

JOÃO PEDRO PEREIRA ARMANI

**GERAÇÃO OTIMIZADA DA TABELA DE JOGOS
DO CAMPEONATO BRASILEIRO DE FUTEBOL
COM CONSIDERAÇÕES DE JUSTIÇA
COMPETITIVA ENTRE AS EQUIPES**

São Paulo
2023

JOÃO PEDRO PEREIRA ARMANI

**GERAÇÃO OTIMIZADA DA TABELA DE JOGOS
DO CÂMPEONATO BRASILEIRO DE FUTEBOL
COM CONSIDERAÇÕES DE JUSTIÇA
COMPETITIVA ENTRE AS EQUIPES**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Diploma de Engenheiro de
Produção.

São Paulo
2023

JOÃO PEDRO PEREIRA ARMANI

**GERAÇÃO OTIMIZADA DA TABELA DE JOGOS
DO CAMPEONATO BRASILEIRO DE FUTEBOL
COM CONSIDERAÇÕES DE JUSTIÇA
COMPETITIVA ENTRE AS EQUIPES**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Diploma de Engenheiro de
Produção.

Orientador:

Leonardo Junqueira

São Paulo
2023

À minha família e meus amigos.

AGRADECIMENTOS

Quero expressar minha gratidão à minha família pelo apoio constante e pela confiança que depositaram em mim.

Agradeço imensamente aos meus pais por serem uma fonte de motivação e por fornecerem as condições necessárias para o meu sucesso.

Aos meus amigos, por todos os momentos compartilhados e pelo suporte incondicional ao longo dos anos.

Agradeço a todos os professores da Escola Politécnica que, ao longo da minha graduação, dedicaram seu tempo para meu ensino e contribuíram significativamente para minha formação pessoal, profissional e acadêmica.

Um agradecimento especial ao meu orientador, Prof. Dr. Leonardo Junqueira, pelo apoio e suporte prestados durante todo o desenvolvimento deste Trabalho de Formatura. Sua contribuição e atenção foram fundamentais.

“

You dream. You plan. You reach. There will be obstacles. There will be doubters. There will be mistakes. But with hard work, with belief, with confidence and trust in yourself and those around you, there are no limits.

”

– Michael Phelps

RESUMO

As ligas esportivas profissionais, dados os altos investimentos, sobretudo em jogadores e estrutura, têm se tornado um grande negócio, passando cada vez mais a serem encaradas como uma forte atividade econômica por todo o mundo. As ligas e os seus respectivos times não desejam ter seus gastos potencializados em virtude de uma tabela de jogos mal elaborada e/ou injusta. A geração de uma tabela é um grande desafio, visto que esse processo envolve uma série de interesses/restrições e múltiplos objetivos a serem atingidos e/ou otimizados. Além desse caráter desafiador, a geração de uma tabela de jogos pode influenciar diretamente aspectos como atratividade e competitividade dos campeonatos, tornando-os mais atrativos (desatrativos) e equilibrados (desequilibrados) o que, por sua vez, impacta positivamente (negativamente) na saúde financeira dos campeonatos e de seus times. A Série A do Campeonato Brasileiro de Futebol é um campeonato anual envolvendo 20 times espalhados por todo o território nacional, e conhecido internacionalmente por ser um dos mais equilibrados e competitivos do mundo. O foco deste Trabalho de Formatura é a aplicação de técnicas de Pesquisa Operacional para a geração da tabela de jogos da Série A do Campeonato Brasileiro de Futebol, visando apresentar uma tabela mais justa entre os times. Após o estudo dos problemas de *scheduling* aplicados a esportes e dos problemas de geração de tabelas de campeonatos, é proposto um modelo de otimização que aborda as principais restrições esportivas básicas nesse contexto e que tem como objetivo evitar muitas sequências de jogos consecutivos como mandante ou visitante, o que geraria impactos tanto na esfera esportiva como nas esferas financeira e de engajamento dos torcedores, através da minimização do número de *breaks* e melhorar a qualidade do torneio, evitando a ocorrência de muitas sequências de enfrentamento iguais para muitos times, através da redução de efeitos de *carry over*. Os objetivos são avaliados de forma simultânea e integrada. Também é proposto um algoritmo contendo uma heurística construtiva, responsável por gerar uma solução factível inicial de maneira rápida e um conjunto de vizinhanças a serem exploradas por meio de um processo iterativo, visando melhorar a solução gerada pela heurística construtiva. Ambos os métodos foram implementados na linguagem *Python* e o modelo foi resolvido através do resolvidor *Gurobi*.

Palavras-chave: Geração de tabelas de futebol, *Breaks*, Efeitos de *carry over*, Modelagem Matemática, Otimização, Algoritmo, Heurística Construtiva, Vizinhanças.

ABSTRACT

Professional sports leagues, given their high investments, especially in players and infrastructure, have become a major business, increasingly seen as a strong economic activity worldwide. Leagues and their respective teams do not wish to have their expenses increased due to a poorly designed and/or unfair game schedule. Creating a game timetable is a significant challenge, as this process involves a series of interests/constraints and multiple objectives to be achieved and/or optimized. Besides this challenging aspect, generating a game timetable can directly influence aspects such as the attractiveness and competitiveness of the championships, making them more appealing (or less) and balanced (or imbalanced), which, in turn, positively (or negatively) impacts the financial health of the championships and their teams. The Brazilian Serie A Football Championship is an annual tournament involving 20 teams spread across the national territory, internationally recognized as one of the most balanced and competitive in the world. The focus of this Graduation Project is the application of Operational Research techniques for generating the game timetable for the Brazilian Serie A Football Championship, aiming to present a fairer schedule among the teams. After studying scheduling problems applied to sports and championship timetable generation problems, an optimization model is proposed that addresses the main basic sports constraints in this context. The objective is to avoid many sequences of consecutive games as host or visitor, which would impact not only the sports sphere but also the financial and fan engagement spheres, by minimizing the number of breaks and improving the tournament's quality, avoiding the occurrence of many identical match sequences for several teams by reducing carry over effects. The objectives are evaluated simultaneously and integrally. An algorithm comprising a constructive heuristic responsible for quickly generating an initial feasible solution, and a set of neighborhoods to be explored, through an iterative process, aiming to improve the solution generated by the constructive heuristic is also proposed. Both methods were implemented in the Python language, and the model was solved using the Gurobi solver.

Keywords: Football timetabling, Breaks, Carry over effects, Mathematical Modeling, Optimization, Algorithm, Heuristic, Neighborhoods.

LISTA DE FIGURAS

1	Estrutura do Capítulo de Fundamentação Teórica.	21
2	Publicações sobre aplicações de PO para Agendamento Esportivo nas últimas 6 décadas.	25
3	Polígono formado para $n = 8$ times (primeira rodada).	55
4	Matriz de adversários consecutivos para $n = 20$	57
5	Aplicação da alternância parcial de rodadas envolvendo os times 2, 3, 7 e 8.	61
6	Evolução de <i>breaks</i> e valores de efeitos de <i>carry over</i> para a melhor execução de instância.	72
7	Evolução de Melhoria da Vizinhança MHAS	76
8	Evolução de Melhoria da Vizinhança RS	77
9	Evolução de Melhoria da Vizinhança PRS	77
10	Fronteira de Pareto para a instância do Campeonato Brasileiro de Futebol - Série A	79

LISTA DE TABELAS

1	Diferentes tipos de simetria para os jogos do time A.	28
2	Exemplo de torneio <i>Single Round-Robin</i> , compacto com 10 times.	29
3	Ocorrência de <i>breaks</i> para o exemplo elaborado.	30
4	Adversários em cada rodada.	31
5	Valores de <i>carry over</i> para o exemplo elaborado.	32
6	Problemas de Geração de Tabelas de Campeonatos.	34
7	Tabela do Campeonato Brasileiro de Futebol - Série A 2023.	39
8	Ocorrência de <i>breaks</i> para o Campeonato Brasileiro de Futebol - Série A 2023.	40
9	Valores de <i>carry over</i> para o Campeonato Brasileiro de Futebol - Série A 2023.	41
10	Índices e conjuntos da modelagem.	47
11	Parâmetros da modelagem.	48
12	Variáveis de decisão da modelagem.	48
13	Tabela fictícia com 8 times.	58
14	Solução vizinha com o mando invertido para o jogo entre os times A e E. .	59
15	Solução vizinha invertendo todos os mandos dos jogos envolvendo o time E.	59
16	Solução vizinha com as rodadas 2 e 3 invertidas.	60
17	Inserção do jogo G x B na sétima rodada.	61
18	Siglas das vizinhanças.	64
19	Resultados obtidos pelo modelo e pelo algoritmo para diferentes instâncias.	69
20	Número de restrições e variáveis por instância.	70
21	Resultados gerados nas 3 execuções de cada instância.	71
22	Desempenho das vizinhanças consideradas nas 9 instâncias propostas. . . .	73

23	Performance das vizinhanças consideradas nas 9 instâncias propostas. . . .	75
24	Valores encontrados de <i>breaks</i> e efeitos de <i>carry over</i> para diferentes penalidades.	79
25	Tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.	94
26	Ocorrência de <i>breaks</i> para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.	95
27	Valores de <i>carry-over</i> para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.	96
28	Tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.	97
29	Ocorrência de <i>breaks</i> para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.	98
30	Valores de <i>carry-over</i> para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.	99
31	Análise de vizinhanças para $n = 6$ - Execução 1.	100
32	Análise de vizinhanças para $n = 6$ - Execução 2.	101
33	Análise de vizinhanças para $n = 6$ - Execução 3.	101
34	Análise de vizinhanças para $n = 8$ - Execução 1.	101
35	Análise de vizinhanças para $n = 8$ - Execução 2.	101
36	Análise de vizinhanças para $n = 8$ - Execução 3.	102
37	Análise de vizinhanças para $n = 10$ - Execução 1.	102
38	Análise de vizinhanças para $n = 10$ - Execução 2.	102
39	Análise de vizinhanças para $n = 10$ - Execução 3.	103
40	Análise de vizinhanças para $n = 12$ - Execução 1.	103
41	Análise de vizinhanças para $n = 12$ - Execução 2.	103
42	Análise de vizinhanças para $n = 12$ - Execução 3.	104
43	Análise de vizinhanças para $n = 14$ - Execução 1.	104

44	Análise de vizinhanças para $n = 14$ - Execução 2.	105
45	Análise de vizinhanças para $n = 14$ - Execução 3.	105
46	Análise de vizinhanças para $n = 16$ - Execução 1.	106
47	Análise de vizinhanças para $n = 16$ - Execução 2.	106
48	Análise de vizinhanças para $n = 16$ - Execução 3.	107
49	Análise de vizinhanças para $n = 18$ - Execução 1.	108
50	Análise de vizinhanças para $n = 18$ - Execução 2.	109
51	Análise de vizinhanças para $n = 18$ - Execução 3.	110
52	Análise de vizinhanças para $n = 20$ - Execução 1.	111
53	Análise de vizinhanças para $n = 20$ - Execução 2.	111
54	Análise de vizinhanças para $n = 20$ - Execução 3.	112
55	Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 1.	113
56	Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 2.	114
57	Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 3.	115

SUMÁRIO

1	Introdução	17
1.1	Objetivos do Trabalho	18
1.2	Estrutura do Trabalho	19
2	Fundamentação Teórica	21
2.1	Aplicações de Pesquisa Operacional em Esportes	21
2.2	Problemas de <i>Scheduling</i> Aplicados em Esportes	24
2.3	Problemas de Geração de Tabelas de Campeonatos	26
2.3.1	Terminologia	26
2.3.2	Definição de Variáveis	28
2.3.3	Funções Objetivo	28
2.3.4	Restrições Contempladas	33
2.4	Abordagens de Solução para Problemas de Geração de Tabelas de Campeonatos	35
3	Descrição do Problema	37
3.1	Campeonato Estudado	37
3.2	Problema Abordado	38
3.3	Análise da Tabela	42
3.4	Considerações Sobre o Problema	44
4	Modelagem Matemática	46
4.1	Definição de Elementos, Conjuntos e Variáveis do Modelo Matemático . . .	46
4.2	Função Objetivo e Restrições do Modelo Matemático	48
5	Abordagem de Solução	54

5.1	Rápida Heurística Construtiva	54
5.1.1	Criação da Tabela com Times Fictícios	54
5.1.2	Definição dos Mandos de Campo	55
5.1.3	Alocação dos Times Reais	56
5.2	Vizinhanças	58
5.2.1	Alternância de Mando de uma Partida	58
5.2.2	Alternância dos Mandos de Campo para um Time	59
5.2.3	Alternância de Rodadas	60
5.2.4	Alternância Parcial de Rodadas	60
5.2.5	Rotação de Jogos	61
5.3	Procedimento Iterativo	63
5.4	Abordagem Completa e Pseudo-código	63
6	Experimentos Computacionais e Resultados	67
6.1	Eficiência do Algoritmo	67
6.2	Análise de Vizinhanças	72
6.3	Trabalhando com Diferentes Penalidades	78
6.4	Consolidação das Análises Realizadas e Principais <i>Insights</i>	81
7	Conclusões e Perspectivas Futuras	84
	Referências	86
	Apêndice A – Tabelas Geradas Através do Modelo e do Algoritmo	92
	Apêndice B – Execuções do Algoritmo	100

1 INTRODUÇÃO

Esportes, de um modo geral, têm se tornado cada vez mais um grande negócio. Em uma economia globalizada, muitos países e cidades lutam pelos direitos de organizar grandes eventos como os Jogos Olímpicos e a Copa do Mundo de Futebol, que muitas vezes trazem milhares de empregos, revitalização urbana e oportunidades econômicas para seus anfitriões (KENDALL et al., 2010). Fãs conferem jornais, rádio, televisão e internet em busca de informações sobre suas equipes locais e nacionais, jogadores preferidos e pontuações (RIBEIRO; URRUTIA, 2012).

Esportes são estruturados, orientados a um objetivo, baseados em enfrentamento e competitivos. Logo, eles promovem um cenário natural para a tomada de decisão antes ou durante o enfrentamento (GERCHAK, 1994). Nesse contexto, o uso de técnicas de Pesquisa Operacional (PO) é bastante recorrente, visando solucionar os mais diversos problemas, incluindo decisões a serem tomadas por jogadores para maximizar suas chances de vitória, ranqueamento de times e jogadores, análise das regras atuais de alguns esportes e impactos de mudanças das mesmas, entre outros. O tema que será abordado neste Trabalho de Formatura é o de *sports timetabling*.

Desde que existe esporte competitivo, há a necessidade de tabelas esportivas. Nos últimos 30 anos, o agendamento esportivo se tornou uma área de pesquisa própria dentro das comunidades de Pesquisa Operacional e Ciência da Computação. Embora possa parecer trivial agendar um torneio, e a matemática combinatória possui métodos para agendar torneios simples, quando requisitos adicionais são considerados, o problema se torna um problema de otimização combinatória muito difícil (RASMUSSEN; TRICK, 2008). A dificuldade se dá, segundo Rasmussen (2008), por conta de uma ampla gama de interesses muitas vezes conflitantes vindos de equipes, redes de televisão, espectadores e a associação organizadora do torneio.

