minimizing Breaks is Sometimes Necessary

- **Fair Competition**: Reducing the number of breaks ensures that all teams have a balanced distribution between home and away games, preventing any team from gaining an unfair competitive advantage due to excessive consecutive home or away games.

- **Fitness Recovery**: Minimizing the number of breaks can help athletes have more balanced time to rest and recover at home, thereby reducing the risk of fatigue and injury.

- **Audience Satisfaction**: A balanced schedule can increase the excitement and competitiveness of the games, preventing a decline in audience interest due to certain teams having too many consecutive home or away games.

2.2.3 Why Maximizing Breaks is Sometimes Necessary

- **Special Scheduling Needs**: In certain situations, such as venue availability or scheduling needs for major events, maximizing breaks may be required. For example, if a venue needs to undergo maintenance or host other major events, the tournament schedule may need to be adjusted flexibly, resulting in some teams having consecutive home or away games.

- **Optimizing Travel Arrangements**: In certain geographical conditions, maximizing breaks can reduce the total travel distance. For example, if travel between certain game locations is very time-consuming and expensive, clustering these games together can reduce the frequency and total cost of travel.

- **Strategic Considerations**: Some teams might choose to cluster their home or away games at specific stages of the season based on their own situation, allowing them to have better preparation or rest periods at other times.

In summary, the management of breaks plays a crucial role in sports scheduling. Depending on the specific needs and goals of the tournament, choosing to minimize or maximize breaks can help optimize the fairness of the competition, the physical management of athletes, and logistical costs. In addition, since break is closely related to HAP, this will help us construct HAP using break.

2.3 HATS under the restriction of break number

Different numbers of breaks correspond to different HATs. We separately analyze HATs with few breaks and HATs with the maximum number of breaks. Some foundational settings regarding breaks, HAP, and HAT will be introduced in this section.

2.3.1 The break setting conditions in this paper

We consider a single round robin tournament with an even number of teams n, whereby each team plays against every other team exactly once. Since each team has one game per round, the number of matches in each round is $n/2$, and there are $n-1$ rounds in total.

Since home games are generally considered to be advantageous, in order to create a fair game table, the number of home and away games of each team should not vary greatly. Moreover, if consecutive home games are played, the audience for each game will be correspondingly reduced, so many studies also set to avoid continuous home or away games as much as possible (see, for example, [17, 50, 42, 51]).

If a team plays home or away games in both rounds r and $r+1$, the team is said to have a *break* in round $r+1$. As discussed in Section 2.2, when a team plays at home, they receive greater support and encouragement from their home audience. Additionally, players are more accustomed to their home environment and conserve energy by not having to travel to away venues. Therefore, home games are considered more advantageous compared to away games. As a result, it is crucial to balance the number of home and away games for each team and avoid an excessive number of consecutive home or away games. Consequently, the following two conditions are applied in our study:

- **Condition 1:** Home games or away games for three or more consecutive rounds are prohibited. That is, breaks in consecutive rounds are not allowed.
- **Condition 2:** The difference between the number of home and away games of each team is 1.

We call a HAP satisfying these conditions *2c-HAP* and a HAT composed of 2c-HAPs *2c-HAT*. Let $break(p)$ be the number of breaks in HAP p. When a HAP consists entirely of individual 'A' or 'H' sequences, the $break(p)$ for that HAP is equal to 0. Since the first round does not have a break, $0 \leq break(p) \leq \frac{n-2}{2} = \frac{n}{2} - 1$ for any 2c-HAP p. (Note that some research regards that a team has a break in the first round when the team plays the same home or away games in the first and last round apart from our definition.)

Figurer 2.6 provides a more intuitive representation of the three HATs with different numbers of breaks, all satisfying conditions 1 and 2. The HAT (a) with the minimum number of breaks includes two 2c-HAPs with $break(p) = 0$, while all other HAPs have only one break. Each 2c-HAP in the HAT (b) has exactly one break. The HAT (c) represents a feasible configuration with the maximum number of breaks. The feasible 2c-HATs with few breaks and 2c-HATs holding the most breaks will be introduced in Chapter 3 and Chapter 4.

2.3.2 2c-HATs with few breaks

From the viewpoint of fairness, it is important to consider consecutive home/away games in HAPs. Sometimes, the fewer breaks the better. In this study, for 2c-HATs with few breaks, three distinct classes of HATs are defined: mb-HAT, eq-HAT, and sr-HAT.

Theorem 2 (de Werra[17]) *A feasible HAT for n teams has at least $n-2$ breaks. In addition, there exists a feasible HAT for n teams with exactly $n-2$ breaks.*

A 2c-HAT that has exactly $n-2$ breaks is called a *minimum-break HAT* (mb-HAT). Since all HAPs are different from each other in any HAT, there are at most two HAPs without any breaks. Thus, in any mb-HAT, each HAP has at most one break. Accordingly, the number of breaks among teams is approximately balanced, and the numbers of home and away games are approximately balanced in each team. Since n breaks are evenly distributed to each team, a 2c-HAT that each HAP had exactly one break is called *equitable* (eq-HAT) (de Werra, 1980). Günneç and Demir[41] dealt with mb-HATs consisting of HAPs in which no breaks were allowed in the second round and the last round. This restriction avoids consecutive breaks in the first and second rounds of the next period, which employs its mirrored schedule for a double round-robin tournament, i.e., the games scheduled in the second period follow exactly the same order as those played in the previous period but with exchanged venues. If a HAP has a break, the venues of the first round and the last round are opposite since the number of rounds is odd. In this case, the first round in the second period has to have a break. Thus, it is better to avoid breaks in the second and last rounds. In tournaments, the outcome of the first match often sets the tone for the entire event. Essentially, this suggests that consecutive away games can put a team at a disadvantage, as the results of these early matches significantly influence the trajectory

Round / TEAM	1	2	3	4	5	6	7
1	H	A	H	A	H	A	H
2	A	H	A	H	A	H	H
3	H	A	H	A	H	A	A
4	A	H	A	H	H	A	H
5	H	A	H	A	A	H	A
6	A	H	H	A	H	A	H
7	H	A	A	H	A	H	A
8	A	H	A	H	A	H	A

(a) A HAT with minimum number of breaks

Round / TEAM	1	2	3	4	5	6	7
1	H	H	A	H	A	H	A
2	A	A	H	A	H	A	H
3	H	A	A	H	A	H	A
4	A	H	H	A	H	A	H
5	H	A	H	A	A	H	A
6	A	H	A	H	H	A	H
7	H	A	H	A	H	H	A
8	A	H	A	H	A	A	H

(b) A HAT that each team has one break

Round / TEAM	1	2	3	4	5	6	7
1	H	H	A	A	H	H	A
2	H	H	A	H	H	A	A
3	H	A	A	H	H	A	A
4	A	A	H	H	A	A	H
5	A	A	H	A	A	H	H
6	A	H	H	A	A	H	H
7	H	A	A	H	A	A	H
8	A	H	H	A	H	H	A

(c) A HAT with maximum number of breaks

Figure 2.6: HATs with different numbers of breaks.

of the tournament. This phenomenon becomes even more pronounced in double round-robin tournaments, where matches follow a mirrored format—teams face each other twice, once at home and once away. Any breaks at the beginning or end, could lead to teams facing more than two consecutive home or away games, thereby amplifying the impact of the scheduling on team performance. The figure 2.7 provides two examples in a double round-robin tournament where three consecutive away games occur. In these examples, rounds 8-14 are a mirror image of rounds 1-7. This phenomenon occurs because these two teams have a break in either the second round or the last round. A 2c-HAT satisfying this restriction, i.e., no break in the second and last rounds, is called a *strong restricted HAT* (sr-HAT). A strong restricted and minimum-break HAT is denoted by sr-mb-HAT and a strong restricted and equitable HAT by sr-eq-HAT.

Team	Round													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	H	H	A	H	A	H	**A**	**A**	**A**	H	A	H	A	H
2	H	A	H	A	H	**A**	**A**	**A**	H	A	H	A	H	A

Figure 2.7: An example of three consecutive away games in double round-robin tournament.

These restrictions for HATs reduce the possible HAPs, although the number of all HAPs is 2^{n-1}. Clearly, the number of HAPs each of which has at most one break is $2(n-1)$. Figure 2.8 illustrates all HAPs with at most one break when n is 6, totaling 10 HAPs. Among them, HAP 1 is a 2c-HAP without a break. Starting from HAP 2, from the second round onwards, the break of each HAP moves back one round until the break reaches the last round. HAPs 6-10 are the mirroring of HAP 1-5. Notably, mb-HATs must include HAP 1 and HAP 6, while eq-HATs are unable to include HAP 1 and HAP 6. Additionally, sr-HATs should not contain HAPs(2, 5, 7, 10) with breaks in the second and final rounds to satisfy the requirements. Moreover, the following property reduces the number of possible combinations of HAPs for our considered HATs.

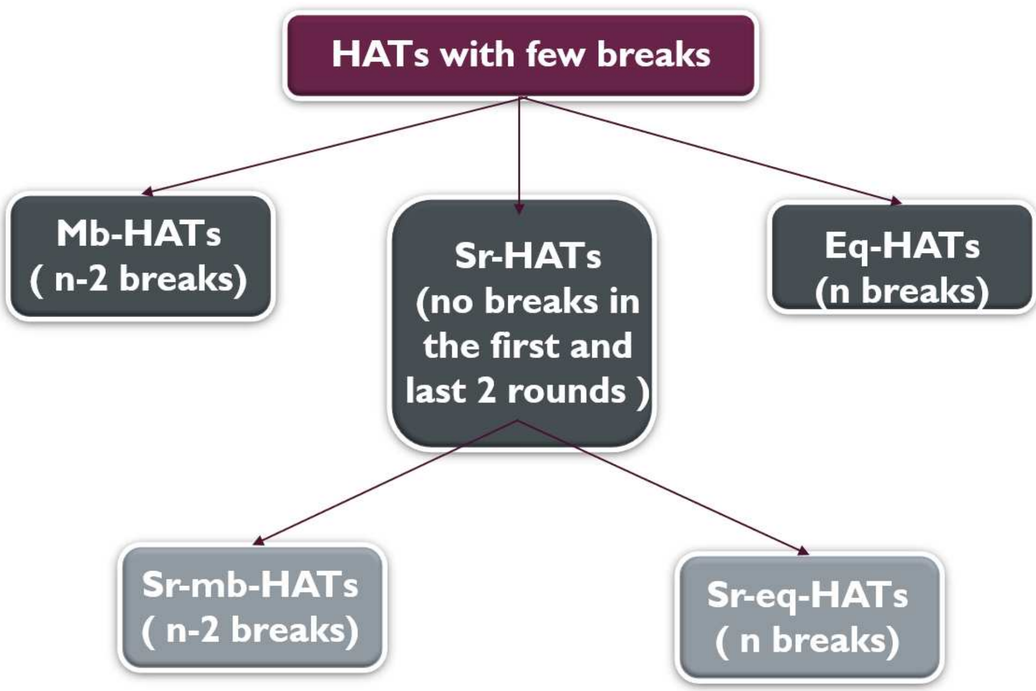
HAP	Round				
	1	2	3	4	5
1	H	A	H	A	H
2	**A**	**A**	H	A	H
3	A	**H**	**H**	A	H
4	A	H	**A**	**A**	H
5	A	H	A	**H**	**H**
6	A	H	A	H	A
7	**H**	**H**	A	H	A
8	H	**A**	**A**	H	A
9	H	A	**H**	**H**	A
10	H	A	H	**A**	**A**

Figure 2.8: The enumeration of the HAPs with at most one break for $n = 6$.

Through Figure 2.9, we can more intuitively see the relationship between the classifi-

HAP	Round							
	1	2	3	4	···	n-3	n-2	n-1
1	H	A	H	A	···	H	A	H
2	**A**	**A**	H	A	···	H	A	H
3	A	**H**	**H**	A	···	H	A	H
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n-1	A	H	A	H	···	A	**H**	**H**
n	A	H	A	H	···	A	H	A
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2n-2	H	A	H	A	···	H	**A**	**A**

(a) The 2c-HAPs with few breaks



(b) The types of HATs with few breaks

Figure 2.9: Classification of HATs with few breaks.

cation of a few break HATs and HAPs. Any mb-HAT must contain HAP 1 and HAP n; eq-HAT cannot contain HAP 1 and HAP n; sr-hat cannot contain HAP with a break in the second round and the final round, that is, HAP 2, HAP $n-1$, HAP $n+1$ and HAP $2n-2$.

Theorem 3 (de Werra[17]; Miyashiro et al.[42]) *Any feasible mb-HAT and eq-HAT consist of pairs of HAP and its complement, i.e., replacing home and away.*

In order to obtain feasible mb-HATs (resp. sr-mb-HATs, sr-eq-HATs), this theorem implies that we only consider all possible combinations of $n/2$ HAPs assigned a home game in the last round, each of which, together with their complements, forms a candidate HAT. Miyashiro et al.[42] showed that any mb-HAT satisfying Eq. (2.1) is feasible for $n \leq 26$. Although the formulation of Eq. (2.1) needs to verify its feasibility for any subset Q of HAPs, Miyashiro et al.[42] proposed a polynomial time method to do this.

2.3.3 2c-HATs with the most breaks

The feasibility of a 2c-HAT with the maximum number of breaks also needs to be considered. When only considering HAPs with the largest number of breaks, it may not be possible to form feasible HATs. Therefore, it becomes necessary to utilize HAPs with the second-largest number of breaks.

We consider a single round robin tournament with an even number of teams $n = 2n'$ with $n' \geq 2$, whereby each team plays against every other team exactly once. Since each team has one game per round, the number of matches in each round is n', and there are $n-1$ rounds in total. Since the number of teams n is even, the number of rounds $n-1$ is odd. Excluding one independent round, all pairs of rounds can form a break. It is easy to understand that the maximum value of $break(p)$ is $n'-1$. Therefore, the $break(p)$ for the 2c-HAP with the second largest number of breaks is $n'-2$. HAPs a and b in the figure 2.10 represent examples of two types of 2c-HAPs, where bold text indicates the rounds constituting a break.

HAP	Round								
	1	2	3	4	5	6	7	8	9
a	H	**A**	**A**	**H**	**H**	**A**	**A**	**H**	**H**
b	H	**A**	**A**	**H**	**H**	**A**	**A**	H	A

Figure 2.10: The two 2c-HAPs with $n'-1$ and $n'-2$ breaks

The selection of these two types of HAPs impacts the total number of breaks in the HATs, making the choice of the number of HAPs with the largest number of breaks crucial. The objective of Chapter 4 is to identify the HAT with the maximum number of breaks and assess its feasibility.

Let B_n denote the maximum number of breaks among feasible 2c-HATs for n teams. The following theorem is known about B_n.

Theorem 4 (Russell et al.[47])

- *When $n' \geq 3$, any HAT consisting of n' HAPs having $n' - 1$ breaks is infeasible. Thus,*
$$B_n < n(n' - 1)$$
- *A feasible HAT is obtained from four 2c-HAPs having $n' - 1$ breaks and $n - 4$ 2c-HAPs having $n' - 2$ breaks. Thus,*
$$B_n \geq 4(n' - 1) + (n - 4)(n' - 2)$$

We can obtain upper bound $B_n \leq n(n' - 1) - 2$ from the proof of [47], which conducts the exact value of B_n for $n' = 3$, i.e., $B_6 = 10$. However, the gap of these upper and lower bounds are larger when n' is greater.

To obtain a better bound for B_n, it is key to determine how many 2c-HAPs having $n' - 1$ breaks can be contained in a feasible HAT. We denote the set of 2c-HAPs with $n' - 1$ breaks by P^*. To find a size of a subset of P^* constructing feasible HATs, we use a necessary condition for feasible HATs discussed in [42](Therorem 1).

To find actual feasible 2c-HATs with the maximum number of breaks, we employ the method of match schedule construction by [19].

Chapter 3

The feasible 2c-HATs with few breaks

Chapter 3 primarily focuses on identifying feasible HATs with few breaks. Three distinct classes of 2c-HATs with few breaks are defined(mb-HAT, eq-HAT, and sr-HAT) in Chapter 2. The concept of isomorphic HATs will be introduced in this chapter, along with the proposal of a space-sequence, which aids in efficiently classifying 2c-HATs and eliminating infeasible 2c-HATs.

3.1 Isomorphic 2c-HATs

It is known that mb-HATs and eq-HATs are essentially equivalent because any eq-HAT can be obtained by cyclic rotation of rounds from a HAP with no break. However, in the meaning of sr-HAT, sr-mb-HATs and sr-eq-HATs are distinguished. In the following, three kinds of HATs: mb-HATs (eq-HATs), sr-mb-HATs, and sr-eq-HATs are considered. Corresponding to HAPs constructing HATs we consider, i.e., mb-HATs (eq-HATs), sr-mb-HATs and sr-eq-HATs, some notations are introduced.

Let $\mathcal{P}$ be the set of HAPs, each of which has at most one break, and $\mathcal{P}_H(\subset \mathcal{P})$ be the set of HAPs assigned a home game in the last round. Since breaks occur from the second round to the last round, the number of HAPs with one break is $2(n-2)$. Because a HAP with no breaks in $\mathcal{P}_H$ is decided to be only one, $|\mathcal{P}_H| = n-1$ holds. We denote the HAP that belongs to $\mathcal{P}_H$ and whose break at round r by p_r. For convenience, we denote the HAP with no break in $\mathcal{P}_H$ by p_1. In addition, the complement of p_r, i.e., swaping home and away in p_r, is denoted by $\bar{p}_r$. Figure 3.1 shows the labels of HAPs in Figure 2.8.

