Grouping and scheduling multiple sports leagues: an integrated approach

ABSTRACT

This paper introduces the multi-league grouping and scheduling problem, which integrates the sport teams grouping problem and the multi-league sports scheduling problem. The problem consists of grouping a number of teams from a set of clubs into leagues and determining a round robin schedule for each league. It involves two possibly conflicting objectives, namely minimizing the total distance travelled by all teams and minimizing the total venue capacity violations over all clubs. We formulate this problem as a bi-objective mixed-integer programming model. Given the NP-hardness of the sport teams grouping problem, the integrated problem is particularly challenging. Hence, we design a two-layer constructive heuristic targeted at efficiently producing an approximation of the Pareto set. This algorithm uses simulated annealing on the outer layer and an integer programming model on the inner layer. We further develop a speed-up version where the inner layer is heuristically solved. We develop a series of problem instances, including one based on real-life data from the Royal Belgian Football Association, with a size exceeding what has been explored in the existing literature. In a computational study, we compare our algorithms with an epsilon-constraint method and evaluate their results using various multi-objective solution quality metrics.

KEYWORDS

OR in sports; sport teams grouping; multi-league scheduling; round robin; different league sizes; bi-objective optimization

1. Introduction

Amateur and youth sports offer non-professional players the opportunity to work out and develop athletic skills. The scale of these competitions is huge, involving hundreds of clubs, delegating thousands of teams distributed over hundreds of leagues (see e.g. Davari, Goossens, Beliën, Lambers, and Spieksma (2020); Li, Davari, and Goossens (2023). Surprisingly, even though the vast majority of sport events take place in an amateur context, only a small fraction of the sports scheduling literature deals with non-professional sports, leading M. B. Wright (2009) to the conclusion that not nearly enough has been done for amateur sport in the past 50 years of research.

Considerable efforts are required to organize youth and amateur leagues. Teams are to be grouped into leagues, a problem known as the *sport teams grouping problem* (STGP), or the sport teams realignment problem. Besides obvious criteria such as age, gender and team strength, leagues are typically composed based on travel distance. Indeed, travel distance has a potentially deleterious impact on physical performance of players (Bean & Birge, 1980; Fowler, Duffield, & Vaile, 2015), but a longer travel distance also increases carbon dioxide (CO_2) emissions and transportation costs. In the setting considered in this paper, each team travels to its opponents' venue and back home with every away game. Note that this differs from the approach where teams make away trips without returning home, which is the setting adopted in the well-known traveling tournament problem (see e.g. Easton, Nemhauser, and Trick (2001)), and common in (professional) sports in the US.

	Rounds									
	1	2	3	4	5	6	7	8	9	10
t_1	A	H	A	H	A	H	A	H	A	H
t_2	A	H	H	A	H	H	A	A	H	A
t_3	H	A	A	H	A	A	H	H	A	H
t_4	A	H	A	H	H	H	A	H	A	A
t_5	H	A	H	A	A	A	H	A	H	H
t_6	H	A	H	A	H	A	H	A	H	A

Figure 1.: A HAP set for a 2RR tournament consisting of six teams (t_1 to t_6).

Another challenge is setting up schedules for these leagues. Each league is usually played according to a so-called double round robin format, where each team faces each other team twice, once at its home venue, and once at the venue of the opponent. For any team, matches played at its home venue should be hosted by its club. All teams from the same club share the same infrastructure. This can create a capacity problem for clubs, since the maximum number of matches each club can host simultaneously at each round (i.e. day, weekend) is restricted by its venue capacity. Since most clubs have multiple teams competing in different leagues, the venue capacity issue creates dependencies between the leagues, such that scheduling them one by one may be suboptimal. Determining for each team when it plays a home game, while minimizing the clubs' venue capacity violations, is known as the *multi-league sports scheduling problem* (MLSP). This is tackled by assigning each team to a so-called Home-Away Pattern (HAP), which determines for each round whether that team plays a home game (H) or an away game (A). Figure 1 illustrates a set of HAPs for a double round robin (2RR) with six teams. Given the venue capacity of each club, determining the HAP for each team is enough to fully determine the capacity violations. While this still allows multiple schedules, these schedules all have the same travel distance and venue capacity violations. Which of these schedules is adopted is typically not considered to be important in youth and amateur sports (see e.g. Davari et al. (2020)), and will therefore not be discussed in this paper.

The reason that so far both problems have been dealt with separately is likely their computational complexity. The STGP generalizes the clique partitioning problem, and is, consequently, an NP-hard problem (Toffolo, Christiaens, Spieksma, & Vanden Berghe, 2019). The complexity of the MSLP is still open: while a special case with leagues of the same even size has been shown to be solvable in polynomial time, some straightforward generalizations are NP-hard (Davari et al., 2020). Moreover, it is common practice to solve the sport teams grouping problem first, and only to focus on the scheduling decision later when the league composition has been publicly announced. However, some league compositions may allow a schedule where few clubs face a venue capacity issue, while others may be more problematic. From personal communication with several sports associations in Flanders, we learned that capacity violations are at least as important as travel distance, which makes the current hierarchical approach questionable.

In order to explore how the league composition is inextricably linked to the quality of the schedules, this paper integrates both decision problems as the *multi-league grouping and scheduling problem* (MLGSP). This problem determines for each team to which league it should be assigned, and according to which HAP it should play. Since minimizing the total travel distance and the total capacity violations could be conflicting objectives, we investigate their trade-off by means of establishing a so-called Pareto front. A Pareto front is a set of nondominated solutions, i.e. no objective can be improved without degrading at some other objective (Steuer, 1986).

We do not aim to contribute methodologically to the field of multi-objective optimization, but rather to further introduce this approach in sports scheduling. We integrate the STGP and the MSLP, by introducing the bi-objective MLGSP (Section 3.1), where one objective relates to grouping teams into leagues (i.e., total distance travelled) and the other to scheduling leagues (i.e., total venue capacity violations). While we outlined this idea earlier in an extended abstract (Li & Goossens, 2022), we now study the trade-offs between both objectives. We also provide a mathematical formulation (Section 3.2), and demonstrate that both objectives are not conflicting under certain conditions. Furthermore, we describe a two-layer constructive heuristic targeted at efficiently producing an approximation of the Pareto front (Section 4). This algorithm comprises a simulated annealing heuristic on the outer layer and an integer programming model on the inner layer. Moreover, we develop a speed-up version where the inner-layer is heuristically solved. Finally, we offer a computational study based on a set of problem instances involving over 2,000 teams (Section 5). The size of these instances is much larger and more representative of amateur sports than what is typically tackled in sport teams grouping problems in the literature (e.g. Toffolo et al. (2019) consider less than 200 teams). One instance is (partially) based on real-life data from the Royal Belgian Football Association. We compare our algorithms with an ϵ-constraint method on small-scale instances, and assess their results using several multi-objective solution quality metrics for larger instances.

2. Related literature

This section offers a brief discussion of multi-objective optimization problems in sports (Section 2.1), followed by a review of the sport teams grouping and the multi-league scheduling problem (Section 2.2).

2.1. Multi-objective optimization in sports

Although most problems in sports scheduling face multiple stakeholders, each with their wishes and objectives to be taken into account in the literature, they are typically tackled using a single objective function that consists of minimizing the weighted sum of violated constraints (Van Bulck, Goossens, Schönberger, & Guajardo, 2020). However, the following multi-objective approaches in sports scheduling share our aim of developing an approximation of the Pareto-front, but focus on different objectives and settings. Duarte and Ribeiro (2008) tackle a bi-objective referee assignment problem arising in amateur leagues, which includes minimizing the difference between the target and the actual number of games assigned to each referee, and the idle time between consecutive games for each referee. They use a greedy constructive heuristic to create an initial solution, with an ILS (iterated local search) based repair heuristic making the initial solution feasible if necessary. Finally, the feasible solution is improved by another ILS-based procedure. Barone, While, Hughes, and Hingston (2006), While and Barone (2007), Craig, While, and Barone (2009) and While and Kendall (2014) present a multi-objective evolutionary algorithm to derive schedules for the Australian Football League, the Super 14 Ruby tournament the National Hockey League (NHL), and the English Football League, respectively. They all show their method can quickly produce superior solutions that dominate the official schedule, whilst offering a range of trade-off solutions to the organization body.Barone et al. (2006) consider four competing objectives: minimizing the interstate travel of the teams, maximizing revenue, fairly distributing the games over venues, and balancing the number of home games each team plays. While and Barone (2007) identify the following conflicting objectives: minimizing the travel between different regions, spreading the games as equally as possible to maximize revenue from advertising and broadcasting, and competition fairness via balancing the home games.Craig et al. (2009) focus on minimizing travel distance for teams, sequences of consecutive home or away games, and ensuring equity in rest time between games. Two principal objectives considered by While and Kendall (2014) are minimizing travel distance for teams and supporters, and the number of so-called pair clashes. More recently, Çavdaroğlu and Atan (2019) and Goossens, Yi, and Van Bulck (2020) both propose an integrated approach to incorporate the minimization of breaks and balancing of so-called carryover effects in sports scheduling, and study their trade-off. Goossens et al. (2020) also construct a Pareto frontier reflecting the trade-off between distribution equity and criterion efficiency of rest times. They propose a bi-criteria evolutionary algorithm generating a rich set of equitably-efficient compromise solutions, and apply it to an amateur indoor football league.

2.2. Sport teams grouping and multi-league scheduling

The bi-objective MLGSP treated in this paper studies the trade-off between travel distance and venue capacity violations, objectives which the literature so far has studied only separately through the STGP and the MLSP respectively.

The STGP deals with dividing a fixed number of sport teams into leagues or divisions while respecting given regulations. Although the STGP is practically relevant, it has not been widely studied in the literature. All studies use a geographical criterion to partition the teams in order to reduce total travel distance.

Saltzman and Bradford (1996) address realigning the teams for National Football League (NFL), which expanded to 30 teams in 1996. They set up a quadratic programming model to partition the teams into 6 divisions of 5 teams, with the aim to minimize total intradivisional travel by all teams. The results suggest that their realignment proposal is fairer than the one from 1995, and creates a substantial decrease in travel. Macdonald and Pulleyblank (2014) develop a geometric method to construct realignments that minimizes total travel by all teams over a season. This method is examined on practical instances with 30 or 32 teams from the Major League Baseball (MLB), the National Basketball Association (NBA), the NHL and the NFL.

