ABORDAGENS BASEADAS EM COLORAÇÃO DE ARESTAS PARA PROBLEMAS DE PROGRAMAÇÃO DE TABELAS ESPORTIVAS

TIAGO DE OLIVEIRA JANUARIO

ABORDAGENS BASEADAS EM COLORAÇÃO DE ARESTAS PARA PROBLEMAS DE PROGRAMAÇÃO DE TABELAS ESPORTIVAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Sebastián Alberto Urrutia

Belo Horizonte
Agosto de 2015

TIAGO DE OLIVEIRA JANUARIO

EDGE COLORING APPROACHES TO ROUND-ROBIN TOURNAMENT SCHEDULING PROBLEMS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Sebastián Alberto Urrutia

Belo Horizonte
August 2015

Januario, Tiago de Oliveira

J35a Abordagens baseadas em coloração de arestas para problemas de programação de tabelas esportivas / Tiago de Oliveira Januario. — Belo Horizonte, 2015
xx, 69 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: Sebastián Alberto Urrutia

1. Computação — Teses. 2. Teoria dos Grafos — Teses. I. Orientador. II. Título.

CDU 519.6*62(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Edge Coloring approaches to round-robin tournament scheduling problems

TIAGO DE OLIVEIRA JANUARIO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Sebastián Alberto Urrutia - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Antonio Alfredo Ferreira Loureiro
Departamento de Ciência da Computação - UFMG

Prof. Jayme Luiz Szwarcfiter
Programa de Engenharia de Sistemas e Computação - UFRJ

Prof. José Elias Claudio Arroyo
Departamento de Informática - UFV

Prof. Maurício Cardoso de Souza
Departamento de Engenharia de Produção - UFMG

Thiago F. Noronha
Prof. Thiago Ferreira de Noronha
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 12 de agosto de 2015.

Sempre em meus pensamentos e no meu coração
Ana e Antônio
Mariana e José
Meus queridos avós

Always in my thoughts and in my heart
Ana and Antônio
Mariana and José
My dear grandparents

Agradecimentos

Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES
Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais - PPGCC/UFMG
Microsoft Azure for Research

Acknowledgments

National Counsel of Technological and Scientific Development - CNPq
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES
Graduate Program in Computer Science at Universidade Federal de Minas Gerais - PPGCC/UFMG
Microsoft Azure for Research

"I believe in the ability of focusing strongly in something, then you are able to extract even more out of it. It's been like this all my life, and it's been only a question of improving it, and learning more and more and there is almost no end. As you go through you just keep finding more and more. It's very interesting, it's fascinating."

(Ayrton Senna)

Resumo

O planejamento de tabelas esportivas é um tema que vem ganhando espaço na área de pesquisa operacional. Métodos de buscas locais estão entre as técnicas mais eficazes para a construção de tabelas de jogos. Estudos prévios apontam a existência de não-conectividade nas estruturas de vizinhanças utilizadas na programação de torneios com rodadas simples. Tal característica afeta o desempenho de buscas locais que fazem uso dessas vizinhanças.

É sabido que grafos são importantes ferramentas matemáticas usadas na modelagem problemas reais. Uma contribuição desta tese é justamente a modelagem das estruturas de vizinhanças para o problema de programação de tabelas esportivas por meio de coloração de arestas.

Esta tese apresenta evidências empíricas e teóricas que relacionam a conectividade de estruturas de vizinhanças com a técnica de embaralhamento perfeito de cartas. Prova-se que, quando a permutação em um embaralhamento possui uma órbita permutacional de tamanho $(n-2)$, uma determinada vizinhança analisada é desconexa, assim as buscas locais baseadas em tal vizinhança não são capazes de obter tabelas de jogos que não sejam isomórficas à tabela inicial. Um novo método construtivo baseado na técnica de embaralhamento perfeito de cartas é apresentado. O método proposto é usado na análise de conectividade de uma das vizinhança estudadas.

Por fim, uma nova estrutura de vizinhança para o problema de programação de tabelas esportivas em torneios com rodadas é proposta. Essa estrutura de vizinhança é descrita em termos de coloração de arestas e a sua corretude é provada. Mostramos que a nova estrutura de vizinhança aumenta a conectividade do espaço de soluções. O seu desempenho é avaliado usando o Problema de Torneio com Viagens com Estádios Predefinidos e o Weighted Carry-Over Effects Value Minimization Problem como estudos de caso.

Palavras-chave: Teoria dos Grafos, Coloração de Arestas, Pesquisa Operacional em Esportes, Programação de tabelas esportivas, Busca local, Estruturas de vizinhanças.

Abstract

Sport scheduling is a trending research topic in operations research. Local search heuristics are among the most effective methods to construct schedules. Prior studies have noted the existance of non-connectivity in neighborhoods used in local search heuristics for single round-robin tournament scheduling. This non-connectivity affects the performance of local search heuristics.

It is well-known that graphs are useful tools for modeling relevant parts of reality. A contribution from this thesis is the modeling of the existing neighborhood structures in edge coloring terms.

This thesis presents empirical and theoretical evidences relating the non-connectivity of the neighborhood structures with the perfect riffle shuffle permutations of a deck of playing cards (faro shuffle). It is proved that, when the faro shuffle permutation has an $(n-2)$-cycle, the neighborhood is not connected and it does not allow the search to escape from schedules isomorphic to the initial one.

A new constructive method based on the faro shuffle of playing cards (faro method) is presented. The faro method is used in the analysis of neighborhood connectivity of one of the existing neighborhood structures.

This thesis introduces a novel neighborhood structure for single round-robin sport scheduling problems. The neighborhood structure is described in edge coloring terms and its correctness is proven. We show that the new neighborhood structure increases the connectivity of the solution space and evaluate its performance using the Traveling Tournament Problem with Predefined Venues and the Weighted Carry-Over Effects Value Minimization Problem as case studies.

Keywords: Graph Theory, Edge Coloring, Operations Research in Sports, Round-robin tournament scheduling, Local Search, Neighborhood structures

List of Figures

List of Tables

List of Acronyms

COEV	*Carry-over effects value*
HAP	*Home-away Pattern*
ILS	*Iterated Local Search*
MTTP	*Mirrored Traveling Tournament Problem*
P1F	*Perfect One-Factorization*
PRS	*Partial Round Swap*
PTS	*Partial Team Swap*
RS	*Round Swap*
SRR	*Single Round-Robin*
TARS	*Teams and Rounds Swap*
TS	*Team Swap*
TSP	*Traveling Salesman Problem*
TTP	*Traveling Tournament Problem*
TTPPV	*Traveling Tournament Problem with Predefined Venues*
WCOEVMP	*Weighted Carry-over Effects Value Minimization Problem*

Contents

Chapter 1

Introduction

Sports competitions attract worldwide attention. *The Rites of Men: Manhood, Politics, and the Culture of Sport* by [Burstyn, 1999] states: “The rituals of sport engage more people in a shared experience than any other institution of cultural activity”. This chapter begins with the motivation of this work. Next, we define some concepts in graph theory and sport scheduling that will be used throughout the text. Finally, we present the objectives of this work and the organization of the text.

1.1 Motivation

Several optimization techniques have been applied to solve problems arising from sports scheduling. The difficulty of these problems leads to the use of a number of exact and approximate approaches, including integer programming, constraint programming, metaheuristics, and hybrid methods [Kendall et al., 2010].

Round-robin scheduling problems are very hard to solve even for small values of n, therefore many heuristics have been developed for these kinds of problems. Local search heuristics are among the most effective heuristic algorithms to solve round-robin scheduling problems as in Ribeiro and Urrutia [2007], Anagnostopoulos et al. [2006] and Di Gaspero and Schaerf [2007].

The performance of local search algorithms crucially depends on structural aspects of the space being searched. Studying the nature of this dependency can significantly improve our understanding of local search behavior and facilitate the further improvement and successful application of local search methods.

Hoos and Stützle [2004] introduces various aspects of search space structure and discuss their impact on local search performance. These include fundamental properties of a given search space and neighborhood graph, such as size, connectivity, diameter and solution density, as well as global and local properties of the search landscapes encountered by local search algorithms, such as the number and distribution of local minima, fitness distance correlation, measures of ruggedness, and detailed information on the plateau and basin structure of the given space. Some of these search space features can be determined analytically, but most have to be measured empirically, often involving rather complex search methods.

Jhonson and McGeoch [1997] exemplifies an analysis of search space features and their impact on local search performance for the *Traveling Salesman Problem* (TSP). Among simple local search algorithms for the TSP, the most famous are 2-Opt and 3-Opt. The authors describe what is known theoretically about these algorithms in the worst and average case. They also present some experimental results that show that the algorithms perform much better in practice than the theoretical bounds might indicate, for a specific set of instances.

Han [2006] also considers the TSP as a case study in another analysis of local search algorithms. This time, the authors investigate the behavior of the 2-Opt heuristic. Empirical evidences show that when applying the 2-opt heuristic to the travelling salesman problem, selecting the best improvement at each iteration gives worse results on average than selecting the first improvement, if the initial solution is chosen at random.

Di Gaspero and Schaerf [2007] presents an investigation of the properties of the known neighborhood structures for the *Traveling Tournament Problem* (TTP). They determined that the intersection between the two most used neighborhood structures in the context of round-robin scheduling is not empty and proposed strategies to avoid redundant moves in local search algorithms.

In a recent paper Costa et al. [2012] an *Iterated Local Search* (ILS) was proposed for the *Traveling Tournament Problem with Predefined Venues* (TTPPV). During the analysis of their computational results, it was noted that the used neighborhood structures were not connected for some instance sizes, entrapping the local search procedures in a small portion of the solution space. The motivation behind this thesis started from the study about the issue of neighborhood non-connectivity noted in Costa et al. [2012].

This thesis considers concepts of graph theory to formulate and solve problems faced by tournament organizers. Its first contribution is the modeling of the existing neighborhood structures in edge coloring terms. Traditional models for constructing sports schedules are based on graphs. It is well-known that graphs are useful tools for modeling relevant parts of reality. Therefore, the research area of this thesis will be bounded by round-robin sport scheduling problems related to graph theory focused on edge coloring problems.

During the investigation of the neighborhood non-connectivity noted in Costa et al. [2012], it was noted that part of this issue can be explained by a classical result in graph theory, related to perfect one-factorizations of complete graphs. The research was extended and an empirical study revealed a previously unknown relation between the neighborhood non-connectivity and the perfect riffle shuffle permutation of a deck of playing cards (faro shuffle).

This thesis contributes with a formal relation between the connectivity of the solution space for the studied neighborhood structures and the faro shuffle of playing cards. According to [Goossens and Spieksma, 2012b] the circle method is the most used constructive method in the literature. Here, a new constructive method based on the faro shuffle of playing cards (faro method) is presented and its equivalence with the circle method is proved. The faro method is used in the analysis of neighborhood connectivity of one of the existing neighborhood structures.

In this thesis, a novel neighborhood structure for round-robin scheduling problems to be used by local search procedures is proposed: the *Teams and Rounds Swap* (TARS). Using two problems from the literature as case studies, it is shown that the proposed neighborhood structure may circumvent the issue of non-connectivity in the existing ones. Moreover, the use of TARS lead to good results for both case study problems regardless the initial solution or the problem instance.

1.2 Elements of sport scheduling

A round-robin tournament is a competition involving n different teams indexed by $t \in T = \{0, \dots, n-1\}$, in which T is the set of teams. The games of the competition must be scheduled in a number r of rounds in such a way that each team plays one game at each round, and each team is required to play against all other teams exactly m times. Since the number of teams participating in a tournament is usually even, we assume that n is even.

Single Round-Robin (SRR) tournament is a competition in which all teams face each other once, there are $^{n}/_{2}$ games in each round and each team plays exactly once in each round. A common way to describe a solution of an SRR scheduling problem is by using a *timetable*. Usually, a timetable is described by an opponent schedule combined with a *Home-away Pattern* (HAP) set. An opponent schedule may be represented by an $n \times (n-1)$-matrix of opponents $O = [o_{t,r}]$ where each entry $o_{t,r} \in \{0, \dots, n-1\} \setminus \{t\}$ specifies the opponent of team t in round r. The HAP set determines in which conditions (home or away) each team plays at each round. It is defined as an $n \times (n-1)$-matrix $H = [h_{t,r}]$, where $h_{t,r} = +$ (resp. $h_{t,r} = -$) when team t has a home (resp. away) game in round r. The opponent schedule and the home-away pattern set together determine the tournament schedule, as seen on Table 1.1.

teams	rounds				
	0	1	2	3	4
0	−4	−5	+3	−1	−2
1	−3	+4	−2	+0	−5
2	−5	+3	+1	−4	+0
3	+1	−2	−0	−5	+4
4	+0	−1	+5	+2	−3
5	+2	+0	−4	+3	+1

Table 1.1: A tournament schedule for 6 teams comprising 5 rounds.

Concerning the geographical position and the distance traveled by the teams, we assume that each team is associated with a home venue located in a city and the distances between the venues are associated with an already known $n \times n$ distance matrix $D = [d_{t_1,t_2}]$. It is considered that each team starts the tournament at its home venue. At the beginning of the tournament, if the team plays its first game away from home, it must travel from its own venue to the opponent's venue. Whenever a team plays two consecutive away games, it goes directly from the city of the first opponent to the other without returning to its own home city. At the end of the tournament, if the team played its last match away from home, it must return to its home venue.

The cost associated with a team in a given schedule is the sum of all distances it has to travel between each game according to the schedule. The cost of a solution is defined as the sum of the costs of all teams in the tournament. For instance, according to the scheduled on Table 1.1, team 0 has to successively play against teams 4 away, team 5 away, team 3 at home, team 1 away and finally team 2 away. Accordingly, the travel cost of team 0 would be $d_{0,4} + d_{4,5} + d_{5,0} + d_{0,1} + d_{1,2} + d_{2,0}$.

It is said that there is a break when a team plays two consecutive home games or two consecutive away games. A road trip is a sequence of consecutive away games for a team, while a home stand is a sequence of consecutive home games. Breaks are convenient because in a large country (e.g. USA, Chile and Brazil), where the distances between the home cities can be very large, it is advantageous to organize a tour for a number of consecutive away games.

1.3 Elements of graph theory

Graph theory, a major branch of mathematics, has been studied intensively for hundreds of years. Many important and useful properties of graphs have been discovered, many important algorithms were developed, and many difficult problems are still actively being studied [Sedgewick and Wayne, 2011]. This section follows [Diestel, 2005; Gabow et al., 1985; Misra and Gries, 1992] and introduces a variety of fundamental graph definition that are important for a better understanding of this work.

A graph is a pair $G = (V, E)$ in which the n elements of V are the vertices of G and the m elements of E are its edges. Each edge $e \in E$ is composed of a pair of vertices $v, w \in V$, usually called its end vertices, endpoints or ends. The edge $e = (v, w)$ is said to be incident to v and w. An adjacent vertex of a vertex v in a graph is a vertex that is connected to v by an edge. Two edges of a graph are called adjacent if they share a common vertex. The degree (or valency) of a vertex of a graph is the number of edges incident to the vertex. The degre of a graph, denoted by Δ, is given by its vertex with maximum degree. If $V' \subseteq V$ and $E' \subseteq E$, then $G' = (V', E')$ is a subgraph of G (and G is a supergraph of G'), written as $G' \subseteq G$. If $G' \subseteq G$ and $G' \neq G$, then G' is a proper subgraph of G. If $G' \subseteq G$ and G' contains all the edges $(v, w) \in E$ with $v, w \in V'$, then G' is an induced subgraph of G. All graphs in this work are complete, undirected, finite and without multiple edges or self-loops.

Edge coloring is an important problem in graph theory. It consists in coloring the edges of a graph such that every two edges incident to the same vertex get different colors. An assignment of colors to the edges of a graph $G = (V, E)$ is a map $c : E \to C$, where C is a non-empty set of colors. Let $\mathcal{C}(v, w)$ be the color of the edge (v, w) and let $adj(v, c)$ be the vertex w such that $\mathcal{C}(v, w) = c$. A color c is incident to a vertex v if any edge incident to v has that color, otherwise, the color is free or absent on that vertex. A conflict in an assignment of colors is the existence of adjacent edges with the same color.

A proper edge coloring of G is an assignment of colors to every edge of G without conflicts and a proper partial edge coloring is an assignment of colors without conflicts such that some edges of G may be uncolored. A proper edge coloring is said to be minimum if the cardinality of C is minimum.

A k-coloring is a proper edge coloring using at most k colors. The minimum number of colors necessary to obtain a proper edge coloring is called the *chromatic index* $\chi'(G)$, so that $\chi'(G) = min\{k | \exists k\text{-coloring of } G\}$. Let Δ be the degree of the graph G. It is easy to see that $\chi'(G) \geq \Delta$ because different colors have to be assigned to all edges incident to a vertex with the maximum degree. Vizing's theorem proves, in a constructive way, that every simple graph can be colored, in polynomial time, with no more than $\Delta+1$ colors [Vizing, 1964]. The proof immediately yields an $O(|E| \cdot |V|)$ time edge coloring algorithm.

A simple graph G is said to be in class 1 if $\chi'(G) = \Delta$ and in class 2 if $\chi'(G) = \Delta + 1$. Vizing's theorem states that there are no other possibilities: all graphs are either in class 1 or class 2. However, it is $\mathcal{NP}$-Complete to determine whether a graph is in class 1 or in class 2 [Holyer, 1981].

Two structures, introduced by [Misra and Gries, 1992], called $cd - path$ and fan will be used in this work. A $cd - path$ is a path in which its edges are alternately colored with colors c and d. A closed $cd - path$ is called a $cd - cycle$.

A fan at v is an ordered sequence of distinct edges $fan(v) = \{(v, w_1), \ldots, (v, w_s)\}$, in which $s = |fan(v)|$. We say that v is the center of $fan(v)$ and w_i are its leaves, for $i \in \{1, ..., s\}$. A shift of $fan(v)$ means to circularly shift the colors of its edges. In a shift forward, for $1 \leq i < s$ and $j = i+1$, (v, w_j) gets the color of (v, w_i) and (v, w_1) gets the color of (v, w_s). In a shift backward, for $1 \leq i < s$ and $j = i + 1$, (v, w_i) gets the color of (v, w_j) and (v, w_s) gets the color of (v, w_1). Given a valid coloring, a shifting of $fan(v)$ yields another coloring of G with the same set of colors.

A one-factor of graph $G = (V, E)$ is a set of edges $R \subseteq E$, such that all vertices in V have degree equal to 1 in the subgraph $G' = (V, R)$ (R is also called a perfect matching). A one-factorization of $G = (V, E)$ is a partition of E into one-factors, i.e., a set $R = \{f_1, f_2, \ldots, f_\gamma\}$ of disjoint one-factors such that $\bigcup_{i=1}^{\gamma} f_i = E$.

A one-factorization is said to be perfect, called *Perfect One-Factorization* (P1F), when the graph induced by the edges in $f_i \cup f_j$ ($G = (V, f_i \cup f_j)$) is a hamiltonian cycle for each pair of distinct one-factors f_i and f_j, wherein a hamiltonian cycle is a cycle through a graph that visits each vertex exactly once. Figure 1.1 gives an example of a P1F for a complete graph with 4 vertices.

1.4 Edge coloring for sport scheduling

To model an SRR tournament, we label the teams by $\{v_1, v_2, ..., v_n\}$, and represent the tournament by a complete graph, where the vertices represent the n teams, and each edge $e = (v_i, v_j)$ represents the match in which teams i and j play against each other. Hence, the schedule of an SRR tournament can be seen as an edge-coloring of a complete graph with n vertices. In the proposed model, a schedule is a partition of the edge set of the graph into rounds.

As noted by [de Werra, 1981], a one-factor represents a round, therefore schedules of SRR tournaments have a one to one relation with ordered one-factorizations of complete graphs K_n. Figure 1.1 gives an SRR schedule represented by a one-factorization of K_4. If we consider the relation between the factors and rounds $R = \{(\textit{dashed line}, \textit{round}\ 1), (\textit{solid line}, \textit{round}\ 2), (\textit{double line}, \textit{round}\ 3)\}$, the matches in the first round, for example, will be team 1 against team 4 and team 2 against team 3.

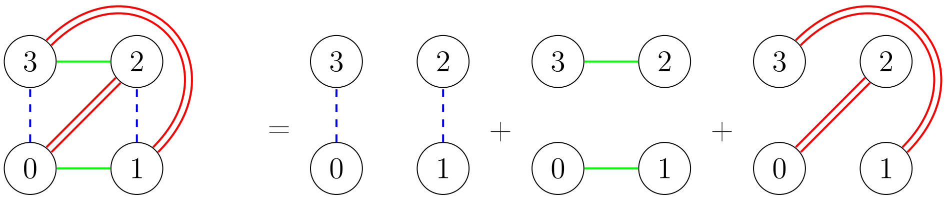


Figure 1.1: An example of a tournament represented by a one-factorization of K_4.

Two one-factorizations $F = \{f_1, f_2, \ldots, f_k\}$ and $J = \{j_1, j_2, \ldots, j_k\}$ of a graph $G = (V, E)$ are called *isomorphic* if there is a bijection $\varphi : V \to V$ such that $j_i = \{(\varphi(x), \varphi(y)) : (x, y) \in f_i\}$ for each $i = 1, \ldots, k$ [Dinitz et al., 1994].

The concept of isomorphic one-factorizations can also be extended to isomorphic schedules and isomorphic colorings, therefore two schedules (resp. colorings) are said to be isomorphic if, and only if, their associated one-factorizations are isomorphic. Note that one-factorization and minimum proper edge coloring are interchangeable concepts for complete graphs with an even number of vertices, e.g., a one-factor f_c represents the edge-set wherein all edges are colored with color c.

From now on, teams will be represented by vertices of K_n, matches will be represented by edges and rounds will be represented by the colors of the edges. For instance, if an edge $e = (v, w)$ is colored by a color c, then the game between teams t_v and t_w is played in round $r = c$.

