sustainability

MDPI

Article

Variable Neighborhood Search for Major League Baseball Scheduling Problem

Yun-Chia Liang [1,*], Yen-Yu Lin [1], Angela Hsiang-Ling Chen [2,*] and Wei-Sheng Chen [2]

[1] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 320, Taiwan; s1045422@mail.yzu.edu.tw
[2] Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taoyuan 320, Taiwan; s10624254@cycu.edu.tw
* Correspondence: ycliang@saturn.yzu.edu.tw (Y.-C.L.); achen@cycu.edu.tw (A.H.-L.C.)

Abstract: Modern society pays more and more attention to leisure activities, and watching sports is one of the most popular activities for people. In professional leagues, sports scheduling plays a very critical role. To efficiently arrange a schedule while complying with the relevant rules in a sports league has become a challenge for schedule planners. This research uses Major League Baseball (MLB) of the year 2016 as a case study. The study proposed the Variable Neighborhood Search (VNS) algorithm with different coding structures to optimize the objective function—minimize the total travelling distance of all teams in the league. We have compared the algorithmic schedules with the 2016 and 2019 MLB regular-season schedules in the real-world case for its performance evaluation. The results have confirmed success in reducing the total travelling distances by 2.48% for 2016 and 6.02% in 2019 while lowering the standard deviation of total travelling distances by 7.06% for 2016.

Keywords: sports scheduling; metaheuristics; optimization; Major League Baseball

check for updates

Citation: Liang, Y.-C.; Lin, Y.-Y.; Chen, A.H.-L.; Chen, W.-S. Variable Neighborhood Search for Major League Baseball Scheduling Problem. *Sustainability* **2021**, *13*, 4000. https://doi.org/10.3390/su13074000

Academic Editors: Franklin G. Mixon

Received: 17 February 2021
Accepted: 1 April 2021
Published: 3 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Professional sports have already become a dynamic and fast-growing industry. Regardless of national and international tournaments, millions of people across the world follow game schedules. Fans pay billions to attend; television networks pay billions to broadcast. In sports, all value chains contribute to significant economic benefits from athletes, teams and leagues, business and the media, and on host cities and countries. This important contribution may justify the promotion of sports management as an independent academic discipline. Within this discipline, several fundamental issues are often associated with sports leagues' operation arise. Most sports leagues organize tournaments via a competitive season of a predefined length; league competitions follow many complex league regulations and restrictions, with teams playing each other at a specific time and place. Such arrangements are often very time-consuming, and the sports business often operates to maximize profits.

Nevertheless, there have been doubts about this profit-oriented operation in team sports. Questions are raised in recognition of the growing environmental problems regarding resource allocation, social and behavioral issues, e.g., talent distribution, athletic quality, the design of the competition, and competition balance within leagues. All pose different challenges to align to the principles of sustainability fully. Linked to the empirical verification, this paper focuses on determining a highly attractive, justifiable, and sustainable schedule beneficial for the competition balance of a Major League Baseball (MLB) case.

2. Literature Review

The demand for a more competitive balance within a league has been an exciting research area over the past decades. There have been extensive review papers written to show successes in theory and practice [1–7]. In principle, many aspects are considered

in various applications. As in the case of the travelling tournament problem (TTP), the real-life situation for Major League Baseball aims to optimize the regular-season schedule at the minimum total travelled distance [8–21]. Similarly, with the travelling umpire problem (TUP), the primary objective is to minimize umpire travel [21–26]. In another case, the goal is to minimize the total number of breaks [2,6,12,27–30]. The ranking-balancedness [6,31], and the carry-over effect [6,32] are problems regarding game fairness. Lastly, in the case of venue management, the maximization of venue availability is tackled by [33].

Among the above specific scheduling topics, most sports leagues tackle two fundamental variations: a round-robin (RR) tournament and a knock-out (KO) tournament. In RR tournaments, all participants play against each other. In KO tournaments, all losers in each round are eliminated, and all winners are promoted to the next game. Many works of literature focused on the round-robin schedule format. Most sports leagues play double rounds [2,8,12,19,34–36], some play single [14,17,32], few triple and quadruple rounds. Other works discussed the knock-out setting [37–41]. As far as we know, few have considered multiple leagues or divisions [42,43]. The differences in league format and number of teams result in different matches played per group in other competitions.

Since practical sports scheduling often leads to many conflicting constraints, the computational methods for creating schedules applicable to real sports leagues have been challenging. Several metaheuristic algorithms have been popular and widely considered in sports scheduling. For example, many researchers used Simulated Annealing (SA) [9,44–47] and the Tabu Search (TS) [4,11,48] to solve travelling tournament problems. Chen et al. [10] and Uthus et al. [49] used the Ant Colony Optimization (ACO) hybridizing with a forward checking and conflicting-directed backjumping algorithm. Januario and Urrutia [17], Khelifa et al. [19], and Guiqian [50] all used a Genetic Algorithm (GA) related hybrid methods to improve the best solutions. Biajoli and Lorena [51] combined the clustering search approach and Variable Neighborhood Search (VNS). Khelifa and Boughaci [15,16] also proposed a VNS and its hybrid form to solve the optimal minimum travelling distance. Their later attempt focused on the mirrored version of the travelling tournament problem with reversed venues and proposed a VNS with Harmony Search (V-HS).

Among the scheduling of popular sports leagues, the baseball league captures the fundamental difficulties in problem formulation, especially Major League Baseball in the United States. Robinson [37] used a linear program to eliminate baseball teams from the playoff. Russell and Leung [52] set a Texas Baseball League schedule with minimizing travel costs as their goal. Nemhauser and Trick [34] applied a combination of integer programming and constraint programming to find feasible schedules for MLB. Anagnostopoulos et al. [9] modified Easton et al. [8] model for MLB and used a simulated annealing approach to obtain schedules at minimum travel distances. Hoshino and Kawarabayashi [53] applied Dijkstra's Algorithm to generate the optimal multiple rounds schedules for Nippon Professional Baseball. Hoshino and Kawarabayashi [54] considered a double round-robin tournament schedule that incorporated geographical information to determine optimal or nearly optimal solutions at the minimum total travel distance for the MLB. Trick and Yildiz [22,23] and Trick et al. [24] all focused on the travelling umpire problem of MLB. They considered the hybrid use of various heuristics to minimize the travel of the umpires. Guiqian [50] and Ye [55] used genetic algorithms and tabu search to optimize Major League Baseball schedules. Hoshino and Kawarabayashi [54] aimed to find the shortest path by using heuristics based on the Graph Theory for the Nippon Professional Baseball. Lin and Chen [56] applied the concept and methods of "efficiency and productivity" to deal with match-fixing players for the Chinese Professional Baseball League. Ko et al. [57] and Kim [58] discussed the Korea Baseball League with several constraints to obtain an efficient annual match schedule.

At present, when discussing sports scheduling problems, most pieces of literature have proposed effective solutions to the problem instances [3,8] and the scheduling of small-scale sports events [1,18,34]. However, in some professional sports, the size of leagues, often between 20 and 30 groups, is a large-scale sports scheduling problem, but few studies have in-depth discussions on this issue. Therefore, this study deals with the United States Major League Baseball scheduling problem, a large-scale professional sports scheduling, and aims to minimize the team's total travelling distances. Such a problem belongs to the NP-hard problem, in which various soft and hard constraints of the MLB must be met.

3. Problem Description and Formulation

The MLB schedule can be regarded as a Constraints Satisfaction Problem (CSP) in scheduling problems. Its schedule characteristics are unique from other sports. The factors to be considered are complex factors such as cross-league events, home-and-away matches, and series matches. Under the conditions that they must fully comply with the league's system rules (Hard Constraints), the season schedule is then completed. This study considers the factors based on the actual 2016 regular program of the season and lists them below.

First, it focuses the season dates on the regular season of the major leagues in the United States. In MLB, there are two leagues: the American League (AL) with 15 teams and the National League (NL) with 15 teams. Each league includes three divisions—East, Central, and West, and each division includes five teams. Each team plays 162 games throughout the season where the total number of games in the same league is 142, and the total number of inter-league matches is 20. These games are then divided into 52 series among the same club and inter-league tournaments. In all, the entire league plays 2430 games in total. Except for certain factors where there is a need for extra play, each team plays one game per day.

In MLB, each team has its venue at its home city and plays at the venue of either one of the two teams in the competition. The home and away games have each play 26 series, equivalent to 81 games; each club comprises two- to four-game series according to the division's restrictions. That means a team may play with an opponent for two, three or four consecutive games. However, teams cannot play against the same opponent for over two series in the same venue. No single team with either home stands or road trips can play over three series consecutively without an off day.