The mb-HATs and eq-HATs are known as essentially equivalent because any eq-HAT can be obtained by cyclic rotation of rounds from a round no breaks. When one HAT can be obtained by cyclic rotations of a mb-HAT, we call these two HATs *isomorphic*. For example, HATs displayed in Fig. 3.2 are isomorphic. In these HATs, "H" and "A" stand for a home game and an away game, respectively. We can observe that HAP p_r (resp. $\bar{p}_r$) is changed to $\bar{p}_{r+1}$ (resp. p_{r+1}) when the last round is moved to the first, where p_n (resp. $\bar{p}_n$) is interpreted as $\bar{p}_1$ (resp. p_1) for convenience. All three HATs mentioned, HAT(a), HAT(b), and HAT(c), are isomorphic HATs. However, HAT(a) and HAT(c) fall into the category of mb-HATs, while HAT(b) is classified as an eq-HAT. Fig.3.3 represents a graph

HAP	Round				
	1	2	3	4	5
p_1	H	A	H	A	H
p_2	**A**	**A**	H	A	H
p_3	A	**H**	**H**	A	H
p_4	A	H	**A**	**A**	H
p_5	A	H	A	**H**	**H**
$\bar{p}_1$	A	H	A	H	A
$\bar{p}_2$	**H**	**H**	A	H	A
$\bar{p}_3$	H	**A**	**A**	H	A
$\bar{p}_4$	H	A	**H**	**H**	A
$\bar{p}_5$	H	A	H	**A**	**A**

Figure 3.1: Labels of 2c-HAPs in Figure 2.8.

that corresponds to the HATs illustrated in Fig. 3.2. When the number of teams is 8, the number of 2c-HAPs with few breaks is 14, which includes two 2c-HAPs without breaks (p_1 and $\bar{p}_1$). In accordance with Theorem 3, the graph exclusively considers the partial $\mathcal{P}_H$ case, so 4 2c-HAPs need to be chosen from the 7 2c-HATs in $\mathcal{P}_H$ when n is 8. Each vertex in the graphs in Figure 3.3 represents a 2c-HAP in $\mathcal{P}_H$, and the gray vertices indicate the utilized HAPs. Taking HAT(a) as an example, it uses p_1, p_2, p_4, and p_5. Therefore, the points p_1, p_2, p_4, and p_5 on the circle of HAT(a) in Figure 3.3 are gray. By moving the seventh round of HAT(a) to the first round, and shifting the original first to sixth rounds one position back, HAT(b) is obtained. Corresponding to Figure 3.3, this is equivalent to rotating the circle of HAT(a) one position to the right, while the 2c-HAP indices and the positions of the gray vertices remain the same, resulting in the graph of HAT(b). Similarly, by moving the last four columns of HAT(a) to the front and shifting the other three columns four positions back, HAT(c) is obtained. Corresponding to Figure 3.3, this is equivalent to rotating the graph of HAT(a) four positions to the right, with the gray vertex initially at p_1 moving to p_5 (p_2 to p_6, p_4 to p_1, p_6 to p_2).

Together with Theorem 3, we obtain the following property.

Lemma 5 *Let $\sigma(i)$ represent an index of HAP in the $\mathcal{P}_H$ chosen ith ($i = 1, \ldots, n/2$), and let $P_\sigma = \{p_{\sigma(i)} \mid i = 1, \ldots, n/2\}$ and $\bar{P}_\sigma = \{\bar{p}_r \mid p_r \in P_\sigma\}$. For any $k = 1, \ldots, n-1$, the HAT constructing $P_\sigma \cup \bar{P}_\sigma$ is isomorphic to the HAT constructing $P_{\sigma^k} \cup \bar{P}_{\sigma^k}$, where*

$$\sigma^k(i) = \begin{cases} \sigma(i) + k & (\sigma(i) + k \le n-1) \\ \sigma(i) + k - (n-1) & (\sigma(i) + k > n-1) \end{cases}.$$

It is clear that we have $n-1$ isomorphic HATs for any HAT by cyclic rotations. Figure 3.4 is an example of isomorphic HATs when $n = 8$. After the first circle rotates once to become the second circle and rotates six times to become the seventh circle, rotating seven times will return it to its original state. Therefore, these seven circles are isomorphic HATs to each other.

HAT (a)

HAP	round 1	2	3	4	5	6	7
p_1	H	A	H	A	H	A	H
p_2	A	A	H	A	H	A	H
p_4	A	H	A	A	H	A	H
p_5	A	H	A	H	H	A	H
$\bar{p}_1$	A	H	A	H	A	H	A
$\bar{p}_2$	H	H	A	H	A	H	A
$\bar{p}_4$	H	A	H	H	A	H	A
$\bar{p}_5$	H	A	H	A	A	H	A

HAT (b)

HAP	round 1	2	3	4	5	6	7
$\bar{p}_2$	H	H	A	H	A	H	A
$\bar{p}_3$	H	A	A	H	A	H	A
$\bar{p}_5$	H	A	H	A	A	H	A
$\bar{p}_6$	H	A	H	A	H	H	A
p_2	A	A	H	A	H	A	H
p_3	A	H	H	A	H	A	H
p_5	A	H	A	H	H	A	H
p_6	A	H	A	H	A	A	H

HAT (c)

HAP	round 1	2	3	4	5	6	7
p_5	A	H	A	H	H	A	H
p_6	A	H	A	H	A	A	H
$\bar{p}_1$	A	H	A	H	A	H	A
$\bar{p}_2$	H	H	A	H	A	H	A
$\bar{p}_5$	H	A	H	A	A	H	A
$\bar{p}_6$	H	A	H	A	H	H	A
p_1	H	A	H	A	H	A	H
p_2	A	A	H	A	H	A	H

Figure 3.2: Example for isomorphic HATs. The HAT(b) is obtained by rotation of HAT(a) where the last round is moved to the first. The HAT(c) is obtained from HAT(a) where the fourth round is moved to the first.

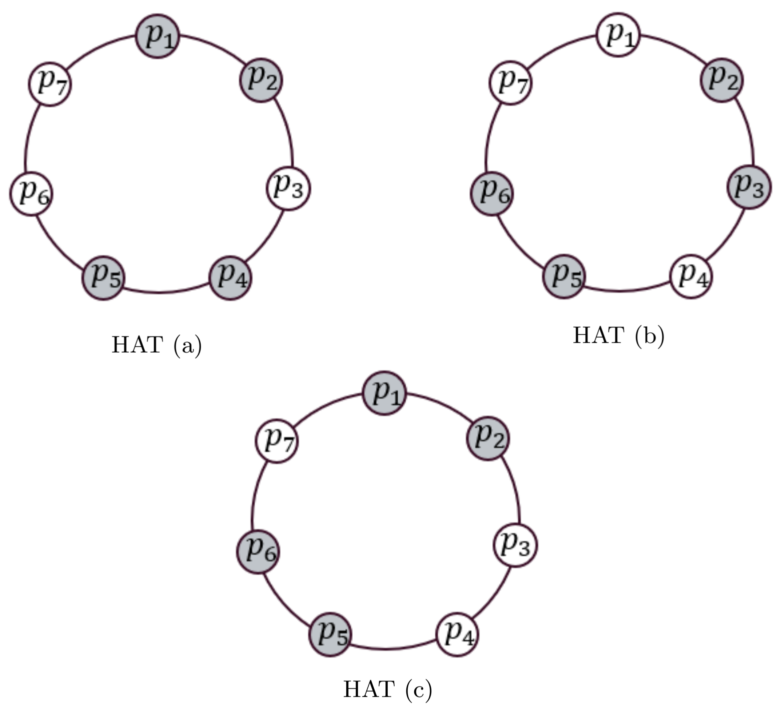


Figure 3.3: The graph corresponding to the HATs in Figure 3.2.

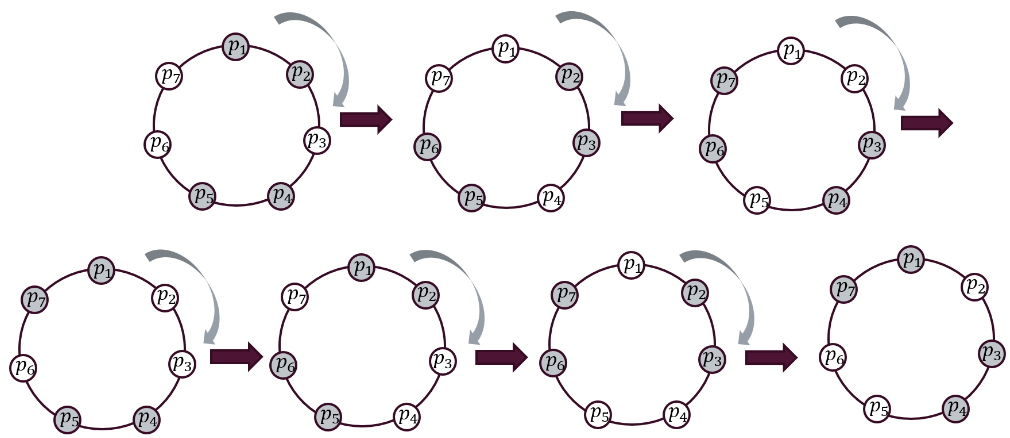


Figure 3.4: An example of isomorphic HATs when $n = 8$.

3.2 Space-sequences

The isomorphic 2c-HATs with few breaks share similar characteristics, aiding in the calculation of various indicators, such as the Carry-Over Effect (COE) value. Building upon this concept, a sequence called the *space-sequence* is proposed, which further facilitates the analysis and classification of 2c-HATs with few breaks based on their isomorphism. The space-sequence plays a crucial role in efficiently exploring the search space of 2c-HATs and determining their feasibility. To understand the characteristics of feasible 2c-HATs with few breaks and enumerate feasible 2c-HATs up to isomorphism, we introduce a notion called space-sequences.

Definition 1 *Let $\sigma(i)$ represent an index of HAP in the $\mathcal{P}_H$ chosen ith $(i = 1, \ldots, n/2)$ and satisfy $\sigma(i) < \sigma(i+1)$ for any i. For a set of HAPs $P_\sigma = \{p_{\sigma(i)} \mid i = 1, \ldots, n/2\}$, we give a sequence $(s_1, \ldots, s_{n/2})$, where*

$$s_i = \begin{cases} \sigma(i+1) - \sigma(i) - 1 & (i < n/2) \\ (n-1) - \sigma(i) + \sigma(1) - 1 & (i = n/2) \end{cases} .$$

We call this sequence $(s_1, \ldots, s_{n/2})$ space-sequence with respect to σ.

Figure 3.5 shows how we get the space-sequence. This space-sequence can be regarded as a sequence of numbers, each of which represents the count of the number of unused HAPs between the two used HAPs according to the cyclic order $p_1, \ldots, p_{n-1}$. It is obvious that the space-sequences of the isomorphic HATs are in a sequential shift or a reverse shift relationship. For instance, the space-sequences corresponding to HATs (a) (b), and (c) in Figure 3.2 are (0, 1, 0, 2), (0, 1, 0, 2), and (0, 2, 0, 1), respectively.

For any space-sequence $(s_1, \ldots, s_{n/2})$, we have $\sum_{i=1}^{n/2} s_i = (n-1) - n/2 = n/2 - 1$. Conversely, for any $n/2$ length nonnegative integer sequence whose sum of elements is equal to $n/2 - 1$, we construct a set of HAPs in $\mathcal{P}_H$ containing p_1. Therefore, there is a one-to-one correspondence relation between such sequences $(s_1, \ldots, s_{n/2})$ and 2c-HATs containing p_1.

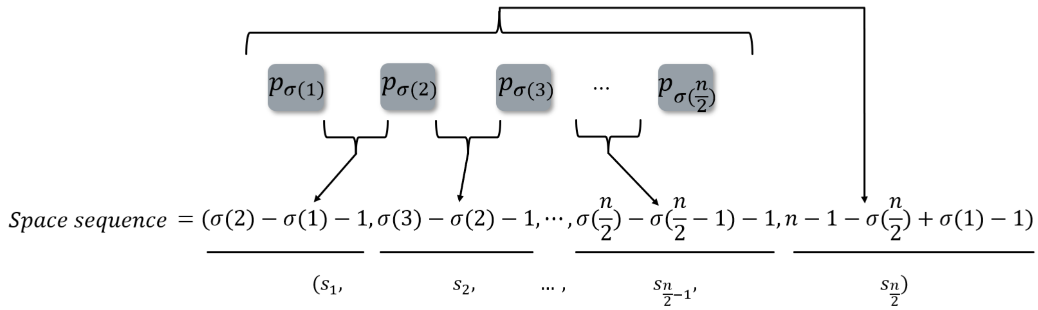
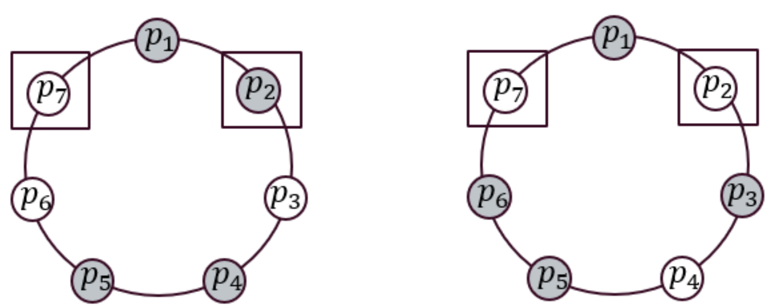


Figure 3.5: Space-sequence calculation process

Theorem 6 *2c-HAT with few breaks is isomorphic to an sr-mb-HAT if and only if the corresponding space-sequence contains two consecutive nonzero elements.*

proof According to the definition of sr-mb-HATs, p_2 and p_{n-1} are not selected, while p_1 is selected. Thus, by rotating the space-sequence that satisfies the conditions of Theorem 6, we can obtain a sequence with $s_1 \geq 1$ and $s_{n/2} \geq 1$.

On the other hand, assume that s_i and s_{i+1} are nonzero in the space-sequence $(s_1, \ldots, s_{n/2})$ with respect to σ. We rotate the given HAT by moving the $\sigma(i+1)$ round to the first round. As a result of this rotation, HAP $p_{\sigma(i+1)}$ becomes p_1. Since $p_{\sigma(i+1)-1}$ and $p_{\sigma(i+1)+1}$ are not used in the given HAT, p_{n-1} and p_2 are also not used in the rotated HAT, which implies that the obtained HAT becomes sr-mb-HAT.



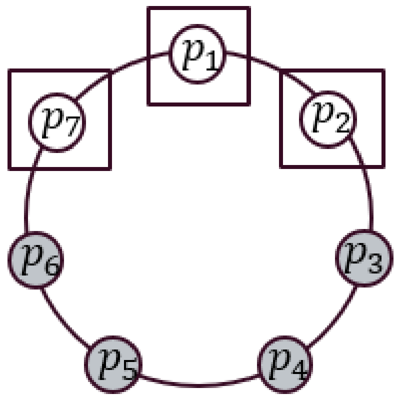
(a) space-sequence$= (0, 1, 0, 2)$ (b) space-sequence$= (1, 1, 0, 1)$

Figure 3.6: Illustration of the relationship between sr-mb-HAT and space-sequence.

This property implies that the HATs shown in Figure. 3.2 are not isomorphic to sr-mb-HAT. In Figure.3.6, the relationship between sr-mb-HAT and space-sequence is depicted. Specifically, it is observed that regardless of the rotation of the graph in graph (a), it is not possible to avoid selecting p_2 and p_7 without break when selecting p_1. However, in contrast, graph (b) demonstrates the feasibility of avoiding such a situation.

Theorem 7 *A 2c-HAT with few breaks is isomorphic to an sr-eq-HAT if and only if the corresponding space-sequence contains an element no less than 3.*

proof According to the definition of sr-eq-HATs, p_1, p_2 and p_{n-1} cannot be selected, so three consecutive unselected HAPs are necessary. When $s_i \geq 3$ in the *space-sequence*, by rotating the given HAT by moving the $\sigma(i)+2$ round to the first round, we obtain an sr-eq-HAT.



space-sequence= $(0, 0, 0, 3)$

Figure 3.7: Illustration of a sr-eq-HAT.

Figure. 3.7 demonstrates that in order to construct an sr-eq-HAT for $n = 8$, the space-sequence must contain a value greater than or equal to 3. This ensures the avoidance of selecting p_1, p_2, and p_7 simultaneously. Consequently, p_3, p_4, p_5, and p_6 must be selected in this scenario. However, it is important to note that the sr-eq-HAT formed under these conditions is infeasible, and the reasons will be further elucidated in the subsequent section.

3.3 Feasible space-sequence

We say that a space-sequence is *feasible* if the corresponding HAT is feasible. We now give a property for feasible space-sequences. When we discuss the feasibility of space-sequences, we use the following lemma together with Eq. (2.1) of Theorem 1.

Lemma 8 *For a subset Q of 2c-HAPs with few breaks in $\mathcal{P}_H$, let q_i and q_j be the minimum and maximum indices of HAPs in Q. Then we have*

$$\sum_{r=1}^{n-1} \min\{H(Q,r), A(Q,r)\} = \sum_{r=q_i}^{q_j-1} \min\{H(Q,r), A(Q,r)\}$$

proof Subsequences of HAPs in Q between the first round and the $(q_i - 1)$th round coincide with each other. The subsequences of HAPs in Q between the q_j round and the last round coincide with each other. Hence, for each of these rounds r, we have $\min\{H(Q,r), A(Q,r)\} = 0$.

Theorem 9 *Any feasible space-sequence does not contain two consecutive 0s.*

proof If $s_i = s_{i+1} = 0$, then we have $p_{\sigma(i+1)} = p_{\sigma(i)+1}$ and $p_{\sigma(i+2)} = p_{\sigma(i)+2}$, where σ represents an index of HAP. By applying Theorem 1 for $Q = \{p_{\sigma(i)}, p_{\sigma(i+1)}, p_{\sigma(i+2)}\}$, LHS of

HAP	round							
	1	2	$\cdots$	$\sigma(i)-1$	$\sigma(i)$	$\sigma(i)+1$	$\sigma(i)+2$	$\cdots$
$\vdots$								
$p_{\sigma(i)}$	A	H	$\cdots$	A	A	H	A	$\cdots$
$p_{\sigma(i)+1}$	A	H	$\cdots$	A	H	H	A	$\cdots$
$p_{\sigma(i)+2}$	A	H	$\cdots$	A	H	A	A	$\cdots$
$\vdots$								
$\min\{H(Q,r),A(Q,r)\}$	0	0	$\cdots$	0	1	1	0	$\cdots$

Figure 3.8: Example of a partial 2d-HAT with few breaks for proof of Theorem 9. Note that there exist cases where Hs and As are replaced by each other according to the parity of $\sigma(i)$. We can check that the LHS of Eq. (2.1) for $Q = \{p_{\sigma(i)}, p_{\sigma(i)+1}, p_{\sigma(i)+2}\}$ is 2.

HAP	round												
	$\cdots$	$\sigma(i)-1$	$\sigma(i)$									$\sigma(i)+9$	$\cdots$
$\vdots$													
$p_{\sigma(i)}$	$\cdots$	A	A	H	A	H	A	H	A	H	A	H	$\cdots$
$p_{\sigma(i)+2}$	$\cdots$	A	H	A	A	H	A	H	A	H	A	H	$\cdots$
$p_{\sigma(i)+3}$	$\cdots$	A	H	A	H	H	A	H	A	H	A	H	$\cdots$
$p_{\sigma(i)+5}$	$\cdots$	A	H	A	H	A	H	H	A	H	A	H	$\cdots$
$p_{\sigma(i)+7}$	$\cdots$	A	H	A	H	A	H	A	H	H	A	H	$\cdots$
$p_{\sigma(i)+8}$	$\cdots$	A	H	A	H	A	H	A	H	A	A	H	$\cdots$
$p_{\sigma(i)+10}$	$\cdots$	A	H	A	H	A	H	A	H	A	H	A	$\cdots$
$\vdots$													
$\min\{H(Q,r),A(Q,r)\}$	$\cdots$	0	1	1	2	3	3	3	3	2	1	1	$\cdots$

Figure 3.9: Example of a partial 2c-HAT with few breaks for proof of Theorem 10. Note that there exist cases where Hs and As are replaced by each other according to the parity of $\sigma(i)$. We can check that the LHS of Eq. (2.1) for $Q = \{p_{\sigma(i)}, p_{\sigma(i)+2}, p_{\sigma(i)+3}, p_{\sigma(i)+5}, p_{\sigma(i)+7}, p_{\sigma(i)+8}, p_{\sigma(i)+10}\}$ is 20.