1.5 Objectives

It was noted in [Costa et al., 2012] that the solution space defined by some commonly used neighborhoods in round-robin sport scheduling literature are not connected in the case of SRR tournaments, which explains the hardness of finding high-quality solutions to some problem instances. They showed that initial solutions obtained by the circle method should not be used in some situations. As an alternative, the vizing algorithm led to considerably better initial solutions as the authors were able to obtain better results than the ones published by [Melo et al., 2009].

The main objective of this thesis is to investigate the connectivity of neighborhood structures applied to SRR scheduling problems making use of graph theory concepts, focused on edge coloring of complete graphs. Hereof, this thesis establishes a formal relation between the most used method to construct initial solutions to tackle SRR scheduling problems with the connectivity of the solution space for the studied neighborhood structures. When constructing an initial solution with the circle method and for specific numbers of participating teams, the existing neighborhoods are not connected and local search procedures stay trapped in a tiny portion of the solution space. In consequence, search procedures on those neighborhoods are not likely to find good schedules for the tournament.

This thesis presents empirical and theoretical evidences relating the non-connectivity of the neighborhood structures with the perfect riffle shuffle permutations of a deck of playing cards (faro shuffle). It is proved that, when the faro shuffle permutation has an $(n-2)$-cycle, the neighborhood is not connected and it does not allow the search to escape from schedules isomorphic to the initial one. A new constructive method based on the faro shuffle of playing cards (faro method) is presented. The faro method is used in the analysis of neighborhood connectivity of one of the studied neighborhood structures.

Hereof, a novel neighborhood structure for SRR sport scheduling problems to be used by local search procedures is proposed: the *Teams and Rounds Swap* (TARS). Using the *Traveling Tournament Problem with Predefined Venues* (TTPPV) and the *Weighted Carry-over Effects Value Minimization Problem* (WCOEVMP) as case studies, experimental results show that this neighborhood structure diminishes the issue of non-connectivity in SRR scheduling problems. Moreover, the use of TARS can lead to good results for both case study problems regardless the initial solution or the problem instance.

1.6 Text organization

This thesis is organized as follows. Chapter 2 describes the circle method, the vizing algorithm, and a new method for constructing SRR schedules based on faro shuffle permutation of playing cards. Such method, called faro method is proved to be equivalent to the circle method. Given a first non-empirical relation between the circle method and the faro shuffle of playing cards, chapter 3 describes the existing neighborhood structures for round-robin sport scheduling problems and analyze their connectivity. We give a theoretical proof characterizing the situations in which the existing neighborhood structures are disconnected. A detailed description of the proposed TARS neighborhood structure followed by the proof of its correctness is given in Chapter 4, followed by a discussion about how TARS increases the solution space connectivity. An experimental analysis of the TARS neighborhood structure is performed in Chapter 5 using the TTPPV and the WCOEVMP as case studies. Empirical results show that the theoretical findings of this thesis have implications in the performance of local search heuristics under consideration. The last chapter presents some concluding remarks and describes some future works that can be produced as outcomes of this thesis.

Chapter 2

Building an SRR Schedule from Scratch

Initial solutions for round-robin sport scheduling problems can be obtained with edge coloring algorithms applied to complete graphs as the circle method [de Werra, 1981] and the vizing algorithm [Vizing, 1964; Misra and Gries, 1992]. This chapter begins with the description of those two edge coloring methods. Next, we introduce a constructive method based on the faro shuffle of playing cards and show its equivalence with the circle method. The faro method is used to analyze the connectivity of the existing neighborhood structures for round-robin scheduling problems given a first non-empirical relation between the circle method and the faro shuffle of playing cards.

2.1 The circle method

The circle method has been widely used to construct initial solutions for round-robin tournaments. The one-factorization generated by the circle method is called canonical coloring. According to [Goossens and Spieksma, 2012b], this method is at least a century old, can be found in most textbooks on graph theory, and it is the most used method in practice.

Denote the set of n teams by $T = \{0, 1, ..., n-1\}$ and the set of $n-1$ rounds by $R = \{0, 1, ..., n-2\}$. To schedule a tournament, first draw a regular $(n-1)$-sided polygon. Each vertex of the polygon is numbered from 0 to $n-2$ and an extra vertex $n-1$ is placed in the middle of the polygon. See the initial setup of the circle method on the top leftmost side of Figure 2.1.

At each iteration i, starting from iteration 0 to iteration $n-2$, in order to determine ${}^{n}/_{2}$ games at the i-th round of the tournament, a straight line is drawn from the

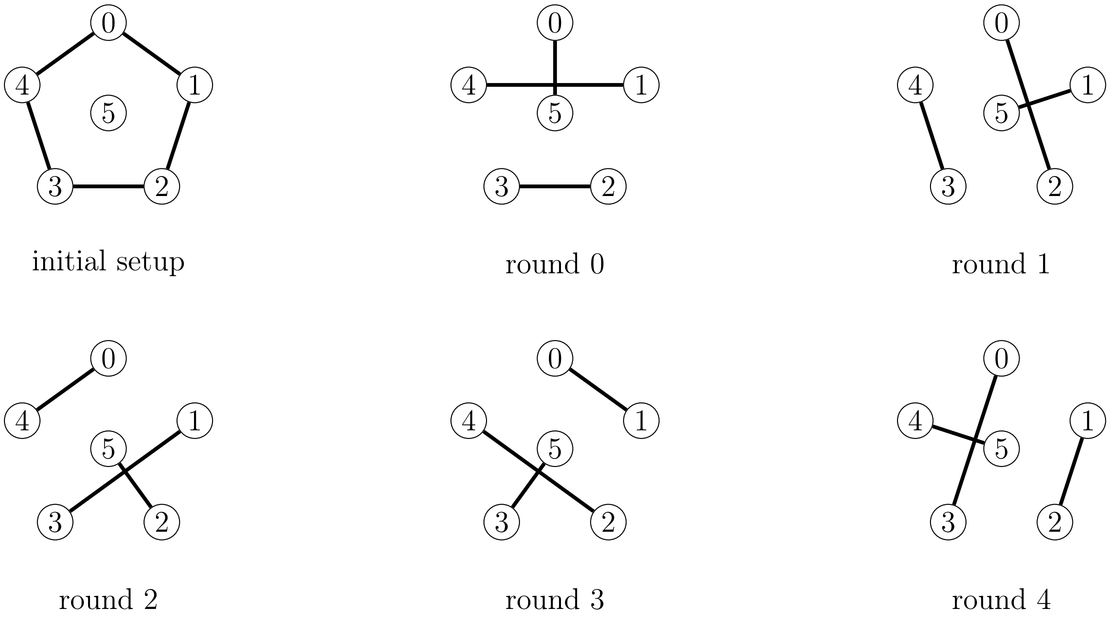


Figure 2.1: Circle method for a tournament with $n = 6$ teams.

vertex $n-1$ to the vertex i of the polygon, together with all possible straight lines that lie perpendicular to $(i, n-1)$. All matches that can be obtained this way determine one round of the tournament. In the next iteration, the lines are rotated $^1/_{(n-1)}$-th of a circle (i.e. one vertex point) in clockwise direction. All new straight lines represent the games for the new round. The procedure is repeated until all rounds are defined.

A description of the circle method based on [Miyashiro and Matsui, 2006] is given as follows:

In round $r \in R = \{0, 1, \dots, n-2\}$,

- team $t \in T \setminus \{n-1\}$ plays against team $t' \in T \setminus t$ for $(t + t') \equiv 2r \mod (n-1)$.
- team $n-1$ plays against team r;

Based on the previous definition, we can represent the circle method by the following opponent schedule function:

$$\Upsilon(t, r) = \begin{cases} r & \text{if } t = n-1 \\ n-1 & \text{if } t = r \\ (2r - t) \mod (n-1) & \text{otherwise} \end{cases} \tag{2.1}$$

Function $\Upsilon(t, r)$ computes the opponent of team $t \in T$ in round $r \in R$ based on the circle method. Table 2.1 shows a schedule constructed with the circle method for a tournament with $n = 6$ teams.

		r				
		0	1	2	3	4
t	0	5	2	4	1	3
	1	4	5	3	0	2
	2	3	0	5	4	1
	3	2	4	1	5	0
	4	1	3	0	2	5
	5	0	1	2	3	4

Table 2.1: Schedule constructed by the circle method for a tournament with $n = 6$ teams.

2.2 An algorithm based on Vizing's theorem

The following result on edge coloring was published by [Vizing, 1964]:

Theorem 1. *All graphs are either in class 1 or class 2, i.e.,* $\Delta \leq \chi'(G) \leq \Delta + 1$.

There are several proofs of Vizing's theorem in the literature, see for example [Diestel, 2005; Dijkstra and Rao, 1990; Gabow et al., 1985; Gould, 1988; Misra and Gries, 1992]. All proofs are based on augmenting the coloring of the graph by assigning, at each iteration, a new edge with color $c \in \{c_1, c_2, \ldots c_{\Delta+1}\}$. The proofs show how to color an uncolored edge of an arbitrary partially colored graph, which may require changing the color of some edges to maintain validity, never exceeding $\Delta + 1$ different colors. This procedure is repeated until all edges are colored. The following proof may be found in [Gould, 1988].

Let $e_0 = (w, v_0)$ be an uncolored edge of a graph G partially colored with no more than $\Delta + 1$ colors. Observe that, since both w and v_0 have at most $\Delta - 1$ incident colored edges, there are at least two available colors in both of them. Let us call $Free(v)$ the set of colors available at vertex v.

If $Free(w) \cap Free(v_0) \neq \varnothing$ we can simply choose one color from that intersection to color the edge e_0.

Assume now, that the intersection is empty. Let c be an available color at v_0 and let d be an available color at w. Let $cd - path$ be the maximal path that starts at v_0 in such a manner its edges are alternately colored with colors c and d. Two cases may arise: $cd - path$ ends at a vertex different from w or $cd - path$ ends at w. Whenever the first case happens, the coloring may be augmented. First, exchange the colors of the edges on the $cd - path$. Edges colored with c are recolored with d and vice-versa. Finally, color the edge e_0 with d, which in turn is available at vertex v_0 (see Fig. 2.2).

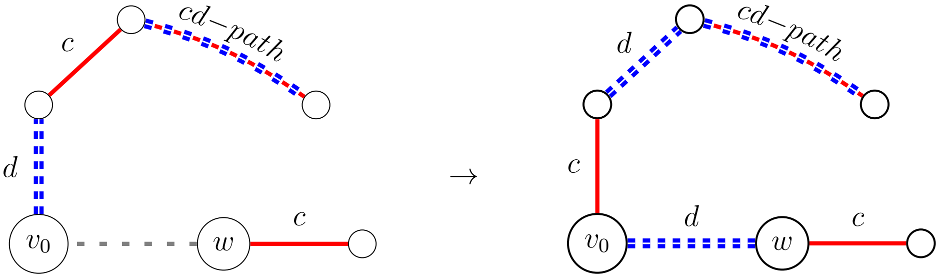


Figure 2.2: Color d is free at v_0 after the exchanging of colors.

Assume now that the $cd-path$ ends at w. If we exchange the colors of the edges on the $cd-path$, then color d will be available in v_0 but, on the other hand, it will not be available in w (see Fig. 2.3).

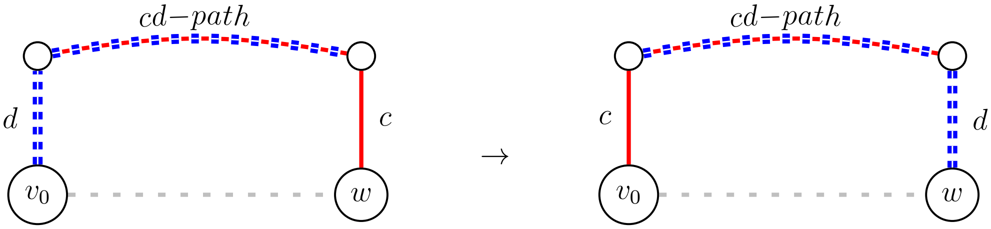


Figure 2.3: An example of the $cd-path$ ending at w.

Let v_1 be the vertex adjacent to w in the $cd-path$ and let $e_1 = (w, v_1)$ be an edge colored with c. Remove the color c from e_1 and color $e_0 = (w, v_0)$ with that color (see Fig. 2.4).

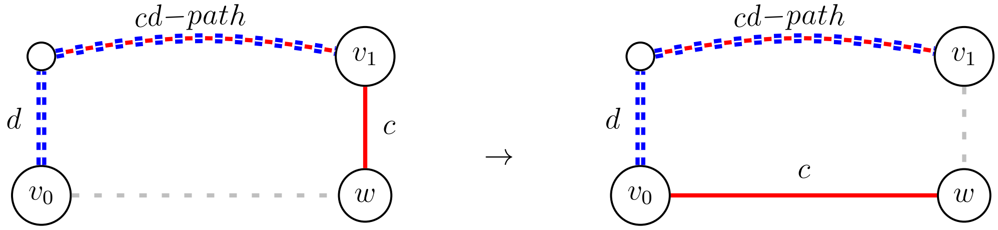


Figure 2.4: Color c moves from (w, v_1) to (w, v_0).

Now the problem consists in re-coloring edge e_1. The same sequence of steps that was used to color e_0 can also be applied in order to assign a color to e_1. Note that, if $Free(w) \cap Free(v_1)$ is also empty then the color c_1, available at v_1, must be different from c in order to avoid cycling. Such color always exists because v_1 has at most $\Delta - 1$ colored edges.

We may continue this way until $Free(w) \cap Free(v_i)$ is not empty or the $cd-path$ does not end at w. The remaining of the proof of Vizing's theorem shows that one of these cases will eventually occur after at most Δ iterations [Gould, 1988]. This proof of Vizing's theorem immediately yields an $O(|E| \cdot |V| \cdot \Delta)$ time algorithm to obtain a $\Delta + 1$ proper edge coloring of a simple graph.

In order to find an edge coloring of a complete graph K_n using $\Delta = n - 1$ colors, color the edges of a complete graph K_{n-1} using exactly $\Delta + 1 = n - 1$ colors. Note that there is one distinct color that is not used to color the edges incident to each of its vertices. Then, add a new vertex v_n to K_{n-1} and new edges $e_i = (v_i, v_n)$ for $1 \leq i \leq n - 1$ connecting each vertex v_i of K_{n-1} to v_n, thus building a complete graph with n vertices. Each new edge e_i is colored with the unique element of $Free(v_i)$, leading to an $(n-1)$-edge coloring of K_n. An example of how this procedure could be applied to obtain an edge coloring for K_4 is depicted in Figure 2.5.

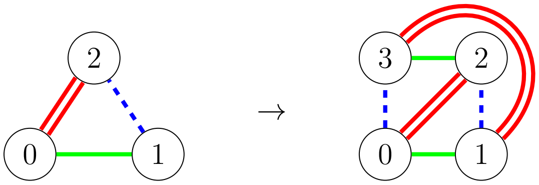


Figure 2.5: Finding an edge coloring of K_4 based on an edge coloring of K_3.

The edge coloring of K_n with $n-1$ colors constructed by Vizing algorithm depends on the order in which the yet uncolored edges are considered. In fact, one may get any possible edge coloring since there always exists an order for which Vizing algorithm produces a given edge coloring of K_n. This is clearly not the case for the circle method.

2.3 The faro method

Suppose you have a deck with an even number n of playing cards you want to shuffle. Take the deck, split it into two parts, and interleave the cards in each part of the deck, by dropping cards from the bottoms of the two half-decks. The top and bottom cards of the deck are always left unaltered. Such shuffling technique is known as riffle shuffle, see [Aigner and Ziegler, 2004]. The faro shuffle, an idealized riffle shuffle, is a term used to denote a perfect riffle shuffle performed in such a manner that the deck is split exactly in half and all its cards are perfectly alternated, see [Morris and Hartwig, 1976; Diaconis et al., 1983]. For a visual representation of a faro shuffle of a deck with 6 playing cards, see Figure 2.6.

We can mathematically model a deck of n playing cards by a sequence of integers $\{0, 1, \ldots, n-1\}$. The card at position $i = 0$ is the top card, and the card at position $i = n - 1$ is the bottom card. In order to perform a faro shuffle, a card at position $i \in$

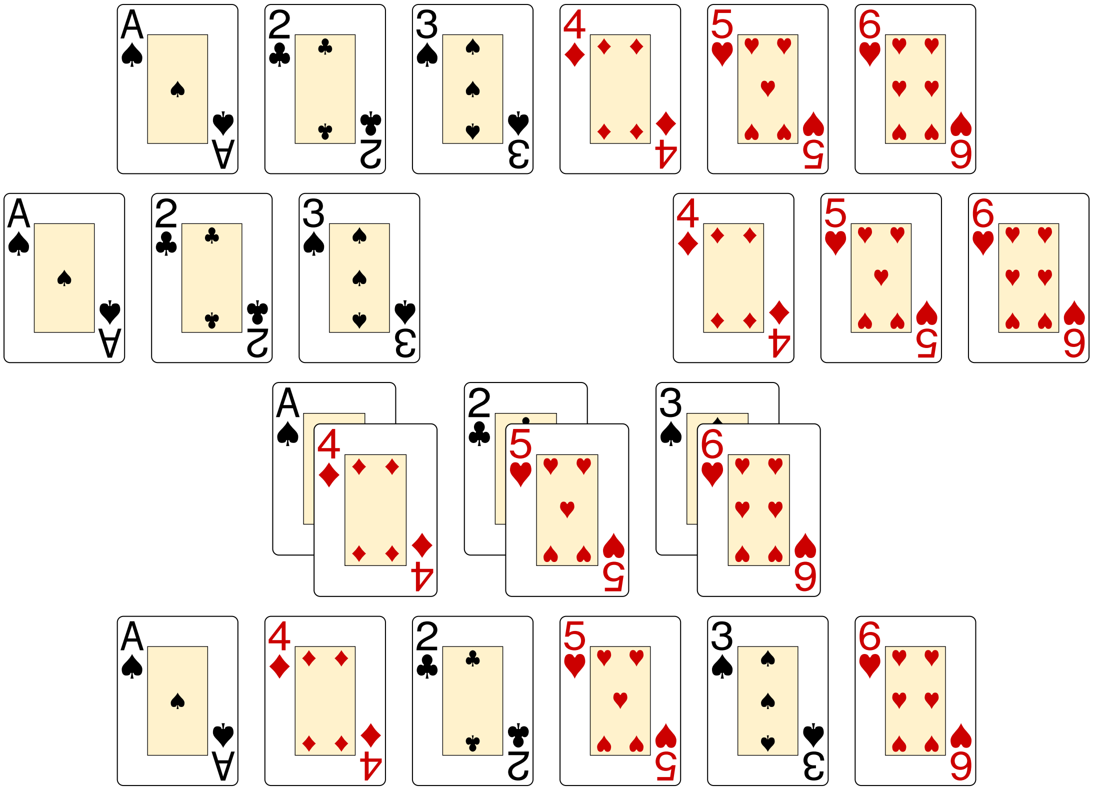


Figure 2.6: The faro shuffle. We begin with an ordered deck, and then we divide it into two packets of the same size and riffle them together. The last line of cards shows the resulting shuffled deck.

$\{0, \ldots, {}^{n}\!/\!{}_{2} - 1\}$ is moved to the position $2i$. If a card is at position $i \in \{{}^{n}\!/\!{}_{2}, \ldots, n-2\}$, then it is moved to the position $2i - (n-1) = 2i \mod (n-1)$ and the card at the position $n-1$ stays at its position. Since $2i < n-1$ for $i \in \{0, \ldots, {}^{n}\!/\!{}_{2} - 1\}$, we can represent a faro shuffle of a deck of n cards by the mapping

$$\Re(i) = \begin{cases} n-1 & \text{if } i = n-1 \\ 2i \mod (n-1) & \text{otherwise} \end{cases} \tag{2.2}$$

A concise mathematical way to think about changing orderings of the deck is given by permutations [Mann, 1995]. A permutation of n cards may be seen as a one-to-one map from the set of integers, between 0 and $n-1$ inclusive, to itself.

According to [Ellis et al., 2002], a faro shuffle permutation of cards can be described as a set of permutation cycles. A permutation cycle is a subset of a permutation whose elements trade places with one another. Permutation cycles are also called "orbits" by [Comtet, 1974]. For example, in a faro shuffle that transforms

$[0,1,2,3,4,5,6,7,8,9]$ into $[0,5,1,6,2,7,3,8,4,9]$, (0) is a 1-cycle, (3,6) is a 2-cycle, $(1,5,7,8,4,2)$ is a 6-cycle and (9) is a 1-cycle. Here, the notation (0)(3,6)(1,5,7,8,4,2)(9) means that, starting from the natural ordering of cards $\{0,1,2,3,4,5,6,7,8,9\}$, the first and last cards stay in their positions, cards at positions 3 and 6 are switched, card at position 1 is replaced by the card at position 5, card at position 5 is replaced by the card at position 7, and so on until the card at position 2 is replaced by card at position 1. The order of the cycles does not change the permutation cycle. Therefore (0)(3,6)(5,7,8,4,2,1)(9), (9)(0)(3,6)(7,8,4,2,1,5), (8,4,2,1,5,7)(9)(0)(3,6) and (9)(4,2,1,5,7,8)(0)(3,6) all describe the same permutation cycle. Note that a faro shuffle can be performed several times in a row, however, despite the cards changing positions, the permutation cycles will always be the same.

2.3.1 Scheduling a tournament with faro shuffles

In the following, a new method for constructing initial solutions for round-robin tournaments is presented. At each iteration i, starting from iteration 0 to iteration $n-2$, the faro method determines all opponents of team t, for $0 \leq t \leq n-1$. Consider two decks of n playing cards represented by an ordered set of integers $\{0,\ldots,n-1\}$. One deck will be used to determine the opponents of team t while the other deck will be used to identify the rounds in which t faces its opponents.