The series of games can be arranged according to the following rules: (a) the teams need to play with each of the other teams in the same division for 6 series, half home series and a half road series, equivalent to 19 games in total. (b) The teams play with each of the other teams in the same league, but in different divisions for two series, one home and one away, equal to six or seven games in total. (c) The teams play with other teams in the different league 8 series (20 games) in total.

The final arrangement is for traditional popular events such as the Subway series (New York Yankees vs. New York Mets), the Citrus series (Tampa Bay Rays vs. Miami Marlins), the Windy City Showdown series (Chicago Cubs vs. Chicago White Sox), the Battle for Ohio series (Cleveland Indians vs. Cincinnati Reds), the I-70 series (Kansas City Royals vs. St. Louis Cardinals), the Border Battle series (Minnesota Twins vs. Milwaukee Brewers), the Cal Freeway series (Anaheim Angels vs. Los Angeles Dodgers), the Bay Bridge series (Oakland Athletics vs. San Francisco Giants), and the Beltway series (Baltimore Orioles vs. Washington Nationals). Each series above plays one home series and one away series, and the purpose of such series is to stimulate the market, increase topicality and game attraction.

This study considers a situation where there are many competing teams and series. Let T present a set of the teams, and S denote a set of the series. The goal is to optimize the team's total distance of travel when the constraints are met. The proposed mathematical model is as follows:

$$Min \quad \sum_i \sum_j \sum_k \sum_s d_{jk} X_{ijks} \tag{1}$$

$$st \quad \sum_s \sum_{j \neq i} V_{ijs} = \sum_s \sum_{j \neq i} W_{ijs} \qquad \forall i \in T \tag{2}$$

$$\sum_s \left[\sum_{j \neq i} V_{ijs} + \sum_{j \neq i} W_{ijs} \right] = S_{max} \qquad \forall i \in T \tag{3}$$

$$H_{iS} + H_{i,S+1} + H_{i,S+2} < C \qquad \forall i \in T,\ s \in \mathrm{S} - 2 \tag{4}$$

$$A_{iS} + A_{i,S+1} + A_{i,S+2} < C \qquad \forall i \in T,\ s \in \mathrm{S} - 2 \tag{5}$$

$$H_{iS} = X_{iiis} \qquad \forall i \in T,\ s \in \mathrm{S} \tag{6}$$

$$A_{iS} = X_{ijks} \qquad \forall i \in T,\ s \in \mathrm{S},\ i \neq j,\ i \neq k \tag{7}$$

$$\sum_S V_{ijs} = D \qquad \forall i, j \in Div \tag{8}$$

$$\sum_S V_{ijs} = E \qquad \forall i, j \in League - Div \tag{9}$$

$$\sum_S V_{ijs} \leq F \qquad \forall i, j \in T - League \tag{10}$$

$$\sum_S \sum_{(i,j) \in Nonlea} V_{ijs} = G \qquad \forall (i,j) \in DifferentLeague \tag{11}$$

$$V_{ijs} + V_{ijs+1} \leq 1 \qquad \forall i \in T,\ \forall j \in T,\ j \neq i, \forall s \in S \tag{12}$$

$$\sum_{j \neq i} V_{ijs} + W_{ijs+1} \leq 2 \cdot X_{iijs} + \left[\left(2 - \sum_{j \neq i} V_{ijs} - W_{ijs+1} \right) \cdot M \right] \tag{13}$$

$$\sum_{j \neq i} V_{ijs} + W_{ijs+1} \geq 2 \cdot X_{iijs} + \left[\left(2 - \sum_{j \neq i} V_{ijs} - W_{ijs+1} \right) \cdot M \right] \tag{14}$$

$$W_{ijs} + \sum_{j \neq i} V_{ijs+1} \leq 2 \cdot X_{ijis} + \left[\left(2 - W_{ijs} - \sum_{j \neq i} V_{ijs+1} \right) \cdot M \right] \tag{15}$$

$$W_{ijs} + \sum_{j \neq i} V_{ijs+1} \geq 2 \cdot X_{ijis} + \left[\left(2 - W_{ijs} - \sum_{j \neq i} V_{ijs+1} \right) \cdot M \right] \tag{16}$$

$$W_{ijs} + W_{ijs+1} \leq 2 \cdot X_{ijks} + \left[(2 - W_{ijs} - W_{ijs+1}) \cdot M \right] \tag{17}$$

$$W_{ijs} + W_{ijs+1} \geq 2 \cdot X_{ijks} + \left[(2 - W_{ijs} - W_{ijs+1}) \cdot M \right] \tag{18}$$

Equation (1) aims to minimize the teams' total distance travelled in a regular season. Equations (2) and (3) indicate the number of home series, away series, and all series. Equations (4) and (5) show the maximum number of consecutive home or away series; this implies that a single team cannot play over three series in consecutive home or away games. Equation (6) describes team i plays at home in the s and $s + 1$ series, while Equation (7) depicts team i plays on the road in the s and $s + 1$ series. Equation (8) limits the number of times a single team can play another team also belonging to the same division. Equation (9) defines the number of matchups for a single team in distinct divisions of the same league in the regular season. Equation (10) sets the maximum number of matchups in the interleague play series for any two teams, while Equation (11) represents the total number of matchups in the interleague play series for a team.

Equation (12) defines that the same matchup of any two teams cannot play over one series in the same venue consecutively. For instance, when Team A and Team B play against each other at Team A's home venue, this matchup cannot be played for two consecutive series in the same arena. Equations (13)–(18) depict three different movements for a team. Equations (13) and (14) denote flows that a team initially plays at its venue, then a road game. Equations (15) and (16) are the team on the road that later gets to play a home game. Equations (17) and (18) are the away team going to the next away game (in a different city and arena). Equations (15)–(18) use the same concept designing movement restriction in the mathematical formulas. All decision variables such as x_{ijks}, W_{ijs}, H_{is}, A_{is}, and V_{ijs} are binary, i.e., two possible values 0 or 1.

4. Methodology

Given the teams' geography and the MLB regulations, our framework assumes all 30 teams, regardless of whether these teams start the season with home games or away games, start to move from their home city. The travel distance is 0 if the first series is a home series; otherwise, the distance between the home venue and the away field for the road game is calculated. To determine the distances, we use Google Maps to find the latitude and longitude of each team's home field and calculates the distance among their home fields. From there, we then make a distance matrix to provide a basis for calculating the travel cost. Furthermore, in the MLB scheduling problem, each city is visited more than once, and it is unnecessary to visit all cities.

To solve the MLB scheduling problem, we consider the variable neighborhood search framework proposed by Mladenović and Hansen in 1997. When searching for solutions, the VNS uses a simple systematic search method to change the neighborhood structure, expand the search range, jump out of the local optimum, and find the global optimum after searching through different neighborhoods. The difference from other global optimizer algorithms is that the variable neighborhood search method does not require complicated parameter settings. As Hansen and Mladenović [59] stated some desirable properties, the VNS exhibits a simple and straightforward principle, with flexible design, and provides good solutions for various problems within a reasonable time. Thus, the variable neighborhood search method is applied in this study. The details of the proposed VNS algorithm are described in the following sections.

Since the MLB scheduling problem is highly constrained, generating a feasible schedule is not an easy task. The procedure may be stuck with an infinite loop if only feasible solutions are accepted. Therefore, this study proposed a procedure that tries to satisfy as many constraints as possible and then fix the infeasible part through a repair mechanism. The procedure continues till a feasible solution that fully meets all constraints is found. Each team's matchplay list is divided into two categories based on the home and away series, including 26 opponents (teams) each. For the team to play against other teams, this arrangement has considered the home and away restrictions and the number of matches in the same division, in different divisions of the same league and across leagues. Figure 1 illustrates the conceptual framework of a home and away schedule.

4.1. Initial Schedule Generation

Figure 2 below depicts the initial schedule generation scheme. Starting from a randomly selected team, we set all the predefined rules and assigned the current team either in a homestand or an away stand. Afterward, we randomly select its opponent team. Suppose that team i is randomly selected to play in an away series, say the sth series, and then another team must be randomly selected as the home team for this series' match combination. Next, the match combination is copied to the selected competing team in the same series. Through this procedure, the arrangement of the 30 teams in the sth series can be completed. Similarly, the next series (s + 1) arrangement is completed until the 52 series are completed.