Eq. (2.1) is equal to $\min\{H(Q,\sigma(i)), A(Q,\sigma(i))\} + \min\{H(Q,\sigma(i)+1), A(Q,\sigma(i)+1)\} = 2$ (see Figure 3.8), which is less than the RHS of Eq. (2.1). Thus, the HAT is not feasible.

Theorem 9 implies that a feasible *space-sequence* has $s_{n/2} > 0$ when $s_1 = 0$ because two consecutive 0s are prohibited in any cyclic rotation of the space-sequence.

Theorem 10 *Any feasible space-sequence has no proper subsequence*$(1,0,1,1,0,1)$.

proof If the subsequence $(1,0,1,1,0,1)$ is contained, by applying Theorem 1 for

$$Q = \{p_{\sigma(i)}, p_{\sigma(i)+2}, p_{\sigma(i)+3}, p_{\sigma(i)+5}, p_{\sigma(i)+7}, p_{\sigma(i)+8}, p_{\sigma(i)+10}\},$$

LHS of Eq. (2.1) is equal to 20 (see Fig. 3.9), which is less than the RHS of Eq. (2.1). Thus, the HAT is not feasible.

Theorem 11 *Any feasible space-sequence has no proper subsequence, such as* $(\underbrace{1,1,...1}_{k},0,\underbrace{1,1,...,1}_{k},0)$ *and* $(0,\underbrace{1,1,...1}_{k},0,\underbrace{1,1,...,1}_{k})$ *for any* $k \geq 0$.

proof Assume that a *space-sequence* has $(\underbrace{1,1,...1}_{k},0,\underbrace{1,1,...,1}_{k},0)$ as its proper subsequence. Without loss of generality, a HAT containing of $Q^k = \{p_1, p_3, \ldots, p_{2k+1}\} \cup \{p_{2k+2}, p_{2k+4} \cdot \ldots, p_{4k+2}\} \cup \{p_{4k+3}\}$ is considered. By the induction hypothesis with respect to k, we shall show that Q^k does not satisfy Eq. (2.1) of Theorem 1.

When $k = 0$, we have shown that Eq. (2.1) is not satisfied for Q^0 from Theorem 9.

It is assumed that Eq. (2.1) is not satisfied for Q^{k-1}. By rotation of rounds, Eq. (2.1) is not satisfied for $\tilde{Q}^{k-1} = \{p_3, \ldots, p_{2k+1}\} \cup \{p_{2k+2}, p_{2k+4} \cdot \ldots, p_{4k}\} \cup \{p_{4k+1}\} = Q^k \setminus \{p_1, p_{4k+2}, p_{4k+3}\} \cup \{p_{4k+1}\}$. Additionally, $\tilde{\tilde{Q}}^{k-1} = Q^k \setminus \{p_1, p_{4k+3}\} = \tilde{Q}^{k-1} \setminus \{p_{4k+1}\} \cup \{p_{4k+2}\}$ is defined. Since each venue in p_{4k+1} and p_{4k+2} coincides with each other except for the $(4k+1)$th round and the subsequences of p_1 and p_{4k+3} between the first round and the $(4k+2)$th round have opposite venues in each round (see Fig. 3.10), we have

$$\begin{aligned}\min\{H(Q^k, r), A(Q^k, r)\} &= \min\{H(\tilde{\tilde{Q}}^{k-1}, r), A(\tilde{\tilde{Q}}^{k-1}, r)\} + 1 \\ &= \min\{H(\tilde{Q}^{k-1}, r), A(\tilde{Q}^{k-1}, r)\} + 1, \quad \forall r = 1, \ldots, 4k, 4k+2.\end{aligned}$$

Since $\min\{H(\tilde{Q}^{k-1}, 4k+1), A(\tilde{Q}^{k-1}, 4k+1)\} = 0$, then $\min\{H(\tilde{\tilde{Q}}^{k-1}, 4k+1), A(\tilde{\tilde{Q}}^{k-1}, 4k+1)\} = 1$, which implies that

$$\min\{H(Q^k, 4k+1), A(Q^k, 4k+1)\} = \min\{H(\tilde{Q}^{k-1}, 4k+1), A(\tilde{Q}^{k-1}, 4k+1)\} + 2.$$

Thus, we have

$$\begin{aligned}&\sum_{r=1}^{n-1} \min\{H(Q^k, r), A(Q^k, r)\} \\ &= \sum_{r=1}^{4k+2} \min\{H(Q^k, r), A(Q^k, r)\} \\ &= \sum_{r=1}^{4k+2} \min\{H(\tilde{Q}^{k-1}, r), A(\tilde{Q}^{k-1}, r)\} + 4k + 3 \\ &= \sum_{r=1}^{n-1} \min\{H(\tilde{Q}^{k-1}, r), A(\tilde{Q}^{k-1}, r)\} + 4k + 3 \\ &< \frac{(2k+1)2k}{2} + 4k + 3 = \frac{(2k+3)(2k+2)}{2} = \frac{|Q^k|(|Q^k| - 1)}{2}\end{aligned}$$

	round															
HAP	1	2	3	4	5	$\cdots$	$2k+1$	$2k+2$	$2k+3$	$2k+4$	$\cdots$	$4k+1$	$4k+2$	$4k+3$	$4k+4$	$\cdots$
p_1	A	H	A	H	A	$\cdots$	A	H	A	H	$\cdots$	A	H	A	H	$\cdots$
p_3	H	A	A	H	A	$\cdots$	A	H	A	H	$\cdots$	A	H	A	H	$\cdots$
p_5	H	A	H	A	A	$\cdots$	A	H	A	H	$\cdots$	A	H	A	H	$\cdots$
$\vdots$																
p_{2k+1}	H	A	H	A	H	$\cdots$	A	H	A	H	$\cdots$	A	H	A	H	$\cdots$
p_{2k+2}	H	A	H	A	H	$\cdots$	H	H	A	H	$\cdots$	A	H	A	H	$\cdots$
p_{2k+4}	H	A	H	A	H	$\cdots$	H	A	H	H	$\cdots$	A	H	A	H	$\cdots$
$\vdots$																
p_{4k+2}	H	A	H	A	H	$\cdots$	H	A	H	A	$\cdots$	H	H	A	H	$\cdots$
p_{4k+3}	H	A	H	A	H	$\cdots$	H	A	H	A	$\cdots$	H	A	A	H	$\cdots$
$\vdots$																

Figure 3.10: Example of a partial 2c-HAT with few breaks for proof of Theorem 11

Corollary 12 *Any feasible space-sequence does not contain any subsequence, which consists of more than two 0s and several 1s.*

proof Because of the feasibility of space-sequences, such a subsequence avoids two consecutive 0s. Thus, it contains a subsequence like as $(0\underbrace{1,\ldots,1}_{k},0,\underbrace{1,\ldots,1}_{l},0)$. If $k \leq l$, then it contains $(0,\underbrace{1,\ldots,1}_{k},0,\underbrace{1,\ldots,1}_{k})$, otherwise it contains $(\underbrace{1,\ldots,1}_{l},0,\underbrace{1,\ldots,1}_{l},0)$.

By applying these properties, we can enumerate HATs efficiently by avoiding infeasible 2c-HATs with few breaks . In the rest of this section, we prove that there are no feasible sr-eq-HATs for $n \leq 14$ and $n = 18$.

Theorem 13 *When $n \leq 14$, there is no feasible sr-eq-HAT.*

proof From Theorem 7, the corresponding space-sequence for a feasible sr-eq-HAT has an element $s_i \geq 3$. Since a space-sequence $(s_1, \ldots, s_{n/2})$ is given by distributing the value of $n/2 - 1$ into $n/2$ nonnegative integer values, we need no less than three 0s in the sequence when it has $s_i = 3$. Corollary 12 implies that there are at least two elements more than 1 when the sequence has at least three 0s. This means we need no less than four 0s in the sequence. To avoid two consecutive 0s, there must be at least four nonzero elements. Therefore, the length of the space-sequence is at least 8, which implies that $n \geq 16$ for feasible sr-eq-HATs.

Theorem 14 *When $n = 18$, there is no feasible sr-eq-HAT.*

proof As we discuss in the proof of Theorem 13, any space-sequence has to have at least two elements more than 1 and four 0s when assuming that $s_i \geq 3$. If the sequence contains at least two elements no less than 3, we need five 0s, which is impossible when $n < 20$ because we need five nonzero elements to avoid two consecutive 0s. Hence, the sequence has one 3, one 2, four 0 s and three 1 s. The possible sequence prohibiting consecutive 0s and patterns described in Theorem 11 is only $(0, 1, 0, 2, 0, 1, 1, 0, 3)$ and its cyclic rotation or inverse rotation. From this *space-sequence*, we obtain a set of HAPs $\{p_1, p_2, p_4, p_5, p_8, p_9, p_{11}, p_{13}, p_{14}\}$. Note that, although this HAT contains p_2, a sr-eq-HAT can be obtained by rotation from this HAT as shown in Theorem 7. We now consider Eq. (2.1) in Theorem 1 for this set of HAPs. As we shown in Fig. 3.11, LHS is equal to 35, which is a shortage for the number of games, i.e., RHS of Eq. (2.1).

3.4 Enumeration of 2c-HATs with few breaks

This section describes how to exhaustively enumerate non-isomorphic 2c-HATs with few breaks and how to use space-sequence to classify these 2c-HATs. The process are shown below.

Step 1 Enumerate all the number sequences that meet all the following conditions.

1 The first element in the sequence is 0.

2 The number of elements remaining in the sequence is $n/2 - 1$.

HAP	round																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p_1	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H
p_2	A	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H
p_4	A	H	A	A	H	A	H	A	H	A	H	A	H	A	H	A	H
p_5	A	H	A	H	H	A	H	A	H	A	H	A	H	A	H	A	H
p_8	A	H	A	H	A	H	A	A	H	A	H	A	H	A	H	A	H
p_9	A	H	A	H	A	H	A	H	H	A	H	A	H	A	H	A	H
p_{11}	A	H	A	H	A	H	A	H	A	H	H	A	H	A	H	A	H
p_{13}	A	H	A	H	A	H	A	H	A	H	A	H	H	A	H	A	H
p_{14}	A	H	A	H	A	H	A	H	A	H	A	H	A	A	H	A	H
$\min\{H(Q,r), A(Q,r)\}$	1	2	2	3	4	4	4	4	3	3	2	2	1	0	0	0	0

Figure 3.11: Partial 2c-HAT with few break for proof of Theorem 14. We can check that the LHS of Eq. (2.1) for $Q = \{p_1, p_2, p_4, p_5, p_8, p_9, p_{11}, p_{13}, p_{14}\}$ is 35.

3 The sum of all elements in the sequence is $n/2 - 1$.

4 The sum of two consecutive elements in the sequence is greater than or equal to 1.

5 The last element in the sequence is greater than or equal to 1.

Step 2 Combine theorems to delete infeasible space-sequences.

1 Since the first step already prevents consecutive zeros in the space-sequence, the space-sequences related to Theorem 9 no longer need to be considered.

2 Based on Theorem 10, any feasible space-sequence cannot contain the subsequence $(1, 0, 1, 1, 0, 1)$, so the related sequences need to be removed.

3 Based on Theorem 11, any feasible space-sequence has no proper subsequence, such as $(\underbrace{1, 1, ...1}_{k}, 0, \underbrace{1, 1, ..., 1}_{k}, 0)$ and $(0, \underbrace{1, 1, ...1}_{k}, 0, \underbrace{1, 1, ..., 1}_{k})$ for any $k \geq 0$, so the related sequences need to be removed.

Step 3 Classify HATs based on theorems, with the main focus on determining the existence of sr-mb-HAT and sr-eq-HAT.

1 In sr-mb-HAT, p_1 without a break must be included, and the HAPs with breaks in the second round(p_2) and last round(p_{n-1}) must not be included. To satisfy both conditions, the space-sequence must contain at least two consecutive elements greater than zero.

2 In sr-eq-HAT, p_1, p_2, and p_{n-1} must all be avoided, so the space-sequence must contain at least one element greater than 3.

Since the space-sequence corresponding to the isomorphic 2c-HAT has the characteristic of circulation, in order to avoid repeated calculations, the first element in the sequence is fixed to 1(condition 1). From Theorem 9, it can be seen that a space-sequence containing two consecutive 0s is infeasible, so in order to facilitate the subsequent calculations, this operation is performed in step 1 first(conditions 4 and 5). If the length of the sequence reaches the expected length, it checks whether the sequence is valid (whether it meets the conditions). If it is valid, it is added to the list of all sequences to be output. Otherwise, continue adding numbers until all possible sequences are found. Then, we use Excel to delete the infeasible space-sequence(Step 2 and Step 3).

Figure 3.12 provides an example when n equals 10. We enumerate all sequences that satisfy the definition of the space-sequence, as shown in the second column of Figure 3.12. Combining with Theorem 11, we conclude that sequences No. 1, 3, 4, and 7 in Figure 3.12 are not feasible. Furthermore, we can deduce that sequences No. 3 and 5 are isomorphic HATs. It only takes a short time to enumerate the space-sequences for teams, so this allows us to quickly identify feasible 2c-HATs with few breaks and classify them.

No.	Space-sequence	Feasibility	Characteristic
1	(0, 1, 0, 1, 2)	x	
2	(0, 1, 1, 0, 2)		Isomorphic to 5
3	(0, 2, 1, 0, 1)	x	
4	(0,1, 2, 0, 1)	x	
5	(0, 2, 0, 1, 1)		Isomorphic to 2
6	(0, 1, 1, 1, 1)		Independent
7	(0, 1, 0, 2, 1)	x	

Figure 3.12: An example of *space-sequence* when $n = 10$.

Table 3.1 shows the number of non-isomorphic 2c-HATs with few breaks, which compares the number of combinations to choose HAPs, the numbers of feasible HATs shown in the columns of "# feasible", and the numbers of non-isomorphic feasible HATs shown in the columns of "# noniso". In the "# feasible" column, we first enumerate all space-sequences with $n/2 - 1$ elements without restricting the first element to 0 and remove the infeasible space-sequences. The remaining space-sequences corresponding to all HATs are then verified using the IP model.

As shown in Table 3.1, the number of non-isomorphic feasible HATs is very small which will play a role in calculating the COE value in Chapter 5.

Table 3.1: The numbers of candidate HATs.

	mb-HATs			sr-mb-HATs			sr-eq-HATs		
n	$C^{n-2}_{\frac{n}{2}-1}$	# feasible	# non-iso.	$C^{n-4}_{\frac{n}{2}-1}$	# feasible	# non-iso.	$C^{n-4}_{\frac{n}{2}}$	# feasible	# non-iso.
6	6	3	1	1	1	1	-	-	-
8	20	8	2	4	2	1	1	-	-
10	79	10	2	15	5	2	6	-	-
12	252	30	4	56	12	3	28	-	-
14	924	49	5	210	29	5	120	-	-
16	3,432	136	12	792	71	10	495	1	1
18	12,870	216	13	3,003	171	13	2,002	-	-

Identifying feasible Home-Away Tables HATs that satisfy specific break number conditions is a complex task. In the case of a HAT with the minimum number of breaks, Miyashiro et al. proved a theorem (Theorem 1) that leads to a polynomial-time algorithm to check whether the given HAT satisfies the necessary conditions. On the other hand, regarding the space-sequence, the order of the HAP is very important. This method utilizes Theorem 1 to make use of the ordered HAP. Even with known necessary conditions for feasible HATs, the challenge escalates as the number of teams increases, leading to an exponential rise in computational complexity. This problem becomes particularly daunting in large-scale tournaments. The proposal of space-sequence significantly enhances the process of categorizing 2c-HATs with few breaks. This method allows for rapid classification and efficient elimination of infeasible solutions. The space-sequence framework also facilitates the quick identification of non-isomorphic 2c-HATs, which is crucial for optimizing the COE values. In summary, while the task of finding feasible HATs that meet break number conditions is inherently difficult, the innovative use of space-sequence and isomorphic HATs concepts provides a powerful toolset for tackling this challenge. These methodologies offer a promising path forward in the quest for optimizing tournament schedules, balancing fairness, and minimizing computational complexity. From the calculation results, we also know that when n is less than 18, if a HAT-related space-sequence does not contain the sequence of Theorems 9, 10, 11 and Corollary 12, HAT is feasible.

Chapter 4

The feasible 2c-HATs with most breaks

The feasibility of a HAT with the maximum number of breaks is considered. When only considering 2c-HAPs with the largest number of breaks, it may not be possible to form feasible HATs. Therefore, it becomes necessary to utilize 2c-HAPs with the second-largest number of breaks. The selection of these two types of HAPs impacts the total number of breaks in the HATs, making the choice of the number of HAPs with the largest number of breaks crucial. The objective of this chapter is to identify the 2c-HAT with the maximum number of breaks and assess its feasibility.

4.1 The upper bound of t_{max}

We consider a single round robin tournament with an even number of teams $n = 2n'$ with $n' \geq 2$, whereby each team plays against every other team exactly once in this chapter. Since each team has one game per round, the number of matches in each round is n', and there are $n-1$ rounds in total.

We denote the set of 2c-HAPs with $n'-1$ breaks by P^*. To find a size of a subset of P^* constructing feasible HATs, we use a necessary condition for feasible HATs discussed in [42](Theorem 1).

Let B_n denote the maximum number of breaks among feasible 2c-HATs for n teams and t_{max} represents the number of HAPs used from P^*. We need to use as many HAPs from P^* as possible to maximize the number of breaks. Let $\mathcal{N} = \{N \subseteq P^* \mid \sum_{r=1}^{n-1} \min\{H(Q,r), A(Q,r)\} \geq \frac{|Q|(|Q|-1)}{2}, \forall Q \subseteq N\}$ and $t_{max} = \max_{N \in \mathcal{N}} |N|$. Then, we obtain an upper bound of B_n as follows from Theorem1,

$$B_n \leq t_{max}(n'-1) + (n - t_{max})(n'-2). \tag{4.1}$$

Our contribution is to give a better upper bound for B_n by estimating t_{max} and to give an exact value of B_n for $n \leq 36$ by constructing actual feasible HATs. In the following sections, we show Theorem15.

Theorem 15 *(1) If n' is even, we have*

$$B_n \leq t(n'-1) + (n-t)(n'-2), \tag{4.2}$$

with $t = 2(4\lfloor\frac{n'}{7}\rfloor + \lceil\frac{2(n' \mod 7)}{3}\rceil)$*, where* $n' \mod 7$ *represents the remainder and this inequality holds with equality when* $n' \leq 18$*.*
(2) If n' *is odd, we have*

$$B_n \leq (n'+1)(n'-1) + (n'-1)(n'-2), \tag{4.3}$$

and this inequality holds with equality when $n' \leq 17$*.*

To find actual feasible HATs, we employ the method of match schedule construction by [19]. Firstly lists all the HAPs that meet the conditions, and uses these HAPs to form the HAT. Then uses the integer programming model to judge whether it is feasible. The next two sections describe the enumeration method of 2c-HAPs having at least $n' - 2$ breaks and discuss the properties of these 2c-HAPs.

