2000

Athletic game scheduling

Hyoung-Ro Lee
Iowa State University

Athletic game scheduling

by

Hyoung-Ro Lee

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Major Professor: Douglas D. Gemmill

Iowa State University

Ames, Iowa

2000

UMI Number: 9977336

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Hyoung-Ro Lee

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Major Professor

e was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

An athletic game scheduling (AGS) problem can be formulated using 0-1 integer program with minor constraints and can be solved partially using traditional operations research methods. However, the constraints that each team and the league requested of the AGS scheduler make the problem no longer practical by typical operations research methods. In most real world problems, many constraints cannot be formulated into mathematical forms. We, therefore, developed heuristic algorithms that can generate reasonably good solutions in a short period. In this research, we propose two algorithms, multi echelon simulated annealing and multi echelon tabu search. In the beginning of the research, our efforts concentrated on developing and defining general athletic game scheduling algorithms that can solve schedules for every athletic league with minor changes. We developed a schedule for year 2000 for the Southern League Baseball, an AA minor league. The procedures of this development process are provided. Finally, an ***A/B/C/D/E*** format is defined to determine the different types of AGS problems depending on these characteristics: the number of meetings between two teams, the number of divisions in the league, the availability of fixed time slots, and the number of teams in each division.

GENERAL INTRODUCTION

Introduction

In recent years, researchers have looked into different areas other than the traditional operations research areas: production scheduling in manufacturing process, vehicle routing problems, materials resource planning, etc. Finding a solution to the problem of athletic game scheduling (AGS) is one of the areas that operations researchers have been investigating recently. They formulated the AGS problem using 0-1 integer programming with minor constraints. However, instead of using traditional operational research methods, in most cases they implemented heuristics to trace feasible solutions because the AGS problem is too complicated to solve and many of the constraints cannot be formulated.

Athletic events are held everywhere all year long. They can be scheduled as individual events, team events, regional events, or as national competition. Especially in America, professional and college athletic events attract multi media interest and generate huge amounts of revenue. Today, these teams are not only playing and competing but also they are making a profit. Therefore, cutting costs and increasing revenue have become major issues to an athletic league. One way of reducing cost and increasing revenue is to determine a good AGS -- the sequence in which games are scheduled among the teams.

Determining a good AGS is a complicated problem that can include many constraints. Hence, finding the global optimal solution is not likely. The prevalent objective of previous researchers has been to determine a local optimal solution. No researchers have fully applied operations research methods. However, some of them found the best solution by using enumeration and integer programming.

Literature Review

Many studies have been published regarding the AGS problem. Table 1 shows the type of sport and games format between two teams that previous researchers investigated. As we see in Table 1, due to the complexity of AGS problems most research has focused on finding heuristics to specific problems. Otherwise, simple enumeration of all the possible solutions was used to find the best solution. Also, most problems included a fixed number of round robin games, where round robin games means two teams meet again after meeting all remaining teams once. In Chapter V, we will explain and classify previous research in terms of the number of round robin games in the league. Most of the previous research dealt with more than double round robin games excepts two studies, for example, Wright (1994) and Armstrong and Willis (1993).

Table 1. Previous research on AGS

Sport	Researcher(s)	Year	Search algorithm	Game format
Basketball	Nemhauser and Trick	1998	Enumeration and optimization	Double round robin
	Bean and Birge	1980	Heuristics	No information
	Ball and Webster	1977	Heuristics	Double round robin
	Campbell and Chen	1976	Heuristics	Double round robin
Baseball	Russell and Leung	1994	Heuristics	Multi round robin
	Cain	1977	Heuristics	Multi round robin
Ice Hockey	Ferland and Fleurent	1991	Heuristics	No information
Soccer	Schreuder	1992	Heuristics	Double round robin
Cricket	Willis and Terrill	1994	Heuristics	No information
	Wright	1994	Heuristics	Single round robin
	Armstrong and Willis	1993	Lotus 1-2-3 (hand) : heuristics	Single round robin

The constraints and solution method of previous research are as follows. Nemhauser and Trick (1998) implemented a procedure to schedule college basketball in the Atlantic Coast Conference (ACC). The ACC consists of nine universities, Clemson (Clem), Duke, Florida State (FSU), Georgia Tech (GT), Maryland (UMD), North Carolina (UNC), North Carolina State (NCSt), Virginia (UVA), and Wake Forest (Wake), in the Southeastern United States. The biggest share of the revenue that the ACC generates comes from television and radio networks who show the games, and from the tickets that fans purchase. The revenue is also greatly affected by the scheduling of the teams. The ACC was interested in developing a good schedule to maximize the revenue. The schedule would be determined by the constraints shown below:

1. The games are a double round robin.
2. Every team plays twice in a week. A team, therefore, has 8 home games, 8 away games and 2 bye games. To be fair every team requires 4 home weekends, 4 away weekends, and 1 bye weekend.
3. No teams are allowed to play two consecutive home or away games.
4. Long series of away weekend games are not allowed.
5. A team must have at least two home weekends or one home and one bye weekend among the first five slots for recruiting student athletes.
6. No team has two consecutive away games in the final week. The final week is reserved for rival games.

7. Duke vs UNC games are reserved at slot 10 and 17 for television network. UNC vs Clem plays at slot 1. The following pairs must be scheduled at least once in February: Wake vs UNC, Wake vs Duke, GT vs UNC, and GT vs Duke.
8. No team should play UNC, Duke, and Wake consecutively and no team should play UNC and Duke consecutively due to the strength of the three teams.
9. Duke has a bye in slot 15. Wake should not have a home game in slot 16 and should have a bye in slot 0.
10. Clem, Duke, UMD, and Wake should not end the season with an away game.
11. Clem, FSU, GT, and Wake should not begin with an away game. UNC should not begin with bye.
12. Neither FSU nor NCSt should end with bye.

Instead of using a combinatorial design, Nemhauser and Trick combined integer programming and enumeration methods to determine existing scheduling patterns. Three steps were used to find a feasible solution. Patterns of strings representing home and away games (HAP) were generated by enumeration combined with integer programming in step 1. In step 2 games were assigned to the HAP using integer programming to make a timetable. And teams were assigned to the timetable using enumeration in step 3. Their objective was simply to find a feasible solution that met all constraints. The final solution was accepted by the ACC.

Ball and Webster (1977) developed scheduling heuristics for the Big Eight and the Southeastern conference college basketball games. They investigated two methods, a 0-1 integer program and an iterative heuristic approach. The constraints that they considered were

1. An even number of teams.
2. Two consecutive road games were allowed without returning home if the trip did not span an entire week.
3. Each team had two games with all the other teams, once at home and once away.
4. A team never played four games in a row either at home or away.
5. The schedule needed to be double round-robin so that every team had to play every other team once before any two teams play twice.
6. Each team had two byes during the entire season.

The final solution revealed a significant reduction in total travel miles.

Campbell and Chen (1976) studied the Southeastern Conference (SEC) basketball scheduling. The conference consisted of 10 universities in 1973-74. The constraints that the authors considered were:

1. The games were double round robin. Every team would have two games with all the other teams, once at home and once away.
2. A maximum of two consecutive away games would be allowed, and they should be scheduled on Saturday and Monday.
3. There would be a minimum of 4 Saturday home games for each team.
4. Having away games on two consecutive weekends was not allowed.

They used a two-phase heuristic model. In the first phase minimum travel distance for all teams was generated by using a combinatorial method. This method uses nodes to represent the home of a team and arcs to represent travel between two nodes. When generating minimum travel distance, only the minimum constraint--two consecutive away games were allowed--was considered. Phase two added the remaining constraints and enumerated the schedules as needed in order to maintain feasibility. The final solution showed that the reduction on total travel distance was 29.3 % compared to the schedule that was used in the previous season.

Bean and Birge (1980) investigated National Basketball Association (NBA) scheduling and attempted to minimize the total travel distance. The NBA consistes of 22 teams. Each team has 82 games, 41 at home and 41 away. They modeled the problem in two ways, a combinatorial and a 0-1 integer program. The combinatorial description interprets the AGS schedule as a collection of nodes representing games, and of arcs representing the teams travel from one game to the next. Colors distinguish the arcs as different travel days. However, the combinatorial approach would be extremely difficult due to the size of the graph, 902 nodes and 73,062 arcs. The 0-1 integer program model is also impractical because of the great size (41,976 constraints and 873,136 variables). In addition, Bean and Birge applied a version of the multi-traveling salesman problem to each team, where each salesman (team) can travel to a maximum of 5 cities (road games) in one trip. But, because of the tightness of the constraints for the NBA problem, no feasible solution was found. Consequently, a heuristic model that divides the shortest tour that cannot be placed into partial tours was developed to find feasible solution. The final solution reduced travel costs about 20% or $757,000 compared to the previous year's schedule.

Ferland and Fleurent (1991) proposed a computer-based expert system to schedule the National Hockey League (NHL) and other sports. NHL had 21 teams in two conferences, each of which is divided into two divisions. Each team had 80 games during the season, 40 at home and 40 away. The objective was to minimize total travel distances and to maximize the number of weekend games. The constraints for the NHL were:

1. Availability of arenas.
2. Two teams in the same division could not meet twice within 14 days. Also, two teams in different divisions required 30 days of time to elapse before meeting again.
3. A team could not travel more than 900 miles between games played on consecutive days.
4. A team could not play more than one game a day.
5. No more than two games on three consecutive days were allowed.
6. No more than three games on five consecutive days were allowed.
7. A team had at least two games in a week.

Even though the problem can be formulated as 0-1 integer program, it was impossible to derive an optimal solution because of the size of the problem--750,000 constraints and 150,000 variables, and hidden constraints that cannot be formulated. Therefore, Ferland and Fleurent developed an expert system that allows an expert user to collect and update the data required to specify the schedule. The expert manually inputs an initial schedule that enables the expert system to exploit its knowledge to further improve the schedule.

Armstrong and Willis (1993), Willis and Terrill (1994) and Wright (1994) studied the scheduling of cricket matches. Armstrong and Willis created a schedule for the World Cup cricket matches held throughout Australia and New Zealand in 1992. The schedule initially included 9 teams representing 9 countries where each team played every other team only once within 26 days. For scheduling they took into consideration the travel distance of each team and TV broadcasting. The main constraints were:

1. A team had at least two days of byes--one for travel one for rest.
2. No team had a game on the same day when Australia was playing New Zealand in Auckland. There would be eight teams competing on the last two days to increase the chance of a major event that decides the finalists.
3. Games were not allowed on Monday in New Zealand. Also, Monday and Friday should be avoided in Australia.
4. Televised Australian games would be played on Saturday, Sunday, and Wednesday. Additionally, only the best 11 games would be televised.
5. Day/night games on Wednesdays and Thursdays, as well as the weekend games, would be televised in Australia.
6. Games that were not televised would have to schedule on a week day, during idle weekend days in New Zealand, or during the same time that televised game was held.

Armstrong and Willis (1994) developed the problem with a 0-1 integer program but used a heuristic to search for feasible solutions because the size of the problem and the requirements that could not be formulated as a mathematical model. They utilized Lotus 1-2-

3 as a tool and developed two procedures, one that generated a schedule automatically and the other that allowed interaction with the user. As a result, they found 24 acceptable solutions.

In other research Willis and Terrill applied the simulated annealing algorithm to search for feasible schedules for the Australian cricket league. The league consisted of 6 teams in six states (NSW, Queensland, South Australia, Tasmania, Victoria and Western Australia). The games had to be scheduled around international games in such a way that the final solutions had to satisfy both domestic and international constraints. The domestic game had one, two, and four game series. The main constraints that they considered were:

1. One-day games are played either on Saturday or Sunday.
2. Each team had a 2/3 or a 3/2 home and away game ratio for one-day games.
3. A one-day game should be avoided during the Motor Racing Formula 1 Grand Prix in Adelaide.
4. A team could not have two games on one day.
5. A maximum total of two games over the three weekends was allowed in late February and early march.
6. A maximum of two consecutive away series was allowed for the four game series.
7. One game series on specific dates, nominated by the TV broadcasting company, must be satisfied.
8. Two teams should not meet twice within 4 weeks.
9. International games scheduling on Saturday and Sunday should be avoided.

10. Scheduling games in Victoria and Tasmania in October and in Queensland in February and March should be avoided due to expected weather.

The objective was to minimize total penalties assigned when a schedule violates constraints. The final solution from simulated annealing was manually modified to produce a feasible schedule.

Wright (1994) developed a schedule for an English county cricket tournament using tabu search. Eighteen teams were in the tournament and each team played only once with every other team. He handed out questionnaires to all teams asking for a preferred day of home and away game, the rival team, and the team that did not want to have a home game on a specific day. Penalty costs occurred when a team had long-distance overnight travel or when a team had to travel to the same county twice within a two-week period. The final solution would need to be modified manually if complaints were received from the counties or tournament committee.

A Dutch Professional Football schedule, consisting of 18 teams, was developed by Schreuder (1992). When developing the game schedule, he considered requirements from different parties including municipalities, police, railway, International Football Federation (FIFA), the teams, and press. The final solution was chosen based on the balance of the requirements from all parties. In constructing a fair schedule for all the teams, Schreuder categorized the requirements into three parts: fans, team ranking, and behavior of hooligans. To generate a fair schedule he first developed HAP with edge colorings of complete graphs, where vertices represent teams and edges represent games between teams. Once acceptable or feasible HAP is found teams are assigned to the HAP, $n!$ solution space, to maximize the satisfaction of the requirements. He considered 126 requirements into the problem and 80% of them are satisfied by the final solution.

A Major League Baseball (MLB) schedule was developed by Cain (1977). In 1975 the MLB had twelve teams in each league, the National League and the American League. Each league was divided into two divisions, with six teams in each division. A team had a three-game series either over a weekend or during a week day. Also, the league allowed three series a week, two for a two-game series and one for a three-game series. The author found that the fan attendance had greater influence to a team's total revenue than the travel distance. The schedule, therefore, had to be generated in the way that attracted fans, which led him to consider three aspects when making the schedule: travel distance, attendance, and fairness. The constraints that MLB required were:

1. No two teams would have back to back games. Meeting twice between two teams in three series was not recommended.
2. A maximum of four consecutive home or away series and a minimum of two consecutive home series were allowed. However, less than two single home game series for a team in a season was also acceptable.
3. Cities that had two teams such as New York, San Francisco, Los Angeles, and Chicago could not have two home games at the same time.
4. The number of weekend games at home and away should be the same. Every team would have home weekend games with each of eleven other teams.
5. Home games in a month had to be evenly distributed to all teams.
6. Each team should have at least four home series in a month.

7. A team should not have either strong or weak opponents in a short period of time. In other words, schedule strength had to be balanced.
8. Montreal preferred to be at home on Jean-Baptiste and Dominion days. Boston also wanted a home game on Patriots' Day.
9. The Fourth of July was assigned "at home" to teams that had played an away game on that holiday in the previous year.

In addition, the league required that rival games occur at a designated time and place (i.e. Dodgers and Giants at San Francisco and Cardinals and Cubs at Chicago).

Cain divided a season schedule into three phases. During the first two phases a team had two series with every other team in the league, one at home and one away. The third phase was composed of 10 series for all teams, five at home and five away with teams in the same division. Each of the first two phases was further subdivided into 10 intra-division series and 12 inter-division series. The schedule thus consisted of five separate schedules. For each of them HAP was developed. Then, teams were assigned to the HAPs to find a schedule that satisfies the constraints. A total of 500 man hours and five hours of central processing unit (CPU) computer time were used to solve the problem. The final schedule was better in every aspect, including constraints such as travel distance and fairness, than the schedule used in the year 1969.

Russell and Leung (1994) presented two heuristics for finding a low cost schedule for the Texas Baseball League, a AA Minor League. The league had eight teams in two divisions. The constraints that determined the schedule were:

1. Each team could not play more than 14 consecutive games, either at home or away.
2. A team should have time off every 21 days.
3. Back to back series between two teams were not allowed.
4. Each team would play every other team in the first half of the season.
5. Games had to be distributed evenly throughout the season.

In addition, all teams preferred to have a home game on the Fourth of July holiday. Therefore, priority must be given to a team that had an away game in the previous season. The first heuristic that they studied was composed of two stages. The algorithm first generated HAP in stage 1, and then assigned teams to the given HAP using enumeration of teams order in stage 2 to find the minimum total travel distance. The final solution suggested that a total of 6.5 % reduction in costs and a 5.6 % reduction in total travel distance were achieved compared to the previous season.

The second heuristic they developed had three stages. The algorithm first solves a matching problem on *n* teams and then determines the order of the paired teams obtained in stage 1. The final schedule was obtained combining the open (bye) series to the order of paired teams determined in stage 2. The final solution, the minimum travel distances, was obtained from the enumeration of all teams to the final schedule.

Werra (1980, 1988) examined the combinatorial aspect of AGS, where several constraints are implied by the teams' locations in particular, and developed *Home-and-Away Pattern* (HAP). He considered $2n$ teams, where every team played only once with every other team and proved that the minimum number of break or bye is $2n$-2.

Table 2. Solution methods used in previous research

Researcher(s)	Solution method
Nemhauser and Trick	Stage 1: Integer programming for HAP. Stage 2: Assign game Stage 3: Enumeration of teams
Bean and Birge	Stage1 : Combinatorial algorithm to each team's schedule Stage 2: Change schedule if conflict exists
Ball and Webster	Minimum distance pairs with mirror schedule
Campbell and Chen	Stage 1: Find solution with minimum constraints Stage 2: Add constraint to the solution
Russell and Leung	Heuristic 1: Stage 1: Find HAP Stage 2: Assign teams to the HAP using enumeration Heuristic 2: Stage 1: Solve matching problem Stage 2: Determine the order of paired teams among matches Stage 3: Add open (bye)
Cain	Stage 1: Find HAP Stage 2: Assign teams to the HAP
Ferland and Fleurent	Expert system with ability of user interface
Schreuder	Stage 1: Find HAP Stage 2: Assign teams to the HAP
Willis and Terrill	Simulated annealing
Wright	Tabu search
Armstrong and Willis	User interactive program using Lotus 1-2-3

The types of solution methods that previous scientists used are summarized in Table 2. Table 3 shows the main constraints that were included and the objectives of the previous research. Since Cain introduced HAP method, scientists adopted HAP for developing AGS problems, for example, Nemhauser and Trick, Russell and Leung, and Schreuder. After generating HAP, they assigned or enumerated the different order of teams to find a good solution. Hence, the final solution is greatly depends on the initial HAP that they generated. Willis and Terrill (1994) and Wright (1994) used simulated annealing and tabu search for determining the acceptance/rejection of current solution. Campbell and Chen (1976) used minimum constraints to obtain initial solution and the final solution is found after adding additional constraints to the initial solution. Therefore, the final solution depends on the initial solution that they obtained.

Many previous research efforts concerned minimizing travel distance, for example, Bean and Birge (1980), Ball and Webster (1977), Campbell and Chen (1976), Russell and Leung (1994), Cain (1977), Armstrong and Willis (1993). Also, scientists studied "fair schedules", i.e. Cain (1977) and Schreuder (1992). In some research, scientists attempted to find a schedule that satisfied the most constraints: Shreuder (1992), Willis and Terrill (1994), and Wright (1994). Nemhauser and Trick (19980 and Ferland and Fleurent (1991) found feasible solutions. The requirements that we can find in the previous research

Table 3. Objectives and main constraints used to build schedule

Researcher(s)	Objective	Constraints
Nemhauser and Trick	Feasible solution	Consecutive home/away games
Bean and Birge	Min. travel distance	Building availability
Ball and Webster	Min. travel distance	Consecutive home/away games
Campbell and Chen	Min. travel distance	Consecutive home/away games
Russell and Leung	Min. travel distance	
Cain	Min. travel distance Fair schedule	
Ferland and Fleurent	Feasible solution	Arena availability Consecutive home/away games Day off for traveling over 900 miles
Schreuder	Max. constraint satisfaction Fair schedule	Fans behavior Previous season ranking Teams sharing same fans
Willis and Terrill	Max. constraint satisfaction	Consecutive home/away games
Wright	Max. constraint satisfaction	Consecutive away games
Armstrong and Willis	Consider travel distance	Televised game

include consecutive home/away game constraint, building availability, fans, day off constraints for long travel. In this research, we developed a general procedure that can be applicable to every AGS problem with the minimum constraints found in most all previous research. Hence, our algorithm is not problem specific. Also, instead of finding local solution our solution search for a solution globally. First two Chapters are devoted to developing a general multi echelon algorithm. We used simulated annealing and tabu search for acceptance/rejection mechanic for the current solution. Also, the fair game scheduling issue will be addressed in Chapter III in terms of travel distance. As a real world example we provided the Southern League Baseball schedule in Chapter IV implemented using the multi echelon algorithm. The detailed dissertation organization in as follow.

Dissertation Organization

The research on AGS was conducted in five stages. The general background and objectives of this dissertation are briefly described in the general introduction.

CHAPTER I and II, we take the constraints required to construct an AGS and show the general procedure that is applicable to most AGS problems. CHAPTER I, "Paired home and away athletic game scheduling using simulated annealing", describes the general AGS problem. The problem consists of *n* teams, an even number, and *n* sites. We formulate the problem as a 0/1 integer program and demonstrate the initial solution procedure. The integer program provides good insight and understanding of the AGS problem and its constraints. A multi echelon simulated annealing (SA) algorithm was implemented to search for optimality.

CHAPTER II, "Athletic game scheduling using tabu search", demonstrates application of a multi-echelon tabu search algorithm (Tabu) to the AGS problem defined in CHAPTER I. The algorithm attempts to minimize the total travel distance. The final results are compared to those achieved with SA in CHAPTER I.

CHAPTER III, " Fair athletic game scheduling", finds fair schedules in order to satisfy parties in a league. As we see in the literature review, fairness is an important factor when determining the acceptance of the schedule proposed. We investigate the fairness in terms of travel distance. Two methods are introduced: minimizing the variance of each team's travel distance (MVTD) and minimizing the greatest distance traveled (MDLT) by any team. A multi echelon algorithm that employed SA in echelon 3 is implemented. The final results suggest that the fair schedule may not find the minimum total travel distance but it provides a solution on which most teams can agree. MDLT outperforms MVTD in terms of the total travel distance. On the other hand, in most examples MVTD generates a more balanced schedule to all teams.

CHAPTER IV, "The Southern League baseball scheduling using simulated annealing", presents a real world example that includes two divisions containing 5 teams in each division. Three objectives--minimizing travel distance, minimizing penalty costs imposed when a schedule cannot satisfy the league requirements, and minimizing the combination of travel distance and penalty costs--are evaluated. We formulate the problem as 0-1 integer program with minor constraints. The problem is deterministic, and a multi echelon heuristic is used because not all of the constraints can be formulated into mathematical form. SA is implemented in echelon 3 to overcome entrapment in local optimal solution. The final solution provides fairness to all teams in the league.

CHAPTER V, "Determining different types of athletic game scheduling", categorizes the different types of AGS problems depending on their characteristics; the number of meetings between two teams, the number of divisions in the league, availability of fixed time slots, and the number of teams in a division. Then the representation method for AGS problems is given using ***A/B/C/D/E*** format. Also, a simple solution technique is provided for a single round robin game schedule with fixed time slots. Complex game schedules can be developed using the single round robin game schedule.

Finally, general concluding remarks are made including future work. The final solutions from minimizing travel distance, minimizing penalty costs, and minimizing the combination of travel distance and penalty costs for the Southern League in CHAPTER IV are presented in Appendix A. References cited in the introduction are shown at the end of the dissertation.

CHAPTER I. PAIRED HOME AND AWAY ATHLETIC GAME SCHEDULING USING SIMULATED ANNEALING

Introduction

Athletic events are held everywhere all year long. They are held as individual events, team events, regional events, or national competition. Especially in America, professional and college athletic events attract multi media and generate huge amounts of revenue. Today, those teams are not only competing with each other but also the teams are making profit. Therefore, cutting cost has become an important issue. One way of reducing cost and increasing revenue is to determine a good athletic game schedule (AGS), or the sequence in which games are scheduled among the teams.

Finding a good AGS is a complicated problem that can include many constraints. Hence, finding the global optimal solution is not likely. The prevalent objective of previous researchers has been to obtain a local optimal solution using heuristics. Most of the previous heuristic methods are very problem specific, the final solution often depends on the initial solution, and they find only local optimal solutions or simply feasible solutions. This led us to develop a multi echelon algorithm which can solve a variety of AGS problems with minor adjustments in the program coding. The method should be capable of finding good initial solutions with little input data, it should be easy to permute from one state to another state, and the method should search for the global optimum rather than a local optimum.

When applied to actual athletic games, previous research required the inclusion of many constraints as evidenced by the literature review. In this research we initially eliminate most of the constraints. We will describe a 0-1 integer program that helps to gain an understanding and insight into the AGS problem. We assume two different scenarios. In the first scenario there is no constraint on the number of consecutive home or away games a team might play. We add a constraint on consecutive home or away games in the second scenario.

Problem Formulation

This problem consists of n teams, an even number, and n sites. Teams must be at their home site at the beginning and after finishing their schedule. Each team needs to visit each of the other team's home site once. A team can visit another site only when the homeowner is at the site. We assume symmetric distances between teams. The notation used throughout the paper is:

i, j : teams,

t: time slot or game day,

n: number of teams,

D_{ij}: distance between team i's home and team j's home,

m: maximum number of consecutive home or road games allowed,

y^t_{ijk} : game between team i and j at location k at time t where k is either i or j,

G_{it}: game schedule or opponent of team i at time t,

h_{it}: home and away schedule for team i at time t,

Z: the total travel distance completed by all teams.

The objective is to minimize the total distance traveled by all teams. The problem can be formulated as a 0-1 integer program. The objective function is

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{t=1}^{2n-1} y^{t-1}_{ijk}\, y^{t}_{ijl}\, D_{kl} ,$$

where

$$y_{ijk}^{t} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ has game at location } k \text{ at time } t \text{ where } t = 1, 2, \cdots, 2(n-1) \\ 0 & \text{otherwise} \end{cases}.$$

The problem we are considering has home and away game. Thus k equals i if the game is held at i's home and j otherwise. Since game between team i and j can be held at only one place, $y_{iji}^{t} + y_{ijj}^{t} = 1$ for $i \neq j$.

$$\sum_{k=1}^{n} \sum_{t=1}^{2(n-1)} y_{ijk}^{t} = 2 \quad i \neq j, \; for \; i = 1, 2 \cdots, n \; and \; j = 1, 2, \cdots, n.$$

Since team i and j have a home and a road game, there is a total of 2 games between them during a season.

$$\sum_{j=1}^{n} \sum_{k=1}^{n} y_{ijk}^{t} = 1 \qquad for \; i = 1, 2 \cdots, n \; and \; t = 1, 2, \cdots, 2(n-1)$$

$$\sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{t=1}^{2(n-1)} y_{ijk}^{t} = 2(n-1) \qquad for \; i = 1, 2 \cdots, n$$

A team i can play in only one game on a game day. In a season, therefore, each team has $2(n-1)$ games. During the season a team cannot have a game against themselves, so $y_{iii}^{t} = 0$ *for* $t = 1, \cdots\cdots, 2(n-1)$. In order to position the team at home at the beginning and end of the season, we define $y_{iii}^{0} = 1$, $y_{iii}^{2n-1} = 1$, $y_{ijk, i \neq j}^{0} = 0$ and $y_{ijk, i \neq j}^{2n-1} = 0$.