In order to determine the opponents of the team 0, take the first deck and arrange all its cards side-by-side on ascending order. Then, take the second deck and arrange all its cards as a faro shuffle of the first deck, keeping every card of the second deck right below every card of the first deck. Figure 2.7 shows an example of such arrangements of cards for decks of size $n=6$.

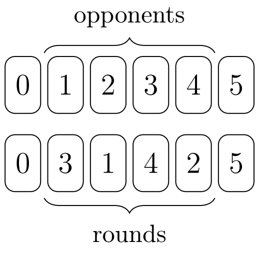


Figure 2.7: Scheduling a tournament with $n = 6$ teams based on the faro shuffle of playing cards. Here the opponents of team 0 are: team 1 on round 3, team 2 on round 1, team 3 on round 4, team 4 on round 2 and team 5 on round 0.

Note that the card 0, representing the team 0, is at position 0 and the card $n-1$, representing the team $n-1$, is at position $n-1$. For each position $i \in \{1,2,\ldots,n-2\}$, each card at the position i of the second row denotes a round $r = i$ and each card at the position i in the first row denotes the opponent of team 0 in round $r = i$. Team 0 faces team $n - 1$ at round $r = 0$.

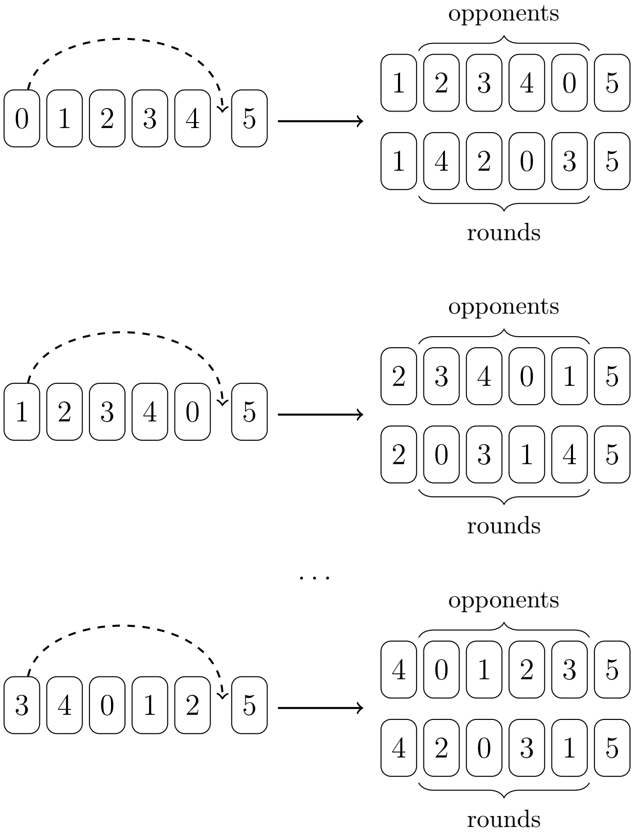


Figure 2.8: The faro method for scheduling an SRR tournament.

In order to determine the opponents of the team t, for $0 < t \leq n - 2$, take the deck of the first row on previous iteration and rotate it, except for the last card, by placing the card from the position 0 between the cards at positions $n - 2$ and $n - 1$. After the rotation, the card that represents the team t will be displayed at position 0. As before, take the second deck and arrange all its cards as a faro shuffle of the first deck, keeping every card of the second deck right below every card of the first deck. The opponents of team $n - 1$ are determined by the first card on the top of the deck. Therefore, team $n - 1$ faces team t at round $r = t$. Figure 2.8 shows an example with $n = 6$.

A tournament schedule as presented in Table 2.1 can be obtained through the faro method by extracting the information about the opponents of every team from Figure 2.8.

2.4 The faro method and circle method are equivalent

In the faro method, during $n-2$ iterations, we circularly shift the cards in the deck, except the card $n-1$, thus every other card i will appear at the top of the deck at iteration i. The shifting move can be done by placing the card on the top between the cards at positions $n-2$ and $n-1$. Likewise, at each iteration $j \in \{0, \ldots, n-2\}$ the position i such that $0 \leq i < n-1$ is occupied by the card numbered $(i+j) \mod (n-1)$. This shifting yields another permutation of the cards in which $n-1$ is always fixed at position $n-1$ and its permutation cycle can be written as $(0,1,\ldots,n-3,n-2)(n-1)$. Function $\mathcal{F}(j,i)$ computes the position of the card of value i in the shifted n-size deck at iteration $j \in \{0, \ldots, n-2\}$

$$\mathcal{F}(j,i) = \begin{cases} n-1 & \text{if } i = n-1 \\ i-j \mod (n-1) & \text{otherwise} \end{cases} \tag{2.3}$$

Function $\overline{\mathcal{F}}(j,i)$ computes the value of the card that is placed at position i at iteration $j \in \{0, \ldots, n-2\}$

$$\overline{\mathcal{F}}(j,i) = \begin{cases} n-1 & \text{if } i = n-1 \\ i+j \mod (n-1) & \text{otherwise} \end{cases} \tag{2.4}$$

For a deck of size n at iteration j, we can represent the faro shuffle permutation of a shifted deck by the mapping $\Re^{\star}(j,i)$ based on of Eq. 2.2, for each iteration $j \in \{0, \ldots, n-2\}$ and for each card at position $i \in \{0, \ldots, n-1\}$

$$\Re^{\star}(j,i) = \begin{cases} n-1 & \text{if } i = n-1 \\ (2\mathcal{F}(j,i)) \mod (n-1) & \text{otherwise} \end{cases} \tag{2.5}$$

In a deck of n cards, the card i ends at position $\Re^{\star}(j,i)$ after j shifts and one faro shuffle. Since the values of $\mathcal{F}(j,i)$ are never computed for the first case in Eq. 2.3 (when $i = n-1$), we can apply the following substitution:

$$\begin{aligned}(2(\mathcal{F}(j,i))) \mod (n-1) &= (2(i-j)(\mod (n-1))) \mod (n-1)\\ &= (2(i-j)) \mod (n-1)\\ &= (2i-2j) \mod (n-1)\end{aligned}$$

Hence, we may rewrite $\Re^\star(j,i)$ as:

$$\Re^\star(j,i) = \begin{cases} n-1 & \text{if } i = n-1 \\ (2i-2j) \mod (n-1) & \text{otherwise} \end{cases} \tag{2.6}$$

Finally, we define the function $\mathcal{A}(j,i) = \overline{\mathcal{F}}(j, \Re^\star(j,i))$ to recover the value of the card that is placed at position $\Re^\star(j,i)$ in the shifted deck of size n at iteration j after a faro shuffle:

$$\begin{aligned}\mathcal{A}(j,i) &= \overline{\mathcal{F}}(j, \Re^\star(j,i))\\ &= \overline{\mathcal{F}}(j, (2i-2j) \mod (n-1)) && i < n-1\\ &= (j + (2i-2j) \mod (n-1)) \mod (n-1)\\ &= (j+2i-2j) \mod (n-1)\\ &= (2i-j) \mod (n-1)\end{aligned}$$

Function $\mathcal{A}(t,r)$ determines the opponent of team t in round r for $r \neq t$. Considering that team i plays against team $n-1$ at round i the full opponent schedule for the faro method is:

$$\mathcal{A}(t,r) = \begin{cases} r & \text{if } t = n-1 \\ n-1 & \text{if } t = r \\ (2r-t) \mod (n-1) & \text{otherwise} \end{cases} \tag{2.7}$$

Such schedule is equal to the one constructed by the circle method since Eq. 2.7 is equal to Eq. 2.1. Therefore, the two methods are equivalent.

Chapter 3

Exploring the Solution Space

This chapter describes, in edge coloring terms, four well-known neighborhood structures for round-robin sport scheduling problems. These neighborhood structures are analyzed as functions that operate on edge-colored graphs. Each new coloring obtained by each function is a neighbor of the current schedule in the neighborhood structure under consideration. At the end of this chapter, a proof of a theorem relating the faro shuffle of playing cards and the neighborhood connectivity of one of the existing neighborhood structures is presented.

3.1 Existing neighborhood structures

Connectivity is a key feature of neighborhood structures that drastically affects the performance and search capability of local search algorithms. Let s be a schedule of an SRR tournament with n teams and let the solution space $\mathcal{S}$ be the set of all possible schedules for those n teams. A neighborhood structure is a mapping that assigns to each schedule $s \in \mathcal{S}$, a set of schedules $N(s)$ that are neighbors of s. Local search procedures use the concept of neighborhoods to move from one schedule s to a neighbor schedule $s' \in N(s)$. The solution space of a neighborhood structure is said to be connected if it is possible to move from any solution s to any other solution $s' \in \mathcal{S}$ through a series of neighborhood moves.

The idea of a neighborhood graph, in which vertices are SRR schedules, and each edge is a transition from a particular SRR schedule to another, will be used throughout the text. All neighborhood structures will be modeled in terms of operators over ordered one-factorizations (or proper edge colorings with a minimum number of colors) of complete graphs.

Due to several research groups working in parallel on sports scheduling problems, it is not uncommon to find in the literature different names corresponding to the same concept, including the nomenclature of the existing neighborhood structures. As can be seen in [Anagnostopoulos et al., 2006], [Costa et al., 2012], [Di Gaspero and Schaerf, 2007] and [Ribeiro and Urrutia, 2007], four different neighborhood structures have been used in local search procedures for round-robin sport scheduling problems in the literature. They are mostly known as *Round Swap* (RS), *Partial Round Swap* (PRS), *Team Swap* (TS) and *Partial Team Swap* (PTS). PRS is a generalization of RS and PTS is a generalization of TS, which makes PTS and PRS the two neighborhood structures most commonly used in local search based algorithms.

Round Swap (RS) and *Partial Round Swap* (PRS) neighborhood structures are based on permutation of opponents between rounds. Given two different rounds, each neighbor in the RS neighborhood structure is obtained by exchanging the opponents of all teams in those rounds. Figure 3.1 gives an example of a move in RS. On the left, there is an initial schedule in which rounds 2 and 4 were selected as parameters of a move in RS. The resulting schedule after changing the opponents in the selected rounds is shown on the right side of the Figure 3.1.

					r			
		0	1	2	3	4	5	6
t	0	7	2	4	6	1	3	5
	1	6	7	3	5	0	2	4
	2	5	0	7	4	6	1	3
	3	4	6	1	7	5	0	2
	4	3	5	0	2	7	6	1
	5	2	4	6	1	3	7	0
	6	1	3	5	0	2	4	7
	7	0	1	2	3	4	5	6

					r			
		0	1	2	3	4	5	6
t	0	7	2	1	6	4	3	5
	1	6	7	0	5	3	2	4
	2	5	0	6	4	7	1	3
	3	4	6	5	7	1	0	2
	4	3	5	7	2	0	6	1
	5	2	4	3	1	6	7	0
	6	1	3	2	0	5	4	7
	7	0	1	4	3	2	5	6

Figure 3.1: A move in RS neighborhood for a tournament with $n = 8$ teams.

The PRS neighborhood is a generalization of RS in which each neighbor is obtained by exchanging the opponents between two different rounds for a subset of teams. For any team $t \in T$ and for any two rounds $r_1 \in R$ and $r_2 \in R$, with $r_1 \neq r_2$, let T' be a minimum cardinality subset of teams, including team t, in which the opponents of the teams in T', in rounds r_1 and r_2, are the same. In other words, $T' = \{1, \dots, k\} \subseteq T$ is minimum and such that $t \in T'$ and $\{i \in T : \exists u \in T'$ such that $adj(i, r_1) = u\} = \{i \in T : \exists u \in T'$ such that $adj(i, r_2) = u\}$. A schedule obtained by exchanging the opponents of each team $t \in T'$ in rounds r_1 and r_2 is a neighbor of the initial schedule in PRS neighborhood. Table 3.1 gives an illustration of the subset

$T' \in T$ when considering team 1 and rounds 0 and 3. Observe that, when T' is equal to T a move in PRS neighborhood is equal to a move in RS neighborhood.

						r				
		0	1	2	3	4	5	6	7	8
	0	9	2	4	6	8	1	3	5	7
	1	8	9	3	5	7	0	2	4	6
	2	7	0	9	4	6	8	1	3	5
	3	6	8	1	9	5	7	0	2	4
t	4	5	7	0	2	9	6	8	1	3
	5	4	6	8	1	3	9	7	0	2
	6	3	5	7	0	2	4	9	8	1
	7	2	4	6	8	1	3	5	9	0
	8	1	3	5	7	0	2	4	6	9
	9	0	1	2	3	4	5	6	7	8

						r				
		0	1	2	3	4	5	6	7	8
	0	9	2	4	6	8	1	3	5	7
	1	5	9	3	8	7	0	2	4	6
	2	7	0	9	4	6	8	1	3	5
	3	6	8	1	9	5	7	0	2	4
t	4	2	7	0	5	9	6	8	1	3
	5	4	6	8	1	3	9	7	0	2
	6	3	5	7	0	2	4	9	8	1
	7	8	4	6	2	1	3	5	9	0
	8	1	3	5	7	0	2	4	6	9
	9	0	1	2	3	4	5	6	7	8

Table 3.1: The minimum cardinality subset of teams T', including team 1, in which the opponents of every team $t \in T'$, in rounds $r = 0$ and $r = 3$, are the same, for a tournament with $n = 10$ teams, is $\{1, 4, 7\}$.

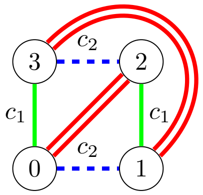


Figure 3.2: A graph obtained through a move in RS neighborhood with colors c_1 and c_2 as parameters.

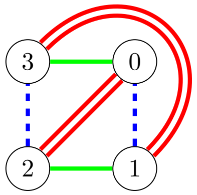


Figure 3.3: A graph obtained through a move in TS neighborhood with teams 0 and 2 as parameters.

Using edge coloring terms in order to obtain a neighbor in the RS neighborhood structure, we take two different used colors, c_j and c_k, and exchange them for all edges colored with any of those colors, i.e., we exchange the two factors r_j and r_k associated to the colors c_j and c_k. Figure 3.2 illustrates the application of RS on the tournament depicted in Figure 1.1, using colors c_1 (represented by the single solid line) and c_2 (represented by the dashed line) as parameters.

For a move in the PRS neighborhood structure, we select any two distinct colors, for instance, c_1 and c_2, and consider a cycle in the subgraph induced by the edges that have those colors. Then, we exchange the colors in the cycle to get a neighbor schedule. If the involved edges are incident to all vertices of the graph, i.e., if they form a hamiltonian cycle, then this move is equivalent to a move in the RS neighborhood structure. As a consequence, PRS and RS are equivalent for perfect one-factorizations. Figure 3.4 gives an illustration of a move in PRS neighborhood.

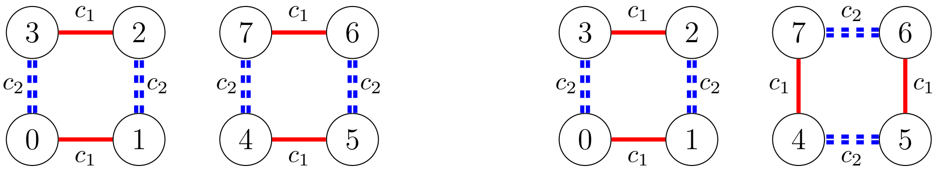


Figure 3.4: On the left: a subgraph induced by the edges with colors c_1 (solid lines) and c_2 (double-dashed lines). On the right: a neighbor coloring obtained with a move in PRS neighborhood, by exchanging the assignment of colors in one cycle.

Each neighbor in the TS neighborhood is obtained by exchanging all opponents of two distinct teams t_i and t_j, in all rounds except for round r such that $\Upsilon(t_i, r) = t_j$ (see Figure 3.5). In this neighborhood, each neighbor coloring is isomorphic to the initial one.

					r			
		0	1	2	3	4	5	6
	0	7	2	4	6	1	3	5
	1	6	7	3	5	0	2	4
	2	5	0	7	4	6	1	3
t	3	4	6	1	7	5	0	2
	4	3	5	0	2	7	6	1
	5	2	4	6	1	3	7	0
	6	1	3	5	0	2	4	7
	7	0	1	2	3	4	5	6

					r			
		0	1	2	3	4	5	6
	0	7	2	4	6	1	3	5
	1	6	7	3	5	0	2	4
	2	5	4	6	1	3	7	0
t	3	4	6	1	7	5	0	2
	4	3	5	0	2	7	6	1
	5	2	0	7	4	6	1	3
	6	1	3	5	0	2	4	7
	7	0	1	2	3	4	5	6

Figure 3.5: A move in TS neighborhood using teams 2 and 5 as parameters.

The PTS neighborhood structure is a generalization of the TS neighborhood structure. For any round r and for any two distinct teams t_1 and t_2, where $adj(t_1, r) \neq t_2$, let R' be a minimum cardinality subset of rounds including r in which the opponents of t_1 and t_2 are the same, i.e., $R' = \{r_1, ..., r_k\} \subseteq R$ is minimal and such that $r \in R'$ and $\{t \in T : \exists r_j \in R'$ such that $adj(t, r_j) = t_1\} = \{t \in T : \exists r_j \in R'$ such that $adj(t, r_j) = t_2\}$. A schedule obtained by exchanging the opponents of teams t_1 and t_2 in all rounds in R' is a neighbor in PTS. Figure 3.6 gives an illustration of the subset $R' \in R$ when we consider round 0 and teams 2 and 7. In the PTS neighborhood structure, the opponents of the teams in a given round are exchanged and a procedure of *repair chain* must be used in order to remove any violation created by PTS (see [Di Gaspero and Schaerf, 2007] for further details). Observe that if R' is equal to $R \setminus r : adj(t_1, r) = t_2$ then PTS is equivalent to TS.

					r			
		0	1	2	3	4	5	6
	0	7	2	4	6	1	3	5
	1	6	7	3	5	0	2	4
	2	5	0	7	4	6	1	3
t	3	4	6	1	7	5	0	2
	4	3	5	0	2	7	6	1
	5	2	4	6	1	3	7	0
	6	1	3	5	0	2	4	7
	7	0	1	2	3	4	5	6

					r			
		0	1	2	3	4	5	6
	0	7	2	4	6	1	3	5
	1	6	7	3	5	0	2	4
	2	0	1	7	4	6	5	3
t	3	4	6	1	7	5	0	2
	4	3	5	0	2	7	6	1
	5	2	4	6	1	3	7	0
	6	1	3	5	0	2	4	7
	7	5	0	2	3	4	1	6

Figure 3.6: The minimum cardinality subset of rounds including $r = 0$ in which the opponents of $t = 5$ and $t = 7$ are the same, for a tournament with $n = 8$ teams, is $\{0, 1, 5\}$.

In edge coloring terms, each neighbor in the TS neighborhood is obtained by exchanging two distinct vertices v_1 and v_2. After this move, v_1 will have the assignment of colors to its incident edges that previously belonged to v_2 and vice-versa. Figure 3.3 illustrates the application of TS, using team 0 and team 2 as parameters, based on the schedule previously presented in Figure 1.1.

In graph theoretical terms, a move in PTS neighborhood structure consists of the following steps. First select two distinct vertices v_1 and v_2 of the complete edge-colored graph, then consider the subgraph induced by the edges connecting every other vertex to v_1 and v_2. This subgraph is isomorphic to the complete bipartite graph $K_{2,n-2}$ and colored with $n-2$ colors, (all colors in K_n but the color of the edge (v_1, v_2)). Compute a subset of vertices $W \subseteq V - \{v_1, v_2\}$ such that the set of colors assigned to edges joining v_1 to vertices in W is equal to the set of colors assigned to edges joining v_2 to vertices in W. Next, exchange the color assignment of edges (v_1, w_i) and (v_2, w_i) for each vertex w_i in W. Figure 3.7 shows such a subgraph $K_{2,n-2}$ with $W = \{w_5, w_6\}$ and Figure 3.8 illustrates the coloring after exchanging colors around W. Note that if $W = V - \{v_1, v_2\}$, the move is equivalent to a TS move.

3.2 An analytical study in connectivity of RS and PRS neighborhood structures

Let us analyze some aspects of the connectivity of PRS. Note that moves in the RS neighborhood affect the order of one-factors in the one-factorization, but it does not modify the one-factorization itself. Also note that, whenever the current one-factorization is perfect, PRS neighborhood is equivalent to the RS neighborhood. In-

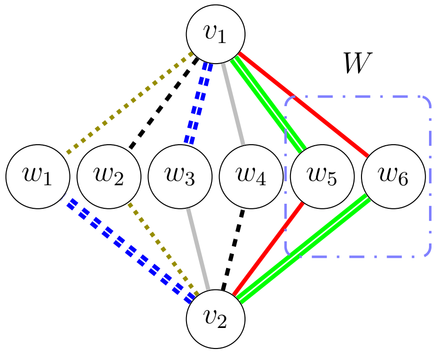


Figure 3.7: A subgraph $K_{2,n-2}$ with $W = \{w_5, w_6\}$

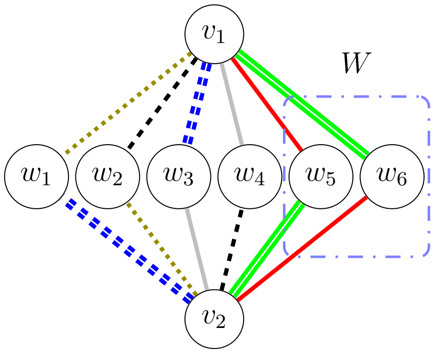


Figure 3.8: The subgraph of Figure 3.7 after exchanging of colors around W.

deed, exchanging opponents between two rounds corresponds to exchanging edges between two one-factors. In order to maintain each of the involved factors as one-factors, the edges exchanged must form a set of cycles. If the one factorization is perfect, the only cycle that can be exchanged between two rounds is a Hamiltonian cycle, implying that all edges of both one-factors must be exchanged.