No caso da Série A do Campeonato Brasileiro de Futebol, por exemplo, a geração da tabela de jogos do final da primeira década deste século foi modelada em conjunto pela Confederação Brasileira de Futebol (CBF), entidade máxima do futebol no Brasil,

responsável pelos campeonatos nacionais, e pela TV Globo, maior grupo de mídia e televisão do país. Os objetivos, restrições e demais considerações feitas à época podem ser encontrados no trabalho de Ribeiro e Urrutia (2012).

Este Trabalho de Formatura tem como objeto de estudo o Problema de Geração da Tabela de Jogos do Campeonato Brasileiro de Futebol - Série A. O estudo envolverá a realização da revisão da literatura sobre a aplicação das técnicas de Pesquisa Operacional para os problemas dessa classe e dois principais critérios serão considerados para garantir a justiça e a qualidade da competição. O primeiro deles diz respeito a ocorrência de muitas sequências de jogos consecutivos como mandante ou visitante. Para cada sequência de dois jogos consecutivos em uma dessas condições dá-se o nome de *break*. Em Buraimo, Forrest e Simmons (2009), por exemplo, é exposta a relação entre jogar partidas consecutivas em uma mesma condição (em casa ou fora) e a presença de público nos estádios, o que impacta diretamente nas finanças dos clubes mandantes. O segundo critério considerado diz respeito a ocorrência de muitas sequências de enfrentamento iguais para muitos times, gerando muitos padrões ao longo do campeonato. Em Briskorn e Knust (2010), é mostrado como jogar contra times considerados fortes ou fracos tem um impacto direto no desempenho dos times, podendo causar desgaste, lesões ou suspensões. Um cenário em que muitos times enfrentem de maneira consecutiva os mesmos dois times poderia impactar diretamente nos resultados das partidas. Basta imaginar que um time poderia ser beneficiado ao enfrentar uma equipe que na rodada anterior teve como adversário um time muito forte. O contrário também é válido, um time poderia ser prejudicado ao enfrentar uma equipe que acabou de ter um jogo considerado fácil. Construir uma tabela de jogos em que determinado time jogue muitas vezes contra uma equipe qualquer que, na rodada anterior, jogou contra um mesmo terceiro time pode influenciar diretamente no campeonato. À presença de muitos padrões pode ser observada a partir de um elevado número de efeitos de *carry over*. Posteriormente, a metodologia de PO será aplicada, uma tabela contendo os 380 jogos, distribuídos em 38 rodadas, será gerada através de dois métodos de solução e os resultados serão analisados frente a frente com a tabela construída pela CBF para o ano de 2023.

1.1 Objetivos do Trabalho

O objetivo fundamental deste Trabalho de Formatura consiste em aplicar a teoria e as técnicas de Pesquisa Operacional na geração da tabela de jogos da Série A do Campeonato Brasileiro de Futebol de maneira automatizada, contemplando uma série de restrições a

serem atendidas pelo modelo desenvolvido, visando gerar uma tabela final que possa ser considerada mais justa para as equipes como um todo, conforme a literatura sobre o tema. Dessa forma, o modelo matemático que representará o problema será implementado através da linguagem *Python* e resolvido através do resolvidor *Gurobi* e o algoritmo envolvendo a heurística, e a exploração das vizinhanças que será utilizado para a obtenção de soluções mais rápidas e/ou melhores serão implementados, também, com a linguagem de programação *Python*. Ambos os métodos de geração da tabela levarão em conta as restrições impostas, visando atender os objetivos de justiça e igualdade propostos.

1.2 Estrutura do Trabalho

Este Trabalho de Formatura está estruturado em 7 capítulos, sendo eles:

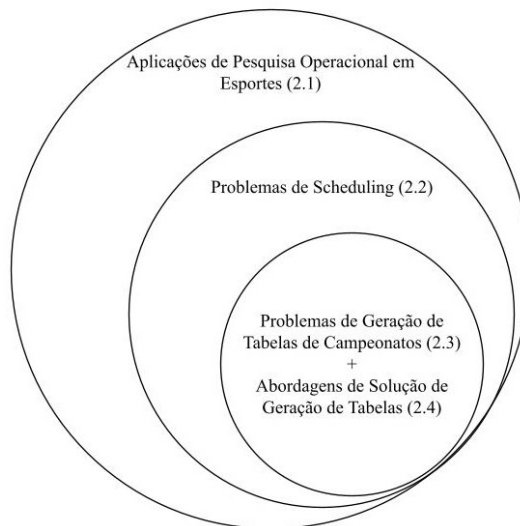
- **Introdução:** Apresentação inicial abrangendo a temática do presente Trabalho de Formatura, estabelecendo os objetivos específicos e as motivações fundamentais que impulsionaram a realização deste estudo.
- **Fundamentação Teórica:** Apresentação do estudo teórico aprofundado, baseado em uma revisão abrangente da literatura existente, abordando a temática do trabalho desde a aplicação de Pesquisa Operacional em Esportes até uma análise detalhada dos Problemas de Geração de Tabelas de Jogos de Campeonatos e suas soluções atuais. São discutidos conceitos, métodos e abordagens relevantes que sustentam o embasamento teórico deste estudo.
- **Descrição do Problema:** São fornecidos detalhes abrangentes sobre o Campeonato Brasileiro de Futebol - Série A, campeonato que será abordado neste trabalho. São destacadas as peculiaridades e características relevantes que devem ser levadas em consideração na modelagem matemática, a fim de atender às especificações e requisitos do campeonato em questão e, assim, propor uma solução viável para o problema em questão.
- **Modelagem Matemática:** Descrição minuciosa do modelo matemático elaborado para representar e descrever de forma precisa o problema abordado. São definidos os índices e conjuntos utilizados, bem como os parâmetros envolvidos. Além disso, são especificadas as variáveis de decisão, a função objetivo que busca otimizar determinados critérios e as restrições necessárias para garantir a coerência e a factibilidade do modelo.

- **Abordagem de Solução:** É descrito o algoritmo que será implementado, com o objetivo de encontrar soluções rápidas e com melhores resultados. O algoritmo conta com a implementação de uma heurística construtiva que, em 3 etapas, gera rapidamente uma solução inicial e com um conjunto de vizinhanças que são exploradas em um processo iterativo, buscando encontrar melhores resultados.
- **Experimentos Computacionais e Resultados:** São conduzidos experimentos computacionais para validar e testar a eficácia do modelo matemático e do algoritmo propostos para um conjunto de 9 instâncias diferentes. São apresentados e analisados os resultados obtidos a partir desses experimentos. Para a instância do Campeonato Brasileiro - Série A 2023, os resultados reais são confrontados com os obtidos através dos dois métodos de solução. O comportamento do algoritmo, a eficiência e impacto de cada uma das vizinhanças e a influência de atribuir penalidades diferentes para os dois objetivos também são analisados.
- **Conclusões e Perspectivas Futuras:** Apresenta as principais conclusões derivadas do estudo realizado, destacando os resultados mais relevantes e as contribuições obtidas. Além disso, são discutidas possíveis perspectivas e direções futuras para o aprimoramento do modelo, visando abordar outras questões e desafios relacionados ao problema e do algoritmo, discutindo a possibilidade de explorar novas vizinhanças e estratégias.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a base teórica do Trabalho de Formatura, resultado do estudo de referências em Pesquisa Operacional aplicada ao campo de Esportes. Inicialmente, aborda-se o tema de forma ampla, explorando a relevância e a aplicabilidade dessa linha de pesquisa no contexto esportivo de maneira geral. Em seguida, é realizado um estudo mais específico, focado nos Problemas de *Scheduling*, abordando aspectos relacionados a diferentes esportes e suas particularidades. Por fim, é dado um maior detalhamento aos Problemas de *Timetabling* em campeonatos de futebol. A estruturação deste capítulo é ilustrada na Figura 1, destacando os tópicos abordados e suas respectivas seções.

Figura 1: Estrutura do Capítulo de Fundamentação Teórica.



Fonte: Elaborado pelo autor.

2.1 Aplicações de Pesquisa Operacional em Esportes

O termo Pesquisa Operacional foi cunhado durante a Segunda Guerra Mundial, quando líderes militares britânicos solicitaram a cientistas e engenheiros a análise de diversos pro-

blemas militares, como o posicionamento de radares e a gestão de comboios, bombardeios, operações antissubmarino e mineração (WINSTON, 2022).

No entanto, a comunidade acadêmica levou algum tempo para abraçar amplamente e com entusiasmo a realização de pesquisas que combinam Pesquisa Operacional e Esportes. Segundo Wright (2016), esse interesse ganhou força a partir da década de 1990, impulsionado por diversos fatores. Entre eles, destacam-se o avanço exponencial no poder de processamento dos computadores, o aprimoramento da teoria matemática aplicada e um crescente interesse nas análises esportivas, beneficiado pela disponibilidade em massa de dados volumosos e complexos, também conhecidos como *big data*. Essa conjunção de circunstâncias contribuiu para a consolidação de um campo de estudo cada vez mais dinâmico e promissor.

Atualmente, à medida que a área de Pesquisa Operacional em Esportes amadurece, surgem diversos temas de destaque que merecem atenção especial. Esses temas abrangem uma ampla gama de tópicos e aspectos relacionados à interseção entre a Pesquisa Operacional e o mundo dos esportes. Entre os temas proeminentes que emergem nesse contexto, destacam-se:

- I. **Ranqueamento:** Essa classe de problemas tem como objetivo responder a questões que sempre permearam a mente de todos os torcedores: “Quem é o melhor jogador/time? Como podemos estabelecer um ranking para eles?” À medida que se avança no campo da Pesquisa Operacional em Esportes, o desafio de realizar ranqueamentos justos e precisos se destaca como uma questão central. Através de métodos e técnicas analíticas avançadas, busca-se fornecer respostas embasadas e confiáveis para essas perguntas, levando em consideração uma variedade de critérios e medidas de desempenho.
 - Um exemplo de estudo é apresentado por Lozano et al. (2002), ao fazer uso de uma ferramenta amplamente reconhecida, denominada Análise por Envoltória de Dados (DEA), para avaliar a eficiência relativa e medir o desempenho das nações participantes nos últimos cinco Jogos Olímpicos de Verão até então. A abordagem proposta leva em consideração dois fatores de entrada, o Produto Nacional Bruto (PNB) e a população, e três fatores de saída, representados pelo número de medalhas de ouro, prata e bronze conquistadas. Os resultados desses últimos cinco Jogos Olímpicos de Verão são minuciosamente analisados, calculando-se índices de desempenho específicos e estabelecendo pontos de referência para cada país. Além disso, ao acompanhar a evolução do desempenho

de um país em diferentes edições dos jogos, é possível identificar tendências e avaliar de forma objetiva tanto os êxitos quanto as decepções.

II. Táticas e Estratégias: Essa categoria de problemas envolve a tomada de decisões estratégicas pelos jogadores com o objetivo de maximizarem suas probabilidades de alcançar a vitória.

- É exposto em Wright e Hirotsu (2003) um modelo calibrado com base em dados de todos os jogos da *Premier League* de 1999-2000 para determinar as circunstâncias em que um jogador “deveria” cometer uma falta tática visando beneficiar sua equipe. Os resultados são ilustrados usando dois jogos hipotéticos, um entre equipes equilibradas e outro em que uma das equipes é muito mais forte que a outra.
- Já em Clarke e Norman (2012), é utilizada Programação Dinâmica para investigar a estratégia ideal de desafio e obter algumas regras gerais. Por exemplo, em um jogo tradicional de tênis, os jogadores devem ser mais agressivos ao desafiar nos estágios finais dos sets e quando seu oponente está à frente. A estratégia ideal de desafio pode aumentar a chance de um jogador vencer uma partida de cinco sets equilibrada para 59%.

III. Regras de campeonatos: Essa classe de problemas visa explorar hipotéticas ou reais mudanças nos formatos e regras de uma competição, com o intuito de analisar seus eventuais impactos.

- Em Percy (2009), por exemplo, são analisados os impactos da mudança do sistema de pontuação no badminton. Inicialmente, a partida terminava em 11 pontos e só era possível pontuar em caso de saque próprio (quando se fazia “ponto” no saque do adversário, apenas alterava-se o sacador, não havendo pontuações). Posteriormente, as partidas passaram a se encerrar aos 21 pontos, sendo possível pontuar independentemente do sacador. O autor utilizou de probabilidade, simulação e Análise Bayesiana para mostrar que o novo sistema de pontuação poderia ser considerado mais justo e ainda tornava as partidas mais atrativas.

IV. Problemas de *Scheduling* Aplicados em Esportes: Essa classe de problemas é amplamente reconhecida e possui grande destaque entre acadêmicos e pesquisadores. Dentre os problemas mais comuns, destacam-se os desafios de otimização combinatória que envolvem a criação de cronogramas para definir os adversários, os horários e

os locais das partidas das equipes. A literatura registra uma diversidade significativa de restrições e objetivos a serem considerados. Essa complexidade e variedade tornam esses problemas um campo de estudo fascinante e desafiador para a comunidade acadêmica. Também são encontrados problemas relacionados à atribuição de árbitros às partidas e a encontrar um cronograma que maximize as oportunidades de realizar as atividades planejadas por um time ou atleta como viajar, treinar, jogar e descansar, por exemplo, de acordo com a periodização estabelecida.

Na próxima seção, serão detalhados os Problemas de *Scheduling* Aplicados em Esportes, dando uma visão geral sobre os principais temas presentes na literatura.

2.2 Problemas de *Scheduling* Aplicados em Esportes

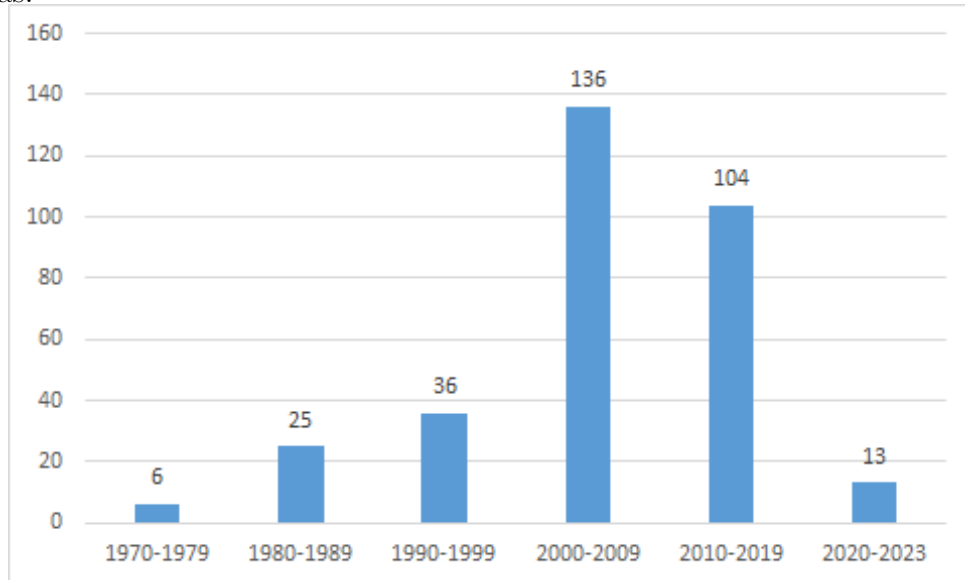
O estudo de agendamento remonta à década de 1950. Pesquisadores em Pesquisa Operacional, Engenharia Industrial e Administração se depararam com o problema de gerenciar várias atividades que ocorrem em uma oficina (LEUNG, 2004). Apesar de sua origem em sistemas de manufatura e em serviços industriais, hoje, é possível encontrar aplicações em diversas áreas como saúde, transporte, educação e até mesmo esportes, já que, de maneira simplista, conforme Pinedo (2012), problemas de agendamento (ou *scheduling problems*) lidam com a alocação de recursos para tarefas ao longo de períodos de tempo determinados e têm como objetivo otimizar um ou mais objetivos.