Let

$$h_{it} = \begin{cases} 1 & if \; y_{ijk}^{t} = 1 \; and \; j = k \; for \; t = 1, \cdots, 2(n-1) \\ 0 & if \; y_{ijk}^{t} = 1 \; and \; i = k \; for \; t = 1, \cdots, 2(n-1) \end{cases}$$

then

$$\sum_{t=1}^{2(n-1)} h_{it} = n-1 \qquad i=1,\cdots,n \text{ and } j=1,\cdots,n,$$

the total number of road games for each team is *n-1* and

$$\sum_{i=1}^{n} h_{it} = \frac{n}{2} \qquad t=1,\cdots,2(n-1) \text{ and } j=1,\cdots,n,$$

there are a total of *n/2* games played on each game day.

$$h_{it} + h_{it+1} + \cdots\cdots + h_{it+m-1} + h_{it+m} \geq 1.$$

$$h_{it} + h_{it+1} + \cdots\cdots + h_{it+m-1} + h_{it+m} \leq m.$$

The total number of consecutive home or road games is limited to *m*.

Algorithm

Generating the initial athletic game schedule

We assume that the number of teams and the distance between each team are known. The initial game schedule (G_{it}) is generated as follows (see Table 4 for example):

Step 1. At time $t = 0$, all teams begin at home.

Step 2. Generate the first team's schedule for the first half of the season (t = 1 to n-1) by simply scheduling 1 versus 2 at t = 1, then 1 versus 3 at t = 2 etc. to 1

Table 4. Half and initial full season schedule (G_{it})

	Time (t)			
i	0	1	2	3
1	1	2	3	4
2	2	1	4	3
3	3	4	1	2
4	4	3	2	1

	Time (t)							
i	0	1	2	3	4	5	6	7
1	1	2	2	3	3	4	4	1
2	2	1	1	4	4	3	3	2
3	3	4	4	1	1	2	2	3
4	4	3	3	2	2	1	1	4

versus n at $t = n$-1. Schedule a game with team 1 on each of the other team's schedule corresponding to the initial half season schedule of team 1.

Step 3. Schedule for next team i, i = 2 to nth team, for a half season.

3-1. From t = 1 to n-1, assign any team k who has not yet been scheduled to play at time t.

3-2. Place team i on team k's schedule at time t.

Step 4. When the first half season schedule is complete, simply duplicate columns to produce the full season schedule as shown in Table 4.

Step 5. After finishing all scheduled games, all teams return home at $t = 2n$-1.

Table 5. Initial home and away schedule (h_{it}) for 4 teams

	Time (t)								
i	0	1	2	3	4	5	6	7	Total
1	0	1	0	1	0	1	0	0	3
2	0	0	1	1	0	1	0	0	3
3	0	1	0	0	1	0	1	0	3
4	0	0	1	0	1	0	1	0	3

0: home game 1: road game

After obtaining the initial G_{it} (game schedule), an initial h_{it} (home and away schedule) will be generated (see Table 5).

Step 1. All teams stay home at t = 0 and $2n$-1 (i.e. all h_{it} = 0 for $i = 1, 2, \cdots, n$).

Step 2. Choose team i = 1 to n.

Step 2-1. Set t = 1,

Step 2-2. If h_{it} is not 0 or 1, assign 1 (Make first meeting a road game).

Step 2-2-1. For $j = t + 1$ to $2(n-1)$, if $G_{it} = G_{ij}$, then $h_{ij} = 0$ and set $s = j$ (Make second meeting a home game).

Step 2-2-2. For $k = i + 1$ to n, if $G_{kt} = i$, then $h_{kt} = 0$ and $h_{ks} = 1$ (Set complementary home/road schedule for team i's competitor).

Step 2-3. If $t < 2(n-1)$, $t = t + 1$ and go to Step 2-2. Otherwise, go to Step 2.

Step 3. If consecutive home/road games constraint is given (m) proceed to Step 4. Otherwise, stop.

Step 4. If there is not more than m consecutive home or road games for any team in the schedule, then stop. Otherwise, let h_{it} be the (m+1)th consecutive home or road game. Change h_{it} to its complementary number (i.e. if $h_{it} = 0$, the new $h_{it} = 1$, and vise versa).

Step 4-1. For $j = 1$ to $2(n-1)$, if $G_{it} = G_{ij}$ and $j \neq t$, then change h_{ij} to its complementary number and set $s = j$ (Change location of other meeting of the two teams).

Step 4-2. For $k = 1$ to n, if $G_{kt} = i$ and $i \neq k$, then change h_{kt} and h_{ks} to their complementary numbers and go to Step 4 (Change home/road schedule for team i's competitor).

Multi echelon simulated annealing

The simulated annealing (SA) algorithm is motivated from the behavior of physical systems in the presence of a heat bath and based on ideas from statistical mechanics (Johnson *et al.*, 1989). The SA algorithm was developed by Kirkpatrick, Gelatt and Vecchi (1983). To find a global optimal solution, they applied the Metropolis procedure (Metropolis *et al.*, 1953) within SA's inner loop.

The Metropolis procedure defines 1) Z, the value for the current best solution, 2) Z', the value for the state being evaluated, 3) T, the current temperature from the annealing schedule, 4) $F(x)$, the probability of accepting a worse solution, and 5) U, a uniform random variate on the interval [0,1]. For a minimization problem, if Z' is less than Z, we accept the new state as the current state, otherwise we accept Z' depending on a negative exponential probability distribution. The probability of accepting the state with a higher objective function value is given by

$$F(x) = \exp\left[\frac{-(Z'-Z)}{T}\right].$$

After a value for U is generated, we accept Z' if $U < F(x)$.

Johnson *et al.* (1989) showed SA performs similar or better than many other search algorithms. However, many researchers have observed that SA requires a long run time in order to perform well. Since SA is applied as a heuristic using random number generation, a final solution found with the SA algorithm is not guaranteed to be a global optimum.

We design a multi echelon simulated annealing algorithm. Echelon 1 swaps two columns in the game schedule (G_{it}) and sends the new game schedule to echelon 2. Two rows in the game schedule are swapped at echelon 2 and the home or road schedule is initialized. Echelon 3 iterates different home or road game schedules. At echelon 1, the game days are chosen randomly, then the corresponding columns in the game schedule are swapped. The new G_{it} is then used in echelon 2. In echelon 2 two rows of the game schedule are chosen randomly and swapped. The procedure for swapping rows at echelon 2 is as follows:

Step 1. Choose two random numbers, a and b, representing two different teams.

Step 2. For t = 1 to 2(n-1), if $G_{at} \neq b$, swap G_{at} and G_{bt}.

Step 3 Set i = 1.

Step 4. If $i \neq a$ and $i \neq b$, go to Step 4-1.

Otherwise, if $i < n$, increase i by 1 else stop.

Step 4-1. For t = 1 to $2(n-1)$, if $G_{it} = a$ then set $G_{it} = b$ or if $G_{it} = b$ then set $G_{it} = a$. Increment i and go to Step 4.

For example, let a and b be 1 and 3 respectively. In Step 2, the circled numbers will be swapped. The numbers indicated with squares will be swapped in Step 4 (see Tables 6 and 7)

For a given G_{it} (game schedule) from echelon 2, an initial h_{it} (home and away schedule) is generated for echelon 3 in the same manner as discussed above. At echelon 3 h_{it} is permutated until the annealing procedure is frozen. The Z' value at the frozen state from

Table 6. Original game schedule (G_{it})

	Time (t)							
i	0	1	2	3	4	5	6	7
1	1	2	2	3	3	4	4	1
2	2	1	1	4	4	3	3	2
3	3	4	4	1	1	2	2	3
4	4	3	3	2	2	1	1	4

Table 7. New game schedule (G'_{it}) after swapping in echelon 2

	Time (t)							
i	0	1	2	3	4	5	6	7
1	1	4	4	3	3	2	2	1
2	2	3	3	4	4	1	1	2
3	3	2	2	1	1	4	4	3
4	4	1	1	2	2	3	3	4

echelon 3 will be accepted/rejected dependent upon the annealing state at echelon 2. The new h_{it}'s at echelon 3 are obtained as follows:

Step 1. Randomly generate two numbers, a and c, representing team and time.

Step 2. Change h_{ac} to its complementary number.

Step 2-1. For $j = 1$ to $2(n-1)$, if $G_{ac} = G_{aj}$ and $j \neq c$, then change h_{aj} to its complementary number and set $s = j$.

Step 2-2. For $k = 1$ to n, if $G_{kc} = a$ and $a \neq k$, then change h_{kc} and h_{ks} to their complementary numbers.

Step 3. If there is not more than m consecutive home or road games for any team in the schedule, then stop. Otherwise, let h_{it} be the $(m+1)$th consecutive home or road game. Change h_{it} to its complementary number.

Step 3-1. For $j = 1$ to $2(n-1)$, if $G_{it} = G_{ij}$ and $j \neq t$, then change h_{ij} to its complementary number and set $s = j$ (Changing location of other meeting of the two teams).

Step 3-2. For $k = 1$ to n, if $G_{kt} = i$ and $i \neq k$, then change h_{kt} and h_{ks} with their complementary numbers and go to Step 3 (Changing home/road schedule for team i's competitor).

The multi echelon simulated annealing algorithm is:

Step 1. If echelon 1 is not frozen, choose game days a and b, ($a \neq b$), at random. Swap G_{ia} and G_{ib} for $i = 1$ to n.

Step 2. If echelon 2 is frozen, go to Step 1. Otherwise, choose two rows and swap them as explained before, G_{it}' (see Tables 6 and 7).

Step 3. Initialize h_{it}, calculate Z, and set better = 0.

Step 4. Choose team *i*, and time *t*, $1 \le t \le 2(n-1)$, at random.

Step 5. Generate new h_{it}' and calculate Z'.

Step 6. If $Z' < Z$, replace Z, G_{it}, and h_{it} with Z', G_{it}', and h_{it}'. Otherwise,

use metropolis method to determine the acceptance of new state.

If the current solution is strictly better than the best solution so far, increase

better = better+1.

Step 7. If echelon 3 is frozen or *better* ≥ 2, go to Step 2. Otherwise, go to Step 4.

We have $2n$-2 time zones for each team for the entire season. Also, each team has two games with all the other teams. Therefore, the total number of possible permutations at echelon 1 is $(2n-2)!$. The total number of possible permutations at each echelon 2 is $n!$. The total number of variables in h_{it} is $n \times (2n-2)$. Also, at each permutation 4 related elements are changed. Hence, the total number of possible permutations at echelon 3 is $2^{n(2n-2)/4} = 2^{(n^2-n)/2}$.

The total possible combinations of all echelons are $(2n-2)! \times n \times 2^{(n^2-n)/2}$. It would take a prohibitive amount of time to permute through all possible combinations. Within the SA algorithm we limited the total number of iterations at each echelon. Echelon 1 iterates $n!$ times after which it is considered frozen and the procedure is terminated. Echelons 2 and 3 iterate $n\times100$ and $n\times500$ times respectively. We use an annealing factor of 0.9 for all echelons. m is equal to 2 for all examples when a constraint on consecutive home or road games is included. The algorithm was implemented using C++ and was run on Pentium II 233 *Mhz* computers.

Application and Results

Examples for 4 teams

Five different example problems consisting of four teams were developed. The distances between teams were generated randomly (see Table 8). Table 9 shows the initial game schedule and corresponding home or road game schedule. The initial schedules are the same with or without a consecutive home or road game constraint m. The total travel distances for the initial solutions to examples 1 through 5 are given in Table 9

When there is no constraint on consecutive home or road games, the final solution for example 1 improved 33% from the initial solution. Example 2 through 5 improved 38%, 39%, 35%, and 36% respectively from their initial solutions (Table 10). Table 11 shows the final results when the consecutive home or road game constraint was included with $m = 2$.

Table 8. Distance (D_{ij}) tables for 4 team examples

Team	1	2	3	4	Team	1	2	3	4
1	0	150	200	350	1	0	35	86	13
2	150	0	230	300	2	35	0	98	28
3	200	230	0	270	3	86	98	0	37
4	350	300	270	0	4	13	28	37	0
Example 1					Example 2				
1	0	154	116	23	1	0	155	177	283
2	154	0	89	31	2	155	0	295	209
3	116	89	0	79	3	177	295	0	211
4	23	31	79	0	4	283	209	211	0
Example 3					Example 4				

Team	1	2	3	4
1	0	46	48	45
2	46	0	5	47
3	48	5	0	12
4	45	47	12	0
Example 5				

Examples 1 to 5 improved 19%, 22%, 20%, 21%, and 20% respectively from the initial solutions.

Examples for 6 teams

Examples 6 to 10 include 6 teams. The initial travel distances for examples 6 through 10 are given in Table 12. The final results from multi echelon simulated annealing (SA) without including a consecutive home or road game constraint are also given in Table 12. The range of improvement over the initial solutions was 38% to 57%. With m = 2, the multi echelon simulated annealing yielded improvements of 19% to 36% from their initial solutions (see Table 12).

Table 9. Initial solutions for examples 1 to 5

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	i	0	1	2	3	4	5	6	7
1	1	2	2	3	3	4	4	1	1	0	1	0	1	0	1	0	0
2	2	1	1	4	4	3	3	2	2	0	0	1	1	0	1	0	0
3	3	4	4	1	1	2	2	3	3	0	1	0	0	1	0	1	0
4	4	3	3	2	2	1	1	4	4	0	0	1	0	1	0	1	0
Game schedule (G_{it})									Home/road schedule(h_{it})								
Initial total travel distance for example 1 = 5900																	
Initial total travel distance for example 2 = 1138																	
Initial total travel distance for example 3 = 1806																	
Initial total travel distance for example 4 = 5239																	
Initial total travel distance for example 5 = 764																	

Table 10. Final result without *m* for example 1 to 5.

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	4	2	4	3	1	1	0	0	0	0	1	1	1	0
2	2	1	4	3	1	3	4	2	2	0	1	0	0	0	1	1	0
3	3	4	1	2	4	2	1	3	3	0	1	1	1	0	0	0	0
4	4	3	2	1	3	1	2	4	4	0	0	1	1	1	0	0	0
Game schedule									Home or road schedule								
Final total travel distance for example 1= 3940																	
	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	4	3	2	3	4	2	1	1	0	0	0	0	1	1	1	0
2	2	3	4	1	4	3	1	2	2	0	0	0	1	1	1	0	0
3	3	2	1	4	1	2	4	3	3	0	1	1	1	0	0	0	0
4	4	1	2	3	2	1	3	4	4	0	1	1	0	0	0	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 2= 702																	
	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	2	4	3	2	4	1	1	0	1	1	1	0	0	0	0
2	2	4	1	3	4	1	3	2	2	0	0	0	0	1	1	1	0
3	3	1	4	2	1	4	2	3	3	0	0	0	1	1	1	0	0
4	4	2	3	1	2	3	1	4	4	0	1	1	0	0	0	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 3= 1108																	
	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	2	4	3	2	1	1	0	1	1	0	0	0	1	0
2	2	4	3	1	3	4	1	2	2	0	1	1	1	0	0	0	0
3	3	1	2	4	2	1	4	3	3	0	0	0	1	1	1	0	0
4	4	2	1	3	1	2	3	4	4	0	0	0	0	1	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 4 = 3429																	
	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	4	2	3	4	1	1	0	1	1	1	0	0	0	0
2	2	1	4	3	1	4	3	2	2	0	0	1	1	1	0	0	0
3	3	4	1	2	4	1	2	3	3	0	1	0	0	0	1	1	0
4	4	3	2	1	3	2	1	4	4	0	0	0	0	1	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 5 = 492																	

Table 11. Final result with m = 2 for example 1 to 5.

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	4	3	4	2	1	1	0	1	0	0	1	1	0	0
2	2	1	4	3	4	3	1	2	2	0	0	1	0	0	1	1	0
3	3	4	1	2	1	2	4	3	3	0	0	1	1	0	0	1	0
4	4	3	2	1	2	1	3	4	4	0	1	0	1	1	0	0	0
Game schedule									Home or road schedule								
Final total traveling distance for example 1= 4760																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	2	4	3	4	3	1	1	0	0	1	0	0	1	1	0
2	2	1	1	3	4	3	4	2	2	0	1	0	1	1	0	0	0
3	3	4	4	2	1	2	1	3	3	0	0	1	0	1	1	0	0
4	4	3	3	1	2	1	2	4	4	0	1	0	1	0	0	1	0
Game schedule									Home or road schedule								
Final total traveling distance for example 2 = 888																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	4	4	2	3	2	3	1	1	0	0	1	0	1	1	0	0
2	2	3	3	1	4	1	4	2	2	0	1	0	1	1	0	0	0
3	3	2	2	4	1	4	1	3	3	0	0	1	0	0	1	1	0
4	4	1	1	3	2	3	2	4	4	0	1	0	1	0	0	1	0
Game schedule									Home or road schedule								
Final total traveling distance for example 3 = 1453																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	2	4	3	4	3	1	1	0	0	1	0	0	1	1	0
2	2	1	1	3	4	3	4	2	2	0	1	0	1	1	0	0	0
3	3	4	4	2	1	2	1	3	3	0	1	0	0	1	1	0	0
4	4	3	3	1	2	1	2	4	4	0	0	1	1	0	0	1	0
Game schedule									Home or road schedule								
Final total traveling distance for example 4 = 4144																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	2	3	4	4	1	1	0	0	1	1	0	0	1	0
2	2	1	4	1	4	3	3	2	2	0	1	0	0	1	1	0	0
3	3	4	1	4	1	2	2	3	3	0	0	0	1	1	0	1	0
4	4	3	2	3	2	1	1	4	4	0	1	1	0	0	1	0	0
Game schedule									Home or road schedule								
Final total traveling distance for example 5 = 614																	

Table 12. Comparison of the initial and final solutions

Example #	Final Solution				
	Initial	Final w/o *m*	% improvement	Final w/ *m*	% improvement
1	5900	3940	33	4760	19
2	1138	702	38	888	22
3	1806	1108	39	1453	20
4	5236	3429	35	4144	21
5	764	492	36	614	20
6	12292	7659	38	9925	19
7	9653	5659	41	7780	19
8	1797	994	45	1314	27
9	2618	1379	47	1928	26
10	6294	2730	57	4070	35
11	25084	15875	37	20207	19

Example for 8 teams

One example was completed with 8 teams (example 11). The total travel distance for the initial schedule was 25,048. The final result without *m* is 15,875 (a 37% improvement from its initial solution) and the final schedule with $m = 2$ was 20,207 (a 19% improvement from the initial solution). The results can be seen in Table 12.

Conclusions

In this research, we have introduced AGS formulated as a 0-1 integer program, and shown a systematical approach to the AGS problem either with or without a constraint on the number of consecutive home or road games. We used a multi echelon SA algorithm to search for good solutions. In every example, regardless of whether a constraint on consecutive home or road games was included, we showed substantial improvement over the initial solution. When the consecutive home or road game constraint is included, the final solution is always somewhat worse than when there is no constraint. Figure 1 shows percent of improvement from the initial solutions to the final solutions.

The methodology proposed in this paper may not be directly applicable to some existing AGS problems such as the National Basketball Association (NBA), the National Football League (NFL), Major League Baseball (MLB), the National Hockey League (NHL), etc. However, it provides good insight to the AGS problem and it will be fairly simple to include other constraints in order to apply the method to various AGS problems. In the future this research will be extended to include additional situations such as odd number of teams, allowing game postponement, non symmetric traveling distances, inter-league games for two or more leagues, consideration of the previous season's schedule, TV broadcasting schedule, rival games, etc. Other performance measures may also be examined such as maximizing the total number of fans.

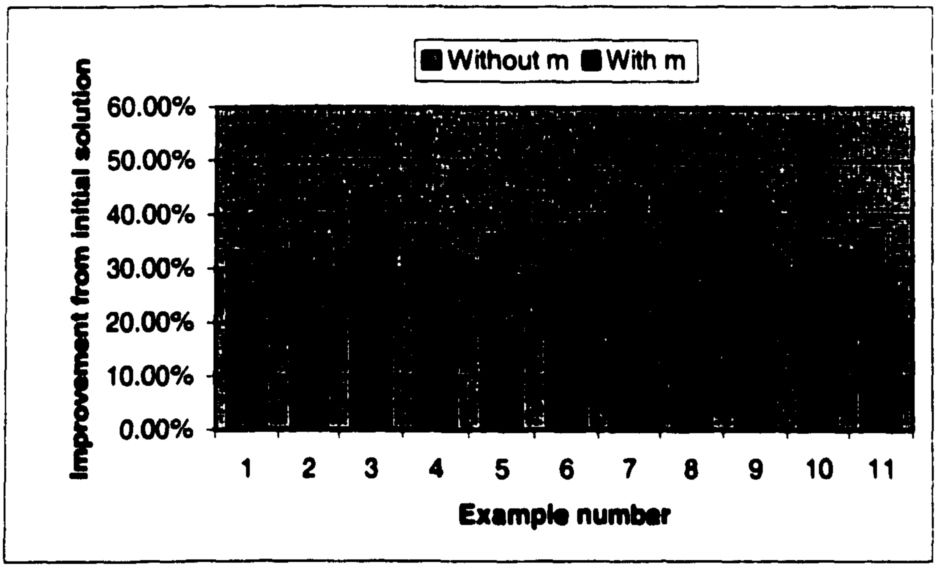

Figure 1. The improvement of the final solutions from the initial solutions

CHAPTER II. ATHLETIC GAME SCHEDULING USING TABU SEARCH

Introduction

In PART I, we developed a multi echelon heuristic to the athletic game scheduling (AGS) problem that adopted simulated annealing (SA) in the final echelon. We exploit multi echelon Tabu search (Tabu) algorithm in this research and the final solutions will be compared to the results in PART I.

Finding a good athletic game schedule is a complicated problem that can include many constraints. Most of the previous research has been done on developing heuristics because of the large size of the problem when formulating in mathematical form and constraints that can not be even formulated. Hence, finding the global optimal solution is not likely. The major objective of previous researchers has been to obtain of a local optimal solution.

Many of the previous heuristics are very problem specific. The final solution often depends on the initial solution, and often they find only local optimal solutions or simply feasible solutions. We developed the use of a multi-echelon algorithm that can solve a variety of scheduling problems with minor adjustments. Our goal was to develop a method capable of finding good initial solutions with little input data, it must be easy to permute from one state to another state, and our method would search for the global optimum rather than a local optimum. We utilized a multi echelon Tabu search procedure to search for the schedule that minimizes the total travel distance of all teams during the course of a season.

Since Glover (1989) introduced Tabu search, many researchers have applied it to combinatorial problems. The advantage of Tabu search is the capability of escaping from local optimality. The ability of Tabu search to find a good solution depends on how users define three important measures: the size of the Tabu list, the life of each element in the Tabu list, and the aspiration criteria. Tabu search keeps *Tabu moves* (individual solutions found previously) in *Tabu lists* in an attempt to force the procedure to identify new solutions. At each iteration, the Tabu search determines if the *new move* (or solution) is in the *Tabu list*. If the *new move* is not in the *Tabu list*, it will be accepted as the most recent solution and be added to the *Tabu list*. Otherwise, an aspiration criteria is utilized to determine whether to accept the new solution even though it is presently *Tabu*. Also, the life of elements in the *Tabu lists* is considered at each iteration so that if an element has remained *Tabu* for a certain length of time (its life), then it is deleted from the *Tabu list*.

In this research we initially eliminated most of the constraints. We described a 0-1 integer program that helped to gain an understanding and insight into the AGS problem. We assumed two different scenarios in PART II, no constraint on the number of consecutive home or away games a team might play and then we added a constraint on the number of consecutive home or away games.

Algorithm

Generating the initial G_{it} and initial h_{it} is the same as given in PART I and provide an initial feasible solution to the AGS problem. We then search for improved solutions using Tabu search. Within the Tabu search procedure we use the following definitions:

- *Zopt*: performance measure value for the current best solution,

- *Znew*: performance measure value for the state being evaluated at echelon three,
- *Zaspire*: performance measure value for the state being evaluated in aspiration step.

We design a multi-echelon Tabu search algorithm. Each succeeding echelon searches for an optimal solution given the information provided by the preceding echelon. At echelon one, two game days are chosen randomly and the corresponding columns of the game schedule are swapped. Similarly, two rows of the game schedule from echelon one are chosen randomly and swapped at echelon two. The swap procedure at echelon two and the permutation procedure of h_{it} are the same as given in PART I.

Figure 2 shows the procedure used at echelon one of the multi-echelon Tabu search algorithm. *Tabu list* (the list of moves that are Tabu to make) is initialized before beginning the search procedure at echelon one. At echelon one $n!$ iterations are performed. For each iteration, two columns are chosen randomly and swapped. If the randomly chosen columns are in the *Tabu list*, the move may still be made if the *aspiration criteria* is met (that is, if certain criteria are met, the move can be made even though it is presently Tabu). If the new move is not presently on the Tabu list, then it is accepted as the next move and added to *Tabu list* (in this way, the same move will not be allowed in the immediate future). At each iteration the algorithm removes from the *Tabu list* any moves that have resided in the list for a predetermined length of time. Hence, those moves are no longer Tabu.

The aspiration criteria utilized in order to allow a Tabu move to be made is as follows. For a Tabu move, an initial home and road schedule is generated and the performance value, *Zaspire*, is computed (total travel distance). If *Zaspire* is less than the best solution found so far (*Zopt*), the new move is allowed. Once a new move is generated and accepted at echelon one, the resulting schedule is passed to echelon 2. The procedure used at echelon two is given in Figure 3. At echelon two, $n\times 100$ iterations are made each time a new move is received from echelon one. At each iteration, two teams (rows) are chosen randomly and swapped. The remaining Tabu search procedure is identical to that used at echelon one.

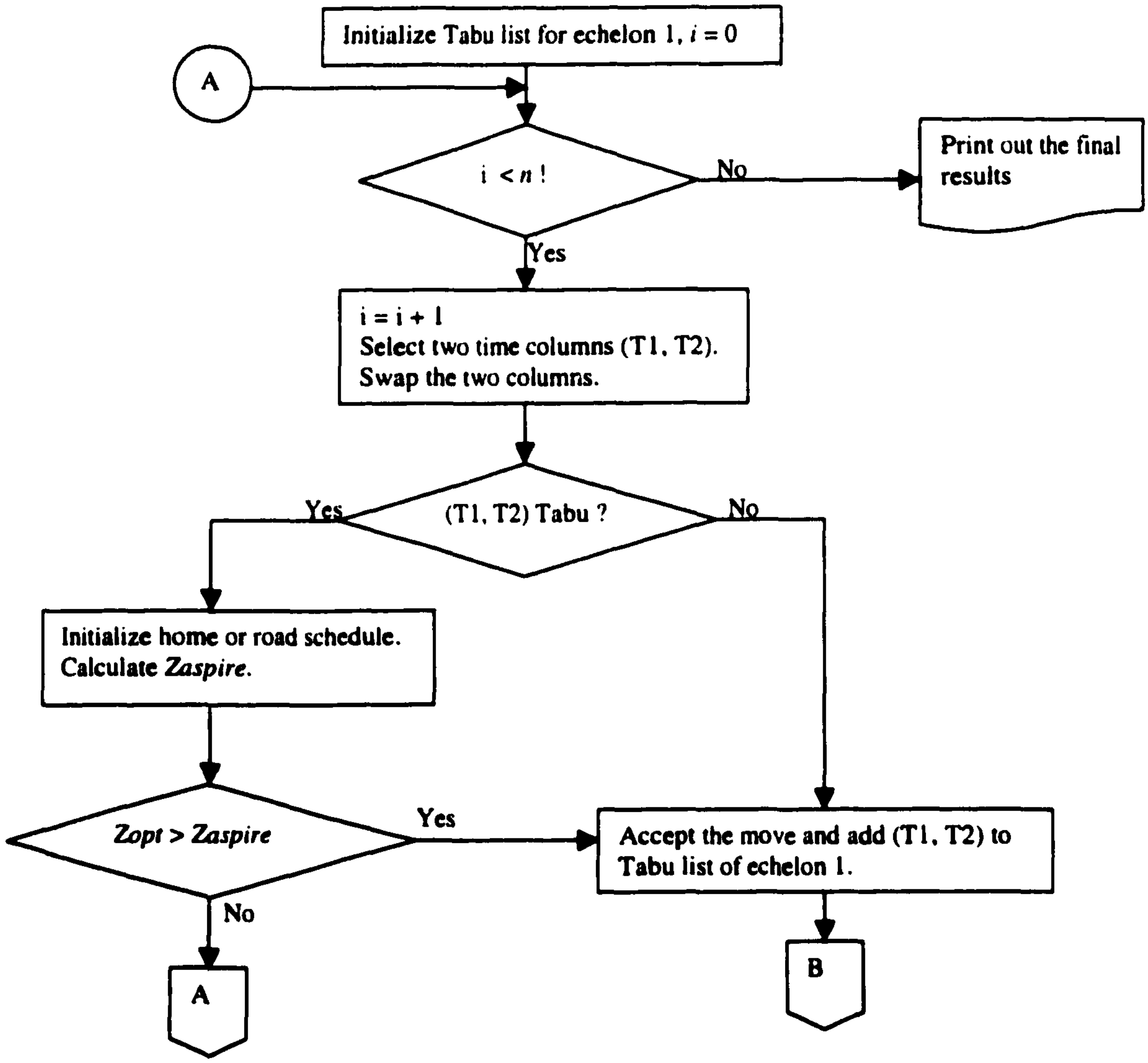

Figure 2. Tabu search at echelon 1 with *m*.