Let n be the number of participating teams in a tournament. For tournaments with $n \leq 100$, whose initial construction was obtained with the circle method, the existing neighborhoods are not connected for several values of $n = p + 1$, where p is a prime number. It means that not all the solution space is reachable using those neighborhood structures. Moreover, when a schedule is constructed with the circle method and using the existing neighborhood structures it is not possible to move from an initial schedule to other schedules that are nonisomorphic to the initial one.

The circle method itself, as well as the faro method, generates P1Fs of K_n wherein n is equal to 4, 6, 8, 12, 14, 18, 20, 24, 30 and so on, that is, whenever $n = p+1$, being p is a prime number [Kobayashi, 1989], therefore the subgraph induced by the edges belonging to any two one-factors is a hamiltonian cycle.

As a result, whenever a coloring is generated using the circle method, and n is equal to a prime number plus one, it is not possible to obtain new non-isomorphic edge colorings using the PRS neighborhood structure, since all moves in this neighborhood are equivalent to moves in the RS neighborhood and all schedules obtained with them are isomorphic to the original one.

As an example, K_{14} has 1,132,835,421,602,062,347 non-isomorphic colorings to be explored, see [Kaski and Ostergard, 2009]. However, using the RS and PRS neighborhoods any local search procedure may stay trapped in P1Fs, as the one constructed with the circle method.

3.3 An analytical study in connectivity of TS and PTS neighborhood structures

Now, we study the connectivity of TS and PTS. Each neighbor in a TS neighborhood is obtained by swapping the label of two distinct vertices of the graph. Therefore, all neighbor schedules generated with TS moves are isomorphic to each other.

PTS neighborhood, however, may modify the underlying one-factorization of the schedule, for that reason it is frequently used in local search procedures to solve round-robin sport scheduling problems. In [Costa et al., 2012] and [Januario and Urrutia, 2015] it is established that, for several values of $n = p + 1$ being p a prime number, the one-factorization that is built by the circle method is such that any possible PTS move is equivalent to a TS move. In those one-factorizations, the smallest set of factors in which any pair of vertices shares all their adjacent vertices has size $n - 2$.

The previous chapter introduced a method to construct SRR schedules based on faro shuffles of a deck of playing cards and showed the equivalence between the introduced method and the circle method. In consequence, it is reasonable to think that properties of the faro shuffle may be translated into characteristics of the schedule constructed by the circle method. So far, [Januario and Urrutia, 2015] is the only work that has related these two research areas.

In the following, an analysis of the connectivity of PTS based on permutation cycles is presented. Table 3.2 gives an example of a construction, step by step, of a permutation cycle that describes a move in a PTS neighborhood for a tournament with $n = 10$, built with the circle method, taking round 0 and teams 5 and 9 as parameters. The values of the opponents were computed by Eq. 2.1.

Opponents		Permutation cycle
$adj(5,0) = 4$	$adj(9,4) = 4$	(0)
$adj(5,4) = 3$	$adj(9,3) = 3$	(0,4)
$adj(5,3) = 1$	$adj(9,1) = 1$	(0,4,3)
$adj(5,1) = 6$	$adj(9,6) = 6$	(0,4,3,1)
$adj(5,6) = 7$	$adj(9,7) = 7$	(0,4,3,1,6)
$adj(5,7) = 0$	$adj(9,0) = 0$	(0,4,3,1,6,7)

Table 3.2: A construction of a permutation cycle based on the move illustrated in Table 3.6.

The set (0,4,3,1,6,7) is a 6-cycle equals to the minimum cardinality subset of rounds including $r = 0$ in which the opponents of $t = 5$ and $t = 9$ are the same. From Table 3.6 we can also obtain another minimum cardinality subset of rounds including

$r = 2$ in which the opponents of $t = 5$ and $t = 9$ are the same, described by the set (2,8).

It is important to emphasize that, when a move in PTS has an $(n-2)$-size permutation cycle, such move will be equivalent to a move in TS neighborhood structure. Therefore, the underlying one-factorization of the schedule is not affected by the move.

The connectivity of PTS when the initial schedule is constructed with the circle method is investigated for values of n in which the faro shuffle permutation with n elements has an orbit of size $n-2$. These values of n match the list of numbers for which a faro shuffle permutation permutes all except the first and the last cards, listed in [OEIS, 2015], that were experimentally obtained. So far, there is no known formula to compute them. Additionally, those numbers are equal to $p + 1$, being p a prime number. For such values of n and using the studied neighborhoods, it is not possible to move from a one-factorization constructed by the circle method to a one-factorization non-isomorphic to it using the existing neighborhoods.

3.3.1 When not to use Partial Team Swap

For the remainder of this chapter, vertices and colors will be referred by their indexes. In that way, a given integer t may refer either to a vertex v_t or to a color c_t. In each context, it should be clear whether we are referring to vertices or to colors.

Consider the complete bipartite graph $B = (X, W)$, isomorphic to $K_{2,n-2}$, spanned by a set $X = \{u, v\}$, $u < v$, and a set $W = V \setminus \{u, v\} = \{w_1, w_2, \dots, w_{n-2}\}$. Without loss of generality, we can assume that the edges of $B = (X, W)$ are colored with colors $1, 2, \dots, n-2$, as can be seen in Figure 3.9. Let $S_{u,v}$ be a partition of W into subsets $\{S^1_{u,v}, \dots, S^q_{u,v}, \dots, S^{\psi}_{u,v}\}$ such that for any $S^q_{u,v}$, with $1 \leq q \leq \psi < |W|$, the set of colors of edges (u, w), for $w \in S^q_{u,v}$, is the same as the set of colors of edges (v, w), for $w \in S^q_{u,v}$, and each $S^q_{u,v}$ is inclusionwise minimal. These colors define a permutation π of $\{1, \dots, n-2\}$ by setting $\pi(\mathcal{C}(u, w_i)) = \mathcal{C}(v, w_i)$.

In order to construct the partition $S_{u,v}$, we build an auxiliary graph $\mathcal{G}$ as follows (see Figure 3.10): each color $\{1, \dots, n-2\}$ is associated to a vertex in $\mathcal{G}$. Each vertex w_i in W corresponds to an oriented arc w_i of $\mathcal{G}$. The arc w_i is oriented from vertex k to vertex l if and only if (u, w_i) has color k and (v, w_i) has color l.

$\mathcal{G}$ has $n-2$ vertices and $n-2$ arcs. Note that $\mathcal{G}$ defines at least one permutation of colors without fixed points. A fixed point would mean that, for a given i, edges (u, w_i) and (v, w_i) would have the same color. For that reason, $\mathcal{G}$ has no (directed) loops. In the graph B, each color occurs once on edges adjacent to u and once on edges adjacent to v, as a result, each vertex in $\mathcal{G}$ has exactly one incoming arc and one

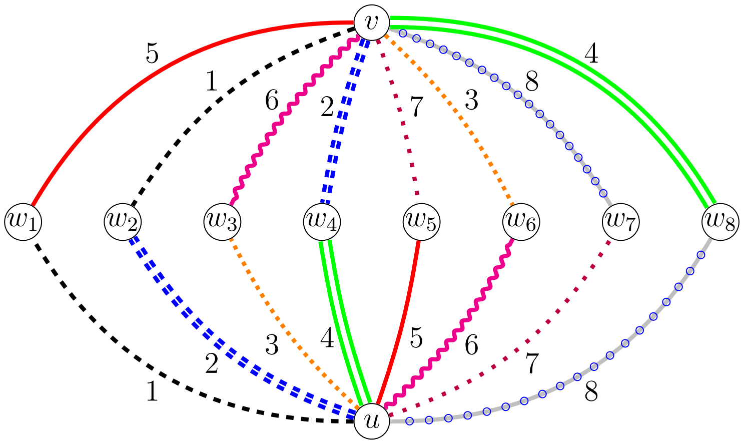


Figure 3.9: A complete bipartite graph $B = (X, W)$ isomorphic to $K_{2,8}$.

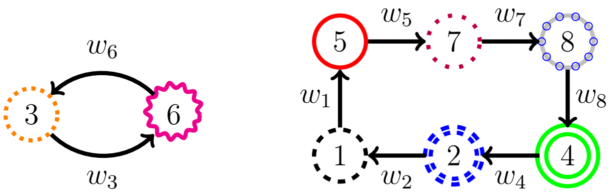


Figure 3.10: The graph $\mathcal{G}$, generated from the bipartite graph in Figure 3.9

outgoing arc. Hence, $\mathcal{G}$ consists of vertex-disjoint cycles of length at least two. Clearly there is a one-to-one correspondence between the orbits of π and the oriented cycles of $\mathcal{G}$. The sets of edges of the cycles within $\mathcal{G}$ are the classes $S^q_{u,v}$ of the partition of W.

Let $d(i)$ be the class of vertex i of K_n in the partition $S_{u,v}$, that is, $i \in S^{d(i)}_{u,v}$. A recursive definition of $S^{d(i)}_{u,v}$ is given as follows:

- $i \in S^{d(i)}_{u,v}$.
- $j \in S^{d(i)}_{u,v} \rightarrow adj(u, \mathcal{C}(v, j)) \in S^{d(i)}_{u,v}$.
- nothing else is in $S^{d(i)}_{u,v}$.

Let u, v and w be three distinct vertices of V such that $u < v$. Observe that $2 \leq |S^{d(w)}_{u,v}| \leq n - 2$ and that $x \in S^{d(w)}_{u,v} \leftrightarrow d(x) = d(w)$.

The following mathematical arguments were designed to characterize the values of n that allows PTS to move from a coloring constructed by the circle method to a coloring non-isomorphic to the original one. It is analogous to define the values of n for which there is at least one orbit of size smaller than $n-2$ in the canonical coloring of K_n. Functions $\Upsilon(v, k)$, defined by Eq. 2.1, and $\mathcal{C}(v, w)$, defined in Chapter 1, refer to canonical colorings.

Given three distinct vertices u, v and k, Lemma 1 shows how to compute the value of another vertex of K_n that belongs to the class $S_{u,v}^{d(k)}$.

Lemma 1. *If u, v and k are three distinct vertices of K_n, such that $u < v < n-1$, $k \leq n-1$, and $i = d(k)$, then the following holds:*

(I) *if $k = (2u - v) \mod (n-1)$ then $n - 1 \in S_{u,v}^i$*

(II) *if $k = n-1$ then $(2v - u) \mod (n-1) \in S_{u,v}^i$*

(III) *else vertex $(k + v - u) \mod (n-1) \in S_{u,v}^i$*

Proof. Let r be the color of the edge (v, k), i.e, $\mathcal{C}(v, k) = r$. By the recursive definition of $S_{u,v}^i$, $k \in S_{u,v}^i$ and $\Upsilon(u, \mathcal{C}(v, k)) \in S_{u,v}^i$.

To prove (I), note that we have $\Upsilon(v, r) = k$ and, since $v < (n-1)$ and $k < (n-1)$, we can apply the following substitution

$$\begin{aligned} \Upsilon(v, r) &= k \\ &= (2r - v) \mod (n-1) && \text{third case of Eq. 2.1} \\ &= (2u - v) \mod (n-1) && \text{value of } k \text{ in (I)} \end{aligned}$$

implying that $r = u$. Consequently, we have

$$\begin{aligned} \Upsilon(u, \mathcal{C}(v, k)) &= \Upsilon(u, r) && \text{since } \mathcal{C}(v, k) = r \\ &= \Upsilon(u, u) && \text{since } r = u \\ &= n - 1 && \text{second case of Eq. 2.1} \end{aligned}$$

therefore, $n - 1$ must be in $S_{u,v}^i$

To prove (II), since $\Upsilon(v, r) = k$, we can apply the following substitution:

$$\begin{aligned} \Upsilon(v, r) &= k \\ &= n - 1 && \text{value of } k \text{ in (II)} \end{aligned}$$

which implies in $r = v$, according to the second case of Eq. 2.1. Consequently, we have

$$\begin{aligned} \Upsilon(u, \mathcal{C}(v, k)) &= \Upsilon(u, r) && \text{since } \mathcal{C}(v, k) = r \\ &= \Upsilon(u, v) && \text{since } r = v \\ &= (2v - u) \mod (n-1) && \text{third case of Eq. 2.1} \end{aligned}$$

therefore, $(2v - u) \mod (n-1)$ must be in $S_{u,v}^i$

To prove (III), since k and v are different from each other and different from $n-1$, we can apply the following substitution:

$$\begin{aligned} \Upsilon(v,r) &= k && \\ &= (2r-v) \mod (n-1) && \text{third case of Eq. 2.1} \\ 2r - v &\equiv k \mod (n-1) && \text{property of congruence} \\ 2r &\equiv (k+v) \mod (n-1) && \text{adding } v \text{ to both sides} \end{aligned}$$

Observe that

$$\begin{aligned} n &\equiv 1 \mod (n-1) && \\ 2({}^{n}\!/\!{}_{2}) &\equiv 1 \mod (n-1) && \\ 2({}^{n}\!/\!{}_{2})(k+v) &\equiv (k+v) \mod (n-1) && \text{multiplying both sides by } k+v \end{aligned}$$

This implies that $r = ({}^{n}\!/\!{}_{2})(k+v) \mod (n-1)$.

$$\begin{aligned} \Upsilon(u, \mathcal{C}(v,k)) &= \Upsilon(u,r) && \text{since } \mathcal{C}(v,k) = r \\ &= \Upsilon(u, ({}^{n}\!/\!{}_{2})(k+v) \mod (n-1)) && \text{since } r = ({}^{n}\!/\!{}_{2})(k+v) \mod (n-1) \end{aligned}$$

There are two cases to be considered. Assume first that $r \neq u$.

$$\begin{aligned} \Upsilon(u, \mathcal{C}(v,k)) &= (2(({}^{n}\!/\!{}_{2})(k+v) \mod (n-1)) - u) \mod (n-1) && \text{third case of Eq. 2.1} \\ &= (2(({}^{n}\!/\!{}_{2})(k+v)) - u) \mod (n-1) && \\ &= (n(k+v) - u) \mod (n-1) && \\ &= (k+v-u) \mod (n-1) && n \equiv 1 \mod (n-1) \end{aligned}$$

Therefore, vertex $(k+v-u) \mod (n-1)$ must be in $S^{i}_{u,v}$.

In the second case, if $r = u$ then $\Upsilon(u,r) = (n-1)$. But in this case, since $2r \equiv k+v \mod (n-1)$, we have $2u \equiv k+v \mod (n-1)$. After subtracting v from both sides we have $k = (2u-v) \mod (n-1)$ and then statement (I) of Lemma 1 should be applied. □

Lemma 2 takes into account the case where vertex $v = (n-1)$, which is ommited fom Lemma 1, and shows how to compute the value of another vertex $(2k-u) \mod (n-1)$ that belongs to the class $S^{d(k)}_{u,n-1}$.

Lemma 2. *If u and k are two distinct vertices of K_n, such that $u < n-1$, $k < n-1$, and $i = d(k)$, then vertex $(2k-u) \mod (n-1) \in S^i_{u,n-1}$.*

Proof. Let r be the color of edge $(n-1, k)$. Note that, by the first case of Eq. 2.1, $r = k$. Then, vertex $\Upsilon(u, \mathcal{C}(n-1,k)) = \Upsilon(u,r) = \Upsilon(u,k)$ must be in $S^i_{u,n-1}$. Since $k \neq u$, applying the third case of Eq. 2.1, we have $\Upsilon(u,r) = (2r-u) \mod (n-1)$ and, since $r = k$, $\Upsilon(u,r) = (2k-u) \mod (n-1)$. Therefore, the vertex $(2k-u) \mod (n-1)$ must be in $S^i_{u,(n-1)}$. □

In Lemma 3, if n is chosen so that $n-1$ is not a prime then $|S^{d(1)}_{0,s}| < n-2$. Note that in Lemma 3 n must be at least 10 since $n-1$ is prime for smaller even values of $n > 2$.

Lemma 3. *If $s \cdot t = n-1$, with $1 < s \leq t < n-1$, then the class $S^{d(1)}_{0,s}$ of $S_{0,s}$ has size $t < n-2$.*

Proof. By definition, $1 \in S^{d(1)}_{0,s}$. Note that

$$\begin{aligned}
\Upsilon(0,s) &= (2\cdot 0 - s) \mod (n-1) && \text{third case of Eq. 2.1}\\
&= n-1-s && \text{congruence modulo } (n-1)\\
&= s\cdot(t-1) && \text{because } s\cdot t = n-1
\end{aligned}$$

Since $(s \cdot i + 1) \mod (n-1)$ is not a multiple of s, for any positive value of i, then $(s \cdot i + 1) \mod (n-1) \neq (2 \cdot 0 - s) \mod (n-1)$. In consequence, statement (III) of Lemma 1 applies.

By statement (III) of Lemma 1, $(1+s-0) \mod (n-1) = (s+1) \mod (n-1) \in S^{d(1)}_{0,s}$. The fact that $(s+1) \mod (n-1) \in S^{d(1)}_{0,s}$ implies $(2s+1) \mod (n-1) \in S^{d(1)}_{0,s}$. Following this way and applying the statement (III) of Lemma 1 repeatedly, we show that in fact $1 + i \cdot s \in S^{d(1)}_{0,s}$ for all $0 \leq i < t$ (observe that $1 + t \cdot s \equiv 1 \mod (n-1)$). Note that $(1+i\cdot s) \mod (n-1)$ is never equal to $n-1-s$ because they are not multiple of s while $n-1-s$ is equal to $s\cdot(t-1)$. Since no other vertex can be obtained by the recursive definition, the class $S^{d(1)}_{0,s}$ has exactly t vertices. To show that $t < n-2$, note that $t = {}^{(n-1)}/_{s}$, which is at most ${}^{(n-1)}/_{3}$ and this is strictly smaller than $n-2$ for n larger than or equal to 4. □

Corollary 1 is a consequence of Lemma 3.

Corollary 1. *When $n-1$ is not prime there exists u and v in K_n such that partition $S_{u,v}$ of $V \setminus \{u,v\}$ generated by the canonical coloring has more than one class.*

Lemma 4 states that, if one chooses n such that $n-1$ is a prime number then $S_{u,v}$ has one single class of size $n-2$, whenever u and v are distinct and smaller than $n-1$. In such conditions, a move in PTS neighborhood will have the same effect as a move in TS neighborhood structure.

Lemma 4. *If $(n-1)$ is prime, $S_{u,v}$ with $u < v < n-1$ has a unique class of size $n-2$.*

Proof. Choose $i = d((2v-u) \mod (n-1))$. By definition, the vertex $(2v-u) \mod (n-1) \in S^i_{u,v}$. Let l be the smallest integer such that $2v-u+(l-1)(v-u) \equiv 2u-v \mod (n-1)$. By the recursive application of statement (III) in Lemma 1, we show that all vertices $(2v-u+(j-1)(v-u)) \mod (n-1) \in S^i_{u,v}$ for all vaues of $2 \leq j \leq l$.

In order to determe l note that

$$\begin{aligned}
2v-u+(l-1)(v-u) &\equiv 2u-v \mod (n-1)\\
2v-u+l\cdot v-l\cdot u-v+u &\equiv 2u-v \mod (n-1)\\
(l+1)v-l\cdot u &\equiv 2u-v \mod (n-1)\\
(l+2)v-(l+2)u &\equiv 0 \mod (n-1) \qquad \text{subtract } 2u-v \text{ from both sides}\\
(l+2)(v-u) &\equiv 0 \mod (n-1)
\end{aligned}$$

since $(v-u)$ and $(n-1)$ are relatively prime then

$$\begin{aligned}
l+2 &\equiv 0 \mod (n-1)\\
l &= (n-3) \mod (n-1)
\end{aligned}$$

and since l has to be as small as possible, then $l = n-3$.

This implies that, recursively applying the statement (III) of Lemma 1, we show that $n-4$ different vertices (all vertices $(2v-u+(j-1)(v-u)) \mod (n-1)$ for $2 \leq j \leq n-3$) are in $S^i_{u,v}$ in addition to vertex $(2v-u) \mod (n-1)$. Then applying statement (I) of Lemma 1 we show that $n-1$ also belongs to $S^i_{u,v}$. Therefore, $S^i_{u,v}$ has size $n-2$. □

Lemma 5 is a generalization of Lemma 2, by applying the latter t consecutive times.

Lemma 5. *Let $i = d(k)$, if vertex $k \in S^i_{u,n-1}$ then, for each $t \geq 1$, vertices $\mathcal{P}(k,t) = (2^t k - (2^t-1)u) \mod (n-1) \in S^i_{u,(n-1)}$.*

Proof. Observe that, when $t = 1$, Lemma 5 is equal to Lemma 2. Suppose that Lemma 5 is valid for a given value of $t \geq 1$. We show that it is also valid for $t+1$ by applying Lemma 2. That is, if $(2^tk - (2^t - 1)u) \mod (n-1) \in S^i_{u,(n-1)}$ then:

$$\begin{aligned}(2(2^tk - (2^t - 1)u \mod (n-1)) - u) \mod (n-1) &= (2^{t+1}k - 2(2^t - 1)u - u) \mod (n-1)\\ &= (2^{t+1}k - 2^{t+1}u + 2u - u) \mod (n-1)\\ &= (2^{t+1}k - (2^{t+1} - 1)u) \mod (n-1)\\ &\in S^i_{u,(n-1)}\end{aligned}$$

□

Function $\mathcal{P}(k,t)$ gives indeed the value of the t-th vertex $(2^tk - (2^t - 1)u) \mod (n-1)$ that belongs to the class $S^{d(k)}_{u,(n-1)}$. For instance, when $t = 1$

$$\begin{aligned}(2^tk - (2^t - 1)u) \mod (n-1) &= (2^1k - (2^1 - 1)u) \mod (n-1) \qquad t = 1\\ &= (2k - (2 - 1)u) \mod (n-1)\\ &= (2k - u) \mod (n-1)\end{aligned}$$

which is the same result as the one of Lemma 2.