Problemas de *Scheduling* Aplicados em Esportes envolvem a organização eficiente de eventos esportivos, como jogos, competições e treinamentos. O objetivo é encontrar uma distribuição adequada de tempo e recursos para atender às necessidades de diferentes partes interessadas, como equipes, jogadores, árbitros e instalações esportivas. O agendamento adequado busca equilibrar fatores como disponibilidade de recursos, restrições específicas, preferências e objetivos desejados. Esses problemas podem surgir em vários contextos esportivos, desde ligas e torneios até eventos de equipe ou individuais. A otimização do agendamento é crucial para garantir condições iguais para as equipes, maximizar a utilização de instalações esportivas, minimizar conflitos de agenda, considerar restrições de viagem e promover a justiça competitiva. A literatura sobre Problemas de *Scheduling* Aplicados em Esportes abrange uma ampla gama de tópicos e aplicações, demonstrando a importância desse campo no planejamento eficiente e na organização de eventos esportivos.

Na Figura 2, é ilustrado o aumento das contribuições de Pesquisa Operacional relaci-

onadas ao agendamento esportivo nas últimas 6 décadas.

Figura 2: Publicações sobre aplicações de PO para Agendamento Esportivo nas últimas 6 décadas.



Fonte: Knust (2018). Elaborado pelo autor.

Segundo Bulck et al. (2020), o pico no número de publicações na primeira década deste século é parcialmente explicado pela introdução do *Travelling Tournament Problem*, um dos problemas dessa classe, cujo objetivo é minimizar a distância total percorrida pelos times ao longo de um campeonato.

Os três principais temas encontrados na literatura são: Atribuição de Árbitros, Geração de Cronograma de Atividades e Geração de Tabelas de Campeonatos. Neste último está contido o problema abordado neste Trabalho de Formatura. Os três temas levantados são mais bem descritos a seguir:

- I. Atribuição de Árbitros: Um problema inovador na gestão esportiva é o Problema de Atribuição de Árbitros, no qual um número limitado de árbitros com diferentes qualificações e disponibilidades deve ser atribuído a um conjunto de jogos já agendados (DUARTE; RIBEIRO; URRUTIA, 2007). Segundo Alarcón, Durán e Guajardo (2014), esse problema tem aumentado o interesse recente e, tipicamente, tem uma estrutura combinatória que é praticamente impossível de ser resolvida de forma manual.
- II. Geração de Cronograma de Atividades: Conceitualmente, o problema consiste em encontrar um cronograma de voos e treinos para a pré-temporada e temporada regu-

lar que maximize as oportunidades para executar os conteúdos periodizados (como viagens, treinos, jogos e folgas). Essa atividade é realizada levando em consideração as restrições conhecidas (como calendário competitivo, disponibilidade do elenco, condições climáticas, eventos especiais, feriados e o efeito emocional dos dias ausentes) (FRY; OHLMANN, 2012).

- III. Problemas de Geração de Tabelas de Campeonatos: Todas as ligas esportivas precisam agendar suas partidas. Frequentemente, esse é um processo simples, utilizando padrões prontamente disponíveis. No entanto, na prática, pode haver mais objetivos e/ou restrições do que os padrões convencionais podem lidar facilmente, e, portanto, é frequentemente necessário o uso de *software* especializado, utilizando algum tipo de algoritmo de Pesquisa Operacional (WRIGHT, 2009). Conforme Ribeiro (2012), o problema consiste em determinar a data e o local em que cada partida de um torneio será realizada, e tal aplicação pode ser encontrada nos mais diversos esportes, como futebol, beisebol, basquete, críquete e hóquei.

2.3 Problemas de Geração de Tabelas de Campeonatos

O processo típico de desenvolvimento de tabelas esportivas na prática é iterativo e consiste em determinar como formular matematicamente os objetivos e requisitos expressos pelos interessados, elaborar uma abordagem de solução para obter uma ou mais tabelas, realizar várias rodadas de apresentação da(s) tabela(s) aos interessados para obter *feedback* e incorporar o *feedback* na forma de novas restrições ou objetivos para gerar novos cronogramas (FRY; OHLMANN, 2012).

2.3.1 Terminologia

Na vasta literatura sobre Problemas de Geração de Tabelas de Campeonatos, é possível encontrar uma ampla variedade de casos, cada um com suas particularidades e diferenças específicas. Compreender e dominar a terminologia utilizada nessa área é de suma importância para uma análise aprofundada e precisa desses problemas. A seguir, serão apresentados alguns dos principais conceitos no que concerne a tabelas de campeonatos esportivos.

- I. Formato: na literatura, os campeonatos esportivos são frequentemente categorizados

em dois grupos distintos com base no formato adotado: aqueles que seguem o formato *Round-Robin* e aqueles que não o seguem. Torneios *Round-Robin* são aqueles que envolvem n equipes, em que cada equipe deve jogar exatamente m vezes contra todas as outras equipes dentro de um número fixo de rodadas. Os tipos mais comuns são os *Round-Robins* Únicos (*Single Round-Robin* - SRR), em que $m = 1$, e os *Round-Robins* Duplos (*Double Round-Robin* - DRR), em que $m = 2$ (nestes últimos, as equipes normalmente são programadas para se enfrentarem uma vez no local de cada uma das equipes) (LEWIS; THOMPSON, 2011).

- II. Compacidade: segundo Trick (2003), um torneio pode ser dito compacto se, e somente se, cada time jogar apenas uma vez por rodada, totalizando, assim $n - 1$ (sendo n o número de times que disputa a competição) rodadas em um campeonato de turno único (torneio categorizado como *SRR*), por exemplo.
- III. Simetria: Válida apenas para torneios com mais de um turno (ou seja, com $m > 1$, adotando a nomenclatura utilizada), a simetria pode ser classificada de diferentes maneiras detalhadas a seguir e ilustradas na Tabela 1, a partir dos jogos do time A dentro de um campeonato com 6 equipes e 2 turnos.
 - Espelhada (ou *Mirrored*): Nessa configuração, os jogos da rodada 1 serão os mesmos da rodada $1 + n$ (ou primeira rodada do segundo turno), os jogos da rodada 2 serão os mesmos da rodada $2 + n$ (ou segunda rodada do segundo turno) e assim em diante até a rodada $n - 1$ (para o caso de um torneio do tipo DRR), invertendo apenas os mandos de campo entre as equipes.
 - Invertida: Nessa configuração, as rodadas do segundo turno são jogadas na ordem inversa às rodadas do primeiro turno (para o caso de um torneio do tipo DRR).
 - Inglesa: Nessa configuração, os confrontos da primeira rodada do segundo turno (considerando, mais uma vez, um torneio do tipo DRR), são os mesmos da última rodada do primeiro turno. As demais partidas são distribuídas de forma que as duas partidas envolvendo os mesmos dois times estejam sempre separadas por um intervalo de n rodadas.
 - Francesa: Nessa configuração, os jogos disputados na primeira rodada do primeiro turno são os mesmos disputados na última rodada do segundo turno (para o caso de um torneio do tipo DRR). As demais partidas são distribuídas de forma que as duas partidas envolvendo os mesmos dois times estejam sempre separadas por um intervalo de $n - 2$ rodadas.

Tabela 1: Diferentes tipos de simetria para os jogos do time A.

	1	2	3	4	5		6	7	8	9	10
Espelhada	A x B	A x C	D x A	A x F	E x A		B x A	C x A	A x D	F x A	A x E
Invertida	A x B	A x C	D x A	A x F	E x A		A x E	F x A	A x D	C x A	B x A
Inglesa	A x B	A x C	D x A	A x F	E x A		A x E	B x A	C x A	A x D	F x A
Francesa	A x B	A x C	D x A	A x F	E x A		C x A	A x D	F x A	A x E	B x A

Fonte: Elaborado pelo autor.

2.3.2 Definição de Variáveis

Os modelos de Problemas de Geração de Tabelas de Campeonatos empregam diversas variáveis que estão diretamente relacionadas à realização de partidas entre dois times ou dois atletas. Essas variáveis são essenciais para determinar o momento e o local em que essas equipes (ou atletas) se enfrentarão ao longo do campeonato. É importante ressaltar que não há um padrão único de variáveis de decisão, pois isso depende do modelo específico utilizado. A seguir, serão apresentados alguns exemplos de variáveis frequentemente encontradas em modelos de Problemas de Geração de Tabelas de Campeonatos.

- Em Rasmussen (2008) é utilizada uma variável binária que indica se determinado time receberá um determinado padrão de sequência de jogos como mandante e visitante.
- Em Rasmussen e Trick (2009) é utilizada uma variável binária que indica se um determinado time jogará em determinada rodada em casa ou fora.
- Em Trick (2004) é utilizada uma variável binária que indica se dois times se enfrentarão em determinada rodada.
- Em Melo, Urrutia e Ribeiro (2007) é utilizada uma variável binária que indica se um determinado time joga como mandante contra outro time determinado em uma rodada específica.

2.3.3 Funções Objetivo

Ao analisar a literatura, é possível identificar a abordagem de cinco principais objetivos no contexto de Problemas de Agendamento Esportivo. Esses objetivos representam as

metas a serem alcançadas no desenvolvimento de um cronograma otimizado. Cada objetivo possui características distintas e pode ser priorizado de acordo com as necessidades e preferências específicas de cada competição. Vale destacar que, em caso de não haver um objetivo bem definido, o problema se reduz a encontrar uma tabela de jogos que respeite as restrições existentes.

Na Tabela 2, é exibido um exemplo de torneio *Single Round-Robin*, ou seja, com $m = 1$, compacto, composto por 10 times a fim de ajudar a ilustrar alguns conceitos. Destaca-se que em todo este trabalho, o time identificado primeiro dentro de uma partida é o time mandante e, conseqüentemente, o time identificado na sequência é o visitante, ou seja a partida “A x B” indica que o time “A” joga em casa contra o time “B” naquela ocasião.

Tabela 2: Exemplo de torneio *Single Round-Robin*, compacto com 10 times.

Rodadas								
1	2	3	4	5	6	7	8	9
D x B	E x H	D x A	F x D	I x H	A x C	C x I	E x A	G x J
H x F	G x D	J x E	A x B	C x J	F x G	D x J	F x C	B x I
J x A	A x I	I x F	G x C	D x E	E x B	B x H	H x G	C x D
I x G	F x J	C x H	E x I	B x F	H x D	G x E	J x B	A x H
C x E	B x C	B x G	H x J	G x A	J x I	A x F	I x D	F x E

Fonte: Elaborado pelo autor.

- I. Minimização do número de *Breaks*: Um dos objetivos clássicos nessa categoria de problemas é a minimização do número de ocorrências em que as equipes jogam partidas consecutivas como mandante ou como visitante. Geralmente, é preferível uma tabela com poucas *breaks* em relação a uma tabela com muitas *breaks* (MIYASHIRO; IWASAKI; MATSUI, 2003). Em outras palavras, em um cenário ideal, os times alternariam partidas em casa e fora ao longo de todo o campeonato. Existem várias razões pelas quais as *breaks* devem ser minimizadas em um cronograma esportivo: os fãs não gostam de períodos longos sem jogos em casa, jogos consecutivos em casa reduzem a receita das bilheterias e sequências longas de jogos em casa ou fora podem influenciar a posição atual da equipe no torneio (NURMI et al., 2010).

A fim de ilustração, é exibida na Tabela 3 o número de *breaks* para cada time dentro do campeonato exemplificado na Tabela 2. O valor “c” representa que o time jogará a rodada em questão como mandante e “f” indica que o time jogará fora de casa naquela rodada. Valores destacados com sublinhado e negrito indicam a ocorrência de uma *break*.

Tabela 3: Ocorrência de *breaks* para o exemplo elaborado.

Times	Rodadas									<i>breaks</i>
	1	2	3	4	5	6	7	8	9	
A	f	c	f	c	f	c	<u>c</u>	f	c	1
B	f	c	<u>c</u>	f	c	f	c	f	c	1
C	c	f	c	f	c	f	c	f	c	0
D	c	f	c	f	c	f	c	f	<u>f</u>	1
E	f	c	f	c	f	c	f	c	f	0
F	f	c	f	c	f	c	f	c	<u>c</u>	1
G	f	c	f	c	<u>c</u>	f	c	f	c	1
H	c	f	<u>f</u>	c	f	c	f	c	f	1
I	c	f	c	f	c	f	<u>f</u>	c	f	1
J	c	f	c	f	<u>f</u>	c	f	c	f	1

Fonte: Elaborado pelo autor.

- II. *Traveling Tournament Problem*: Como mencionado na Seção 2.2, esse problema surgiu no início deste século como um importante impulsionador no aumento do número de pesquisas sobre o agendamento esportivo. O objetivo do *Traveling Tournament Problem* é encontrar uma tabela de jogos que minimize a soma das distâncias viajadas pelos times durante a competição. As instâncias desse problema parecem ser muito difíceis mesmo para um número muito pequeno de equipes, o que o torna um desafio interessante para técnicas de otimização combinatória (EASTON; NEMHAUSER; TRICK, 2001). Segundo Durán et al. (2021), as tabelas de jogos geradas podem resultar em diferenças significativas nas distâncias de viagens entre as equipes, o que é uma situação frustrante tanto para os dirigentes, treinadores e jogadores dos times, mas quase impossível de evitar com técnicas manuais de agendamento da temporada.
- III. Minimização de Custos: Tem-se em Briskorn e Drexler (2009a) que, para cada partida possível, é associado um custo, isto é, realizar determinada partida em determinado estádio em determinada data gera um custo. Analogamente, pode-se pensar em associar uma receita a cada um dos enfrentamentos, como a obtida com a venda de ingressos. Dessa forma, essa categoria de problemas visa construir uma tabela que minimize a soma dos custos, maximize as receitas, ou, até mesmo, combine esses dois objetivos.
- IV. Minimização de Efeitos de *Carry Over*: Qualquer tabela para um torneio *Round-Robin* envolve uma ordem em que cada equipe enfrenta seus oponentes. Dizemos

que uma equipe i dá um *carry over* para uma equipe j se o jogo de alguma outra equipe t contra i for seguido por um jogo de t contra a equipe j . Por exemplo, se a equipe i for muito forte e jogar de forma intensa, pode-se imaginar que seu oponente, equipe t , fique enfraquecido por lesões ou fadiga, o que poderia ser uma vantagem para seu próximo oponente, equipe j . Além disso, o efeito do *carry over* também pode ser relevante em uma interpretação estritamente psicológica, quando a equipe t perde confiança e moral após uma derrota severa contra a equipe forte i , novamente em benefício de seu próximo oponente, equipe j . O oposto pode ser verdadeiro se a equipe i for fraca. Claramente, os efeitos de *carry over* são inevitáveis em qualquer tabela, mas as tabelas podem diferir na medida em que os efeitos de *carry over* são equilibrados entre as equipes (GOOSSENS; SPIEKSMAN, 2012). Segundo Guedes e Ribeiro (2011), o valor dos efeitos de *carry over* é uma das várias medidas que se pode considerar para avaliar a qualidade de um torneio do tipo *Round-Robin*.