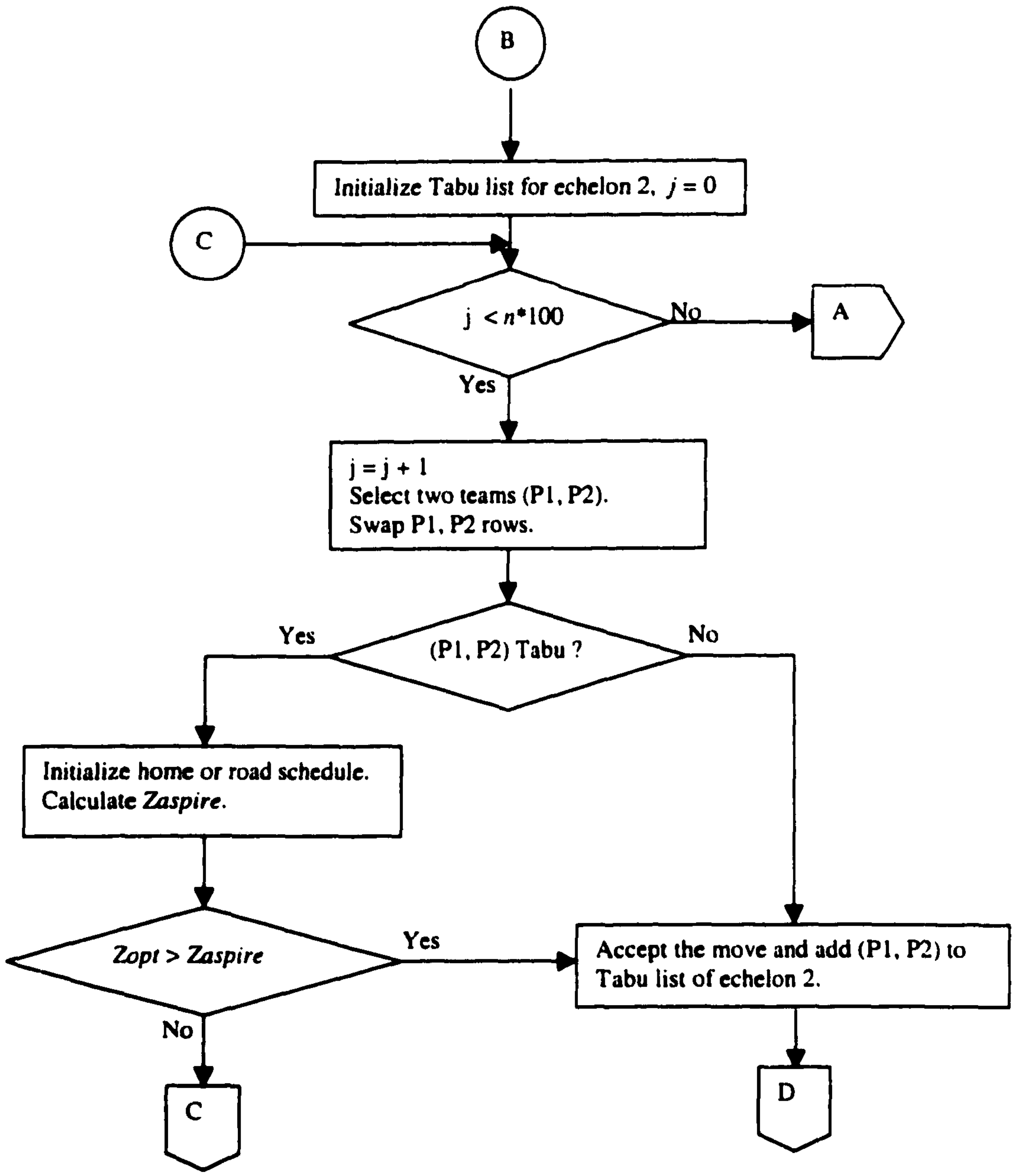

Figure 3. Tabu search at echelon 2 with *m*.

Again, each time a new move is generated and accepted at echelon two, the resulting schedule is passed to echelon three (Figure 4) where $n \times 500$ iterations are made. At each iteration, a team and a time are randomly generated and the corresponding home and away game schedules are changed. The Tabu search procedure then continues as before. The best solution found at echelon three is passed back up to echelon two, and the best solution found at echelon two is passed back up to echelon one. In this manner the procedure searches for the optimal solution to the AGS problems. Although the procedure cannot guarantee that the optimal solution will be found, good solutions are identified. The Tabu search algorithm was implemented using C++ and tested on a Pentium II 233 *Mhz* personal computer.

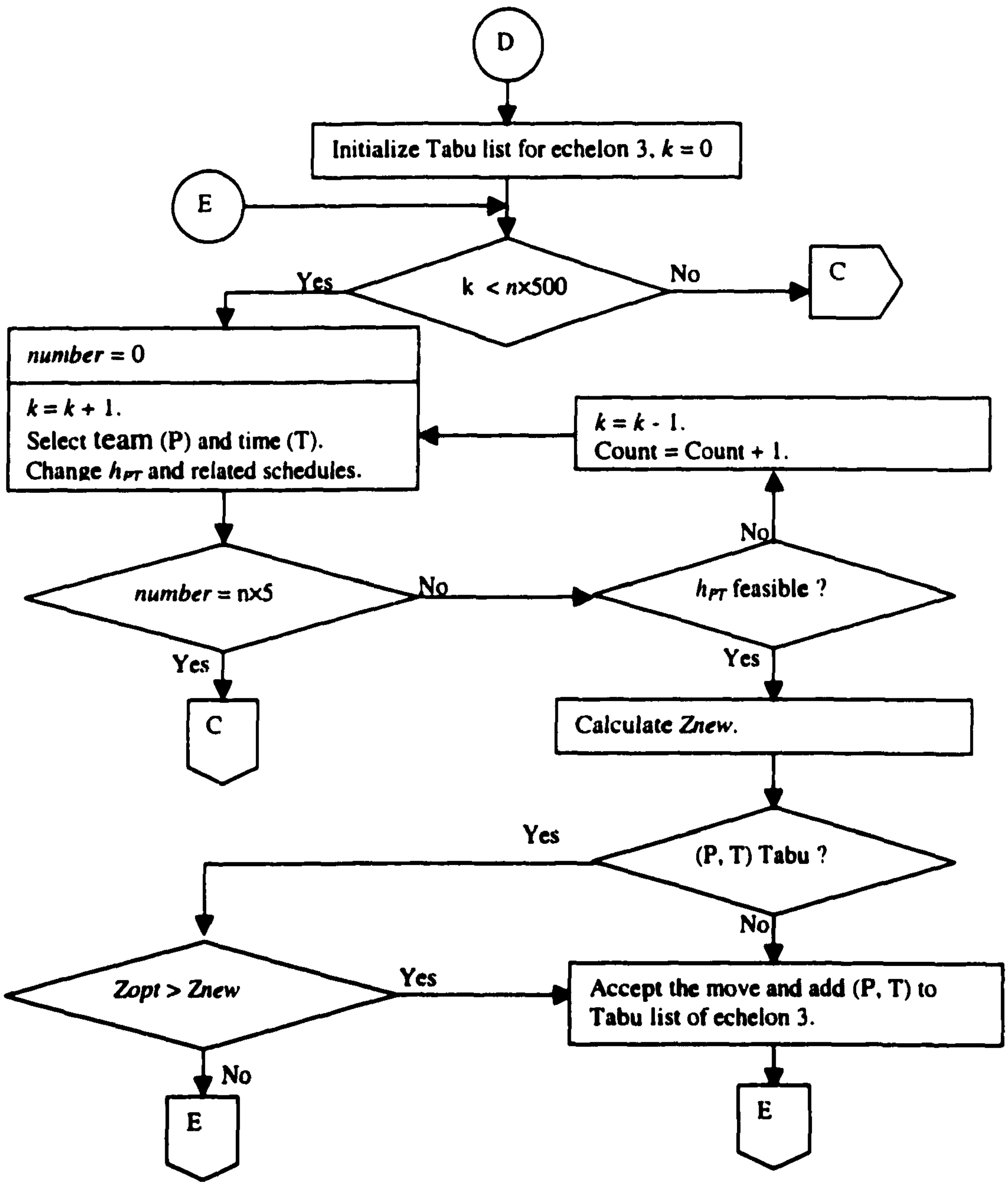

Figure 4. Tabu search at echelon 3 with *m*.

Examples

Examples for 4 teams

Five different example problems consisting of four teams were developed. The distances between teams were generated randomly (see Table 8 in PART I). We use the same *Tabu sizes* to all three echelons with $n\times3$. Table 9 shows the initial game schedule and corresponding home or road game schedule. The initial schedules are the same with or without a consecutive home or road game constraint m. The total travel distances for the initial solutions for example 1, 2, 3, 4 and, 5 are 5900, 1138, 1806, 5239, and 764 respectively as seen in Table 9. When there is no constraint on consecutive home or road games, the final solutions improved 33% to 38% from the initial solutions (Table 13). Table 14 shows the final results with allowing maximum 2 consecutive home or road games. And they are 4760, 888, 1453, 4144, and 614 for example 1, 2, 3, 4, and 5 respectively which improved between 19% to 22% from the initial solution.

Examples for 6 teams

Examples from example 6 to 10 are concerning with 6 teams. The *Tabu sizes* for all echelons are $n\times5$, and *life* of *Tabu lists* to all three echelons with $n\times4$. The initial travel distances for examples 6 through 10 are given in Table 15. The final results from multi-echelon Tabu search without including a consecutive home or road game constraint are also given in Table 15. The final solutions improved over the initial solutions 39%, 40%, 45%, 49%, and 57% to example 1, 2, 3, 4, and 5 respectively.

Table 13. Final results for example 1 to 5 without *m*

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	2	3	4	2	1	1	0	1	1	1	0	0	0	0
2	2	4	3	1	4	3	1	2	2	0	1	1	0	0	0	1	0
3	3	1	2	4	1	2	4	3	3	0	0	0	1	1	1	0	0
4	4	2	1	3	2	1	3	4	4	0	0	0	0	1	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 1= 3940																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	4	3	2	3	4	1	1	0	0	1	1	1	0	0	0
2	2	1	3	4	1	4	3	2	2	0	1	1	1	0	0	0	0
3	3	4	2	1	4	1	2	3	3	0	0	0	0	1	1	1	0
4	4	3	1	2	3	2	1	4	4	0	1	0	0	0	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 2= 702																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	2	4	4	2	3	1	1	0	0	0	0	1	1	1	0
2	2	4	1	3	3	1	4	2	2	0	1	1	0	1	0	0	0
3	3	1	4	2	2	4	1	3	3	0	1	1	1	0	0	0	0
4	4	2	3	1	1	3	2	4	4	0	0	0	1	0	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 3 = 1149																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	2	3	4	2	1	1	0	1	1	0	0	0	1	0
2	2	4	3	1	4	3	1	2	2	0	1	1	1	0	0	0	0
3	3	1	2	4	1	2	4	3	3	0	0	0	1	1	1	0	0
4	4	2	1	3	2	1	3	4	4	0	0	0	0	1	1	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 4 = 3429																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	4	3	2	2	3	4	1	1	0	0	0	0	1	1	1	0
2	2	3	4	1	1	4	3	2	2	0	1	1	1	0	0	0	0
3	3	2	1	4	4	1	2	3	3	0	0	1	1	0	0	1	0
4	4	1	2	3	3	2	1	4	4	0	1	0	0	1	1	0	0
Game schedule									Home or road schedule								
Final total travel distance for example 5 = 485																	

Table 14. Final result for example 1 to 5 with $m = 2$.

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	3	4	2	2	1	1	0	1	1	0	0	1	0	0
2	2	4	3	4	3	1	1	2	2	0	0	1	1	0	0	1	0
3	3	1	2	1	2	4	4	3	3	0	0	0	1	1	0	1	0
4	4	2	1	2	1	3	3	4	4	0	1	0	0	1	1	0	0
Game schedule									Home or road schedule								
Final total travel distance for example 1 = 4760																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	3	4	2	2	1	1	0	1	1	0	0	1	0	0
2	2	4	3	4	3	1	1	2	2	0	0	0	1	1	0	1	0
3	3	1	2	1	2	4	4	3	3	0	0	1	1	0	1	0	0
4	4	2	1	2	1	3	3	4	4	0	1	0	0	1	0	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 2 = 888																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	2	3	4	4	1	1	0	1	1	0	0	1	0	0
2	2	1	4	1	4	3	3	2	2	0	0	1	1	0	1	0	0
3	3	4	1	4	1	2	2	3	3	0	0	0	1	1	0	1	0
4	4	3	2	3	2	1	1	4	4	0	1	0	0	1	0	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 3 = 1453																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	3	4	3	4	2	2	1	1	0	1	1	0	0	1	0	0
2	2	4	3	4	3	1	1	2	2	0	0	1	1	0	0	1	0
3	3	1	2	1	2	4	4	3	3	0	0	0	1	1	0	1	0
4	4	2	1	2	1	3	3	4	4	0	1	0	0	1	1	0	0
Game schedule									Home or road schedule								
Final total travel distance for example 4 = 4144																	

	Time (t)									Time (t)							
i	0	1	2	3	4	5	6	7	*i*	0	1	2	3	4	5	6	7
1	1	2	3	2	3	4	4	1	1	0	1	1	0	0	1	0	0
2	2	1	4	1	4	3	3	2	2	0	0	1	1	0	0	1	0
3	3	4	1	4	1	2	2	3	3	0	1	0	0	1	1	0	0
4	4	3	2	3	2	1	1	4	4	0	0	0	1	1	0	1	0
Game schedule									Home or road schedule								
Final total travel distance for example 5 = 614																	

Table 15. Comparison of the initial and final solutions by Tabu search

Example Number	Initial Solution	Final Solution			
		Final w/o *m*	Improvement	Final w/ *m*	Improvement
1	5900	3940	33%	4760	19%
2	1138	702	38%	888	22%
3	1806	1149	36%	1453	20%
4	5236	3429	35%	4144	21%
5	764	485	37%	614	20%
6	12292	7493	39%	9976	19%
7	9653	5753	40%	7825	19%
8	1797	981	45%	1330	26%
9	2618	1324	49%	1949	26%
10	6294	2729	57%	4120	35%
11	25084	16109	36%	20413	19%

When we included constraint on consecutive home or road games (m = 2), the final result improved 18%, 19%, 26%, 26%, and 35% from their initial solutions for 6, 7, and 8, 9, and 10 respectively.

Example for 8 teams

We have developed one example for 8 teams (example 11). The *Tabu sizes* for all echelons are n×9, and *life* of *Tabu lists* to all three echelons with n×8. The total travel distance at initial schedule was 25048. The final result without m improved a 37% (16109) from its initial solution and the final schedule with m = 2 was 20207 (a 19% improvement form the initial solution). The results can be seen in Table 12.

Conclusions

In this research, we have solved AGS problems using multi echelon Tabu search. The percent of improvement from the initial results are given in Figure 5. As expected, AGS without constraining on consecutive home or road games, m, found a much better solution than with *m* since the best solution without *m* can be found when a team has consecutive road games without interruption. Therefore, it is the same as solving as many TSP as possible for all teams. However, if we consider *m* to mean that a team cannot play all road games consecutively, then this is the same as solving a multi travel salesmen problem (MTSP) for an individual team.

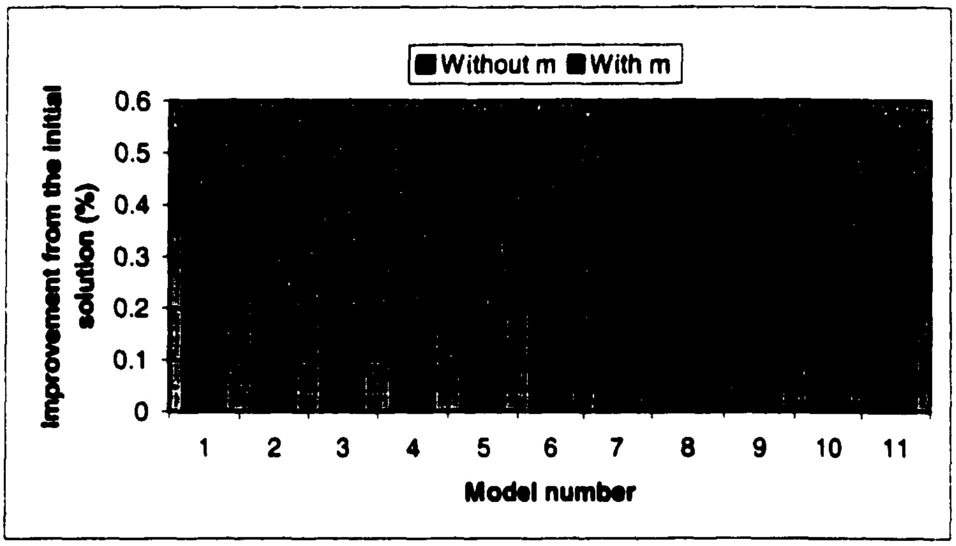

Figure 5. Comparison of the initial and final solutions by Tabu search

Comparison with SA without *m*

As we can see in Tables 16 and 17 and Figure 6, the final results from three examples are tied between Tabu search and SA and SA algorithm outperformed in three examples while Tabu search generated a better solution in 5 examples. The total CPU running times for all examples by SA and Tabu are shown in Table 17. Tabu search algorithm took less CPU times for all 4 teams examples; however, SA performed less CPU times in 3 examples for 6 teams game and 8 teams game.

Table 16. Comparison of final results from Tabu and SA (without *m*)

Example #	Initial	Final Solution		Tabu-SA
		Tabu	SA	
1	4040	3940	3940	0
2	845	702	702	0
3	1587	1149	1108	41
4	3689	3429	3429	0
5	492	485	492	-7
6	9294	7493	7659	-166
7	8156	5753	5659	94
8	1215	981	994	-13
9	1945	1324	1379	-55
10	3940	2729	2730	-1
11	18360	16109	15875	234

Table 17. Comparison of computation times (without *m*)

Example number	SA	Tabu
1	90.31	58.42
2	89.14	71.54
3	89.64	56.71
4	90.06	56.75
5	90.03	56.69
4 teams game scheduling (seconds)		
Example number	SA	Tabu
1	2.824598	2.777558
2	2.708283	2.792763
3	2.864085	2.801017
4	2.703226	2.787528
5	2.752611	2.781658
6 teams game scheduling (hours)		
Example number	SA	Tabu
1	383.771586	473.093219
8 teams game scheduling (hours)		

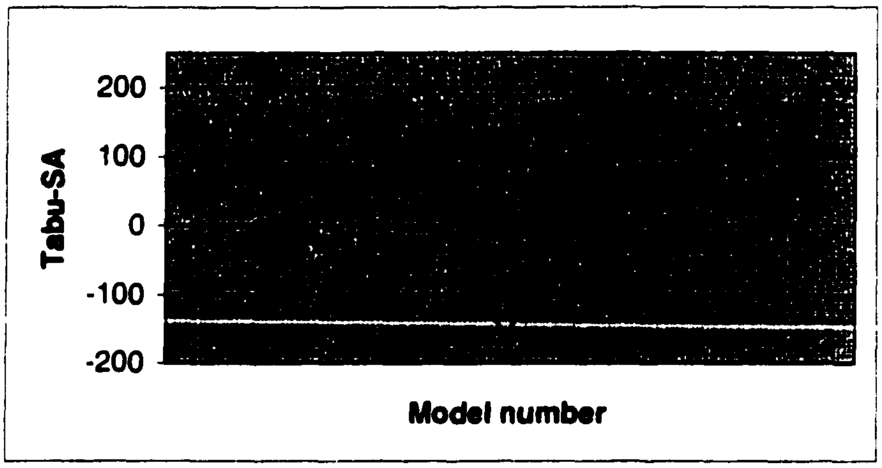

Figure 6. Differences of the final results between Tabu and SA

Comparison with SA with *m*

The results from both SA and Tabu are compared in Table 18 and Figure 7. As we see in Table 18, Both Tabu search and SA are tied in terms of travel distance for 4 teams game while SA took longer CPU times. In 2nd example of 6 teams game (7th overall) Tabu search found lower travel distance and spent less CPU times compared to the results from SA. On the other hand, in 8 teams game (11th example overall), SA algorithm performed better than Tabu search in both CPU time and travel distance.

According to Table 17 and 18, the longer CPU time does not guarantee the better results. Since both algorithms employ random numbers to exploit a different feasible region, the final objective function value and the total CPU time vary depending upon the random numbers and stopping criteria.

Though the constraints we have considered in this research are not realistic to the existing AGS problem such as NBA, NFL, MLB, NHL, etc, research can be extended to include odd number of teams, game postponement allowed, non symmetric traveling distances, inter-league game for two or more leagues, considering previous season schedule, TV broadcasting schedule, rival game, using different objective functions, *e.g.*, maximizing the total profit of all teams, etc.

Table 18. Comparison of the results from SA and Tabu with *m*

Total travel distances			Total CPU times (hrs)		
SA	Tabu	SA-Tabu	SA	Tabu	SA-Tabu
4760	4760	0	0.058	0.034	0.024
888	888	0	0.054	0.034	0.021
1453	1453	0	0.055	0.034	0.021
4144	4144	0	0.054	0.034	0.020
614	614	0	0.052	0.034	0.018
9925	9976	-51	3.39	1.22	2.17
7880	7825	55	3.47	1.35	2.12
1314	1330	-16	3.61	1.20	2.41
1928	1949	-21	3.25	1.16	2.09
4070	4120	-50	3.53	1.24	2.29
20207	20413	-206	81.34	154.04	-72.70

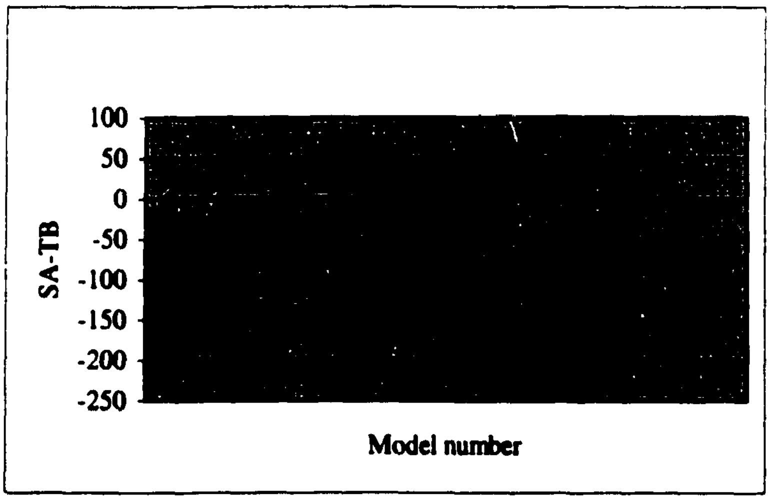

Figure 7. The difference of travel distances between SA and Tabu

CHAPTER III. FAIR ATHLETIC GAME SCHEDULING

Introduction

We have introduced an athletic game scheduling (AGS) using multi echelon heuristic in previous research and tried to minimize the total travel distance completed by all teams. The final results in PART I and PART II showed that reduction on travel distance by the heuristic is large. On the other hand, even in an optimal travel distance, a team might travel quite longer than the others, or vice versa. This causes a problem when determining the acceptance of an AGS schedule in real world since all teams in a league along with committees join and vote for the acceptance. It is therefore very important that a schedule not only minimizes the travel distance but also it needs to satisfy all parties that participate in the process of decision making for their future schedule. In this research we propose two methods for developing a fair AGS: minimizing the variance of the each team's travel distance (MVTD) and minimizing the distance of longest travel team (MDLT).

Problem Formulation

The objective function that minimizes the total travel distance is presented PART I and PART II as

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{2n-1} y_{ijk}^{t-1} y_{ijl}^{t} D_{kl},$$

where

$$y_{ijk}^{t} = \begin{cases} 1 & \textit{if } i \textit{ and } j \textit{ has game at location } k \textit{ at time } t \textit{ where } t = 1, 2, \cdots, 2(n-1) \\ 0 & \textit{otherwise} \end{cases}.$$

The model includes n teams represented as i and j and assumed symmetric distance between team i's home and team j's home (D_{ij}). h_{it} indicates whether team i is at home (0) or away (1) at time t and y^t_{ijk} is of game between team i and j at location k at time t. Therefore, if there exists a game between team i and j, $y^t_{ijk} = 1$, and if it is held at i's home then $k = i$, otherwise $k = j$. Since game between team i and j can be held at only one place, $y^t_{iji} + y^t_{ijj} = 1$ for $i \neq j$.

Let Z_i be a travel distance by team i at the end of season then

$$Z_i = \sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{2n-1} y^{t-1}_{ijk}\, y^t_{ijl}\, D_{kl}.$$

The mean travel distance is thus

$$\bar{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{Z}{n}$$

and the variance of all teams travel distance that we want to minimize (MVTD) is

$$Min\ \sigma^2_z = \frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \bar{Z})^2.$$

Also, MDLT can be obtained as

$$Min\ \{Max\ Z_i \text{ for } i = 1, \ldots, n\}.$$

The two performance measures are optimized using multi echelon heuristics that utilize simulated annealing (SA) in the last echelon. SA developed by Kirkpatrick, Gelatt and Vecchi (1983) applied the Metropolis procedure (Metropolis *et al.*, 1953) within SA's inner loop. The Metropolis procedure uses a negative exponential probability distribution to determine the acceptance of the current solution over the best solution found so far, which allows the SA algorithm to overcome entrapment in a local optimal solution.

Algorithm

The multi echelon heuristic consists of three nested echelons. Echelon 1 generates a new game schedule G_{it} by swapping two game days and then sends it to echelon 2. Echelon 2 swaps two rows of game schedule G_{it} received from echelon 1. Also, home or away schedule, h_{it}, is initialized at echelon 2 then sent to echelon 3 to permute a different h_{it} scenario. h_{it} consists of binary numbers, where 0 and 1 mean home and road game respectively for team i at time t. Swapping columns and rows of G_{it} and initialing and permuting h_{it} are explained in detail in PART I and PART II. The multi echelon algorithm for MVTD is as follows:

Step 1. If echelon 1 is not frozen, choose game days a and b, $(a \neq b)$, at random. Swap G_{ia} and G_{ib} for $i = 1$ to n.