4.2 The total number of the 2c-HAPs having $n' - 1$ breaks

Recall that P^* is the set of 2c-HAPs with $n' - 1$ breaks. The set P^* can be divided into the two sets, P^*_H consisting of 2c-HAPs starting with home games, and P^*_A consisting of 2c-HAPs starting with away games. For any HAP, its complement, i.e., replacing home and away, keeps the number of breaks. Thus, the operation of complement gives one to one relationship among P^*_H and P^*_A. Since, as long as P^*_H can be enumerated, P^*_A can be obtained by taking complement them, the following enumeration object is only P^*_H.

Theorem 16 *The total number of the 2c-HAPs having* $n - 1$ *breaks is*
(1) n*, if* n' *is even*
(2) $n' + 1$*, if* n' *is odd*

proof (1) Let $s = (s_1, ..., s_n)$ be an H-A sequence of length n, where

$$s_{4k+i} = \begin{cases} H & i = 1,2 \\ A & i = 3,4 \end{cases}, k = 0, ..., \frac{n'}{2} - 1,$$

i.e., it is obtained by repeating $\frac{n'}{2}$ times of subsequence $(HHAA)$. In this sequence, the number of Hs and the number of As are equal. Regarding H as a home game and A as an away game, we have $break(s) = n'$. Therefore, by deleting an element of any break in the sequence, a 2c-HAP of length $n - 1$ can be obtained. Thus, n' pieces of 2c-HAPs in P^*_H can be created from the sequence s. By taking complement on these 2c-HAPs, n' pieces of 2c-HAPs in P^*_A can be obtained.

Conversely, in the H-A sequence representing a 2c-HAP in P^*_H, there is exactly one single H or A, although the remaining elements have two consecutive Hs or two consecutive as. By adding H or A after such an element, the H-A sequence s can be obtained. Therefore, all 2c-HAPs in P^*_H can be constructed by the operation removing one element in a break from s, and the number of 2c-HAPs is n. The process can be seen in Figure. 4.1.

(2) In the same way as (1), let $s = (s_1, ..., s_n)$ be the H-A sequence of length n, where

$$s_{4k+i} = \begin{cases} H & i = 1,2 \\ A & i = 3,4 \end{cases}, k = 0, ..., \frac{n'-1}{2}.$$

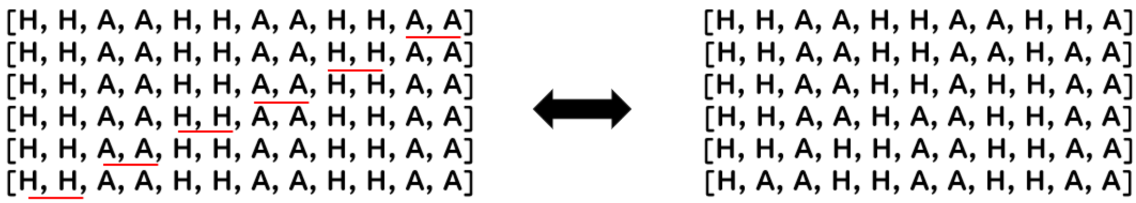


Figure 4.1: Generation Process of P_H^* for $n = 12$

It is obtained by repeating $\frac{n'-1}{2}$ times of subsequence $(HHAA)$ and by adding (HH) at the end. Then, we have $break(s) = n'$, and the number of Hs is two more than As. Therefore, a 2c-HAP of length $n-1$ can be obtained by deleting the element H in the round of break from s. By this way, $\frac{n'+1}{2}$ pieces of 2c-HAPs in P_H^* can be made, and $\frac{n'+1}{2}$ pieces of 2c-HAPs in P_A^* can be obtained by taking complement to these 2c-HAPs. We also know that all 2c-HAPs in P_H^* is obtained in this way since there is a one-to-one relationship between the break deleted from the H-A sequence s and the 2c-HAP in P_H^*. Thus, we obtain the number of 2c-HAPs, $n'+1$. The process can be seen in Figure. 4.2

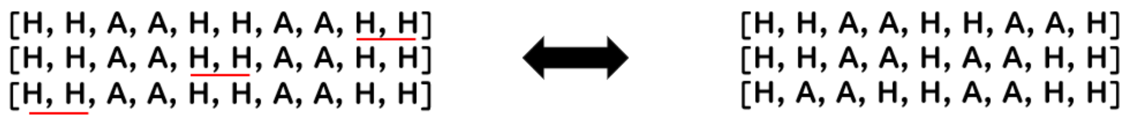


Figure 4.2: Generation Process of P_H^* for $n = 10$

From the operation obtained a 2c-HAP from an H-A sequence s discussed in the above proof, we have the following property for 2c-HAPs in P_H^*.

Corollary 17 *In any 2c-HAP in P_H^*, each game at $4k+1$ round is home and a game at $4k+3$ round is away for $k = 0, \dots, \lceil \frac{n'}{2} \rceil - 1$.*

We now discuss a property of 2c-HAPs having $n'-1$ breaks in order to evaluate t_{max}. Let p_l be a 2c-HAP obtained by deleting the $(2l)$th element in the sequence s. We can observe that, in addition to $4k+1$ round from Corollary 17, home games are assigned in p_l at $4k+2$ round for $k \leq \frac{l}{2} - 1$ and $4k+4$ round for $k \geq \frac{l}{2} - 1$.

We now consider the property of 2c-HAPs for even n'. Note that P_H^* is given by $\{p_l \mid l = 1, \dots, n'\}$ when n' is even.

Lemma 18 *Any set of 2c-HAPs containing p_l, p_{l+1}, p_{l+2} for any $l = 1, \dots, n'-2$ does not form a feasible HAT.*

proof By applying for $Q = \{p_l, p_{l+1}, p_{l+2}\}$, the LHS of Eq. (2.1) is equal to $\min\{H(Q, 2l), A(Q, 2l)\} + \min\{H(Q, 2l+2), A(Q, 2l+2)\} = 2$ (see Figure. 4.4), which is less than the RHS of Eq. (2.1), $|Q|(|Q|-1)/2 = 3$. Thus, any HAT containing Q is not feasible.

This result shows that we can select at most $\lceil \frac{2}{3}k \rceil$ 2c-HAPs from any k consecutive seriese, which derives the upper bound of t_{max}, i.e., $t_{max} \leq 2\lceil \frac{2}{3}n' \rceil$. To obtain a tighter upper bound, the following lemma shows prohibited longer patterns of 2c-HAPs for any feasible HAT.

	round										
p	1	2	3	4	5	6	7	8	9	10	11
1	**H**	A	**A**	H	**H**	A	**A**	H	**H**	A	**A**
2	**H**	H	**A**	H	**H**	A	**A**	H	**H**	A	**A**
3	**H**	H	**A**	A	**H**	A	**A**	H	**H**	A	**A**
4	**H**	H	**A**	A	**H**	H	**A**	H	**H**	A	**A**
5	**H**	H	**A**	A	**H**	H	**A**	A	**H**	A	**A**
6	**H**	H	**A**	A	**H**	H	**A**	A	**H**	H	**A**

(a)

	Round				
p	2	4	6	8	10
1	A	H	A	H	A
2	H	H	A	H	A
3	H	A	A	H	A
4	H	A	H	H	A
5	H	A	H	A	A
6	H	A	H	A	H

(b)

Figure 4.3: Delete columns holding the same H and A.

$$\begin{array}{c} \\ p_l \\ p_{l+1} \\ p_{l+2} \end{array} \begin{array}{c} \begin{array}{cccccccccc} & & & & 2l & & & & 2l+4 & \end{array} \\ \begin{pmatrix} H & H & \cdots & H & A & A & H & H & A & \cdots \\ H & H & \cdots & H & H & A & H & H & A & \cdots \\ H & H & \cdots & H & H & A & A & H & A & \cdots \end{pmatrix} \end{array}$$

Figure 4.4: Example of set of 2c-HAPs $\{p_l, p_{l+1}, p_{l+2}\}$, where l is odd.

Lemma 19 *Any set of 2c-HAPs containing each of the following sets does not form a feasible HAT.*

1. $p_l, p_{l+1}, p_{l+3}, p_{l+4}, p_{l+6}$ *for any* $l = 1, \ldots, n' - 6$
2. $p_l, p_{l+1}, p_{l+3}, p_{l+5}, p_{l+6}, p_{l+8}, p_{l+10}, p_{l+11}$ *for any* $l = 1, \ldots, n' - 11$
3. $p_l, p_{l+1}, p_{l+3}, p_{l+5}, p_{l+6}, p_{l+9}, p_{l+10}, p_{l+12}, p_{l+13}$ *for any* $l = 1, \ldots, n' - 13$

proof By applying for Q of Eq. (2.1) the set of 2c-HAPs in the statement, we check its feasibility.

1. By applying for Q, the LHS of Eq. (1) is equal to $\sum_{l \le i < l+6} \min\{H(Q, 2i), A(Q, 2i)\} = 9$ (see Figure. 4.5). It is less than the RHS of Eq. (1), $|Q|(|Q| - 1)/2 = 10$. Thus, any HAT containing Q is not feasible.

2. By applying for Q of Eq. (2.1), the LHS is 26, and the RHS is 28. Thus, any HAT containing Q is not feasible.

3. By applying for Q of Eq. (2.1), the LHS is 35, and the RHS is 36. Thus, any HAT containing Q is not feasible.

$$
\begin{array}{c}
 \\ p_l \\ p_{l+1} \\ p_{l+3} \\ p_{l+4} \\ p_{l+6}
\end{array}
\begin{array}{ccccccccccccc}
2l & & & & & & & & & & & & 2(l+6) \\
\cdots A & A & H & H & A & A & H & H & A & A & H & H & A\cdots \\
\cdots H & A & H & H & A & A & H & H & A & A & H & H & A\cdots \\
\cdots H & A & A & H & H & A & H & H & A & A & H & H & A\cdots \\
\cdots H & A & A & H & H & A & A & H & A & A & H & H & A\cdots \\
\cdots H & A & A & H & H & A & A & H & H & A & A & H & A\cdots
\end{array}
$$

Figure 4.5: Example of set of 2c-HAPs $\{p_l, p_{l+1}, p_{l+3}, p_{l+4}, p_{l+6}\}$, where l is odd.

Figure 4.6 provides a more intuitive examination of the subsets that cannot form feasible 2c-HATs. The "HAP" column enumerates all 2c-HAPs with the maximum number of breaks, and black dots represent selected HAPs. If a 2c-HAT contains a subset matching any of the four scenarios depicted in the figure, then that 2c-HAT is considered infeasible. This will help us better understand the theorem presented below.

Theorem 20 *For even n', we have $t_{max} \leq 2(4\lfloor \frac{n'}{7} \rfloor + \lceil \frac{2(n' \mod 7)}{3} \rceil)$.*

proof Let us consecutive 2c-HAPs in P_H^*, $p_1, \ldots, p_n$ be divided into blocks containing exactly seven consecutive series except for the last block that contains $(n' \mod 7)$ 2c-HAPs. Namely, each block is formed as $\{p_{7k+1}, p_{7k+2}, \ldots, p_{7k+7}\}$ for $k = 0, \ldots, \lfloor \frac{n'}{7} \rfloor$. We call a block containing seven 2c-HAPs as full. Lemma 18 shows that we can select at most five 2c-HAPs for a feasible HAT in each full block, and $\lceil \frac{2(n' \mod 7)}{3} \rceil$ 2c-HAPs from the last block.

Assume that $t_{max} > 2(4\lfloor \frac{n'}{7} \rfloor + \lceil \frac{2(n' \mod 7)}{3} \rceil)$. Then there exists a full block from which we can select five 2c-HAPs. When we select five 2c-HAPs for a feasible HAT from a full block, there is only one selection mode $p_{7k+1}, p_{7k+2}, p_{7k+4}, p_{7k+6}, p_{7k+7}$ that satisfies Lemma 18. To avoid including series prohibited by Lemma 19, only three HAPs can be selected from the preceding and following full blocks. Hence, a possible case is when five 2c-HAPs are selected from each of odd-numbered full blocks, three 2c-HAPs from each of even-numbered full blocks, and $\lfloor \frac{n'}{7} \rfloor$ is odd. Since n' is even, $(n' \mod 7)$ is not zero. If $\lfloor \frac{n'}{7} \rfloor$th block contains five 2c-HAPs, we can not choice $\lceil \frac{2(n' \mod 7)}{3} \rceil)$ 2c-HAPs containing no prohibited sequence from the last block.

We now turn to the case of odd n', where P_H^* can be represented by $\{p_{2l+1} \mid l = 0, \ldots, \frac{n'-1}{2}\}$.

Theorem 21 *For odd n', we have $P^* \in \mathcal{N}$ and $t_{max} = n' + 1$.*

proof For any subset $Q \subseteq P^*$, we will prove Eq. (2.1). Let $\widetilde{Q}$ be the set of 2c-HAPs in Q such that it complement is not contained in Q, and set $|P_H^* \cap \widetilde{Q}| = c_1$, $|P_A^* \cap \widetilde{Q}| = c_0$, and $|P_H^* \cap (Q \setminus \widetilde{Q})| = |P_A^* \cap (Q \setminus \widetilde{Q})| = c_2$. Notice that $|Q| = c_0 + c_1 + 2c_2 \leq n' + 1$. Since the number of home games is equal to the number of away games at each round in $Q \setminus \widetilde{Q}$,

(1)

HAP	●/○
p_l	●
p_{l+1}	●
p_{l+2}	●

(2)

HAP	●/○
p_l	●
p_{l+1}	●
p_{l+2}	○
p_{l+3}	●
p_{l+4}	●
p_{l+5}	○
p_{l+6}	●

(3)

HAP	●/○
p_l	●
p_{l+1}	●
p_{l+2}	○
p_{l+3}	●
p_{l+4}	○
p_{l+5}	●
p_{l+6}	●
p_{l+7}	○
p_{l+8}	●
p_{l+9}	○
p_{l+10}	●
p_{l+11}	●

(4)

HAP	●/○
p_l	●
p_{l+1}	●
p_{l+2}	○
p_{l+3}	●
p_{l+4}	○
p_{l+5}	●
p_{l+6}	●
p_{l+7}	○
p_{l+8}	○
p_{l+9}	●
p_{l+10}	●
p_{l+11}	○
p_{l+12}	●
p_{l+13}	●

Figure 4.6: If 2c-HATs contains these four partial sets, then the 2c-HATs is not feasible.

i.e., $H(Q \setminus \widetilde{Q}, r) = A(Q \setminus \widetilde{Q}, r)$ for any round r, we have

$$\begin{aligned}
&\sum_{r=1}^{2n-1} \min\{H(Q,r), A(Q,r)\} \\
&= \sum_{r=1}^{2n-1} (\min\{H(\widetilde{Q},r), A(\widetilde{Q},r)\} + H(Q \setminus \widetilde{Q}, r)) \\
&= \sum_{r=1}^{2n-1} \min\{H(\widetilde{Q},r), A(\widetilde{Q},r)\} + (2n-1)c_2.
\end{aligned}$$

Without loss of generality, we assume that $c_1 \geq c_0$. At $4k+1$ round, we have $\min\{H(\widetilde{Q}, 4k+1), A(\widetilde{Q}, 4k+1)\} = c_0$ since every games are assigned to home in P_H^* and every games are assigned to away in P_A^*. Similarly, we have $\min\{H(\widetilde{Q}, 4k+3), A(\widetilde{Q}, 4k+3)\} = c_0$ for $k = 0, \dots, \frac{n'-1}{2}$. Thus, by summing up in odd rounds, we have

$$\sum_{k=0}^{n-1} \min\{H(\widetilde{Q}, 2k+1), A(\widetilde{Q}, 2k+1)\} = n'c_0.$$

Next, we estimate $\min\{H(P_H^* \cap \widetilde{Q}, r), A(P_H^* \cap \widetilde{Q}, r)\}$ for even round r, instead of $\min\{H(\widetilde{Q}, r), A(\widetilde{Q}, r)\}$. Assume that $P_H^* \cap \widetilde{Q} = \{p_{2k_1+1}, p_{2k_2+1}, \dots, p_{2k_{c_1}+1}\}$, with $k_1 < k_2 < \cdots < k_{c_1}$.

$$\begin{aligned}
&\min\{H(P_H^* \cap \widetilde{Q}, 2k), A(P_H^* \cap \widetilde{Q}, 2k)\} \\
&= \begin{cases}
0 & k < 2k_1 + 1 \\
1 & 2k_1 + 1 \leq k < 2k_2 + 1 \\
2 & 2k_2 + 1 \leq k < 2k_3 + 1 \\
\vdots & \\
\lfloor \frac{c_1}{2} \rfloor - 1 & 2k_{\lfloor \frac{c_1}{2} \rfloor - 1} + 1 \leq k < 2k_{\lfloor \frac{c_1}{2} \rfloor} + 1 \\
\lfloor \frac{c_1}{2} \rfloor & 2k_{\lfloor \frac{c_1}{2} \rfloor} + 1 \leq k < 2k_{\lceil \frac{c_1}{2} \rceil + 1} + 1 \\
\lfloor \frac{c_1}{2} \rfloor - 1 & 2k_{\lceil \frac{c_1}{2} \rceil + 1} + 1 \leq k < 2k_{\lceil \frac{c_1}{2} \rceil + 2} + 1 \\
\vdots & \\
1 & 2k_{c_1 - 1} + 1 \leq k < 2k_{c_1} + 1 \\
0 & k \geq 2k_{c_1} + 1
\end{cases}
\end{aligned}$$

To minimize $\sum_{k=1}^{n-1} \min\{H(P_H^* \cap \widetilde{Q}, 2k), A(P_H^* \cap \widetilde{Q}, 2k)\}$, $p_{2k_{i+1}+1} = p_{2(k_i+1)+1}$ for $i = 1, \dots, c_1 - 1$. In this case, we have

$$\sum_{k=1}^{n-1} \min\{H(P_H^* \cap \widetilde{Q}, 2k), A(P_H^* \cap \widetilde{Q}, 2k)\} = 4(1 + 2 + \cdots + \frac{c_1 - 1}{2}),$$

for odd c_1 and

$$\sum_{k=1}^{n-1} \min\{H(P_H^* \cap \widetilde{Q}, 2k), A(P_H^* \cap \widetilde{Q}, 2k)\} = 4(1 + 2 + \cdots + (\frac{c_1}{2} - 1)) + 2\frac{c_1}{2},$$

for even c_1, which implies that

$$\begin{aligned}
&\sum_{k=1}^{n-1}\min\{H(\widetilde{Q},2k),A(\widetilde{Q},2k)\}\\
&\geq\sum_{k=1}^{n-1}\min\{H(P^*_H\cap\widetilde{Q},2k),A(P^*_H\cap\widetilde{Q},2k)\}\geq\frac{c_1^2-1}{2}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\sum_{r=1}^{2n-1}\min\{H(Q,r),A(Q,r)\}\\
=&nc_0+\frac{c_1^2-1}{2}+(2n-1)c_2\\
\geq&\frac{c_0+c_1+2c_2-1}{2}c_0+\frac{c_1}{2}c_0+\frac{c_1(c_1-1)}{2}\\
&+2(c_0+c_1+2c_2-1)c_2+c_1c_2\\
=&\frac{(c_0+c_1+2c_2)(c_0+c_1+2c_2-1)}{2}=\frac{|Q|(|Q|-1)}{2}.
\end{aligned}$$

Thus, we obtain the result of $P^*\in\mathcal{N}$.