In the case that teams need to be partitioned into divisions of equal size, the sport teams realignment problem can be modeled as a so-called k-way equipartition problem. An example of this case is given by Mitchell (2003), who optimally solve the realignment of the NFL for 8 divisions of 4 teams each using branch-and-cut. The alignment resulting from their algorithm reduces the total intradivisional travel distance by 45%. Later, Ji and Mitchell (2005) solve the same problems using branch-and-price. For the NBA and the NHL (3 divisions of 5 teams each), they improve the current realignment in terms of travel intradivisional travel distance. For the NFL, they match the optimal solution computed by Mitchell (2003).

More recently, Recalde, Severín, Torres, and Vaca (2016) define a team realignment problem faced by the Ecuadorian football association as a variant of the so-called balanced k-clique partitioning problem. An integer programming formulation and a heuristic are proposed. Their instance, where 44 teams need to be split into 8 leagues, could however not be solved to optimality after 4 hours of computation. In a subsequent work, Recalde, Severín, Torres, and Vaca (2018) reframe the problem as the so-called balanced k-way partitioning problem and develop a branch-and-cut algorithm and a tabu search meta-heuristic, which is able to find an optimal solution for this instance in less than a second.

While the above described research concerns professional sports leagues, Toffolo et al. (2019) study a setting from the Royal Belgian Football Association (RBFA) to organize youth football teams into leagues. They present several integer programming formulations, minimizing the total travel distance of the participating teams. Given the computational complexity of the STGP, and the scale of the instances they consider, they propose a simulated annealing heuristic with a ruin-and-recreate procedure. They demonstrate this heuristic can effectively find high-quality solutions for a set of instances involving up to 167 teams.

Given the league composition (i.e. the output of the STGP), the MLSP describes the simultaneous scheduling of multiple sports leagues. Multi-league sport scheduling construction involves inter-league constraints such as venue capacity constraints (Davari et al., 2020; Li et al., 2023; Schönberger, 2015, 2017), which rules out an approach where the leagues are scheduled one by one as an exact method. For a comprehensive overview of multi-league sports scheduling, we refer to Li et al. (2023).

Table 1.: Overview of considered characteristics; '✓' indicates the study includes this characteristic.

Characteristics	Toffolo et al. (2019)	Davari et al. (2020)	Li et al. (2023)	This study
Multi-league grouping	✓			✓
Multi-league scheduling		✓	✓	✓
Different league sizes	✓		✓	✓
Min. travel distance	✓			✓
Min. capacity violations		✓	✓	✓

We position our work to the literature as follows. Our focus lies on multi-league problems as faced by youth and amateur sports, in accordance with Toffolo et al. (2019), Davari et al. (2020) and Li et al. (2023), which involve many more teams, clubs, and leagues compared to professional sports. In line with Toffolo et al. (2019) and Li et al. (2023) (but contrary to Davari et al. (2020)), we consider different league sizes, however, where Toffolo et al. (2019) set minimum and maximum values to limit the number of teams grouped in a league, in our setting the size of each league is given. We follow the approach of Davari et al. (2020) and Li et al. (2023) to assign a HAP to each team for each league, in order to minimize the total venue capacity violation. The set of HAPs that is used for each league is given, and assumed to be feasible, in the sense that it allows a compatible schedule. The issue of deciding whether a feasible schedule exists for a given HAP set is a well-researched topic (see Briskorn (2008); Goossens and Spieksma (2011); Horbach (2010); Miyashiro, Iwasaki, and Matsui (2003); Van Bulck and Goossens (2020)); we do not go into details here.

Li et al. (2023) report that the so-called canonical HAP set, when used in all leagues, causes fewer capacity violations, which is accordingly adopted in this research. Furthermore, based on their artificial instances and a real-life case study, they show that optimizing the starting date of leagues is a clever strategy to avoid capacity violations compared to starting all leagues at the same date. Although we consider the starting round of each league given in order not to render the problem overly complex, we have used their insights to set the corresponding parameters (see Section 5.1).

3. Problem description

3.1. Problem definition

We are given a set of clubs C and a set of teams T with $|T|$ even, where each team belongs to exactly one club. We define $\hat{T}_c \subseteq T$ as the set of teams belonging to club

c. Each team is associated with a location corresponding to their club (the circles in Figure 2). The symmetric distance between teams t and p is denoted by $d_{t,p}$.

Let L be a set of leagues, with for each league $\ell \in L$ a given (possibly different) even league size s_ℓ, such that $\sum_{\ell \in L} s_\ell = |T|$. We make the common assumption, in order to avoid collusion and match-fixing, that no league can have multiple teams of the same club (see also Burrows and Tuffley (2015); Davari et al. (2020); Durán, Durán, Marenco, Mascialino, and Rey (2019); Li et al. (2023); Mitchell (2003); Saltzman and Bradford (1996); Schönberger (2017)). Furthermore, based on age, gender, or team strength, teams can typically not play in just any league. Hence, for each league ℓ, we define $\bar{T}_\ell \subseteq T$ as the set of teams that are eligible to play in league ℓ. Moreover, the set of leagues L is partitioned into $|G|$ league groups L_g such that all leagues in L_g share the same set of eligible teams.

During the season, all leagues play a compact k-round robin tournament (kRR), with k even. In a k-round robin tournament, each team faces each other team k times; in a compact tournament with an even number of teams, each team plays exactly once per round. Each league ℓ takes place as a series of $k(s_\ell - 1)$ consecutive rounds without interruption. Leagues can be played in overlapping rounds though, as depicted in Figure 2. The season consists of R rounds, where $R = \{1, 2, \dots, k(s_{\max} - 1)\}$ with $s_{max} = \max_{\ell \in L} s_\ell$. A starting round $r_\ell \in R$ is given for each league ℓ, which is a simplification of the multi-league scheduling problem by Li et al. (2023), where the starting round is a decision variable.

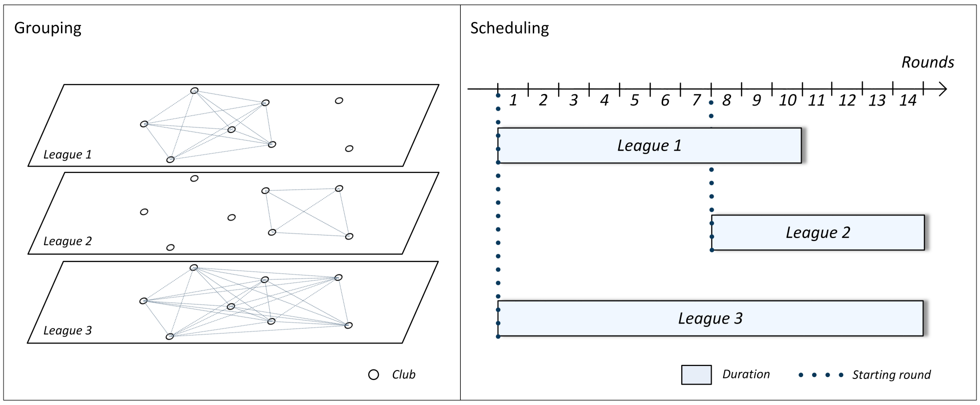

Figure 2.: A small example with 18 teams from 8 clubs, grouped into three 2RR leagues of size 8, 6, and 4, respectively, and $|R| = 14$.

Within a league, every team faces every other team k times and after each away game, the visiting team returns home. Hence, for $k = 2$, the total distance travelled by the teams is the summation of distances between teams and their opponents in the league.

We are given a feasible HAP set $\mathcal{H}_\ell$ consisting of complementary pairs of HAPs for each league $\ell \in L$ (as assumed by Davari et al. (2020); Li et al. (2023)), and denote $\mathcal{H} = \bigcup_{\ell \in L} \mathcal{H}_\ell$. Each HAP h is defined by a binary parameter $U_{h,r}$ which equals one if the team assigned to HAP h plays home in round $r \in R$, and zero if it plays away or has no game in round r. A pair of HAPs h and h' are complementary if $U_{h,r} + U_{h',r} \leqslant 1$ for all rounds r; a HAP set is feasible if it allows a compact k-round robin schedule. Leagues that start on the same round and have the same number of teams may use the same HAP set: n_h indicates how many leagues can use HAP h.

A capacity δ_c is given for each club c, corresponding to the number of matches it can host per round. Venue capacity violations occur when the number of teams of a club that play home simultaneously exceeds its venue capacity. For each club, the capacity violation in a round is measured by a scalar value that is either the number of its teams that play a home game in that round minus the club's capacity (if the venue capacity is exceeded), or zero if there is no venue capacity violation.

The MLGSP consists of determining, for each team $t \in T$, a league $\ell \in L$ as well as a HAP $h \in \mathcal{H}_\ell$, while minimizing total distance travelled and total venue capacity violations over the clubs. Both objectives can be conflicting, forcing the league organizer to make a trade-off. The problem instance in Figure 2 involves three leagues with eight, six and four teams, respectively (corresponding to instance S1-1 in Table 2). Figure 3 illustrates that in this case, no solution exists that minimizes both venue capacity violations and travel distance. Indeed, improving one objective degrades the other objective value.

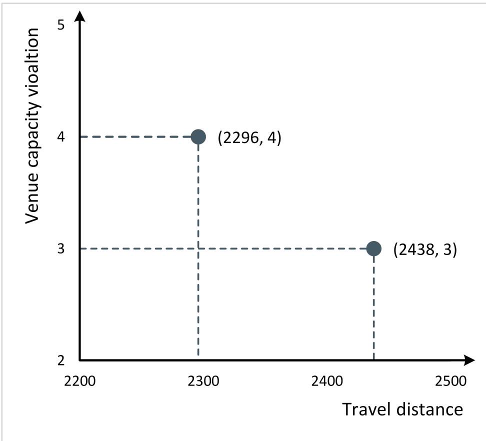

Figure 3.: Pareto front generated for the problem instance in Figure 2.

Observation. *If all leagues have the same (even) league size s and play according to the same complementary HAP set, there is no trade-off between both objectives.*

This observation follows from the result by Davari et al. (2020), who state that the minimum number of venue capacity violations for this special case is given by

$$k\,(s-1)\sum_{c\in C^-}\left(\frac{|T_c|}{2}-\delta_c\right) \tag{1}$$

where C^- is the set of clubs for which the capacity δ_c is less than half the total number of home games to be played by their teams. Note that in this special case, the minimal venue capacity violation can be obtained with any league composition. Furthermore, Davari et al. (2020) describe a polynomial-time algorithm that achieves this.