Step 2. If echelon 2 is frozen, go to Step 1. Otherwise, choose two rows and swap them as explained before, G_{it}' (see Table 6 and 7 in PART I).

Step 3. Initialize h_{it} and calculate Z.

Step 4. Choose team i and time t, $1 \leq t \leq 2(n-1)$, at random.

Step 5. Generate new h_{it}' and calculate Z'.

Step 6. If $Z' < Z$, replace Z, G_{it}, and h_{it} with Z', G_{it}', and h_{it}'. Otherwise, use metropolis method to determine the acceptance of new state.

Step 7. If echelon 3 is frozen, go to Step 2. Otherwise, go to Step 4.

Also, the multi echelon algorithm for MDLT is:

Step 1. If echelon 1 is not frozen, choose game days a and b, $(a \neq b)$, at random. Swap G_{ia} and G_{ib} for $i = 1$ to n.

Step 2. If echelon 2 is frozen, go to Step 1. Otherwise, choose two rows and swap them as explained before, G_{it}' (see Table 6 and 7 in PART I).

Step 3. Initialize h_{it}, calculate Z_i and find Z_l, the longest travel.

Step 4. Find longest travel team x (Z_x) and choose time t, $1 \leq t \leq 2(n-1)$, at random.

Step 5. Generate new h_{it}' after changing h_{xt} with the complementary number, calculate Z_i' and find the current longest travel Z_l'.

Step 6. If $Z_l' < Z_l$, replace Z_l, G_{it}, and h_{it} with Z_l', G_{it}', and h_{it}'. Otherwise, use metropolis method to determine the acceptance of new state.

Step 7. If echelon 3 is frozen, go to Step 2. Otherwise, go to Step 4.

Echelon 1 and 2 are permuted for $n!$ and $n\times100$ times respectively. Also, echelon 3 is permuted for $n\times500$ times for MVTD and $n\times250$ times for MDLT model. We use annealing factor 0.9 for all echelons. The heuristics were implemented using C++ and were run on Pentium II 300 *Mhz* computers.

Examples

Examples for 4 teams

We used the same examples that were used in PART I and PART II. The final results from multi echelon algorithm improved to 5580, 1028, 1553, 5009, and 659 from their initial solutions for 1, 2, and 3, 4, and 5 respectively for MVTD model as seen in Table 19. When we minimized the distance of maximum traveling team (MDLT), the final solutions are 4920, 924, 1453, 4184, and 622 for examples from 1 to 5 respectively (Table 20). The results suggest that MDLT model is better than MVTD model in terms of total travel distance.

According to Figure 8 and 9, the MVDT model found more balanced travel distances to all team in example 1.

Examples for 6 teams

The final results from MVTD improved to 11345, 8566, 1398, 2291, and 5242 from their initial solutions for 6, 7, and 8, 9, and 10 respectively. We obtained 10042, 7966, 1357,

Table 19. Solutions from MVTD

Example Number	Travel distance		Average		Variance	
	Initial	Final	Initial	Final	Initial	Final
1	5900	5580	1475.0	1395.0	63566.7	633.3
2	1138	1028	284.5	257.0	13915.7	1988.7
3	1806	1553	451.5	388.3	23454.3	280.3
4	5236	5009	1309.8	1252.3	8026.9	543.6
5	764	659	191.0	164.8	4476.0	267.6
6	12292	11345	2032.0	1890.8	37924.0	29.8
7	9653	8566	1608.8	1427.7	55747.0	75.9
8	1797	1398	299.5	233.0	3773.9	51.2
9	2618	2291	436.3	381.8	3210.7	7.4
10	6294	5242	1049.0	873.7	11720.8	43.9
11	25084	23521	3131.0	2940.1	67539.1	64.1

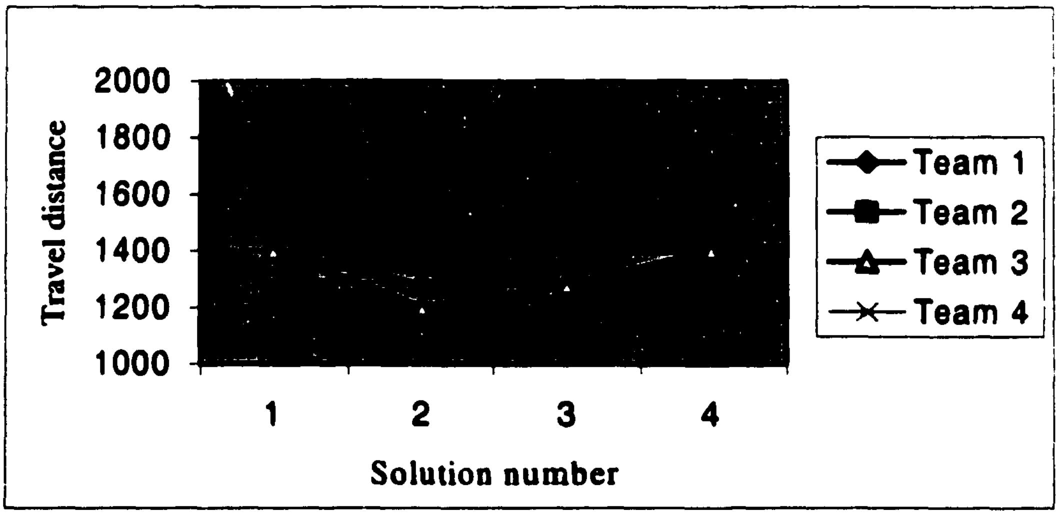

Figure 8. Individual travel distances model for example 1 by MVTD

Table 20. Solutions from MDLT

Example Number	Traveling distance		Average		Percentile Improvement
	Initial	Final	Initial	Final	
1	5900	4920	1475.0	1230.0	12%
2	1138	924	284.5	231.0	10%
3	1806	1453	451.5	363.3	6%
4	5236	4184	1309.8	1046.0	16%
5	764	622	191.0	155.5	6%
6	12292	10042	2032.0	1673.7	11%
7	9653	7966	1608.8	1327.7	7%
8	1797	1357	299.5	226.2	3%
9	2618	1973	436.3	328.8	14%
10	6294	4195	1049.0	699.2	20%
11	25084	20676	3131.0	2584.5	12%

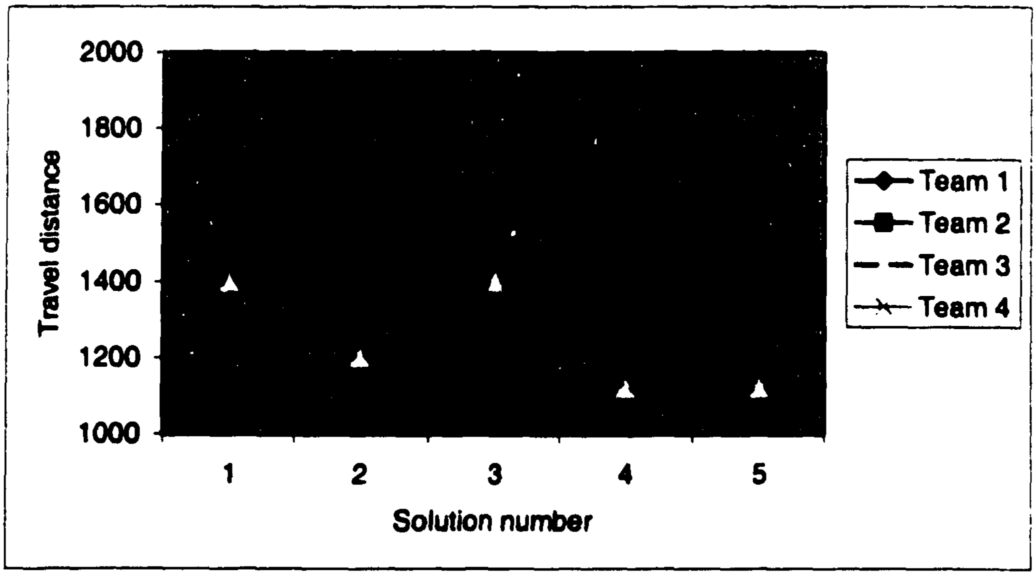

Figure 9. Individual travel distances for example 1 by MDLT

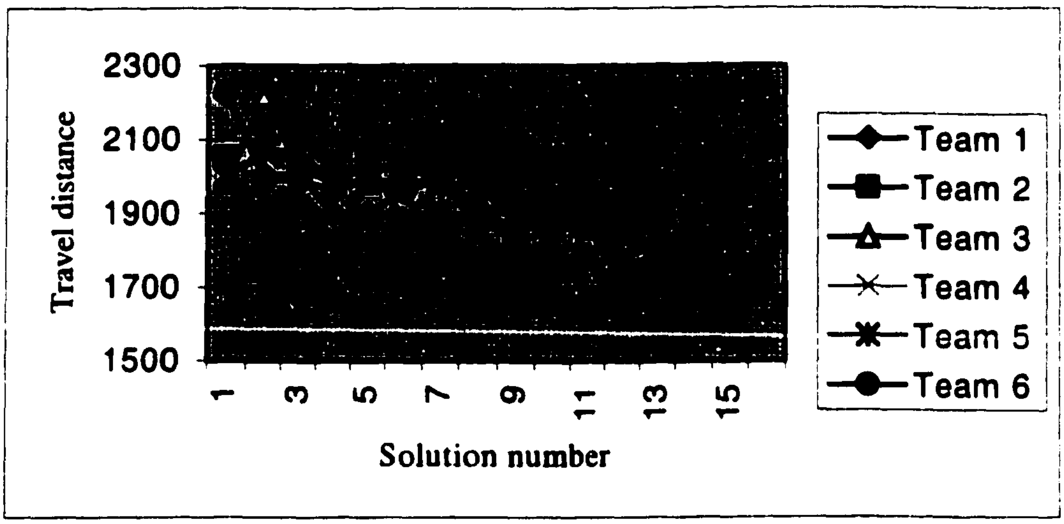

Figure 10. Individual travel distances for example 6 by MVTD

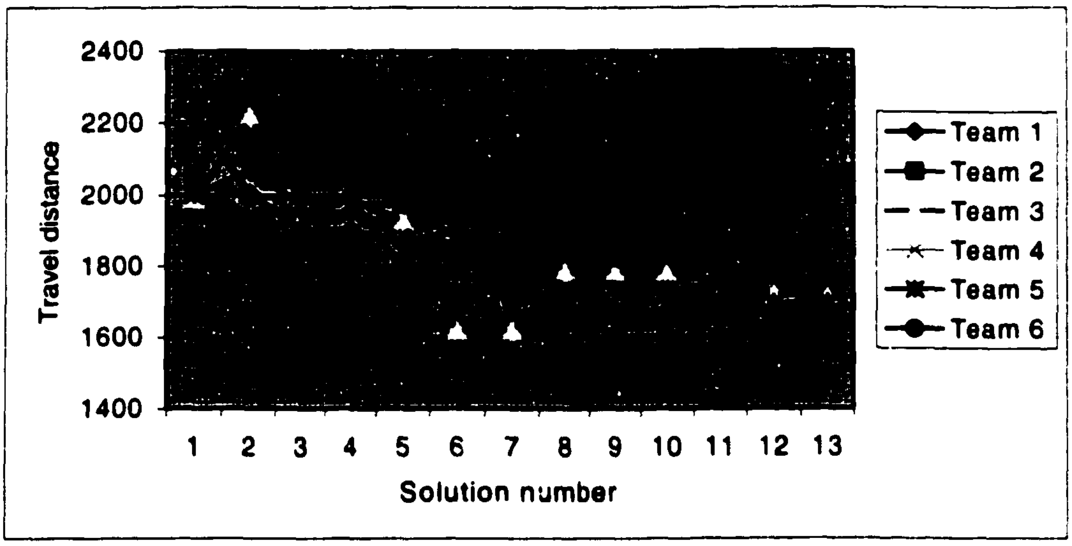

Figure 11. Individual travel distances for example 6 by MDLT

1973, and 4195 for examples from 6 to 10 respectively from the MDLT model. MVTD again generated a more balanced schedule to all teams in example 6 as shown in Figure 10 and 11.

Example for 8 teams

The final result from the MVTD model is 23521. And we obtained 20676 as a final solution from the MDLT method (see Table 21 and Figures 12 and 13). The MVTD model found more balanced travel distances to all teams than the MDLT model.

Conclusions

As we see in Table 19, the total travel distances from the MDLT method improved between 3% to 20% compared to those of the MVTD method. The minimum and maximum travel distances and variances of the final solutions are given in Table 21. It suggests that the

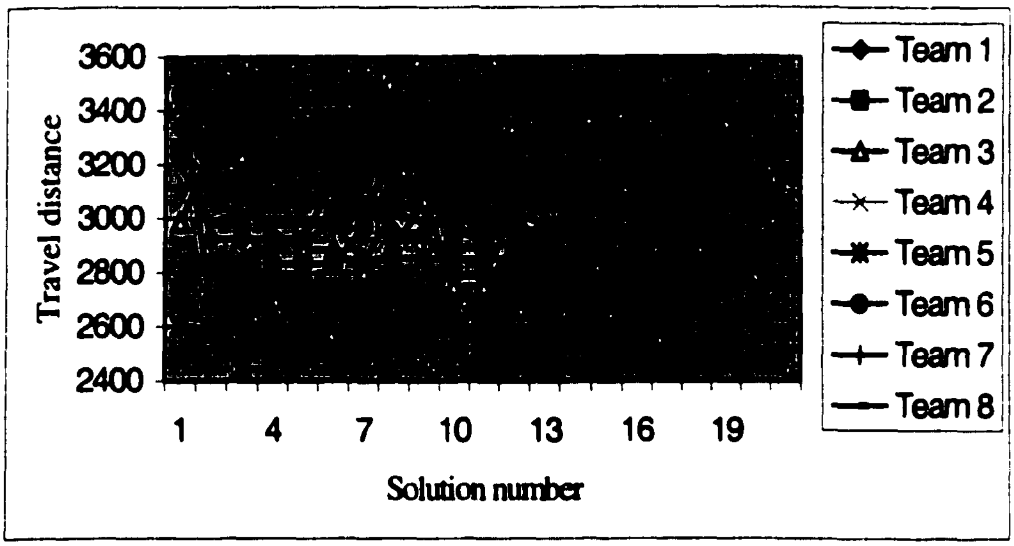

Figure 12. Individual travel distances for example 11 by MVTD

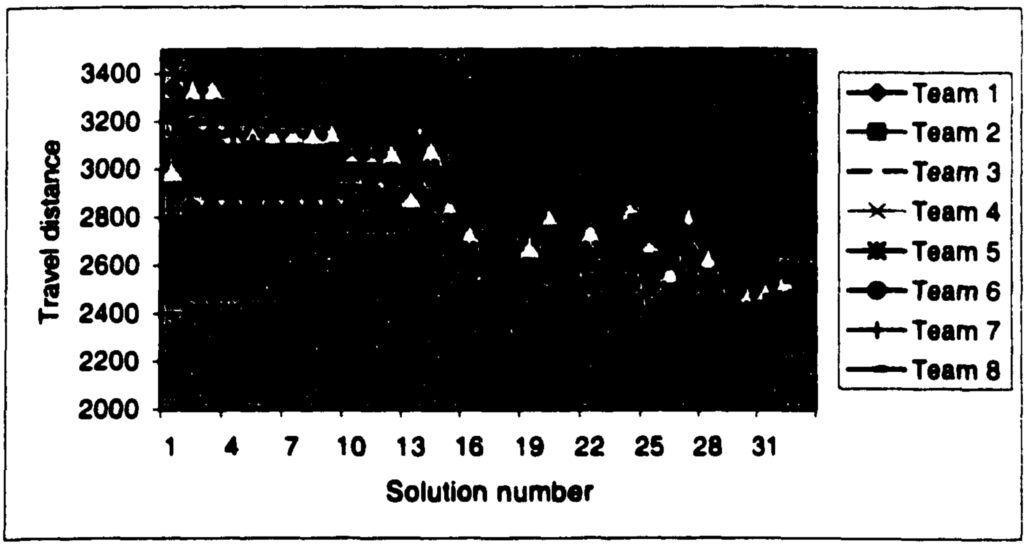

Figure 13. Individual travel distances for example 11 by MDLT

MDLT method is recommended if a league is interested in average travel distance or total travel distance. However, the MVTD model found more balanced travel distance to all teams. In most examples, the MDLT method had lower maximum values in most examples except example 7, 8, and 9. The reason is that the MVTD method resulted in twice-as-long iterations in echelon 3.

Table 21. Comparisons of final results between MVTD and MDLT

Ex	Initial Solutions				Final from MVTD				Final from MDLT			
	Min	Max	Variance	Average	Min	Max	Variance	Average	Min	Max	Variance	Average
1	1260	1840	63566.7	1475.0	1360	1420	633.3	1395.0	1100	1420	23600.0	1190.0
2	156	442	13915.7	284.5	192	293	1988.7	257.0	150	293	4236.3	230.3
3	266	586	23454.3	451.5	366	405	280.3	388.3	266	405	4263.6	363.3
4	1230	1406	8026.9	1309.8	1230	1281	543.6	1252.3	981	1069	1774.7	1036.0
5	130	278	4476.0	191.0	154	189	267.6	164.8	123	189	727.0	155.5
6	1699	2230	37924.0	2032.0	1887	1901	29.8	1890.8	1483	1773	13009.9	1670.7
7	1377	2062	55747.0	1608.8	1414	1438	75.9	1427.7	1191	1445	10565.9	1326.3
8	221	378	3773.9	299.5	225	243	51.2	233.0	150	290	2917.5	224.7
9	379	520	3210.7	436.3	379	386	7.4	381.8	287	389	1237.6	328.0
10	906	1185	11720.8	1049.0	861	880	43.9	873.7	585	779	4634.6	699.2
11	2604	3460	67539.1	3131.0	2932	2951	64.1	2940.1	2409	2787	14918.8	2563.3

Fair game scheduling does not necessarily find the minimum travel distance as we have seen PART I and PART II; however, it finds a solution that balances each team's travel distance and therefore increases the chance to be accepted. The fair game can be extended in the way that includes requests from all the teams so that a fair schedule can be justified as satisfying those requests.

CHAPTER IV. THE SOUTHERN LEAGUE BASEBALL SCHEDULING USING SIMULATED ANNEALING

Introduction

We have so far discussed the general AGS problems to gain insight to the real AGS problem. The problem was further analyzed by 0-1 integer program and was exploited using multi echelon heuristics to search for the optimal solution. In PART IV the multi echelon heuristic is applied to the Southern League baseball scheduling.

The Southern League consists of 10 AA minor league teams in two divisions (see Table 22). Most teams are located in the southeastern area of United States (Florida, Alabama, Tennessee, North Carolina, and South Carolina). A team currently plays 139 games in 152 days with series made up of two to four games. There is an all Star game that is held toward the middle of the second half of the season. The league wants to change the current schedule requirements in the year 2000 so that each team plays 140 games in 152 days, the All Star game is held exactly in the middle of the season, and only two and four games series are allowed. The schedule also allows only 10 days off during the season (composed of 4 days off in the first half and 6 days off in the second half). This scheduling is very tight for all of the teams.

Table 22. Teams in the Southern League

Western Division	Abbr.	Eastern Division	Abbr.
1: West Tennessee (Jackson), TN	WTN	6: Knoxville, TN	KNX
2: Mobile, AL	MOB	7: Greenville, SC	GRN
3: Birmingham, AL	BIR	8: Jacksonville, FL	JAX
4: Huntsville, AL	HNT	9: Orlando, FL	ORL
5: Chattanooga, TN	CHT	10: Raleigh/Durham, NC	CAR

Teams are transported by buses as they travel from one place to the next location. Therefore, minimizing the total traveling distance is a major issue not only to reduce the travel costs but also to reduce players' fatigue.

Problem Formulation

This problem consists of 10 teams, $n = 10$, in 10 different sites. Teams must be at their home site at the beginning and after finishing the season schedule. A team can visit another site only when the homeowner is at the site. We use the same notation as in PART I and PART II.

i, j: teams,

t: game series or time slot,

n: number of teams, $n = 10$,

D_{ij}: distance between team i's home and team j's home,

m: maximum number of consecutive home or road games allowed, $m = 3$,

y^t_{ijk} : game between team i and j at location k at time t where k is either i or j,

G_{it}: game schedule or opponent of team i at time t,

h_{it}: indicates whether team i is at home or away at time t,

Z: the total travel distance completed by all teams.

The first objective is to minimize the total distance traveled by all teams. The objective function is

$$Z = \sum_{i=1}^{10}\sum_{j=1}^{10}\sum_{k=1}^{n}\sum_{l=1}^{n}\sum_{t=1}^{37} y^{t-1}_{ijk}\, y^t_{ijl}\, D_{kl},$$

where

$$y_{ijk}^{t} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ have a game at location } k \text{ at time } t \text{ where } t = 1, 2, \cdots, 2(n-1) \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} y_{ijk}^{t} = 1 \qquad for\ i = 1, 2\cdots, n\ \ and\ t = 1, 2, \cdots, 36$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{t=1}^{36} y_{ijk}^{t} = 36 \qquad for\ i = 1, 2\cdots, n$$

Since game between team i and j can be held at only one place, $y_{iji}^{t} + y_{ijj}^{t} = 1$ for $i \neq j$. Team i can play in only one game series in each time slot, t = 1, ……, 36. In a season, therefore, each team has 36 series of games. During the season a team cannot have a game series against themselves, so $y_{iii}^{t} = 0$ for t = 1, ……, 36. In order to position the team at home at the beginning and end of the season, we define $y_{iii}^{0} = 1$, $y_{iii}^{37} = 1$, $y_{ijk,i\neq j}^{0} = 0$ and $y_{ijk,i\neq j}^{37} = 0$. Also, a back-to-back game series is not allowed between two teams. Therefore,

$$y_{ijk}^{t} + y_{ijk}^{t+1} \leq 1 \text{ for } t = 1, \ldots, 36.$$

$$h_{it} = \begin{cases} 1 & if\ y_{ijk}^{t} = 1\ and\ j = k\ \ for\ t = 1, \cdots, 36 \\ 0 & if\ y_{ijk}^{t} = 1\ and\ i = k\ \ for\ t = 1, \cdots, 36 \end{cases}$$

$$\sum_{t=1}^{36} h_{it} = 18 \qquad i = 1, \cdots, n,$$

the total number of road/home game series for each team is 18 and

$$\sum_{i=1}^{n} h_{it} = 5 \qquad t = 1, \cdots, 36,$$

there are a total of 5 home game series played in each time slot.

$$h_{it} + h_{it+1} + \cdots + h_{it+m-1} + h_{it+m} \geq 1.$$

$$h_{it} + h_{it+1} + \cdots + h_{it+m-1} + h_{it+m} \leq m.$$

The total number of consecutive home or road games is limited to m, $m = 3$.

Each team has 140 games, 70 at home and 70 on the road. The season starts on April 6 and ends on September 4. There is an All Star game break on June 19 and 20. The season is divided into the first half and the second half with respect to games before and after the All Star break. Each team plays 24 games with each of the 4 teams in the division, all four-game series, and 8 games with each of the 5 teams out of division, four-game series in each half season. Also, there are two-game series with the nearest team, one series each in the first half and second half. Two-game series pairs are WTN vs. HNT, MOB vs. BIR, CHT vs. KNX, GRN vs. CAR, and JAX vs. ORL. The number of game series requirements given above adds more constraints. The total number of game series between two teams during a season varies depending on the teams:

$$\sum_{k=1}^{n}\sum_{t=1}^{36} y_{ijk}^{t} = \begin{cases} 6 & i \neq j,\ i \text{ and } j \text{ are in the same division but not closest each other,} \\ 8 & i \neq j,\ i \text{ and } j \text{ are closest teams in the same division,} \\ 4 & i \text{ is CHT and } j \text{ is KNX or vice versa,} \\ 2 & i \neq j,\ i \text{ and } j \text{ are in the different division and not in closest team pair.} \end{cases}$$

Maximizing the Southern League requirements

The second objective is to minimize the penalty cost that accrued when the schedule violates the requirements provided by the league. The league requirements and team preferences are given below. Since some requirements are more important than other requirements, we assign different penalty costs with regard to the importance of the requirements.

The following are the requirements that the Southern League requested. No team may travel more than 500 miles without a scheduled day off; penalty cost of 100 is given each occurrence of violation. All teams should have at least one scheduled day off in each 30 day period, penalty cost 10 for each occurrence of violation. The total number of home weekend dates, Friday and Saturday, must balance to within plus or minus one date of the average, 22, in the season, 2 penalty costs for each occurrence of violation. Fair distribution of home weekend games per month. Travel distance of all teams in the season should not exceed 105,000 miles in total, penalty cost of 10,000. The final schedule should have the best possible balance of home games per month. Every team prefers at home on the 4th of July; however, only CAR must be scheduled at home on July 4 since the team had an away game in the past two years, 1000 penalty cost if violated. Teams that played at home on one side of the All Star break should play on the road on the other side, and vice versa; 100 penalty costs for each occurrence of a violation. No series starts on Sunday, if possible. No more than 12 consecutive home games are allowed. Back-to-back series between two teams are not allowed; 10,000 penalty costs for a violation.

The following constraints must be satisfied and, therefore, used in determining the feasibility of the current game schedule that was generated by swapping two columns in echelon 2. KNX must be on the road for the first 12 days due to ballpark construction and will accept 20 weekend days in exchange for the first 16 days on the road; 10,000 penalty costs for violation. CAR must be scheduled on the road for the first 8 days due to ballpark construction, 10,000 penalty costs for violation. ORL will play on the road on April 6-9, May 19-22, and June 21 due to pre-existing events in its ballpark, penalty costs of 10,000 is given on each occurrence of violation.

Individual team preferences for the year 2000 (penalty costs: 2)

- WTN: Likes to open on the road, but will open at home because it had road openers in the past two years. Wants more home games in May and fewer games in August.
- MOB: Likes to open at home against BIR for rivalry purpose.
- BIR: Wants their home opener to be on a weekend and their road games on the third May weekend. Doesn't want a Monday game at home.
- HNT: Likes to open on the road in Florida and wants to be at home the first week of May for the annual sold-out promotion.
- CHT: Would like to open on the road.
- KNX: Wants to close on the road and doesn't like direct trips to or from Florida.
- GRN: Doesn't want to play at home on Easter Sunday. Likes more weekend home games in June and July and fewer games in April and May.
- JAX: Likes to open at home and likes home games on the weekend in April and May. Doesn't want to play at home in August and September, especially on weekends, because of conflicted schedule with the National Football League (NFL) Jacksonville Jaguars.
- CAR: Wants fewer home games in August.