A partir da Tabela 2, é possível gerar uma tabela adicional que auxilia na compreensão dos valores de *carry over*. A Tabela 4 indica quais serão os adversários de cada time em cada rodada.

Tabela 4: Adversários em cada rodada.

Times	Rodadas								
	1	2	3	4	5	6	7	8	9
A	J	I	D	B	G	C	F	E	H
B	D	C	G	A	F	E	H	J	I
C	E	B	H	G	J	A	I	F	D
D	B	G	A	F	E	H	J	I	C
E	C	H	J	I	D	B	G	A	F
F	H	J	I	D	B	G	A	C	E
G	I	D	B	C	A	F	E	H	J
H	F	E	C	J	I	D	B	G	A
I	G	A	F	E	H	J	C	D	B
J	A	F	E	H	C	I	D	B	G

Fonte: Elaborado pelo autor.

A partir da Tabela 4, é possível, enfim, montar a Tabela 5 contendo a soma dos *carry over* dados pelo time i para os demais times j , e a soma dos valores de *carry over* ao quadrado dados pelo time i para os demais times j .

Tabela 5: Valores de *carry over* para o exemplo elaborado.

Times	Times										SCO ¹	SCOQ ²
	A	B	C	D	E	F	G	H	I	J		
A	0	0	1	0	0	7	0	0	1	0	9	51
B	0	0	1	0	0	0	7	1	0	0	9	51
C	1	1	0	1	1	1	1	1	1	1	9	9
D	0	7	1	0	1	0	0	0	0	0	9	51
E	0	1	1	0	0	0	0	7	0	0	9	51
F	0	0	1	1	7	0	0	0	0	0	9	51
G	7	0	1	0	0	0	0	0	0	1	9	51
H	0	0	1	0	0	0	1	0	0	7	9	51
I	0	0	1	7	0	1	0	0	0	0	9	51
J	1	0	1	0	0	0	0	0	7	0	9	51

Fonte: Elaborado pelo autor.

Em um campeonato de turno único, a soma de *carry over* dados por cada time i sempre será igual a $n - 1$. Isso ocorre devido ao fato de que para cada rodada em que o time i enfrenta o time t , haverá um time j que jogará contra o time t na rodada seguinte. Nesse contexto, na literatura, o objetivo é minimizar a soma de cada quantidade de *carry over* dada por cada time i para cada time j elevada ao quadrado, como uma forma de balancear os valores entre os times, conferindo condições iguais entre os mesmos. Em outras palavras, o objetivo é minimizar a soma de todos os elementos a_{ij} da Tabela 5 quando elevados ao quadrado. A essa soma será dado o nome efeitos de *carry over*.

- V. Minimização de *Soft Constraints*: Quando não há um objetivo específico, como os mencionados anteriormente, essa classe de problemas busca minimizar as penalidades decorrentes de violações das restrições *soft*, ao mesmo tempo em que respeita as restrições *hard*. Isso implica em encontrar soluções que busquem reduzir ao máximo as violações das restrições *soft*, considerando-as como penalidades a serem evitadas, enquanto garantem o cumprimento das restrições *hard* que são consideradas obrigatórias e não podem ser violadas. As restrições *soft* contempladas podem ser de qualquer classe citada a seguir.

¹SCO: Soma de *carry over* dados pelo time i para os demais times j .

²SCOQ: Soma de *carry over* dados pelo time i para os demais times j ao quadrado.

2.3.4 Restrições Contempladas

Outra distinção importante entre os problemas é o conjunto de restrições empregado. Em Bulck et al. (2020), as restrições são categorizadas em 5 classes, a saber:

1. Restrições de Capacidade: Obrigam um time a jogar como mandante ou visitante e controlam o número total de partidas disputadas por um time ou conjunto de times.
2. Restrições de Jogo: Impõem ou obrigam partidas específicas a acontecerem em datas ou rodadas determinadas.
3. Restrições de *Break*: Controlam a frequência e ocorrência das *breaks* ao longo da competição.
4. Restrições de justiça competitiva e atratividade: Contemplam as restrições que visam aumentar a competitividade do torneio e ampliar a atratividade dos fãs. Essa é uma categoria muito ampla de restrições. Knust e Thaden (2006) trazem o conceito de tabela balanceada como sendo aquela em que a diferença de jogos como mandante e visitante é no máximo 1 ao final do torneio, o que pode ser de grande importância em campeonatos nos quais os times disputem uma quantidade ímpar de jogos, como em um campeonato do tipo *Round-Robin* simples com um número par de times.
5. Restrições de Separação: De modo geral, são responsáveis por controlar o número de rodadas entre partidas envolvendo os mesmos times, garantindo assim, a simetria desejada para o torneio.

Na Tabela 6 estão disponíveis publicações de Problemas de Geração de Tabelas de Campeonatos. Tanto a estrutura, como as restrições contempladas e o objetivo dos modelos estão classificados conforme Bulck et al. (2020). Foi adicionada ao final, uma linha referente a este Trabalho de Formatura.

Tabela 6: Problemas de Geração de Tabelas de Campeonatos.

Referência	Estrutura			Restrições Contempladas					Objetivo ³
	Formato ⁴	Compacidade ⁵	Simetria ⁶	Capacidade	Jogo	Breaks	Just. e Atrat.	Separação	
Nemhauser e Trick (1998)	2RR	C		X	X	X		X	∅
Zhang (2002)	NRR	C		X					∅
Voorhis (2005)	NRR	NC		X	X				∅
Kostuk e Willoughby (2012)	NRR	NC		X	X	X			∅
Werra (1985)	1RR	C		X		X			BR
Croce e Oliveri (2006)	2RR	C	M	X					BR
Saur et al. (2012)	1RR	C		X		X			BR
Durán, Guajardo e Sauré (2017)	2RR	C	F	X					BR
Ball e Webster (1977)	2RR	NC	P	X	X				TR
Bean e Birge (1980)	NRR	NC		X					TR
Russell e Leung (1994)	NRR	C		X					TR
Easton, Nemhauser e Trick (2001)	2RR	C		X				X	TR
Ribeiro e Urrutia (2007a)	2RR	C	M	X				X	TR
Hoshino e Kawarabayashi (2011a)	8RR	C	P	X			X	X	TR
Hoshino e Kawarabayashi (2011b)	2RR	C	M	X				X	TR
Bonomo et al. (2012)	2RR	C	M	X		X			TR
Durán et al. (2007)	1RR	C		X	X	X			CR
Briskorn e Drexel (2009b)	1RR	C		X	X	X			CR
Fiallos et al. (2010)	2RR	C	M	X		X			CR
Durán, Guajardo e Wolf-Yadlin (2012)	4RR	C	M	X	X	X	X		CR
Guedes e Ribeiro (2011)	1RR	C							CO
Günneç e Demir (2019)	2RR	C	M			X			CO
Ferland e Fleurent (1991)	NRR	NC		X	X		X	X	SC
Schreuder (1992)	2RR	C	M	X		X			SC
Costa (1995)	NRR	NC		X	X		X	X	SC
Wright (1994)	NRR	NC		X	X		X		SC
Schonberger, Mattfeld e Kopfer (2000)	1RR	NC		X			X		SC
Easton (2003)	2RR	C		X		X		X	SC
Wright (2005)	4RR	NC		X	X		X	X	SC
Bartsch, Drexel e Kröger (2006)	2RR	C	M	X	X	X		X	SC
Wright (2006)	2RR	NC		X	X		X	X	SC
Ribeiro e Urrutia (2007b)	2RR	C	M	X		X	X		SC
Rasmussen (2008)	3RR	C	P	X	X	X		X	SC
Goossens e Spieksma (2009)	2RR	C	M	X	X	X			SC
Kyngas e Nurmi (2009)	NRR	NC		X	X	X	X	X	SC
Knust (2010)	1RR	NC		X	X	X	X		SC
Lewis e Thompson (2011)	2RR	C		X				X	SC
Ribeiro e Urrutia (2012)	2RR	C	M	X	X	X			SC
Nurmi, Goossens e Kyngäs (2014)	3RR	NC		X	X	X	X	X	SC
Recalde, Torres e Vaca (2013)	2RR	C	I	X		X			SC
Larson e Johansson (2014)	NRR	C		X		X	X	X	SC
Westphal (2014)	2RR	C	P	X	X	X	X		SC
Kyngäs et al. (2017)	NRR	NC		X	X	X	X	X	SC
Cocchi et al. (2018)	2RR	C	M	X	X	X	X		SC
Bulck, Goossens e Spieksma (2019)	2RR	NC		X				X	SC
<i>Este Trabalho de Formatura</i>	2RR	C	M	X		X			BR, CO

Fonte: Bulck et al. (2020). Adaptado pelo autor.

³∅ : Não existe um objetivo associado; BR: Minimização do número de Breaks; TR: *Traveling Tournament Problem*; CR: *Minimização de Custos*; CO: *Minimização de Efeitos de Carry Over*

⁴NRR: Campeonatos que não possuem formato *Round-Robin*; mRR: Campeonatos com formato *Round-Robin* com *m* turnos.

⁵C: Campeonatos compactos; NC: Campeonatos não compactos.

⁶M: Simetria Espelhada; I: Simetria Invertida; P: Campeonatos faseados sem simetrias adicionais.

2.4 Abordagens de Solução para Problemas de Geração de Tabelas de Campeonatos

Nesta seção serão explorados alguns dos principais métodos de solução empregados em publicações sobre Problemas de Geração de Tabelas de Campeonatos. Serão discutidas Abordagens de Decomposição, Programação Inteira, *Constraint Programming*, Algoritmos de Busca Local e Algoritmos Evolucionários.

1. Abordagens de Decomposição: Problemas de geração de tabelas esportivas frequentemente são decompostos em subproblemas que são resolvidos sequencialmente por algoritmos exatos ou heurísticos. Duas abordagens importantes podem ser distinguidas:
 - (a) *First-schedule-then-break*: Primeiramente, são determinados os confrontos entre as n equipes para cada rodada (ou seja, quais equipes jogam uma contra a outra em uma rodada). Em seguida, é calculado um padrão correspondente de jogos em casa e fora (com um número mínimo de *breaks*) para esses confrontos.
 - (b) *First-break-then-schedule*: Primeiramente, é determinado um padrão viável de jogos em casa e fora (com um número mínimo de *breaks*), e posteriormente são fixados os confrontos correspondentes a esse padrão (DREXL; KNUST, 2007).
2. Programação Inteira: A programação inteira é uma área da otimização matemática que se concentra na resolução de problemas de otimização nos quais as variáveis de decisão são limitadas a valores inteiros. Ela é aplicada em diversas situações em que as decisões precisam ser tomadas de forma discreta, como na distribuição de recursos, no planejamento de projetos e na elaboração de cronogramas. O objetivo da programação inteira é encontrar a combinação ideal de valores inteiros para as variáveis de decisão, de modo a otimizar uma determinada função objetivo, ao mesmo tempo em que são respeitadas as restrições impostas pelo problema. Existe uma tradição relativamente longa de usar a programação inteira para agendar torneios esportivos. Seu uso tem sido particularmente popular no futebol (DURÁN; GUAJARDO; SAURÉ, 2017).
3. *Constraint Programming*: Poderoso paradigma para resolver problemas de busca combinatória que se baseia em uma ampla gama de técnicas provenientes da Inteligência Artificial, Pesquisa Operacional, Algoritmos, Teoria dos Grafos e outras áreas. A ideia básica em *Constraint Programming* é que o usuário especifica as res-

trições e um solucionador de restrições de propósito geral é utilizado para resolvê-las (ROSSI; BEEK; WALSH, 2008).

4. Algoritmos Baseados em Busca Local: Essencialmente, a busca local consiste em mover-se de uma solução para outra em seu espaço de solução de acordo com regras bem definidas (PIRLOT, 1996). No contexto de Problemas de Geração de Tabelas de Campeonatos dois algoritmos aparecem com destaque:

- Busca Tabu: O princípio básico da busca tabu é prosseguir com a busca sempre que um ótimo local for encontrado, permitindo movimentos que não melhorem a solução. O retorno a soluções anteriormente visitadas é evitado pelo uso de memórias, chamadas de listas tabu, que registram o histórico recente da busca (GENDREAU; POTVIN, 2005).
- *Simulated Annealing*: Muitos problemas de otimização combinatória têm se mostrado como NP-difíceis, o que significa que o tempo de execução para quaisquer algoritmos atualmente conhecidos que garantam uma solução ótima é uma função exponencial do tamanho do problema. O SA (*Simulated Annealing*) é uma das muitas abordagens meta-heurísticas projetadas para fornecer uma solução boa, embora não necessariamente ótima, dentro de um tempo computacional razoável (EGLESE, 1990).

5. Algoritmos Evolucionários: Algoritmos que realizam tarefas de otimização ou aprendizado com capacidade de evoluir (YU; GEN, 2010).

3 DESCRIÇÃO DO PROBLEMA

Neste capítulo é fornecida uma visão geral do campeonato esportivo em análise, juntamente com a problemática abordada. Será apresentada uma breve descrição do referido campeonato, utilizando os termos e conceitos discutidos na Seção 2.3.1. Em seguida, serão discutidas as características específicas do problema, que serão, no capítulo seguinte, traduzidas em um modelo matemático, incluindo variáveis e restrições cruciais, bem como índices, conjuntos e parâmetros relevantes.

3.1 Campeonato Estudado

O Campeonato Brasileiro de Futebol é a principal competição de futebol no Brasil, compreendendo quatro divisões principais: Série A, Série B, Série C e Série D. Organizado pela Confederação Brasileira de Futebol (CBF), o torneio atrai a atenção de milhões de torcedores brasileiros e entusiastas do futebol em todo o mundo. Com uma história rica e uma base de fãs apaixonados, o Campeonato Brasileiro de Futebol - Série A apresenta, atualmente, um formato de pontos corridos, segundo a estrutura de *Mirrored Double Round-Robin* (ou seja, possui 2 turnos com formato do tipo *Round-Robin* e possui simetria do tipo espelhada) compacto, com o objetivo de conquistar o título e garantir vagas em competições internacionais, além de assegurar a permanência na divisão principal, evitando o rebaixamento para a segunda divisão.

A Série A do Campeonato Brasileiro de Futebol reúne os principais clubes do país, conhecidos por sua tradição, rivalidades históricas e elencos estrelados. A cada temporada, essas equipes se enfrentam em jogos emocionantes, em busca do título de campeão brasileiro. Já a Série B é marcada pela luta pela promoção à elite do futebol brasileiro, com clubes buscando o acesso à Série A e a oportunidade de competir com os grandes times do país. Ambas as divisões oferecem intensidade, emoção e disputas acirradas, proporcionando aos torcedores uma experiência única e apaixonante. O Campeonato Brasileiro de Futebol é um dos eventos mais aguardados do calendário esportivo brasileiro, celebrando

a grandeza e a paixão pelo futebol em todas as suas formas.

Neste Trabalho de Formatura, será abordada a Série A do Campeonato Brasileiro de Futebol, mas com a ressalva de que o mesmo problema poderia ser estendido para a segunda divisão que, por sua vez, segue o mesmo formato da divisão principal.