3.2. Model formulation

In this subsection, we present a bi-objective MILP formulation for MLGSP, where the main decision variables are $y_{t,\ell}$, which is one if team t is assigned to league ℓ and zero otherwise, and $x_{t,h}$, which equals one if team t is assigned to HAP h and zero otherwise. Furthermore, we have variables $z_{t,p}$ to indicate whether teams t and p are allocated to the same league ($z_{t,p} = 1$) or not ($z_{t,p} = 0$). The variable $v_{c,r}$ is used to keep track of venue capacity violations for each club c and round r. The problem can now be formulated as follows:

$$\min f_1 = \sum_{t,p \in T} d_{t,p} z_{t,p} \tag{2}$$

$$\min f_2 = \sum_{c \in C} \sum_{r \in R} v_{c,r} \tag{3}$$

s.t.

$$\sum_{\ell \in L : t \in \bar{T}_\ell} y_{t,\ell} = 1 \qquad \forall t \in T \tag{4}$$

$$\sum_{t \in \bar{T}_\ell} y_{t,\ell} = s_\ell \qquad \forall \ell \in L \tag{5}$$

$$\sum_{t \in \hat{T}_c \cap \bar{T}_\ell} y_{t,\ell} \leq 1 \qquad \forall \ell \in L, c \in C \tag{6}$$

$$y_{t,\ell} + y_{p,\ell} \leq z_{t,p} + 1 \qquad \forall \ell \in L, t, p \in \bar{T}_\ell \tag{7}$$

$$\sum_{t \in T} x_{t,h} = n_h \qquad \forall h \in \mathcal{H} \tag{8}$$

$$\sum_{h \in \mathcal{H} : \exists \ell : t \in \bar{T}_\ell \wedge h \in H_\ell} x_{t,h} = 1 \qquad \forall t \in T \tag{9}$$

$$y_{t,\ell} \leqslant \sum_{h \in H_\ell} x_{t,h} \qquad \forall \ell \in L, t \in \bar{T}_\ell \tag{10}$$

$$v_{c,r} \geq \sum_{t \in \hat{T}_c} \sum_{h \in \mathcal{H}} x_{t,h} U_{h,r} - \delta_c \qquad \forall c \in C, r \in R \tag{11}$$

$$v_{c,r} \geq 0 \qquad \forall c \in C, r \in R \tag{12}$$

$$x_{t,h} \in \{0,1\} \qquad \forall t \in T, h \in \mathcal{H} \tag{13}$$

$$y_{t,\ell} \in \{0,1\} \qquad \forall t \in T, \ell \in L \tag{14}$$

$$z_{t,p} \in \{0,1\} \qquad \forall t, p \in T, t \neq p. \tag{15}$$

The first objective (2) minimizes the total travel distance and the second objective (3) minimizes the total venue capacity violations. Constraints (4) guarantee that each team is assigned to exactly one league for which it is eligible, while Constraints (5) assert that the number of teams in league ℓ is exactly s_ℓ. Constraints (6) ensure that no teams from the same club are allocated to the same league. Constraints (7) set $z_{t,p}$ to one if teams t and p are in the same league. Constraints (8) enforce that each HAP is assigned to n_h teams. Constraints (9) guarantee that each team is assigned to a HAP of a league for which it is eligible. Similarly, Constraints (10) enforce that if a team is grouped into a league, it has to be assigned to a HAP from that league. Constraints (11)−(12) calculate the number of capacity violations for each club in each round, i.e. the number of the club's teams that play home in that round minus its capacity, or zero if the club has sufficient capacity. Constraints (13)−(15) declare the variable domains.

Note that the linear programming relaxation of formulation (4)−(15) with respect to objective (3) is always zero, since $y_{t,\ell} = y_{p,\ell} = \frac{1}{2}$ satisfies constraints (7), allowing $z_{t,p} = 0$ for all t, p, ℓ. Inspired by Toffolo et al. (2019), we introduce constraints (16) to cut away this fractional solution, forcing each team to have at least $\min(s_\ell) - 1$ and at most $\max(s_\ell) - 1$ opponents over the leagues ℓ for which it is eligible.

$$\min_{\ell \in L: t \in \bar{T}_\ell} s_\ell - 1 \leq \sum_{p \in T} z_{t,p} \leq \max_{\ell \in L: t \in \bar{T}_\ell} s_\ell - 1 \qquad \forall t \in T \tag{16}$$

The formulation may be further improved by reducing symmetry. Indeed, as leagues from the same league group, sharing the same starting round, league size, and HAP set are interchangeable, permuting such leagues creates symmetric solutions. This can be mitigated by stating that

$$\sum_{t \in \bar{T}_\ell} y_{t,\ell} \cdot t \leq \sum_{t \in \bar{T}_\ell} y_{t,\ell'} \cdot t \qquad \forall g \in G, \forall \ell, \ell' \in L_g : \ell < \ell', s_\ell = s_{\ell'}, r_\ell = r_{\ell'}, \mathcal{H}_\ell = \mathcal{H}_{\ell'} \tag{17}$$

Unfortunately, adding constraints (16) and (17) did not help to reduce the model's computation time or performance for our purpose. Indeed, when focussing on travel distance as an objective, the LP relaxation is clearly strenghtened, but still less than 20% of the best known solution on medium-scale instances (see Section 5.1). Furthermore, when imposing a time limit of 1,800s, we obtained either no solution, or a worse solution than without constraints (16) and (17).

4. Solution methods

Being a bi-objective optimization problem, MLGSP may not have a unique solution that simultaneously optimizes total distance travelled and total venue capacity violations. Hence, our aim is to identify a set of Pareto solutions. In this section, we describe the exact ϵ-constraint method (Section 4.1), and we develop a two-layer constructive heuristic (Section 4.2) to obtain an approximation of the Pareto front.

4.1. Epsilon-constraint method

The ϵ-constraint method (ECM), introduced by Haimes (1971), is a classic exact approach to tackle multi-objective optimization problems. The basic idea of this method is to transform the multi-objective decision making problem into a single-objective optimization problem (see e.g. Mesquita-Cunha, Figueira, and Barbosa-Póvoa (2023)). More specifically, decision makers select a primary objective function to be optimized, whereas the remaining objective functions are converted into constraints by imposing an upper bound ϵ. By iteratively solving this problem, while updating ϵ with a positive constant Δ at each iteration, a Pareto front is obtained.

In our problem, travel distance minimization (f_1) is chosen as the primary objective. Indeed, it is more efficient to set steps Δ in terms of capacity violations (f_2), since the range of this integer-valued objective is typically much smaller than that of f_1. Moreover, preliminary experiments showed that this approach solves faster than its counterpart which has f_2 as the primary objective. The ϵ-constraint problem then takes the following form:

$$\min f_1;\ \text{s.t. } f_2 \leq \epsilon \text{ and constraints } (4) - (15). \tag{18}$$

Algorithm 1 ϵ-constraint method (ECM)

Input: An instance of MLGSP
Output: Pareto front PF

Solve the problem for each objective function individually, and find the ideal point (f_1^I, f_2^I). Here, $f_1^I = \{\min f_1;\ s.t.\ (4)-(15)\}$, and $f_2^I = \{\min f_2;\ s.t.\ (4)-(15)\}$.
Solve each objective function individually, with a constraint on the value of the other objective, and find the nadir point (f_1^N, f_2^N). Here, $f_1^N = \{\min f_1;\ s.t.\ f_2 = f_2^I,\ (4)-(15)\}$, and $f_2^N = \{\min f_2;\ s.t.\ f_1 = f_1^I,\ (4)-(15)\}$.
$PF \leftarrow PF \cup \{(f_1^I, f_2^N), (f_2^I, f_1^N)\}$, $\epsilon \leftarrow f_2^N - \Delta$
while $\epsilon > f_2^I$ **do**
  Solve the ϵ-constraint problem (18) and add the resulting solution (f_1, f_2) to PF.
  $\epsilon \leftarrow f_2 - \Delta$
end while
Remove all dominated solutions from PF.

The ϵ-constraint procedure to acquire the entire Pareto front PF is described in Algorithm 1. We use lexicographic optimization (Mavrotas, 2009) to attain the range of each objective function, which guarantees that the solutions resulting from individual optimization are extreme points (i.e., (f_1^I, f_2^N) and (f_2^I, f_1^N)) on the Pareto curve. By setting $\Delta = 1$, all Pareto points can be collected since the total capacity violations are integer.

4.2. Two-layer constructive method

If acquiring the Pareto-optimal set is computationally too expensive, we need to revert to approximation (Borgonjon & Maenhout, 2022; Chircop & Zammit-Mangion, 2013). We therefore develop a Two-layer Constructive heuristic Method, to which we refer as TCM. Algorithm 2 gives an overview of our method, which starts with a mathematical programming approach to create an initial point (line 2) with an optimal venue capacity violation, and a resulting travel distance, which is further improved via local search (line 3). Next (lines 5-27), we opt for a procedure which decomposes the MLGSP into two sub-problems, namely, the STGP and the MLSP. The former is solved on the outer layer (i.e., without any pre-fixed assignment of teams to HAPs), using simulated annealing; the latter is solved on the inner layer (i.e., given an assignment of teams to leagues), using a MILP solver. Hence, at each iteration, we obtain a candidate point characterized by some total travel distance and capacity violation, until the procedure stops according to a cooling criterion. Eventually, given the list of potential Pareto efficient solutions, we use GEN-PARETO, a fast nondominated sorting algorithm (Jensen, 2003) to identify the Pareto set (line 28). Finally, the travel distance of each point in this set is locally improved, after which the sorting method is run again to distill the Pareto front approximation (lines 29-35).

The computational complexity of the integrated problem steers us away from a global multi-objective optimization method, towards a decomposition approach. A natural decomposition is to tackle the league composition and HAP assignment separately. Furthermore, deciding on the league composition first (on the outer layer), and then attempting to minimize the venue capacity violation (on the inner layer) turns out computationally much more efficient than vice versa. Indeed, without a league composition, the multi-league scheduling problem typically results in many optimal solutions. Moreover, fixing the assignment of teams to HAPs does not make the league composition problem much easier, and also offers no guarantee on the existence of a feasible league composition.

In the following, single algorithm components are explained in detail in Sections 4.2.1 to 4.2.3, where the notation mentioned in Algorithm 2 is defined as well. In addition, Sections 4.2.4 and 4.2.5 describe an improvement procedure and a speed-up version of our two-layer constructive method, respectively.