Algorithm

Generating a round robin schedule

Before generating the initial game schedule we need to generate a round robin schedule as in Table 23. Since each division has an odd number of teams (5), the initial algorithm given in PART I and PART II cannot be available. Yet, if we assume $n = 6$, we can easily generate a *round robin schedule* for 6 teams using the initial schedule algorithm given in PART I and PART II. We can then duplicate all the rows in the 6 teams' *round robin schedule* and add 5 to all slots of the duplicated schedule so that we have 6 teams round robin schedule in two divisions (see Table 23), where team 6 in division I and division II are different teams. Since the Southern League has only 10 teams, we remove the last rows from both division I and division II schedule. Also, we delete 6 from each slot in division I and 11 from each slot in division II so that the schedule consists of 5 teams in each division which gives every team a time slot without a game. We can now make a pair with those teams without a game in each slot, (4, 9) in slot 1, (5, 10) in slot 2, (3, 8) in slot 3, (2, 7) in slot 4, and (1, 6) in slot 5, as in Table 23 to generate *10 teams round robin schedule.*

Table 23. Generating round robin game schedule

		Time (t)						*Time (t)*				
Division	*Team*	1	2	3	4	5	*Team*	1	2	3	4	5
	1	2	3	4	5	***6***	1	2	3	4	5	***6***
	2	1	4	5	***6***	3	2	1	4	5	***7***	3
	3	5	1	***6***	4	2	3	5	1	***8***	4	2
I	4	***6***	2	1	3	5	4	***9***	2	1	3	5
	5	3	***6***	2	1	4	5	3	***10***	2	1	4
	6	***4***	***5***	***3***	***2***	***1***						
	6	7	8	9	10	***11***	6	7	8	9	10	***1***
	7	6	9	10	***11***	8	7	6	9	10	***2***	8
	8	10	6	***11***	9	7	8	10	6	***3***	9	7
II	9	***11***	7	6	8	10	9	***4***	7	6	8	10
	10	8	***11***	7	6	9	10	8	***5***	7	6	9
	11	***9***	***10***	***8***	***7***	***6***						
6 teams and two divisions single round robin game schedule							10 teams two divisions round robin game schedule					

Generating an initial game schedule (G_{it})

An initial game schedule can be generated based on *10 teams two division round robin schedule*. The algorithm is as follows (also see Table 24):

Step 1. Copy columns 1 to 5 to columns 6 to 10 and 11 to 15.

Step 2. In division II, rotate columns 6 to 10 clockwise such that column 7 moves to column 6, column 8 moves to column 7, 9 moves to column 8, 10 moves to column 9, and 6 moves to column 10.

Step 3. In division I, rotate columns 11 to 15 such as in Step 2.

Step 4. From column 6 to 15, delete teams that appeared in the different division (team 6 to 10 in division I and team 1 to 5 in division II) so that we have two empty slots in every column 6 to 15.

Step 5. Assign a game between the two teams that we deleted in Step 4 in each column 6 to 15.

Table 24. Initial game schedule (G_{it})

	Time (t)																	
Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2	3	4	5	6	2	3	4	5	***9***	3	4	5	***7***	2	8	10	4
2	1	4	5	7	3	1	4	5	***6***	3	4	5	***8***	3	1	10	9	3
3	5	1	8	4	2	5	1	***7***	4	2	1	***10***	4	2	5	9	6	2
4	9	2	1	3	5	***10***	2	1	3	5	2	1	3	5	***6***	7	8	1
5	3	10	2	1	4	3	***8***	2	1	4	***9***	2	1	4	3	6	7	6
6	7	8	9	10	1	8	9	10	***2***	7	7	8	9	10	***4***	5	3	5
7	6	9	10	2	8	9	10	***3***	8	6	6	9	10	***1***	8	4	5	10
8	10	6	3	9	7	6	***5***	9	7	10	10	6	***2***	9	7	1	4	9
9	4	7	6	8	10	7	6	8	10	***1***	***5***	7	6	8	10	3	2	8
10	8	5	7	6	9	***4***	7	6	9	8	8	***3***	7	6	9	2	1	7

Step 6. In each row 1 to 5 find two teams in division II that do not have a game and assign them to column 16 and 17. Then schedule the corresponding teams in row 6 to 10. Step 6 is the same as finding a single route from team 1 to team 10 (see Appendix B for details).

Step 7. Column 18 is filled with the games between the nearest teams.

A game schedule (G_{it}) for a season is obtained by duplicating column 1 to 18 so that the season schedule has 36 time slots.

Generating the initial home and away game schedule (h_{it})

After a game schedule (G_{it}) for a season is generated, we can initialize the home and away schedule (h_{it}). The method is very similar to the algorithm given PART I.

Step 1. All teams stay home at $t = 0$ and 37 (i.e. all $h_{it} = 0$ for $i = 1, 2, \cdots, n$).

Step 2. Choose team $i = 1$ to 10.

Step 2-1. Set $t = 1$.

Step 2-2. If h_{it} is not 0 or 1, assign 1 (Make the first meeting a road game).

Step 2-2-1. For $j = t + 1$ to 36, if $G_{it} = G_{ij}$, then $h_{ij} = 0$ and set $s = j$ (Make the second meeting a home game).

Step 2-2-2. For $k = i + 1$ to *10*, if $G_{kt} = i$, then $h_{kt} = 0$ and $h_{ks} = 1$ (Set a complementary home/road schedule for team i's competitor).

Step 2-3. If $t < 36$, $t = t + 1$ and go to Step 2-2. Otherwise, go to Step 2.

Step 3. If there is not more than m consecutive home or road games for any team in the schedule, then stop. Otherwise, let h_{it} be the $(m+1)$th consecutive home or road game. Change h_{it} to its complementary number (i.e. if $h_{it} = 0$, the new $h_{it} = 1$, and vise versa).

Step 3-1. For j = 1 to 36, if $G_{it} = G_{ij}$ and $j \neq t$, then change h_{ij} to its complementary number and set $s = j$ (Change the location of the other meeting of the two teams).

Step 3-2. For k = 1 to 10, if $G_{kt} = i$ and $i \neq k$, then change h_{kt} and h_{ks} to their complementary numbers and go to Step 3 (Change the home/road schedule for team i's competitor).

Feasible home and away game schedule

To make the home and away schedule feasible, which will satisfy all the constraints with penalty of more than 10,000, the initial home and away schedule (h_{it}) needs to be adjusted. The algorithm forces an assignment of four game series to the first 3 time slots to satisfy the constraints that must be on the road. For example, KNX has to be on the road for the first 3 time slots ($h_{6,1} = h_{6,2} = h_{6,3} = 0$) while CAR must be on the road for the first two time slots ($h_{10,1} = h_{10,2} = 0$). And ORL is not allowed at home on time slot 1, 12, and 19 ($h_{9,1} = h_{9,12} = h_{9,19} = 0$) due to the preexisting events. A feasible home and away game schedule is obtained as explained below:

Step 1. If all teams and time slots mentioned above have road games, then go to Step 2. Otherwise, let h_{it} be the home game that violates the constraints.

Step 1-1. Change $h_{it} = 1$. For j = 1 to 36, if $G_{it} = G_{ij}$ and $j \neq t$, then change $h_{ij} = 0$. (Change location of the other meeting of the two teams).

Step 1-2. For k = 1 to 10, if $G_{kt} = i$ and $i \neq k$, then change $h_{kt} = 0$ and $h_{ks} = 1$. (Change the home/road schedule for team i's competitor).

Step 2. Check for consecutive home or road games using Step 3 of *generating an initial home and away game schedule*.

Multi echelon heuristic

The problem is very complicated and may not be practical using traditional operations research tools. In addition, some constraints cannot be formulated. This leads us to develop a multi echelon heuristic that can solve the problem efficiently and generate a feasible solution easily from a given infeasible solution. It turns out that the problem becomes infeasible almost every time the algorithm is permuted in every echelon because of the imposed constraints. It is, therefore, important to find a closest feasible solution from the current infeasible solution.

We implemented three echelon heuristics. Echelon 1 (see Figure 14) swaps two rows of game schedule (G_{it}) and if the new game schedule is feasible, it is sent to the echelon 2. In

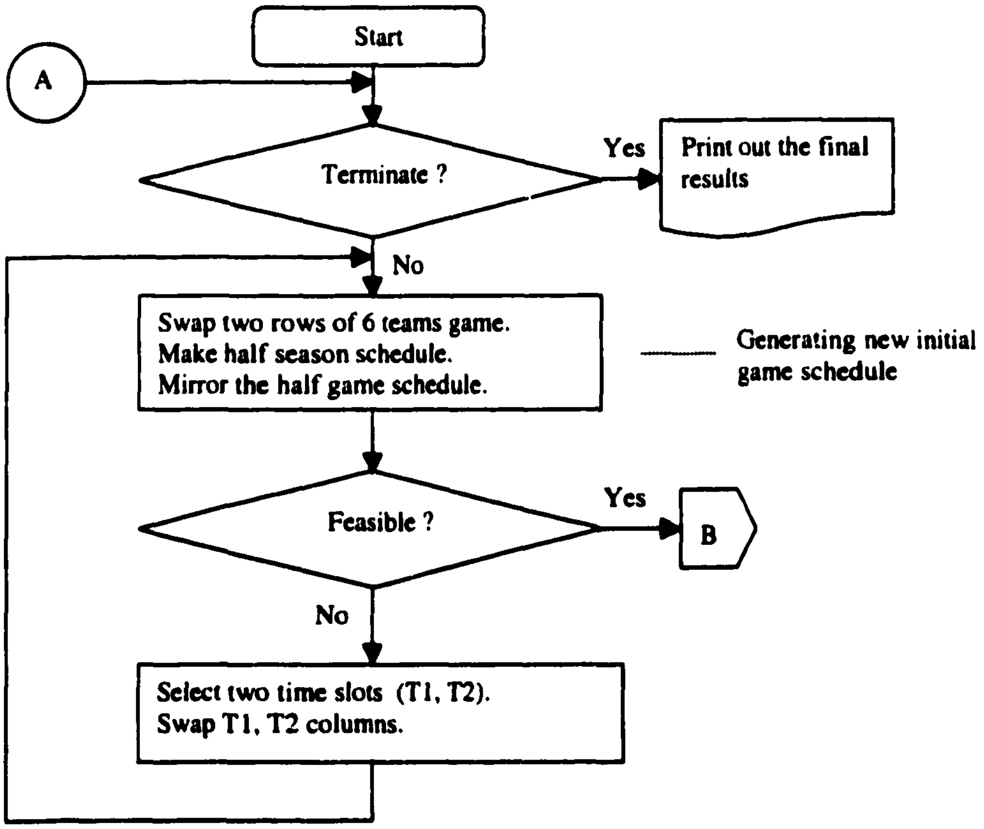

Figure 14. The Southern League scheduling echelon 1.

echelon 1 a game schedule is considered feasible if it does not have back-to-back series between two teams. Echelon 1 permutes $n\times100$ times.

Two columns in the game schedule are swapped until we obtain a feasible solution in Echelon 2 (see Figure 15). The feasible solution in echelon 2 is defined such that the two teams that must be on the road in the same time slots, for example, KNX and CAR at time slot 1 and 2, cannot compete with each other. Also, back-to-back games between two teams should not exist. Based on the new game schedule (G_{it}) that satisfies the feasibility in echelon 2, the home and away game schedule (h_{it}) is initialized as given above and sent to echelon 3 (see Figure 16) for further permutations. Echelon 2 permutes $n\times38$.

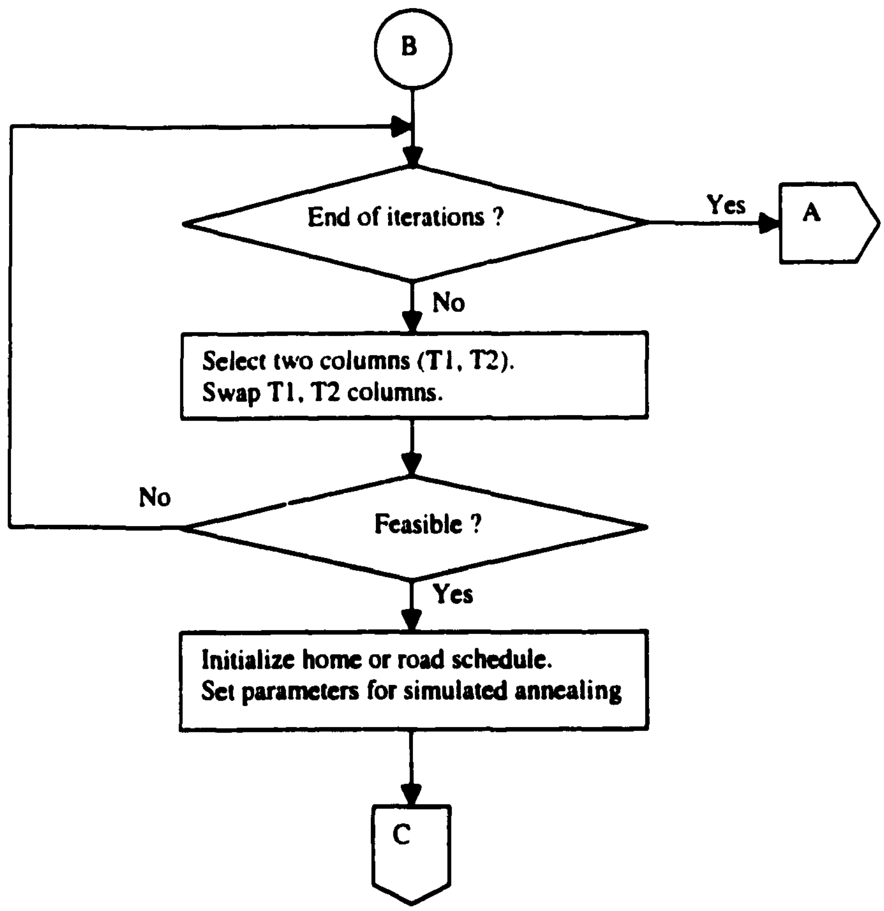

Figure 15. The Southern League scheduling echelon 2.

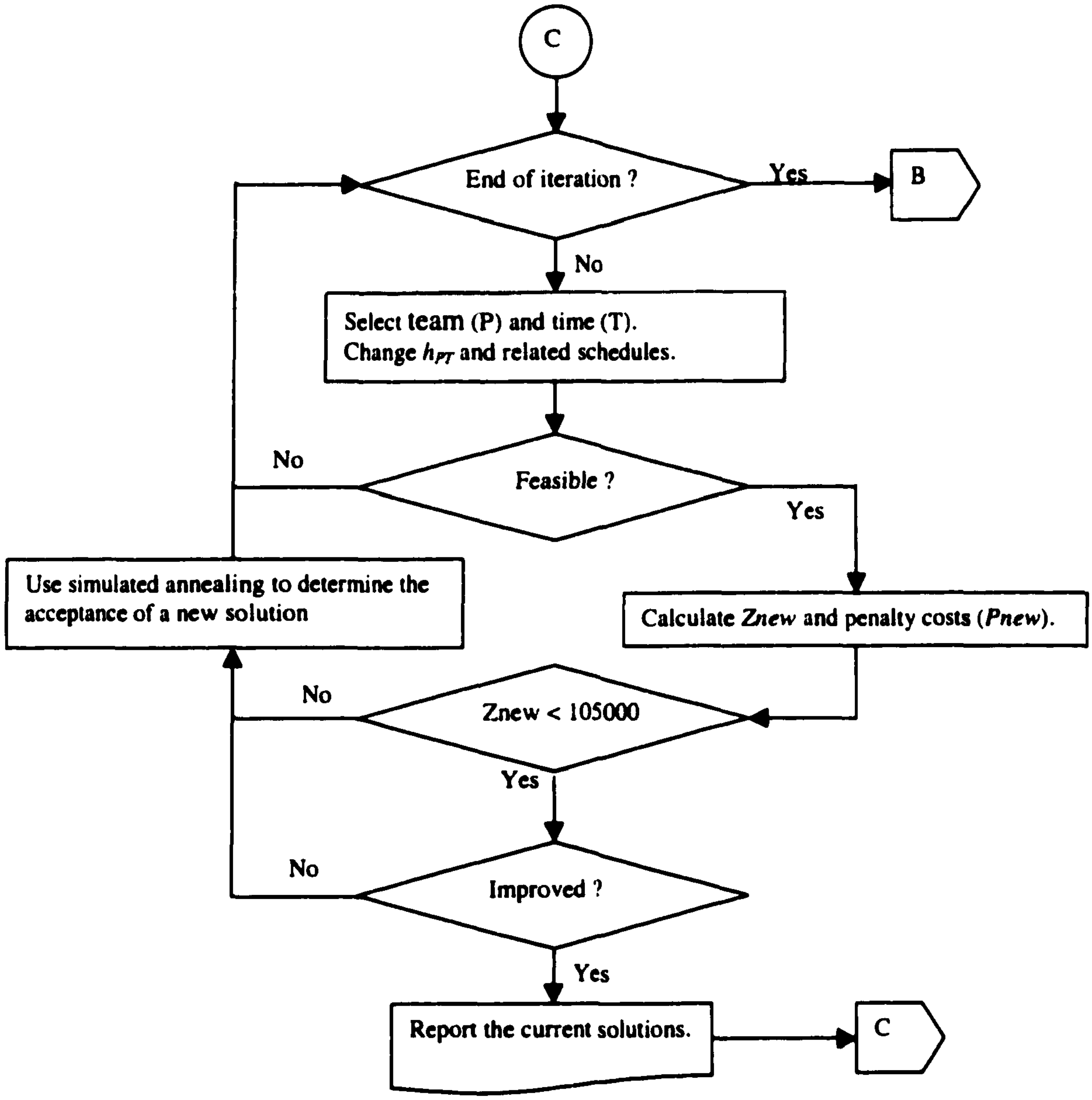

Figure 16. The Southern League scheduling echelon 3.

The simulated annealing (SA) algorithm is employed in echelon 3 to determine the acceptance of the current solution as a new solution because SA has the ability to overcome local optimum solution by using negative exponential probability distribution. The algorithm print out solutions with strictly less than 105,000 in terms of total travel distance to satisfy the league requirement on the maximum allowable travel distance. Echelon 3 permutes $n\times1000$ times also uses 0.9 as the annealing factor.

Generation of a season schedule from the acceptable solutions

The output report in echelon 3 consists of total travel distance, penalty costs, and approximate season schedule (ASCH). The ASCH is obtained from a game schedule (G_{it}) and a home and away schedule (H_{it}). Table 25 shows the beginning of the second half of a season. A normal time slot in G_{it} or H_{it} is extended to 4 days in ASCH and the slots representing the prescheduled game between nearest teams are extended to two days in ASCH. The league asks for a day off whenever a team travels more than 500 miles in a day; however, the empirical results show that a season schedule cannot be accomplished when we allow for a day off for every travel of more than 500 miles. To do that, we need more than 150 days, but the total available days by the Southern League is 150 days--74 days in the first half of the season and 76 days in the second half, except for the two days for the All Star Game.

Table 25. A part of game schedule (G_{it}) and home and away schedule (H_{it})

	Time				Time			
Teams	19	20	21	22	19	20	21	22
1	2	4	7	5	1	1	0	0
2	1	5	4	3	0	1	0	0
3	4	8	5	2	1	0	1	1
4	3	1	2	6	0	0	1	1
5	10	2	3	1	1	0	0	1
6	7	10	9	4	0	1	1	0
7	6	9	1	8	1	1	1	0
8	9	3	10	7	0	1	1	1
9	8	7	6	10	1	0	0	0
10	5	6	8	9	0	0	0	1

Only four off days in the first half of the schedule and six off days in the second half are available to all teams to finish 140 games in a season. Hence, the algorithm selects four longest travels in the first half and six longest travels in the second half of the season. ASCH is then made up of all teams having days off on those ten days (see left half of Table 26). The ASCH is refined further by an expert to minimize the penalty costs as seen in right half of Table 26. The penalty costs of violating a day off for over 500 miles travel in a day, a cost of 500 per instance, are relatively large. Therefore, the reduction of the penalty costs for increasing days off for over 500 miles travel in a day is far greater than the augmentation of the penalty costs caused by violating other constraints after the refinement step. ASCH thus provides the upper bound of the penalty costs for a given G_{it} and H_{it}. As seen in Table 26, the

Table 26. Approximated and refined season schedule

	Teams (ASCH)										Teams (Refined)									
Day	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
77	2	2	4	4	10	6	6	8	8	10	2	2	4	4	10	6	6	8	8	10
78	2	2	4	4	10	6	6	8	8	10	2	2	4	4	10	6	6	8	8	10
79	2	2	4	4	10	6	6	8	8	10	2	2	4	4	10	6	6	8	8	10
80	2	2	4	4	10	6	6	8	8	10	2	2	4	4	10	6	6	8	8	10
81	0	0	0	0	0	0	0	0	0	0	0	5	3	0	5	10	0	3	0	10
82	4	5	3	4	5	10	9	3	9	10	4	5	3	4	5	10	9	3	9	10
83	4	5	3	4	5	10	9	3	9	10	4	5	3	4	5	10	9	3	9	10
84	4	5	3	4	5	10	9	3	9	10	4	5	3	4	5	10	9	3	9	10
85	4	5	3	4	5	10	9	3	9	10	4	0	0	4	0	0	9	0	9	0
86	1	2	5	2	5	9	1	10	9	10	0	2	5	2	5	9	0	10	9	10
87	1	2	5	2	5	9	1	10	9	10	1	2	5	2	5	9	1	10	9	10
88	1	2	5	2	5	9	1	10	9	10	1	2	5	2	5	9	1	10	9	10
89	1	2	5	2	5	9	1	10	9	10	1	2	5	2	5	9	1	10	9	10
90	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
91	1	2	2	6	1	6	7	7	9	9	1	2	2	6	1	6	7	7	9	9

refined season schedule reduces the penalty costs by 1500. In case there are conflicts because more than two teams travel over 500 miles in a day and we have to choose only one, we assign a day off to the team traveling the most miles. Also note that the total number of days off in a day must be an even number when making the refinement schedule since we have a total of 10 teams, an even number of teams. For example, on day 81 four teams have the day off and two teams have the day off on day 86.

Results

Three different objective function values--total travel distance, total penalty costs, and combination of both travel distance and penalty costs--are investigated along with simulated annealing (SA) in echelon 3. The travel distances between two teams that the Southern League provided are symmetric. The distances between the teams are given in Table 27.

Table 27. Distance between each team

Teams	WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
WTN	0	379	223	172	260	316	479	720	816	687
MOB	379	0	262	361	405	517	479	470	498	745
BIR	223	262	0	99	144	258	290	468	563	535
HNT	172	361	99	0	107	221	311	565	660	586
CHT	260	405	144	107	0	114	243	468	563	479
KNX	316	517	258	221	114	0	167	546	663	370
GRN	479	479	290	311	243	167	0	386	526	263
JAX	720	470	468	565	468	546	386	0	140	470
ORL	816	498	563	660	563	663	526	140	0	610
CAR	687	745	535	586	479	370	263	470	610	0

Minimizing travel distance

At first we consider minimizing the total travel distance as an objective function for SA in echelon 3 (Table 28). The result shows that penalty costs are not necessarily decreased as the total travel distance reduces. The minimum total travel distance is 6,642 miles less than the maximum limit (105,000 miles) that the Southern League required and 5982 miles shorter than the actual schedule used in the year 1999 (104,340 miles). The results also show that the penalty costs from the refined schedule is always lower than ASCH.

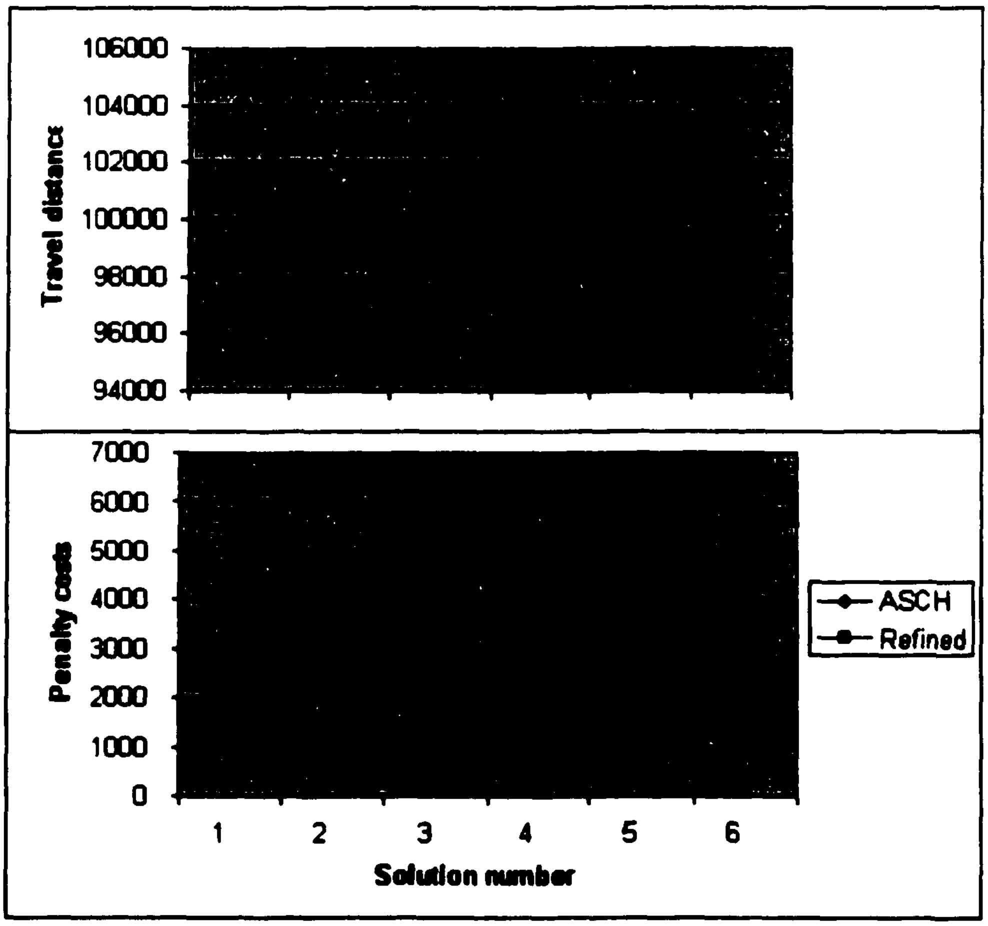

Figure 17. Result from minimizing travel distance

Table 28. Results from minimizing travel distance

Number	Total Travel	Penalty Costs (ASCH)	Penalty Cost (Refined)
1	104127	5400	4687
2	101914	5850	3280
3	99863	4824	3677
4	99562	4644	4366
5	98572	4408	3354
6	98358	5606	3565

Minimizing penalty cost

Table 29 shows the results from minimizing the penalty costs as an objective function for SA in echelon 3. According to the results, we cannot find a strong relationship between penalty cost and travel distance (see also Figure 18). The lowest penalty costs at the ASCH is 3398 and the refined schedule has 2460. It also reveals that the lower penalty cost in the ASCH schedule dose not guarantee a lower penalty cost from the refined schedule.

Table 29. Results from minimizing penalty costs

Number	Total Travel	Penalty Costs (ASCH)	Penalty Cost (Refined)
1	103253	4566	3260
2	103643	4564	3299
3	104754	4468	3382
4	104726	4340	3397
5	102263	4286	3979
6	104539	4178	3659
7	103706	4174	3870
8	102699	4065	3914
9	102820	3814	3772
10	104031	3798	3383
11	102626	3560	3067
12	104066	3554	2863
13	103800	3488	2668
14	102004	3474	3472
15	101625	3456	2460
16	103664	3398	3390

Minimizing the combination of travel distance and penalty costs

We also investigated minimizing the combination of total travel distance and penalty costs as an objective function (Table 30). For output, we put the maximum limit for the summation of penalty costs and travel distance strictly under 105,000. The final results show that the minimum total travel distance saves 5,455 miles compared to the schedule adopted in 1999. Solution number 7 has the most saving (1,004) in terms of the penalty costs after refining the schedule as 2,780.