3.2 Problema Abordado

O problema abordado neste Trabalho de Formatura será a geração da tabela do Campeonato Brasileiro de Futebol - Série A, considerando critérios de justiça competitiva entre as equipes, a partir do uso de técnicas de Pesquisa Operacional. Nesta seção, serão explorados os detalhes do problema com base na revisão de literatura realizada no Capítulo 2.

Como mencionado na Seção 2.3.3, Problemas de Geração de Tabelas de Campeonatos podem possuir objetivos distintos. No caso deste Trabalho de Formatura, o foco será dado a tornar o calendário do Brasileirão (como é popularmente conhecido) mais justo (do inglês “*fair*”), considerando dois critérios: ocorrências de *breaks* e efeitos de *carry over*.

A título de ilustração, na Tabela 7 é possível visualizar os adversários de cada um dos 20 times ao longo das 38 rodadas para a edição do ano de 2023, conforme a tabela gerada e divulgada pela CBF. Já nas Tabelas 8 e 9 são exibidos as ocorrências de *breaks* e os valores de *carry over* considerando os times e jogos envolvidos na mesma edição.

Tabela 7: Tabela do Campeonato Brasileiro de Futebol - Série A 2023.

Times ¹	Rodadas																																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38				
América/MG	FLU	SPO	SAN	CUI	RBR	CRU	FOR	BOT	COR	CAP	GRE	INT	CAM	CTB	VAS	FLA	PAL	BAH	GOI	FLU	SPO	SAN	CUI	RBR	CRU	FOR	BOT	COR	CAP	GRE	INT	CAM	CTB	VAS	FLA	PAL	BAH	GOI				
Athletico/PR	GOI	FLU	CAM	FLA	INT	CTB	RBR	GRE	BOT	AMG	SPO	COR	PAL	FOR	BAH	VAS	CRU	SAN	CUI	GOI	FLU	CAM	FLA	INT	CTB	RBR	GRE	BOT	AMG	SPO	COR	PAL	FOR	BAH	VAS	CRU	SAN	CUI				
Atlético/MG	VAS	SAN	CAP	BOT	CUI	INT	CTB	PAL	FOR	CRU	PAL	FOR	AMG	COR	GOI	GRE	FLA	SPO	AMG	BAH	RBR	BOT	VAS	CTB	SAN	FLA	GOI	INT	FOR	CRU	PAL	FLU	GRE	CUI	CAP	RBR	SAN	CTB	CRU	INT		
Bahia	RBR	BOT	VAS	CTB	SAN	FLA	GOI	INT	FOR	CRU	PAL	FOR	FLU	GRE	CUI	CAP	GRE	SPO	AMG	BAH	RBR	BOT	VAS	CTB	SAN	FLA	GOI	INT	FOR	CRU	PAL	FLU	GRE	CUI	CAP	RBR	SAN	CTB	CRU	INT		
Botafogo	SPO	BAH	FLA	CAM	COR	GOI	FLU	AMG	CAP	FOR	CUI	PAL	VAS	GRE	RBR	SAN	CTB	CRU	INT	SPO	BAH	FLA	CAM	COR	GOI	FOR	BOT	SPO	FLA	FLU	AMG	CAP	FOR	CUI	PAL	VAS	GRE	RBR	SAN	CTB	CRU	INT
Corinthians	CRU	GOI	PAL	FOR	BOT	SPO	FLA	FLU	AMG	CUI	SAN	CAP	RBR	CAM	GRE	BAH	VAS	INT	CTB	CRU	GOI	PAL	FOR	SPO	BAH	VAS	CAP	CAM	CUI	PAL	SAN	INT	GRE	GOI	AMG	CRU	FLU	BOT	RBR	COR		
Coritiba	FLA	FOR	SPO	BAH	VAS	CAP	CAM	CUI	PAL	SAN	INT	GRE	GOI	AMG	CRU	FLU	BOT	RBR	COR	FLA	FOR	SPO	BAH	VAS	CAP	CAM	CUI	PAL	SAN	INT	GRE	GOI	AMG	CRU	FLU	BOT	RBR	COR				
Cruzeiro	COR	GRE	RBR	SAN	FLU	AMG	CUI	FLA	CAM	BAH	FOR	SPO	INT	VAS	CTB	GOI	CAP	BOT	PAL	CAP	PAL	GRE	RBR	SAN	FLU	AMG	CUI	FLA	CAM	BAH	FOR	SPO	INT	VAS	CTB	GOI	CAP	BOT	PAL			
Cuiabá	PAL	RBR	GRE	AMG	CAM	FLU	CRU	CTB	GOI	COR	BOT	VAS	SAN	BAH	FOR	SPO	INT	FLA	CAP	PAL	RBR	GRE	AMG	CAM	FLU	CRU	CTB	GOI	COR	CRU	VAS	GRE	RBR	SAN	FOR	PAL	FLU	AMG	CAM	CUI	SPO	
Flamengo	CTB	INT	BOT	CAP	GOI	BAH	COR	CRU	VAS	GRE	RBR	SAN	FOR	PAL	FLU	AMG	CAM	CUI	SPO	CTB	INT	BOT	CAP	GOI	BAH	CRU	COR	CRU	VAS	GRE	RBR	SAN	FOR	PAL	FLU	AMG	CAM	CUI	SPO			
Fluminense	AMG	CAP	FOR	VAS	CRU	CUI	BOT	COR	RBR	GOI	CAM	BAH	SPO	INT	FLA	CTB	SAN	PAL	GRE	AMG	CAP	FOR	VAS	CRU	CUI	BOT	COR	RBR	GOI	CAM	BAH	SPO	INT	FLA	CTB	SAN	PAL	GRE				
Fortaleza	INT	CTB	FLU	COR	SPO	GRE	AMG	VAS	BAH	BOT	CRU	CAM	FLA	CAP	CUI	PAL	RBR	GOI	SAN	INT	CTB	FLU	COR	SPO	CRU	SPO	GRE	AMG	VAS	BAH	BOT	CRU	CAM	FLA	CAP	CUI	PAL	RBR	GOI	SAN		
Goiás	CAP	COR	INT	PAL	FLA	BOT	BAH	SPO	CUI	FLU	VAS	RBR	CTB	GOI	CAM	CRU	GRE	FOR	AMG	CAP	COR	INT	PAL	FLA	BOT	BAH	SPO	CUI	FLU	VAS	RBR	CTB	GOI	CAM	CRU	GRE	FOR	AMG				
Grêmio	SAN	CRU	CUI	RBR	PAL	FOR	INT	CAP	SPO	FLA	AMG	CTB	BAH	BOT	COR	CAM	GOI	VAS	FLU	SAN	CRU	CUI	RBR	PAL	FOR	INT	CAP	SPO	FLA	AMG	CTB	BAH	BOT	COR	CAM	GOI	VAS	FLU	AMG			
Internacional	FOR	FLA	GOI	SPO	CAP	CAM	GRE	BAH	SAN	VAS	CTB	AMG	CRU	FLU	PAL	RBR	CUI	GOI	SAN	CRU	CUI	RBR	PAL	FOR	INT	CAP	SPO	FLA	AMG	CTB	BAH	BOT	COR	CAM	GOI	VAS	FLU	AMG				
Palmeiras	CUI	VAS	COR	GOI	GRE	RBR	SAN	CAM	CTB	SPO	BAH	BOT	CAP	FLA	INT	FOR	AMG	FLU	CRU	CUI	VAS	COR	GOI	SPO	CAP	CAM	GRE	BAH	SAN	VAS	CTB	AMG	CRU	FLU	PAL	RBR	CUI	COR	BOT			
RB Bragantino	BAH	CUI	CRU	GRE	AMG	PAL	CAP	SAN	FLU	CAM	FLA	GOI	COR	SPO	BOT	INT	FOR	CTB	VAS	BAH	CUI	CRU	GRE	AMG	PAL	CAP	SAN	FLU	CAM	FLA	GOI	COR	SPO	BOT	INT	FOR	CTB	VAS				
Santos	GRE	CAM	AMG	CRU	BAH	VAS	PAL	RBR	INT	CTB	COR	FLA	CUI	GOI	SPO	BOT	FLU	CAP	FOR	GRE	CAM	AMG	CRU	BAH	VAS	PAL	RBR	INT	CTB	CRU	FLA	CUI	GOI	SPO	BOT	FLU	CAP	FOR				
São Paulo	BOT	AMG	CTB	INT	FOR	COR	VAS	GOI	GRE	PAL	CAP	CRU	FLU	RBR	SAN	CUI	BAH	CAM	FLA	BOT	AMG	CTB	INT	FOR	COR	VAS	PAL	RBR	SAN	CUI	BAH	CAM	FLA	FOR								
Vasco	CAM	PAL	BAH	FLU	CTB	SAN	SPO	FOR	FLA	INT	GOI	CUI	BOT	CRU	AMG	CAP	COR	GRE	RBR	CAM	PAL	BAH	FLU	CTB	SAN	SPO	FOR	FLA	INT	GOI	CUI	BOT	CRU	AMG	CAP	COR	GRE	RBR				

Fonte: CBF (2023). Elaborado pelo autor.

¹América/MG: AMG; Athletico/PR: CAP; Atlético/MG: CAM; Bahia: BAH; Botafogo: BOT; Corinthians: COR; Coritiba: CTB; Cruzeiro: CRU; Cuiabá: CUI; Flamengo: FLA; Fluminense: FLU; Fortaleza: FOR; Goiás: GOI; Grêmio: GRE; Internacional: INT; Palmeiras: PAL; RB Bragantino: RBR; Santos: SAN; São Paulo: SPO; Vasco: VAS.

Tabela 8: Ocorrência de *breaks* para o Campeonato Brasileiro de Futebol - Série A 2023.

Times	Rodadas																																						Breaks		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38			
América/MG	c	f	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	f	<u>f</u>	c	f	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	8	
Athletico/PR	c	f	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>c</u>	f	c	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	8	
Atlético/MG	c	f	c	f	<u>f</u>	c	f	c	f	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	c	<u>f</u>	c	f	c	<u>f</u>	c	f	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	8	
Bahia	f	c	f	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>f</u>	c	f	c	f	c	<u>c</u>	f	c	<u>f</u>	c	f	<u>f</u>	c	f	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	8	
Botafogo	c	f	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	f	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	8
Corinthians	c	f	<u>f</u>	c	f	c	f	c	f	c	<u>f</u>	c	f	c	f	c	f	c	f	c	<u>c</u>	f	c	<u>f</u>	c	f	f	c	f	c	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	f	4
Coritiba	f	c	<u>c</u>	f	c	f	c	<u>f</u>	c	<u>c</u>	f	<u>f</u>	c	<u>f</u>	c	f	c	f	c	f	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	f	c	8	
Cruzeiro	f	c	f	c	<u>c</u>	f	c	f	c	<u>f</u>	c	<u>c</u>	f	<u>f</u>	c	<u>c</u>	f	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	8	
Cuiabá	f	c	<u>c</u>	f	c	f	c	<u>c</u>	f	c	f	<u>f</u>	c	f	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	f	c	8
Flamengo	c	f	c	f	c	f	c	<u>c</u>	f	c	f	<u>f</u>	c	f	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	f	c	8
Fluminense	f	c	f	c	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	8
Fortaleza	c	f	c	f	c	f	<u>f</u>	c	<u>c</u>	f	<u>f</u>	c	f	c	<u>c</u>	f	c	f	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	8
Goiás	f	c	f	c	f	c	<u>f</u>	c	<u>c</u>	f	<u>f</u>	c	f	c	f	c	f	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8
Grêmio	c	f	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>c</u>	f	<u>c</u>	f	c	f	c	f	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8
Internacional	f	c	<u>c</u>	f	c	f	<u>f</u>	c	f	c	<u>f</u>	c	f	c	f	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8	
Palmeiras	c	f	c	f	c	<u>c</u>	f	<u>f</u>	c	f	<u>f</u>	c	f	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8	
RB Bragantino	c	f	c	f	c	f	c	<u>c</u>	f	<u>f</u>	c	<u>c</u>	f	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8
Santos	f	c	<u>c</u>	f	c	f	c	f	c	f	<u>c</u>	f	c	f	c	f	c	<u>f</u>	c	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	4
São Paulo	f	c	f	c	f	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	f	c	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8	
Vasco	f	c	<u>c</u>	f	<u>f</u>	c	<u>f</u>	c	f	c	<u>c</u>	f	c	f	c	f	c	f	c	f	<u>f</u>	c	<u>f</u>	c	<u>c</u>	f	c	<u>c</u>	f	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	<u>f</u>	c	8
Total																																							152		

Fonte: CBF (2023). Elaborado pelo autor.

Tabela 9: Valores de *carry over* para o Campeonato Brasileiro de Futebol - Série A 2023.

Times	Times																SCoQ			
	América/MG	Athletico/PR	Atlético/MG	Bahia	Botafogo	Corinthians	Coritiba	Cruzeiro	Cuiabá	Flamengo	Fluminense	Fortaleza	Goiás	Grêmio	Internacional	Palmeiras	RB Bragantino	Santos	São Paulo	Vasco
América/MG	0	8	6	0	0	2	4	6	4	0	2	0	0	0	0	2	0	0	2	2
Athletico/PR	0	0	4	0	4	6	0	2	2	0	0	6	2	2	0	2	2	2	2	0
Atlético/MG	2	0	0	4	0	2	4	2	4	8	2	0	2	4	0	2	2	0	0	0
Bahia	0	0	2	0	6	2	0	0	2	2	2	4	2	0	0	0	0	2	4	10
Botafogo	4	4	0	2	0	6	0	4	2	0	2	2	0	0	2	2	2	0	2	4
Corinthians	0	2	2	0	4	0	0	2	0	4	0	0	6	4	2	2	2	0	6	2
Coritiba	2	0	0	2	0	2	0	4	0	0	2	0	6	0	4	2	2	8	2	4
Cruzeiro	2	0	2	2	0	0	2	0	6	0	6	2	2	4	2	2	2	2	0	2
Cuiabá	0	2	0	2	4	2	0	2	0	2	2	0	2	0	2	6	4	2	2	2
Flamengo	2	4	4	0	4	0	2	0	2	0	2	2	6	0	6	2	0	0	2	0
Fluminense	8	2	4	0	2	2	2	4	0	0	0	2	0	2	0	2	2	2	2	2
Fortaleza	6	0	0	2	4	2	2	2	2	4	0	0	0	2	2	2	0	0	6	2
Goiás	2	2	4	2	0	4	0	0	2	0	6	0	0	6	2	2	0	2	2	2
Grêmio	8	0	2	4	2	0	0	0	2	2	0	2	2	0	2	2	10	0	0	0
Internacional	0	2	2	0	2	0	10	0	0	4	0	8	2	2	0	2	0	0	2	2
Palmeiras	0	4	0	4	0	2	0	2	0	2	4	6	0	2	0	0	8	2	0	2
RB Bragantino	0	0	4	0	2	2	2	2	2	0	2	0	4	4	2	2	0	10	0	0
Santos	0	4	4	2	0	0	2	2	6	2	4	2	0	0	4	2	0	0	2	2
São Paulo	2	2	0	8	4	2	2	0	2	4	0	2	0	2	6	0	0	2	0	0
Vasco	0	2	0	4	0	2	6	4	0	2	2	0	2	4	2	2	2	4	0	0
																Total				3152

Fonte: CBF (2023). Elaborado pelo autor.