Algorithm 2 Two-layer constructive method (TCM)

Input: An instance of MLGSP
Output: Pareto front approximation A

$k \leftarrow k_0$, $AF \leftarrow \varnothing$, $P \leftarrow \varnothing$
Generate an initial solution (f_1^{in}, f_2^I) and resulting assignment of teams to leagues S^{in}.
$(f_1^{in}, S^{in}) \leftarrow$ SWAPOPTOR(S^{in})
$AF \leftarrow AF \cup \{(f_1^{in}, f_2^I)\}$, $P \leftarrow P \cup \{(S^{in}, f_1^{in}, f_2^I)\}$, $S \leftarrow S^{in}$, $f_1 \leftarrow f_1^{in}$.
while $k > k_1$ **do**
  for $iter = 1$ to I **do**
    $S' \leftarrow$ NEIGHBOUR(S)
    $f_1' \leftarrow$ DISTANCE(S')
    if $f_1' < f_1^{in}$ **then**
      if $f_1' < f_1^*$ **then**
        $f_2' \leftarrow$ MLSP-I(S')
        $AF \leftarrow AF \cup \{(f_1', f_2')\}$, $P \leftarrow P \cup \{(S', f_1', f_2')\}$
      else
        Find $(f_1^+, f_2^+) \in AF$ such that $f_1^+ \leqslant f_1'$ and $f_1' - f_1^+$ is minimal.
        $f_2' \leftarrow$ MLSP-II(S', f_2^+)
        $AF \leftarrow AF \cup \{(f_1', f_2')\}$, $P \leftarrow P \cup \{(S', f_1', f_2')\}$
      end if
    end if
    if $f_1' \leq f_1$ **then**
      $S \leftarrow S'$, $f_1 \leftarrow f_1'$
    else
      Generate a random constant $u \in [0, 1]$.
      if $u < e^{-\Delta/k}$ **then** $S \leftarrow S'$, $f_1 \leftarrow f_1'$
    end if
  end for
  $k \leftarrow k\sigma$
end while
$AF \leftarrow$ GEN-PARETO(AF), $A \leftarrow \varnothing$
for $(S, f_1, f_2) \in P$ **do**
  if $(f_1, f_2) \in AF$ **then**
    $(f_1, S) \leftarrow$ SWAPOPTOR(S)
    $A \leftarrow A \cup \{(f_1, f_2)\}$
  end if
end for
$A \leftarrow$ GEN-PARETO(A)

4.2.1. Initial solution

The optimization process begins with a feasible solution composed of an initial value f_1^{in} (i.e., total distance travelled) and the ideal value of f_2^I (i.e., total venue capacity violations). The latter follows from solving the single-objective integrated formulation (19) optimally, while we calculate the former according to the resulting initial assignment of teams to leagues S^{in}.

$$\min\ f_2;\ \text{s.t. constraints (4)} - (6), (8) - (14). \tag{19}$$

We noticed that the above formulation is computationally expensive for large-scale instances. Hence, for such instances, we developed a two-stage approach. In the first stage, formulation (20) minimizes capacity violations for a setting that ignores the league compositions, however only assigning teams to HAPs from leagues in which they are eligible to play, so that the ideal value of f_2^I is obtained.

$$\min\ f_2;\ \text{s.t. constraints (8)} - (9), (11) - (13). \tag{20}$$

For the second stage, we define binary parameters $x'_{t,h}$ representing the assignment of teams to HAPs obtained from the first stage. With this assignment of teams to HAPs given, Formulation (21) then obtains a feasible initial league composition S^{in} and its corresponding f_1^{in} value, however without considering the minimization of total travel distance (in order to keep the computation time limited). The new constraints guarantee that teams assigned with the same HAP do not end up in the same league.

$$\text{s.t.}\ \begin{cases} \text{constraints (4)} - (7), (14), (15) & \\ \sum_{t\in T} y_{t,\ell} x'_{t,h} = 1 & \forall \ell \in L, h \in \mathcal{H}_\ell. \end{cases} \tag{21}$$

Note that the assignment of teams to HAPs from the first stage may lead to an infeasible league composition problem in the second stage. Indeed, the constraint stating that no teams from the same club can be allocated to the same league may be conflicting with the given assignment of teams to HAPs. In this case, formulation (20) will be executed again with additional constraint $\sum_{t\in T}\sum_{h\in\mathcal{H}} x_{t,h}x'_{t,h} < |T| - 1$, which forbids the current assignment of teams to HAPs, until it produces a feasible solution. In our computational experiments (see Section 5), however, we did not run into such situation.

If leagues of the same size share an identical HAP set (which is the case in our experimental setting, see Section 5.1), we can further improve the travel distance of our initial solution using the local search procedure denoted by SWAPOPTOR. This procedure uses a neighbourhood which does not worsen the total capacity violation of a given solution. For any team, its neighbourhood consists of those teams from a different club playing according to the same HAP, playing in a different league from the same league group with the same starting round. Note that swapping the leagues of this team and any team in its neighbourhood will not influence the total capacity violation. It will however change the league composition, which could reduce the total travel distance. In each iteration, a team is selected randomly. We consider swapping this team's league with the league of each team in the neighbourhood, and accept the best move if it improves the total travel distance. When the number of iterations without improvement of the travel distance reaches a predefined maximum value MAXUC, the procedure is stopped.

4.2.2. Outer layer

On the outer layer, we gradually improve the assignment of teams to leagues with respect to the total travel distance (f_1) by means of simulated annealing, which has proven effective in multi-objective optimization problems (Suman & Kumar, 2006; Xu & Qu, 2011) as well as sports scheduling (Kyngäs et al., 2017; M. Wright, 2018). Besides the ease of manipulating a solution, our choice for simulated annealing is based on its good performance for the STGP as demonstrated by Toffolo et al. (2019). With each iteration *iter*, a neighbour (S') of the current league composition (S) is created through one of two neighbourhood operators (denoted by NEIGHBOUR). The exchange operator arbitrarily picks two distinct leagues from the same league group, and randomly exchanges teams between both leagues (see Figure 4). We do not swap teams from the same club though, since exchanging such teams does not change the total distance (both teams have the same distance to any other team).

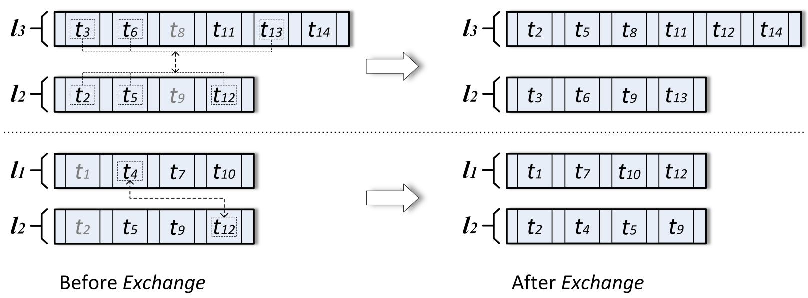

Figure 4.: Exchange operation between two leagues with different (top) or identical (bottom) sizes. Teams marked in grey belong to the same club and cannot be swapped.

The relocate operator (see Figure 5) randomly selects a number of leagues from the range $[2, |L|]$, and removes a team from each selected league. The problem of relocating the teams to these leagues can be cast to an assignment problem based on a weighted bipartite graph. Every edge joining a 'team' vertex and a 'league' vertex has a non-negative weight, which is the total travel distance that results from adding the team to this league. Any edge linking a team to a league that already has a team from the same club, or a league for which this team is not eligible is removed. The Kuhn-Munkres algorithm (Kuhn, 1956; Munkres, 1957) is used to find a minimum weighted perfect matching in polynomial time.

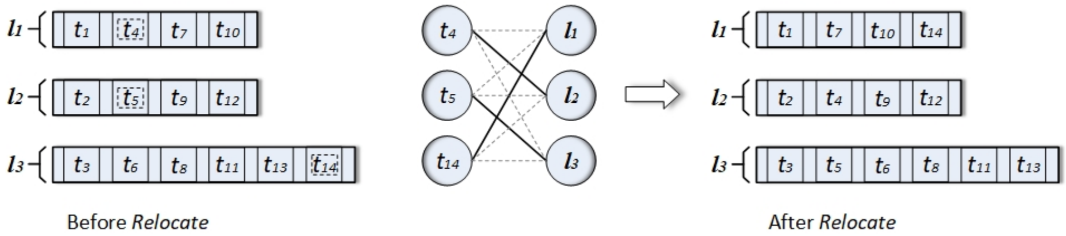

Figure 5.: Relocate operation among three leagues. The full lines represent the obtained minimum weight perfect matching.

The procedure then computes the total travel distance of the neighbour solution S' using the DISTANCE function. If this distance f_1' improves on the travel distance resulting from the previous assignment of teams to leagues, S' is accepted; a deteriorating move is accepted with a probabilistic criterion. The two-layer method runs I times at each temperature, after which the temperature k is updated with a constant factor σ. Finally, the whole outer-layer process terminates when the final temperature k_1 is reached.

4.2.3. Inner layer

The inner layer focuses on minimizing the capacity violation (f_2), with the assignment of teams to leagues as determined by the outer layer as input (we use $\bar{S}_\ell$ to denote the set of teams that form league ℓ). We only consider league compositions S' that have a total travel distance smaller than that of the initial solution (which has the ideal capacity violation). If the neighbour's total travel distance f_1' is better than the best travel distance f_1^* obtained so far in the list of potential Pareto efficient solutions AF, we solve formulation (22), referred to as MLSP-I, to obtain an optimal value f_2' for its total capacity violation, and (f_1', f_2') is added to AF as a new candidate Pareto solution.

$$\min\ f_2;\ \text{s.t.}\ \begin{cases} \sum_{t\in\bar{S}_\ell} x_{t,h} = 1 & \forall \ell \in L, h \in \mathcal{H}_\ell \\ \sum_{h\in\mathcal{H}_\ell} x_{t,h} = 1 & \forall \ell \in L, t \in \bar{S}_\ell \\ \text{constraints } (11) - (13). \end{cases} \tag{22}$$

Otherwise, we find the candidate Pareto solution $(f_1^+, f_2^+) \in AF$ such that f_1' is closest to, but larger than f_1^+. Its corresponding total capacity violation f_2^+ is then added as a constraint $f_2 \leqslant f_2^+ - 1$ on the objective in formulation (22), referred to as MLSP-II. If MLSP-II is feasible, the resulting solution (f_1', f_2') is added to AF. Note that constraint $f_2 \leqslant f_2^+ - 1$ is added to not only sweep dominated points but also to reduce the computation time.