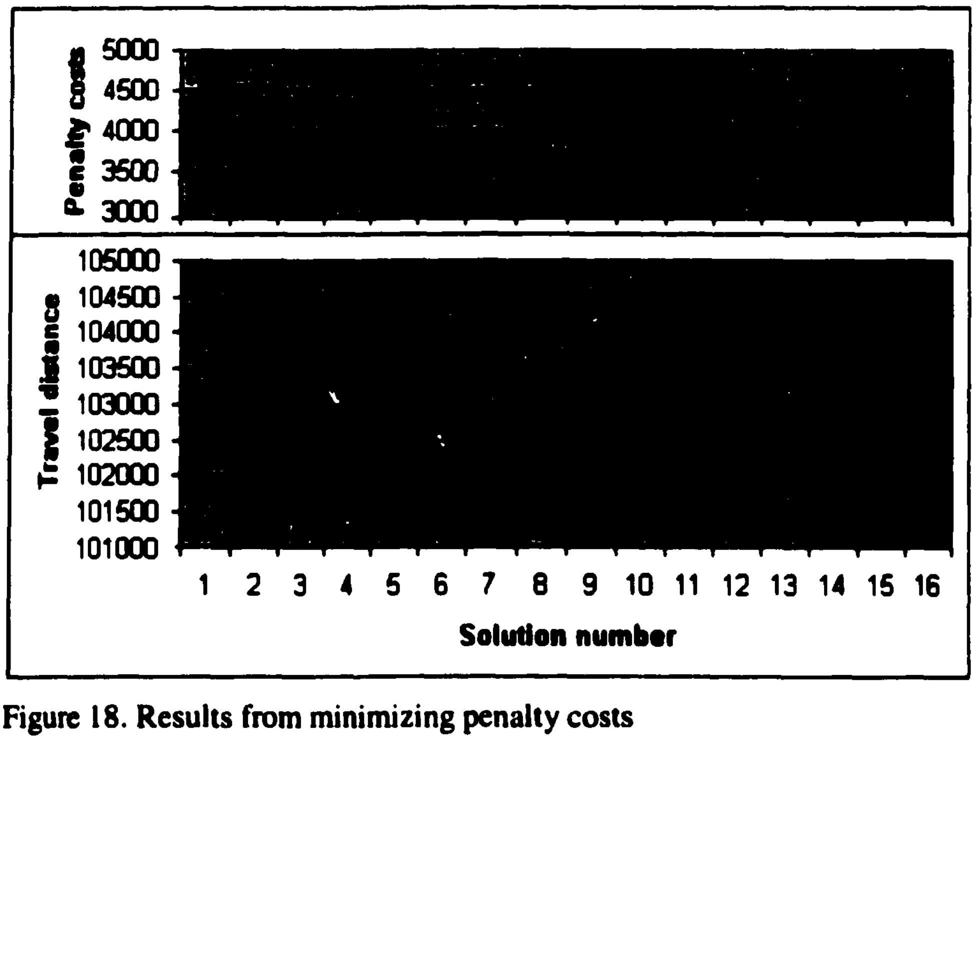

Figure 18. Results from minimizing penalty costs

Table 30. Results from minimizing combination of travel distance and penalty costs

Number	Total Travel	Penalty Costs (ASCH)	Sum	Penalty Cost (Refined)
1	100114	4592	104706	4384
2	100238	4262	104500	3566
3	99804	3760	103564	3758
4	99623	3760	103383	3466
5	99099	3762	102861	3456
6	98927	3880	102807	2982
7	98927	3784	102711	2780
8	98885	3682	102567	3370

Conclusions

In this research, we have introduced the Southern League Baseball scheduling and shown a systematical approach to the problem. We implemented three different objective functions: minimizing the total travel distance and penalty costs, and minimizing a combination of both travel distance and penalty costs. We used three echelons that include simulated annealing (SA) algorithm in the third echelon to search for good solutions. We showed substantial improvement over the league guideline for travel distance. Also, the refinement step is very efficient for reducing the penalty costs.

The Southern League was very interested in the solutions of the research. We sent multiple schedules to the Southern League. None of the schedules, however, was accepted for the year 2000 because the penalty costs we assigned to each constraint were not exactly reflected in the Southern League's intentions. For example, when determining a good schedule, the owners of the Southern League teams mostly concern about their teams' home games on weekends to increase revenue as well as to attract more fans. However, we imposed only 2 points for violation of 22 home games on weekends so that the influence of the weekend home games to the total penalty costs is extremely small. Less affect on the total penalty cost led less influence on simulated annealing when we used the penalty cost to determine the acceptance of the new solution over the current best solution. Consequently, in every developing stage the communication with the Southern League to obtain more detail information about importance of each constraint to the teams and players should not be ignored.

The research will be applied further to the other athletic games, for example, the National Basketball Association (NBA), Major League Baseball (MLB), the National Hockey League (NHL), the Korean Baseball Organization (KBO), and the Japanese Baseball League, etc. We can also use combination of an integer programming and multi echelon algorithm.

CHAPTER V. DETERMINING DIFFERENT TYPES OF ATHLETIC GAME SCHEDULING

Introduction

The athletic game scheduling (AGS) problem is deterministic and NP hard. There is no exact algorithm that solves all the real world AGS problems. In PART I, II, and III we have discussed multi echelon heuristics that are applicable to any kind of AGS with minor changes and we solved the Southern League baseball scheduling problem as an example in PART IV.

The development of the fast computer processor in modern technology has brought new research areas to operations research scientists. The problem of AGS is one of the areas that operations researchers have been investigating recently. Researchers are attempting to solve this problem by using computers. Traditionally, the AGS problems have been solved by hand. That process was tedious and the final solution was not economical. Dozens of research results have been published since 1977. However, most of the heuristic methods used in the previous research are problem specific and cannot solve every AGS problem.

No researchers have been able to provide a general AGS form. Therefore, we analyze the different types of AGS problems in general depending on the specific characteristics of a given AGS problem to help researchers who are involved in AGS. Once the characteristics of the given AGS problem are determined, one should easily build the initial schedule and find different solutions. To help build the initial schedule a simple scheduling algorithm for a single round robin AGS problem is also provided. The single round robin schedule-demonstrated using table-will help researchers to understand the AGS problem. The extended

AGS problems, such as the double round robin problem, can be easily generated from the single round robin AGS schedule.

Types of Ahtletic Game Scheduling Problems

Characteristics of AGS problems

All AGS problems have several distinct characteristics. First, we can characterize AGS problems depending on the number of round robin schedules: single round robin, double round robin, more than triple round robin games, and mixed round robin schedule. Round robin game means that a team meets every other team in the league. Therefore, double or higher round robin games where each team meets all the other teams once before meeting again can be easily obtained by duplicating the single round robin schedule. A mixed round robin schedule does not require that each team should meet all the other teams once before meeting again.

The AGS problems can also be divided by the availability of fixed time slots, i.e. fixed-time slot and nonfixed time slots. Time slots can be explained as the separation between game days and nongame days. If all teams in a league have games or byes on the same day and take a break during the same periods (days), the league has fixed time slots. In the National Football League (NFL) in the U. S., for example, all teams have games or byes once a week such that a single time slot represents a week. Most of the AGS problems have fixed time slots, including the National Basketball Association (NBA), College basketball and football games, NFL, etc. A nonfixed time slot problem can be found in some baseball scheduling problems. For example, baseball's Southern League in PART IV has a schedule whose series cannot be separated into exact time slots.

Finally, the number of leagues and the number of divisions in a league including inter and intra divisional games are important factors for determining the characteristics of a AGS problem. Even though there are many divisions in a league, if no inter divisional game exists, the schedule is the same as solving a single division problem.

Representation of AGS problem

According to the characteristics of a game schedule, AGS problems can be described as ***A/B/C/D/E*** where,

A is the number of divisions regardless of the number of leagues. For example, Major League Baseball has two leagues, the American and the National League with three divisions--Eastern, Central, and Western division--in each league, so it has six divisions (***A*** = 6).

B is whether all divisions in a league have only intra divisional games or not. ***Intra*** represents intra divisional games only and ***inter*** represents both inter and intra divisional games.

C shows the number of round robin schedules in a season. For example, single round robin is ***1***, double round robin is ***2***, ***M*** represents the mixed round robin, and ***S*** means that special promotion games exist for all teams that are predetermined before the start of scheduling. The Southern League in PART IV, for instance, has predetermined games between the closest teams twice a season that can be represented as ***S-2***. More than two elements can be used in ***C*** when inter divisional games are allowed. For example, ***/in-2, out-1, M/*** can be interpreted as a schedule where each team has 2 inter divisional games with teams in different divisions and

1 intra divisional games with every other team in the same division. Also two teams can meet twice before meeting another team once in any period.

D indicates whether the league consists of fixed time slots (***F***) or not (***N***).

E is the number of teams in a division. If more than two divisions have a different number of teams, then choose the largest number.

Therefore, a league whose representation is ***2/inter/1/F/6*** can be interpreted as two divisions including inter divisional games and a season consists of a single round robin with fixed time slots. A division has at most 6 teams.

Scheduling Algorithm for *1/Intra/1/F/n*

The ***1/intra/1/F/n*** AGS problem is fundamental for all the other extended AGS problems that include fixed time slots. We first explain the method to schedule ***1/intra /1/F/n***. The method is provided in PART I and PART II as a half-season schedule. We first need to generate a game schedule as follows:

Step 1. At time $t = 0$, all teams begin at home.

Step 2. Generate the first team's schedule by simply scheduling 1 versus 2 at $t = 1$, then 1 versus 3 at $t = 2$ etc. to 1 versus n at $t = n-1$ (see Table 31 for example). Schedule a game with team 1 on each of the other team's schedule corresponding to the initial half-season schedule of team 1.

Step 3. Schedule for next team i, $i = 2$ to nth team.

3-1. From $t = 1$ to $n-1$, assign any team k who has not yet been scheduled to play at time t.

3-2. Place team *i* on team *k*'s schedule at time *t*.

Table 31. Initial game schedule with corresponding home/away schedule

Team (*i*)	Time (*t*)								
	0	1	2	3	0	1	2	3	
1	1	2	3	4	0	1	1	1	0: Home
2	2	1	4	3	0	0	1	1	1: Road
3	3	4	1	2	0	1	0	0	
4	4	3	2	1	0	0	0	0	

After obtaining the game schedule, the home and away schedule will be generated (see Table 31).

Step 1. All teams stay home at $t = 0$ and n-1. Home and away games are represented by 0 and 1 relatively.

Step 2. For team $i = 1$ to n.

Step 2-1. Set $t = 1$,

Step 2-2. If slot (i, t) is not 0 or 1, assign 1 (Make a road game).

And find a competing team against team *i* at time *t* in the game schedule. Let the team be *k*. Set the (k, t) with complementary number (Set complementary home/away schedule for team *i*'s competitor).

Step 2-3. If $t < n-1$, $t = t+1$ and go to Step 2-2. Otherwise, go to Step 2.

Step 3. If consecutive home/road games do not exist, stop. Otherwise, replace the second game of consecutive home/road games with complementary schedule. Also do the same for the competing team's schedule.

To find a feasible solution for the given constraints a three-echelon algorithm is used. In echelon 1 two rows of a game schedule are swapped and the new game schedule is sent to echelon two. Two columns in the game schedule received from echelon 1 are swapped in echelon two. For the new game schedule from echelon 2, the home and away schedule is generated and permutated for different home/away schedule in echelon 3. The three-echelon procedure is continued until a feasible solution is observed.

The extended AGS problem can be easily obtained from ***1/intra/1/F/n***. For instance, the ***1/intra/k/F/n*** AGS problem is obtained by duplicating ***1/intra/1/F/n*** *k* times. A schedule for a league with odd number of teams can be obtained from schedule of even number of teams obtained after adding a dummy team to the odd number of teams. Let a league have *n* teams and *n* is odd number. At first, we generate schedule for $n + 1$ teams, even number of teams. Then delete $(n+1)^{th}$ row of the schedule. Finally, delete elements that are equal to (n+1) from all time slots of game schedule. Table 32 shows how we obtain a round robin schedule for 3 teams, ***1/intra/1/F/3,*** obtained from the 4 teams' game schedule in Table 31.

Table 32. Game schedule for odd number of teams

Team (*i*)	Time (*t*)			
	0	1	2	3
1	1	2	3	Bye
2	2	1	Bye	3
3	3	Bye	1	2
4	Removed			

Two divisions with an inter divisional game schedule ***2/inter/out-1, in-2, M/F/6,*** for example, can also be derived from single round schedule, ***1/intra/1/F/6***. First, we generate a schedule of ***1/intra/1/F/6*** for division I and duplicate ***1/intra/1/F/6*** for division II. Then we add 6 to all the elements in division II. Thus, we have single round robin schedule for division I and division II. Then, the remaining game schedule is the same as ***1/intra/1/F/12*** and can be easily solved by using single round robin schedule as previously described.

Solving an AGS problem including nonfixed time slots that can be found in many baseball leagues is one of the most difficult AGS problems. Nonfixed time slot problem often appears if a team is required to take off a day when traveling a long distance. For example, in Major League Baseball (MLB) some teams have a day off on Monday while other teams have games such that we cannot separate game days and nongame days. However, we still can apply the multi echelon algorithm in special case, where all series consist of the same number of games or only few series have a different number of games (as an example see the Southern League schedule in PART IV). The Southern League scheduling problem is solved regardless of the day off constraints so that the problem now has fixed time slots. Once we have a problem with fixed time slots, the game schedule and home/away schedule are easily generated by using the method previously provided with the multi echelon algorithm being applied to generate different schedules. Finally, we add the day-off constraints to the final solutions from the Southern League scheduling problem. No research has been published on an AGS problem with a nonfixed time slots.

Analyzing Previous Research

Nemhauser and Trick (1998) scheduled the Atlantic Coast Conference (ACC) basketball problem. The problem can be represented as ***1/intra/2/F/11***, which is 1 division or single league including 11 teams for double round robin schedule with fixed time slots. Therefore, the problem can be modeled as a 12 team schedule, then eliminate row 12 and team 12 from all time slots. A double round robin game schedule can be accomplished duplicating the single round robin schedule. The Big Eight and the Southeastern (SEC) basketball scheduling by Ball and Webster (1977) and SEC basketball schedule by Campbell and Chen (1976) are also double round robin. Since both conferences had an even number of teams, the problem can be characterized as ***1/intra/2/F/n***, where ***n*** is even. The solution procedure is the same as the ACC basketball scheduling problem explained above, except deleting the last row is not necessary.

The National Basketball Association (NBA) schedule was done by Bean and Birge (1980). The problem has 4 divisions that have a maximum of 6 teams in a division. The research does not reveal the exact number of games between teams. Nonetheless, we can predict the requirement from the current NBA game format that every team has x (assumed) intra divisional games and y (assumed) inter divisional games with every other team, and two teams can meet twice before meeting other team once. Hence, the problem can be represented as ***4/inter/in-x, out-y, M/F/6***. Similarly, the National Hockey League (NHL) schedule that was done by Ferland and Fleurent (1991) has ***4/inter/in-x, out-y, M/F/6***.

Armstrong and Willis's (1993) World Cup cricket schedule has 9 teams. Each team had only one game with every other team. Therefore, the solution can be represented as ***1/intra/1/F/9***. The England country cricket schedule that was done by Wright (1994) is the same as the World Cup cricket schedule created by by Armstrong and Willis. The England cricket league has 18 teams. Therefore, it has a format of ***1/intra /1/F/18***. The Australian Cricket League schedule done by Willis and Tereill (1994) includes 6 teams and consists of one, two, and four game series so that the problem, at a glance, looks like nonfixed time slot schedule. However, since every team has four game series on weekdays and two game series on weekends, the problem is actually composed of fixed time slots. We assume x is the number of games between two teams since the research does not show the exact number of games between them. Thus, the solution is ***1/intra /x, M/F/6.***

Schreuder's Dutch Professional Football schedule (1992) concerns 18 teams that meet twice between teams--double round robin. Also, the problem has fixed time slots and, thus, can be represented as ***1/intra /2/F/18***.

In 1977, Cain researched the Major League Baseball (MLB) scheduling. The league consists of the American and the National League including 2 divisions in each league and 6 teams in each division. The schedule consists of two and three game series. Three series of two game series are held in a week so that a week can have either three time slots for two game series and two time slots for three game series. Cain separated the whole schedule by three phases. During the first two phases a team had two series with every other team in the league. Hence, the first two phases are represented as ***1/intra/2, M/F/12*** (also can be represented as ***2/inter/in-2, out-2, M/6***) respectively. The third phase is composed of two series for all the other teams in the division so that there is a solution of ***1/intra/2/F/6***. The whole season schedule is then the same as Table 33.

Table 33. MLB schedule by Cain (1977):. represents ***intra***

League	Season schedule		
	Phase 1	Phase 2	Phase 3
National	***1/./2, M/F/12***	***1/./2, M/F/12***	***1/./2, M/F/6***
			1/./2, M/F/6
American	***1/./2, M/F/12***	***1/./2, M/F/12***	***1/./2, M/F/6***
			1/./2, M/F/6

Russel and Leung (1994) solved the Texas League, a AA Minor League, schedule. The league has two divisions of 4 teams in each division. As seen in Table 34, the authors divided a season into three segments with each segment composed of double round robin schedules.

Table 34. Texas League schedule by Russell and Leung (1994)

Division	Season schedule		
	Segment 1	Segment2	Segment 3
East	*1/.fl2/F/4*	*1/.fl2/F/8*	*1/.fl2/F/4*
West	*1/.fl2/F/4*		*1/.fl2/F/4*

The Southern League Baseball problem described in PART IV does not have fixed time slots. However, we converted the problem to the fixed time slots problem and permutated different schedules by using the multi echelon algorithm. Then, the final solutions from the fixed time slots problem were reverted to nonfixed time slots schedule at the refinement step. The season schedule after converting to fixed time slots problem is given in Table 35.

Table 35. The Southern League baseball scheduling

Division	Season schedule	
	First half	Second half
East	***2/inter/in-3, out-1, S-1, M/F/5***	***2/inter/in-3, out-1, S-1,M/F/5***
West	***2/inter/in-3, out-1, S-1, M/F/5***	***2/inter/in-3, out-1, S-1, M/F/5***

Conclusions

The different types of AGS problems are described depending on characteristics such as the number of meetings between two teams, the number of divisions in the league, the availability of fixed time slots, and the number of teams in a division. We assign a notation to each characteristic as in a ***A/B/C/D/E*** format where ***A*** is the number of divisions, ***B*** shows if there exists ***inter*** divisional games or not, ***C*** represents the number of round robin schedules, ***D*** is if the league consists of fixed time slots (***F***) or not (***N***), and ***E*** is the number of teams in a division. Then, a simple scheduling algorithm for a single round robin AGS problem is also provided which can be extended to other AGS problems such as double round robin problem. Finally, previous research was represented with the ***A/B/C/D/E*** format.

Many athletic leagues are not represented in this research. The NFL is an example. Nevertheless, one can easily analyze athletic leagues to a ***A/B/C/D/E*** format. The multi echelon solution procedure can be applied further to every athletic league including scheduling constraints.

GENERAL CONCLUSIONS

Athletic game scheduling (AGS) problems are deterministic and a class of NP hard problems. Also, most of the problems can only be solved by using heuristic methods because of the size of the problem and constraints that cannot be formulated. Even though a few researchers have applied the traditional operational research methods to the problems, they could not fully take into account all of the constraints an athletic league required. Also, all heuristics that developed in the previous research are problem specific.

Throughout the dissertation, the focus is on the development of a heuristic algorithm that can generate reasonably good solutions in a short period and can be applicable to any kind of AGS problem with minor corrections. The heuristic comprises three echelons. The first and second echelon explore the different combinations of game schedule, then the last echelon assigns the home and away schedule and permutes the different home and away schedule to the game schedule from the second echelon. We implemented Tabu search and simulated annealing (SA) algorithm to overcome entrapment in a local optimal solution.

The multi echelon heuristic is successfully applied to the development of the Southern League Baseball schedule for year 2000. Even though the final solution was not approved by the league committee, we gain great confidence for solving other AGS problems and we have learned a valuable lesson that communication between the league and the scheduler is crucial in every stage of development of AGS.

We also developed a method to determine the types AGS problems using a ***A/B/C/D/E*** format depending on the characteristics where ***A*** is the number of divisions, ***B*** is represented if there exists ***inter*** divisional games or not, ***C*** shows the number of round robin schedules, ***D*** is if the league consists of fixed time slots (***F***) or not (***N***), and ***E*** is the number of teams in a division. All the previous research were represented with the ***A/B/C/D/E*** format as an example. We also formulated a 0-1 integer program that minimized the total travel distance with minimum set of constraints.

Future research can be done in the way that combines the multi echelon heuristics with operational research methods. When solving the Southern League baseball problem, we found that the fair schedule for all teams was the most important factor for determining the acceptance of the final schedule by the committee. Thus, fair distribution of home and away games, especially on weekends, for all teams should be given when solving AGS problems for any leagues. Finally, the multi echelon method will be applied to the existing AGS problem such as NBA, NFL, MLB, and NHL etc. Research can be extended to include nonsymmetric traveling distances, an inter-league game for two or more leagues, consideration of the previous season schedule, the TV broadcasting schedule, when to schedule a rival game, and the use of different objective functions, *e.g.*, maximizing the total profit of all teams.

APPENDIX A. THE SOUTHERN LEAGUE SCHEDULE FOR YEAR 2000

Table A. 1. Game schedule and home/away schedule from minimizing the combination of travel distance and penalty costs

Half	Round	Team 1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	3	6	1	5	4	2	10	9	8	7
	2	4	3	2	1	10	8	9	6	7	5
	3	2	1	5	9	3	10	8	7	4	6
	4	10	7	9	8	6	5	2	4	3	1
	5	2	1	5	6	3	4	10	9	8	7
	6	5	4	10	2	1	8	9	6	7	3
	7	4	3	2	1	6	5	10	9	8	7
	8	5	4	7	2	1	9	3	10	6	8
	9	9	5	4	3	2	10	8	7	1	6
	10	6	10	8	7	9	1	4	3	5	2
	11	8	5	4	3	2	7	6	1	10	9
	12	2	1	5	10	3	8	9	6	7	4
	13	4	3	2	1	7	9	5	10	6	8
	14	3	8	1	5	4	7	6	2	10	9
	15	5	4	6	2	1	3	10	9	8	7
	16	3	9	1	5	4	10	8	7	2	6
	17	7	5	4	3	2	9	1	10	6	8
	18	4	3	2	1	8	7	6	5	10	9
SECOND HALF	19	2	1	5	10	3	8	9	6	7	4
	20	3	8	1	5	4	7	6	2	10	9
	21	4	3	2	1	6	5	10	9	8	7
	22	3	9	1	5	4	10	8	7	2	6
	23	2	1	5	6	3	4	10	9	8	7
	24	9	5	4	3	2	10	8	7	1	6
	25	4	3	2	1	7	9	5	10	6	8
	26	8	5	4	3	2	7	6	1	10	9
	27	4	3	2	1	10	8	9	6	7	5
	28	5	4	7	2	1	9	3	10	6	8
	29	4	3	2	1	8	7	6	5	10	9
	30	6	10	8	7	9	1	4	3	5	2

Half	Round	Team 1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	1	0	0	1	0	1	0	0	1	1
	2	0	1	0	1	0	1	1	0	0	1
	3	0	1	1	0	0	1	0	1	1	0
	4	1	0	1	0	1	0	1	1	0	0
	5	1	0	0	1	1	0	1	0	1	0
	6	1	0	0	1	0	1	1	0	0	1
	7	0	1	0	1	0	1	0	0	1	1
	8	1	1	1	0	0	1	0	1	0	0
	9	1	0	1	0	1	0	0	1	0	1
	10	0	1	1	0	1	1	1	0	0	0
	11	0	1	0	1	0	0	1	1	1	0
	12	1	0	0	0	1	0	0	1	1	1
	13	1	1	0	0	0	0	1	0	1	1
	14	0	1	1	0	1	1	0	0	0	1
	15	0	0	1	1	1	0	1	1	0	0
	16	0	0	1	1	0	1	1	0	1	0
	17	1	0	0	1	1	1	0	1	0	0
	18	1	1	0	0	1	0	1	0	0	1
SECOND HALF	19	0	1	0	1	1	0	0	1	1	0
	20	0	0	1	1	0	1	0	1	1	0
	21	1	0	1	0	1	0	1	1	0	0
	22	1	1	0	0	1	0	1	0	0	1
	23	0	1	1	0	0	1	0	1	0	1
	24	0	1	0	1	0	0	1	0	1	1
	25	1	0	1	0	1	0	0	1	1	0
	26	1	0	1	0	1	0	1	0	1	0
	27	0	0	1	1	1	1	0	0	0	0
	28	1	1	0	0	0	1	1	0	0	1
	29	0	0	1	1	0	1	0	1	0	1
	30	1	0	0	1	0	0	0	1	1	1

Table A. 1. (continued)

	Team									
	1	2	3	4	5	6	7	8	9	10
31	5	4	10	2	1	8	9	6	7	3
32	3	6	1	5	4	2	10	9	8	7
33	2	1	5	9	3	10	8	7	4	6
34	10	7	9	8	6	5	2	4	3	1
35	7	5	4	3	2	9	1	10	6	8
36	5	4	6	2	1	3	10	9	8	7

	Team									
	1	2	3	4	5	6	7	8	9	10
31	0	1	1	0	1	0	0	1	1	0
32	1	1	0	0	1	0	1	0	1	0
33	1	0	1	1	0	1	0	1	0	0
34	0	1	0	1	0	1	0	0	1	1
35	0	1	1	0	0	0	1	0	1	1
36	0	0	0	1	1	1	0	1	0	1

Table A. 2. Final schedule from minimizing the combination of travel distance and penalty costs

Total Penalty Costs: 3370 Total Travel Distance: 98885

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
6-Apr	Thursday		KNX	WTN		HNT		CAR	ORL		
7-Apr	Friday		KNX	WTN		HNT		CAR	ORL		
8-Apr	Saturday		KNX	WTN		HNT		CAR	ORL		
9-Apr	Sunday		KNX	WTN		HNT		CAR	ORL		
10-Apr	Monday	HNT		MOB		CAR			KNX	GRN	
11-Apr	Tuesday	HNT		MOB		CAR			KNX	GRN	
12-Apr	Wednesday	HNT		MOB		CAR			KNX	GRN	
13-Apr	Thursday	HNT		MOB		CAR			KNX	GRN	
14-Apr	Friday	MOB			OFF	BIR		OFF	OFF	OFF	KNX
15-Apr	Saturday	MOB			ORL	BIR		JAX			KNX
16-Apr	Sunday	MOB			ORL	BIR		JAX			KNX
17-Apr	Monday	MOB			ORL	BIR		JAX			KNX
18-Apr	Tuesday	OFF	OFF	OFF	ORL	OFF	OFF	JAX			OFF
19-Apr	Wednesday		GRN		JAX		CHT			BIR	WTN
20-Apr	Thursday		GRN		JAX		CHT			BIR	WTN
21-Apr	Friday		GRN		JAX		CHT			BIR	WTN
22-Apr	Saturday		GRN		JAX		CHT			BIR	WTN
23-Apr	Sunday	OFF	OFF	OFF		OFF	HNT	OFF	OFF	OFF	OFF
24-Apr	Monday		WTN	CHT			HNT		ORL		GRN
25-Apr	Tuesday		WTN	CHT			HNT		ORL		GRN
26-Apr	Wednesday		WTN	CHT			HNT		ORL		GRN
27-Apr	Thursday		WTN	CHT	OFF		OFF		ORL		GRN
28-Apr	Friday		HNT	CAR		WTN			KNX	GRN	
29-Apr	Saturday		HNT	CAR		WTN			KNX	GRN	
30-Apr	Sunday		HNT	CAR		WTN			KNX	GRN	