These properties about t_{max} , together with Eq. (4.1), derive the tighter upper bound for B_{2n} shown in Eqs. (4.2) and (4.3) in Theorem 15.

4.3 The total number of the 2c-HAPs having $n'-2$ breaks

We next count of 2c-HAPs having $n'-2$ breaks. Let P^{**} be the set of such 2c-HAPs, which is divided into two sets, P^{**}_H consisting of 2c-HAPs starting with home games, and P^{**}_A consisting of 2c-HAPs starting with away games.

Theorem 22 *The total number of the 2c-HAPs having $n'-2$ breaks is,*

(1) $2\binom{\frac{n'}{2}+1}{2}\binom{\frac{n'}{2}}{1}+2\binom{\frac{n'}{2}+1}{3}$, *if n' is even*

(2) $4\binom{\frac{n'+1}{2}}{2}\binom{\frac{n'+1}{2}}{1}$, *if n' is odd*

proof The proof starts from the case of (2) that n' is odd. Create an H-A sequence $s=(s_1,...,s_{n+2})$ of length $n+2$, where

$$s_{4k+i}=\begin{cases}H & i=1,2\\ A & i=3,4\end{cases},k=0,...,\frac{n'-1}{2}.$$

We have $break(s)=n'+1$, and the number of consecutive H breaks and consecutive A breaks are both $\frac{n'+1}{2}$. If one (HH) and two (AA) are selected and one round in each break is deleted, it becomes a 2c-HAP with $n-2$ breaks and one more H than A. Similarly, taking two (HH) and one (AA) and deleting one round in each break, it becomes a 2c-HAP with $n-2$ breaks and one more A than H. The number of selected combinations of (HH) and (AA) is

$$\binom{\frac{n'+1}{2}}{1}\binom{\frac{n'+1}{2}}{2}+\binom{\frac{n'+1}{2}}{2}\binom{\frac{n'+1}{2}}{1}=2\binom{\frac{n'+1}{2}}{1}\binom{\frac{n'+1}{2}}{2}.$$

Thus, non-repetitive 2c-HAPs can be formed and the total number of 2c-HAPs becomes to $4\binom{\frac{n'+1}{2}}{2}\binom{\frac{n'+1}{2}}{1}$ after taking complement.

Conversely, an H-A sequence corresponding to any 2c-HAP in P_H^{**} always has three rounds where Hs or As are not consecutive. If round with H is inserted after a round with a non-consecutive H and a round with A is inserted after a round with a non-consecutive A, this becomes the sequence s with a length of $n'+2$. Therefore, every 2c-HAP is created by deleting three slots from s. Thus, we obtain the number of 2c-HAPs having $n'-2$ breaks.

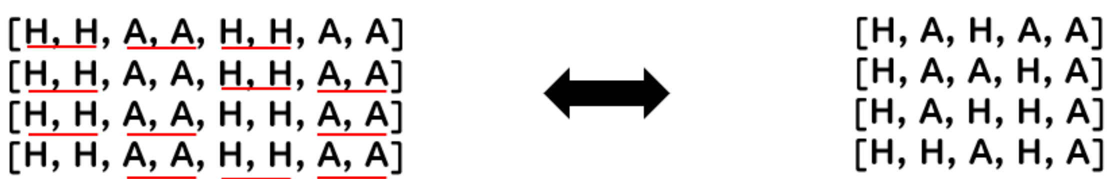


Figure 4.7: Generation Process of P_H^{**} for $n=6$

Back to (1) which creates an H-A sequence $s=(s_1,...,s_{n+2})$ of length $n+2$. That is,

$$s_{4k+i}=\begin{cases} H & i=1,2 \\ A & i=3,4 \end{cases},k=0,...,\frac{n'}{2}.$$

We have $break(s)=n'+1$, where the number of (HH) breaks is $\frac{n'}{2}+1$, and the number of (AA) breaks is $\frac{n'}{2}$. If two (HH) and one (AA) are selected and one round in each break is deleted, it becomes a 2c-HAP with $n'-2$ breaks and one more H than A. Similarly, taking three (HH) and deleting one round in each break, it becomes a 2c-HAP with $n'-2$ breaks and one more A than H. The number of selected combinations of (HH) and (AA) is

$$\binom{\frac{n'}{2}+1}{2}\binom{\frac{n'}{2}}{1}+\binom{\frac{n'}{2}+1}{3}.$$

Thus, non-repetitive 2c-HAPs can be formed and the total number of 2c-HAPs becomes to $2\binom{\frac{n'}{2}+1}{2}\binom{\frac{n'}{2}}{1}+2\binom{\frac{n'}{2}+1}{3}$ after taking complement. Since the all 2c-HAPs in P_H^{**} is obtained by this way, we obtain the number of 2c-HAPs.

The proofs of the Theorems 16 and 22 also provides a method enumerating 2c-HAP having at least $n'-2$ breaks.

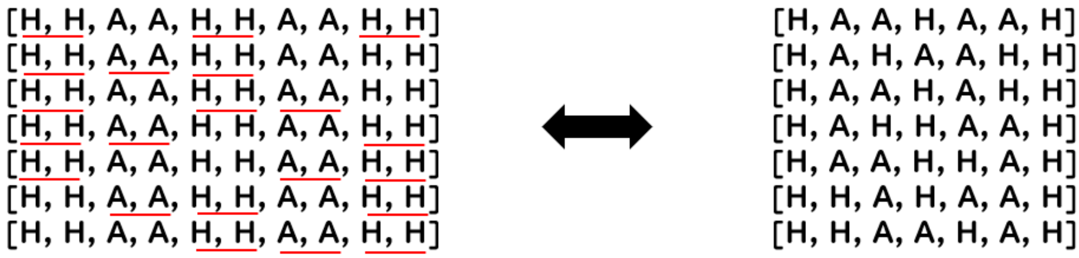


Figure 4.8: Generation Process of P_H^{**} for $n = 8$

4.4 Formulation based on integer programming

Recall that a HAT composed of 2c-HAPs is called a 2c-HAT. In this section, feasible 2c-HATs that maximizes the number of breaks are constructed by using an integer programming model. The integer programming model in this chapter refers to the integer programming model in [19] that handles minimizing the number of breaks.

Let $T = \{1, \ldots, n\}$ and $R = \{1, \ldots, n-1\}$ represent the sets of teams and rounds, respectively. The set of 2c-HAPs without any break number limitation is denoted by $\bar{P}$.

To formulate as an integer programming problem, we prepare four types of 0-1 variables for representing game schedule.

x_{ip}: When team $i \in T$ uses HAP $p \in \bar{P}$, x_{ip} is 1; otherwise, it is 0.

y_p: If HAP $p \in \bar{P}$ is used, y_p is 1; otherwise 0.

x_{ijr}: If team $i \in T$ and team $j \in T$ are playing in round $r \in R$, x_{ijr} is 1; otherwise, it is 0.

h_{ir}: If team $i \in T$ is playing at home in round $r \in R$, h_{ir} is 1; otherwise, it is 0.

We also use a parameter ξ_{pr}, which represents the match position of the corresponding HAP. If HAP $p \in \bar{P}$ is at home in round $r \in R$, it is 1; otherwise, it is 0.

The following integer programming model, called IPM, finds the maximum number of breaks over 2c-HATs.

$$\text{(IPM)} \quad \max \sum_{p \in \bar{P}} break(p) y_p \tag{4.4}$$

$$\text{s.t.} \sum_{p \in \bar{P}} x_{ip} = 1, \quad i \in T \tag{4.5}$$

$$\sum_{i \in T} x_{ip} = y_p, \quad p \in \bar{P} \tag{4.6}$$

$$x_{ijr} = x_{jir}, \quad i, j \in T, r \in R \tag{4.7}$$

$$\sum_{r \in R} x_{ijr} = 1, \quad i, j \in T, i \neq j \tag{4.8}$$

$$\sum_{j \in T \setminus \{i\}} x_{ijr} = 1, \quad i \in T, r \in R \tag{4.9}$$

$$h_{ir} = \sum_{p \in \bar{P}} \xi_{pr} x_{ip}, \quad i \in T, r \in R \tag{4.10}$$

$$x_{ijr} \leq 2 - (h_{ir} + h_{jr}), \quad i, j \in T, r \in R \tag{4.11}$$

$$x_{ijr} \leq (h_{ir} + h_{jr}), \quad i, j \in T, r \in R \tag{4.12}$$

$$\sum_{i \in T} h_{ir} = n, \quad r \in R \tag{4.13}$$

Equation (4.4) is the objective function, which expresses the maximization of the number of breaks in the whole tournament.

Constraint (4.5) stipulates that each team can only be assigned to one HAP.

Constraint (4.6) represents that each used HAP is identified.

Constraint (4.7) means that if team i and team j are playing in round r, team j is playing against i in round r.

Constraint (4.8) means that the same two teams will compete only once in all rounds, and constraint (4.9) means that each team will compete only once in a round.

If team i is assigned HAP p, then its home-away field will be positioned the same as the assigned HAP, and we restrict this with constraint (4.10).

Constraints (4.11) and (4.12) restrict the two teams that are both participating in the home game or away game in a round are not able to play against each other.

Constraint (4.13) means that the number of home games and away games is equal in any round $r \in R$.

The purpose of this IPM is to first allocate n 2c-HAPs $\in P$ to n teams. Constraints (4.5), (4.6) and (4.10) are used to complete this step. According to the HAT composed of the selected HAPs, the teams are configured to compete in each round. Constraints (4.7) (4.9), (4.11) and (4.12) play a role in this step. Finally, calculate the break number for this feasible 2c-HAT and maximize it. However, the computing power of this IPM is limited, so new algorithms need to be proposed.

4.5 Algorithm of finding maximum break 2c-HAT

When the listed 2c-HAPs are restricted to $P'(\subseteq \bar{P})$ instead of $\bar{P}$ in (IPM), we denote such a restricted problem by ($\text{IPM}_{P'}$). The optimal value of ($\text{IPM}_{P'}$) gives a lower bound for B_n. We consider the problem ($\text{IPM}_{P^* \cup P^{**}}$), i.e., the 2c-HAPs are restricted in $P^* \cup P^{**}$ having at least $n-2$ breaks in (IPM). Even with this restriction of $P^* \cup P^{**}$, the problem takes to be solved a long time when n is greater. Instead of solving ($\text{IPM}_{P^* \cup P^{**}}$) directly, we reduce the size of ($\text{IPM}_{P^* \cup P^{**}}$) by setting used 2c-HAPs in P^* as $N \in \mathcal{N}$, i.e., ($\text{IPM}_{N \cup P^{**}}$) and solve it for each $N \in \mathcal{N}$ as like the method of match schedule construction by [19].

The algorithm1 shows our procedure, which repeats checking, in the descending order of size of N, whether each subset $N \in \mathcal{N}$ can be extended to feasible 2c-HAT by using HAPs in P^{**}. As for the initial t in the algorithm, when n is odd, $t_{max} = n' + 1$, and when n' is even, the upper bound $t_{ub} = 2(4\lfloor \frac{n'}{7} \rfloor + \lceil \frac{2(n' \mod 7)}{3} \rceil)$ is used. If the current $\text{IPM}_{N \cup P^{**}}$ is feasible, then the value of t_{fsb} is t, if all the 2c-HATs composed of $N \in \mathcal{N}$ with $|N| = t$ are not feasible, then $t-1$, continue to solve for $N \in \mathcal{N}$ with $|N| = t-1$.The algorithm returns a feasible 2c-HAT with $t_{fsb}(n'-1) + (n - t_{fsb})(n'-2)$ breaks.

---

algorithm1

Initialize t such as $t \geq t_{max}$
repeat
    for $N \in \mathcal{N}$ with $|N| = t$ **do**
        Solve ($\text{IPM}_{N \cup P^{**}}$) adding constraint such that $y_p = 1$ for $p \in N$
        if the problem is feasible **then**
            $t_{fsb} \leftarrow t$
            return
        end if
    end for
    $t \leftarrow t - 1$
until $t \leq 0$
return infeasible

The integer programming problems were solved by using Gurobi 9.5.1 as an integer programming (IP) solver. All computations were run on an AMD Ryzen 5 3600 processor with 16.0 GB of RAM. The output of our algorithm are shown in Tables 4.1 and 4.2.

When $4 \leq n \leq 36$, it can be seen that t_{max} and t_{fsb} are the same, and the number of breaks obtained is the maximum number of breaks B_n.

Figure 4.9 compares the computation time of our algorithm and of solving ($\text{IPM}_{P^* \cup P^{**}}$) directly. These graphs do not draw when computing time is over 10,000 seconds for solving ($\text{IPM}_{P^* \cup P^{**}}$) by the IP solver. When n' is odd, our algorithm is also solve ($\text{IPM}_{P^* \cup P^{**}}$) since $P^* \in \mathcal{N}$. The difference of computation time is due to adding constraints $y_p = 1$ for $p \in P^*$.

Table 4.1: Computing results for odd n'

# teams (n)	6	10	14	18	22	26	30	34
$t_{max}(= n' + 1)$	4	6	8	10	12	14	16	18
t_{fsb}	4	6	8	10	12	14	16	18
# breaks	10	36	78	136	210	300	406	528

Table 4.2: Computing results for even n'

# teams (n)	4	8	12	16	20	24	28	32	36
t_{ub}	4	6	8	10	12	16	16	20	22
t_{fsb}	4	6	8	10	12	16	16	20	22
# breaks	4	22	56	106	172	256	352	468	598

When n' is even, the number of variables in (IPM$_{N \cup P^{**}}$) solved in our algorithm is reduced greatly from in (IPM$_{P^* \cup P^{**}}$), since $t_{max} < n = |P^*|$. So, the computing time for solving (IPM$_{N \cup P^{**}}$) is expected shorter than for solving (IPM$_{P^* \cup P^{**}}$). Meanwhile, since there are a lot of combinations of HAPs, the many iterations of our algorithm may be needed. In fact, however, the algorithm terminates after a few iterations. This is because that $t_{max} = t_{fsb}$ holds fortunately, and also because (IPM$_{N \cup P^{**}}$) is feasible for most of $N \in \mathcal{N}$, which fact is shown in Table 4.3, where the second row shows the number of $N \in \mathcal{N}$ with $|N| = t_{max}$ and the third row shows the number of $N \in \mathcal{N}$ with $|N| = t_{max}$ and (IPM$_{N \cup P^{**}}$) is feasible. Since the number of N achieved t_{max} is large for $n = 28$, we have not checked the number of N so that (IPM$_{N \cup P^{**}}$) is feasible.

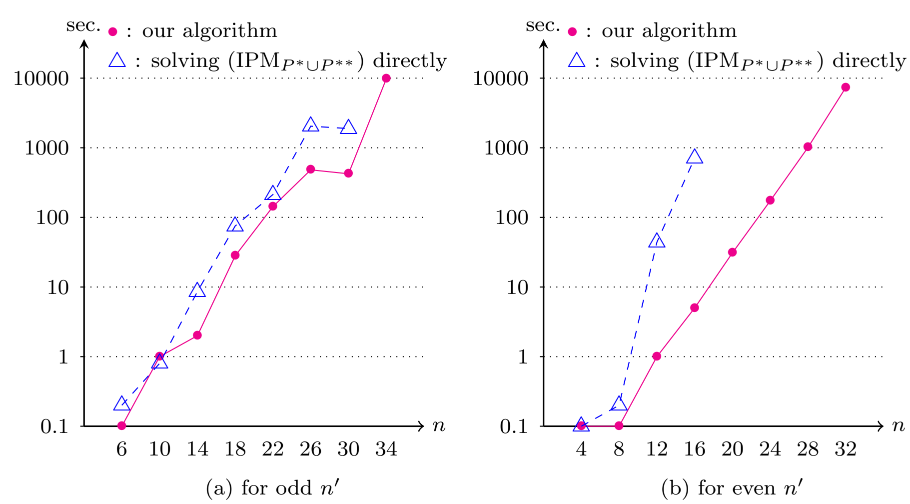


Figure 4.9: Computation time for finding a maximum break HAT

The problem of maximizing the number of breaks in sports scheduling has received relatively less attention compared to the problem of minimizing breaks, thereby highlighting its significance. In this chapter, we introduce a novel method for constructing 2c-HAPs that possess the maximum number of breaks as well as those with the second-highest number of breaks. Leveraging the intrinsic properties of HAPs, we initially derive an upper bound for $t_{\max}$ (Theorem 15). Utilizing these two types of HAPs, we propose an algorithm

Table 4.3: The number of feasible HAP conbinations in $\mathcal{N}$ for even n'

# teams (n)	4	8	12	16	20	24	28	32	36
with $\lvert N \rvert = t_{max}$	1	4	36	144	1024	1	33124	16	36
feasible	1	2	30	130	1024	1	–	16	36

to form feasible HATs that achieve the maximum number of breaks. Our calculations reveal the maximum number of breaks for up to 36 teams, and we establish that this bound holds true when the number of teams is less than 36. Through rigorous mathematical proofs presented in this chapter, we showcase the elegance and effectiveness of employing pure mathematical techniques to address challenges in sports scheduling. This approach not only advances theoretical understanding but also underscores the practical utility of mathematical solutions in optimizing tournament schedules.

So far, serving as the core of this thesis primarily presents the method of schedule generation under the constraint of break numbers. By introducing the concepts of space-sequence and isomorphic HAT, regularities are identified. Starting with HAPs, an effective algorithm is proposed to generate feasible schedules, which are then validated.

Chapter 5

Minimization of carry-over effect value under break constraints

The COE value minimization problem finds a schedule for which the COE value is minimal. Russell[18]proposed an algorithm for constructing a minimum COE value schedule when n is a power of two. Meanwhile, some research developed heuristic algorithms for the problem and improved the upper bounds of the COE value. Although constraint programming and integer programming were employed to obtain the exact optimal COE value, it seemed difficult to find the exact optimal COE values for large n in practical computational time. Recently, Günneç and Demir[41] discussed the problem of creating a fair schedule by minimizing the COE value over sr-mb-HATs. They solved it by integer programming and by a heuristic algorithm. In Chapter 5, we focus on the COE value minimization problems over mb-HATs, sr-mb-HATs, and sr-eq-HATs.