4.2.4. Improvement procedure

Our two-layer method is outer layer-directed, and oriented towards searching for increasingly better solutions with respect to travel distance (f_1), for which the capacity violation (f_2) is subsequently minimized. We initiate the process from a solution with high travel distance but ideal capacity violation. However, the simulated annealing mechanism swiftly steers toward solutions with reduced travel distance, without thoroughly exploring the earlier area where solutions with low venue capacity violations exist. To address this, we introduce an improvement procedure. It involves running the two-layer method once more from the initial point (f_1^{in}, f_2^I) to return to uncharted region of the Pareto front. Indeed, we adjust the final temperature and the number of iterations at each temperature to slow down the simulated annealing process for this second run, in order to focus more on part of the Pareto front where we rushed through at first. A key difference is that if the resulting travel distance f_1' is better than the previous value but exceeds a predefined threshold β, we do not accept its solution S'. The algorithm with the improvement procedure is referred to as TCM-I.

4.2.5. Speed-up strategy

Although the MLSP sub-problem with different league sizes can be solved relatively efficiently using MLSP-I and MLSP-II (up to the largest instance scale covered, see Section 5.1), it is computationally expensive to run it repeatedly, especially when applying the improvement procedure. Therefore, a speed-up version of the two-layer constructive method is proposed, in particular to handle large-scale instances. It replaces the IP models with a heuristic which is based on a polynomial-time algorithm designed by Davari et al. (2020), which is exact for leagues with the same size, and has been extended as a heuristic for problems with leagues of different sizes by Li et al. (2023). Since the venue capacity violations generated by this method are not necessarily optimal, the Pareto front resulting from applying the nondominated sorting method to the candidate points AF may be further improved. Therefore, for each point on the Pareto front, we try to improve its objective f_1 using the SWAPOPTOR local search procedure, and re-solve its objective f_2 optimally using formulation (22), given its league composition as input. Eventually, the nondominated sorting method is re-employed on the resulting solutions to identify an approximate Pareto front. This speed-up version is referred to as TCM-IS.

5. Computational study

This section presents the results of our computational study conducted to analyze the proposed solution methods. First, in Section 5.1, we discuss how problem instances are generated and we provide parameter settings for the heuristics. Next, we describe quality indicators used to evaluate the bi-objective optimization methods in Section 5.2, and report on the performance of these methods in Section 5.3. An application based on data from the Royal Belgian Football Association follows in Section 5.4.

The proposed algorithms were developed in C++, using an $O(n^3)$ implementation of the Kuhn-Munkres algorithm; mathematical models were implemented using CPLEX 20.1.0. For the ECM, we impose a time limit of 3,600s on producing the ideal point and nadir points and 1,800s for each ϵ-constraint problem. All computations were performed on a personal computer with an Intel Core i7-10850H CPU, 2.70 GHz processor and 16.0 GB RAM.

5.1. Instances and parameters

Due to the fact that the multi-league instances in the literature either lack data on venue capacity (Toffolo et al., 2019) or travel distance (Li et al., 2023), we generate our own set of instances as follows. We create ten sets of instances with a varying number of teams, leagues, and clubs, as well as league and club sizes. As league sizes 16 and 8 are the most common in the practical setting we observed with the Royal Belgian Football Association, we opt for these league sizes in large instances. A larger variety of league sizes is maintained in the small instances.

Each set contains five problem instances that differ in the the location and the capacity of clubs, to reduce the impact of randomness that could occur with a single instance. For each club, its horizontal and vertical coordinates are randomly generated in the range $[0, 50]$, from which a distance $d_{t,p}$ as the crow flies follows for each pair of teams t and p. For each club c, a capacity δ_c is a randomly chosen integer between $\max(\lfloor|\hat{T}_c|/2\rfloor - 2, 1)$ and $\min(\lfloor|\hat{T}_c|/2\rfloor + 2, |\hat{T}_c|)$, in line with Li et al. (2023).

We focus on 2RR leagues, where each team meets each other team twice (once at home and once away), since this format is wide-spread in many popular sports (e.g. soccer, see Goossens and Spieksma (2012)). The parameters $U_{h,r}$ are set according to the so-called canonical HAP sets 's_ℓ-C-1' (see Li et al. (2023)) for each league ℓ, with all leagues of the same size using the same HAP set. In the medium- and large-scale instances, the 8-team leagues are split into two groups of equal size, where one group is played in the first half of the season ($r_\ell = 1$), and the other group in the second season half ($r_\ell = 17$), such that there is no overlap between leagues in different groups. Likewise, in the small-scale instances, the overlap between two smaller-sized leagues is avoided as much as possible. As argued by Li et al. (2023), these choices make sense to reduce the total venue capacity violations.

Table 2 summarizes the design of the instance sets, which can be retrieved from our website[1], together with the source code of the instance generator. The instances are categorized as small, medium and large: small instances are those with at most 50 teams, while large instances involve more than 500 teams. In addition to these artificial instances, an instance based on a real-life case from the Royal Belgium Football Association (RBFA) is listed, for which we refer to Section 5.4.

Based on preliminary parameter configuration tests (not reported in this paper), considering the trade-off between solution quality and computation time, in the first run, the simulated annealing in the outer layer cools down from an initial temperature

[1] `https://robinxval.ugent.be/RobinX/multiLeagueRepo.php`

$k_0 = 100$ to a final temperature $k_1 = 1$. The number of iterations per temperature is set to $I = 15$; we use the standard cooling rate $\sigma = 0.99$. While only the exchange operator is used at first, when the current temperature drops to $\frac{1}{2}k_0$, the neighbourhood is constructed by randomly choosing between the exchange operator and the relocate operator. For the swap operator, MAXUC is set to $\frac{3}{2}|T|$. For the improvement procedure, we use $k_0 = 10$ as initial temperature, and $\beta = \frac{1}{100}\max|f_1^{a+1} - f_1^a|$, where a refers to the points in the Pareto set generated by the first run, ordered with increasing travel distance.

Table 2.: Overview of instance types and their features.

Instance type	Instance ID	No. teams	No. clubs	No. leagues	No. leagues per size					Club size range
					16	10	8	6	4	
Small	S1	18	8	3	0	0	1	1	1	[2,3]
	S2	34	16	3	1	1	0	1	0	[2,3]
Medium	M1	80	17	8	2	0	6	0	0	[3,6]
	M2	112	18	11	3	0	8	0	0	[4,8]
	M3	208	25	19	7	0	12	0	0	[5,10]
	M4	336	40	30	12	0	18	0	0	[6,10]
Large	L1	512	60	46	18	0	28	0	0	[3,13]
	L2	1120	70	120	20	0	100	0	0	[1,30]
	L3	1600	110	190	25	0	160	0	0	[1,27]
	L4	2160	120	260	30	0	230	0	0	[1,34]
Real-life	RBFA	2688	366	307	29	0	278	0	0	[1,30]

5.2. Solution quality indicators

Evaluating the quality of an approximation of the Pareto-optimal set is not straightforward (see e.g. Zitzler, Thiele, Laumanns, Fonseca, and da Fonseca (2003) for a discussion). Many metrics have been introduced in the literature, we refer the reader to e.g. Audet, Bigeon, Cartier, Le Digabel, and Salomon (2021) for a comprehensive overview. Performance indicators can be categorized according to what quality properties they capture. In general, without knowing the exact Pareto front, no indicator proves that one approximation is better than another, but the approximation that holds better values is typically preferred (Lust & Tuyttens, 2014). We use the following three widely-used (Borgonjon & Maenhout, 2022; Demir, Bektaş, & Laporte, 2014) indicators to assess four properties, namely, cardinality, convergence, spread and uniformity.

– Unique nondominated solutions (UNFR) (Berry & Vamplew, 2005): this indicator measures the proportion of unique nondominated solutions of a given Pareto set A to the size of a reference set F_{ref}. Without knowledge of the exact Pareto set, the reference set may be chosen as the set of unique nondominated solutions over all obtained (approximate) Pareto sets. Let A_{unf} be the unique nondominated front of a given Pareto set A ($A_{unf} \subseteq A$), then

$$\mathrm{UNFR}(A) = \frac{|a \in A_{unf} | \nexists f \in F_{ref} : f \text{ dominates a}|}{|F_{ref}|} \tag{23}$$

The UNFR value is in the range of [0,1]. A higher value is considered to be better: a UNFR value of one indicates that the combined nondominated front is precisely comprised by solutions in A.

– Hypervolume (HV) (Zitzler et al., 2003): this indicator evaluates the properties of convergence, spread, and cardinality. Given a Pareto set A and a reference point r_a (generally a nadir point), the HV value is the volume of the union of the hypercubes determined by each solution of this set and the reference point. A bi-objective case is depicted in Figure 6, where each point in the Pareto set forms a rectangle (see shaded area) with respect to the reference point; in this case, the HV indicator is the area of the union of all rectangles.

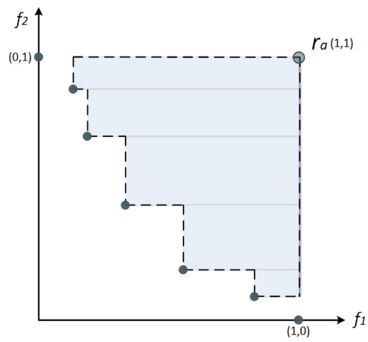

Figure 6.: Illustration of the hypervolume indicator for a bi-objective problem.

A higher HV value is preferred. We calculated this indicator based on normalized objective values and reference point (1,1).

– Spacing (SP) (Schott, 1995): this indicator reflects how uniformly spaced a set of nondominated solutions is. It is computed based on the variation of the distance between solutions in a Pareto set A according to

$$\mathrm{SP}(A) = \sqrt{\frac{1}{|A|-1} \sum_{i=1}^{|A|-1} (d_i - \bar{d})^2} \tag{24}$$

where d_i is the Euclidean distance between two consecutive points and $\bar{d} = \frac{1}{|A|-1} \sum_{i=1}^{|A|-1} d_i$.

We normalize the objective values due to their different magnitudes. Ideally, the SP value equals zero: all distances between two consecutive points on the Pareto front are then identical, and hence the points are uniformly distributed.

5.3. Computational results

First, we evaluate the performance of the solution methods ECM and TCM. Table 3 summarizes for each small-scale instance the Pareto solutions found and the computation time spent by both methods. Taking the Pareto front generated by ECM as the reference set, the UNFR value of TCM is one for all instances. In other words, TCM produces the optimal Pareto front for all small-scale instances. In terms of computation time, TCM operates far more efficiently than ECM, reaching a solution ten times faster on average. Note that, by construction, there is no point in running TCM-I if TCM produces an optimal solution, and that TCM-IS – designed for large-scale instances – requires multiple leagues of the same size, which is not the case for the small-scale instances.

Table 3.: Comparison of results obtained with ECM and TCM on small-scale instances.