Com relação às *breaks*, nota-se um fato interessante. Considerando os 20 times que disputam a edição, 18 têm o mesmo número de *breaks* (8), tendo apenas 2 times com uma quantidade distinta (4). A soma de *breaks* total é de 152.

Já com relação aos *carry over*, salienta-se, primeiramente, que é adotada uma noção de “circularidade” na tabela, tendo a primeira rodada do primeiro turno sendo considerada como seguinte à última do segundo turno. Dessa forma, é possível calcular os valores de *carry over* em todas as rodadas. Nota-se que o maior número de *carry over* dado por um time i para um time j é 10 que, por sua vez, se repete 4 vezes entre os times Bahia e Vasco, Grêmio e RB Bragantino, Internacional e Coritiba e RB Bragantino e Santos. É interessante notar como o RB Bragantino é um dos times que mais recebe *carry over* de um determinado time i , mas também é um dos que mais dá *carry over* para um determinado time j . Esses valores de 10 podem ser considerados altos, levando em conta um campeonato de 38 rodadas. Como última observação, pode-se destacar que todos os valores presentes na tabela são valores pares, o que já era esperado tratando-se de um campeonato de dois turnos com simetria espelhada. Os efeitos de *carry over* são iguais a 3152.

3.3 Análise da Tabela

Nesta seção, é realizada uma análise aprofundada da tabela do Campeonato Brasileiro de Futebol Série A - 2023. O objetivo principal é identificar padrões presentes na tabela. Vale ressaltar que o processo de geração da tabela não é divulgado pela Confederação Brasileira de Futebol (CBF), a entidade responsável pela organização da competição. Portanto, não é possível afirmar com exatidão quais são os critérios e restrições básicas adotados no momento da criação da tabela.

A ausência de informações detalhadas sobre o processo de elaboração da tabela implica que a análise será conduzida com base nos resultados e na disposição das partidas na tabela, sem acesso direto aos critérios específicos estabelecidos pela CBF. Esta abordagem exploratória busca identificar regularidades ou peculiaridades que possam fornecer *insights* sobre a lógica subjacente à distribuição das partidas ao longo da competição.

A investigação se concentrará em elementos como a distribuição de jogos em diferentes rodadas, confrontos diretos entre equipes de destaque, sequências de jogos em casa e fora, entre outros fatores que podem influenciar a dinâmica e a competitividade do campeonato.

A falta de divulgação explícita dos critérios de elaboração da tabela adiciona um

elemento de incerteza a essa análise, tornando-a uma exploração desafiadora e intrigante. A descoberta de padrões ou regularidades na tabela pode fornecer *insights* valiosos sobre a estratégia subjacente à programação dos jogos.

Durante a análise da tabela do Campeonato Brasileiro de Futebol Série A - 2023, pode-se perceber, entre outras coisas, que não existem sequências de 3 ou mais partidas em casa ou fora para nenhum dos times que disputam a competição. Essa distribuição sugere uma preocupação em evitar vantagens ou desvantagens excessivas em relação ao mando de campo. Chama a atenção, também, o fato de que nas duas primeiras rodadas nenhum time disputou as duas partidas com ou sem o mando de campo, ou seja, todos aqueles times que disputaram a primeira rodada em casa, viajaram para jogar a segunda rodada fora de casa e vice-versa. Além disso, todos os times que iniciaram o campeonato em casa, terminaram o mesmo jogando longe de seu estádio e vice-versa. Nota-se que, para cada um dos turnos, o número de partidas em casa disputadas por cada um dos times era no mínimo 9 e no máximo 10. Isso pode indicar uma preocupação em garantir condições parecidas para todos os times.

Pode-se, também, realizar diversas análises em cima dos “clássicos”. Um clássico é por definição um confronto entre equipes com uma notável rivalidade desportiva. Neste Trabalho de Formatura, apenas os clássicos regionais (clássicos, segundo a primeira definição, envolvendo times da mesma região) são considerados clássicos, ou seja, os clássicos inter-regionais não são classificados da mesma forma. Primeiramente, destaca-se que não existem clássicos nem na primeira, nem na última rodada. Nas edições de 2011 e 2012, era determinado que a última rodada de cada turno seria “dedicada” aos clássicos, visando evitar que os times chegassem desinteressados à última rodada. Além disso, constatou-se que em nenhuma das 38 rodadas existe mais de um clássico envolvendo times do mesmo estado em uma mesma rodada, e que sempre existiam rodadas entre clássicos para um mesmo time, ou seja, nenhum time disputa clássicos consecutivos. Tais padrões sugerem, respectivamente, uma preocupação com interesses comerciais, visto que pode ser prejudicial dividir o interesse do público em uma mesma rodada e evitar sequências que podem interferir de maneira forte no psicológico dos times, visto que clássicos são partidas com grande potencial de elevar ou diminuir a moral dos times ao longo do campeonato e, eventualmente, ter duas derrotas consecutivas em clássicos poderia ter um impacto muito negativo sobre os times.

Esses padrões são apenas alguns dos muitos existentes na distribuição dos jogos, sugerindo uma preocupação não apenas com a competitividade, mas também com a experiência global do campeonato, buscando manter um equilíbrio dinâmico e atrativo ao

longo de toda a temporada. Destaca-se que, para esse trabalho não serão incluídos todos esses padrões, pois alguns deles não influenciam diretamente nos objetivos principais da modelagem que será proposta, além do fato de que, foi analisada, em detalhe, apenas a tabela do ano de 2023 e, somente a partir dela, é impossível concluir se esses padrões são recorrentes em todos os anos e, mais do que isso, se serão mantidos para as tabelas seguintes.

3.4 Considerações Sobre o Problema

Considerando os elementos mencionados neste capítulo, a definição precisa das restrições a serem adotadas torna-se crucial para se construir uma tabela que respeite e contemple as necessidades atuais e seja coerente com os objetivos de justiça competitiva apresentados neste Trabalho de Formatura. Inicialmente, é essencial ressaltar a manutenção de certos aspectos como formato e simetria, elementos que identificam a instância selecionada. Evitar mudanças substanciais nesses parâmetros é essencial, uma vez que a proposição de alterações drásticas poderia ser percebida como disruptiva, potencialmente gerando uma recepção desfavorável por parte de torcedores, patrocinadores, equipes e jogadores já familiarizados com o modelo atual do Campeonato Brasileiro de Futebol - Série A. Dessa forma, o formato *Mirrored Double Round-Robin* será mantido.

Os clássicos receberão atenção especial na tabela proposta, buscando seguir os padrões vigentes e identificados na seção anterior. Será estabelecido que clássicos de um mesmo estado não ocorrerão em uma mesma rodada. Além disso, será proibido que um time participe de clássicos em rodadas consecutivas. Quanto ao momento em que esses clássicos serão alocados, não haverá imposições, permitindo sua ocorrência tanto no início quanto no encerramento do campeonato, uma vez que a relação entre a rodada específica em que serão disputados e os aspectos de justiça competitiva não são claros.

Também será proposta uma maneira de evitar sequências de jogos potencialmente mais fáceis ou difíceis para os times envolvidos, de modo a assegurar uma distribuição mais equilibrada dos desafios ao longo do campeonato o que, por sua vez, também pode ser entendido como uma condição para garantir a justiça competitiva entre os times.

Somado a isso, enfatiza-se mais uma vez que, neste Trabalho de Formatura, garantir uma tabela justa competitivamente está prioritariamente relacionado com o número de *breaks* e de efeitos de *carry over* e que ambos os critérios serão os objetos de minimização deste Trabalho.

Por fim, salienta-se que, os outros padrões percebidos durante a análise da tabela do Campeonato Brasileiro de Futebol - Série A 2023, assim como outros que, eventualmente, existam e não foram citados, não serão considerados neste estudo.

4 MODELAGEM MATEMÁTICA

Neste capítulo é apresentado o modelo matemático elaborado para detalhamento do Problema de Geração de Tabelas de Campeonatos aplicado para o Campeonato Brasileiro de Futebol - Série A. O modelo proposto é original e descreve em detalhes as principais restrições consideradas relevantes para garantir o equilíbrio entre as equipes e restrições que puderam ser percebidas como relevantes para a Confederação Brasileira de Futebol (CBF), a partir do estudo das tabelas dos campeonatos já disputados.

4.1 Definição de Elementos, Conjuntos e Variáveis do Modelo Matemático

Nesta seção é apresentada a nomenclatura utilizada no modelo de Programação Inteira elaborado para a solução do Problema de Geração de Tabelas de Campeonatos aplicado para o Campeonato Brasileiro de Futebol - Série A. Os índices e conjuntos utilizados no modelo para a geração da tabela são detalhados na Tabela 10.

Tabela 10: Índices e conjuntos da modelagem.

Elementos	Descrição
$i, j, l \in T$	Em que i, j e l são times e T é o conjunto de todos os times que disputam o campeonato
$k \in K$	Em que k é o número da rodada em questão e K é o conjunto de todas as rodadas
$c \in C$	Em que c é a cidade na qual o time está localizado e C é o conjunto de cidades nas quais dois ou mais times estão localizados (no caso do Campeonato Brasileiro de Futebol - Série A 2023, temos as cidades de São Paulo (1), Rio de Janeiro (2), Belo Horizonte (3), Curitiba (4) e Porto Alegre (5))
$K^1 \subset K$	Em que K^1 é o conjunto de rodadas do primeiro turno do campeonato
$T_c \subset T$	Em que T_c é o conjunto de times que estão localizados na cidade c que disputam clássicos regionais
$TOP10 \subset T$	Em que $TOP10$ é o conjunto de times que ficaram nas dez primeiras posições do campeonato do ano anterior (no caso de 2022, Palmeiras, Internacional, Fluminense, Corinthians, Flamengo, Athletico/PR, Atlético/MG, Fortaleza, São Paulo, América/MG)
$BOTTOM10 \subset T$	Em que $BOTTOM10$ é o conjunto de times que ficaram nas posições 11 a 16 do campeonato do ano anterior mais os quatro times que foram promovidos da segunda divisão no ano anterior (no caso de 2022, Botafogo, Santos, Goiás, RB Bragantino, Coritiba, Cuiabá, Cruzeiro, Grêmio, Bahia, Vasco)

Fonte: Elaborado pelo autor.

A Tabela 11 apresenta os parâmetros definidos no modelo.

Tabela 11: Parâmetros da modelagem.

Elementos	Descrição
$n \in \mathbb{Z}_+$	Número de times envolvidos na competição
$brk^l \in \mathbb{Z}_+$	Número mínimo possível de <i>breaks</i>
$brk^u \in \mathbb{Z}_+$	Número máximo possível de <i>breaks</i>
$coe^l \in \mathbb{Z}_+$	Número mínimo possível de efeitos de <i>carry over</i>
$coe^u \in \mathbb{Z}_+$	Número máximo possível de efeitos de <i>carry over</i>
$\omega_1 \in \mathbb{R}_+$	Penalidade associada ao objetivo de minimização do número de <i>breaks</i>
$\omega_2 \in \mathbb{R}_+$	Penalidade associada ao objetivo de minimização de efeitos de <i>carry over</i>

Fonte: Elaborado pelo autor.

As variáveis de decisão cujos valores serão resultado da solução do problema abordado estão detalhadas na Tabela 12.

Tabela 12: Variáveis de decisão da modelagem.

Elementos	Descrição
$x_{i,j,k} \in \{0, 1\}$	Assume valor unitário se o time i joga em casa contra o time j na rodada k , e valor nulo caso contrário
$c_{i,j,k} \in \{0, 1\}$	Assume valor unitário se o time i dá um <i>carry over</i> para o time j na rodada k , e valor nulo caso contrário
$b_{i,k} \in \{0, 1\}$	Assume valor unitário se o time i tem uma break na rodada k , e valor nulo caso contrário

Fonte: Elaborado pelo autor.

4.2 Função Objetivo e Restrições do Modelo Matemático

Nesta seção é apresentado o modelo matemático elaborado para o Problema de Geração de Tabelas de Campeonatos para o Campeonato Brasileiro de Futebol - Série A. É proposto um modelo de programação quadrática inteira mista (MIQP) para o problema de minimização integrada do número de *breaks* e dos efeitos de *carry over*. Utilizando variáveis binárias, o modelo possui um termo quadrático apenas na função objetivo, mas não há termos não lineares nas restrições. O número resultante de *breaks* e dos efeitos *carry over* são dados, respectivamente, por *breaks* e *coe*. É utilizado o conceito de

soma ponderada que, por sua vez, é uma das abordagens mais populares para avaliar um problema multiobjetivo. A ideia é associar a cada objetivo um peso e minimizar a soma ponderada dos objetivos, para que, dessa forma, o problema multiobjetivo possa ser transformado em um problema de objetivo único. No entanto, para o problema em questão, os valores obtidos de *breaks* e *coe*, bem como os limites associados a cada um deles, possuem escalas diferentes, logo, precisam passar por uma normalização. A normalização adotada é adaptada do trabalho apresentado em Miettinen (1999). Em um problema de otimização multiobjetivo, a função objetivo para cada objetivo pode ser normalizada através de:

$$f'_i(x) = \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}}$$

onde $f_i(x)$ representa o valor da função objetivo para o objetivo correspondente e f_i^{max} e f_i^{min} representam os valores máximo e mínimo, respectivamente, que aquele objetivo pode assumir. O intervalo da nova função objetivo $f'_i(x)$ é $[0,1]$, ou seja, para o problema abordado com dois objetivos, os valores possíveis que a função objetivo pode assumir estarão limitados ao intervalo $[0,2]$. Ressalta-se que essa normalização adotada só é possível pois ambos os objetivos são limitados.

A seguir, tem-se o modelo completo posposto para o problema. Inicialmente, é apresentada a função do modelo. Em seguida, serão expostas e detalhadas as expressões das restrições.