In Lemma 6, the function $\mathcal{R}(x,t)$ is a simplification of the faro shuffle function given in Eq. 2.2. It gives the position of the card x after t consecutive faro shuffles in the same deck of playing cards. Lemma 6 shows that $\mathcal{R}(x,t)$ and $\mathcal{P}(k,t)$ have the same period.

Lemma 6. *Let* $\mathcal{R}(x,t) = 2^tx \mod (n-1)$, $0 < x < n-1$ *and let* $\mathcal{P}(k,t) = (2^tk - (2^t - 1)u) \mod n-1$, $0 \leq k < n-1$, *where* u *is a constant such that* $0 \leq u < n-1$ *and* $k \neq u$. *Then* $\mathcal{R}(x,t) = \mathcal{R}(x,l)$ *if and only if* $\mathcal{P}(k,t) = \mathcal{P}(k,l)$.

Proof. In the following sequence each formula is equivalent to the next one:

$$\begin{aligned}
\mathcal{R}(x,t) - \mathcal{R}(x,l) &= 0 \\
2^t x \mod (n-1) - (2^l x \mod (n-1)) &= 0 \\
(2^t - 2^l)x &\equiv 0 \mod (n-1) \\
2^t - 2^l &\equiv 0 \mod n-1 \qquad 0 < x < (n-1) \\
(2^t - 2^l)(k-u) &\equiv 0 \mod n-1 \qquad k \not\equiv u \mod (n-1) \\
2^t k - (2^t - 1)u - 2^l k + (2^l - 1)u &\equiv 0 \mod (n-1) \\
\mathcal{P}(k,t) - \mathcal{P}(k,l) &= 0
\end{aligned}$$

□

Finnaly, Theorem 2 gives the desired result in the study of PTS connectivity.

Theorem 2. *Given a complete graph K_n with an even number n of vertices, there exist vertices u and v of K_n such that the partition $S_{u,v}$ of $V \setminus \{u,v\}$ generated by the canonical coloring has more than one class if and only if $(n-1)$ is not prime or the faro shuffle permutation with n elements has no orbit of size $n-2$.*

Proof. If $(n-1)$ is not prime, Corollary 1 shows that u and v exist and Lemma 3 shows how they can be found.

If $n-1$ is prime, Lemma 4 shows that if u and v exist then we must have $v = n-1$. In this case, by Lemma 5, we know that the vertices belonging to $S^{d(k)}_{u,n-1}$ are computable by $(2^t k - (2^t - 1)u) \mod (n-1)$, for $t \geq 1$. By Lemma 6, we know that such a function is periodic with the same period as the faro shuffle function. Then, $S_{u,n-1}$ has more than one class if and only if the faro shuffle permutation with n elements has no orbit of size $n-2$. □

Chapter 4

A New Neighborhood

Last chapter showed that it is not always possible to visit all feasible solutions in scheduling problems using the existing neighborhoods.

The aim of this chapter is to present a new neighborhood for round-robin sport scheduling problems that is able to find good schedules regardless the initial solution or the size of the instance under consideration. We show that one can move from a valid schedule to another which is not isomorphic to the initial one even when moves in the existing neighborhood structures cannot. The neighborhood is described in graph theoretical terms and its correctness is proven.

4.1 Teams and Rounds Swap

This chapter introduces a new neighborhood structure, named Teams and Rounds Swap (TARS), that combines characteristics of PTS and PRS in order to generate neighbors that cannot be obtained using either of them.

A move in the proposed neighborhood structure consists of two phases: selection and change. Given a colored graph s, a vertex $v \in s$ and two distinct colors c and d, the selection phase determines a set of subgraphs of s. In the change phase, the color assignment of each subgraph identified in the previous phase is modified. The change phase occurs immediately after the selection phase and it returns a neighbor solution with the best cost-function for the problem under consideration, among those obtained through modifications introduced to the color assignment of the subgraphs selected in the previous phase. Observe that, for any schedule s, TARS(s) $= \{s' = \mathit{change}(s, \mathit{selection}(s, v, c, d)) : v \in V, c, d \in C, c \neq d\}$. The selection phase returns up to $O(n)$ subgraphs. Since there are $O(n^3)$ possible combinations of one vertex and two distinct colors, the size of TARS is $O(n^4)$.

4.1.1 Selection phase

The selection phase has four parameters: two distinct colors c and d, and one vertex v. It selects the subgraphs η^1 and η_p^t, for $t \in \{2, 3\}$, of the colored graph s, for $1 \leq p \leq p_{max}$, that will be used in the change phase. The value of p_{max} depends on the length of a $cd - cycle$ that has vertex v.

```
parameters: a vertex $v$ and a pair of distinct colors $c$ and $d$
    output: the subgraphs $\eta^1$ and $\eta_p^t$ of the colored graph $s$, for $2 \leq t \leq 3$ and
            $1 \leq p \leq p_{max}$
1  begin
2      $p \leftarrow 1$;
3      $\eta^1 \leftarrow \varnothing$;
4      while $\eta^1 = \varnothing$ do
5          $\eta_p^2 \leftarrow \eta_p^3 \leftarrow \varnothing$;
6          Build a $cd - path$ of size $2p$ from $v_1$ to $v_2$ having $v$ as the middle vertex;
7          $w_1 \leftarrow adj(v_2, c)$;
8          $w_{end} \leftarrow adj(v_1, d)$;
9          if $w_1 = w_{end}$ then
10             Add the $cd - path$ of size $2p$ from $v_1$ to $v_2$ having $v$ as the middle vertex
               into $\eta^1$;
11             Add $w_1$, $(w_1, v_1)$ and $(w_1, v_2)$ into $\eta^1$, thus forming a $cd - cycle$;
12             $p_{max} \leftarrow p - 1$;
13         else
14             Add the $cd - path$ of size $2p$ from $v_1$ to $v_2$ having $v$ as the middle vertex
               into $\eta_p^2$;
15             $j \leftarrow 1$;
16             while $w_j \neq w_{end}$ do
17                 $c_j \leftarrow \mathcal{C}(v_1, w_j)$;
18                 $w_{new} \leftarrow adj(v_2, c_j)$;
19                 Add $w_j$, $(v_1, w_j)$ and $(v_2, w_j)$ to both $\eta_p^2$ and $\eta_p^3$;
20                 $j \leftarrow j + 1$;
21                 $w_j \leftarrow w_{new}$;
22             Add $w_{end}$, $(v_1, w_{end})$ and $(v_2, w_{end})$ to both $\eta_p^2$ and $\eta_p^3$;
23             Build a new $cd - path$ from $w_{end}$ to $w_1$, not passing through $v$;
24             Add the new $cd - path$ into $\eta_p^3$;
25             $p \leftarrow p + 1$
```

Figure 4.1: Pseudocode of the selection phase.

The algorithm in Figure 4.1 depicts the full selection process. First, let $p = 1$ (line 2) and let η^1 to be empty (line 3). The while-loop in lines 4-25 constructs, at each iteration, a pair of subgraphs η_p^2 and η_p^3 or, at its last iteration, the η^1 subgraph.

At each iteration, a $cd-path$ of length $2p$ having v as its middle vertex is constructed in line 6. The endpoints of this path are called v_1 and v_2. See Figure 4.2.

The path is then extended by setting w_1 as the vertex adjacent to v_2 such that $\mathcal{C}(v_2, w_1) = c$ and w_{end} as the vertex adjacent to v_1 such that $\mathcal{C}(v_1, w_{end}) = d$ (lines 7 and 8). Observe that, if $w_1 = w_{end}$, the path is now a cycle and the existence of a longer $cd - path$ having v as its middle vertex is not possible anymore. In this case, the subgraph η^1 is set as the $cd - cycle$ and the selection phase ends (lines 10 and 11), see Figure 4.3.

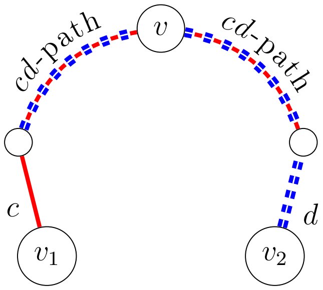


Figure 4.2: The subgraph $\eta_p^1 = \eta_p^2$ where the $cd - path$ from v_1 to v_2 has v as the middle vertex.

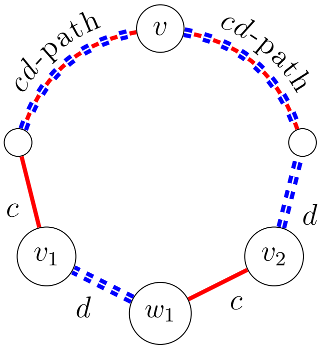


Figure 4.3: The subgraph η_p^1, obtained when $w_i = w_k$, after the inclusion of the vertex w_1 and edges (v_1, w_1) and (v_2, w_1).

If $w_1 \neq w_{end}$, subgraphs η_p^2 and η_p^3 are constructed (lines 14-24). First, the $cd - path$ built in line 6 is added to η_p^2. The while-loop in lines 16-21 determines a set of vertices W, including w_1 and w_{end}, that are connected to v_1 and v_2 with edges colored with the same subset of colors, with the exception of c and d which are the colors of (w_1, v_2) and (w_{end}, v_1). All vertices in W and the edges joining them to v_1 and v_2 are then added to η_p^2 and η_p^3, see Figures 4.4 and 4.5.

In lines 23 and 24 a $cd-path$ from w_{end} to w_1 not passing through v is built and added to η_p^3, as can be seen in Figure 4.6. To show that such a path always exists, observe that we have already built a $cd-path$ from w_1 to w_{end} passing through v and that the subgraph induced by the edges colored with c or d is a collection of $cd-cycles$. Therefore, there must be a way to reconnect w_{end} and w_1 through a new and disjoint $cd-path$. After this last step, both η_p^2 and η_p^3 are ready to be processed by the change phase and the algorithm increases the value of p to start a new while-loop iteration.

In the analysis of time complexity for the selection phase of TARS, we assume that the functions $adj(v, c)$ and $\mathcal{C}(v, w)$ take $O(1)$ time. The comparison $w_1 = w_{end}$, in line 9, is true when the addition of one more vertex is enough to close the $cd - path$ built in line 6. The loop of lines 4-25 runs until $\eta^1 \neq \varnothing$, which takes at most $O(n)$

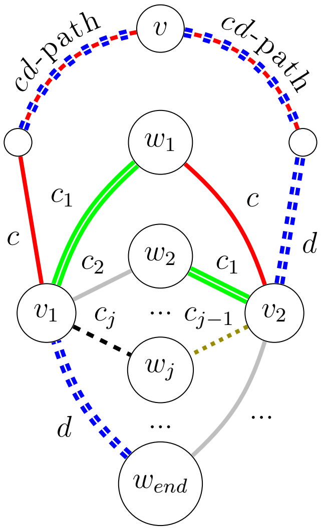


Figure 4.4: A final configuration of a subgraph η_p^2.

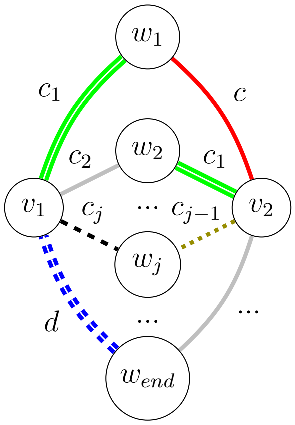


Figure 4.5: A subgraph η_p^3 obtained after the inclusion of all vertices in W and the edges joining them to v_1 and v_2.

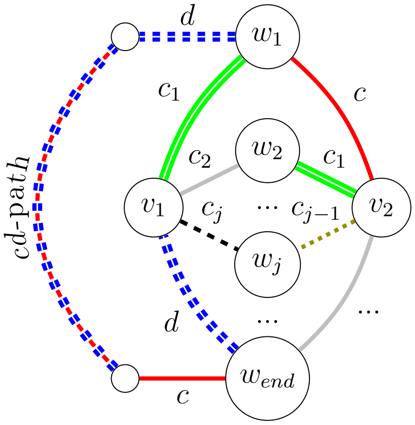


Figure 4.6: A configuration of a subgraph η_p^3 after the inclusion of the $cd - path$ starting from w_{end} and ending at w_1, not passing through v.

iterations since the maximum value of p is limited by ${}^n\!/_2 - 1$. The path constructed in line 6 can be extended from the path used in the previous iteration and takes $O(1)$ to be updated. The loop in lines 16-21 runs for at most n iterations since it processes one vertex at each iteration and cannot process a vertex twice. The construction of the path in line 23 takes $O(n)$ time. All other operations take constant time and the complexity of the entire algorithm is $O(n^2)$.

4.1.2 Change phase

Every subgraph η^1 and η_p^t, for $t \in \{2, 3\}$ and $1 \leq p \leq p_{max}$ obtained in the selection phase is used as input in the change phase in order to perform a move in the TARS neighborhood structure. All neighbor solutions created in this phase are evaluated and the one with best cost function value of the associated problem is returned.

```
   input : the subgraphs $\eta^1$ and $\eta_p^t$ of the colored graph $s$, for $2 \leq t \leq 3$ and
           $1 \leq p \leq p_{max}$
   output: a neighbor solution with the best cost function
 1 begin
 2     foreach subgraph $\eta_p^t$ do
 3         let $t$ be the superindex of the subgraph;
 4         if $t = 1$ then
 5             Perform a move in PRS neighborhood based on a subgraph of $\eta^1$;
 6         else if $t = 2$ then
 7             if $p = 1$ then
 8                 Perform a move in PTS neighborhood based on the subgraph of $\eta_1^2$;
 9             else
10                 Exchange the colors along the $cd-path$ from $v_1$ to $v_2$;
11                 foreach edge $e \in \eta_p^2$ incident to $v_1$ and not in the $cd-path$ do
12                     Let $w$ be the end vertex of $e$ such that $w \neq v_1$;
13                     Exchange the color assignment of $(v_1, w)$ and $(v_2, w)$;
14         else
15             Exchange the colors along the $cd-path$ from $w_{end}$ to $w_1$;
16             Let $fan(v_1) = \{(v_1, w_1), (v_1, w_2), \ldots, (v_1, w_{end})\}$;
17             Let $fan(v_2) = \{(v_2, w_1), (v_2, w_2), \ldots, (v_2, w_{end})\}$;
18             Rotate $fan(v_1)$ by shifting it backward;
19             Rotate $fan(v_2)$ by shifting it forward;
20         $s' \leftarrow$ neighbor solution obtained in the current iteration;
21         if cost function of $s'$ is the best so far then
22             $s^\star \leftarrow s'$;
23         return $s^\star$
```

Figure 4.7: Pseudocode of the change phase.

Figure 4.7 gives the pseudocode of the change phase. For each subgraph, it identifies the value of the superindex t in order to decide what kind of change will be applied to the subgraph under consideration (line 3). If $t = 1$ then it exchanges the colors c and d along the edges of η^1 in order to obtain a subgraph equivalent to the one in Figure 4.8. Note that this procedure (line 5) is equivalent to a move in PRS neighborhood, as depicted in Figure 3.4.

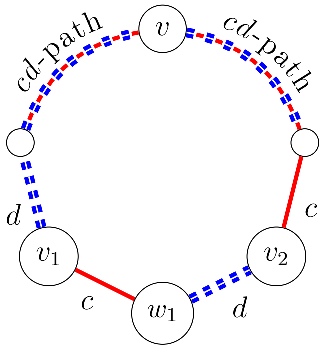
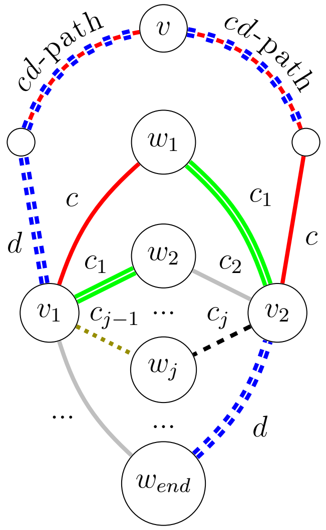


Figure 4.8: A resulting subgraph η^1 obtained in the change phase after exchanging the colors c and d along its edges.

If $t = 2$ and $p = 1$ then the move based on the subgraph η_1^2 is equivalent to a PTS move, as depicted in Figures 3.7 and 3.8 . In case $p \neq 1$ then, in line 10, the algorithm exchanges the colors c and d along the $cd - path$ starting from v_1 and ending at v_2. From line 11 and 13, it exchanges the color of the edges (v_1, w_i) and (v_2, w_i) for each vertex w_i in the set W determined in the selection phase. See Figure 4.9.



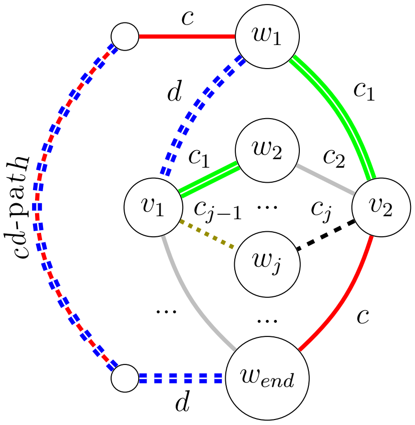


Figure 4.9: A resulting subgraph η_p^2 after exchanging the colors c and d in $cd-path$ and exchanging the color assignment of (v_1, w) and (v_2, w).

Figure 4.10: The resulting subgraph η_p^3 obtained after exchanging the colors c and d in $cd - path$ and rotating $fan(v_1)$ and $fan(v_2)$.

If $t = 3$ then, it exchanges the colors c and d along the $cd - path$ starting from w_{end} and ending at w_1 (line 15). Let $fan(v_1) = \{(v_1, w_1), (v_1, w_2), \ldots, (v_1, w_{end})\}$ and $fan(v_2) = \{(v_2, w_1), (v_2, w_2), \ldots, (v_2, w_{end})\}$ such that $\mathcal{C}(v_1, w_j) = c_j$ and $\mathcal{C}(v_2, w_j) = c_{j-1}$. Next, the algorithm rotates $fan(v_1)$ and $fan(v_2)$ by shifting $fan(v_1)$ backward and shifting $fan(v_2)$ forward, see Figure 4.10.

The rest of the algorithm keeps track of the best neighbor solution and returns it once all neighbors were generated and evaluated according to the problem objective function.

The algorithm processes $2p_{max} + 1$ subgraphs. Since $p_{max} \leq {}^{n}/_{2} - 1$, the outer loop executes $O(n)$ iterations. A move in PRS or in PTS can be performed in $O(n)$. The exchange of colors of the $cd - paths$ in lines 10 and 15, as well as the loop of lines 11-13, can be implemented in $O(n)$. The rotation of each fan in lines 18 and 19 also takes $O(n)$ and the entire algorithm runs in $O(n^2)$ time.

Figure 4.11 shows a subgraph η_3^2 obtained from an edge-colored K_{12}, with parameters $v = 8$, color c represented by solid lines and color d represented by double-dashed lines. In this subgraph, we have $v_1 = 1$, $v_2 = 7$, $cd - path = \{(1,6),(6,10),(10,8),(8,11),(11,9),(9,7)\}$, $2p = |cd - path| = 6$, $w_1 = 4$, $w_2 = 10$, $w_3 = 5$ and $w_4 = w_{end} = 0$. Figure 4.12 gives the resulting subgraph after we exchange the colors along the $cd - path$, starting from $v_1 = 1$ and ending at $v_2 = 7$, and swap the assignment of colors of the edges joining v_1, v_2 and w_j, with $1 \leq j \leq 4$.

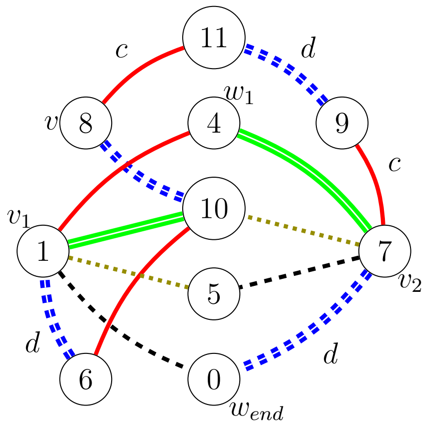


Figure 4.11: A subgraph η_3^2 obtained from an edge-colored K_{12}.

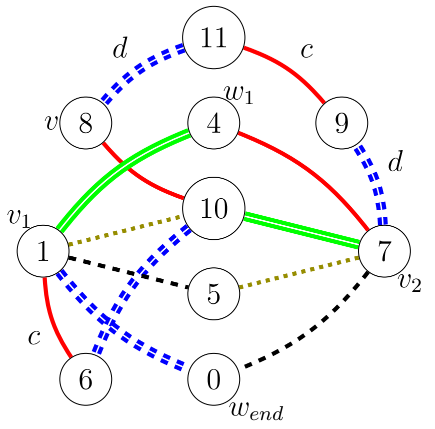
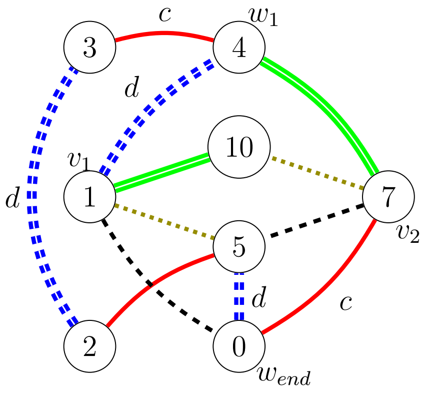


Figure 4.12: A resulting subgraph after we change the assignment of colors in η_3^2.