5.1 Integer programming model

Let $T = \{1, \ldots, n\}$ and $R = \{1, \ldots, n-1\}$ represent the set of teams and rounds, respectively. Recall that $\mathcal{P}$ is a set of HAPs, each of which has at most one break. For our integer programming, we prepare four types of 0-1 variables for representing game schedules and one type of 0-1 variable, and one continuous variable for representing COE values.

x_{ip}: IF team $i \in T$ uses HAP $p \in \mathcal{P}$, x_{ip} is 1; otherwise, it is 0.

y_p : If HAP $p \in \mathcal{P}$ is used, y_p is 1; otherwise 0.

x_{ijr}: If team $i \in T$ and team $j \in T$ are playing in round $r \in R$, x_{ijr} is 1; otherwise, it is 0.

h_{ir}: If team $i \in T$ is playing at home in round $r \in R$, h_{ir} is 1; otherwise, it is 0.

c_{ijr} : If team $i \in T$ gives team $j \in T$ a COE in round $r \in R$, i.e., team i and team j compete with a same other team in round r and round $r+1$,respectively, c_{ijr} is 1; otherwise, it is 0.

c_{ij} is the count that represents the sum of COEs from team $i \in T$ to team $j \in T$. We also use a parameter s_{pr}, which represents the match position of the corresponding mode. If pattern $p \in \mathcal{P}$ is at home in round $r \in R$, it is 1; otherwise, it is 0.

The following integer programming model finds the minimum COE value over mb-HATs. This integer programming model is essentially equivalent to one in Günneç and Demir[41], where round n is regarded as the first round.

$$\min \sum_{i,j \in T} c_{ij}^2 \tag{5.1}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}} x_{ip} = 1, \quad i \in T \tag{5.2}$$

$$\sum_{i \in T} x_{ip} = y_p, \quad p \in \mathcal{P} \tag{5.3}$$

$$x_{ijr} = x_{jir}, \quad i,j \in T, r \in R \tag{5.4}$$

$$x_{iir} = 0, \quad i \in T, r \in R \tag{5.5}$$

$$\sum_{r \in R} x_{ijr} = 1, \quad i,j \in T, i \neq j \tag{5.6}$$

$$\sum_{j \in T \setminus \{i\}} x_{ijr} = 1, \quad i \in T, r \in R \tag{5.7}$$

$$h_{ir} = \sum_{p \in \mathcal{P}} x_{ip} s_{pr}, \quad i \in T, r \in R \tag{5.8}$$

$$x_{ijr} \leq 2 - (h_{ir} + h_{jr}), \quad i,j \in T, r \in R \tag{5.9}$$

$$x_{ijr} \leq (h_{ir} + h_{jr}), \quad i,j \in T, r \in R \tag{5.10}$$

$$x_{ilr} + x_{jl(r+1)} - 1 \leq c_{ijr}, \quad i,j,l \in T, r \in R \tag{5.11}$$

$$\sum_{r \in R} c_{ijr} = c_{ij}, \quad i,j \in T \tag{5.12}$$

Equation (5.1) is the objective function, which expresses the minimization of the COE value in the whole tournament.

Constraint (5.2) stipulates that each team can only be assigned to one HAP.

Constraint (5.3) represents that each used HAP is identified.

Constraint (5.4) and constraint (5.5) mean that if team i and team j are playing in round r, team j is playing against i in round r, and teams cannot play against itself.

Constraint (5.6) means that the same two teams will compete only once in all rounds, and constraint (5.7) means that each team will compete only once in a round.

If team i is assigned HAP p, then its home-away venue will be positioned the same as the assigned HAP, and we restrict this with constraint (5.8).

Constraints (5.9) and (5.10) restrict the two teams that are both participating in the home game or away game in a round are not able to play against each other.

Constraint (5.11) means that after team l and team i compete in round r and team l and team j compete in round $r+1$, team j will obtain a COE from team i.

Constraint (5.12) expresses the total COE of two teams as the sum of their COE values in each round.

The integer programming model first allocates n 2c-HAPs with few breaks to n teams. Constraints (5.2), (5.3) and (5.8) complete this step. Matches are arranged for each team in every round for the HAT composed of these selected n 2c-HAPs. Constraints (5.4) (5.7), (5.9) and (5.10) complete this step. Finally, constraints (5.11) and (5.12) calculate the COE c_{ij} between every two teams, and the objective function minimizes the COE value.

Günneç and Demir[41] also employ the constraint

$$y_p = y_{\bar{p}}, \quad p \in \mathcal{P} \tag{5.13}$$

which is derived from Property 1, and

$$y_{p_1} = 1 \tag{5.14}$$

for mb-HATs. We call this problem given by Eqs. (5.1) – (5.14) as "GD model." By replacing $\mathcal{P}$ with $\mathcal{P} \setminus \{p_2, p_{n-1}\}$ in this problem, we find the minimum COE value over sr-mb-HATs. When we find over sr-eq-HATs, we use $y_{p_1} = 0$ instead of Eq. (5.14) for the problem after replacing. However, the computational power of this integer programming model is extremely limited and it needs to be further improved.

5.1.1 Modifications to the integer programming model

We now introduce additional valid inequalities for this problem and convert the original nonlinear objective function to a linear function to improve the computational speed.

M1 To eliminate symmetry when teams are assigned to patterns, we impose a rule that teams with smaller numbers are assigned to HAPs with smaller numbers, under the assumption that teams 1 to $n/2$ use HAPs in $\mathcal{P}_H$:

$$\sum_{q=1}^{r} x_{ip_q} \geq \sum_{q=1}^{r} x_{(i+1)p_q}, \quad i \in T, i+1 \leq n/2, r \in R$$

$$\sum_{q=1}^{r} x_{(i-1)\bar{p}_q} \geq \sum_{q=1}^{r} x_{i\bar{p}_q}, \quad i \in T, i-1 \geq n/2+1, r \in R.$$

$$\sum_{p \in \mathcal{P}_H} x_{ip} = 1, \quad i = 1, \ldots, n/2.$$

M2 We add the condition for feasible HAT according to Theorem 1. Let N_{iqr} be a variable representing $\min\{H(Q_{iq}, r), A(Q_{iq}, r)\}$ for a set of HAPs $Q_{iq} = \{p_k \mid k = i, i+1, \ldots, i+q-1\}$. The following two constraints stand for $N_{iqr} \le H(Q_{iq}, r)$ and $N_{iqr} \le A(Q_{iq}, r)$.

$$N_{iqr} \le \sum_{k=i}^{i+q-1} h_{kr}, \quad q = 3, \ldots, n/2, i = 1, ..., n/2 - q + 1, r \in R$$

$$N_{iqr} \le q - \sum_{k=i}^{i+q-1} h_{kr}, \quad q = 3, \ldots, n/2, i = 1, ..., n/2 - q + 1, r \in R$$

The following equation corresponds to Eq. (2.1) for Q_{iq}.

$$\sum_{r \in R} N_{iqr} \ge \frac{q(q-1)}{2}, \quad q = 3, \ldots, n/2, i = 1, ..., n/2 - q + 1.$$

M3 According to the characteristics of HAPs, two teams assigned HAPs p_k and p_h $(k < h)$ can compete during the kth round to the $(h-1)$th round. Similarly, if HAPs $\bar{p}_k$ and $\bar{p}_h$ are assigned to two teams, these two teams can compete against during the kth round to the $(h-1)$th round. If HAPs p_k and $\bar{p}_h$ $(k < h)$ or $\bar{p}_k$ and p_h are assigned to two teams, their game can be placed in rounds excluding the kth round to the $(h-1)$th round. These restrictions can be expressed by the following constraints.

$$\sum_{r=k}^{h-1} x_{ijr} \ge x_{ip_k} + x_{jp_h} - 1, \quad i, j \in T, p_k, p_h \in \mathcal{P}_H, k < h$$

$$\sum_{r=k}^{h-1} x_{ijr} \ge x_{i\bar{p}_k} + x_{j\bar{p}_h} - 1, \quad i, j \in T, p_k, p_h \in \mathcal{P}_H, k < h$$

$$\sum_{r=1}^{k-1} x_{ijr} + \sum_{r=h}^{n-1} x_{ijr} \ge x_{ip_k} + x_{j\bar{p}_h} - 1, \quad i, j \in T, p_k, p_h \in \mathcal{P}_H, k < h$$

$$\sum_{r=1}^{k-1} x_{ijr} + \sum_{r=h}^{n-1} x_{ijr} \ge x_{i\bar{p}_k} + x_{jp_h} - 1, \quad i, j \in T, p_k, p_h \in \mathcal{P}_H, k < h$$

M4 It is important to strengthen the relationship between the game and the COE. For any team i, the COE can and can only be obtained from one team in each round r. The same is true for any team j.

$$\sum_{j \in T \setminus \{i\}} c_{ijr} = 1, \quad i \in T, r \in R$$

$$\sum_{i \in T \setminus \{j\}} c_{ijr} = 1, \quad j \in T, r \in R$$

When $c_{ijr} = 1$ and $x_{ilr} = 1$, the x_{jlr+1} must be 1. Similarly, when $c_{ijr} = 1$, if $x_{jlr+1} = 1$, the x_{ilr} must be 1. These relations can be represented by the following constraints.

$$c_{ijr} + x_{ilr} - 1 \leq x_{jlr+1}, \quad i, j, l \in T, r \in R$$

$$c_{ijr} + x_{jlr+1} - 1 \leq x_{ilr}, \quad i, j, l \in T, r \in R$$

M5 To linearize the quadratic form in the objective function, we adopt continuous variabl z_{ijq} with $0 \leq z_{ijq} \leq 1$ for $i, j \in T$ and $q = 1, \ldots, n-1$, and add the constraint

$$c_{ij} = \sum_{q=1}^{n-1} z_{ijq}, \quad i, j \in T.$$

Then, objective function can be converted to piecewise linear function without using c_{ij} by

$$\min \sum_{i,j \in T} \sum_{q=1}^{n-1} (2q-1) z_{ijq}.$$

This linearized technique is discussed in Itoi et al. [52] as efficient linearization of quadratic objective functions, according to which z_{ijr} takes an integer value (1 or 0) even if it is a continuous variable.

5.1.2 Comparison of calculation results

By solving this integer programming problem, we obtain the minimum COE values over mb-HATs, over sr-mb-HATs and over sr-eq-HATs. The integer programming(IP) problem was solved by using Gurobi 9.1.0 as an integer programming solver. All computations were run on an Intel Core i5-8250U with 8.0 GB of RAM. The time limit to solve each problem was fixed to 7200 seconds. Table 5.1 compares the results of the GD model and our model that adds M1–M5 to the GD model. Variables other than x_{ip} and x_{ijr} are changed to continuous variables because the calculation will be faster with fewer integer variables. When n is 10 and the variables are integers, the solution time over sr-mb-HAT to the optimal value is 530.23 seconds, which is greater than 460.52 seconds in Table 5.1. When n is 12 and the variables are integers, the calculation result within 7200 seconds is 260, which is also greater than 258 in Table 5.1. As we observed in Theorem 13, there is no feasible sr-eq-HAT for small n. The results are shown in Table 5.1, where "C.v." stands for the obtained COE values. If the optimum COE values could not be obtained within the time limit, the gaps between the obtained upper and lower limits are shown in parentheses below the COE value. A hyphen "–" means that we could not find any feasible solution within the limited time. The computing time is shown in seconds in the column of "time." We could obtain the optimal COE values for $n = 6, 8$ and 10. In these results, valid inequalities and linearized technique M1–M5 of our model are useful to improve computing time. However, it is difficult to obtain the exact minimum COE values when n becomes slightly large, although our valid inequalities may help greatly raise its lower bounds.

Table 5.1: Comparison of obtained COE values and computing time for IP calculation. The columns of C.v. show the obtained COE values and the columns of time show computing time in seconds. If the optimal COE values could not be obtained within the time limit, 7200sec., its computing time is denoted by “TU” and the gaps between the obtained upper and lower limits are shown in parentheses below the COE value. A hyphen “–” means that we could not find any feasible solution within the limited time.

	over mb-HATS				over sr-mb-HATS				over sr-eq-HATS			
	GD model		our model		GD model		our model		GD model		our model	
n	C.v. (gap)	time (s)	C.v. (gap)	time (s)	C.v. (gap)	time (s)	C.v. (gap)	time (s)	C.v. (gap)	time (s)	C.v. (gap)	tme (s)
6	60	2.01	60	0.28	60	1.85	60	0.04	no feasible HAT			
8	100	246.53	100	6.08	100	580.00	100	1.63	no feasible HAT			
10	168	TU	168	871.7	168	TU	168	460.52	no feasible HAT			
	(69.5%)				(83.5%)							
12	394	TU	260	TU	264	TU	258	TU	no feasible HAT			
	(92.2%)		(40.8%)		(98.6%)		(36.1%)					
14	–	TU	402	TU	–	TU	382	TU	no feasible HAT			
			(61.4%)				(51.66%)					
16	–	TU	692	TU	–	TU	636	TU	–	TU	630	TU
			(69.8%)				(66.7%)				(19.9%)	

5.2 Heuristic algorithm for minimizing carry-over effect value

In this section, we focus on constructing a heuristic algorithm, which is expected to get a good solution more quickly than solving IP discussed in the previous section. Our heuristic method is the basis of the local search discussed on [41]. We employ two neighborhood: round swap and partial swap.

5.2.1 Round swap

Figure 5.1 shows the round swap without considering breaks. The COE value of schedule (a) is 120. When we swap rounds 2, 3, and 4, and swap rounds 6 and 7, it becomes schedule (b), and the COE value becomes 56. Compared with schedule (a), the COE value of schedule (b) has been greatly reduced. Therefore, it's apparent that by round swap, it is possible to find a schedule with a smaller COE value.

Team	Round						
	1	2	3	4	5	6	7
1	2	3	4	6	7	8	5
2	1	4	3	5	8	7	6
3	4	1	2	8	5	6	7
4	3	2	1	7	6	5	8
5	6	7	8	2	3	4	1
6	5	8	7	1	4	3	2
7	8	5	6	4	1	2	3
8	7	6	5	3	2	1	4

(a) The COE value is 120.

Team	Round						
	1	2	3	4	5	6	7
1	2	6	3	4	7	5	8
2	1	5	4	3	8	6	7
3	4	8	1	2	5	7	6
4	3	7	2	1	6	8	5
5	6	2	7	8	3	1	4
6	5	1	8	7	4	2	3
7	8	4	5	6	1	3	2
8	7	3	6	5	2	4	1

(b) The COE value is 56.

Figure 5.1: Comparison of COE values of two schedules.

The round swap under breaks constraints, which is used in [41], exchanges all the games in two rounds (See Figure. 5.2). After the round swap, the positions of home/away of each team will also change. Thus, it is necessary to decide whether the new schedule meets a HAT consisting of home-away of patterns with the constraints of sr-mb-HAT. Miyashiro and Mitsui[33] show polynomial time algorithms to find a HAT with $n-2$ breaks and to find an equitable HAT for a given timetable of a round-robin tournament.

Thus, we can judge whether a HAT consistent with the new schedule exists in $O(n^3)$. If the new COE value becomes lower and the constraints are met, the result of the swap is adopted, otherwise, the next swap will be carried out. Indeed, almost all results of round swap failure to construct HATs.

Team	Round 1	Round 2	Round 3	Round 4	Round 5
A	@F	@D	B	E	C
B	E	@C	@A	@D	F
C	@D	B	E	@F	@A
D	C	A	F	B	@E
E	@B	F	@C	@A	D
F	A	@E	@D	C	@B

Figure 5.2: An example of round swap.

5.2.2 Partial swap

The partial swap exchanges partial games in two rounds. We employ two types of partial swaps. The type I of partial swap does not change its HAT. According to two rounds k, k', let $G_{k,k'} = (T, E_k \cup E_{k'})$ be the directed graph with the vertex set as the team set T and the arc set $E_k \cup E_{k'}$, where $(i, j) \in E_k$ if teams i and j compete against at team j's home. Note that $G_{k,k'}$ consists of disjoint even length cycles. For a directed cycle C in $G_{k,k'}$, all games in the directed cycle C exchange among round k and k' without changing its HAT (See Figure. 5.3). This swap makes the direction of C reverse in $G_{k,k'}$. A cycle C in $G_{k,k'}$ is called alternate if C is directed when the direction of all arcs in $E_{k'}$ are reversed. All games in an alternate cycle C can also be exchanged among round k and k' without changing its HAT (See Figure. 5.4). For a cycle C not directed/alternate, all games in the cycle C exchange among round k and k' but it needs to update HAT according to it. We call this exchange as type 2 of partial swap.

The framework is shown in Algorithm 2.

In our computing, we use as an initial feasible solution the feasible solution found firstly in IP model with M1– M3.

The result is shown in Table 5.2, where RS, and PS stand for round swap and partial swap, respectively. The result of only round swaps was slightly different from [41], which implies that the results are influenced by initial feasible solutions. Combining partial swaps, the values were improved a little.

algorithm2

```
Find a feasible schedule and compute c_ij
repeat
    for k, k' ∈ R s.t. k < k' do
        exchange round k and round k'                                   ▷ round swap
        if the new COE value is smaller than the previous one and a feasible HAT associated with the new table exists then   ▷ the swap is successful
            update the schedule obtained by this swap
        end if
    end for
    for k, k' ∈ R s.t. k < k' do
        Find a cycle not alternate/directed in G_{k,k'} and swap along the cycle   ▷ partial swap type 2
        if the new COE value is smaller than the previous one and a feasible HAT associated with the new table exists then   ▷ the swap is successful
            update the schedule obtained by this swap
        end if
    end for
    for k, k' ∈ R s.t. k < k' do
        Find an alternate/directed cycle in G_{k,k'} and swap along the cycle   ▷ partial swap type 1
        if the new COE value is smaller than the previous one then   ▷ the swap is successful
            update the schedule obtained by this swap
        end if
    end for
until no successful swap occurs
```

Table 5.2: Result of the heuristic algorithm of round/partial swaps.