Função objetivo:

$$Min \omega_1 \cdot \frac{(\sum_{i \in T} \sum_{k \in K} b_{i,k}) - brk^l}{brk^u - brk^l} + \omega_2 \cdot \frac{(\sum_{i,j \in T} (\sum_{k \in K} c_{i,j,k})^2) - coe^l}{coe^u - coe^l} \quad (4.1)$$

Restrições:

$$\sum_{k \in K^1} (x_{i,j,k} + x_{j,i,k}) = 1, \quad \forall i, j \in T, i \neq j \quad (4.2)$$

$$\sum_{k \in K} x_{i,j,k} = 1, \quad \forall i, j \in T, i \neq j \quad (4.3)$$

$$\sum_{j \in T \mid j \neq i} (x_{i,j,k} + x_{j,i,k}) = 1, \quad \forall i \in T, \forall k \in K \quad (4.4)$$

$$x_{i,j,k} = x_{j,i,k+n-1}, \quad \forall i, j \in T, i \neq j, \forall k \in K^1 \quad (4.5)$$

$$x_{i,l,k} + x_{j,l,k+1} + x_{l,i,k} + x_{l,j,k+1} - 1 \leq c_{i,j,k}, \quad \forall i, j, l \in T, i \neq j \neq l, \forall k \in K \setminus \{2 \cdot n - 2\} \quad (4.6)$$

$$x_{i,l,2 \cdot n - 2} + x_{j,l,1} + x_{l,i,2 \cdot n - 2} + x_{l,j,1} - 1 \leq c_{i,j,2 \cdot n - 2}, \quad \forall i, j, l \in T, i \neq j \neq l \quad (4.7)$$

$$x_{i,j,k} + x_{i,l,k+1} - 1 \leq b_{i,k+1}, \quad \forall i, j, l \in T, i \neq j \neq l, \forall k \in K \setminus \{2 \cdot n - 2\} \quad (4.8)$$

$$1 - \sum_{j \in T \mid j \neq i} x_{i,j,k} - \sum_{j \in T \mid j \neq i} x_{i,j,k+1} \leq b_{i,k+1}, \quad \forall i \in T, \forall k \in K \setminus \{2 \cdot n - 2\} \quad (4.9)$$

$$\sum_{j \in T_c \mid j \neq i} (x_{i,j,k} + x_{j,i,k} + x_{i,j,k+1} + x_{j,i,k+1}) \leq 1, \quad \forall c \in C : |T_c| \geq 3, \forall i \in T_c, \forall k \in K^1 \setminus \{n - 1\} \quad (4.10)$$

$$\sum_{i \in T_c} \sum_{j \in T_c \mid j \neq i} x_{i,j,k} \leq 1, \quad \forall c \in C : |T_c| = 4, \forall k \in K^1 \quad (4.11)$$

$$\begin{aligned} & \sum_{j \in T \mid j \notin TOP10, j \neq i} (x_{i,j,k} + x_{i,j,f(k+1)} + x_{i,j,f(k+2)} + x_{i,j,f(k+3)} + x_{i,j,f(k+4)} + x_{i,j,f(k+5)} + x_{j,i,k} \\ & + x_{j,i,f(k+1)} + x_{j,i,f(k+2)} + x_{j,i,f(k+3)} + x_{j,i,f(k+4)} + x_{j,i,f(k+5)}) \geq 1, \quad \forall i \in T, \forall k \in K^1 \end{aligned} \quad (4.12)$$

$$\begin{aligned} & \sum_{j \in T \mid j \notin BOTTOM10, j \neq i} (x_{i,j,k} + x_{i,j,f(k+1)} + x_{i,j,f(k+2)} + x_{i,j,f(k+3)} + x_{i,j,f(k+4)} + x_{i,j,f(k+5)} + x_{j,i,k} \\ & + x_{j,i,f(k+1)} + x_{j,i,f(k+2)} + x_{j,i,f(k+3)} + x_{j,i,f(k+4)} + x_{j,i,f(k+5)}) \geq 1, \quad \forall i \in T, \forall k \in K^1 \end{aligned} \quad (4.13)$$

$$x_{i,j,k}, c_{i,j,k}, b_{i,k} \in \{0, 1\} \quad (4.14)$$

A função objetivo (4.1) do modelo elaborado é composta pela soma de dois termos detalhados a seguir. Cada um dos termos é normalizado da maneira descrita anteriormente. Estes são, ainda, multiplicados por fatores ω entre zero e um que representam a relevância relativa de cada objetivo.

- I A minimização da soma de variáveis $b_{i,k}$ do primeiro termo, que indicam se o time i teve uma *break* na rodada k , ou seja se ele disputou as rodadas $k - 1$ e k em casa ou na casa do adversário, tem como objetivo manter, sempre que possível, partidas alternadas em casa e fora para todos os times ao longo do campeonato.
- II A minimização de $(\sum_{k \in K} c_{i,j,k})^2$ no segundo termo, associada a soma dos efeitos de *carry over* dados pelos times i aos times j ao longo do campeonato elevados ao quadrado, tem como objetivo minimizar os efeitos de *carry over* dentro da competição.

A seguir serão detalhadas as restrições definidas para o modelo em questão.

- A expressão (4.2) garante que, no primeiro turno do campeonato, todos os times só se enfrentarão uma única vez, ou com mando de i ou com mando de j .
- A expressão (4.3) garante que, ao longo do campeonato, os times só terão o mando de jogo contra os outros times uma única vez.
- A expressão (4.4) garante que os times só jogarão uma única vez por rodada, garantindo que o campeonato seja compacto.
- A expressão (4.5) garante que dois times que se enfrentaram na rodada k , se enfrentarão necessariamente na rodada $k + n$, porém com mando de campo invertido, garantindo, assim, a simetria espelhada do campeonato.
- A expressão (4.6) define a variável $c_{i,j,k}$, com base nas variáveis $x_{i,l,k}$, $x_{j,l,+1}$, $x_{l,i,k}$ e $x_{l,j,k+1}$. Essa restrição vale para todas as rodadas com exceção da última, pois, não existe uma rodada seguinte. Se um time i joga com um time l na rodada k , e um time j joga contra o mesmo time l na rodada $k + 1$, então pode-se dizer que o time i deu um *carry over* para o time j na rodada k .
- A expressão (4.7) funciona da mesma maneira que a anterior, considerando que a rodada seguinte à última, seria a primeira, gerando uma noção de circularidade da tabela. Se um time i joga com um time l na última rodada do segundo turno e um time j joga contra o mesmo time l na primeira rodada do primeiro turno, então pode-se dizer que o time i deu um *carry over* para o time j na última rodada.

- A expressão (4.8) define a variável $b_{i,k+1}$, considerando uma sequência de jogos em casa, com base no valor das variáveis $x_{i,j,k}$ e $x_{i,j,k+1}$. Se um time jogou em casa duas rodadas consecutivas, então pode-se dizer que ele teve uma *break* em casa.
- A expressão (4.9) também define a variável $b_{i,k+1}$, mas, agora, considerando uma sequência de jogos fora de casa, com base no valor das somas das variáveis $x_{i,j,k}$ e $x_{i,j,k+1}$. Se um time jogou fora de casa em rodadas consecutivas, então pode-se dizer que ele teve uma *break* fora de casa.
- A expressão (4.10) garante que, para qualquer cidade na qual estão localizados mais de 2 times, haverá sempre uma partida espaçando jogos entre o time i e os times j e l , sendo os 3 times da mesma cidade. Em outras palavras, nenhum time disputará clássicos em rodadas consecutivas. Geralmente, clássicos são jogos mais difíceis para os quais as equipes solicitam mais tempo de preparação. Além de que, perder dois jogos clássicos seguidos afetaria fortemente a motivação de uma equipe e o interesse de seus torcedores.
- A expressão (4.11) garante que, em uma dada rodada, não haverá mais de um clássico envolvendo times de uma mesma cidade para as cidades com exatamente 4 times (São Paulo e Rio de Janeiro no caso do campeonato do ano de 2023). Ter mais de um clássico na mesma rodada poderia dividir o interesse do público, além de potencializar confrontos entre torcedores antes e/ou depois das partidas e gerar problemas de circulação.
- A expressão (4.12) impõe que nenhum time pode jogar mais de cinco jogos consecutivos contra equipes que terminaram o campeonato do ano anterior nas 10 primeiras posições, como uma forma de evitar uma longa sequência de jogos, potencialmente, mais difíceis.
- A expressão (4.13) impõe que nenhum time pode jogar mais de cinco jogos consecutivos contra equipes que terminaram o campeonato do ano anterior nas posições 11 - 16 ou os 4 times que foram promovidos para a Série A no ano atual, como uma forma de evitar uma longa sequência de jogos, potencialmente, menos difíceis.

As expressões (4.12) e (4.13) utilizam a função $f(a)$, definida, conforme (RIBEIRO; URRUTIA, 2012), por:

$$f(a) = \begin{cases} a, & \text{se } a \leq n - 1 \\ a - (n - 1), & \text{caso contrário} \end{cases}$$

e usada para lidar com sequências de jogos consecutivos que começam na primeira fase e terminam na segunda.

- A expressão (4.14) estabelece o domínio das variáveis de decisão do modelo.

Nota-se que este modelo elaborado pode ser eventualmente estendido para incorporar outras características específicas do Campeonato Brasileiro de Futebol - Série A e, mais do que isso, pode ser facilmente adaptado para outros campeonatos que tenham estrutura semelhante. Assim, o modelo configura-se como uma ferramenta de apoio à decisão de organizações e confederações que cuidem dos esportes em suas respectivas localidades.

5 ABORDAGEM DE SOLUÇÃO

Neste capítulo, é apresentada a abordagem de solução escolhida para resolver o problema, a qual se inspira e adapta de forma significativa o trabalho de Ribeiro e Urrutia (2007a).

5.1 Rápida Heurística Construtiva

Nesta seção é apresentada a heurística desenvolvida para gerar uma solução factível para o problema abordado. A mesma é composta de 3 etapas principais: criação de uma tabela que respeite as condições básicas de um campeonato do tipo *Mirrored Double Round-Robin* com times fictícios, definição do mando das partidas definidas na etapa anterior e alocação dos times reais à tabela gerada, respeitando as restrições específicas da instância.

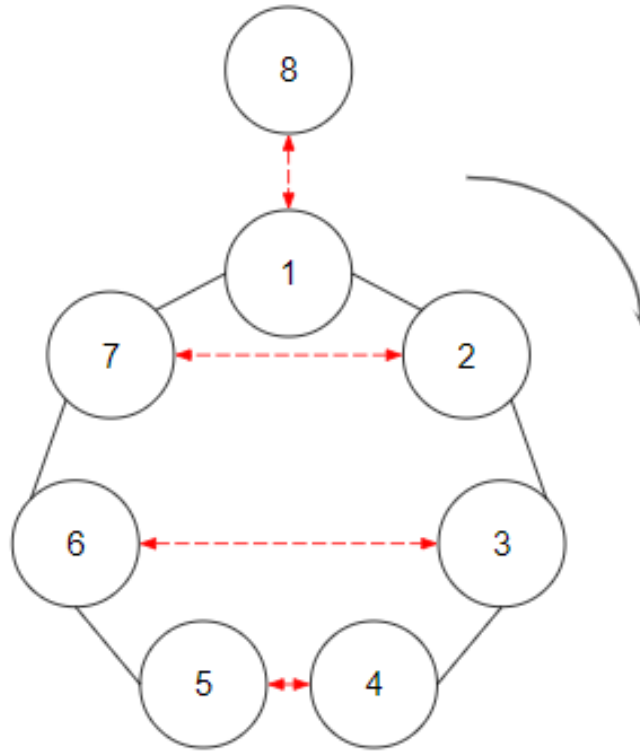
5.1.1 Criação da Tabela com Times Fictícios

Para a construção de uma tabela para um campeonato do tipo *SRR* com n times, sem a necessidade de designar locais para os jogos, pode ser aplicada a “abordagem do polígono”. A adaptação da tabela gerada para um campeonato do tipo *Mirrored Double Round-Robin* (*MDRR*) pode ser feita de maneira fácil, apenas “repetindo” o resultado alcançado por essa abordagem. O espelhamento, no entanto, só é possível uma vez que as partidas para a tabela *SRR* já tiverem seus respectivos mandos definidos.

Na Figura 3, é ilustrada a aplicação dessa abordagem para $n = 8$. Os times fictícios A, B, ..., S (para $n = 20$) são inicialmente colocados em nós numerados consecutivamente no sentido horário de um polígono regular com $n - 1$ nós: a equipe A no nó 1, a equipe B no nó 2 e assim por diante. O time fictício T não é colocado no polígono. Em cada rodada $k = 1, \dots, n - 1$, o time fictício colocado no nó numerado $v = 2, \dots, n/2$ joga contra aquele colocado no nó numerado $n + 1 - v$. Em cada rodada, o time fictício colocado no nó 1

joga contra a equipe fictícia T. Após cada rodada, cada time fictício A, B, ..., S é movido no sentido horário para o próximo nó do polígono (RIBEIRO; URRUTIA, 2007a).

Figura 3: Polígono formado para $n = 8$ times (primeira rodada).



Fonte: Elaborado pelo autor.

5.1.2 Definição dos Mandos de Campo

Nessa etapa, são definidos os mandos de campo para cada uma das partidas criadas pela etapa anterior, ainda levando em conta os times fictícios. O procedimento se dá da seguinte maneira:

- Para as rodadas 1 e $n - 1$, o mando de campo é definido de maneira aleatória.
- Para as rodadas 2, ..., $n - 2$, a estratégia a seguir é adotada. É definida a variável cm_i que representa o número de partidas em que o time fictício i jogou de forma consecutiva ou em casa ou fora e tem seu valor atualizado a cada rodada. Por exemplo, se a variável $cm_D = 3$ ao final da sétima rodada, significa que o time D disputou a quinta, sexta e sétima rodadas todas em seus domínios ou todas fora do mesmo. Nota-se que o valor mínimo que a variável cm_i pode assumir é 1. A partir disso, têm-se as seguintes regras:

- Situação 1: $cm_j > cm_i$
 - * se o time j disputou sua última partida em casa, então o jogo é alocado no estádio de i .
 - * caso contrário, o jogo é alocado no estádio de j .
- Situação 2: $cm_j < cm_i$
 - * se o time i disputou sua última partida em casa, então o jogo é alocado no estádio de j .
 - * caso contrário, o jogo é alocado no estádio de i .
- Situação 3: $cm_j = cm_i$
 - * se o time i disputou sua última partida em casa e o time j disputou sua última partida fora de casa, então o jogo é alocado no estádio de j .
 - * se o time j disputou sua última partida em casa e o time i disputou sua última partida fora de casa, então o jogo é alocado no estádio de i .
 - * caso contrário, o mando de campo é definido de maneira aleatória.

5.1.3 Alocação dos Times Reais

Essa é a etapa mais complexa da construção da solução inicial. Os times reais não podem ser alocados de qualquer forma aleatória como ordená-los em ordem alfabética e alocá-los 1 a 1 aos times fictícios A, B, ..., T, pois isso dificilmente cumpriria com as restrições do modelo, como, por exemplo, a que impõem que os times não podem jogar clássicos estaduais de forma consecutiva. Por esse motivo, é definida a matriz quadrada $n \times n$ que ajuda a entender o padrão gerado pela primeira etapa e permite encontrar uma saída para esse problema. Cada elemento (v_1, v_2) da matriz representa o número de vezes em que os times alocados nos nós v_1 e v_2 foram adversários consecutivos de outros times. Nesse caso, não é levada em consideração a ordem das partidas, ou seja, se um time jogou contra o time i primeiro e depois contra o time j ou se o mesmo jogou contra o time j primeiro e depois contra o time i . Ambos os cenários possuem o mesmo efeito nesse caso e, até mesmo por isso, tem-se uma matriz simétrica em relação a diagonal primária. A Figura 4 ilustra a matriz para um campeonato com 20 times.

Figura 4: Matriz de adversários consecutivos para $n = 20$.

0	1	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	4
1	0	2	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	4
34	2	0	2	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	0	4
0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	0	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	0	4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	0	2	33	4
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	0	2
2	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	2	0
4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	0

Fonte: Elaborado pelo autor.

A matriz gerada não depende dos times envolvidos e nem do campeonato em questão. Para qualquer campeonato envolvendo 20 times que atenda o formato em questão, a mesma matriz é obtida. A partir de sua da matriz, percebe-se que a tabela gerada pela primeira etapa cria vários padrões, tendo muitos pares de times que são adversários consecutivos de outros times. Os times alocados nos nós 1 e 3, por exemplo, são adversários consecutivos para outros times 34 vezes no campeonato. Para o objetivo desse trabalho, essa solução inicial, embora encontre um bom número de *breaks* através da segunda etapa, gera também uma solução em que os efeitos de *carry over* são máximos.