Figure 4.13 shows a subgraph η_3^3 obtained from the same edge-colored K_{12} with the same parameters used to build the subgraph η_3^2 in Figure 4.11. Note that η_3^2 and η_3^3 differ only by the $cd - path$ present in them. In this subgraph, we have $v_1 = 1$, $v_2 = 7$, $fan(v_1) = \{(1,4),(1,10),(1,5),(1,0)\}$, $fan(v_2) = \{(7,4),(7,10),(7,5),(7,0)\}$, $w_1 = 4$, $w_2 = 10$, $w_3 = 5$, $w_4 = w_{end} = 0$, and $cd - path = \{(0,5),(5,2),(2,3),(3,4)\}$. Figure 4.14 gives the resulting subgraph after we exchange the colors along the $cd - path$

starting from $w_4 = 0$ and ending at $w_1 = 4$ and rotate $fan(v_1)$ by shifting it backward and rotate $fan(v_2)$ by shifting it forward. Notice that the vertices w_j, for $1 \leq j \leq 4$, may belong to any $cd - path$ in η_p^t.



→

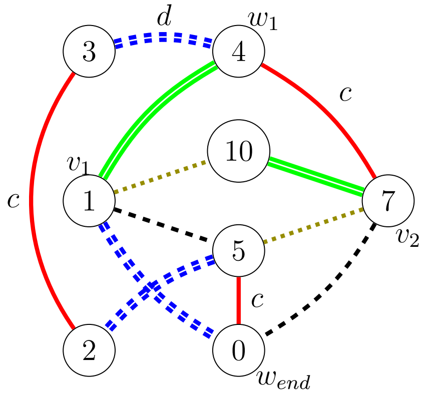


Figure 4.13: A subgraph η_3^3 obtained from an edge-colored K_{12}.

Figure 4.14: A resulting subgraph after we change the assignment of colors in η_3^3.

Figure 4.15 shows an example of how a move in TARS neighborhood structure changes the opponent assignment in a timetable. The timetable on the left was obtained with the circle method, for a tournament with $n = 12$. The teams were chosen in such a way to reflect the underlying subgraph represented in Figure 4.13. On the right, we have a resulting timetable after performing a move in TARS neighborhood structure that considers the subgraph represented in Figure 4.14. Note that such move is quite hard to derive and understand without the graph theory modeling used in this thesis.

4.1.3 Correctness of Teams and Rounds Swap

When applying a move in TARS neighborhood structure that is based on a subgraph η_p^t, the obtained neighbor $s' \in \text{TARS}(s)$ is a proper edge coloring of the graph. To show that, we analyze the changes of colors of the edges involved in the move.

Moves based on η^1 subgraphs are equivalent to PRS moves and are therefore valid.

For moves based on η_p^2 and η_p^3 subgraphs we analyze, for each vertex in the subgraph, the modification in the color assignment of its incident edges and certify that each color is still present in exactly one of them after the move.

First, we analyze η_p^2 subgraphs.

							r					
		0	1	2	3	4	5	6	7	8	9	10
	0	11	2	4	6	8	10	1	3	5	7	9
	1	10	11	3	5	7	9	0	2	4	6	8
	2	9	0	11	4	6	8	10	1	3	5	7
	3	8	10	1	11	5	7	9	0	2	4	6
	4	7	9	0	2	11	6	8	10	1	3	5
t	5	6	8	10	1	3	11	7	9	0	2	4
	6	5	7	9	0	2	4	11	8	10	1	3
	7	4	6	8	10	1	3	5	11	9	0	2
	8	3	5	7	9	0	2	4	6	11	10	1
	9	2	4	6	8	10	1	3	5	7	11	0
	10	1	3	5	7	9	0	2	4	6	8	11
	11	0	1	2	3	4	5	6	7	8	9	10

							r					
		0	1	2	3	4	5	6	7	8	9	10
	0	11	2	4	6	8	10	7	3	1	5	9
	1	4	11	3	10	7	9	5	2	0	6	8
	2	9	0	11	4	6	8	10	1	5	3	7
	3	8	10	1	11	5	7	9	0	4	2	6
	4	1	9	0	2	11	6	8	10	3	7	5
t	5	6	8	10	7	3	11	1	9	2	0	4
	6	5	7	9	0	2	4	11	8	10	1	3
	7	10	6	8	5	1	3	0	11	9	4	2
	8	3	5	7	9	0	2	4	6	11	10	1
	9	2	4	6	8	10	1	3	5	7	11	0
	10	7	3	5	1	9	0	2	4	6	8	11
	11	0	1	2	3	4	5	6	7	8	9	10

Figure 4.15: An example of a move in TARS neighborhood structure for a tournament with $n = 12$.

- Interior vertices of $cd - path$:

  Note that all these vertices have an incident edge colored with c and an incident edge colored with d that are part of $cd - path$. Therefore, by exchanging colors c and d the colors assigned to edges incident to those vertices remain the same.

- Vertices v_1 and v_2:

  These are the end vertices of the $cd - path$. After exchanging the colors in the edges of this path, the color c will not be assigned to any edge incident to v_1 and will be assigned to two edges incident to v_2. An equivalent situation occurs with color d considering vertices v_1 and v_2 in reverse order. After exchanging the color of the remaining edges in η_p^2, the situation is fixed. By construction of η_p^2, every color present in an edge incident to v_1 is also present in an edge incident to v_2. Therefore, after exchanging colors, the set of colors present in edges incident to v_1 (as well as those present in edges incident to v_2) are the same.

- Vertices $w_j \in W$:

  For any vertex w_j, two incident edges exchange their color assignments. The colors assigned to edges incident to those vertices remain the same.

Finally, we analyze moves based on η_p^3 subgraphs.

- Interior vertices of $cd - path$:

  The situation is equivalent to the correspondent case for η_p^2 subgraphs.

- Vertices v_1 and v_2:
  These vertices are the center of the fans. Note that, since the shifts are circular, the set of colors incident to them is unchanged after the rotation.

- Vertices $w_j \in W \setminus \{w_1, w_{end}\}$:
  Observe that, for these vertices w_j, it holds that $\mathcal{C}(v_1, w_j) = c_j$ and $\mathcal{C}(v_2, w_j) = c_{j-1}$. After the shift of both fans the new coloring satisfies that $\mathcal{C}(v_1, w_j) = c_{j-1}$ and $\mathcal{C}(v_2, w_j) = c_j$. All other edges incident to w_j are unchanged, therefore, all colors remain present at it.

- Vertices w_1 and w_{end}:
  For vertex w_1, it holds that $\mathcal{C}(v_1, w_1) = c_1$ and $\mathcal{C}(v_2, w_1) = c$. After rotating both fans it holds that $\mathcal{C}(v_1, w_1) = d$ and $\mathcal{C}(v_2, w_1) = c_1$. Now, in the set of edges incident to w_1, the color c is absent and color d is repeated. By exchanging the colors of $cd-path$, one edge changes its color from d to c restoring the correctness of the coloring. An equivalent analysis shows that the new coloring is also proper in regard to w_{end}.

4.2 A study in connectivity of TARS neighborhood structure

In general, good neighborhood structures offer a high search capability and consequently lead to good results largely independent of the initial solution while the search performance induced by weak neighborhood structures is often highly correlated to the initial solution [Papadimitriou and Steiglitz, 1998].

Chapter 3 exposed conditions in which the existing neighborhood strucures suffer from the issue of non-connectivity. In this chapter, we show that the proposed TARS neighborhood structure is able to get around the issue of non-connectivity, regardless the initial solution or the number of participating teams. Figure 4.16 shows a perfect one-factorization of K_{12}; each row is a one-factor and each pair of vertices indicates an edge. Thus, the first line specifies the one-factor $\{(0,11),(1,10),(2,9),(3,8),(4,7),(5,6)\}$. The union of any two factors builds a hamiltonian cycle, as the one depicted in Figure 4.18, obtained by considering its two first factors. Also, for any pair of vertices, the smallest set of factors in which they share all its adjacent vertices has size $n-2$.

(0,11),	(1,10),	(2,9),	(3, 8),	(4, 7),	(5, 6)
(0,9),	(1, 8),	(2, 7),	(3, 6),	(4, 5),	(10,11)
(0, 7),	(1, 6),	(2, 5),	(3, 4),	(8,10),	(9,11)
(0, 5),	(1, 4),	(2, 3),	(6,10),	(7,9),	(8,11)
(0, 3),	(1, 2),	(4,10),	(5,9),	(6, 8),	(7,11)
(0, 1),	(2,10),	(3,9),	(4, 8),	(5, 7),	(6,11)
(0,10),	(1,9),	(2, 8),	(3, 7),	(4, 6),	(5,11)
(0, 8),	(1, 7),	(2, 6),	(3, 5),	(4,11),	(9,10)
(0, 6),	(1, 5),	(2, 4),	(3,11),	(7,10),	(8,9)
(0, 4),	(1, 3),	(2,11),	(5,10),	(6,9),	(7, 8)
(0, 2),	(1,11),	(3,10),	(4,9),	(5, 8),	(6, 7)

Figure 4.16: A perfect one-factorization of K_{12}.

(0,11),	**(1, 4)**,	(2,9),	(3, 8),	(5, 6),	**(7,10)**
(0,9),	(1, 8),	(2, 7),	(3, 6),	(4, 5),	(10,11)
(0, 5),	(1, 6),	**(2, 3)**,	**(4, 7)**,	(8,10),	(9,11)
(0, 1),	**(2, 5)**,	**(3, 4)**,	(6,10),	(7,9),	(8,11)
(0, 3),	(1, 2),	(4,10),	(5,9),	(6, 8),	(7,11)
(0, 7),	**(1, 5)**,	(2,10),	(3,9),	(4,8),	(6,11)
(0,10),	(1,9),	(2, 8),	(3, 7),	(4, 6),	(5,11)
(0, 8),	(1, 7),	(2, 6),	(3, 5),	(4,11),	(9,10)
(0, 6),	**(1,10)**,	(2, 4),	(3,11),	**(5, 7)**,	(8,9)
(0, 4),	(1, 3),	(2,11),	(5,10),	(6,9),	(7, 8)
(0, 2),	(1,11),	(3,10),	(4,9),	(5, 8),	(6, 7)

Figure 4.17: A one-factorization obtained after a move in TARS neighborhood structure.

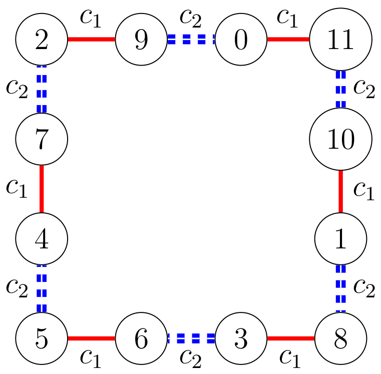


Figure 4.18: A Hamiltonian cycle built using the first two factors of a perfect one factorization of K_{12}.

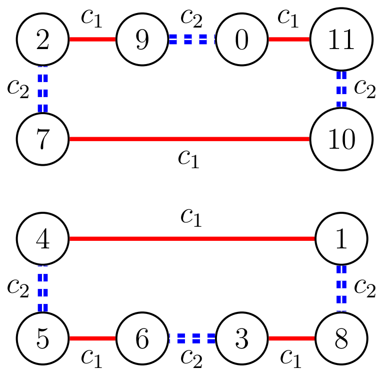


Figure 4.19: After a move in TARS neighborhood, the two first factors form two distinct cycles, showing that the new factorization is no longer perfect.

Figure 4.17 shows a one-factorization of K_{12} obtained from the perfect one-factorization of K_{12} depicted in Figure 4.16, after a single TARS move. A graphical representation of such move is given by Figures 4.13 and 4.14.

In Figure 4.19 we have the subgraph induced by the two first factors from Figure 4.17. As one can see, the two factors do not form a hamiltonian cycle, which shows that the new factorization is no longer perfect. Therefore, by this example, we show that TARS increases the connectivity of the solution space of SRR sport scheduling problems.

For any schedule s, all schedules s' that are connected to s through PTS, PRS, TS and RS moves, are also connected to s through a TARS move. Furthermore, Figures 4.16 and 4.17 show that there is at least one pair of schedules s and s' not connected by PRS or PTS, which are connected by TARS. Whether the solution space is fully connected by TARS is still an open issue.

Chapter 5

Computational Results

This chapter presents a series of experimental analysis that were carried out in order to evaluate the neighborhood structures described in this thesis. We describe two sport scheduling problems: the *Weighted Carry-over Effects Value Minimization Problem* (WCOEVMP), proposed by [Guedes and Ribeiro, 2011], and the *Traveling Tournament Problem with Predefined Venues* (TTPPV), proposed by [Melo et al., 2009]. An ILS heuristic is applied to both problems. At the end of this chapter, we present some computational results that confirm the theoretical foundation introduced in the previous chapters.

5.1 The Weighted Carry-Over Effects Values Minimization Problem

The *Carry-over effects value* (COEV) is one of the various measures that can be considered when assessing the quality of a round-robin tournament schedule. If team a plays against team b and c in two consecutive rounds, team c is said to receive a carry-over effect from team b. The rounds are considered cyclically, i.e., round $n-1$ is followed by the first round. Let C_{ab} denote the number of times team b receives a carry-over effect from team a during a tournament. Hence the COEV of an n-team SRR schedule is given by COEV $= \sum_{a=1}^{n} \sum_{b=1}^{n} {C_{ab}}^2$.

If team b is much stronger than the other competitors, then team c will possibly take some advantage over team a in their game due to the great effort team a went through in its previous game. This sort of situation, in which one of the teams might be benefited, should be avoided or at least minimized. The problem of minimizing the COEV in SRR schedules was originally proposed by [Russell, 1980].

Based on the schedule described in Figure 5.1 (a), one may count the number of carry-over effects each team gives to every other in an SRR tournament and build the carry-over effects matrix, as shown in Table 5.1 (b). Each entry (a, b) in this matrix indicates the number of carry-over effects team a gives to team b. An ideal schedule is one in which all non-diagonal elements of the corresponding matrix are equal to one.

teams	rounds				
	0	1	2	3	4
0	−4	−5	+3	−1	−2
1	−3	+4	−2	+0	−5
2	−5	+3	+1	−4	+0
3	+1	−2	−0	−5	+4
4	+0	−1	+5	+2	−3
5	+2	+0	−4	+3	+1

teams	teams					
	0	1	2	3	4	5
0	0	1	0	0	1	3
1	0	0	3	0	1	1
2	3	0	0	1	1	0
3	1	3	0	0	1	0
4	1	1	1	1	0	1
5	0	0	1	3	1	0

Figure 5.1: (a) A tournament schedule for 6 teams comprising 5 rounds. (b) A carry-over effects matrix.

In the original problem, all carry-over effects have the same weight unit. Taking into account the different strength of teams in real competitions, [Guedes and Ribeiro, 2011] proposed the weighted variant of the problem called *Weighted Carry-over Effects Value Minimization Problem* (WCOEVMP). For instance, a team with a small number of good players may be more susceptible to the carryover effect than a team with a lot of good players, which can more easily deal with injuries or suspensions, or even simply put a completely fresh team on the pitch after a difficult game. In this approach, we assign a weight w_{ab} to every ordered pair (a, b) of teams, based on their relative strengths or handicaps, and minimize the total weighted carry-over effects value. Therefore, the weighted COEV of an SRR schedule with n participating teams is given by the following equation:

$$\text{WCOEV} = \sum_{a=1}^{n} \sum_{b=1}^{n} w_{ab} \times {C_{ab}}^2 \tag{5.1}$$

[Goossens and Spieksma, 2012a] claim that, in general, a football team cannot be considered at a disadvantage because of a schedule that does not balance carryover effects. However, it does not exclude that there may exist specific circumstances where the carryover effect may have an influence.

5.2 The Traveling Tournament Problem with Predefined Venues

The Traveling Tournament Problem with Predefined Venues (TTPPV) [Melo et al., 2009] is a variant of the Traveling Tournament Problem (TTP) [Easton et al., 2001] and consists in scheduling an SRR tournament in which the home team of each game is known beforehand. A game between teams i and j can be represented by the ordered pair (i, j) or by the ordered pair (j, i). In the first case, the game between teams i and j takes place at the venue of team i; otherwise, at the venue of team j. For instance, the home-away assignment for the tournament schedule presented in Figure 5.1 (a) can be represented by $\{(4, 0), (5, 0), (0, 3), (1, 0), (2, 0), (3, 1), (1, 4), (2, 1), (5, 1), (5, 2), (2, 3), (4, 2), (5, 3), (3, 4), (4, 5)\}$.

Given an $n \times n$ distance matrix, assume that each team is at its home venue at the beginning of the tournament and that it returns there after playing its last game. The goal of the problem is to minimize the total distance traveled by all teams while no team plays more than three consecutive home-games or three consecutive away-games.

The circle method and vizing algorithm are used by [Costa et al., 2012] in order to build initial solutions for the TTPPV. Both construction procedures and some moves in neighborhood structures for SRR tournament problems may build schedules that violate the limits on the maximum number of consecutive home or away-games.

teams	rounds				
	0	1	2	3	4
0	-4	-5	+3	-1	-2
1	-3	+4	-2	+0	-5
2	-5	+3	+1	-4	+0
3	+1	-2	-0	-5	+4
4	+0	-1	+5	+2	-3
5	+2	+0	-4	+3	+1

⟶

teams	rounds				
	0	1	2	3	4
0	**-4**	**-5**	**-2**	**-1**	+3
1	-3	+4	-5	+0	-2
2	-5	+3	+0	-4	+1
3	+1	-2	+4	-5	-0
4	+0	-1	-3	+2	+5
5	**+2**	**+0**	**+1**	**+3**	-4

Figure 5.2: A neighbor solution obtained after a move in RS neighborhood having rounds 2 and 4 as parameters.

Figure 5.2 shows a neighbor schedule obtained after a move in RS neighborhood, considering the tournament scheduled presented in Figure 5.1 (a) and rounds 2 and 4 as parameters. After the move, team 0 has four away-games in a row while team 5 has four home-games in a row. Thus, this neighbor solution has two violations.

An ILS heuristic for the TTPPV was proposed in [Costa et al., 2012]. The local search algorithm is allowed to work with solutions with violations in the number of consecutive home or away-games. They showed that, for several values of n, ordered one-factorizations obtained by the circle method should not be used to build initial solutions. Instead, using vizing algorithm for constructing the initial solutions led to better solutions as the authors were able to obtain better results for the problem than the ones previously published in [Melo et al., 2009].

5.3 Experimental results

In this work, we heuristically tackle the WCOEVMP and the TTPPV with a simple heuristic based on the ILS heuristic proposed in [Guedes and Ribeiro, 2011], depicted in Figure 5.3. The main difference from our version of the ILS to the one proposed by [Guedes and Ribeiro, 2011] is the stopping criterion. While they use a fixed number of iterations for each instance, we fix the wall-clock runtime of our experiments to $n^3/2$ seconds. Otherwise, we kept all other features of the original version. The experimental results aim to validate the theoretical findings presented in the previous chapters.

```
1  begin
2      while $runtime < n^3/2$ seconds do
3          Build a solution $s$;
4          repeat
5              Obtain a solution $s'$ by applying a perturbation to $s$;
6              Apply local search to $s'$;
7              Replace $s$ by $s'$ using an acceptance criterion;
8              Update the best-known solution $s^\star$;
9          until non-improving condition is met;
10     return $s^*$
```

Figure 5.3: Iterated Local Search heuristic.

The ILS heuristic depicted in Figure 5.3 starts by constructing an initial solution s with the circle method or vizing algorithm (line 3). An inner loop starts in line 4. The solution s is then perturbed by applying up to five random moves of PRS or PTS neighborhood, randomly chosen (line 5). Then, a local search procedure is applied to the perturbed solution s', using either PRS$\cup$PTS or TARS, with the best improvement strategy, that is, moving the search from the current solution to the best neighbor solution until there is not neighbor solution that improves s' (line 6). Next, the solution s', is compared with the current solution s. The current solution s is replaced by s', in line 7, if the cost function of the latter is less than or equal to $(1+b)$ times the cost function of s. The value of the variable b starts with 0.01 and it doubles after every $2 \times n$ iterations without an update of the current solution. The variable b is reset to its initial value whenever the current solution is updated. Finally, in line 8, the best-known solution s^* is updated if the cost function of s' is smaller than the cost function of $s^\star$. The inner loop stops in line 9 after a maximum number of 1000 non-improving moves have been accepted since the last time the best solution was updated. The same parameter setup was defined for both case study problems.

Computational experiments were performed on an Intel® Xeon® E5-2660 machine with 2.20 GHz and 128 GB of RAM memory. The code was implemented in C++ and compiled with a GNU GCC compiler version 4.9.2 under Ubuntu Linux 14.10. The same set of instances considered in [Guedes and Ribeiro, 2011] for the WCOEVMP and in [Costa et al., 2012] for the TTPPV were used in our experiments. Average and best values were obtained over 30 independent runs of the ILS heuristic for each instance.

Tables 5.1, 5.2 and 5.6 provide an experimental evaluation of the performance of the studied neighborhood structures. The first column shows the instance sizes while the second column gives the instance names. For each instance, an initial solution is obtained with a constructive method. Next, a local search based on the best improvement strategy is applied until a local optimum is found. The third, fourth, fifth and sixth columns give the average local optimum values over 30 repetitions of such procedure. The last two columns give the relative difference between PRS∪PTS and TARS, computed as $\frac{\lambda-\mu}{\mu} \times 100\%$, where λ stands for the values obtained with PRS∪PTS, and μ stands for the values obtained with TARS. The performance of the local search using TARS is considered superior whenever the relative difference is positive, and the performance of the local search using PRS∪PTS is superior otherwise.

The numerical results obtained with ILS heuristic are presented in Tables 5.3, 5.4, 5.5, 5.7 and 5.8. The first column shows the instance sizes while the second column gives the instance names. The third one shows the best-known results for the WCOEVMP as reported in [Guedes and Ribeiro, 2011] or for the TTPPV as reported in [Costa et al., 2012]. Fourth and fifth columns show the best results obtained by the ILS heuristic based on the PRS∪PTS neighborhood structure when the circle method and vizing method were respectively used to build initial solutions. The next two columns display the average results, considering both versions of the ILS heuristic for each instance. The next four columns show the same information for the ILS heuristic based on TARS neighborhood structure. The next two columns present the lower and upper bounds of the confidence interval obtained for the difference of averages observed for the ILS heuristic based on the PRS∪PTS neighborhood structure and the ILS heuristic based on TARS neighborhood structure when both heuristics use the circle method to build initial solutions. The last two columns show the same information when using the vizing algorithm to build initial solutions.