Instance	ECM		TCM	
ID	Pareto front	Time (s)	Pareto set	Time (s)
S1-1	(2296,4);(2438,3)	4	(2296,4);(2438,3)	3
S1-2	(2786,4);(2788,3)	3	(2786,4);(2788,3)	14
S1-3	(2400,4);(2434,3)	3	(2400,4);(2434,3)	6
S1-4	(2384,5);(2458,4)	5	(2384,5);(2458,4)	63
S1-5	(2444,5);(2592,4)	5	(2444,5);(2592,4)	4
S2-1	(10140,8);(10476,7)	2314	(10140,8);(10476,7)	82
S2-2	(9996,8);(10366,7)	1166	(9996,8);(10366,7)	365
S2-3	(9910,10);(10280,9)	581	(9910,10);(10280,9)	48
S2-4	(9172,8);(9540,7)	3941	(9172,8);(9540,7)	185
S2-5	(8198,5);(8308,4)	407	(8198,5);(8308,4)	21
Avg.		843		79

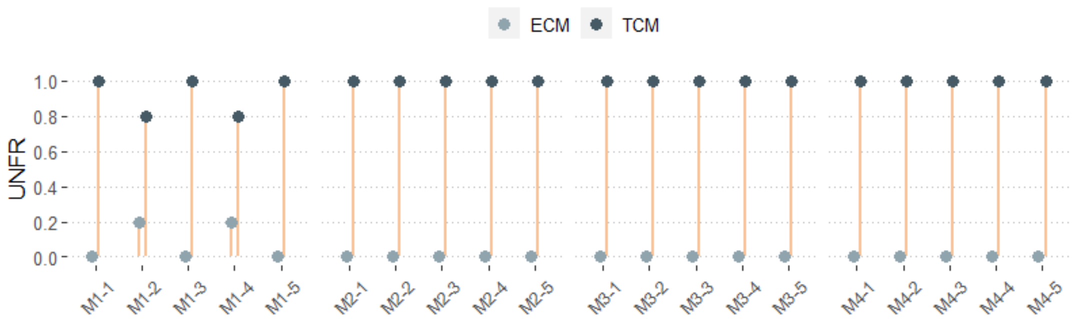

Figure 7.: Comparison of results obtained with ECM and TCM using UNFR indicator for medium-scale instances.

We now turn our attention to medium-scale instances, and the results, as illustrated in Figures 7 and 8, present a comparative analysis between ECM and TCM based on the UNFR indicator and computation time. The result reveals that TCM produces Pareto solutions that entirely dominate those generated by ECM in 18 out of 20 instances, and partially dominate in the remaining two instances. However, ECM consumes on average 76 times more computation time than TCM. Additionally, ECM failed to produce any feasible solutions for large-scale instances within the specified time limit. In other words, the effectiveness of ECM is severely limited when dealing with instances of medium- and large- scales.

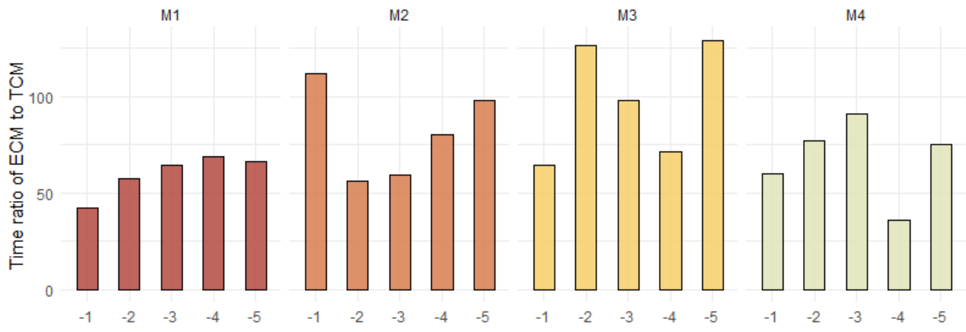

Figure 8.: Computation time of ECM compared to TCM for medium-scale instances.

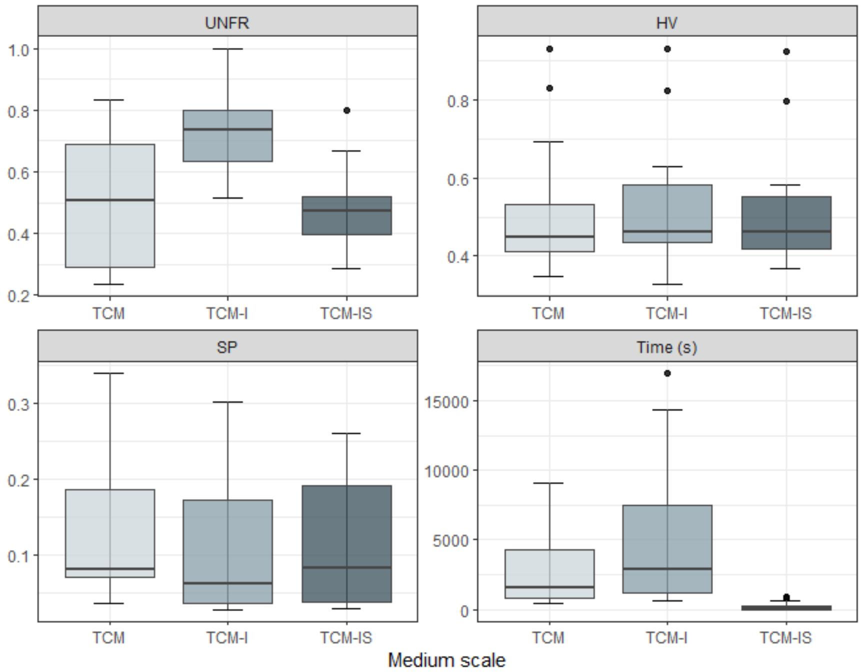

Figure 9.: Performance of the proposed methods displayed via boxplots on medium-scale instances.

Finally, we assess the value derived from the improvement procedure and the speed-up strategy. Figures 9 and 10 provide boxplots on the three evaluation metrics as well as the computation time for the Pareto front approximations obtained with TCM, TCM-I and TCM-IS on medium and large-scale instances; detailed results can be found in the Appendix in Tables A1 and A2, respectively.

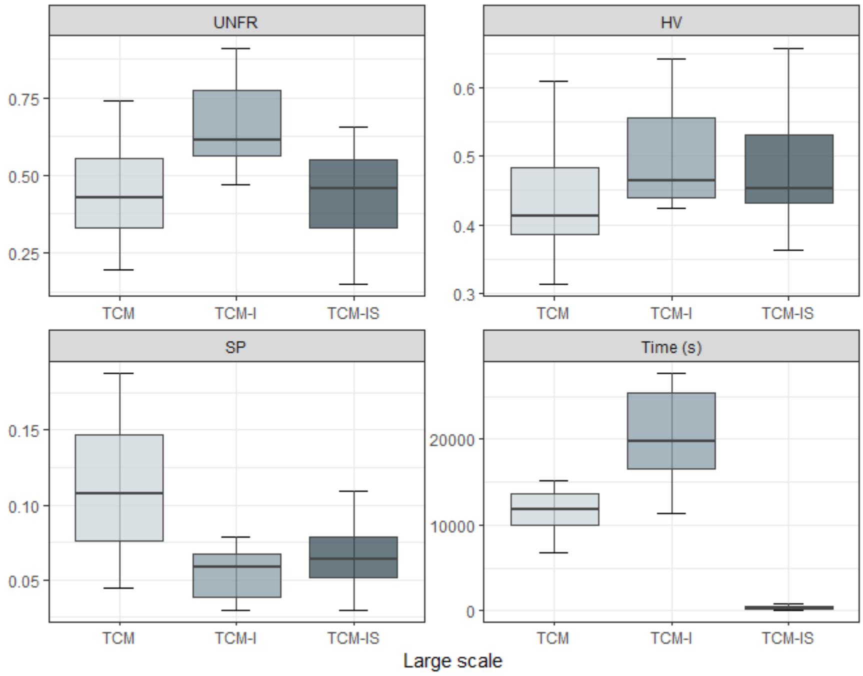

Figure 10.: Performance of the proposed methods displayed via boxplots on large-scale instances.

With respect to the unique nondominated solutions (UNFR), TCM-I is superior to both TCM and TCM-IS, while the latter two exhibit similar performance. This confirms that the improvement procedure (see Section 4.2.4) add a significant number of points to the Pareto front. However, for the medium-scale instances, these points do not fundamentally alter the shape of the Pareto front, as is witnessed by the spacing (SP) and hypervolume (HV) measures. For the large instances, the added value of TCM-I is more pronounced. In this case, it does not only yield more points on the Pareto front, these points are now also more uniformly distributed compared to TCM, and the hypervolume also shows a clear improvement. Reverting to the speed-up version (TCM-IS) eliminates most of the advantage with respect to UNFR, although the improvement with respect to spacing and hypervolume largely remains. Concerning the computation time, there is a considerable difference between TCM-IS and the other two algorithms. Indeed, using the polynomial-time heuristic instead of MIP models to minimize the capacity violations decreases the computation time dramatically to only 6.4% (3.7%) on medium-scale instances and 3.6% (2.1%) on large-scale instances of time required by TCM (TCM-I), respectively. In summary, the incorporation of the improvement procedure significantly enhances the performance of the algorithm, as evidenced by TCM-I outperforming TCM and TCM-IS in terms of uniformity, cardinality, and convergence of the resulting Pareto front approximation. However, this comes at the expense of much more computation time, particularly for larger instances. In contrast, the speed-up version demonstrates commendable performance with a substantial reduction in computation time.

5.4. An instance based on real-life data from football

As a more realistic application, we investigate a setting based on data from youth and amateur football divisions in the Belgian province East-Flanders for the season 2016–17. Note that for simplicity, we disregard a small number of leagues with uncommon league sizes (e.g. two leagues of size 10). The resulting problem instance features 2542 teams and 220 clubs, which are accommodated into 307 leagues. There are in total 29 league groups, based on age, gender and team strength, with a number of leagues ranging from 1 to 30. Note that the team grouping problem constrains teams to be assigned to a league from their original league group. To handle the issue of having an odd number of teams in a number of leagues, we followed common practice (Burrows & Tuffley, 2015) and slightly altered the instance by introducing 146 dummy teams, each with their corresponding dummy club, in order to be in line with our assumption of even-sized leagues. The resulting leagues now all have 8 or 16 teams; consequently, the season has 30 rounds. The 8-team leagues are split into two groups of equal size: half are designated as early leagues (starting in round 1), the remaining half are the late leagues (starting in round 17). Other problem characteristics are summarized in Table 2.