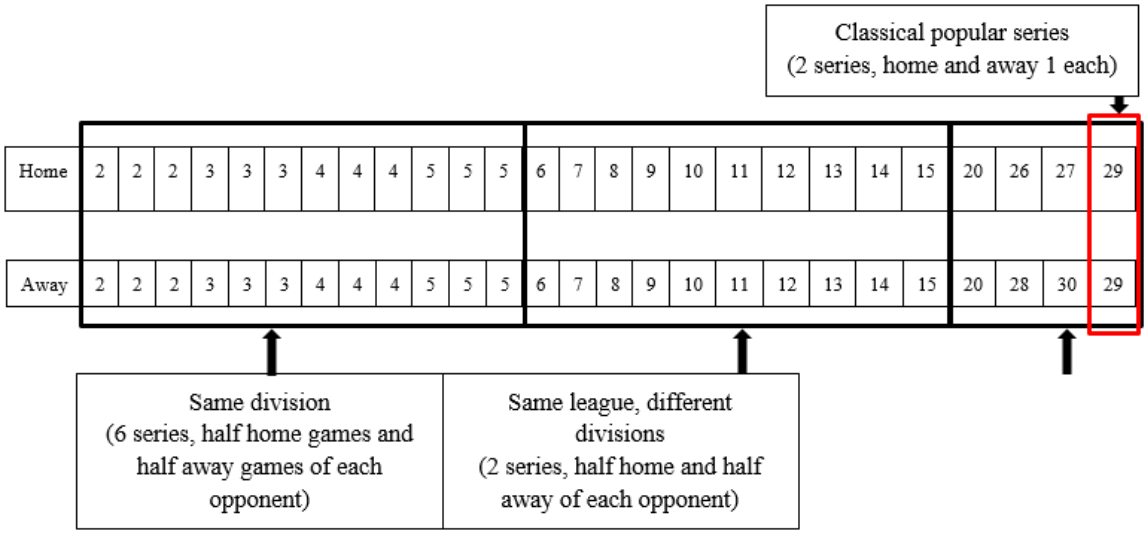


Figure 1. An illustration of an initial solution with a home and away schedule. Note: The numbers in each cell represent the teams.

The initial solution generation must satisfy two constraints: (1) Teams in the same battle combination must not play more than one series on the same venue in a row; (2) The number of consecutive home or away series matches cannot exceed three. Often and potentially possible, this initial schedule does not fully satisfy the constraints; the matching leads to situations where there is no team to choose from, showing an infinite loop. This study attempts to repair the initial infeasible schedule by combining the schedule restoration mechanisms proposed by [55], including the home-and-away escape method, release correction method, and series escape method.

Each team's combination is divided into two categories based on the home and away games, each including 26 series. If opponents for an away series are not selected, no feasible team to play and the home-and-away escape method is activated. That means the away series is switched to a home series to find the feasible opponent. If the feasible opponent is still not available after the switch, the procedure activates the series escape method. The release correction method involves the changes on multi-teams. For example, if for Team 11, opponents' potential options are Team 4 or Team 7. However, both Teams 4 and 7 have been selected by Teams 2 and 6, respectively, in the same series. Team 11 has no feasible option for opponents. If the battle combination of Teams 6 and 7 is released, i.e., Teams 6 and 7 no longer play with each other in this series, Team 7 can be matched with Team 11 to play in this series. The final schedule restoration method is the series escape method. Suppose the previous two restoration methods fail to find a feasible solution after a certain number of trials. In that case, the series escape method is activated and rearranges the current series (s) and two previous series ($s-1$, $s-2$). If the series escape method cannot generate a feasible schedule after a pre-specified number of trials, the schedule is still output as an infeasible one.

The violation of constraints has been reduced through the three restoration methods but may still not be a feasible solution. This study then employs a local search method 2-Opt to repair the initial infeasible schedule further. The 2-Opt is a method that is commonly applied to routing problems. If the sequence of series matches is considered a routing of nodes, the 2-Opt method is to disconnect two places in the sequence and then reverse the series between. An illustration of the 2-Opt method in Figure 3 considers 6 teams and 8 series for each team. The current schedule beginning the 2nd column from the right in Figure 3a denotes the number of consecutive home or away series constraints being violated. For example, in the current schedule for Team 1, the first 4 series are all away games that violate the constraints. Teams 2 and 5 show the same type of violation. Teams 4 and 6 playing for two consecutive series show another type of constraint violation. The

2-Opt method here disconnects Series 1 and 2, Series 7 and 8, respectively (as shown by the red crosses in Figure 3a). The order from the current Series 2 to 7 is reversed, as shown in Figure 3b. The total number of violations is successfully reduced from 5, in Figure 3a, to 2, as illustrated in Figure 3b.

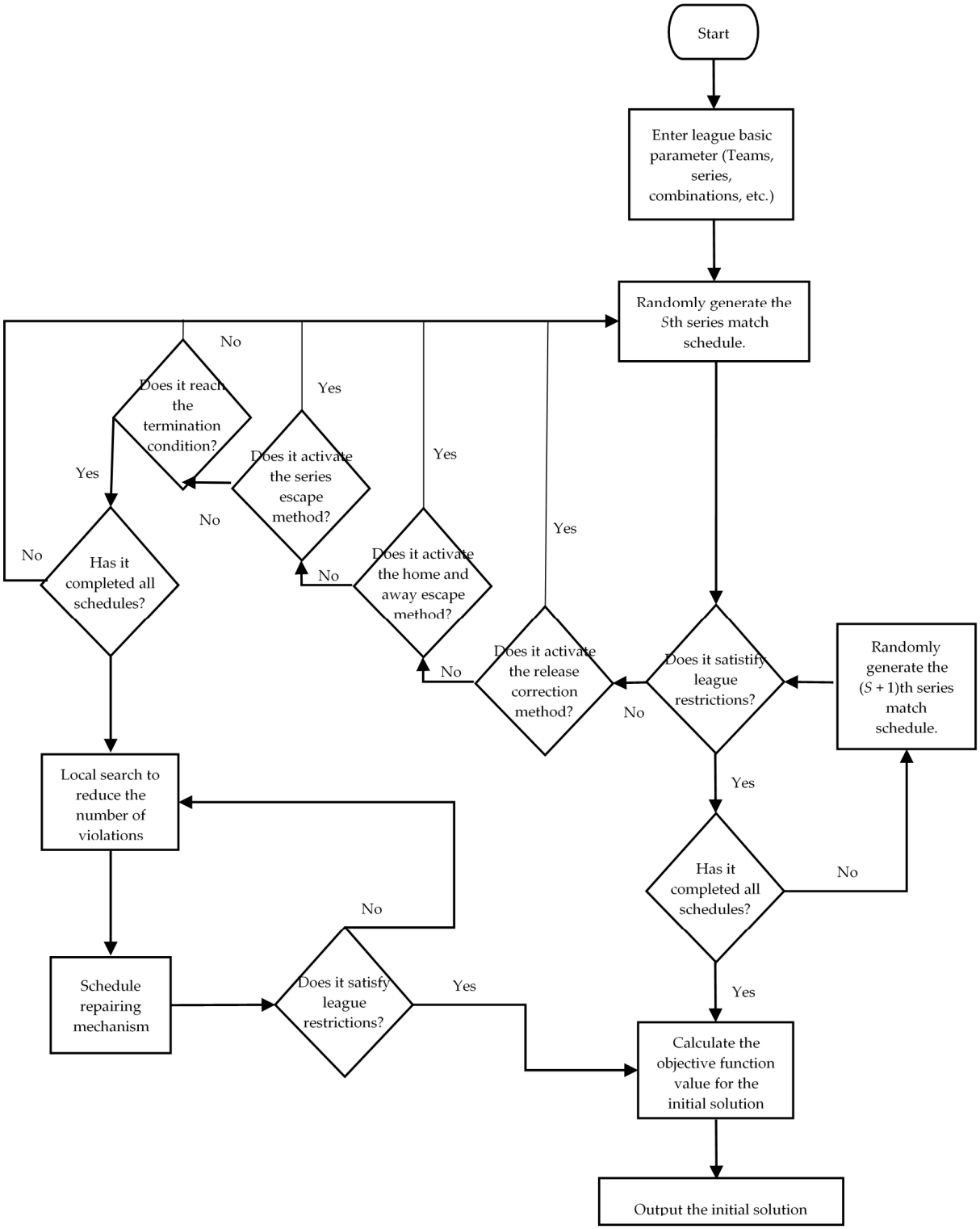


Figure 2. The procedure for generating the initial solution of the schedule.