# teams	local search			best val. by IP
	only RS	RS+PS1	RS+PS1+PS2	
10	190	190	180	168
12	364	332	300	262
14	458	454	450	396
16	684	664	646	554
18	1042	932	932	884

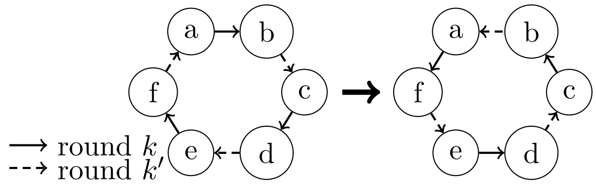


Figure 5.3: Example for type I of the partial swap along a directed cycle with 6 teams a–f

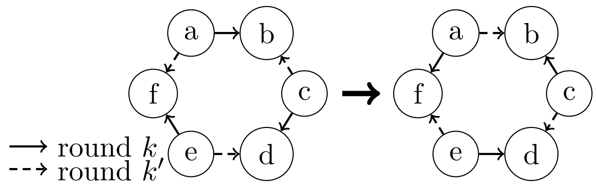


Figure 5.4: Example for type I of the partial swap along an alternate cycle with 6 teams a– f

5.3 Successive method by enumerating home-away tables

The integer programming model for the COE value minimization problem determines both an appropriate HAT and its tournament schedule. Because there are a large number of possible HATs and their tournament schedules, their computing time tends to become so long. To reduce its search space, we added M2 according to Theorem 1. The number of candidate HATs over mb-HATs was discussed in Miyashiro et al.[42]. They showed that the number of feasible mb-HATs is much less than the number of combinations that select $n/2$ HAPs from $\mathcal{P}$. Moreover, as we discuss in Section 2.1, we do not need to search for several isomorphic HATs to find the minimum COE values. Table 3.1 shows the number of candidate 2c-HATs with few breaks, which compares the number of combinations to choose HAPs, the numbers of feasible HATs shown in the columns of "# feasible", and the numbers of nonisomorphic feasible HATs shown in the columns of "# noniso". As shown in Table 3.1, the number of nonisomorphic feasible HATs is very small.

Since the number of nonisomorphic HATs is small, we adopt a method to compute minimum COE values for candidate HATs in order. To enumerate nonisomorphic HATs, we use *space-sequences*. We list up all nonnegative integer sequences of length $n/2$ with the sum of $n/2-1$ and delete a sequence if it contains a subsequence $(0,0)$, $(1,0,1,1,0,1)$, $(1,...,1,0,1,...,1,0)$ and $(0,1,...,1,0,1,...,1)$. Then, we reserve only HATs as a candidate HAT if it is lexicographically small among sequences obtained by cyclic rotation and reverse rotation from itself to avoid duplication of isomorphic HATs.

For $n \leq 18$, this enumeration can be done in a few seconds in our computer environment described in the previous subsection. We then solve the integer programming problem to find minimum COE values for each candidate HAT by using our model described in the previous subsection. Since variables x_{ip}, y_p and h_{ir} are specified according to a given HAT, we solve the problem minimizing Eq.(2) subject to Eqs.(5.4)–(5.7), (5.9)–(5.12) and M3–

algorithm3

enumerate non-isomorphic candidate HATs
set $UB = \infty$ (upper bound for the COE value)
for each candidate HAT **do**
  solve the IP model with the candidate HAT and with the constraints associated with UB
  if the objective value $< UB$ **then**
    update UB by the obtained objective value
  end if
end for

M5. Table 5.3 displays the minimum COE values for each mb-HAT by solving this integer programming problem. In order to facilitate understanding, table 5.4 is an example of a schedule with minimum COE value when n is 12 and the underlined teams play away games. From this result, we conjecture that the HAT corresponding to the space-sequence $(0, 1, 1, \ldots, 1)$ gives the minimum COE value among mb-HATs and sr-mb-HATs. If this is true, the minimum COE value over sr-mb-HATs is equivalent to the minimum COE value over mb-HATs. Moreover, in our experiment, the computing time tended to be short when a space-sequence contained numbers 2 and 3. Thus, a method for finding the minimum COE value solves the problem for HATs in lexicographically decreasing order for space-sequence, which outline is shown in Algorithm 3. We call this method as successive method.

Table 5.5 shows the result of the successive method over sr-mb-HATs together with our IP model described in the previous section and the results in Günneç and Demir[41] of their heuristic algorithm. In addition, the result for sr-eq-HAT for $n = 16$ is shown. Each result of our IP model and successive method is the tentative value when it could not finished within a given time limit of 7200 sec. When the optimal value was obtained within the limited time, an asterisk was added to the value in the table. Günneç and Demir[41] obtained a solution for $n = 18$ by their heuristic algorithm in 23.9 min in their environment. Although the computing time is approximately 5 times for $n = 18$, we could improve the COE values for every n by no more than 18. Unfortunately, despite enumerating mb-HATs, the time limit was reached while solving the problem for the first HAT corresponding to $(0, 1, 1, \cdots, 1)$ when $n \geq 12$. Nevertheless, the solution of successive method can be improved the results of Günneç and Demir[41]. If there are no restrictions on HATs, the minimum COE value is 240 for $n = 16$ (Russell[18]). Therefore, it can be seen that the minimum COE value increases by adding the restriction for HATs. Moreover, when $n = 16$, we took the tentative minimum value 526 of sr-mb-HAT as the upper bound of the successive method to calculate the unique sr-eq-HAT, and it was infeasible. So we know that when $n = 16$, the minimum COE value over sr-eq-HATs is larger than the minimum COE value over mb-HATs/sr-mb-HATs, which implies the trade-off between breaks and COE values.

Simultaneously considering break constraints and COE values is a novel and valuable research topic. In this chapter, we investigate the minimum COE values for 2c-HATs with a small number of breaks using two distinct methods. The first method employs a heuristic algorithm based on graph theory that involves rounds swap, while the second

Table 5.3: The COE values of each candidate HAT. Each candidate HAT is represented by the corresponding space-sequence.

$n = 6$		$n = 8$		$n = 10$		$n = 12$		$n = 14$		$n = 16$	
	C.v.		C.v.		C.v.		C.v.		C.v.		C.v.
(0, 1, 1)	60	(0, 1, 1, 1)	100	(0, 1, 1, 1, 1)	168	(0, 1, 1, 1, 1, 1)	258	(0, 1, 1, 1, 1, 1, 1)	382	(0, 1, 1, 1, 1, 1, 1, 1)	526
		(0, 1, 0, 2)	112	(0, 1, 1, 0, 2)	198	(0, 1, 1, 1, 0, 2)	264	(0, 1, 1, 1, 1, 0, 2)	384	(0, 1, 1, 1, 1, 1, 0, 2)	532
						(0, 1, 1, 0, 1, 2)	278	(0, 1, 1, 1, 0, 1, 2)	396	(0, 1, 1, 1, 1, 0, 1, 2)	560
						(0, 1, 0, 2, 0, 2)	296	(0, 1, 1, 0, 2, 0, 2)	412	(0, 1, 1, 1, 0, 2, 0, 2)	554
								(0, 1, 0, 2, 0, 1, 2)	422	(0, 1, 1, 1, 0, 1, 2, 1)	572
										(0, 1, 1, 1, 0, 1, 1, 2)	548
										(0, 1, 1, 0, 2, 0, 1, 2)	560
										(0, 1, 1, 0, 1, 2, 0, 2)	584
										(0, 1, 0, 2, 1, 0, 1, 2)	566
										(0, 1, 0, 2, 0, 2, 0, 2)	580
										(0, 1, 0, 2, 0, 1, 1, 2)	580
										(0, 1, 0, 2, 0, 1, 0, 3)	620

Table 5.4: The schedule table with minimum COE value when n is 12.

	Round										
n	1	2	3	4	5	6	7	8	9	10	11
1	2	3	7	4	6	5	10	9	8	11	12
2	1	4	3	6	5	7	8	11	9	12	10
3	12	1	2	5	4	9	6	8	7	10	11
4	11	2	12	1	3	8	5	6	10	9	7
5	10	12	11	3	2	1	4	7	6	8	9
6	9	11	10	2	1	12	3	4	5	7	8
7	8	9	1	10	12	2	11	5	3	6	4
8	7	10	9	12	11	4	2	3	1	5	6
9	6	7	8	11	10	3	12	1	2	4	5
10	5	8	6	7	9	11	1	12	4	3	2
11	4	6	5	9	8	10	7	2	12	1	3
12	3	5	4	8	7	6	9	10	11	2	1

Table 5.5: Comparison of the obtained COE values by our IP model, successive method, and heuristic algorithm by Günneç and Demir (2019) which is shown in the columns of GD heuristic. Asterisks (*) mean the COE values are optimal. hyphen(-) implies that a feasible solution could not obtained in a limited time.

	sr-mb-HATs			sr-eq-HATs	
n	our IP mode	successive method	GD heuristic	our IP mode	successive method
6	60*	60*			
8	100*	100*	104		
10	168*	168*	192		
12	260	258	318		
14	408	382	446		
16	638	526	626	630	620
18	-	744	944		
20	-	1172		-	1348

method introduces a new algorithm leveraging the concept that isomorphic HATs possess identical COE values. Although the heuristic algorithm for swapping rounds did not produce highly satisfactory results, identifying swappable rounds through graph theory due to break constraints in HAPs is intriguing. This suggests that there is potential for developing more effective heuristic algorithms in the future. Conversely, the successive method based on isomorphic HATs has demonstrated promising results, achieving lower COE values compared to previous studies.

Chapter 6

The travel distance minimization problem

This study aims to investigate the impact of break numbers on travel distance, employing two different measures of travel distance to examine the results.

6.1 Description

The Traveling Tournament Problem (TTP) is a challenging combinatorial optimization problem in sports scheduling that focuses on minimizing the total travel distance for teams in a round-robin tournament. This problem is particularly important for professional sports leagues, where reducing travel can significantly cut costs and improve player performance by reducing fatigue.

The TTP was first introduced by Easton et al.[32]. The original formulation involves scheduling a double round-robin tournament such that the total distance traveled by all teams is minimized, while satisfying a set of constraints. Since then, numerous researchers have proposed various models and methods to address the TTP, including both exact and heuristic approaches.

6.1.1 Mathematical Formulation

Let $T = \{1, \ldots, n\}$ and $R = \{1, \ldots, 2(n-1)\}$ represent the set of teams and rounds in a double round-robin tournament, respectively. Let d_{ij} denote the distance between the home venues of teams i and j, for $i, j \in T$. When a team plays an away game, it is assumed that the team travels from the home venue to the away court. When playing consecutive away games, the team travels directly from one away venue to the next. Each team starts the game at its home and must return to its home venue after the game.The standard TTP involves the key component: Finding a schedule specifies which teams play against each other in each round and where each game is played, and the schedule S has the minimum total travel distance across all teams

The TTP must satisfy several constraints:

- Each team begins the tournament at home and must return home after its last away game;

- No repeaters are allowed, meaning no two teams can play against each other in consecutive rounds;
- Every sequence of consecutive home games played by any team consists of at least L and at most U games; L and U are integer parameters.
- Every sequence of consecutive away games played by any team consists of at least L and at most U games;
- The sum of the total traveling distance of each team has to be minimized.

Timetable

Team	Round									
	1	2	3	4	5	6	7	8	9	10
1	5	6	3	2	4	5	6	3	2	4
2	6	4	5	1	3	6	4	5	1	3
3	4	5	1	6	2	4	5	1	6	2
4	3	2	6	5	1	3	2	6	5	1
5	1	3	2	4	6	1	3	2	4	6
6	2	1	4	3	5	2	1	4	3	5

Patterns

Team	Round									
	1	2	3	4	5	6	7	8	9	10
1	A	H	A	A	A	H	A	H	H	H
2	H	A	H	H	H	A	H	A	A	A
3	A	H	H	H	A	H	A	A	A	H
4	H	H	A	H	H	A	A	H	A	A
5	H	A	A	A	H	A	H	H	H	A
6	A	A	H	A	A	H	H	A	H	H

Figure 6.1: Timetable and patterns.

The timetable in Figure 6.1 represents a mirrored double round-robin schedule, with the numbers in the table representing the teams. The patterns is the HAT correspond to the timetable, and the number of patterns is determined by L and U. Let S denote the set of slots in the first half of the timetable $\{1, \ldots, n-1\}$. Let P denote the set of patterns. For each pattern $j \in P$ and each $s \in S$, h_{js} is defined to be 1 if round s of pattern j is 1; otherwise, h_{js} is defined to be 0. For each $t \in T$ and each $j \in P$, let d_{tj} denote the distance team t must travel if pattern j is assigned to team t. Note that the distance d_{tj} can be calculated given the timetable and the pattern. Then the Timetable Constrained Distance Minimization Problem(TCDMP) for the given timetable can be formulated as the following integer programming problem:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{j \in P} d_{tj} x_{tj} \\
\text{s.t.} \quad & \sum_{j \in P} x_{tj} = 1 \quad \forall t \in T \\
& \sum_{t \in \{i_1, i_2\}} \sum_{j \in P} h_{js} x_{tj} = 1 \quad \forall i_1, i_2 \in T, s \in S \text{ such that } i_1 < i_2 \\
& x_{tj} \in \{0, 1\} \quad \forall t \in T, j \in P.
\end{aligned}
$$

The above formulation is a specialization of the formulation introduced by Rasmussen and Trick [53] to the mirrored setting.

6.1.2 Exact Approaches and Limitations

Ribeiro and Urrutia [43] proposed an IP formulation that was able to solve small instances of the problem optimally. However, these methods often struggle with larger instances due to the exponential growth in the number of possible schedules.

For $U = 1$, it is easy to see that there is no solution for $n > 2$. In this case, the only feasible HAP is alternating home and away. There are only two such sequences (one beginning at home and one beginning away), so if there are more than 2 teams, two teams will have the same sequence. Such teams, however, cannot play each other so no round robin tournament is possible.

The classic TTP prioritizes distance minimization, thus relaxing the conditions on U. However, as mentioned earlier, too many consecutive home or away games can lead to unfair schedules. Nevertheless, an effective method to solve the minimum travel distance when $U = 2$ has yet to be found. In the TTP, the double round-robin tournament is the most commonly studied format. In a double round-robin tournament, the home and away situations of the first half of the schedule mirror that of the second half, often resulting in sequences of three consecutive home or away games. Consequently, in previous studies addressing the double round-robin tournament, the upper bound U is typically set to 3. To mitigate the occurrence of such sequences, the use of sr-HAT is required. Since Chapters 3 and 4 have already derived 2c-HATs under different break constraints, and 2c-HATs satisfy the condition of $U = 2$, we will use these 2c-HATs to examine the relationship between the number of breaks and travel distance in single round robin.

6.2 The travel distance minimization problem under maximum breaks

The concept of 2c-HATs and finding the maximum breaks over 2c-HATs has been introduced to reduce the total travel distance to execute the tournament schedule while maintaining fairness [47]. In this section, we investigate the effect of these break constraints of HATs on the travel distance.

The problem of finding a tournament schedule minimizing the total travel distance is called a traveling tournament problem [32]. Although the traveling tournament problem, in general, assumes double round robin tournaments, we deal with the single round robin considered in the previous section. The total travel distance considers the distance to go to the game from the home field in the first round and to return to the home field after the tournament. Table 6.1 and Figure 6.2 show the movement of teams 1 in a schedule with 6 teams, and the travel process is $1 \to 4 \to 2 \to 1 \to 6 \to 1$.

Table 6.1: The schedule of Team 1

Round	1	2	3	4	5
Opponent	4	2	3	5	6
Home/Away	A	A	H	H	A

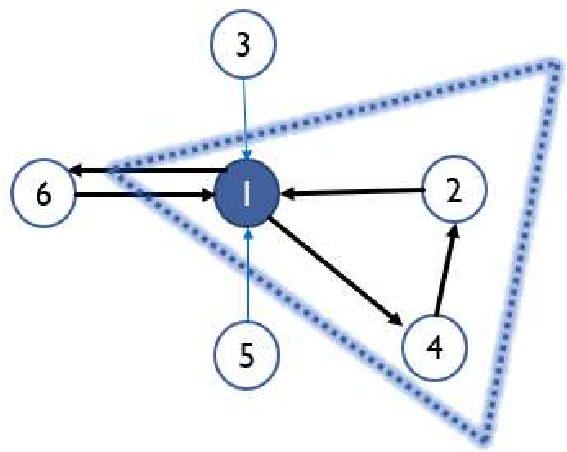


Figure 6.2: The travel process of team 1.

To clarify the influence of the number of breaks, the distance between the home fields of each pair of teams has previously been calculated by both the circular distance and the linear distance that have regularity [37]. In the circular distance, the distance d_{ij} from team i's home field to team j's home field is given by $\min\{i-j, j-i+2n\}$ for $i > j$. In linear distance, d_{ij} is given by $|i-j|$. Examples of both two distance for $n = 8$ are shown in the Figure 6.3 and Figure 6.4.

6.2.1 The integer programming model for the travel distance of a single round robin

The traveling tournament problem for a single round-robin tournament can be formulated as an integer programming problem by using two kinds of binary variables: $\tilde{x}_{ijr}$ represents whether team i plays on team j's home field in round r; z_{ijr} represents whether a team goes from team i's home to team j's home just before round r. We recall that h_{ir} is

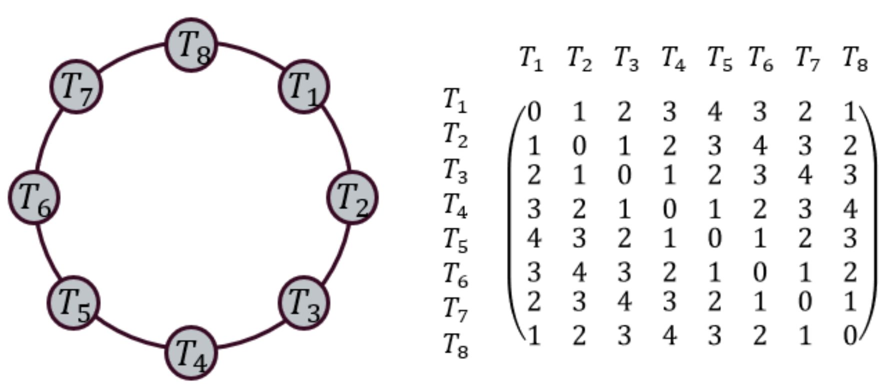
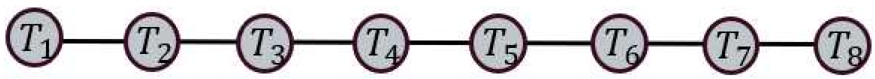


Figure 6.3: The circle distance between each pair of teams.



	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
T_1	0	1	2	3	4	5	6	7
T_2	1	0	1	2	3	4	5	6
T_3	2	1	0	1	2	3	4	5
T_4	3	2	1	0	1	2	3	4
T_5	4	3	2	1	0	1	2	3
T_6	5	4	3	2	1	0	1	2
T_7	6	5	4	3	2	1	0	1
T_8	7	6	5	4	3	2	1	0

Figure 6.4: The linear distance between each pair of teams.

also a binary variable representing whether a team i plays its home field in round r. For convenience of notation, we give $h_{i0} = h_{i(n)} = 1$ for any $i \in T$.

$$\min \sum_{\substack{i,j \in T \\ i \neq j}} \sum_{r \in R \cup \{n\}} d_{ij} z_{ijr} \tag{6.1}$$

$$\text{s.t.} \sum_{j \in T \setminus \{i\}} (\tilde{x}_{ijr} + \tilde{x}_{jir}) = 1, \quad i \in T, r \in R \tag{6.2}$$

$$\sum_{r \in R} (\tilde{x}_{ijr} + \tilde{x}_{jir}) = 1, \quad i, j \in T, i \neq j \tag{6.3}$$

$$\sum_{j \in T \setminus \{i\}} \tilde{x}_{jir} = h_{ir}, \quad i \in T, r \in R \tag{6.4}$$

$$\tilde{x}_{li(r-1)} + \tilde{x}_{ljr} - 1 \leq z_{ijr}, \quad i, j, l \in T, r \in R, r \neq 1 \tag{6.5}$$

$$h_{i(r-1)} + \tilde{x}_{ijr} - 1 \leq z_{ijr}, \quad i, j \in T, r \in R \tag{6.6}$$

$$\tilde{x}_{ji(r-1)} + h_{jr} - 1 \leq z_{ijr}, \quad i, j \in T, r \in R \cup \{2n\}, r \neq 1 \tag{6.7}$$

Constraint (6.2) means that every team plays exactly one game in every round.