After retrieving the longitude and latitude information of each club's location using Google Maps, we calculate the travel distance between any pair of teams. When a team is scheduled to play against a dummy team, it is effectively not playing in that round. Hence, the travel distance between a team and a dummy team is set to zero. The official league composition incurs total travel distance of 275,236 km. We point out that this official solution at times violates the constraint that leagues should not include multiple teams from the same club, which obviously has a positive effect on travel distance.

Unfortunately, the data provided does not include the venue capacity of the clubs. Setting the capacity of each club to the maximal number of home games of its teams at any point during the competition would be an overestimation of its real capacity (indeed, there were capacity violations in the official schedule, but it is unknown to us when and where). Moreover, this essentially takes away most of the challenge for this objective. Therefore, we decided to set the capacity of each club following the approach explained in Section 5.1. The capacity of each dummy club is set to one. Note that, being developed for even league sizes, our method does not ignore home games against dummy teams (which in fact do not consume venue capacity), such that we likely overestimate the capacity violations.

Despite being a challenging instance, TCM-I manages to find 14 Pareto solutions using ten hours of computation time. The approximate Pareto set obtained by TCM-IS lies close to TCM-I's and consists of 12 Pareto solutions found in only 455 seconds. The solutions are depicted in Figure 11. For the reasons explained above, we cannot compare our results with the official schedule.

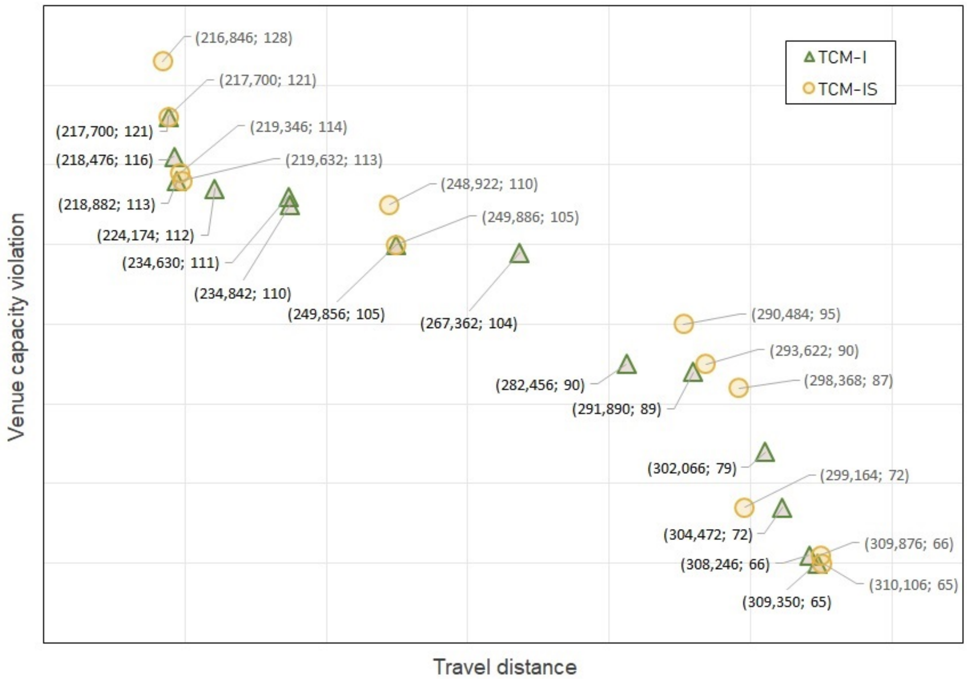

Figure 11.: Objectives scatter plot with the Pareto front approximations by TCM-I and TCM-IS.

Nevertheless, the results show that there is a wide range of solutions to choose from, depending on how the league organizer makes the trade-off between travel distance and venue capacity violation. Indeed, the total venue capacity violation can be halved if travel distance is ignored, and vice versa, the travel distance can be reduced with 30% at the expense of capacity violations. Decision-makers can use our results to strike a balance between travel distance and venue capacity violations based on their preference. However, knowing that the RBFA does not prioritize travel distance over capacity violations, their choice to first decide on league composition, and only then to tackle the scheduling problem does not seem to make much sense.

6. Conclusions

Sport teams grouping and multi-league scheduling are both important for organizing a sports competition. The former deals with minimizing the total travel distance faced by the teams, while the latter focuses on minimizing the venue capacity violations faced by the clubs. We showed that these two objectives can be conflicting when leagues have different sizes. This paper is the first to integrate both aspects.

We developed a bi-objective mixed integer linear programming model, as well as a two-layer constructive heuristic and its speed-up version. In comparison with the exact ϵ-constraint method on small-scale problem instances, the constructive method was found to be competitive with respect to solution quality, but computationally much more efficient. Furthermore, computational experiments on larger instances evaluated with multi-objective performance indicators show that while the constructive method with the improvement procedure obtains superior results, its speed-up version can efficiently approximate the Pareto front in a much shorter time, which is particularly helpful for the largest, realistically-sized instances. In order to demonstrate that our approach is suited for practice, a problem instance based on data from the Royal Belgian Football Association was presented. The resulting Pareto set shows that there is indeed a considerable trade-off to be made between travel distance and venue capacity violation.

This contribution is a first step in integrating the grouping and scheduling problem, which leaves several avenues for future research. First of all, we want to encourage researchers to investigate other multi-objective approaches (e.g. evolutionary algorithms, weighted optimization, etc.) and compare them with our heuristic. To facilitate this, we make the problem instances publicly available, so they can serve as a benchmark, as well as the source code for our instance generator, so more and different instances can easily be created. Second, there is further understanding to be gained about the trade-off between travel distance and venue capacity violations. What is the impact of the assumption that the starting round is given for each league? What is the impact of the size of the leagues? What is the impact of the HAP set that is used in the leagues? Third, it would be interesting to consider that league sizes should be between some given minimum and maximum number of teams, instead of our assumption of given league sizes. Indeed, in practice, the size of a league may be considered somewhat flexible, in particular in youth settings where there is no formal system of promotion or relegation, or comparison between the leagues. Finally, while we only focused on presenting a set of nondominated solutions to choose from, the practitioner will eventually still have to pick one solution. In case of a bi-objective problem with a convex Pareto front, a knee point (or elbow point in case of minimization) is often a preferred trade-off solution (Deb & Gupta, 2011). More in general, one could pick the solution closest to the (generally unattainable) ideal point (Marler & Arora, 2004). We point out that there is a lot of literature guiding the practitioner in selecting the best compromise solution for all stakeholders, including e.g. a fuzzy satisfying approach (see e.g. Gazijahani, Ajoulabadi, Ravadanegh, and Salehi (2020)).

Disclosure of interest

The authors report there are no competing interests to declare.

References

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., & Salomon, L. (2021). Performance indicators in multiobjective optimization. *European Journal of Operational Research*, *292*(2), 397–422.

Barone, L., While, L., Hughes, P., & Hingston, P. (2006). Fixture-scheduling for the Australian Football League using a multi-objective evolutionary algorithm. In *IEEE int. conf. on evolutionary computation* (pp. 954–961).

Bean, J. C., & Birge, J. R. (1980). Reducing travelling costs and player fatigue in the National Basketball Association. *Interfaces*, *10*(3), 98–102.

Berry, A., & Vamplew, P. (2005). The combative accretion model–multiobjective optimisation without explicit Pareto ranking. In *Proceedings of the 3^{th} int. conf. on evolutionary multi-criterion optimization* (pp. 77–91).

Borgonjon, T., & Maenhout, B. (2022). A two-phase Pareto front method for solving the bi-objective personnel task rescheduling problem. *Computers & Operations Research*, *138*, 105624.

Briskorn, D. (2008). Feasibility of home away pattern sets for round robin tournaments. *Operations Research Letters*, *36*(3), 283–284.

Burrows, W., & Tuffley, C. (2015). Maximising common fixtures in a round robin tournament with two divisions. *Australasian Journal of Combinatorics*, *63*(1), 153–169.

Çavdaroğlu, B., & Atan, T. (2019). Integrated break and carryover minimization problem in round robin tournament. In *Proceedings of the 7^{th} int. conf. on mathematics in sport* (pp. 25–32).

Chircop, K., & Zammit-Mangion, D. (2013). On ϵ-constraint based methods for the generation of Pareto frontiers. *J. Mech. Eng. Autom*, *3*(5), 279–289.

Craig, S., While, L., & Barone, L. (2009). Scheduling for the national hockey league using a multi-objective evolutionary algorithm. In *Australasian joint conference on artificial intelligence* (pp. 381–390).

Davari, M., Goossens, D., Beliën, J., Lambers, R., & Spieksma, F. C. (2020). The multi-league sports scheduling problem, or how to schedule thousands of matches. *Operations Research Letters*, *48*(2), 180–187.

Deb, K., & Gupta, S. (2011). Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering Optimization*, *43*(11), 1175–1204.

Demir, E., Bektaş, T., & Laporte, G. (2014). The bi-objective pollution-routing problem. *European Journal of Operational Research*, *232*(3), 464–478.

Duarte, A., & Ribeiro, C. (2008). Referee assignment in sports leagues: approximate and exact multi-objective approaches. In *19^{th} International Conference on Multiple Criteria Decision Making* (pp. 58–60).

Durán, G., Durán, S., Marenco, J., Mascialino, F., & Rey, P. A. (2019). Scheduling Argentina's professional basketball leagues: A variation on the travelling tournament problem. *European Journal of Operational Research*, *275*(3), 1126–1138.

Easton, K., Nemhauser, G., & Trick, M. (2001). The traveling tournament problem description and benchmarks. In T. Walsh (Ed.), *Principles and practice of constraint programming — cp 2001* (pp. 580–584). Berlin, Heidelberg: Springer.

Fowler, P., Duffield, R., & Vaile, J. (2015). Effects of simulated domestic and international air travel on sleep, performance, and recovery for team sports. *Scandinavian Journal of Medicine & Science in Sports*, *25*(3), 441–451.

Gazijahani, F. S., Ajoulabadi, A., Ravadanegh, S. N., & Salehi, J. (2020). Joint energy and reserve scheduling of renewable powered microgrids accommodating price responsive demand by scenario: A risk-based augmented epsilon-constraint approach. *Journal of Cleaner Production*, *262*, 121365.

Goossens, D., & Spieksma, F. (2011). Breaks, cuts, and patterns. *Operations Research Letters*, *39*, 428 - 432.

Goossens, D., & Spieksma, F. C. (2012). Soccer schedules in Europe: an overview. *Journal of*

Scheduling, *15*, 641–651.