Team \ Series	1	2	3	4	5	6	7	8	No. of Violations for Each Team	Total No. of Violations
1	**{5,1}**	**{2,1}**	**{3,1}**	**{4,1}**	{1,6}	{3,1}	{1,4}	{1,5}	1	5
2	{2,3}	{2,1}	{4,2}	**{2,6}**	**{2,3}**	**{2,4}**	**{2,5}**	{3,2}	1	
3	{2,3}	{5,3}	{3,1}	{3,5}	{2,3}	{3,1}	{3,6}	{3,2}	0	
4	**{6,4}**	**{6,4}**	{4,2}	{4,1}	{4,5}	{2,4}	{1,4}	{4,6}	1	
5	{5,1}	{5,3}	{5,6}	**{3,5}**	**{4,5}**	**{6,5}**	**{2,5}**	**{1,5}**	1	
6	**{6,4}**	**{6,4}**	{5,6}	{2,6}	{1,6}	{6,5}	{3,6}	{4,6}	1	

(**a**)

Team Series	1	7 (New 2)	6 (New 3)	5 (New 4)	4 (New 5)	3 (New 6)	2 (New 7)	8	No. of Violations for Each Team	Total No. of Violations
1	{5,1}	{1,4}	{3,1}	{1,6}	{4,1}	{3,1}	{2,1}	{1,5}	0	2
2	**{2,3}**	**{2,5}**	**{2,4}**	**{2,3}**	**{2,6}**	{4,2}	{2,1}	{3,2}	1	
3	{2,3}	{3,6}	{3,1}	{2,3}	{3,5}	{3,1}	{5,3}	{3,2}	0	
4	{6,4}	{1,4}	{2,4}	{4,5}	{4,1}	{4,2}	{6,4}	{4,6}	0	
5	{5,1}	**{2,5}**	**{6,5}**	**{4,5}**	**{3,5}**	{5,6}	{5,3}	{1,5}	1	
6	{6,4}	{3,6}	{6,5}	{1,6}	{2,6}	{5,6}	{6,4}	{4,6}	0	

(**b**)

Figure 3. An illustration of the 2-Opt local search procedure for repairing the schedule's initial solution (**a**) before the repairing, (**b**) after applying the 2-Opt method.

The steps mentioned above (the home-and-away escape method, release correction method, and series escape method, and the 2-Opt method) can reduce the violation of constraints but may still not assure a feasible schedule yet. In this case, a home-and-away swap method is proposed as a final step in the initial schedule generation in this study. The home-and-away swap method chooses one series that caused the violation and tries to see if it can be swapped with a different type of series with the same opponent. For example, in Figure 3b, Team 5 has 4 away series in a row. If the series {6,5} (i.e., Team 6 plays the home games, and Team 5 plays as the away team) can be swapped with the 6th series {5,6}, the new schedule for Team 5 will become {5,1}, {2,5}, {5,6}, {4,5}, {3,5}, {6,5}, {5,3}, {1,5} which has turned into a feasible schedule. The 2-Opt method and the home-and-away swap method are employed in turn until a feasible initial schedule is generated. Then the objective function value of the initial feasible solution is evaluated.

4.2. Schedule Optimization

This research refers to the basic schemes of VNS proposed by [60] to deal with the MLB scheduling problem. The factors that affect the solution quality and efficiency of the VNS include the design of neighborhoods, the order of neighborhood structures, and the conditions for changing the neighborhood structure. Since the best solution in a single neighborhood is not necessarily the best solution in other neighborhoods, it is necessary to explore different neighborhoods to avoid the local optimal solution. This section explains the shaking mechanism and neighborhood search mechanism. Figure 4 exhibits a flow chart of the proposed VNS algorithm for solving MLB scheduling problems.

The schedule optimization procedure begins with the initial solution generation method based on the generation method described in Section 4.1; the initial schedule acts as a single initial solution of the variable neighborhood search method. Once the initial solution is generated, a neighborhood structure is selected for solution exploration. A shaking mechanism is performed using the selected neighborhood structure to generate a neighboring solution randomly. The neighboring solution is used as a base point for the neighborhood search. The shaking mechanism is designed to change the base solution for each neighborhood search so that exploring the space is increased and getting stuck on local optimum can be prevented.

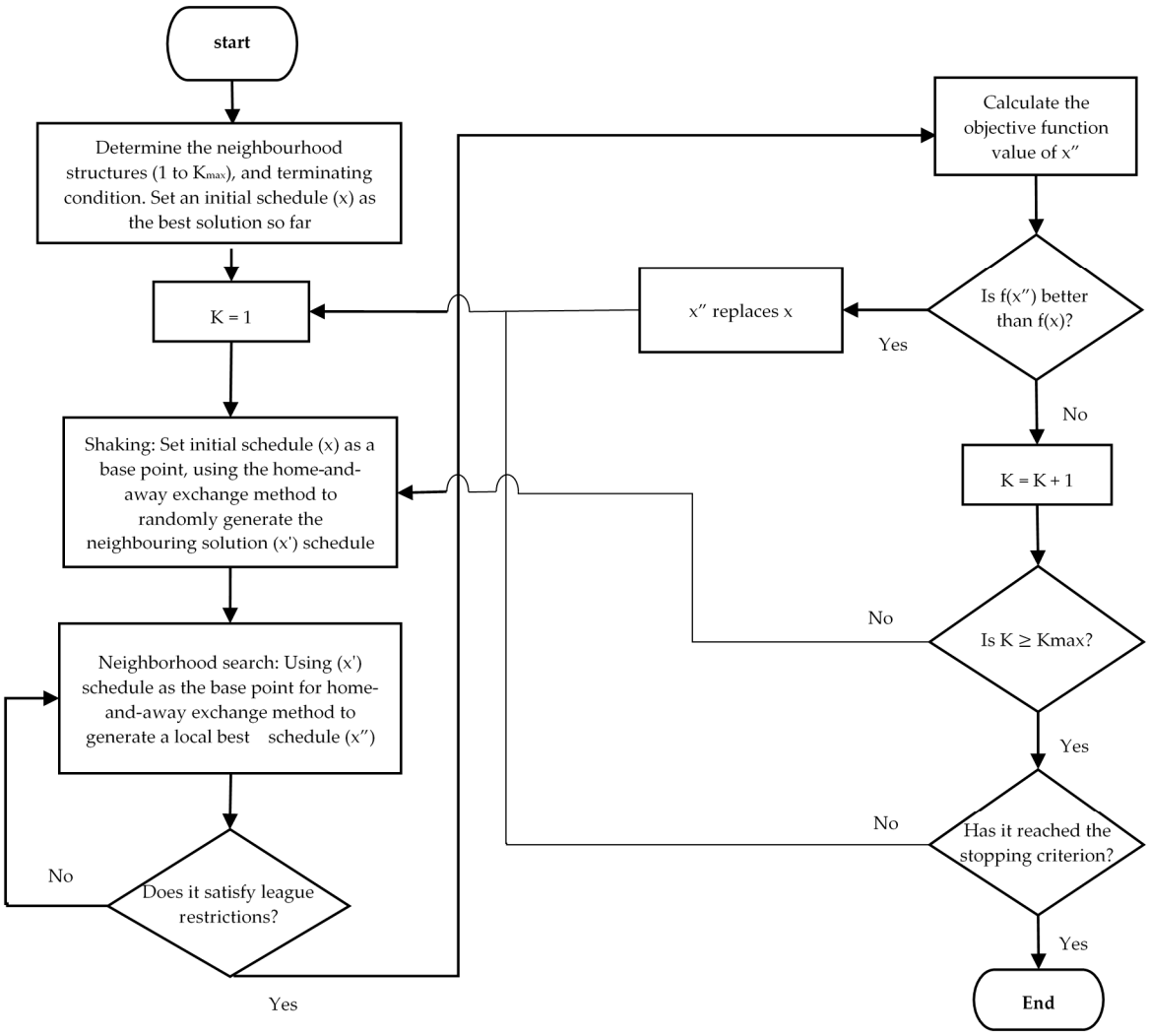


Figure 4. The flow chart of the Variable Neighborhood Search (VNS) algorithm for solving the Major League Baseball (MLB) scheduling problem.

This study proposes a home-and-away swap method as the move operator for the neighborhood. When the home-and-away swap method is executed, one of the 30 teams is selected randomly. As an object of exchange, the result of this shake changes the two teams' schedule. The home-and-away swap method causes four positions in the schedule to change, as highlighted by the red boxes and arrow in Figure 5. The neighborhood search mechanism uses a systematic search method to search for feasible neighboring solutions. In different neighborhoods, a different number of times the swap operation is applied. The number of swaps in each neighborhood and the sequence of different neighborhood structures are examined by parameter analysis in Section 5.1.

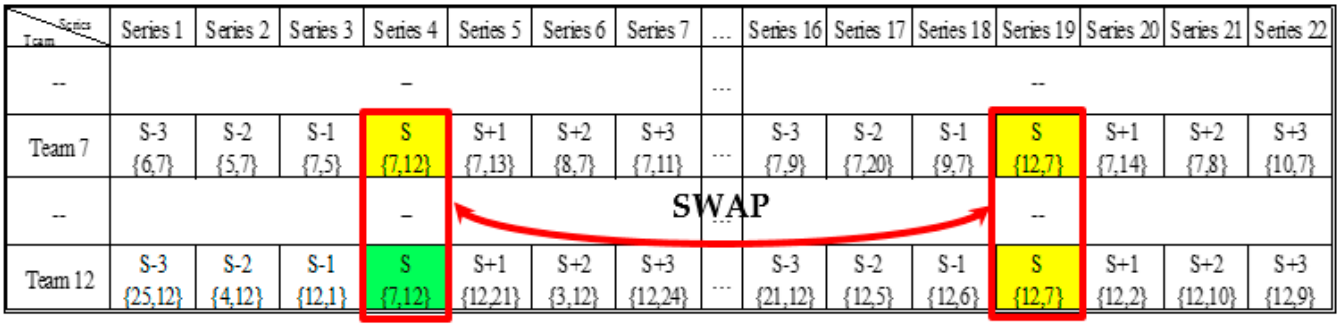


Figure 5. An illustrative example of the shaking mechanism.