Table A. 2. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
1-May	Monday		HNT	CAR		WTN			KNX	GRN	
2-May	Tuesday	HNT		MOB		KNX		CAR	ORL		
3-May	Wednesday	HNT		MOB		KNX		CAR	ORL		
4-May	Thursday				MOB	WTN		BIR		KNX	JAX
5-May	Friday				MOB	WTN		BIR		KNX	JAX
6-May	Saturday				MOB	WTN		BIR		KNX	JAX
7-May	Sunday				MOB	WTN		BIR		KNX	JAX
8-May	Monday		CHT	OFF	OFF		OFF	JAX		WTN	OFF
9-May	Tuesday		CHT		BIR		CAR	JAX		WTN	
10-May	Wednesday		CHT		BIR		CAR	JAX		WTN	
11-May	Thursday		CHT		BIR		CAR	JAX		WTN	
12-May	Friday	OFF	OFF		BIR	OFF	CAR	OFF	OFF	OFF	
13-May	Saturday	KNX			GRN				BIR	CHT	MOB
14-May	Sunday	KNX			GRN				BIR	CHT	MOB
15-May	Monday	KNX			GRN				BIR	CHT	MOB
16-May	Tuesday	KNX			GRN				BIR	CHT	MOB
17-May	Wednesday	OFF		HNT		MOB	GRN		OFF	OFF	OFF
18-May	Thursday	JAX		HNT		MOB	GRN				ORL
19-May	Friday	JAX		HNT		MOB	GRN				ORL
20-May	Saturday	JAX		HNT		MOB	GRN				ORL
21-May	Sunday	JAX	OFF	CHT	OFF		OFF	OFF			ORL
22-May	Monday		WTN	CHT	CAR		JAX	ORL			
23-May	Tuesday		WTN	CHT	CAR		JAX	ORL			
24-May	Wednesday		WTN	CHT	CAR		JAX	ORL			
25-May	Thursday		WTN	OFF	CAR	OFF	JAX	ORL			
26-May	Friday			MOB	WTN	GRN	ORL		CAR		
27-May	Saturday			MOB	WTN	GRN	ORL		CAR		
28-May	Sunday			MOB	WTN	GRN	ORL		CAR		
29-May	Monday			MOB	WTN	GRN	ORL		CAR		
30-May	Tuesday	BIR			CHT			KNX	MOB	CAR	
31-May	Wednesday	BIR			CHT			KNX	MOB	CAR	
1-Jun	Thursday	BIR			CHT			KNX	MOB	CAR	
2-Jun	Friday	BIR			CHT			KNX	MOB	CAR	
3-Jun	Saturday	CHT	HNT				BIR			JAX	GRN
4-Jun	Sunday	CHT	HNT				BIR			JAX	GRN
5-Jun	Monday	CHT	HNT				BIR			JAX	GRN
6-Jun	Tuesday	CHT	HNT				BIR			JAX	GRN
7-Jun	Wednesday	BIR	ORL			HNT			GRN		KNX
8-Jun	Thursday	BIR	ORL			HNT			GRN		KNX
9-Jun	Friday	BIR	ORL			HNT			GRN		KNX
10-Jun	Saturday	BIR	ORL			HNT			GRN		KNX
11-Jun	Sunday		CHT	HNT				WTN		KNX	JAX

Table A. 2. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
12-Jun	Monday		CHT	HNT				WTN		KNX	JAX
13-Jun	Tuesday		CHT	HNT				WTN		KNX	JAX
14-Jun	Wednesday		CHT	HNT				WTN		KNX	JAX
15-Jun	Thursday			MOB	WTN		GRN		CHT	CAR	
16-Jun	Friday			MOB	WTN		GRN		CHT	CAR	
17-Jun	Saturday			MOB	WTN		GRN		CHT	CAR	
18-Jun	Sunday			MOB	WTN		GRN		CHT	CAR	
19-Jun	Monday	All	Star	break							
20-Jun	Tuesday	All	Star	break							
21-Jun	Wednesday	MOB		CHT			JAX	ORL			HNT
22-Jun	Thursday	MOB		CHT			JAX	ORL			HNT
23-Jun	Friday	MOB		CHT			JAX	ORL			HNT
24-Jun	Saturday	MOB		CHT			JAX	ORL			HNT
25-Jun	Sunday	BIR	JAX			HNT		KNX			ORL
26-Jun	Monday	BIR	JAX			HNT		KNX			ORL
27-Jun	Tuesday	BIR	JAX			HNT		KNX			ORL
28-Jun	Wednesday	BIR	JAX			HNT		KNX			ORL
29-Jun	Thursday		BIR		WTN		CHT			JAX	GRN
30-Jun	Friday		BIR		WTN		CHT			JAX	GRN
1-Jul	Saturday			WTN	CHT		CAR		GRN	MOB	
2-Jul	Sunday			WTN	CHT		CAR		GRN	MOB	
3-Jul	Monday			WTN	CHT		CAR		GRN	MOB	
4-Jul	Tuesday			WTN	CHT		CAR		GRN	MOB	
5-Jul	Wednesday	OFF	OFF		KNX	BIR		CAR		JAX	
6-Jul	Thursday	MOB			KNX	BIR		CAR		JAX	
7-Jul	Friday	MOB			KNX	BIR		CAR		JAX	
8-Jul	Saturday	MOB			KNX	BIR		CAR		JAX	
9-Jul	Sunday	MOB		HNT		OFF	CAR		GRN	OFF	
10-Jul	Monday	ORL		HNT		MOB	CAR		GRN		
11-Jul	Tuesday	ORL		HNT		MOB	CAR		GRN		
12-Jul	Wednesday	ORL		HNT		MOB	CAR		GRN		
13-Jul	Thursday	ORL		OFF	OFF	MOB	OFF	OFF	OFF		OFF
14-Jul	Friday		BIR		WTN		ORL	CHT			JAX
15-Jul	Saturday		BIR		WTN		ORL	CHT			JAX
16-Jul	Sunday		BIR		WTN		ORL	CHT			JAX
17-Jul	Monday		BIR		WTN		ORL	CHT			JAX
18-Jul	Tuesday		CHT		BIR		GRN		WTN		ORL
19-Jul	Wednesday		CHT		BIR		GRN		WTN		ORL
20-Jul	Thursday		CHT		BIR		GRN		WTN		ORL
21-Jul	Friday		CHT		BIR		GRN		WTN		ORL
22-Jul	Saturday	OFF	BIR		OFF	OFF	OFF	OFF	OFF	OFF	OFF
23-Jul	Sunday	HNT	BIR						KNX	GRN	CHT

Table A. 2. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
24-Jul	Monday	HNT	BIR						KNX	GRN	CHT
25-Jul	Tuesday	HNT	BIR						KNX	GRN	CHT
26-Jul	Wednesday	HNT	OFF	OFF					KNX	GRN	CHT
27-Jul	Thursday			GRN	MOB	WTN			CAR	KNX	
28-Jul	Friday			GRN	MOB	WTN			CAR	KNX	
29-Jul	Saturday			GRN	MOB	WTN			CAR	KNX	
30-Jul	Sunday			GRN	MOB	WTN			CAR	KNX	
31-Jul	Monday	HNT	BIR			JAX	OFF	OFF		CAR	
1-Aug	Tuesday	HNT	BIR			JAX		KNX		CAR	
2-Aug	Wednesday	HNT	BIR			JAX		KNX		CAR	
3-Aug	Thursday	HNT	BIR			JAX		KNX		CAR	
4-Aug	Friday	OFF	CAR	JAX	OFF	OFF		KNX		OFF	
5-Aug	Saturday		CAR	JAX		ORL	WTN	HNT			
6-Aug	Sunday		CAR	JAX		ORL	WTN	HNT			
7-Aug	Monday		CAR	JAX		ORL	WTN	HNT			
8-Aug	Tuesday		OFF	OFF		ORL	WTN	HNT	OFF		OFF
9-Aug	Wednesday	CHT			MOB		JAX	ORL			BIR
10-Aug	Thursday	CHT			MOB		JAX	ORL			BIR
11-Aug	Friday	CHT			MOB		JAX	ORL			BIR
12-Aug	Saturday	CHT			MOB		JAX	ORL			BIR
13-Aug	Sunday	OFF		OFF	CHT		MOB		OFF	OFF	GRN
14-Aug	Monday			WTN	CHT		MOB		ORL		GRN
15-Aug	Tuesday			WTN	CHT		MOB		ORL		GRN
16-Aug	Wednesday			WTN	CHT		MOB		ORL		GRN
17-Aug	Thursday		OFF	WTN	OFF	OFF		OFF	ORL		KNX
18-Aug	Friday		WTN			BIR		JAX		HNT	KNX
19-Aug	Saturday		WTN			BIR		JAX		HNT	KNX
20-Aug	Sunday		WTN			BIR		JAX		HNT	KNX
21-Aug	Monday		WTN			BIR	OFF	JAX		HNT	OFF
22-Aug	Tuesday	CAR		OFF		KNX		MOB	HNT	OFF	
23-Aug	Wednesday	CAR		ORL		KNX		MOB	HNT		
24-Aug	Thursday	CAR		ORL		KNX		MOB	HNT		
25-Aug	Friday	CAR		ORL		KNX		MOB	HNT		
26-Aug	Saturday	OFF	OFF	ORL	OFF	OFF	OFF	OFF	OFF		OFF
27-Aug	Sunday	GRN			BIR	MOB	ORL		CAR		
28-Aug	Monday	GRN			BIR	MOB	ORL		CAR		
29-Aug	Tuesday	GRN			BIR	MOB	ORL		CAR		
30-Aug	Wednesday	GRN			BIR	MOB	ORL		CAR		
31-Aug	Thursday	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
1-Sep	Friday	CHT	HNT	KNX				CAR		JAX	

Table A. 2. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
2-Sep	Saturday	CHT	HNT	KNX				CAR		JAX	
3-Sep	Sunday	CHT	HNT	KNX				CAR		JAX	
4-Sep	Monday	CHT	HNT	KNX				CAR		JAX	

The constraints that are not satisfied

Team WTN does not have day off for more than 30 days
Team JAX does not have day off for more than 30 days
Team ORL does not have day off for more than 30 days
Team CAR does not have day off for more than 30 days
Team WTN travels 563 miles without day off in the first half
Team WTN travels 565 miles without day off in the second half
Team BIR travels 565 miles without day off in the first half
Team CHT travels 563 miles without day off in the first half
Team KNX travels 563 miles without day off in the first half
Team KNX travels 610 miles without day off in the first half
Team KNX travels 663 miles without day off in the first half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 610 miles without day off in the first half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 563 miles without day off in the second half
Team JAX travels 546 miles without day off in the first half
Team JAX travels 517 miles without day off in the second half
Team ORL travels 660 miles without day off in the first half
Team ORL travels 663 miles without day off in the first half
Team ORL travels 610 miles without day off in the second half
Team CAR travels 535 miles without day off in the first half
Team CAR travels 586 miles without day off in the first half
Team CAR travels 565 miles without day off in the first half
Team CAR travels 610 miles without day off in the first half
Team CAR travels 610 miles without day off in the first half
CAR does not have home game on the Independence day
BIR has to have different schedule before and after the all star break
KNX has to have different schedule before and after the all star break
Weekend games for WTN do not met
Weekend games for MOB do not met
Weekend games for KNX do not met
Weekend games for JAX do not met
WTN needs more home games in May
WTN needs fewer home games in August
MOB wants home opener with BIR

BIR wants road games on the third May weekend
BIR dislikes home game on Monday
HNT likes to open in Florida
HNT prefers home game in the first week of May
CHT likes to open on the road
KNX doesn't likes direct travel to/from Florida
GRN wants fewer home games on weekend in April and May
JAX likes to open at home
JAX wants more home games on weekend in April and May
JAX doesn't want home games in August and September

Table A. 3. Game schedule and home/away schedule from minimizing the penalty costs

Half	Team 0	1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	7	6	9	8	10	2	1	4	3	5
	2	5	9	4	3	1	8	10	6	2	7
	3	4	5	6	1	2	3	9	10	7	8
	4	6	10	7	9	8	1	3	5	4	2
	5	4	5	10	1	2	9	8	7	6	3
	6	3	4	1	2	9	8	10	6	5	7
	7	5	8	4	3	1	7	6	2	10	9
	8	3	4	1	2	7	10	5	9	8	6
	9	9	3	2	5	4	8	10	6	1	7
	10	5	7	4	3	1	10	2	9	8	6
	11	10	3	2	5	4	9	8	7	6	1
	12	2	1	5	6	3	4	9	10	7	8
	13	4	3	2	1	6	5	10	9	8	7
	14	2	1	5	10	3	9	8	7	6	4
	15	8	3	2	5	4	7	6	1	10	9
	16	3	4	1	2	6	5	9	10	7	8
	17	2	1	5	7	3	10	4	9	8	6
	18	4	5	8	1	2	7	6	3	10	9
SECOND HALF	19	9	3	2	5	4	8	10	6	1	7
	20	4	5	10	1	2	9	8	7	6	3
	21	3	4	1	2	7	10	5	9	8	6
	22	2	1	5	10	3	9	8	7	6	4
	23	3	4	1	2	6	5	9	10	7	8
	24	2	1	5	7	3	10	4	9	8	6
	25	4	5	6	1	2	3	9	10	7	8
	26	7	6	9	8	10	2	1	4	3	5
	27	3	4	1	2	9	8	10	6	5	7
	28	4	5	8	1	2	7	6	3	10	9
	29	6	10	7	9	8	1	3	5	4	2
	30	4	3	2	1	6	5	10	9	8	7
	31	5	8	4	3	1	7	6	2	10	9
	32	10	3	2	5	4	9	8	7	6	1
	33	5	7	4	3	1	10	2	9	8	6
	34	8	3	2	5	4	7	6	1	10	9
	35	2	1	5	6	3	4	9	10	7	8
	36	5	9	4	3	1	8	10	6	2	7

Half	Team 0	1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	1	0	0	1	0	1	0	0	1	1
	2	0	0	0	1	1	1	0	0	1	1
	3	1	1	0	0	0	1	1	1	0	0
	4	1	1	1	0	0	0	0	1	1	0
	5	0	0	0	1	1	1	1	0	0	1
	6	0	0	1	1	1	1	0	0	0	1
	7	1	0	0	1	0	0	1	1	1	0
	8	0	1	1	0	1	1	0	0	1	0
	9	1	1	0	1	0	1	1	0	0	0
	10	0	1	1	0	1	0	0	1	0	1
	11	0	0	1	0	1	0	1	0	1	1
	12	1	0	1	0	0	1	0	1	1	0
	13	0	1	0	1	0	1	0	1	0	1
	14	1	0	1	1	0	0	0	1	1	0
	15	1	0	1	0	1	0	1	0	1	0
	16	0	0	1	1	0	1	1	0	0	1
	17	0	1	0	1	1	0	0	1	0	1
	18	1	0	1	0	1	1	0	0	0	1
SECOND HALF	19	0	1	0	1	0	0	1	1	1	0
	20	0	1	1	1	0	1	0	1	0	0
	21	1	0	0	1	0	0	1	1	0	1
	22	0	1	0	0	1	0	1	0	1	1
	23	1	1	0	0	1	0	0	1	1	0
	24	1	0	1	0	0	1	1	0	1	0
	25	0	1	1	1	0	0	1	0	0	1
	26	0	1	1	0	1	0	1	1	0	0
	27	1	1	0	0	0	0	0	1	1	1
	28	1	0	0	0	1	1	0	1	0	1
	29	0	0	0	1	1	1	1	0	0	1
	30	1	0	1	0	1	0	1	0	1	0
	31	1	1	1	0	0	0	1	0	0	1
	32	1	0	1	1	0	1	0	1	0	0
	33	0	0	0	1	1	1	1	0	1	0
	34	0	1	0	0	1	1	0	1	1	0
	35	0	1	0	1	1	0	0	0	1	1
	36	1	1	1	0	0	0	1	1	0	0

Table A. 4. Final schedule from minimizing the penalty costs

Total Penalty Costs: 3770 Total Travel Distance: 103664

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
6-Apr	Thursday		KNX	ORL		CAR		WTN	HNT		
7-Apr	Friday		KNX	ORL		CAR		WTN	HNT		
8-Apr	Saturday		KNX	ORL		CAR		WTN	HNT		
9-Apr	Sunday		KNX	ORL		CAR		WTN	HNT		
10-Apr	Monday	CHT	ORL	HNT				CAR	KNX		
11-Apr	Tuesday	CHT	ORL	HNT				CAR	KNX		
12-Apr	Wednesday	CHT	ORL	HNT				CAR	KNX		
13-Apr	Thursday	CHT	ORL	HNT				CAR	KNX		
14-Apr	Friday			KNX	WTN	MOB				GRN	JAX
15-Apr	Saturday			KNX	WTN	MOB				GRN	JAX
16-Apr	Sunday			KNX	WTN	MOB				GRN	JAX
17-Apr	Monday			KNX	WTN	MOB				GRN	JAX
18-Apr	Tuesday				ORL	JAX	WTN	BIR			MOB
19-Apr	Wednesday				ORL	JAX	WTN	BIR			MOB
20-Apr	Thursday				ORL	JAX	WTN	BIR			MOB
21-Apr	Friday				ORL	JAX	WTN	BIR			MOB
22-Apr	Saturday	HNT	OFF	CAR		OFF	OFF		GRN	OFF	
23-Apr	Sunday	HNT	CHT	CAR					GRN	KNX	
24-Apr	Monday	HNT	CHT	CAR					GRN	KNX	
25-Apr	Tuesday	HNT	CHT	CAR					GRN	KNX	
26-Apr	Wednesday	BIR	CHT		OFF			CAR	OFF	KNX	
27-Apr	Thursday	BIR	HNT					CAR	KNX	CHT	
28-Apr	Friday	BIR	HNT					CAR	KNX	CHT	
29-Apr	Saturday	BIR	HNT					CAR	KNX	CHT	
30-Apr	Sunday	OFF	HNT	OFF				OFF	KNX	CHT	OFF
1-May	Monday		JAX	HNT		WTN	GRN				ORL
2-May	Tuesday		JAX	HNT		WTN	GRN				ORL
3-May	Wednesday		JAX	HNT		WTN	GRN				ORL
4-May	Thursday		JAX	HNT		WTN	GRN				ORL
5-May	Friday	BIR			MOB			CHT	ORL		KNX
6-May	Saturday	BIR			MOB			CHT	ORL		KNX
7-May	Sunday	BIR			MOB			CHT	ORL		KNX
8-May	Monday	BIR			MOB			CHT	ORL		KNX
9-May	Tuesday	OFF		MOB	OFF	OFF			KNX	OFF	GRN
10-May	Wednesday			MOB		HNT			KNX	WTN	GRN
11-May	Thursday			MOB		HNT			KNX	WTN	GRN
12-May	Friday			MOB		HNT			KNX	WTN	GRN
13-May	Saturday		OFF	OFF		HNT	OFF	OFF	OFF	WTN	OFF
14-May	Sunday	OFF			BIR	OFF	CAR	MOB		JAX	

Table A. 4. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
15-May	Monday	CHT			BIR		CAR	MOB		JAX	
16-May	Tuesday	CHT			BIR		CAR	MOB		JAX	
17-May	Wednesday	CHT			BIR		CAR	MOB		JAX	
18-May	Thursday	CHT	BIR		OFF		ORL		GRN		OFF
19-May	Friday	CAR	BIR		CHT		ORL		GRN		
20-May	Saturday	CAR	BIR		CHT		ORL		GRN		
21-May	Sunday	CAR	BIR		CHT		ORL		GRN		
22-May	Monday	CAR		CHT		OFF	ORL	OFF			
23-May	Tuesday		WTN		KNX	BIR		ORL			JAX
24-May	Wednesday		WTN		KNX	BIR		ORL			JAX
25-May	Thursday		WTN		KNX	BIR		ORL			JAX
26-May	Friday			MOB	KNX		OFF		OFF	JAX	
27-May	Saturday	HNT		MOB		KNX		CAR		JAX	
28-May	Sunday	HNT	OFF	OFF		KNX		CAR		JAX	
29-May	Monday		WTN		OFF	BIR	OFF	JAX		OFF	OFF
30-May	Tuesday		WTN			BIR	ORL	JAX			HNT
31-May	Wednesday		WTN			BIR	ORL	JAX			HNT
1-Jun	Thursday		WTN			BIR	ORL	JAX			HNT
2-Jun	Friday		BIR			OFF	ORL	OFF	WTN		HNT
3-Jun	Saturday		BIR		CHT		GRN		WTN		ORL
4-Jun	Sunday		BIR		CHT		GRN		WTN		ORL
5-Jun	Monday		BIR		CHT		GRN		WTN		ORL
6-Jun	Tuesday	OFF	OFF	OFF	CHT		GRN		OFF		ORL
7-Jun	Wednesday	BIR	HNT			KNX			CAR	GRN	
8-Jun	Thursday	BIR	HNT			KNX			CAR	GRN	
9-Jun	Friday	BIR	HNT			KNX			CAR	GRN	
10-Jun	Saturday	BIR	HNT			KNX			CAR	GRN	
11-Jun	Sunday	MOB		CHT			CAR	HNT		JAX	
12-Jun	Monday	MOB		CHT			CAR	HNT		JAX	
13-Jun	Tuesday	MOB		CHT			CAR	HNT		JAX	
14-Jun	Wednesday	MOB		CHT			CAR	HNT		JAX	
15-Jun	Thursday		CHT		WTN			KNX	BIR	CAR	
16-Jun	Friday		CHT		WTN			KNX	BIR	CAR	
17-Jun	Saturday		CHT		WTN			KNX	BIR	CAR	
18-Jun	Sunday		CHT		WTN			KNX	BIR	CAR	
19-Jun	Monday	All Star break									
20-Jun	Tuesday	All Star break									
21-Jun	Wednesday	ORL		MOB		HNT	JAX				GRN
22-Jun	Thursday	ORL		MOB		HNT	JAX				GRN
23-Jun	Friday	ORL		MOB		HNT	JAX				GRN
24-Jun	Saturday	ORL		MOB		HNT	JAX				GRN

Table A. 4. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
25-Jun	Sunday	HNT				MOB	OFF	JAX		OFF	BIR
26-Jun	Monday	HNT				MOB		JAX		KNX	BIR
27-Jun	Tuesday	HNT				MOB		JAX		KNX	BIR
28-Jun	Wednesday	HNT				MOB		JAX		KNX	BIR
29-Jun	Thursday		HNT	WTN		GRN			OFF	KNX	OFF
30-Jun	Friday		HNT	WTN		GRN	CAR			JAX	
1-Jul	Saturday		HNT	WTN		GRN	CAR			JAX	
2-Jul	Sunday		HNT	WTN		GRN	CAR			JAX	
3-Jul	Monday	MOB		CHT	OFF		CAR	OFF		JAX	
4-Jul	Tuesday	MOB		CHT	CAR		ORL		GRN		
5-Jul	Wednesday	MOB		CHT	CAR		ORL		GRN		
6-Jul	Thursday	MOB		CHT	CAR		ORL		GRN		
7-Jul	Friday		OFF	WTN	CAR	OFF	ORL		GRN		
8-Jul	Saturday			WTN	MOB		CHT	ORL			JAX
9-Jul	Sunday			WTN	MOB		CHT	ORL			JAX
10-Jul	Monday			WTN	MOB		CHT	ORL			JAX
11-Jul	Tuesday	OFF		OFF	MOB		CHT	ORL			JAX
12-Jul	Wednesday		WTN		GRN	BIR			ORL		KNX
13-Jul	Thursday		WTN		GRN	BIR			ORL		KNX
14-Jul	Friday		WTN		GRN	BIR			ORL		KNX
15-Jul	Saturday		WTN		GRN	BIR			ORL		KNX
16-Jul	Sunday	HNT				MOB	BIR		CAR	GRN	
17-Jul	Monday	HNT				MOB	BIR		CAR	GRN	
18-Jul	Tuesday	HNT				MOB	BIR		CAR	GRN	
19-Jul	Wednesday	HNT				MOB	BIR		CAR	GRN	
20-Jul	Thursday	OFF		OFF	OFF		MOB	OFF	OFF	OFF	CHT
21-Jul	Friday	GRN			JAX		MOB			BIR	CHT
22-Jul	Saturday	GRN			JAX		MOB			BIR	CHT
23-Jul	Sunday	GRN			JAX		MOB			BIR	CHT
24-Jul	Monday	GRN	OFF		JAX	OFF	OFF			BIR	OFF
25-Jul	Tuesday			WTN	MOB	ORL	JAX	CAR			
26-Jul	Wednesday			WTN	MOB	ORL	JAX	CAR			
27-Jul	Thursday			WTN	MOB	ORL	JAX	CAR			
28-Jul	Friday			WTN	MOB	ORL	JAX	CAR			
29-Jul	Saturday		CHT	JAX	WTN			KNX		CAR	
30-Jul	Sunday		CHT	JAX	WTN			KNX		CAR	
31-Jul	Monday		CHT	JAX	WTN			KNX		CAR	
1-Aug	Tuesday		CHT	JAX	WTN			KNX		CAR	
2-Aug	Wednesday	KNX	CAR	GRN					CHT	HNT	
3-Aug	Thursday	KNX	CAR	GRN					CHT	HNT	
4-Aug	Friday	KNX	CAR	GRN					CHT	HNT	

Table A. 4. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
5-Aug	Saturday	KNX	CAR	GRN					CHT	HNT	
6-Aug	Sunday	OFF	BIR		OFF		CHT	OFF	ORL		OFF
7-Aug	Monday		BIR		WTN		CHT		ORL		GRN
8-Aug	Tuesday			OFF	WTN	OFF	OFF		MOB	OFF	GRN
9-Aug	Wednesday				BIR	WTN	GRN		MOB	CAR	
10-Aug	Thursday				BIR	WTN	GRN		MOB	CAR	
11-Aug	Friday				BIR	WTN	GRN		MOB	CAR	
12-Aug	Saturday		OFF		BIR	WTN	GRN		OFF	CAR	
13-Aug	Sunday		BIR			HNT		JAX		KNX	WTN
14-Aug	Monday		BIR			HNT		JAX		KNX	WTN
15-Aug	Tuesday		BIR			HNT		JAX		KNX	WTN
16-Aug	Wednesday		BIR			HNT		JAX		KNX	WTN
17-Aug	Thursday	OFF	GRN	HNT		OFF	OFF		ORL		OFF
18-Aug	Friday	CHT	GRN	HNT					ORL		KNX
19-Aug	Saturday	CHT	GRN	HNT					ORL		KNX
20-Aug	Sunday	CHT	GRN	HNT					ORL		KNX
21-Aug	Monday	CHT	OFF	OFF	OFF			OFF	OFF	OFF	KNX
22-Aug	Tuesday	JAX		MOB	CHT			KNX			ORL
23-Aug	Wednesday	JAX		MOB	CHT			KNX			ORL
24-Aug	Thursday	JAX		MOB	CHT			KNX			ORL
25-Aug	Friday	JAX		MOB	CHT			KNX			ORL
26-Aug	Saturday	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
27-Aug	Sunday	MOB		CHT			HNT	ORL	CAR		
28-Aug	Monday	MOB		CHT			HNT	ORL	CAR		
29-Aug	Tuesday	MOB		CHT			HNT	ORL	CAR		
30-Aug	Wednesday	MOB		CHT			HNT	ORL	CAR		
31-Aug	Thursday	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
1-Sep	Friday				BIR	WTN	JAX			MOB	GRN
2-Sep	Saturday				BIR	WTN	JAX			MOB	GRN
3-Sep	Sunday				BIR	WTN	JAX			MOB	GRN
4-Sep	Monday				BIR	WTN	JAX			MOB	GRN