Goossens, D., Yi, X., & Van Bulck, D. (2020). Fairness trade-offs in sports timetabling. In *Science meets sports: When statistics are more than numbers* (pp. 213–244).

Haimes, Y. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, *1*(3), 296–297.

Horbach, A. (2010). A combinatorial property of the maximum round robin tournament problem. *Operations Research Letters*, *38*(2), 121–122.

Jensen, M. T. (2003). Reducing the run-time complexity of multiobjective eas: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(5), 503–515.

Ji, X., & Mitchell, J. E. (2005). Finding optimal realignments in sports leagues using a branch-and-cut-and-price approach. *Int. Journal of Operational Research*, *1*(1-2), 101–122.

Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. *Naval Research Logistics Quarterly*, *3*(4), 253–258.

Kyngäs, J., Nurmi, K., Kyngäs, N., Lilley, G., Salter, T., & Goossens, D. (2017). Scheduling the Australian Football League. *Journal of the Operational Research Society*, *68*(8), 973-982.

Li, M., Davari, M., & Goossens, D. (2023). Multi-league sports scheduling with different leagues sizes. *European Journal of Operational Research*, *307*(1), 313–327.

Li, M., & Goossens, D. (2022). Grouping and timetabling for multi-league sports competitions. In *Proceedings of the 13th int. conf. on the Practice and Theory of Automated Timetabling (PATAT), volume III* (pp. 192–194).

Lust, T., & Tuyttens, D. (2014). Variable and large neighborhood search to solve the multiobjective set covering problem. *Journal of Heuristics*, *20*(2), 165–188.

Macdonald, B., & Pulleyblank, W. (2014). Realignment in the NHL, MLB, NFL, and NBA. *Journal of Quantitative Analysis in Sports*, *10*(2), 225–240.

Marler, R., & Arora, J. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*, 369-395.

Mavrotas, G. (2009). Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, *213*(2), 455–465.

Mesquita-Cunha, M., Figueira, J. R., & Barbosa-Póvoa, A. P. (2023). New ϵ-constraint methods for multi-objective integer linear programming: A Pareto front representation approach. *European Journal of Operational Research*, *306*(1), 286-307.

Mitchell, J. E. (2003). Realignment in the National Football League: Did they do it right? *Naval Research Logistics*, *50*(7), 683–701.

Miyashiro, R., Iwasaki, H., & Matsui, T. (2003). Characterizing feasible pattern sets with a minimum number of breaks. In *Proceedings of the 4^{th} int. conf. on the Practice And Theory of Automated Timetabling* (pp. 78–99).

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, *5*(1), 32–38.

Recalde, D., Severín, D., Torres, R., & Vaca, P. (2016). Balanced partition of a graph for football team realignment in Ecuador. In *International symposium on combinatorial optimization* (pp. 357–368).

Recalde, D., Severín, D., Torres, R., & Vaca, P. (2018). An exact approach for the balanced k-way partitioning problem with weight constraints and its application to sports team realignment. *Journal of Combinatorial Optimization*, *36*(3), 916–936.

Saltzman, R. M., & Bradford, R. M. (1996). Optimal realignments of the teams in the National Football League. *European Journal of Operational Research*, *93*(3), 469–475.

Schönberger, J. (2015). Scheduling of sport league systems with inter-league constraints. In

Proceedings of the 5^{th} int. conf. on mathematics in sport (pp. 171–176).

Schönberger, J. (2017). The championship timetabling problem-construction and justification of test cases. In *Proceedings of the 6^{th} int. conf. on mathematics in sport* (pp. 330–339).

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application.* John Wiley and Sons.

Suman, B., & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, *57*(10), 1143-1160.

Toffolo, T. A., Christiaens, J., Spieksma, F. C., & Vanden Berghe, G. (2019). The sport teams grouping problem. *Annals of Operations Research*, *275*(1), 223–243.

Van Bulck, D., & Goossens, D. (2020). On the complexity of pattern feasibility problems in time-relaxed sports timetabling. *Operations Research Letters*, *48*(4), 452-459.

Van Bulck, D., Goossens, D., Schönberger, J., & Guajardo, M. (2020). RobinX: a three-field classiffication and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, *280*(2), 568–580.

While, L., & Barone, L. (2007). Super 14 rugby fixture scheduling using a multi-objective evolutionary algorithm. In *IEEE Symposium on Computational Intelligence in Scheduling* (pp. 35–42).

While, L., & Kendall, G. (2014). Scheduling the English football league with a multi-objective evolutionary algorithm. In *Int. conf. on parallel problem solving from nature* (pp. 842–851).

Wright, M. (2018). Scheduling an amateur cricket league over a nine-year period. *Journal of the Operational Research Society*, *69*(11), 1854–1862.

Wright, M. B. (2009). 50 years of OR in sport. *Journal of the Operational Research Society*, *60*(sup1), S161-S168.

Xu, Y., & Qu, R. (2011). Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighbourhoods. *Journal of the Operational Research Society*, *62*(2), 313–325.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117-132.

Appendix

Table A1.: Quality indicators for the Pareto front approximations on medium-scale instances. Arrows indicate whether a higher (↑) or lower (↓) value is better.

Instance	UNFR (↑)			HV (↑)			SP (↓)			Time (s)		
ID	TCM	-I	-IS	TCM	-I	-IS	TCM	-I	-IS	TCM	-I	-IS
M1-1	0.750	1.000	0.500	0.413	0.327	0.392	0.340	0.233	0.192	564	874	7
M1-2	0.800	0.800	0.400	0.829	0.823	0.797	0.311	0.302	0.214	442	640	6
M1-3	0.667	0.833	0.667	0.664	0.630	0.424	0.137	0.194	0.261	593	712	8
M1-4	0.800	0.800	0.800	0.693	0.611	0.553	0.199	0.190	0.192	732	1102	10
M1-5	0.833	1.000	0.500	0.933	0.933	0.927	0.221	0.219	0.186	689	1036	13
M2-1	0.545	0.636	0.636	0.511	0.439	0.584	0.080	0.097	0.105	985	1573	24
M2-2	0.667	0.750	0.417	0.503	0.512	0.527	0.077	0.061	0.092	1211	1915	47
M2-3	0.444	0.667	0.444	0.553	0.580	0.469	0.111	0.096	0.210	1167	2030	31
M2-4	0.500	0.800	0.300	0.503	0.566	0.554	0.182	0.087	0.091	843	1414	61
M2-5	0.800	0.900	0.500	0.436	0.590	0.519	0.339	0.167	0.205	845	1260	43
M3-1	0.519	0.741	0.296	0.374	0.408	0.369	0.116	0.058	0.074	3051	5066	87
M3-2	0.297	0.514	0.541	0.384	0.420	0.403	0.036	0.033	0.030	2936	5579	140
M3-3	0.268	0.707	0.390	0.349	0.447	0.378	0.079	0.036	0.039	3408	6129	236
M3-4	0.273	0.576	0.606	0.527	0.498	0.561	0.043	0.034	0.035	1928	3700	89
M3-5	0.514	0.649	0.514	0.427	0.458	0.431	0.079	0.056	0.044	2297	4446	101
M4-1	0.455	0.727	0.333	0.420	0.450	0.432	0.074	0.046	0.035	8312	13174	322
M4-2	0.244	0.578	0.511	0.420	0.457	0.457	0.037	0.028	0.034	8345	14328	592
M4-3	0.262	0.786	0.286	0.464	0.467	0.490	0.049	0.030	0.034	6995	12243	781
M4-4	0.467	0.633	0.400	0.404	0.417	0.389	0.082	0.062	0.070	6862	11549	380
M4-5	0.234	0.617	0.404	0.406	0.402	0.425	0.058	0.036	0.038	9065	16960	961
Avg.	0.517	0.736	0.472	0.511	0.522	0.504	0.133	0.103	0.109	3064	5287	197

Table A2.: Quality indicators for the Pareto front approximations on large-scale instances. Arrows indicate whether a higher (↑) or lower (↓) value is better.

Instance	UNFR (↑)			HV (↑)			SP (↓)			Time (s)		
ID	TCM	-I	-IS	TCM	-I	-IS	TCM	-I	-IS	TCM	-I	-IS
L1-1	0.281	0.469	0.656	0.359	0.424	0.363	0.132	0.048	0.064	13785	22489	623
L1-2	0.220	0.659	0.439	0.329	0.440	0.432	0.113	0.038	0.052	13496	26913	876
L1-3	0.375	0.675	0.425	0.397	0.434	0.431	0.072	0.038	0.061	13717	27686	624
L1-4	0.488	0.610	0.488	0.376	0.432	0.436	0.077	0.057	0.039	10479	15917	423
L1-5	0.227	0.614	0.432	0.389	0.444	0.382	0.045	0.034	0.030	13309	23485	736
L2-1	0.632	0.842	0.263	0.500	0.575	0.444	0.094	0.069	0.087	6833	11320	111
L2-2	0.455	0.523	0.636	0.609	0.643	0.658	0.052	0.039	0.034	7978	13223	262
L2-3	0.346	0.577	0.577	0.399	0.438	0.432	0.100	0.056	0.076	8033	14764	191
L2-4	0.194	0.516	0.516	0.313	0.447	0.371	0.178	0.067	0.050	9671	17157	254
L2-5	0.514	0.571	0.457	0.562	0.575	0.527	0.053	0.030	0.037	8469	14962	251
L3-1	0.250	0.550	0.550	0.391	0.483	0.540	0.102	0.073	0.080	11101	17683	299
L3-2	0.543	0.600	0.457	0.507	0.551	0.452	0.058	0.036	0.053	10074	16702	277
L3-3	0.739	0.826	0.304	0.470	0.448	0.451	0.140	0.060	0.078	13113	22652	290
L3-4	0.348	0.565	0.565	0.365	0.492	0.512	0.184	0.075	0.078	10430	17377	292
L3-5	0.588	0.912	0.147	0.429	0.481	0.531	0.146	0.049	0.053	10688	17769	325
L4-1	0.720	0.880	0.200	0.523	0.569	0.531	0.087	0.061	0.074	12478	21689	430
L4-2	0.533	0.767	0.333	0.405	0.587	0.554	0.188	0.060	0.093	14466	25388	513
L4-3	0.400	0.650	0.550	0.478	0.540	0.624	0.154	0.079	0.109	13581	25338	473
L4-4	0.640	0.800	0.320	0.418	0.446	0.470	0.121	0.079	0.107	15189	26001	470
L4-5	0.345	0.483	0.552	0.438	0.426	0.454	0.148	0.061	0.063	14829	25411	666
Avg.	0.442	0.654	0.443	0.433	0.494	0.480	0.112	0.055	0.066	11586	20196	419