A t-test for paired observations, as presented in [Jain, 1991], was applied in order to compare the different configurations of the ILS. A 99.9% confidence level was chosen. 30 experiments on each configuration were conducted such that there is a one-to-one correspondence between the ith test on each configuration. For each pair of configuration, the difference in performance can be computed and a confidence interval can be constructed for the difference. If the confidence interval includes zero, the systems are not significantly different.

Whenever the lower-bound of the confidence interval is greater than zero, one can be 99.9% confident that the ILS heuristic based on TARS neighborhood structure outperforms the same ILS heuristic using the existing neighborhood structures for the given instance and for the given method that is used to build an initial solution.

5.3.1 Numerical results for the Weighted Carry-Over Effects Values Minimization Problem

Four classes of instances for the WCOEVPM, generated by [Guedes and Ribeiro, 2011], were used in order to evaluate the performance of the studied neighborhoods. The weights of the Random instances are randomly generated in the interval $[1, 2n]$. Three matrices identified by letters A, B and C were generated for each value of n. For the Linear instances, a strength in the interval $[1, n]$ is assigned to each team. For simplicity, the strength of team i is made equal to i. Each weight w_{ij} is defined as the absolute value of the difference between the strengths of teams i and j. For the real-life inspired instances, six instances are derived from six consecutive issues of the Brazilian football championship. The strength of each team is given by the number of points it obtained in the previous year. As with the linear instances, each weight w_{ij} is defined as the absolute value of the difference between the strengths of teams i and j. For the Perturbed instances, each weight of a linear instance is increased by an individual perturbation, randomly generated in the interval $[-n/2, n/2]$. Three different instances identified by letters A, B and C are generated for each value of n. Absolute values are taken whenever the perturbation leads to a negative weight.

As can be seen from Tables 5.1 and 5.2, the performance of the local search using TARS is slightly superior when vizing method is used to build the initial solutions, for almost all considered instances. The relative difference is greater when the circle method is considered for building the initial solutions, specially for those values of instance sizes listed in [OEIS, 2015], as 12, 14 and 20.

For the Random instances in Table 5.3, when the circle method is used to build the initial solutions and for any instance of size greater than or equal to 12, TARS outperforms the existing neighborhood structures. The proposed algorithm was able to improve the best-known solution for 14 of the 27 evaluated instances. Besides the draw in two results for instances of size 10, the ILS version using the existing neighborhoods found two best results for inst12randomB and inst16randomB. Note that those results were obtained when the vizing algorithm was used to build the initial solutions.

Table 5.4 gives the numerical results for the classes of linear instances and real-life inspired instances. For linear instances, regardless the initial solution, TARS outperforms the existing neighborhood structures for instances of sizes greater than 12. For the real-life inspired instances, TARS outperforms the existing neighborhood structures in all scenarios, except the one with instances of size 24 and when the initial solution is obtained with the circle method.

For the Perturbed instances in Table 5.4, although the numerical results present an outlier, for instance, inst16perturbedlinearA, TARS still outperforms the existing neighborhood structures at 99.9% confidence level for instances of sizes greater than 12, regardless the initial solution used.

5.3.2 Numerical results for the Traveling Tournament Problem with Predefined Venues

Two sets of instances for the TTPPV, generated by [Costa et al., 2012], were used in order to evaluate the performance of the studied neighborhoods in Table 5.6 and in the computational experiments presented in Tables 5.7 and 5.8. In these instances, the distance between team i and j, for $i > j$ is given by $min\{i-j, j-i+n\}$, which are the same as in instances circ18 and circ20 of the TTP.

As can be seen from Table 5.6, the performance of the local search using TARS is superior regardless the method used to build the initial solutions, for all considered instances. The relative difference is at least 17.1% when the circle method is considered for building the initial solutions for instances of size 20.

For instances of size 18 of the TTPPV, TARS was able to improve the best-known solution for 13 of the 17 evaluated instances. For all instances of size 20 of the TTPPV, TARS outperforms the existing neighborhood with a 99.9% confidence level for all 18 evaluated instances. This result was expected since 20 is a special case of

Instance		PRS∪PTS		TARS		Relative difference	
Size	Name	Circle	Vizing	Circle	Vizing	Circle	Vizing
4	inst4randomA	63.0	63.0	63.0	63.0	0.0%	0.0%
4	inst4randomB	53.0	53.0	53.0	53.0	0.0%	0.0%
4	inst4randomC	45.0	45.0	45.0	45.0	0.0%	0.0%
6	inst6randomA	233.0	233.0	233.0	233.0	0.0%	0.0%
6	inst6randomB	274.6	274.5	274.6	274.5	0.0%	0.0%
6	inst6randomC	244.4	242.0	244.0	241.6	0.2%	0.2%
8	inst8randomA	553.5	564.3	564.9	556.9	-2.0%	1.3%
8	inst8randomB	572.4	552.6	544.3	552.4	5.2%	0.0%
8	inst8randomC	545.5	554.5	523.4	549.9	4.2%	0.8%
10	inst10randomA	1176.5	1184.7	1123.6	1133.8	4.7%	4.5%
10	inst10randomB	1031.6	1034.0	996.7	975.4	3.5%	6.0%
10	inst10randomC	976.5	977.9	931.9	935.8	4.8%	4.5%
12	inst12randomA	2098.3	1925.0	1832.9	1836.6	14.5%	4.8%
12	inst12randomB	2067.9	1857.7	1789.8	1784.0	15.5%	4.1%
12	inst12randomC	2197.6	1930.8	1915.9	1866.2	14.7%	3.5%
14	inst14randomA	3430.1	3108.8	2990.8	2957.8	14.7%	5.1%
14	inst14randomB	3745.1	3499.4	3317.2	3275.5	12.9%	6.8%
14	inst14randomC	3570.1	3292.7	3223.3	3219.5	10.8%	2.3%
16	inst16randomA	4748.3	4538.0	4429.9	4359.6	7.2%	4.1%
16	inst16randomB	4678.9	4569.4	4384.8	4362.1	6.7%	4.8%
16	inst16randomC	4723.3	4539.1	4382.3	4326.4	7.8%	4.9%
18	inst18randomA	7024.3	6629.9	6393.4	6251.5	9.9%	6.1%
18	inst18randomB	7351.9	6978.8	6754.6	6675.9	8.8%	4.5%
18	inst18randomC	7092.3	6576.5	6387.3	6207.2	11.0%	6.0%
20	inst20randomA	9932.9	9316.2	8909.3	8628.5	11.5%	8.0%
20	inst20randomB	10196.0	9440.9	9050.3	8853.2	12.7%	6.6%
20	inst20randomC	9822.2	9139.3	8797.2	8773.0	11.7%	4.2%
4	inst4linear	20.0	20.0	20.0	20.0	0.0%	0.0%
6	inst6linear	114.0	114.0	114.0	114.0	0.0%	0.0%
8	inst8linear	178.1	185.3	176.8	181.9	0.7%	1.9%
10	inst10linear	405.0	403.6	383.2	391.3	5.7%	3.2%
12	inst12linear	783.7	688.3	660.3	680.2	18.7%	1.2%
14	inst14linear	1225.8	1078.7	1042.0	1043.6	17.6%	3.4%
16	inst16linear	1675.7	1620.8	1561.5	1506.1	7.3%	7.6%
18	inst18linear	2454.4	2299.1	2185.7	2162.7	12.3%	6.3%
20	inst20linear	3471.3	3085.1	2960.9	2942.1	17.2%	4.9%

Table 5.1: Performance evaluation of the existing neighborhoods for the Weighted Carry-Over Effects Value Minimization Problem. The local optimum values were obtained after applying each local search procedure to solutions obtained with constructive methods presented in this thesis.

Instance		PRS∪PTS		TARS		Relative difference	
Size	Name	Circle	Vizing	Circle	Vizing	Circle	Vizing
4	inst4perturbedlinearA	16.0	16.0	16.0	16.0	0.0%	0.0%
4	inst4perturbedlinearB	17.0	17.0	17.0	17.0	0.0%	0.0%
4	inst4perturbedlinearC	22.0	22.0	22.0	22.0	0.0%	0.0%
6	inst6perturbedlinearA	72.2	72.2	72.2	71.9	0.0%	0.4%
6	inst6perturbedlinearB	73.0	73.0	73.0	73.0	0.0%	0.0%
6	inst6perturbedlinearC	60.0	60.0	60.0	60.0	0.0%	0.0%
8	inst8perturbedlinearA	173.7	176.1	171.1	173.0	1.5%	1.8%
8	inst8perturbedlinearB	180.5	176.9	172.2	177.2	4.8%	-0.2%
8	inst8perturbedlinearC	195.1	188.5	185.2	185.6	5.3%	1.6%
10	inst10perturbedlinearA	434.7	432.3	409.2	415.9	6.2%	3.9%
10	inst10perturbedlinearB	354.2	357.8	336.3	335.5	5.3%	6.6%
10	inst10perturbedlinearC	392.4	392.3	374.8	375.6	4.7%	4.4%
12	inst12perturbedlinearA	872.6	768.8	719.1	731.3	21.4%	5.1%
12	inst12perturbedlinearB	774.3	683.2	643.1	654.6	20.4%	4.4%
12	inst12perturbedlinearC	785.9	664.3	637.5	629.3	23.3%	5.6%
14	inst14perturbedlinearA	1341.8	1150.1	1077.8	1069.7	24.5%	7.5%
14	inst14perturbedlinearB	1304.9	1108.9	1034.5	1027.2	26.1%	8.0%
14	inst14perturbedlinearC	1323.8	1150.2	1112.7	1094.8	19.0%	5.1%
16	inst16perturbedlinearA	1846.8	1784.5	1705.7	1670.8	8.3%	6.8%
16	inst16perturbedlinearB	1767.9	1714.8	1631.2	1595.4	8.4%	7.5%
16	inst16perturbedlinearC	1528.0	1473.4	1374.3	1369.1	11.2%	7.6%
18	inst18perturbedlinearA	2842.3	2541.9	2447.1	2404.9	16.2%	5.7%
18	inst18perturbedlinearB	2645.6	2408.5	2290.1	2293.1	15.5%	5.0%
18	inst18perturbedlinearC	2546.2	2275.7	2117.8	2107.4	20.2%	8.0%
20	inst20perturbedlinearA	4002.1	3574.0	3486.4	3363.8	14.8%	6.3%
20	inst20perturbedlinearB	3883.2	3342.5	3200.5	3173.6	21.3%	5.3%
20	inst20perturbedlinearC	3785.0	3281.2	3110.9	3041.5	21.7%	7.9%
24	inst24brazil2003	10136.7	9741.2	9441.1	9200.8	7.4%	5.9%
24	inst24brazil2004	10377.2	9723.5	9543.0	9258.3	8.7%	5.0%
22	inst22brazil2005	6898.1	6670.6	6525.3	6327.9	5.7%	5.4%
20	inst20brazil2006	6978.7	6670.0	6400.6	6340.7	9.0%	5.2%
20	inst20brazil2007	6140.7	5865.7	5736.6	5622.5	7.0%	4.3%
20	inst20brazil2008	6285.1	5705.1	5402.5	5329.4	16.3%	7.0%

Table 5.2: Performance evaluation of the existing neighborhoods for the Weighted Carry-Over Effects Value Minimization Problem. The local optimum values were obtained after applying each local search procedure to solutions obtained with constructive methods presented in this thesis.

			PRS∪PTS				TARS				Confidence interval			
Instance			Best result		Average		Best result		Average		Circle		Vizing	
Size	Name	BKS	Circle	Vizing	Circle	Vizing	Circle	Vizing	Circle	Vizing	LB	UB	LB	UB
4	inst4randomA	63	63	63	63.0	63.0	63	63	63.0	63.0	0.00	0.00	0.00	0.00
4	inst4randomB	53	53	53	53.0	53.0	53	53	53.0	53.0	0.00	0.00	0.00	0.00
4	inst4randomC	45	45	45	45.0	45.0	45	45	45.0	45.0	0.00	0.00	0.00	0.00
6	inst6randomA	233	233	233	233.0	233.0	233	233	233.0	233.0	0.00	0.00	0.00	0.00
6	inst6randomB	274	274	274	274.0	274.0	274	274	274.0	274.0	0.00	0.00	0.00	0.00
6	inst6randomC	235	235	235	235.0	235.0	235	235	235.0	235.0	0.00	0.00	0.00	0.00
8	inst8randomA	505	505	505	505.0	505.0	505	505	505.0	505.0	0.00	0.00	0.00	0.00
8	inst8randomB	495	495	495	495.0	495.0	495	495	495.0	495.0	0.00	0.00	0.00	0.00
8	inst8randomC	470	470	470	470.0	470.0	470	470	470.0	470.0	0.00	0.00	0.00	0.00
10	inst10randomA	895	895	895	909.9	905.1	895	895	915.3	910.8	-9.00	-1.80	-9.69	-1.84
10	inst10randomB	781	*__778__	*__778__	791.0	787.1	*__778__	*__778__	795.0	792.3	-6.81	-1.19	-7.34	-2.92
10	inst10randomC	737	*__714__	*__714__	721.1	719.9	*__714__	*__714__	732.2	729.0	-16.88	-5.38	-14.34	-3.73
12	inst12randomA	1521	1584	1505	1656.6	1550.5	1500	*__1486__	1558.9	1552.8	92.06	103.48	-7.40	2.73
12	inst12randomB	1489	1529	*__1468__	1600.5	1504.0	1478	1471	1516.0	1509.3	76.40	92.74	-6.86	-3.67
12	inst12randomC	1572	1650	1541	1696.3	1582.6	1541	*__1535__	1587.7	1582.8	105.90	111.37	-5.11	4.85
14	inst14randomA	2608	2648	2470	2738.8	2538.6	2475	*__2464__	2530.7	2531.4	196.82	219.24	2.27	12.13
14	inst14randomB	2870	2891	2821	3006.6	2879.1	*__2778__	2828	2871.3	2888.7	126.31	144.29	-13.77	-5.57
14	inst14randomC	2760	2714	2676	2892.2	2786.8	2747	*__2674__	2792.7	2784.0	79.23	119.90	-5.09	10.69
16	inst16randomA	3756	3721	3740	3840.8	3823.2	3715	*__3623__	3787.1	3778.9	44.97	62.56	32.27	56.46
16	inst16randomB	3797	3722	*__3645__	3816.6	3800.5	3692	3653	3764.9	3762.6	42.01	61.25	25.63	50.04
16	inst16randomC	3755	3554	3602	3732.0	3706.9	3532	*__3497__	3648.4	3635.4	70.72	96.48	52.76	90.18
18	inst18randomA	5515	5403	5490	5632.9	5580.5	5301	*__5283__	5466.7	5473.4	144.86	181.07	83.58	130.75
18	inst18randomB	5745	5902	5836	6030.3	5988.3	5779	5780	5904.3	5914.7	119.50	132.63	63.10	84.10
18	inst18randomC	5548	5327	5316	5525.3	5483.7	5197	*__5158__	5381.7	5385.8	132.34	154.86	77.65	118.21
20	inst20randomA	7760	7969	7686	8218.2	7848.7	7431	*__7424__	7629.8	7646.8	561.04	615.70	178.22	225.65
20	inst20randomB	7888	8169	7826	8394.9	7983.1	7477	*__7419__	7772.4	7750.7	583.65	661.35	179.95	284.79
20	inst20randomC	7636	7873	7623	8172.9	7852.3	*__7486__	7526	7657.4	7688.9	481.12	549.81	146.43	180.37

Table 5.3: Numerical results for the Weighted Carry-Over Effects Value Minimization Problem. All values marked with * indicate new best results. BKS stands for best-known solution reported in [Guedes and Ribeiro, 2011].

			PRS∪PTS				TARS				Confidence interval			
Instance			Best result		Average		Best result		Average		Circle		Vizing	
Size	Name	BKS	Circle	Vizing	Circle	Vizing	Circle	Vizing	Circle	Vizing	LB	UB	LB	UB
4	inst4linear	20	20	20	20.0	20.0	20	20	20.0	20.0	0.00	0.00	0.00	0.00
6	inst6linear	114	114	114	114.0	114.0	114	114	114.0	114.0	0.00	0.00	0.00	0.00
8	inst8linear	168	168	168	168.0	168.0	168	168	168.0	168.0	0.00	0.00	0.00	0.00
10	inst10linear	318	318	318	318.0	318.0	318	318	318.0	319.3	0.00	0.00	-2.57	-0.10
12	inst12linear	496	628	496	632.3	517.4	496	496	518.6	521.2	106.64	120.69	-5.08	-2.52
14	inst14linear	958	944	812	950.7	850.8	⋆**796**	810	834.7	832.5	107.44	124.43	14.57	22.10
16	inst16linear	1076	1116	1200	1230.9	1258.5	1152	1128	1217.3	1204.4	4.69	22.64	49.26	59.01
18	inst18linear	1660	1656	1704	1746.2	1818.5	1668	⋆**1624**	1735.0	1730.4	3.15	19.25	78.02	98.12
20	inst20linear	2212	2542	2374	2579.7	2516.5	2228	2246	2376.3	2393.3	174.22	232.45	114.05	132.22
24	inst24brazil2003	7542	⋆**7494**	8228	7759.7	8446.5	7504	7752	7879.9	7974.9	-151.56	-88.84	444.48	498.72
24	inst24brazil2004	7088	7290	7834	7493.7	8218.5	7276	7182	7672.7	7671.3	-224.50	-133.50	522.75	571.65
22	inst22brazil2005	5158	5164	5416	5342.7	5546.1	5096	⋆**5008**	5246.3	5223.9	72.60	120.20	298.18	346.22
20	inst20brazil2006	5236	5528	5562	5656.3	5748.9	5358	5462	5511.9	5573.2	122.41	166.39	156.89	194.45
20	inst20brazil2007	4834	5018	4878	5071.0	5077.8	4742	⋆**4732**	4886.9	4885.5	135.94	232.33	175.92	208.75
20	inst20brazil2008	3944	4904	4266	4946.0	4555.3	3966	3968	4225.7	4245.0	652.84	787.69	280.50	340.17

Table 5.4: Numerical results for the Weighted Carry-Over Effects Value Minimization Problem. All values marked with ⋆ indicate new best results. BKS stands for best-known solution reported in [Guedes and Ribeiro, 2011].

Instance			PRS∪PTS				TARS				Confidence interval			
			Best result		Average		Best result		Average		Circle		Vizing	
Size	Name	BKS	Circle	Vizing	Circle	Vizing	Circle	Vizing	Circle	Vizing	LB	UB	LB	UB
4	inst4perturbedlinearA	16	16	16	16.0	16.0	16	16	16.0	16.0	0.00	0.00	0.00	0.00
4	inst4perturbedlinearB	17	17	17	17.0	17.0	17	17	17.0	17.0	0.00	0.00	0.00	0.00
4	inst4perturbedlinearC	22	22	22	22.0	22.0	22	22	22.0	22.0	0.00	0.00	0.00	0.00
6	inst6perturbedlinearA	68	68	68	68.0	68.0	68	68	68.0	68.0	0.00	0.00	0.00	0.00
6	inst6perturbedlinearB	73	73	73	73.0	73.0	73	73	73.0	73.0	0.00	0.00	0.00	0.00
6	inst6perturbedlinearC	60	60	60	60.0	60.0	60	60	60.0	60.0	0.00	0.00	0.00	0.00
8	inst8perturbedlinearA	137	137	137	137.0	137.0	137	137	137.0	137.0	0.00	0.00	0.00	0.00
8	inst8perturbedlinearB	141	141	141	141.0	141.0	141	141	141.0	141.0	0.00	0.00	0.00	0.00
8	inst8perturbedlinearC	162	162	162	162.0	162.0	162	162	162.0	162.0	0.00	0.00	0.00	0.00
10	inst10perturbedlinearA	326	***315**	***315**	322.1	318.3	***315**	***315**	323.2	321.3	-3.82	1.48	-5.69	-0.38
10	inst10perturbedlinearB	274	274	274	275.5	275.8	274	274	277.5	277.6	-3.01	-1.12	-2.54	-1.13
10	inst10perturbedlinearC	284	***283**	***283**	288.5	287.1	***283**	***283**	288.1	286.3	-1.07	1.81	-0.40	1.87
12	inst12perturbedlinearA	587	647	598	667.9	607.3	***583**	584	603.2	604.9	63.19	66.34	-1.18	5.98
12	inst12perturbedlinearB	525	574	504	593.9	529.8	514	***502**	529.0	526.3	63.33	66.60	1.46	5.54
12	inst12perturbedlinearC	478	550	481	574.0	499.3	466	***460**	498.1	495.6	72.57	79.17	0.76	6.71
14	inst14perturbedlinearA	920	960	772	1025.1	838.9	***728**	751	785.9	798.7	229.79	248.61	31.64	48.70
14	inst14perturbedlinearB	932	924	829	986.2	869.1	***821**	824	854.9	857.8	125.74	136.86	8.21	14.39
14	inst14perturbedlinearC	990	1000	898	1037.3	918.6	***854**	871	896.0	902.5	138.65	143.95	11.12	21.08
16	inst16perturbedlinearA	1376	***1284**	1357	1415.6	1422.1	1292	1289	1377.1	1356.0	30.51	46.49	55.84	76.36
16	inst16perturbedlinearB	1348	1269	1287	1345.0	1351.7	1248	***1202**	1298.8	1307.9	37.39	55.01	36.87	50.73
16	inst16perturbedlinearC	1098	1052	1049	1111.3	1113.3	***962**	1004	1051.1	1064.0	49.43	70.90	45.23	53.37
18	inst18perturbedlinearA	2005	1976	2017	2051.7	2061.9	1825	***1801**	1949.3	1927.2	81.91	117.42	110.51	158.89
18	inst18perturbedlinearB	1921	1864	1884	1953.9	1949.6	1814	***1784**	1879.2	1878.3	68.76	80.57	62.70	79.90
18	inst18perturbedlinearC	1585	1655	1565	1739.4	1754.2	***1493**	***1493**	1636.3	1623.9	87.73	118.47	120.12	140.42
20	inst20perturbedlinearA	3065	3180	2898	3281.6	3004.3	2799	***2733**	2894.3	2891.3	380.08	394.52	103.09	122.97
20	inst20perturbedlinearB	2800	3006	2634	3099.0	2758.4	***2536**	2540	2646.7	2651.9	440.51	463.95	94.22	118.78
20	inst20perturbedlinearC	2791	2925	2556	3029.4	2652.0	***2372**	2400	2524.4	2524.8	489.95	520.05	115.52	138.88

Table 5.5: Numerical results for the Weighted Carry-Over Effects Value Minimization Problem. All values marked with * indicate new best results. BKS stands for best-known solution reported in [Guedes and Ribeiro, 2011].