The constraints that are not satisfied

Team BIR travels 535 miles without day off in the second half
Team BIR travels 535 miles without day off in the second half
Team BIR travels 563 miles without day off in the second half
Team HNT travels 586 miles without day off in the first half
Team HNT travels 660 miles without day off in the second half
Team CHT travels 563 miles without day off in the first half
Team CHT travels 546 miles without day off in the second half
Team KNX travels 546 miles without day off in the first half
Team KNX travels 663 miles without day off in the second half
Team KNX travels 663 miles without day off in the second half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 663 miles without day off in the first half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 660 miles without day off in the second half
Team JAX travels 610 miles without day off in the first half
Team JAX travels 526 miles without day off in the first half
Team ORL travels 660 miles without day off in the first half
Team ORL travels 610 miles without day off in the first half
Team ORL travels 663 miles without day off in the first half
Team ORL travels 610 miles without day off in the first half
Team ORL travels 663 miles without day off in the second half
Team ORL travels 563 miles without day off in the second half
Team ORL travels 563 miles without day off in the second half
Team CAR travels 535 miles without day off in the first half
Team CAR travels 687 miles without day off in the first half
Team CAR travels 546 miles without day off in the first half
Team CAR travels 663 miles without day off in the first half
Team CAR travels 586 miles without day off in the second half
Team CAR travels 526 miles without day off in the second half
Team CAR travels 610 miles without day off in the second half
Team CAR travels 610 miles without day off in the second half
CAR does not have home game on the Independence day
Weekend games for WTN do not met
Weekend games for HNT do not met
Weekend games for KNX do not met
Weekend games for GRN do not met
Weekend games for JAX do not met
WTN needs more home games in May
WTN needs fewer home games in August
MOB wants home opener with BIR
BIR dislikes home game on Monday

HNT likes to open in Florida
HNT prefers home game in the first week of May
HNT prefers home game in the first week of May
CHT likes to open on the road
KNX likes to close on the road
GRN wants fewer home games on weekend in April and May
JAX likes to open at home
JAX wants more home games on weekend in April and May
JAX doesn't want home games in August and September
CAR prefers less home games in August

Table A. 5. Game schedule and home/away schedule from minimizing the travel distance

	Team										
	0	1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	5	3	2	10	1	7	6	9	8	4
	2	7	6	10	9	8	2	1	5	4	3
	3	5	3	2	8	1	9	10	4	6	7
	4	4	5	7	1	2	8	3	6	10	9
	5	2	1	4	3	6	5	9	10	7	8
	6	8	4	5	2	3	9	10	1	6	7
	7	4	5	9	1	2	10	8	7	3	6
	8	3	10	1	5	4	7	6	9	8	2
	9	4	5	6	1	2	3	9	10	7	8
	10	2	1	4	3	7	8	5	6	10	9
	11	3	8	1	5	4	9	10	2	6	7
	12	9	4	5	2	3	10	8	7	1	6
	13	4	3	2	1	6	5	10	9	8	7
	14	6	4	5	2	3	1	9	10	7	8
	15	3	9	1	5	4	10	8	7	2	6
	16	10	7	8	6	9	4	2	3	5	1
	17	5	3	2	7	1	8	4	6	10	9
	18	2	1	4	3	10	7	6	9	8	5
SECOND HALF	19	4	5	6	1	2	3	9	10	7	8
	20	8	4	5	2	3	9	10	1	6	7
	21	2	1	4	3	6	5	9	10	7	8
	22	3	8	1	5	4	9	10	2	6	7
	23	5	3	2	10	1	7	6	9	8	4
	24	4	5	7	1	2	8	3	6	10	9
	25	5	3	2	8	1	9	10	4	6	7
	26	7	6	10	9	8	2	1	5	4	3
	27	4	5	9	1	2	10	8	7	3	6
	28	3	10	1	5	4	7	6	9	8	2
	29	5	3	2	7	1	8	4	6	10	9
	30	3	9	1	5	4	10	8	7	2	6
	31	6	4	5	2	3	1	9	10	7	8
	32	10	7	8	6	9	4	2	3	5	1
	33	2	1	4	3	10	7	6	9	8	5
	34	9	4	5	2	3	10	8	7	1	6
	35	2	1	4	3	7	8	5	6	10	9
	36	4	3	2	1	6	5	10	9	8	7

	Team										
	0	1	2	3	4	5	6	7	8	9	10
FIRST HALF	1	1	0	1	0	0	1	0	0	1	1
	2	1	0	0	1	1	1	0	0	0	1
	3	1	1	0	1	0	1	1	0	0	0
	4	0	1	0	1	0	0	1	1	0	1
	5	0	1	1	0	1	0	0	0	1	1
	6	0	0	0	1	1	0	1	1	1	0
	7	1	1	0	0	0	1	0	1	1	0
	8	0	1	1	0	1	1	0	1	0	0
	9	1	0	1	0	1	0	1	0	0	1
	10	1	0	0	1	0	0	1	1	0	1
	11	1	1	0	1	0	0	0	0	1	1
	12	0	1	0	0	1	1	1	0	1	0
	13	1	0	1	0	0	1	0	0	1	1
	14	0	0	0	1	1	1	1	1	0	0
	15	0	0	1	1	0	0	1	0	1	1
	16	1	1	1	0	0	1	0	0	1	0
	17	0	1	0	0	1	1	1	0	0	1
	18	0	1	1	0	1	0	1	1	0	0
SECOND HALF	19	0	0	0	1	1	1	0	1	1	0
	20	1	1	1	0	0	0	0	0	1	1
	21	1	0	0	1	0	1	0	0	1	1
	22	0	0	1	0	1	1	1	1	0	0
	23	0	0	1	1	1	0	1	1	0	0
	24	0	1	1	1	0	1	0	0	1	0
	25	1	1	0	0	0	1	0	1	0	1
	26	0	1	1	0	0	0	1	1	1	0
	27	1	0	1	0	1	0	1	0	0	1
	28	1	0	0	1	0	0	1	0	1	1
	29	0	0	1	1	1	1	0	0	1	0
	30	1	1	0	0	1	1	0	1	0	0
	31	1	1	1	0	0	0	1	1	0	0
	32	0	0	0	1	1	0	1	1	0	1
	33	1	0	0	1	0	1	0	0	1	1
	34	1	0	1	1	0	0	0	1	0	1
	35	0	1	1	0	1	0	0	1	1	0
	36	0	1	0	1	1	0	1	1	0	0

Table A. 6. Final schedule from minimizing the travel distance

Total Penalty Costs: 3565 Total Travel Distance: 98358

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
6-Apr	Thursday		BIR		CAR	WTN		KNX	ORL		
7-Apr	Friday		BIR		CAR	WTN		KNX	ORL		
8-Apr	Saturday		BIR		CAR	WTN		KNX	ORL		
9-Apr	Sunday		BIR		CAR	WTN		KNX	ORL		
10-Apr	Monday		KNX	CAR	OFF			WTN	CHT	OFF	
11-Apr	Tuesday		KNX	CAR				WTN	CHT	HNT	
12-Apr	Wednesday		KNX	CAR				WTN	CHT	HNT	
13-Apr	Thursday		KNX	CAR				WTN	CHT	HNT	
14-Apr	Friday	OFF		MOB		OFF	OFF		OFF	HNT	GRN
15-Apr	Saturday			MOB		WTN			HNT	KNX	GRN
16-Apr	Sunday			MOB		WTN			HNT	KNX	GRN
17-Apr	Monday			MOB		WTN			HNT	KNX	GRN
18-Apr	Tuesday		OFF	OFF		WTN		OFF	HNT	KNX	OFF
19-Apr	Wednesday	OFF		GRN	OFF	MOB	OFF		OFF	CAR	
20-Apr	Thursday	HNT		GRN		MOB	JAX			CAR	
21-Apr	Friday	HNT		GRN		MOB	JAX			CAR	
22-Apr	Saturday	HNT		GRN		MOB	JAX			CAR	
23-Apr	Sunday	HNT	OFF	OFF		OFF	JAX	ORL			
24-Apr	Monday	MOB			BIR		CHT	ORL	CAR		
25-Apr	Tuesday	MOB			BIR		CHT	ORL	CAR		
26-Apr	Wednesday	MOB			BIR		CHT	ORL	CAR		
27-Apr	Thursday	MOB			BIR		CHT	OFF	OFF	OFF	
28-Apr	Friday	OFF	HNT	CHT			ORL		OFF		GRN
29-Apr	Saturday	JAX	HNT	CHT			ORL				GRN
30-Apr	Sunday	JAX	HNT	CHT			ORL				GRN
1-May	Monday	JAX	HNT	CHT			ORL				GRN
2-May	Tuesday	JAX		ORL	OFF	MOB		OFF			KNX
3-May	Wednesday			ORL	WTN	MOB		JAX			KNX
4-May	Thursday			ORL	WTN	MOB		JAX			KNX
5-May	Friday			ORL	WTN	MOB		JAX			KNX
6-May	Saturday				WTN	OFF	OFF	JAX			MOB
7-May	Sunday	BIR			CHT			KNX		JAX	MOB
8-May	Monday	BIR			CHT			KNX		JAX	MOB
9-May	Tuesday	BIR			CHT			KNX		JAX	MOB
10-May	Wednesday	BIR	OFF		CHT			KNX		JAX	OFF
11-May	Thursday		CHT		WTN		BIR		CAR	GRN	
12-May	Friday		CHT		WTN		BIR		CAR	GRN	
13-May	Saturday		CHT		WTN		BIR		CAR	GRN	
14-May	Sunday		CHT		WTN		BIR		CAR	GRN	

Table A. 6. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
15-May	Monday		WTN	HNT		GRN	JAX			CAR	
16-May	Tuesday		WTN	HNT		GRN	JAX			CAR	
17-May	Wednesday		WTN	HNT		GRN	JAX			CAR	
18-May	Thursday		WTN	HNT		GRN	JAX			CAR	
19-May	Friday			WTN		HNT	ORL	CAR	MOB		
20-May	Saturday			WTN		HNT	ORL	CAR	MOB		
21-May	Sunday			WTN		HNT	ORL	CAR	MOB		
22-May	Monday			WTN		HNT	ORL	CAR	MOB		
23-May	Tuesday	ORL		CHT	MOB				GRN		KNX
24-May	Wednesday	ORL		CHT	MOB				GRN		KNX
25-May	Thursday	ORL		CHT	MOB				GRN		KNX
26-May	Friday	ORL		CHT	MOB				GRN		KNX
27-May	Saturday		BIR		WTN	KNX		CAR	OFF	OFF	
28-May	Sunday		BIR		WTN	KNX		CAR	ORL		
29-May	Monday	KNX	HNT	CHT				OFF	ORL		OFF
30-May	Tuesday	KNX	HNT	CHT						GRN	JAX
31-May	Wednesday	KNX	HNT	CHT						GRN	JAX
1-Jun	Thursday	KNX	HNT	CHT						GRN	JAX
2-Jun	Friday	BIR	OFF			HNT	OFF			GRN	JAX
3-Jun	Saturday	BIR	ORL			HNT	CAR		GRN		
4-Jun	Sunday	BIR	ORL			HNT	CAR		GRN		
5-Jun	Monday	BIR	ORL			HNT	CAR		GRN		
6-Jun	Tuesday	OFF	ORL	OFF	OFF	OFF	CAR		GRN		
7-Jun	Wednesday				KNX	ORL		MOB	BIR		WTN
8-Jun	Thursday				KNX	ORL		MOB	BIR		WTN
9-Jun	Friday				KNX	ORL		MOB	BIR		WTN
10-Jun	Saturday				KNX	ORL		MOB	BIR		WTN
11-Jun	Sunday	CHT		MOB	GRN				KNX	CAR	
12-Jun	Monday	CHT		MOB	GRN				KNX	CAR	
13-Jun	Tuesday	CHT		MOB	GRN				KNX	CAR	
14-Jun	Wednesday	CHT		MOB	GRN				KNX	CAR	
15-Jun	Thursday	MOB			BIR		GRN			JAX	CHT
16-Jun	Friday	MOB			BIR		GRN			JAX	CHT
17-Jun	Saturday	MOB			BIR		GRN			JAX	CHT
18-Jun	Sunday	MOB			BIR		GRN			JAX	CHT
19-Jun	Monday	All Star break									
20-Jun	Tuesday	All Star break									
21-Jun	Wednesday	HNT	CHT	KNX				ORL			JAX
22-Jun	Thursday	HNT	CHT	KNX				ORL			JAX
23-Jun	Friday	HNT	CHT	KNX				ORL			JAX
24-Jun	Saturday	HNT	CHT	KNX				ORL			JAX

Table A. 6. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
25-Jun	Sunday	OFF			MOB	BIR	ORL	CAR	OFF		
26-Jun	Monday				MOB	BIR	ORL	CAR	WTN		
27-Jun	Tuesday				MOB	BIR	ORL	CAR	WTN		
28-Jun	Wednesday				MOB	BIR	ORL	CAR	WTN		
29-Jun	Thursday		OFF	HNT		KNX		ORL	WTN		OFF
30-Jun	Friday		WTN	HNT		KNX		ORL	CAR		
1-Jul	Saturday		WTN	HNT		KNX		ORL	CAR		
2-Jul	Sunday		WTN	HNT		KNX		ORL	CAR		
3-Jul	Monday		WTN	OFF	CHT			OFF	CAR	KNX	
4-Jul	Tuesday	BIR	JAX		CHT					KNX	GRN
5-Jul	Wednesday	BIR	JAX		CHT					KNX	GRN
6-Jul	Thursday	BIR	JAX		CHT					KNX	GRN
7-Jul	Friday	BIR	JAX		OFF	OFF	OFF			OFF	GRN
8-Jul	Saturday	CHT	BIR				GRN			JAX	HNT
9-Jul	Sunday	CHT	BIR				GRN			JAX	HNT
10-Jul	Monday	CHT	BIR				GRN			JAX	HNT
11-Jul	Tuesday	CHT	BIR				GRN			JAX	HNT
12-Jul	Wednesday	OFF			OFF	MOB		BIR	KNX	OFF	OFF
13-Jul	Thursday	HNT				MOB		BIR	KNX		ORL
14-Jul	Friday	HNT				MOB		BIR	KNX		ORL
15-Jul	Saturday	HNT				MOB		BIR	KNX		ORL
16-Jul	Sunday	HNT	OFF	OFF		OFF	OFF	OFF	OFF		ORL
17-Jul	Monday			MOB	JAX	WTN		CAR		KNX	
18-Jul	Tuesday			MOB	JAX	WTN		CAR		KNX	
19-Jul	Wednesday			MOB	JAX	WTN		CAR		KNX	
20-Jul	Thursday			MOB	JAX	WTN		CAR		KNX	
21-Jul	Friday	GRN	OFF		OFF	JAX	OFF			OFF	BIR
22-Jul	Saturday	GRN			ORL	JAX	MOB				BIR
23-Jul	Sunday	GRN			ORL	JAX	MOB				BIR
24-Jul	Monday	GRN			ORL	JAX	MOB				BIR
25-Jul	Tuesday	OFF		OFF	ORL	OFF	MOB	OFF	OFF		OFF
26-Jul	Wednesday		CHT		WTN		CAR		GRN	BIR	
27-Jul	Thursday		CHT		WTN		CAR		GRN	BIR	
28-Jul	Friday		CHT		WTN		CAR		GRN	BIR	
29-Jul	Saturday		CHT		WTN		CAR		GRN	BIR	
30-Jul	Sunday		CAR	WTN		HNT	GRN		OFF	OFF	
31-Jul	Monday		CAR	WTN		HNT	GRN		ORL		
1-Aug	Tuesday		CAR	WTN		HNT	GRN		ORL		
2-Aug	Wednesday		CAR	WTN		HNT	GRN		ORL		
3-Aug	Thursday	CHT	BIR				OFF	HNT	ORL		OFF
4-Aug	Friday	CHT	BIR					HNT	KNX		ORL

Table A. 6. (continued)

Home team		WTN	MOB	BIR	HNT	CHT	KNX	GRN	JAX	ORL	CAR
5-Aug	Saturday	CHT	BIR					HNT	KNX		ORL
6-Aug	Sunday	CHT	BIR					HNT	KNX		ORL
7-Aug	Monday		OFF	WTN	OFF	OFF		OFF	KNX		ORL
8-Aug	Tuesday			WTN	CHT			JAX		MOB	KNX
9-Aug	Wednesday			WTN	CHT			JAX		MOB	KNX
10-Aug	Thursday			WTN	CHT			JAX		MOB	KNX
11-Aug	Friday	OFF		OFF	CHT			JAX		MOB	KNX
12-Aug	Saturday		OFF		OFF	BIR	WTN			GRN	JAX
13-Aug	Sunday				MOB	BIR	WTN			GRN	JAX
14-Aug	Monday				MOB	BIR	WTN			GRN	JAX
15-Aug	Tuesday				MOB	BIR	WTN			GRN	JAX
16-Aug	Wednesday	OFF		OFF	MOB		OFF	OFF	OFF	CHT	OFF
17-Aug	Thursday	CAR	GRN	JAX			HNT			CHT	
18-Aug	Friday	CAR	GRN	JAX			HNT			CHT	
19-Aug	Saturday	CAR	GRN	JAX			HNT			CHT	
20-Aug	Sunday	CAR	GRN	JAX		OFF	HNT			OFF	
21-Aug	Monday		WTN	HNT		CAR		KNX	ORL		
22-Aug	Tuesday		WTN	HNT		CAR		KNX	ORL		
23-Aug	Wednesday		WTN	HNT		CAR		KNX	ORL		
24-Aug	Thursday		WTN	HNT		CAR		KNX	ORL		
25-Aug	Friday		HNT			BIR	CAR	JAX		WTN	
26-Aug	Saturday		HNT			BIR	CAR	JAX		WTN	
27-Aug	Sunday		HNT			BIR	CAR	JAX		WTN	
28-Aug	Monday		HNT			BIR	CAR	JAX		WTN	
29-Aug	Tuesday	OFF	OFF		BIR	OFF	JAX	OFF		OFF	OFF
30-Aug	Wednesday	MOB			BIR		JAX	CHT			ORL
31-Aug	Thursday	MOB			BIR		JAX	CHT			ORL
1-Sep	Friday	MOB			BIR		JAX	CHT			ORL
2-Sep	Saturday	MOB		OFF	OFF		OFF	CHT	OFF		ORL
3-Sep	Sunday	HNT		MOB			CHT			JAX	GRN
4-Sep	Monday	HNT		MOB			CHT			JAX	GRN

The constraints that are not satisfied

Team WTN does not have day off for more than 30 days
Team HNT does not have day off for more than 30 days
Team CHT does not have day off for more than 30 days
Team WTN travels 687 miles without day off in the first half
Team MOB travels 565 miles without day off in the first half
Team MOB travels 517 miles without day off in the second half
Team BIR travels 535 miles without day off in the second half
Team BIR travels 563 miles without day off in the second half
Team CHT travels 687 miles without day off in the first half
Team CHT travels 563 miles without day off in the second half
Team KNX travels 565 miles without day off in the first half
Team KNX travels 546 miles without day off in the first half
Team KNX travels 563 miles without day off in the second half
Team KNX travels 546 miles without day off in the second half
Team GRN travels 526 miles without day off in the first half
Team GRN travels 563 miles without day off in the first half
Team GRN travels 546 miles without day off in the second half
Team GRN travels 526 miles without day off in the second half
Team JAX travels 546 miles without day off in the first half
Team JAX travels 526 miles without day off in the first half
Team JAX travels 546 miles without day off in the first half
Team JAX travels 546 miles without day off in the first half
Team ORL travels 526 miles without day off in the first half
Team ORL travels 563 miles without day off in the first half
Team ORL travels 663 miles without day off in the first half
Team ORL travels 563 miles without day off in the first half
Team ORL travels 526 miles without day off in the second half
Team ORL travels 610 miles without day off in the second half
Team ORL travels 660 miles without day off in the second half
Team ORL travels 610 miles without day off in the second half
Team ORL travels 610 miles without day off in the second half
Team CAR travels 535 miles without day off in the first half
Team CAR travels 526 miles without day off in the first half
Team CAR travels 610 miles without day off in the first half
Team CAR travels 610 miles without day off in the first half
Team CAR travels 517 miles without day off in the second half
WTN has to have different schedule before and after the All Star break
CAR has to have different schedule before and after the All Star break
Weekend games for BIR do not met
Weekend games for HNT do not met
Weekend games for JAX do not met
Weekend games for ORL do not met

Weekend games for CAR has not met
WTN needs more home games in May
BIR wants road games on the third May weekend
BIR dislikes home game on Monday
HNT likes to open in Florida
HNT prefers home game in the first week of May
CHT likes to open on the road
KNX likes to close on the road
GRN doesn't want home game on Easter Sunday
GRN wants fewer home games on weekend in April and May
JAX likes to open at home
JAX wants more home games on weekend in April and May
JAX doesn't want home games in August and September

APPENDIX B. ASSIGNING TEAMS TO COLUMN 16 AND 17 IN THE GAME SCHEDULE IN THE SOUTHERN LEAGUE SCHEDULE

Column 17 is automatically determined if column 16 is scheduled. Assigning teams in column 16 is the same as the matching problem or assignment problem. We formulate the problem using 0-1 integer program method and provide combinatorial algorithm that finds the matching between teams easily.

0 -1 Integer Programming

$$\text{Let } X_{ij} = \begin{cases} 1 & \textit{if team i teavels to team j or vice versa} \\ 0 & \textit{otherwise} \end{cases},$$

$$C_{ij} = \begin{cases} 1 & \textit{j is admissible from i} \\ M & \textit{otherwise} \end{cases},$$

and Z is the total cost. The admissible team j from i is the team that does not have a game with team i between column 1 to 15. For example, Team 1 and 10 are admissible from team 1 (see Table 21 on Page 63 in PART IV). M is imaginary large number. Therefore, the objective function is

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}X_{ij}$$

where,

$$\sum_{i=1}^{n} X_{ij} = 1,\ \sum_{j=1}^{n} X_{ij} = 1, \text{ and } X_{ij} \geq 0.$$

Combinatorial Algorithm

To find the matching in column 16 we only need to consider team 1 through 5 since the competing teams for team 6 through 10 are automatically determined if matching teams for team 1 to 5 are determined. For example, team 1 has a game with team 8 in column 16 also means that team 8 has game against team 1.

We will use nodes to represent teams and arcs to represent travel from team to team. Then we have 10 nodes and 10 arcs since each team has two admissible arcs as seen in Figure B. 1.

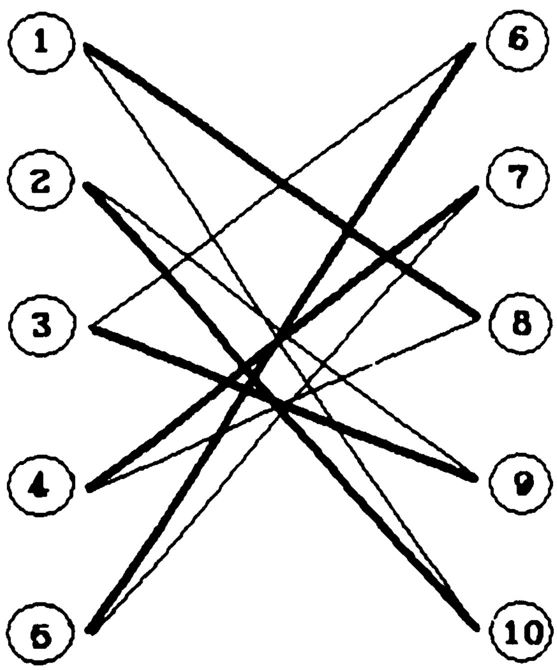

Figure B. 1. Finding matching between teams

Matching Algorithm

Step 1. **LIST1 = LIST2 = [ ]**. And start from node 1, choose any admissible team j and add 1 to **LIST1** and *j* to **LIST2**.

Step 2. If **LIST2** contains all nodes in right hand side, stop. We found matching.

Otherwise, continue.

Step 3. Find an admissible node *i* from node *j* that is not in the **LIST1**.

Add *i* into **LIST1**.

Step 3. Find an admissible node *j* from node *i* that is not in the **LIST2**.

Add *j* into **LIST2**. Go to Step 2.

At the end of the matching algorithm we have **LIST1** = [1, 4, 5, 3, 2] and **LIST2** = [8, 7, 6, 9, 10] from the example above. Finally, we assign games between **LIST1**(*i*) and **LSIT2**(*i*) where **LIST1**(*i*) represents i^{th} element of **LIST1**.

REFERENCES

Armstrong, J. and Willis, R. J. (1993). Scheduling the cricket world cup-A case study. J Opl Res Soc **44**: 1067-1072.

Ball, B. C. and Webster, D. B. (1977). Optimal scheduling for even-numbered team athletic conferences. AIIE Trans **9**: 161-169.

Bean, J. C. and Birge, J.R. (1980). Reducing traveling costs and player fatigue in the National Basketball Association. Interfaces **10**: 98-102.

Cain Jr., W. O. (1977). A computer assisted heuristic approach used to schedule the Major League Baseball clubs. In: S. P. Ladany and R. E. Machol (eds). Optimal Strategies in Sports. Amsterdam. North-Holland Pub.Co.: 32-41.

Campbell, R. T. and Chen, D.-S. (1976). A minimum distance basketball scheduling problem. In: R. E. Machol, S. P. Ladany, and D. G. Morrison (eds). Management Science in Sports. Amsterdam. North-Holland Pub. Co.: 15-25

De Werra, D. (1980). Geography, Games and Graphs. Discrete Appl Math **2**: 327-337.

De Werra, D. (1988). Some Models of Graphs for Scheduling Sports Competitions. Discrete Appl Math **21**: 47-65.

Ferland, J. A. and Fleurent, C. (1991). Computer aided scheduling for a sports league. INFOR **29**: 14-24.

Glover, F. (1989a). Tabu search: A tutorial. Interfaces **20**: 74-94.

Glover, F. (1989b). Tabu search – Part I. ORSA Journal on Computing **1**: 190-206.

Glover, F. (1990). Tabu search – Part II. ORSA Journal on Computing **2**: 4-32

Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. (1989). Optimization by simulated annealing: an experimental evaluation; part 1, graph partitioning. Ops Res **37**: 865-892.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. Sci **220**: 671-679.

Metropolis, W., Rosenbluth, A., Rosenbluth, M., Teller H., and Teller, E. (1953). Equation of state calculations by fast computing machines. J Chem Phys **21**: 1087-1092.

Nemhauser, G. L. and Trick, M. A. (1998). Scheduling a major college basketball conference. Ops Res **46**:1-8.

Russell, R. A. and Leung, J. M. (1994). Devising a cost effective schedule for a baseball league. Ops Res **42**: 614-625.

Schreuder, J. A. M. (1992). Combinatorial aspects of construction of competition Dutch Professional Football League. Disc Appl Math **35**: 301-312.

Willis, R. J. and Terrill, B. J. (1994). Scheduling the Australian State Cricket season using simulated annealing. J Opl Res Soc **45**: 276-280.

Wright, M. (1994). Timetabling county cricket fixtures using a form of tabu search. J Opl Res Soc **45**: 758-770.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Douglas D. Gemmill, for his guidance for Ph. D. He is very enthusiastic for helping me. I also thank to Dr. John Wacker, Dr. Sigurdur Olafsson, Dr. Timothy Van Voorhis, and Dr. Yoshinori Suzuki to be on my committee. I am very grateful to Dr. Jo Min for his kindness and guidance at Iowa State University.

Also, my special thanks are extended to Dr. Satish Udpa. I would not finish my Ph.D. without his research assistantship.

I am deeply appreciate to my parents, sister, brother, and brother-in-row. They believe in me even in hard times and support me always. Also, special thanks go to Toshiyuki Nakayame, Hareyo Nakayama, Konish Yoshiki, and Keiko Yoshiki. They are always very friendly.

Finally, my sincere gratitude goes to my wife, Yuki Nakayama, for her believe, patience, advice, and encouragement during my Ph. D. study. She brought an angel, Marie, to our family.