This study proposes two versions, VNS-I and VNS-II, as swap coding methods to arrange the game combinations in the neighborhood search. For VNS-I, the two competing teams are based on the lines (e.g., the sth series) in the schedule, and the corresponding team must be in the same line (series); the aligned arrangement is used as the coding method for the swap operation. For VNS-II, the two teams in the schedule do not have to be aligned in the same series (e.g., the sth series) but can shift the corresponding position of plus or minus one position (e.g., the $(s + 1)$th series and the $(s - 1)$th series) based on the series. The corresponding position is selected from these three positions, the $(s + 1)$th series, the sth series, and the $(s - 1)$th series. After completing the neighborhood search, each team's MLB schedule is determined and satisfied with all league rules. Once the sequence of each team's series has been determined, the objective function values, the total travelling distance for all teams, can be calculated and determines if the new solution replaces the current base solution.

The final step is to arrange the MLB daily schedule based on a series of optimal schedules obtained above. After completing the series schedule, a single game must be scheduled according to the matches' series. Each series may consist of two consecutive games, three consecutive games, or four consecutive games. The daily scheduling steps are: First, read the optimized schedule from 1 to 52 series in order; then, use the number of matches to determine the remaining number of consecutive game combinations, and arrange for further consecutive games. Finally, determine whether the corresponding dates of the two teams in the game combination are the same. If it is the same, then the match is held; otherwise, an off-day can be arranged to be played at the same date for both teams.

5. Case Study Results and Discussions

This proposed VNS algorithm was coded using Microsoft Visual C ++ 2012. The computer environment is Inter® Core™ i7-2600 CPU 3.4 GHz, and the memory is 16 GB. The following sections introduce the case instance, discuss the parameter settings, and summarize the computational results.

5.1. Case Instance

The case study is based on the 2016 and 2019 regular seasons of Major League Baseball. As shown in Table 1, MLB consists of two leagues (American and National) with three divisions (East, Central, and West, respectively); each division includes five teams. Therefore, there are 30 teams in MLB, and each team plays 52 series (equivalent to 162 games) in a regular season, i.e., 780 series (equivalent to 2430 games) in the whole season overall. When Houston Astros switched to the West Division of the American League in 2013, the number of teams in each division is even; since then, there has not been any change.

Table 1. The list of teams in MLB.

League	Division	Team
American	East	Baltimore Orioles
		Boston Red Sox
		New York Yankees
		Tampa Bay Rays
		Toronto Blue Jays
	Central	Chicago White Sox
		Cleveland Indians
		Detroit Tigers
		Kansas City Royals
		Minnesota Twins
	West	Houston Astros
		Los Angeles Angels
		Oakland Athletics
		Seattle Mariners
		Texas Rangers
National	East	Atlanta Braves
		Miami Marlins
		New York Mets
		Philadelphia Phillies
		Washington Nationals
	Central	Chicago Cubs
		Cincinnati Reds
		Milwaukee Brewers
		Pittsburgh Pirates
		St. Louis Cardinals
	West	Arizona Diamondbacks
		Colorado Rockies
		Los Angeles Dodgers
		San Diego Padres
		San Francisco Giants

5.2. Parameter Settings

The performance of VNS is related to the setting of parameters such as the type of neighborhood structure, the number of neighboring solutions searched in each neighborhood, and the order of neighborhoods. Therefore, this section focus on the experimental design of parameters in VNS.

The size of the neighborhood structure (K_max) is set to three (i.e., N_1, N_2, N_3, respectively). The number of shaking each time is set to 1, 25, or 50, respectively, and the number of the neighboring solutions searched is 10, 50, or 100, respectively. The combination of the three neighborhoods is determined by the number of teams involved each time. Thus, the combinations of (N_1, N_2, N_3) consists of (1, 5, 10), (5, 10, 15), (10, 15, 20), (15, 20, 25), (10, 5, 1), (15, 10, 5), (20, 15, 10), and (25, 20, 15). There are two types of coding for the home-and-away swap method, VNS-I and VNS-II. Each setting is run 10 times using different random number seeds. Overall, there are $3 \times 3 \times 8 \times 2 = 144$

experiments equivalent to 1 run for parameter analysis. Lastly, the maximum number of iterations is set to 60,000 as a stopping criterion.

The appropriate parameter settings obtained are shown in the following table. The ANOVA results in Table 2 show that the performance is significantly affected by methods design (i.e., VNS-I and VNS-II). As we ran the experiments with the same number of neighborhoods, shakes, neighboring solutions, and neighborhood structures, VNS-II obtains overall better results than VNS-I. Furthermore, under the design of several parameter settings for VNS-1, only the number of shakings significantly affects its performance; on the other hand, for VNS-2, the number of neighboring solutions, the number of teams involved in N1, N2 and N3 all show significant effects on the solution quality (i.e., the total travelling distance). Finally, Table 3 summarizes the parameter setting of both versions of the proposed VNS algorithm.

Table 2. One-way ANOVA: Average Total Traveling Distance versus Different Parameters.

Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS Type	1	1.56322×10^{12}	1.56322×10^{12}	7807.68	0.000
Error	142	2.84306×10^{10}	2.00215×10^{8}		
Total	143	1.59165×10^{12}			
S = 14150 R-Sq = 98.21% R-Sq(adj) = 98.20%					
Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS-I_Shaking	2	2.81981×10^{9}	1.40990×10^{9}	16.36	0.000
Error	69	5.94718×10^{9}	8.61910×10^{7}		
Total	71	8.76699×10^{9}			
S = 9284 R-Sq = 32.16% R-Sq(adj) = 30.20%					
Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS-II_No. N. S. [1]	2	3.01533×10^{9}	1.50767×10^{9}	6.25	0.003
Error	69	1.66483×10^{10}	2.41279×10^{8}		
Total	71	1.96636×10^{10}			
S = 15533 R-Sq = 15.33% R-Sq(adj) = 12.88%					
Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS-II_N1	5	5.21903×10^{9}	1.04381×10^{9}	4.77	0.001
Error	66	1.44446×10^{10}	2.18857×10^{8}		
Total	71	1.96636×10^{10}			
S = 13366 R-Sq = 38.22% R-Sq(adj) = 35.49%					
Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS-II_N2	5	7.51511×10^{9}	2.50504×10^{9}	14.02	0.000
Error	66	1.21485×10^{10}	1.78654×10^{8}		
Total	71	1.96636×10^{10}			
Source	**DF**	**SS**	**MS**	***F*-Value**	***p*-Value**
VNS-II_N3	5	3.23669×10^{9}	6.47337×10^{8}	2.60	0.033
Error	66	1.64269×10^{10}	2.48892×10^{8}		
Total	71	1.96636×10^{10}			
S = 15776 R-Sq = 16.46% R-Sq(adj) = 10.13%					

[1] VNS-II_No. N. S. refers to VNS-II_Number of Neighboring Solutions.

Table 3. Parameter setting of the variable neighborhood search methods.

Parameter	**VNS-I**	**VNS-II**
Number of Shakings	50	1
Number of Neighboring Solutions	100	50
Number of Neighborhoods (K_{max})	3	3
Neighborhood Structure (N_1)	5 teams	1 team
Neighborhood structure (N_2)	10 teams	5 teams
Neighborhood structure (N_3)	15 teams	10 teams
Stopping criterion	Max. No. of Iterations = 60,000	
Number of runs	10	

5.3. Result Comparison

This research compared the best solution and the average solution over 10 runs among the experimental result. To verify the proposed VNS algorithms' quality, the 2016 and 2019 MLB seasons' actual schedules, respectively, have been used. Table 4 shows the detailed mileages for each team obtained by the actual 2016 LB schedule, the average and the best total travelling distances by VNS-I and VNS-II, respectively. The percentage of gap between the VNS variations and the actual MLB schedule is also calculated. From Table 4, VNS-II is superior to the actual MLB schedule and the VNS-I in both average and the best performance.

Table 4. Results comparison among VNS-I, VNS-II, and actual 2016 MLB seasonal schedules (all distances are in miles).