Instance		PRS∪PTS		TARS		Relative difference	
Size	Name	Circle	Vizing	Circle	Vizing	Circle	Vizing
18	circ18abal	962.8	947.6	909.9	903.5	5.8%	4.9%
18	circ18bbal	972.8	952.1	922.1	916.2	5.5%	3.9%
18	circ18cbal	955.5	946.0	904.2	905.6	5.7%	4.5%
18	circ18dbal	951.5	946.9	916.1	903.8	3.9%	4.8%
18	circ18ebal	958.5	937.1	905.4	908.0	5.9%	3.2%
18	circ18fbal	957.9	948.0	917.3	912.3	4.4%	3.9%
18	circ18gbal	966.7	942.3	904.5	908.4	6.9%	3.7%
18	circ18hbal	950.5	942.1	906.6	904.8	4.8%	4.1%
18	circ18ibal	951.6	946.5	913.2	912.9	4.2%	3.7%
18	circ18jbal	958.7	944.2	901.8	910.5	6.3%	3.7%
18	circ18anonbal	965.1	964.2	928.7	927.2	3.9%	4.0%
18	circ18dnonbal	969.6	957.9	918.9	921.7	5.5%	3.9%
18	circ18enonbal	967.4	963.7	932.7	936.9	3.7%	2.9%
18	circ18fnonbal	983.4	970.6	936.8	939.6	5.0%	3.3%
18	circ18gnonbal	959.9	966.8	926.1	933.7	3.6%	3.5%
18	circ18hnonbal	973.1	970.7	941.7	936.5	3.3%	3.7%
18	circ18jnonbal	951.3	945.7	920.2	919.7	3.4%	2.8%
20	circ20abal.txt	1502.6	1290.9	1237.5	1232.9	21.4%	4.7%
20	circ20bbal.txt	1498.2	1293.0	1241.9	1244.9	20.6%	3.9%
20	circ20cbal.txt	1489.1	1283.7	1240.7	1233.3	20.0%	4.1%
20	circ20dbal.txt	1502.1	1287.7	1231.1	1231.5	22.0%	4.6%
20	circ20ebal.txt	1500.7	1290.3	1241.9	1234.2	20.8%	4.5%
20	circ20fbal.txt	1491.3	1279.1	1230.3	1239.7	21.2%	3.2%
20	circ20gbal.txt	1492.1	1288.5	1237.6	1239.4	20.6%	4.0%
20	circ20hbal.txt	1521.7	1295.7	1252.5	1250.6	21.5%	3.6%
20	circ20ibal.txt	1505.5	1288.9	1235.5	1242.1	21.9%	3.8%
20	circ20jbal.txt	1517.4	1280.0	1235.5	1239.1	22.8%	3.3%
20	circ20cnonbal.txt	1510.5	1305.2	1270.2	1271.7	18.9%	2.6%
20	circ20dnonbal.txt	1519.9	1321.4	1279.7	1268.2	18.8%	4.2%
20	circ20enonbal.txt	1532.3	1350.8	1292.8	1302.2	18.5%	3.7%
20	circ20inonbal.txt	1514.7	1310.3	1255.9	1267.4	20.6%	3.4%
20	circ20anonbal.txt	1495.1	1314.7	1270.1	1272.1	17.7%	3.3%
20	circ20bnonbal.txt	1510.0	1315.9	1268.1	1270.5	19.1%	3.6%
20	circ20gnonbal.txt	1512.9	1321.8	1292.1	1283.3	17.1%	3.0%
20	circ20jnonbal.txt	1496.5	1299.4	1247.1	1249.3	20.0%	4.0%

Table 5.6: Performance evaluation of the existing neighborhoods for the Traveling Tournament Problem with Predefined Venues. The local optimum values were obtained after applying each local search procedure to solutions obtained with constructive methods presented in this thesis.

			PRS∪PTS				TARS				Confidence interval			
Instance			Best result		Average		Best result		Average		Circle		Vizing	
Size	Name	BKS	Circle	Vizing	Circle	Vizing	Circle	Vizing	Circle	Vizing	LB	UB	LB	UB
18	circ18abal	850	802	842	826.5	823.6	826	***798**	814.7	812.5	1.92	21.68	0.96	21.31
18	circ18bbal	844	842	***804**	828.9	827.4	816	834	813.6	820.2	3.97	26.56	-4.81	19.21
18	circ18cbal	856	***816**	824	823.8	823.1	822	820	815.0	817.1	-0.65	18.25	-2.77	14.77
18	circ18dbal	842	820	***806**	822.3	819.3	822	810	809.2	812.7	3.14	23.12	-2.23	15.30
18	circ18ebal	838	838	836	822.6	822.2	814	***802**	811.5	812.9	-0.24	22.51	-0.59	19.12
18	circ18fbal	834	828	828	824.1	828.5	***802**	834	821.2	821.0	-6.51	12.38	-0.60	15.67
18	circ18gbal	842	812	824	824.0	824.3	***810**	816	810.7	810.7	3.94	22.59	5.23	21.97
18	circ18hbal	838	834	820	818.6	818.7	***796**	808	807.0	810.1	0.68	22.52	1.31	16.02
18	circ18ibal	848	818	830	821.6	823.5	826	***790**	816.4	811.9	-5.03	15.43	0.23	22.84
18	circ18jbal	830	816	834	817.4	822.6	818	***794**	806.8	806.7	-0.31	21.51	5.47	26.40
18	circ18anonbal	862	850	832	834.9	834.6	832	***822**	824.7	825.3	3.05	17.21	-1.12	19.66
18	circ18dnonbal	854	***810**	822	827.4	825.1	824	***810**	819.9	817.2	-5.02	20.09	1.00	14.73
18	circ18enonbal	854	844	834	833.3	835.6	832	***802**	821.6	824.9	0.59	22.74	-1.03	22.50
18	circ18fnonbal	864	826	***824**	836.6	837.4	844	848	831.9	830.5	-5.80	15.13	-4.74	18.48
18	circ18gnonbal	856	838	836	828.2	832.1	824	***818**	817.2	818.8	0.92	21.08	4.43	22.11
18	circ18hnonbal	860	828	830	836.1	836.1	***818**	832	829.0	821.3	-0.68	14.95	4.25	25.35
18	circ18jnonbal	858	834	834	826.0	826.1	***798**	818	817.8	813.0	-3.30	19.70	4.33	21.80

Table 5.7: Numerical results for the Traveling Tournament Problem with Predefined Venues. All values marked with * indicate new best results. BKS stands for best-known solution reported in [Costa et al., 2012].

			PRS∪PTS				TARS				Confidence interval			
Instance			Best result		Average		Best result		Average		Circle		Vizing	
Size	Name	BKS	Circle	Vizing	Circle	Vizing	Circle	Vizing	Circle	Vizing	LB	UB	LB	UB
20	circ20abal.txt	1170	1320	1098	1342.3	1129.9	⋆**1086**	1090	1116.9	1113.7	219.98	230.82	14.08	18.32
20	circ20bbal.txt	1200	1284	1108	1318.6	1132.1	⋆**1084**	1086	1113.3	1115.9	201.40	209.26	14.42	18.11
20	circ20cbal.txt	1192	1310	1100	1313.5	1123.4	⋆**1078**	1080	1104.4	1108.3	202.06	216.21	13.03	17.10
20	circ20dbal.txt	1172	1304	1096	1331.3	1128.2	⋆**1084**	1092	1116.0	1117.3	212.17	218.50	8.77	12.96
20	circ20ebal.txt	1202	1312	1116	1331.3	1136.2	⋆**1068**	1082	1115.9	1118.5	208.68	222.12	14.51	20.83
20	circ20fbal.txt	1202	1282	1096	1319.3	1120.5	1084	⋆**1082**	1107.1	1105.8	209.07	215.46	11.79	17.68
20	circ20gbal.txt	1188	1294	1106	1321.4	1123.9	⋆**1072**	1082	1106.3	1105.3	212.14	218.13	16.58	20.62
20	circ20hbal.txt	1190	1298	1098	1323.6	1139.4	1098	⋆**1076**	1123.7	1124.3	196.67	203.20	13.59	16.54
20	circ20ibal.txt	1172	1310	⋆**1074**	1330.2	1127.9	⋆**1074**	1084	1115.6	1114.7	210.08	219.12	7.94	18.46
20	circ20jbal.txt	1182	1286	⋆**1080**	1327.5	1126.0	1082	⋆**1080**	1110.4	1112.1	213.89	220.38	11.65	16.08
20	circ20cnonbal.txt	1208	1380	1102	1410.7	1140.2	⋆**1084**	⋆**1084**	1129.1	1126.3	277.90	285.30	11.06	16.67
20	circ20dnonbal.txt	1222	1378	1136	1420.4	1160.6	⋆**1114**	1116	1142.1	1141.5	274.79	281.88	14.82	23.32
20	circ20enonbal.txt	1190	1410	1134	1440.9	1169.9	⋆**1108**	1126	1151.0	1148.9	285.12	294.61	18.91	23.22
20	circ20inonbal.txt	1234	1338	1112	1361.0	1140.5	1104	⋆**1100**	1128.1	1127.0	230.37	235.36	10.80	16.27
20	circ20anonbal.txt	1232	1416	1114	1437.5	1144.7	⋆**1088**	1110	1129.1	1131.2	300.67	316.00	11.14	15.93
20	circ20bnonbal.txt	1244	1334	1106	1368.3	1143.8	1090	⋆ 1080	1128.6	1128.9	236.32	243.15	11.88	17.99
20	circ20gnonbal.txt	1208	1376	1130	1425.9	1158.7	⋆**1086**	1112	1132.5	1138.9	291.32	295.48	17.52	22.08
20	circ20jnonbal.txt	1194	1312	1112	1355.5	1130.7	1096	⋆**1086**	1114.1	1110.9	236.44	246.36	17.68	21.92

Table 5.8: Numerical results for the Traveling Tournament Problem with Predefined Venues. All values marked with ⋆ indicate new best results. BKS stands for best-known solution reported in [Costa et al., 2012].

prime number plus one reported by [Costa et al., 2012] and [Januario and Urrutia, 2015] in which the connectivity issue of the search space arises. As discussed in the previous chapter, when the initial solutions are constructed using the circle method, any local search procedure based on the existing neighborhood structures cannot move from the schedules that are isomorphic to the canonical one.

We note that regardless the initial solution, in general, the quality of the solutions obtained with the use of TARS is statistically superior than the ones obtained with the previously known neighborhood structures. For the TTPPV, the heuristic using TARS in its local search procedure was able to improve the best-known solution for 31 of the 35 evaluated instances. For the WCOEVMP, the same algorithm was able to improve the best-known solution for 34 of the 69 evaluated instances.

Chapter 6

Conclusion

This chapter gives a brief discussion about the findings reported in this thesis. Future works can be produced as outcomes of this thesis. Some are presented afterward.

6.1 Concluding remarks

This work built on a proper modeling of *Single Round-Robin* (SRR) scheduling problems in terms of edge coloring of complete graphs. Describing the neighborhood structures for sport scheduling problems without using the formalism of graph theory would undoubtedly be a challenging - but tedious - task which we can, fortunately, avoid.

A new constructive method for scheduling SRR tournaments, the faro method, was introduced and its equivalence to the classic circle method was established. The proposed method was used to study the relation between schedules constructed by the circle method and the faro shuffle of playing cards. There are many ways to construct initial solutions for scheduling problems but, according to [Goossens and Spieksma, 2012b], the most popular method is definitely the circle method.

In this thesis, we analyzed the connectivity of the most commonly used neighborhoods in the literature. When $n = p + 1$ with p being a prime number PRS is not connected and if the initial schedule is constructed with the circle method it is not possible to move to other non-isomorphic schedules with that neighborhood. Analyzing PTS, due to the relation established between the faro method and the circle method it is clear to see that such neighborhood is also not connected. It was proven that, when n is such that a faro shuffle permutation on a deck of n cards has an $(n-2)$-cycle, it is again not possible to move to other non-isomorphic schedules with it. A new theorem were introduced in this thesis, giving theoretical evidences that local search algorithms that make use of the existing neighborhoods may lead to poor results when initial schedules are constructed with the circle method, for certain values of n.

An important contribution is this work is a new neighborhood for SRR scheduling problems, called *Teams and Rounds Swap* (TARS). Using the Weighted Carry-Over Effects Values Minimization Problem and the Traveling Tournament Problem with Predefined Venues as case studies, we showed that this neighborhood structure avoids the issue of non-connectivity of the existing neighborhood structures. Moreover, and as a consequence, the ILS heuristic using the TARS neighborhood structure led to several best results for both case problems, regardless the initial solution or the tackled instance.

The dominance of TARS over the existing neighborhood structures is confirmed by a test t for paired observation at 99.9% confidence level. For the WCOEV, we note that the improvement over the existing neighborhood structures is larger when comparing results for instances with of size 12, 14 and 20 while the circle method is used to generate the initial coloring. This expected result is an outcome of the fact that TARS is able to move from perfect colorings to non-perfect colorings even in cases where the existing neighborhood structures cannot.

6.1.1 Future works

As shown in this work, TARS increases the connectivity of the search space of single round-robin problems. Whether the search space is fully connected by TARS is still an open issue.

Approaches based on graph theory concepts applied to round-robin sport scheduling problems seem promising. [de Werra, 1981] was one of the first authors who discussed the application of graph theory concepts to sport scheduling problems and, after more than 30 years, it is still an inspiring subject. [Lewis and Thompson, 2011] presented models based on vertex coloring of complete graphs in order to solve the TTP.

We aim to include the usage of TARS in other round-robin scheduling problems and even propose new neighborhood structures. In particular, we are interested in evaluating TARS' performance in a heuristic approach to *Mirrored Traveling Tournament Problem* (MTTP) and TTP.

The MTTP is a variant of the TTP with an SRR tournament in the first $n - 1$ rounds (first leg), followed by the same tournament with reversed venues in the last $n - 1$ rounds (second leg). After years of investigation, the optimal solution for even small instances of the TTP remains unknown. The best results for instances of the TTP are being tracked on Michael Trick's webpage [Trick, 2015].

Besides, many other discrete problems in several areas can be transformed into equivalent graph problems and then solved using graph algorithms [Ebert, 1987]. TARS neighborhood may be used in local search algorithms for solving problems that can be modeled as edge coloring of graphs.

Bibliography

(2006). First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5):802–817.

Aigner, M. and Ziegler, G. M. (2004). *Proofs from THE BOOK*. Springer.

Anagnostopoulos, A., Michel, L., Hentenryck, P. V., and Vergados, Y. (2006). A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2):177–193.

Burstyn, V. (1999). *The Rites of Men: Manhood, Politics, and the Culture of Sport*. University of Toronto Press, first edition.

Comtet, L. (1974). *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. D. Reidel Publishing Company.

Costa, F. N., Urrutia, S., and Ribeiro, C. C. (2012). An ILS heuristic for the traveling tournament problem with predefined venues. *Annals of Operations Research*, 194(1):137–150.

de Werra, D. (1981). Scheduling in sports. In Hansen, P., editor, *Studies on Graphs and Discrete Programming*, volume 11 of *Annals of Discrete Mathematics*, pages 381–395. North-Holland.

de Werra, D. and Blazewicz, J. (1992). Some preemptive open shop scheduling problems with a renewable or a nonrenewable resource. *Discrete Applied Mathematics*, 35(3):205–219.

Di Gaspero, L. and Schaerf, A. (2007). A composite-neighborhood tabu search approach to the traveling tournament problem. *Journal of Heuristics*, 13(2):189–207.

Diaconis, P., Graham, R., and Kantor, W. M. (1983). The mathematics of perfect shuffles. *Advances in Applied Mathematics*, 4(2):175–196.

Diestel, R. (2005). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer.

Dijkstra, E. W. and Rao, J. (1990). Constructing the proof of Vizing's Theorem. Technical report EWD1075, University of Texas at Austin.

Dinitz, J. H., Garnick, D. K., and McKay, B. D. (1994). There are 526,915,620 nonisomorphic one-factorizations of k_{12}. *Journal of Combinatorial Designs*, 2(4):273–285.

Easton, K., Nemhauser, G., and Trick, M. (2001). The traveling tournament problem description and benchmarks. In Walsh, T., editor, *Principles and Practice of Constraint Programming*, volume 2239 of *Lecture Notes in Computer Science*, pages 580–584. Springer.

Ebert, J. (1987). A versatile data structure for edge-oriented graph algorithms. *Communications of the ACM*, 30(6):513–519.

Ellis, J., Fan, H., and Shallit, J. (2002). The cycles of the multiway perfect shuffle permutation. *Discrete Mathematics and Theoretical Computer Science*, 5(1):169–180.

Gabow, H. N., Nishizeki, T., Kariv, O. Leven, D., and Terada, O. (1985). Algorithms for edge-coloring graphs. Technical report TRECIS-8501, Tohoku University.

Gandham, S., Dawande, M., and Prakash, R. (2008). Link scheduling in wireless sensor networks: Distributed edge-coloring revisited. *Journal of Parallel and Distributed Computing*, 68(8):1122–1134.

Goossens, D. R. and Spieksma, F. C. (2012a). The carryover effect does not influence football results. *Journal of Sports Economics*, 13(3):261–278.

Goossens, D. R. and Spieksma, F. C. R. (2012b). Soccer schedules in europe: an overview. *Journal of Scheduling*, 15(5):641–651.

Gould, R. (1988). *Graph Theory*. Benjamin-Cummings.

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

Holyer, I. (1981). The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720.

Hoos, H. and Stützle, T. (2004). *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann Publishers Inc.

Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, first edition.

Januario, T. and Urrutia, S. (2015). An analytical study in connectivity of neighborhoods for single round robin tournaments. In Brian Borchers, J. P. B. and McLay, L., editors, *Operations Research and Computing: Algorithms and Software for Analytics*, pages 188–199. INFORMS.

Jhonson, D. S. and McGeoch, L. A. (1997). *Local Search in Combinatorial Optimization*, chapter The Traveling Salesman Problem: A Case Study in Local Optimization. John Wiley and Sons, London.

Kaski, P. and Ostergard, P. R. J. (2009). There are 1,132,835,421,602,062,347 nonisomorphic one-factorizations of K_{14}. *Journal of Combinatorial Designs*, 17(2):147–159.

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

Kobayashi, M. (1989). On perfect one-factorization of the complete graph K_{2p}. *Graphs and Combinatorics*, 5(1):351–353.

Lewis, R. and Thompson, J. (2011). On the application of graph colouring techniques in round-robin sports scheduling. *Computers & Operations Research*, 38(1):190–204.

Mann, B. (1995). How many times should you shuffle a deck of cards? *Topics in Contemporary Probability and Its Applications*, 15:1–33.

Melo, R. A., Urrutia, S., and Ribeiro, C. C. (2009). The traveling tournament problem with predefined venues. *Journal of Scheduling*, 12(6):607–622.

Misra, J. and Gries, D. (1992). A constructive proof of vizing's theorem. *Information Processing Letters*, 41(3):131–133.

Miyashiro, R. and Matsui, T. (2006). Minimizing the carry-over effects value in a round- robin tournament. In Burke, E. and Rudova, H., editors, *Proceedings of the 6th international conference on the practice and theory of automated timetabling*, pages 216–230. PATAT.

Morris, S. B. and Hartwig, R. E. (1976). The generalized faro shuffle. *Discrete Mathematics*, 15(4):333–346.

OEIS (2015). List of numbers 2n for which the riffle permutation permutes all except the first and last of the 2n cards. `http://oeis.org/A217948`. last visited on April 23, 2015.

Papadimitriou, C. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover Books on Computer Science Series. Dover Publications, first edition.

Rasmussen, R. V. and Trick, M. A. (2008). Round robin scheduling - A survey. *European Journal of Operational Research*, 188(3):617–636.

Ribeiro, C. C. and Urrutia, S. (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3):775–787.

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

Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Pearson Education.

Tainsky, S., Salaga, S., and Santos, C. (2012). Estimating attendance for the ultimate fighting championship: A demand theory approach. *International Journal of Sport Management and Marketing*, 11(3):206–224.

Tainsky, S., Salaga, S., and Santos, C. (2013). Determinants of pay-per-view broadcast viewership in sports: The case of the ultimate fighting championship. *Journal of Sport Management*, 27(1):43–58.

Trick, M. (2015). Challenge traveling tournament problems benchmark. `http://mat.gsia.cmu.edu/TOURN/`. Accessed: 2015-03-28.

Vizing, V. G. (1964). On an estimate of the chromatic class of a p-graph. *Discrete Analysis*, 3:25–30. in Russian.

Watanabe, N. (2012). Demand for pay-per-view consumption of ultimate fighting championship events. *International Journal of Sport Management and Marketing*, 11(3):225–238.