Constraint (6.3) means that every pair of two teams plays one game during the tournament.

Constraint (6.4) stipulates the consistency of the home/away game.

Constraints (6.5)–(6.7) describe the travel of the teams, whether z_{ijr} is 1 when there is a team l who plays in i's home in round $r-1$ and j's home in round r, when team i plays a home game in round $r-1$ and in j's home in round r, and when teams j play at i's home in round $r-1$ and team j plays at home in round r.

When the home-away assignment is restricted to be according to 2c-HATs, we need additional constraints

$$h_{ir} + h_{i(r+1)} + h_{i(r+2)} \leq 2, \quad \forall i \in T, \forall r \in R, r \neq n-2, n-1$$

for Condition 1 and

$$n' - 2 \leq \sum_{r \in R} h_{ir} \leq n' - 1, \quad \forall i \in T$$

for Condition 2.

When the number of breaks of 2c-HATs is restricted, we assign teams to the given HAPs satisfying the break constraint to minimize the total travel distance rather than adding constraints that express the number of breaks. The minimum value among the candidate HATs is then found. For 2c-HATs with maximum breaks, we solve the integer model together with $\mathrm{IPM}_{N \cup P^{**}}$ for $N \in \mathcal{N}_{fsb}$. Instead of (6.2), we introduce constraints

$$\sum_{p \in N \cup P^{**}} x_{ip} = 1, \quad i \in T$$

$$\sum_{i \in T} x_{ip} = 1, \quad p \in N$$

$$h_{ir} = \sum_{p \in N \cup P^{**}} \xi_{pr} x_{ip}, \quad i \in T, r \in R.$$

In addition, in contrast to (6.3), we add

$$\sum_{j \in T \setminus \{i\}} \tilde{x}_{ijr} = 1 - h_{ir}, \quad i \in T, r \in R$$

Let $d(N \cup P^{**})$ be the minimum travel distance for $N \cup P^{**}$. Then, we compute $\min_{N \in \mathcal{N}_{fsb}} d(N \cup P^{**})$ by solving the integer programming problem repeatedly.

When we solve the traveling tournament problem on 2c-HATs with minimum breaks, the candidate HATs can be obtained by applying the enumerating method in [51]. In [51], the method enumerating all nonisomorphic HATs having the minimum breaks was demonstrated, where HATs obtained by cyclic rotations of rounds and by inverse rotation of rounds in a HAT were called isomorphic. In the traveling tournament problem, however, the result might be different even for isomorphic HATs. Let $\tilde{\mathcal{N}}$ be the set of candidate HATs obtained by cyclic rotation of rounds from every nonisomorphic HAT enumerating by the method in [51]. Then, we calculate $\min_{N \in \tilde{\mathcal{N}}} d(N)$, where $d(N)$ is the optimal value of the integer model replacing $N \cup P^{**}$ by N.

6.2.2 Numerical results and investigation (the advantage of 2c-HAT in moving distance)

Table 6.2 compares the minimum travel distances for three cases under 2c-HATs: the 2c-HAT without any break number constraints (2c-HAT), the 2c-HAT with the maximum breaks (max bre.), and the 2c-HAT with the minimum breaks, i.e., $n-2$ breaks (min bre.). The number with an asterisk is the optimal solution. When n is 10, although the optimal solution is not obtained, the lower bound of 2c-HAT with the minimum breaks is higher than the tentative solution of 2c-HAT with the minimum breaks during the solution

Table 6.2: Minimum travel distance under 2c-HAT with difference breaks

	circular distance			linear distance		
n	2c-HAT	max bre.	min bre.	2c-HAT	max bre.	min bre.
6	40*	40*	46*	48*	48*	56*
8	84*	84*	110*	104*	104*	138*
10	168	168	220	210	210	256

process. Although results are available for a small number of teams to compare the exact minimum distances, maximizing the number of breaks helps minimize the travel distances.

The solution to the Traveling Tournament Problem is highly complex. From previous studies and our computational experiments, we understand that the optimal solution can only be found when the number of teams is very small. Therefore, identifying a HAT initially can enhance efficiency slightly. To achieve an optimal solution, the classic TTP problem often relaxes the restrictions on breaks, sometimes permitting three or more consecutive home or away games, which might compromise fairness.

In this chapter, we primarily focus on 2c-HATs, which enforce strict break requirements. We not only investigate the circular distance, which is more commonly utilized in previous studies, but also incorporate linear distance to examine 2c-HATs comprehensively.

Chapter 7

Conclusion and future work

This study studies the fair round-robin tournament schedule from three aspects, which are the creation of feasible HAT under break constraints, the minimization of the COE value under break constraints, and the impact of the number of breaks on the moving distance. The conclusion in the three directions and future prospects will be discussed below.

First of all, the concepts and relationships of break, HAP and HAT are introduced. Whether it is the HAT with the smallest number of breaks or the HAT with the largest number of breaks, HAPs that meet the conditions for the number of breaks are first constructed, and then the HAT is composed of HAPs and its feasibility is judged. For the case where the number of breaks is the smallest, all HAPs with only one break or no break are enumerated, and the composed HATs are divided into three categories: mb-HAT, sr-HAT and eq-HAT. The sr-HAT can also be divided into sr mb-HAT and sr-eq-HAT in detail. By proposing space-sequence, we proposed a method to find feasible HATs and classify them more quickly and classified the teams under 18. From the results, we learned that sr-mb-HAT only exists when the number of teams is 16. On the other hand, in order to maximize the number of breaks, we first constructed two types of HAPs, including HAP set P^* with the largest number of breaks and HAP set P^{**} with the second largest number of breaks. We combined the two types of HAPs to find the largest number of HAPs in P^* that can be contained in the HAT which means the upper bound of t_{max}. we built an integer programming model and proposed the algorithm using iteratively the model. The algorithm found permissible 2c-HATs for up to 36 teams with maximum breaks. Many basic theories are proved in chapter 2, which greatly facilitates the production of feasible HAT under the break number constraint. It is a future work to find the 2c-HAT with the maximum number of breaks for a larger number of teams and to determine whether our proposed upper bound on the number of breaks is tight when n is an even number.

Secondly, we dealt with the COE value minimization problem under restriction of breaks, which is discussed in Günneç and Demir's[41]. In Chapter 2, the concept of isomorphic HAT was proposed. Since when calculating COE, the first round is used to calculate the COE of the last round, we can regard it as a cycle. The concept of isomorphic HAT greatly reduces the time to calculate the minimum COE value. The existing integer programming model is very limited in calculating the COE value. While trying the heuristic algorithm (partial transformation), we made six modifications to the integer programming model and confirmed its effect with data calculation results. And the COE values of isomorphic HATs are the same. Therefor, when calculating the COE value, we only need to calculate one of the isomorphic HATs. Since the HATs have been classified and the heterogeneous HATs have been found in Chapter 2, we based on the improved integer programming The model proposes an algorithm for heterogeneous HATs and obtains COE value results for less than 20 teams. This method succeeded in improving the COE values obtained by Günneç and Demir[41]. Future work will improve the COE value calculation of each HAT. If our conjecture that space-sequence $(0, 1, 1, \cdots, 1)$ achieves the minimum COE value is true, The COE value minimization problem over mb-HAT/sr-mb-HAT can be solved by finding the minimum COE value for only that HAT. Therefore, it is expected that it will efficiently solve the minimum COE value for the HAT corresponding to $(0, 1, 1, \cdots, 1)$. In addition, without the break number restriction, more heuristic algorithms can be considered, and research in this area is also need to be studied.

Finally, the travel distance of the HAT based on different break number limits is calculated. By using the two different distance, it can be seen that the HAT with the largest number of breaks does have the smallest moving distance. However, there are too few teams to be counted. It is worth looking forward to more teams being counted, and more different movement distances can be utilized.

Acknowledgements

Completing this PhD thesis has been a long journey, and I am deeply grateful for the support and encouragement I have received from numerous individuals along the way. It is with immense gratitude that I acknowledge their contributions.

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Maiko Shigeno. Your unwavering support, insightful guidance, and constant encouragement have been invaluable throughout my doctoral studies. When I was confused and wanted to give up countless times, you didn't give up on me. Your expertise and dedication have greatly influenced my academic development, and your belief in my potential has been a constant source of motivation.

I am also grateful to the members of my dissertation committee, Professor Ilic, Professor Hachimori, Professor Miao and Professor Sano, for their valuable feedback and constructive criticism. Your suggestions and insights have significantly improved the quality of my research and writing.

A special thank you goes to my colleagues and fellow graduate students in the Shigeno Lab. Your camaraderie, collaboration, and the many stimulating discussions have enriched my academic experience and made the journey enjoyable. I am particularly thankful to Mr.Ma for helping me find the sports scheduling when I was confused, as well as other members of the sports scheduling team. It was very happy to study with you.

I would also like to acknowledge the administrative and technical staff at University of Tsukuba, especially Mrs.Kawai. Your assistance with the various logistical aspects of my research has been greatly appreciated.

My heartfelt thanks go to my family for their unwavering support and encouragement. To my parents, thank you for your endless love, patience, and for believing in me even when I doubted myself. Your constant support, understanding, and sacrifices have been instrumental in helping me reach this milestone. I could not have done this without you.

Finally, I am grateful to JST, University of Tsukuba and JSPS KAKENHI for Scientific Research (B) for providing the financial support that made this research possible.

This thesis is dedicated to all those who have helped and inspired me throughout this journey. Thank you.

Bibliography

[1] R. A. Melo, S. Urrutia, and C. C. Ribeiro. The traveling tournament problem with predefined venues. *Journal of Scheduling*, 12:607–622, 2009.

[2] D. Briskorn. Feasibility of home-away-pattern sets for round-robin tournaments. *Operations Research Letters*, 36(3):283–384, 2008.

[3] T. Briskorn and A. Drexl. IP models for round robin tournaments. *Computers & Operations Research*, 36(3):837–852, 2009.

[4] T. F. Noronha, C. C. Ribeiro, G. Duran, S. Souyris, and A. Weintraub. A branch-and-cut algorithm for scheduling the highly-constrained chilean soccer tournament. In *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, pages 174–186. Springer, Berlin, 2007.

[5] F. Della Croce and D. Oliveri. Scheduling the Italian football league: an ILP-based approach. *Computers & Operations Research*, 33(7):1963–1974, 2006.

[6] T. Bartsch, A. Drexl, and S. Kroger. Scheduling the professional soccer leagues of Austria and Germany. *Computers & Operations Research*, 33(7):1907–1937, 2006.

[7] M. Henz. Constraint-based round robin tournament planning. In D. de Schreye, editor, *Proceedings of the international conference on logic programming*, pages 545–557, Las Cruces, New Mexico, 1999. MIT Press.

[8] K. Marriott and P. Stuckey. *Programming with constraints.* The MIT Press, Cambridge, 1998.

[9] J. C. Regin. Minimization of the number of breaks in sports scheduling problems using constraint programming. *Constraint Programming and Large Scale Discrete Optimization*, 57:115–130, 2001.

[10] M. Henz, T. Müller, and S. Thiel. Global constraints for round robin tournament scheduling. *European Journal of Operational Research*, 153(1):92–101, 2004.

[11] L. Castillo, D. Borrajo, M. Salido, and A. Oddi. *Planning, scheduling and constraints satisfaction: From theory to practice*. IOS Press, 2005.

[12] R. V. Rasmussen and M. A. Trick. A Benders approach for the constrained minimum break problem. *European Journal of Operational Research*, 177(1):2198–213, 2007.

[13] A. Lim, B. Rodrigues, and X. Zhang. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174:1459–1478, 2006.

[14] C. C. Ribeiro and S. Urrutia. Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179:775–787, 2007.

[15] A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. A simulated annealing approach to the traveling tournament problem. In *Proceedings of CPAIOR'03*, 2003.

[16] D. Costa. An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR*, 33(3):161–178, 1995.

[17] D. De Werra. Geography, games and graphs. *Discrete Applied Mathematics*, 2(4):327–337, 1980.

[18] K. G. Russel. Balancing carry-over effects in round robin tournaments. *Biometrika*, 67(1):127–131, 1980.

[19] G. L. Nemhauser and M . A. Trick. Scheduling a major college basketball conference. 46(1):1–8, 1998.

[20] J. C. Bean and J. R. Birge. Reductions for degeneracy in mathematical programming models of the traveling salesman problem. *Operations Research*, 28(5):1100–1119, 1980.

[21] G. Bao. Modeling and solving the NBA scheduling problem. *European Journal of Operational Research*, 194(1):41–56, 2009.

[22] C. Fleurent and J. Ferland. Allocating games for the nhl using integer programming. *Operations Research*, 41(6):958–968, 1993.

[23] M. Karpovich C. T. Craig, T. Hartman and S. Knust. A multilevel search algorithm for the NHL scheduling problem. *Annals of Operations Research*, 171(1):1–20, 2009.

[24] S. Knust. Classification of amateur sports league scheduling problems. *European Journal of Operational Research*, 205(3):582–589, 2010.

[25] R. J. Willis and M. J. Terrill. Scheduling the Australian cricket tour. *Journal of the Operational Research Society*, 45(2):191–198, 1994.

[26] M. B. Wright. Timetabling county cricket fixtures using a formulation of the traveling tournament problem. *Journal of the Operational Research Society*, 45(5):543–554, 1994.

[27] M. B. Wright. Sports scheduling: Issues and applications. *Annals of Operations Research*, 108(1):201–204, 2005.

[28] G. Kendall, S.Knust, C.Ribeiro, and S.Urrutia. Scheduling in sports: An annotated bibliography. *Computers and Operations Research*, 37(1):1–19, 2010.

[29] C. C. Ribeiro. Sports scheduling: Problems and applications. *International Transactions in Operational Research*, 193(1-2):201–226, 2012.

[30] G.Durán. Sports scheduling and other topics in sports analytics: a survey with special reference to latin america. *TOP*, 29:125–155, 2021.

[31] D. Van Bulck, D. Goossens, J. Sch onberger, and M. Guajardo. Robinx: A three-field classification and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, 280(2):568–580, 2020.

[32] K. Easton, G. Nemhauser, and M. Trick. The traveling tournament problem: description and benchmarks. *Proceedings of the Seventh International Conference on Principles and Practice of Constraint Programmings*, pages 580–584, 2001.

[33] R. Miyashiro and T. Matsui. A polynomial-time algorithm to find an equitable home-away assignment. *Operations Research Letters*, 3(3):235–241, 2005.

[34] D. Goossens and F. Spieksma. Soccer schedules in Europe: An overview. *Journal of Scheduling*, 14(5):641–651, 2012.

[35] I. Anderson. Balancing carry-over effects in round robin tournaments. In C. Rowley F. C. Holroyd, K. A. S. Quinn and B. S. Webb, editors, *Combinatorial Designs and Their Applications, CRC Research Notes in Mathematics Chapman Hall/CRC*, pages 1–16, 1999.

[36] M. Beintema, J. Bonn, R. Fitzgerald, and J. Yucas. Orderings of finite fields and balanced tournaments. *Ars Combinatoria*, 49:41–48, 1998.

[37] M. A. rick. A schedule-then-break approach to sports timetabling. In E. Burke and W. Erben, editors, *Practice and theory of automated timetabling III, Springer-Verlag, Berlin*, pages 324–343, 2000.

[38] R. Miyashiro and T. Matsui. Minimizing the carry-over effects value in a round robin tournament. *Proceedings of the 6th international conference on the practice and theory of automated timetabling, Brno*, pages 460–463, 2006.

[39] M. P. Kidd. A tabu-search for minimising the carry-over effects value of a round-robin tournament. *Journal of the Operations Research Society of South Africa*, 26:125–141, 2010.

[40] Y. Cao, W. Wu, and M. Yagiura. A metaheuristic algorithm for the carry-over effect value minimization problem in round-robin tournaments. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 16(4):JAMDSM0042, 2022.

[41] D. Günneç and E. Demir. Fair-fixture: minimizing carry-over effects in football leagues. *Journal of Industrial and Management Optimization*, 15(4):1565–1577, 2019.

[42] R. Miyashiro, H. Iwasaki, and T. Matsui. Characterizing feasible pattern sets with a minimum number of breaks. *Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science*, 2740:78–99, 2003.

[43] S. Urrutia and C. C. Ribeiro. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Applied Mathematics*, 154(13):1932–1938, 2006.

[44] Y. Fujiwara, M. Matsui, and T. Matsumoto. Constructive methods for the round-robin tournament problem. *Journal of the Operational Research Society*, 51(11):1264–1272, 2000.

[45] S. Irnich. A new branch-and-price algorithm for the traveling tournament problem. Technical Report OR-01-2009, RWTH Aachen, 2009.

[46] D. Briskorn and A. Drexl. A branch-and-price algorithm for scheduling sport leagues. *Journal of the Operational Research Society*, 60:84–93, 2009.

[47] R. A. Russell and J. M. Y. Leung. Devising a cost effective schedule for a baseball league. 42(4):614–625, 1994.

[48] A. Suzuka, R. Miyashiro, A. Yoshise, and T.Matsui. The home-away assignment problems and break minimization/maximization problems in sports scheduling. *Pacific Journal of Optimization*, 3:113–133, 1994.

[49] D. Van Bulck and D. Goossens. First-break-heuristically-schedule: Constructing highly-constrained sports timetables. *Operation Research Letters*, 51(3):326–331, 2023.

[50] M. Elf, M. Júnger, and G. Rinaldi. Minimizing breaks by maximizing cuts. *Operations Research Letters*, 36(5):343–349, 2003.

[51] F. Xue, H. Ma, and M. Shigeno. Minimizing carry-over effect values under restricted breaks for round-robin tournament. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 16(4):JAMDSM0040, 2022.

[52] M. Itoi, W. Wu, and A. Ikegami. Resource balancing for mixed integer programming problems. *Proceedings of the Scheduling Symposium*, pages 84–89, 2019.

[53] R.V. Rasmussen and M.A. Trick. The timetable constrained distance minimization problem. In J.C. Beck and B.M. Smith, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3990 of *Lecture Notes in Computer Science*, pages 167–181. Springer, Berlin, Heidelberg, 2006.