Team	MLB 2016 Schedule	VNS-I				VNS-II			
		Avg.	Best	Avg. Gap (%)	Best Gap (%)	Avg.	Best	Avg. Gap (%)	Best Gap (%)
Baltimore Orioles	37,803	47,301	38,408	25.12	1.60	40,608	37,367	7.42	−1.15
Boston Red Sox	44,846	52,571	52,456	17.23	16.97	43,987	41,584	−1.92	−7.27
New York Yankees	41,135	44,986	44,576	9.36	8.37	41,903	40,972	1.87	−0.40
Tampa Bay Rays	44,004	55,200	58,443	25.44	32.81	43,683	42,013	−0.73	−4.52
Toronto Blue Jays	51,993	58,109	53,295	11.76	2.50	48,473	51,584	−6.77	−0.79
Chicago White Sox	31,236	35,675	37,654	14.21	20.55	31,396	30,705	0.51	−1.70
Cleveland Indians	30,445	33,718	29,758	10.75	−2.26	30,128	29,120	−1.04	−4.35
Detroit Tigers	30,907	35,293	33,611	14.19	8.75	29,930	30,956	−3.16	0.16
Minnesota Twins	34,919	38,143	37,780	9.23	8.19	34,212	34,037	−2.02	−2.53
Kansas City Royals	35,294	37,640	37,123	6.65	5.18	34,265	34,784	−2.92	−1.45
Houston Astros	46,728	54,729	59,696	17.12	27.75	44,898	41,686	−3.92	−10.79
Los Angeles Angels	53,971	61,335	57,801	13.64	7.10	52,154	51,311	−3.37	−4.93
Oakland Athletics	49,342	63,922	54,734	29.55	10.93	51,494	48,913	4.36	−0.87
Seattle Mariners	58,909	68,710	66,227	16.64	12.42	59,642	57,200	1.24	−2.90
Texas Rangers	49,390	50,948	49,098	3.15	−0.59	44,649	45,074	−9.60	−8.74
Atlanta Braves	33,826	41,702	42,894	23.28	26.81	34,748	33,948	2.73	0.36
New York Mets	41,003	49,457	51,546	20.62	25.71	38,554	37,154	−5.97	−9.39
Miami Marlins	31,211	41,560	44,343	33.16	42.07	32,784	32,503	5.04	4.14
Philadelphia Phillies	33,398	40,636	41,324	21.67	23.73	30,298	30,197	−9.28	−9.58
Washington Nationals	28,557	36,587	35,516	28.12	24.37	29,267	28,390	2.49	−0.58
Chicago Cubs	28,184	36,211	37,321	28.48	32.42	29,397	29,056	4.30	3.09
Cincinnati Reds	28,956	35,377	33,955	22.18	17.26	28,529	28,704	−1.47	−0.87
Milwaukee Brewers	30,350	38,242	38,823	26.00	27.92	31,421	30,140	3.53	−0.69
Pittsburgh Pirates	30,425	39,632	41,773	30.26	37.30	30,493	30,781	0.22	1.17
St. Louis Cardinals	30,753	38,264	39,716	24.42	29.15	31,892	31,003	3.70	0.81
Arizona Diamondbacks	42,220	51,811	51,221	22.72	21.32	41,904	41,829	−0.75	−0.93
Colorado Rockies	39,051	51,181	39,402	31.06	0.90	39,607	39,697	1.42	1.65
Los Angeles Dodgers	48,245	53,286	50,780	10.45	5.25	47,615	46,589	−1.31	−3.43
San Diego Padres	46,125	55,377	51,893	20.06	12.51	46,062	46,401	−0.14	0.60
San Francisco Giants	45,795	58,373	56,229	27.47	22.78	45,299	46,098	−1.08	0.66
Total Distance	1,179,021	1,405,976	1,367,396	19.25	13.78	1,169,292	1,149,796	−0.83	−2.48
Average distance	39,301	46,866	45,580	19.25	13.78	38,976	38,327	−0.83	−2.48
Standard Deviation	8592	9700	9122	12.89	5.81	8197	7986	−4.60	−7.06
Max Distance	58,909	68,710	66,227			59,642	57,200		
Min Distance	28,184	33,718	29,758			28,529	28,390		

With less flexibility in code setting, the VNS-I meets hard constraints that the league must abide by in the solution process. However, its ability to find solutions has been greatly restricted. Moreover, the quality of the solutions is lowered compared to the VNS-II. On the other hand, VNS-II provides much flexibility in finding solutions. The average performance of VNS-II can improve the total travelling distance of the actual 2016 MLB schedule by 0.83%. Its best solution reduces the total travelling distance of the league by 2.48%, and the distances in 21 out of 30 teams can be improved by VNS-II while comparing with the 2016 actual schedule. In addition, VNS-II has a 7.06% reduction in comparing the standard deviation of each team's travelling distance. The VNS-II shortens uneven travelling distances and improves the team's fairness.

Similarly, Table 5 summarizes the comparison of VNS-I, VNS-II and the 2019 actual MLB schedules. Once again, the best performance of VNS-II can improve the travelling distance in 22 out of 30 teams, and the average gap percentage from the actual schedule is 6.02%. The standard deviation of the travelling distance over teams (7820 miles) in the actual schedule is slightly lower than the one (7986 miles) obtained by VNS-II. Nevertheless, the 2019 actual MLB schedule increases the competition balance, or team fairness, over teams by raising the travelling distance in most teams which may not be a sustainable strategy for running a sports league.

Table 5. Results comparison among VNS-I, VNS-II and actual 2019 MLB seasonal schedules (all distances are in miles).

Team	MLB 2019 Schedule	VNS-I				VNS-II			
		Avg.	Best	Avg. Gap (%)	Best Gap (%)	Avg.	Best	Avg. Gap (%)	Best Gap (%)
Baltimore Orioles	34,070	47,301	38,408	38.83	12.73	40,608	37,367	19.19	9.68
Boston Red Sox	39,657	52,571	52,456	32.56	32.27	43,987	41,584	10.92	4.86
New York Yankees	35,317	44,986	44,576	27.38	26.22	41,903	40,972	18.65	16.01
Tampa Bay Rays	50,374	55,200	58,443	9.58	16.02	43,683	42,013	−13.28	−16.60
Toronto Blue Jays	56,357	58,109	53,295	3.11	−5.43	48,473	51,584	−13.99	−8.47
Chicago White Sox	33,102	35,675	37,654	7.77	13.75	31,396	30,705	−5.15	−7.24
Cleveland Indians	35,216	33,718	29,758	−4.25	−15.50	30,128	29,120	−14.45	−17.31
Detroit Tigers	29,360	35,293	33,611	20.21	14.48	29,930	30,956	1.94	5.44
Minnesota Twins	37,802	38,143	37,780	0.90	−0.06	34,212	34,037	−9.50	−9.96
Kansas City Royals	36,627	37,640	37,123	2.77	1.35	34,265	34,784	−6.45	−5.03
Houston Astros	46,755	54,729	59,696	17.05	27.68	44,898	41,686	−3.97	−10.84
Los Angeles Angels	50,407	61,335	57,801	21.68	14.67	52,154	51,311	3.47	1.79
Oakland Athletics	52,523	63,922	54,734	21.70	4.21	51,494	48,913	−1.96	−6.87
Seattle Mariners	55,161	68,710	66,227	24.56	20.06	59,642	57,200	8.12	3.70
Texas Rangers	46,093	50,948	49,098	10.53	6.52	44,649	45,074	−3.13	−2.21
Atlanta Braves	38,157	41,702	42,894	9.29	12.41	34,748	33,948	−8.93	−11.03
New York Mets	46,374	49,457	51,546	6.65	11.15	38,554	37,154	−16.86	−19.88
Miami Marlins	39,029	41,560	44,343	6.48	13.62	32,784	32,503	−16.00	−16.72
Philadelphia Phillies	33,665	40,636	41,324	20.71	22.75	30,298	30,197	−10.00	−10.30
Washington Nationals	36,546	36,587	35,516	0.11	−2.82	29,267	28,390	−19.92	−22.32
Chicago Cubs	32,065	36,211	37,321	12.93	16.39	29,397	29,056	−8.32	−9.38
Cincinnati Reds	33,516	35,377	33,955	5.55	1.31	28,529	28,704	−14.88	−14.36
Milwaukee Brewers	34,060	38,242	38,823	12.28	13.98	31,421	30,140	−7.75	−11.51
Pittsburgh Pirates	36,323	39,632	41,773	9.11	15.00	30,493	30,781	−16.05	−15.26
St. Louis Cardinals	31,735	38,264	39,716	20.57	25.15	31,892	31,003	0.49	−2.31
Arizona Diamondbacks	43,262	51,811	51,221	19.76	18.40	41,904	41,829	−3.14	−3.31
Colorado Rockies	35,568	51,181	39,402	43.90	10.78	39,607	39,697	11.36	11.61
Los Angeles Dodgers	43,275	53,286	50,780	23.13	17.34	47,615	46,589	10.03	7.66
San Diego Padres	50,478	55,377	51,893	9.71	2.80	46,062	46,401	−8.75	−8.08
San Francisco Giants	50,592	58,373	56,229	15.38	11.14	45,299	46,098	−10.46	−8.88
Total Distance	1,223,466	1,405,976	1,367,396	14.92	11.76	1,169,292	1,149,796	−4.43	−6.02
Average distance	40,782	46,866	45,580	14.92	11.76	38,976	38,327	−4.43	−6.02
Standard Deviation	7820	9700	9122	26.15	18.63	8197	7986	6.60	3.85
Max Distance	56,357	68,710	66,227			59,642	57,200		
Min Distance	29,360	33,718	29,758			28,529	28,390		

Lastly, the convergence performance of both VNS versions can be seen in Table 6. Both VNS-I and VNS-II start the procedure with the initial objective value of over 1,500,000 miles. VNS-II can reduce to the objective value lower than 1,200,000 within 10,000 iterations when VNS-I struggles with a slower convergence with the proper design. At the end of the search procedure, the VNS-II improves the initial objective value by 24.62% with a computational time of 720 s. Considering the MLB scheduling problem as a long-term planning problem, VNS-II provides very competitive performance within a considerably short computational time.

Table 6. Convergence analysis of VNS-I and VNS-II.

Number of Iterations	VNS-I	VNS-II
1	1,519,607	1,525,334
10,000	1,389,212	1,163,451
20,000	1,386,232	1,150,402
30,000	1,385,316	1,150,402
40,000	1,379,673	1,149,796
50,000	1,373,940	1,149,796
60,000	1,367,396	1,149,796
CPU time (seconds)	646	720
Initial solution distance (miles)	1,519,607	1,525,334
Distance after optimization (miles)	1,367,396	1,149,796
Improvement from the initial solution (%)	10.02%	24.62%

6. Conclusions

Professional sports leagues like the US Major Leagues are designed with win-maximizing behaviour. Differing optimality levels of schedules can result in different monetary values for various organizations associated with the leagues. Thus, sports scheduling plays a critical role, and such a task becomes quite challenging for league organizers. They must distinguish between short-, medium- and long-term regulatory policies. To simplify the complexity of the problem, most of the research on MLB event scheduling only discussed the scheduling of the series. Few documents discussed the scheduling of daily events. However, this study has successfully considered them and constructed the scheduling model with hard and soft constraints. Inevitably some constraints are very hard to quantify and require vast information to determine if a potential schedule works as the league intends.

For this reason, we hope league management teams can provide more valuable information in the future. In the meantime, being fans of MLB and to the best of our knowledge, we consider our VNS to be a potential tool to determine a schedule works as the league intends. Our objective in this study is to minimize the team's total travel distance, which shapes the balance of competition or fairness. The Variable Neighborhood Search algorithm was proposed to deal with the complexity of the Major League Baseball schedule and constraints. The quality of schedules was achieved by applying different coding schemes. The actual scheduling of the 2016 and 2019 MLB are two case examples to evaluate algorithmic performance. VNS-II with higher coding flexibility, in turn, provides much better results than VNS-I. VNS-II outperforms the actual 2016 MLB schedule in both total travelling distance and the standard deviation of travelling distance over 30 teams 2.48% and 7.06%, respectively. Comparing the actual 2019 MLB schedule, VNS-II provides better results in the total travelling distance. In short, the proposed VNS algorithm can provide cost-saving schedules that also ensure the athletic quality and balance of competitions within leagues. The VNS algorithm shows its merit to create such schedules and its potential for other sports scheduling problems.

Essentially all the sports scheduling researches that we have reviewed discussed actual schedules obtained for some specific leagues and affirmed that the sports schedule's optimality varies as the set of constraints associated with the league's changes. The minor details of the constraints have a significant impact on the optimal schedule's final result. Hence, our future research directions focus on finding a way to determine which constraints can be slightly modified for algorithmic purposes that potentially lead to optimum or near-optimal schedules much more effectively and efficiently.

Author Contributions: Conceptualization, Y.-C.L. and Y.-Y.L.; methodology, Y.-C.L. and Y.-Y.L.; software, Y.-Y.L.; validation, Y.-Y.L. and Y.-C.L.; formal analysis, Y.-Y.L. and Y.-C.L.; investigation, Y.-Y.L. and Y.-C.L.; resources, Y.-Y.L. and Y.-C.L.; data curation, Y.-Y.L. and W.-S.C.; writing—original draft preparation, A.H.-L.C., W.-S.C., and Y.-C.L.; writing—review and editing, A.H.-L.C. and Y.-C.L.; visualization, A.H.-L.C. and Y.-C.L.; supervision, Y.-C.L.; project administration, Y.-C.L. and A.H.-L.C.; funding acquisition, A.H.-L.C. and Y.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research receives no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bartsch, T.; Drexl, A.; Kröger, S. Scheduling the professional soccer leagues of Austria and Germany. *Comput. Oper. Res.* **2006**, *33*, 1907–1937. [CrossRef]
2. Rasmussen, R.V.; Trick, M.A. A Benders approach for the constrained minimum break problem. *Eur. J. Oper. Res.* **2007**, *177*, 198–213. [CrossRef]
3. Nurmi, K.; Goossens, D.; Bartsch, T.; Bonomo, F.; Briskorn, D.; Duran, G.; Kyngäs, J.; Marenco, J.; Ribeiro, C.C.; Spieksma, F.; et al. A Framework for a Highly Constrained Sports Scheduling Problem. In Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2010), Hong Kong, 17–19 March 2010; International Association of Engineers: London, UK, 2010; Volume III, pp. 1991–1997.
4. Kendall, G.; Knust, S.; Ribeiro, C.C.; Urrutia, S. Scheduling in sports: An annotated bibliography. *Comput. Oper. Res.* **2010**, *37*, 1–19. [CrossRef]
5. Ribeiro, C.C. Sports scheduling: Problems and applications. *Int. Trans. Oper. Res.* **2012**, *19*, 201–226. [CrossRef]
6. Goossens, D.R.; Spieksma, F.C. Soccer Schedules in Europe: An Overview. *J. Sched.* **2012**, *15*, 641–651. [CrossRef]
7. Wright, M.B. OR Analysis of Sporting Rules–A survey. *Eur. J. Oper. Res.* **2014**, *232*, 1–8. [CrossRef]
8. Easton, K.; Nemhauser, G.; Trick, M. The traveling tournament problem description and benchmarks. In Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming, Paphos, Cyprus, 26 November–1 December 2001; pp. 580–584.
9. Anagnostopoulos, A.; Michel, L.; Van Hentenryck, P.; Vergados, Y. A simulated annealing approach to the traveling tournament problem. *J. Sched.* **2006**, *9*, 177–193. [CrossRef]
10. Chen, P.-C.; Kendall, G.; Berghe, G.V. An Ant Based Hyper-heuristic for the Travelling Tournament Problem. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), Honolulu, Hawaii, USA, 1–5 April 2007; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2007; pp. 19–26.
11. Di Gaspero, L.; Schaerf, A. A composite-neighborhood tabu search approach to the traveling tournament problem. *J. Heuristics* **2007**, *13*, 189–207. [CrossRef]
12. Trick, M.A. Sports Scheduling. *Hybrid Optim.* **2011**, *45*, 489–508.
13. Cáceres, L.P.; Riff, M.C. AISTTP: An Artificial Immune Algorithm to Solve Traveling Tournament Problems. *Int. J. Comput. Intell. Appl.* **2012**, *11*, 1250008. [CrossRef]
14. Januario, T.; Urrutia, S. A new neighborhood structure for round robin scheduling problems. *Comput. Oper. Res.* **2016**, *70*, 127–139. [CrossRef]
15. Khelifa, M.; Boughaci, D. A Variable Neighborhood Search Method for Solving the Traveling Tournaments Problem. *Electron. Notes Discret. Math.* **2015**, *47*, 157–164. [CrossRef]
16. Khelifa, M.; Boughaci, D. Hybrid harmony search combined with variable neighborhood search for the traveling tournament problem. In Proceedings of the International Conference on Computational Collective Intelligence, Halkidiki, Greece, 28–30 September 2016; Springer: Cham, Switzerland, 2016; pp. 520–530.
17. Januario, T.; Urrutia, S.; Ribeiro, C.C.; De Werra, D. Edge coloring: A natural model for sports scheduling. *Eur. J. Oper. Res.* **2016**, *254*, 1–8. [CrossRef]
18. Alarcón, F.; Durán, G.; Guajardo, M.; Miranda, J.; Muñoz, H.; Ramírez, L.; Ramírez, M.; Sauré, D.; Siebert, M.; Souyris, S.; et al. Operations Research Transforms the Scheduling of Chilean Soccer Leagues and South American World Cup Qualifiers. *Interfaces* **2017**, *47*, 52–69. [CrossRef]
19. Khelifa, M.; Boughaci, D.; Aimeur, E. An enhanced genetic algorithm with a new crossover operator for the traveling tournament problem. In Proceedings of the 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT), Barcelona, Spain, 5–7 April 2017; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2017; pp. 1072–1077.

20. Durán, G.; Guajardo, M.; Sauré, D. Scheduling the South American Qualifiers to the 2018 FIFA World Cup by integer programming. *Eur. J. Oper. Res.* **2017**, *262*, 1109–1115. [CrossRef]
21. Durán, G. Sports scheduling and other topics in sports analytics: A survey with special reference to Latin America. *Top* **2021**, *29*, 125–155. [CrossRef]
22. Trick, M.A.; Yildiz, H. Benders' cuts guided large neighborhood search for the traveling umpire problem. *Nav. Res. Logist.* **2011**, *58*, 771–781. [CrossRef]
23. Trick, M.A.; Yildiz, H. Locally Optimized Crossover for the Traveling Umpire Problem. *Eur. J. Oper. Res.* **2012**, *216*, 286–292. [CrossRef]
24. Trick, M.A.; Yildiz, H.; Yunes, T. Scheduling Major League Baseball Umpires and the Traveling Umpire Problem. *Interfaces* **2012**, *42*, 232–244. [CrossRef]
25. Xue, L.; Luo, Z.; Lim, A. Two exact algorithms for the traveling umpire problem. *Eur. J. Oper. Res.* **2015**, *243*, 932–943. [CrossRef]
26. Toffolo, T.A.M.; Christiaens, J.; Spieksma, F.C.R.; Berghe, G.V. The sport teams grouping problem. *Ann. Oper. Res.* **2017**, *275*, 223–243. [CrossRef]
27. Westphal, S. Scheduling the German Basketball League. *Interfaces* **2014**, *44*, 498–508. [CrossRef]
28. Goossens, D. Optimization in Sports League Scheduling: Experiences from the Belgian Pro League Soccer. In Proceedings of the 7th International Conference on Operations Research and Enterprise Systems, Funchal, Madeira, Portugal, 24–26 January 2018; Springer: Cham, Switzerland, 2018; pp. 3–19.
29. Günneç, D.; Demir, E. Fair-fixture: Minimizing carry-over effects in football leagues. *J. Ind. Manag. Optim.* **2019**, *15*, 1565–1577. [CrossRef]
30. Çavdaroğlu, B.; Atan, T. Determining matchdays in sports league schedules to minimize rest differences. *Oper. Res. Lett.* **2020**, *48*, 209–216. [CrossRef]
31. Gotzes, U.; Hoppmann, K. Bounding the final rank during a round robin tournament with integer programming. *Oper. Res.* **2020**, 1–9. [CrossRef]
32. Lambrechts, E.; Ficker, A.M.C.; Goossens, D.R.; Spieksma, F.C.R. Round-robin tournaments generated by the Circle Method have maximum carry-over. *Math. Program.* **2018**, *172*, 277–302. [CrossRef]
33. Melouk, S.H.; Keskin, B.B. Team assignments and scheduling for the NCAA basketball tournament. *J. Oper. Res. Soc.* **2012**, *63*, 620–630. [CrossRef]
34. Nemhauser, G.L.; Trick, M.A. Scheduling A Major College Basketball Conference. *Oper. Res.* **1998**, *46*, 1–8. [CrossRef]
35. Henz, M. Scheduling a Major College Basketball Conference—Revisited. *Oper. Res.* **2001**, *49*, 163–168. [CrossRef]
36. Faella, M.; Sauro, L. Irrelevant matches in round-robin tournaments. *Auton. Agents Multi-Agent Syst.* **2021**, *35*, 1–34. [CrossRef]
37. Robinson, L.W. Baseball playoff eliminations: An application of linear programming. *Oper. Res. Lett.* **1991**, *10*, 67–74. [CrossRef]
38. Gusfield; Martel The Structure and Complexity of Sports Elimination Numbers. *Algorithmica* **2002**, *32*, 73–86. [CrossRef]
39. Rutjanisarakul, T.; Jiarasuksakun, T. A Sport Tournament Scheduling by Genetic Algorithm with Swapping Method. *arXiv* **2017**, arXiv:1704.04879.
40. Raof, A.R.A.; Sudin, S.; Mahrom, N.; Rosli, A.N.C. Sport Tournament Automated Scheduling System. *MATEC Web Conf. EDP Sci.* **2018**, *150*, e05027. [CrossRef]
41. Della Croce, F.; Dragotto, G.; Scatamacchia, R. On fairness and diversification in WTA and ATP tennis tournaments generation. *Ann. Oper. Res.* **2020**, 1–13. [CrossRef]
42. Burrows, W.; Tuffley, C. Maximising Common Fixtures in a Round Robin Tournament with Two Divisions. *arXiv* **2015**, arXiv:1502.06664.
43. Davari, M.; Goossens, D.; Beliën, J.; Lambers, R.; Spieksma, F.C. The multi-league sports scheduling problem, or how to schedule thousands of matches. *Oper. Res. Lett.* **2020**, *48*, 180–187. [CrossRef]
44. Lim, A.; Rodrigues, B.; Zhang, X. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *Eur. J. Oper. Res.* **2006**, *174*, 1459–1478. [CrossRef]
45. Wright, M. Scheduling fixtures for Basketball New Zealand. *Comput. Oper. Res.* **2006**, *33*, 1875–1893. [CrossRef]
46. Van Hentenryck, P.; Vergados, Y. Population-Based Simulated Annealing for Traveling Tournaments. In Proceedings of the 22nd National Conference on Artificial Intelligence, Vancouver, Canada, 22–26 July 2007; AAAI Press: Palo Alto, CA, USA, 2007; pp. 267–272.
47. Khelifa, M.; Boughaci, D. A Cooperative Local Search Method for Solving the Traveling Tournament Problem. *Comput. Inform.* **2019**, *37*, 1386–1410. [CrossRef]
48. Van Bulck, D.; Goossens, D.R.; Spieksma, F.C.R. Scheduling a non-professional indoor football league: A tabu search based approach. *Ann. Oper. Res.* **2018**, *275*, 715–730. [CrossRef]
49. Uthus, D.C.; Riddle, P.J.; Guesgen, H.W. An ant colony optimization approach to the traveling tournament problem. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation—GECCO '09, Montreal, Canada, 8–12 July 2009; ACM: New York, NY, USA, 2009; pp. 81–88.
50. Guiqian, T. The Optimization of Professional Sporting Management System Based on Gene Algorithm. In *Advanced Technology in Teaching, Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009), Shenzhen, China, 19–20 December 2009*; Advances in Intelligent and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2012; Volume 116, pp. 679–687.

51. Biajoli, F.L.; Lorena, L.A.N. Clustering search approach for the traveling tournament problem. In *Advances in Artificial Intelligence. Lecture Notes in Computer Science*; Gelbukh, A., Kuri Morales, Á.F., Eds.; Springer: Berlin, Heidelberg, Germany, 2007; Volume 4827, pp. 83–93.
52. Russell, R.A.; Leung, J.M.Y. Devising a Cost Effective Schedule for a Baseball League. *Oper. Res.* **1994**, *42*, 614–625. [CrossRef]
53. Hoshino, R.; Kawarabayashi, K.I. A Multi-round Generalization of the Traveling Tournament Problem and its Application to Japanese Baseball. *Eur. J. Oper. Res.* **2011**, *215*, 481–497. [CrossRef]
54. Hoshino, R.; Kawarabayashi, K.I. Generating Approximate Solutions to the TTP using a Linear Distance Relaxa-tion. *J. Artif. Intell. Res.* **2012**, *45*, 257–286. [CrossRef]
55. Yeh, C.W. Tabu search algorithm for major league baseball scheduling. Master's Thesis, Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan City, Taiwan, 2013.
56. Lin, W.-B.; Chen, M.-Y. A study of efficiency monitoring systems for match-fixing players in the Chinese Professional Baseball League. *Eur. Sport Manag. Q.* **2015**, *15*, 301–322. [CrossRef]
57. Ko, Y.D.; Jung, S.H.; Kim, S.H.; Lee, S.W. Sustainable Sport Scheduling Approach Considering Team Equity for the Korean Professional Baseball League. *Sustainability* **2018**, *10*, 429. [CrossRef]
58. Kim, T. Optimal approach to game scheduling of multiple round-robin tournament: Korea professional baseball league in focus. *Comput. Ind. Eng.* **2019**, *136*, 95–105. [CrossRef]
59. Hansen, P.; Mladenović, N. *An Introduction to Variable Neighbourhood Search. Metaheuristic*; Springer: Boston, MA, USA, 1999; pp. 433–458.
60. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]