Tendo em vista que o modelo procura por uma solução em que os times não disputem clássicos estaduais de maneira consecutiva, uma solução é alocar os times de uma mesma cidade que estão envolvidos em clássicos em posições em que nunca serão adversários consecutivos de nenhum outro time. Para o caso da cidade de São Paulo, por exemplo, em que 4 times estão envolvidos em clássicos, todos eles devem ser posicionados em nós v_1, v_2, v_3 e v_4 , tal que todas as combinações duas a duas de nós leve a um valor 0 na matriz de adversários consecutivos.

Dessa forma, os times reais são alocados aos times fictícios de acordo com a quantidade de times presentes nas suas cidades, começando por aquelas cidades com maior número de times (consequentemente, maior número de clássicos). São muitas as possíveis escolhas de nós que satisfaçam a restrição que impede clássicos consecutivos para um mesmo time. Entretanto, nem todas elas garantem outras restrições do modelo, como a que impede

que clássicos de um mesmo estado sejam jogados na mesma rodada (restrição (4.11) do modelo) ou as que impedem sequências, potencialmente, mais difíceis e fáceis (restrições (4.12) e (4.13) do modelo).

Por tal razão, a cada cidade que terá seus times alocados, todos os conjuntos de vértices que satisfazem a restrição (4.10) que impede que um time dispute clássicos seguidos, são guardados e um deles é escolhido de maneira aleatória, visando atender todas as restrições do modelo.

Essa terceira etapa pode ser executada mais de uma vez, testando diferentes alocações, até que uma alocação não infrinja nenhuma restrição do modelo. É importante destacar que, ao final dessa etapa, a solução obtida é uma solução factível inicial para o problema em questão.

5.2 Vizinhanças

Nessa seção são apresentadas 5 diferentes vizinhanças que podem ser exploradas com o objetivo de melhorar a solução existente.

A fim de exibir o potencial de melhoria causado por cada uma das vizinhanças, tem-se a Tabela 13, contendo as partidas de um campeonato com 8 times fictícios.

Tabela 13: Tabela fictícia com 8 times.

Rodadas													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A x H	B x E	B x C	C x G	B x F	A x C	B x A	H x A	E x B	C x B	G x C	B x F	C x A	A x B
B x G	C x D	D x H	E x A	D x A	D x B	C x H	G x B	D x C	H x D	A x E	A x D	B x D	H x C
C x F	G x A	E x G	F x D	E x C	E x H	E x F	F x C	A x G	G x E	D x F	C x E	H x E	F x E
D x E	H x F	F x A	H x B	G x H	F x G	G x D	E x D	F x H	A x F	B x H	H x G	G x F	D x G

Fonte: Elaborado pelo autor.

Neste exemplo, os valores encontrados de *breaks* e de efeitos de *carry over* (valores que deseja-se minimizar) são, respectivamente, de 40 e 784.

5.2.1 Alternância de Mando de uma Partida

Cada solução dessa vizinhança é obtida invertendo o mando de campo de uma partida específica (e da sua correspondente no segundo turno). Cada solução possui $n \cdot (n - 1)/2$

vizinhos. A Tabela 14 mostra um vizinho da solução inicial, exibida na Tabela 13, onde a partida envolvendo os times A e E na rodada 4 teve seu mando invertido.

Tabela 14: Solução vizinha com o mando invertido para o jogo entre os times A e E.

Rodadas													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A x H	B x E	B x C	C x G	B x F	A x C	B x A	H x A	E x B	C x B	G x C	B x F	C x A	A x B
B x G	C x D	D x H	A x E	D x A	D x B	C x H	G x B	D x C	H x D	E x A	A x D	B x D	H x C
C x F	G x A	E x G	F x D	E x C	E x H	E x F	F x C	A x G	G x E	D x F	C x E	H x E	F x E
D x E	H x F	F x A	H x B	G x H	F x G	G x D	E x D	F x H	A x F	B x H	H x G	G x F	D x G

Fonte: Elaborado pelo autor.

A solução vizinha gerada apresenta um número de *breaks* igual a 32 que, por sua vez, é inferior ao da primeira solução (40). Os efeitos de *carry over*, por sua vez, permanecem inalterados, uma vez que essa vizinhança não explora novas sequências de adversários.

5.2.2 Alternância dos Mandos de Campo para um Time

Cada solução dessa vizinhança é obtida invertendo o mando de campo de todas as partidas do campeonato para um time específico. Cada solução possui n vizinhos. A Tabela 15 mostra um vizinho da solução inicial, exibida na Tabela 13, onde todas as partidas do time E tiveram seus mandos invertidos.

Tabela 15: Solução vizinha invertendo todos os mandos dos jogos envolvendo o time E.

Rodadas													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A x H	E x B	B x C	C x G	B x F	A x C	B x A	H x A	B x E	C x B	G x C	B x F	C x A	A x B
B x G	C x D	D x H	A x E	D x A	D x B	C x H	G x B	D x C	H x D	E x A	A x D	B x D	H x C
C x F	G x A	G x E	F x D	C x E	H x E	F x E	F x C	A x G	E x G	D x F	E x C	E x H	E x F
E x D	H x F	F x A	H x B	G x H	F x G	G x D	D x E	F x H	A x F	B x H	H x G	G x F	D x G

Fonte: Elaborado pelo autor.

A solução vizinha gerada apresenta um número de *breaks* igual a 32 que, por sua vez, é inferior ao da primeira solução (40). Os efeitos de *carry over*, por sua vez, permanecem inalterados, uma vez que essa vizinhança não explora novas sequências de adversários.

5.2.3 Alternância de Rodadas

Cada solução dessa vizinhança é obtida invertendo duas rodadas do primeiro turno (consequentemente, as respectivas rodadas no segundo turno também são invertidas). Cada solução possui $(n - 1) \cdot (n - 2)$ vizinhos. A Tabela 16 mostra um vizinho da solução inicial, exibida na Tabela 13, onde as rodadas 2 e 3 foram invertidas.

Tabela 16: Solução vizinha com as rodadas 2 e 3 invertidas.

Rodadas													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A x H	B x C	B x E	C x G	B x F	A x C	B x A	H x A	C x B	E x B	G x C	B x F	C x A	A x B
B x G	D x H	C x D	E x A	D x A	D x B	C x H	G x B	H x D	D x C	A x E	A x D	B x D	H x C
C x F	E x G	G x A	F x D	E x C	E x H	E x F	F x C	G x E	A x G	D x F	C x E	H x E	F x E
D x E	F x A	H x F	H x B	G x H	F x G	G x D	E x D	A x F	F x H	B x H	H x G	G x F	D x G

Fonte: Elaborado pelo autor.

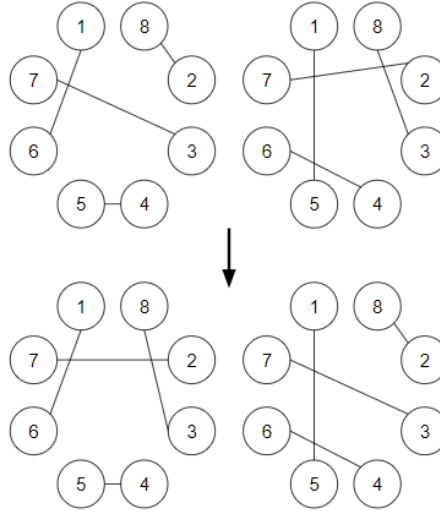
A solução vizinha gerada apresenta um número de *breaks* igual ao da primeira solução (40), o que não é uma regra, mas apresenta uma redução nos efeitos de *carry over*, caindo de 784, na solução inicial, para 512.

5.2.4 Alternância Parcial de Rodadas

Para gerar uma solução nessa vizinhança é necessário que a solução inicial tenha, pelo menos, duas rodadas $k1$ e $k2$ em que quatro times $t1, t2, t3$ e $t4$ se enfrentem entre si. Por exemplo, deve existir uma rodada $k1$ em que existam as partidas $t1$ x $t3$ e $t2$ x $t4$ e uma rodada $k2$ em que existam as partidas $t1$ x $t4$ e $t2$ x $t3$. A solução vizinha envolve a troca das rodadas em que esses jogos acontecem. Os jogos $t1$ x $t3$ e $t2$ x $t4$ são movidos para a rodada $k2$, enquanto os jogos $t1$ x $t4$ e $t2$ x $t3$ são movidos para a rodada $k1$. Qualquer uma das 6 possibilidades (3 adversários possíveis para $t1$ em $k1$ vezes 2 adversários possíveis para $t1$ em $k2$, assumindo que os times desse quarteto que não enfrentam $t1$ em $k1$ jogam entre si, assim como os times do quarteto que não enfrentam $t1$ em $k2$) podem gerar soluções vizinhas. Em resumo, essa vizinhança explora a alternância de apenas algumas partidas em duas rodadas distintas. O número de 4 times definido é arbitrário e pode ser alterado para qualquer valor par (caso, esse valor seja 20, essa vizinhança passa a ser a mesma que a anterior). A Figura 5 ilustra o procedimento de inverter as rodadas em que os times 2, 3, 7 e 8 se enfrentam. Inicialmente, 2 jogava contra 8 e 3 contra 7 na rodada $k1$ (figura no canto superior esquerdo) e 2 jogava contra 7 e 3 contra 8 na rodada

$k2$ (figura no canto superior direito) Com a alternância das rodadas apenas para essas partidas, 2 passa a jogar contra 7 e 3 contra 8 em $k1$ (figura no canto inferior esquerdo) e 2 passa a jogar contra 8 e 3 contra 7 em $k2$ (figura no canto inferior direito). Percebe-se que as demais partidas em $k1$ e $k2$ permanecem inalteradas.

Figura 5: Aplicação da alternância parcial de rodadas envolvendo os times 2, 3, 7 e 8.



Fonte: Elaborado pelo autor.

5.2.5 Rotação de Jogos

A exploração de soluções nessa vizinhança consiste em inserir um jogo envolvendo dois times arbitrários em uma rodada também arbitrária. No entanto, a inserção de um jogo específico cria uma reação em cadeia na solução atual, visto que, a partir do momento da inserção, ambos os times envolvidos passam a jogar duas partidas na mesma rodada, o que não é um cenário válido. A reação em cadeia, portanto, é composta por uma sequência de remoções e inserções de jogos de modo a tornar a solução mais uma vez factível. A Tabela 17 mostra a solução gerada pela inserção da partida G x B na sétima rodada que, antes era jogada na rodada 8 na solução inicial, exibida na Tabela 13.

Tabela 17: Inserção do jogo G x B na sétima rodada.

Rodadas													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
A x B	B x E	B x C	C x G	B x F	A x C	D x A	B x A	E x B	C x B	G x C	B x F	C x A	A x D
G x H	C x D	D x H	E x A	A x H	D x B	C x H	H x G	D x C	H x D	A x E	H x A	B x D	H x C
C x F	G x A	E x G	F x D	E x C	E x H	E x F	F x C	A x G	G x E	D x F	C x E	H x E	F x E
D x E	H x F	F x A	H x B	G x D	F x G	G x B	E x D	F x H	A x F	B x H	D x G	G x F	B x G

Fonte: Elaborado pelo autor.

Para que a inserção do jogo G x B, uma sequência de remoções e inserções foi realizada:

1. A partida G x B foi inserida na rodada 7.
2. As partidas G x D e B x A que envolviam os times G e B foram removidas da rodada 7.
3. A partida D x A, envolvendo os antigos adversários de G e B (D e A, respectivamente), foi adicionada na rodada 7.
Com isso, o jogo envolvendo as equipes D e A passa a existir em duas rodadas distintas (5 e 7).
4. A partida envolvendo D e A é removida da rodada 5
5. A partida envolvendo o time G (nesse caso, G é escolhido por ser o mandante na partida que tem-se o desejo de inserir) é removida da rodada 5 (G x H)
6. A partida envolvendo o time G com o time D (partida removida na “iteração” anterior) é inserida na rodada 5
7. A partida envolvendo o time A e o antigo adversário de G na rodada 5 (H) é inserida
Com isso, o jogo envolvendo as equipes A e H passa a existir em duas rodadas distintas (1 e 5).
8. A partida envolvendo A e H é removida da rodada 1
9. A partida envolvendo o time G (time mandante da partida que tem-se o desejo de inserir) é removida da rodada 1 (B x G)
10. A partida envolvendo o time G com o time H (partida removida na “iteração” anterior) é inserida na rodada 1
11. A partida envolvendo o time A e o antigo adversário de G na rodada 1 (B) é inserida
A tabela está correta e não possui mais inconformidades a partir do momento em que a partida envolvendo os dois times que estavam na partida que desejava-se inserir (G e B) é removida (B x G na rodada 1).

Destaca-se que as inserções e remoções para o segundo turno são realizadas de maneira simultânea aos passos descritos.

Nota-se como os procedimentos são padronizados e uma abordagem recursiva pode ser utilizada para realizar a sequência de inserções e remoções de partidas.

A solução vizinha gerada, embora apresente um número ligeiramente maior de *breaks* (42) quando comparado ao da primeira solução (40), apresenta uma redução nos efeitos de *carry over*, caindo de 784, na solução inicial, para 672.

5.3 Procedimento Iterativo

Para a exploração das diferentes vizinhanças apresentadas é proposto um método iterativo. Inicialmente, definem-se dois parâmetros, o número máximo de iterações e o número máximo de iterações sem melhoria. O processo iterativo possui, portanto, dois critérios de parada, ou seja, o mesmo será encerrado uma vez que o número de iterações atingir o número máximo de iterações definido ou quando um número determinado de iterações não gerar nenhuma melhor solução.

A cada iteração, uma vizinhança, definida de maneira aleatória é explorada. Todas as vizinhanças possuem chances iguais de serem escolhidas. A partir disso, 3 cenários são possíveis: a nova solução explorada pode ser inactível, pode ser factível mas pior que a solução atual ou factível e melhor que a solução atual. Nos dois primeiros casos, a solução explorada é ignorada e se avança para a próxima iteração. No último cenário, a solução explorada passa a ser a solução atual e, conseqüentemente, a que será explorada na iteração seguinte.

A ideia por trás dessa abordagem é combinar a busca por refinamentos locais, através da exploração das vizinhanças que alteram pouco as soluções correntes, com a exploração de mudanças mais radicais para escapar de ótimos locais e explorar novas regiões do espaço de solução, alternando as vizinhanças exploradas. Essa diversificação de vizinhanças permite uma exploração mais ampla do espaço de busca, potencialmente encontrando soluções melhores ou mais diversas.

A escolha aleatória e equitativa entre as diferentes vizinhanças oferece uma abordagem flexível e adaptável, permitindo que o algoritmo explore uma variedade de estratégias de melhoria das soluções, buscando um equilíbrio entre refinamentos locais e mudanças mais disruptivas.

5.4 Abordagem Completa e Pseudo-código

Na Tabela 18, é possível visualizar todas as vizinhanças exploradas, seguidas de suas siglas que serão utilizadas nas análises a seguir.

Tabela 18: Siglas das vizinhanças.

Vizinhança	Sigla
Alternância de mando de uma partida	MHAS
Alternância dos mandos de campo para um time	THAS
Alternância de rodadas	RS
Alternância parcial de rodadas	PRS
Rotação de jogos	GR

Fonte: Elaborado pelo autor.

As siglas utilizadas vêm do inglês (MHAS para *Match Home Away Swap*, THAS para *Team Home Away Swap*, RS para *Round Swap*, PRS para *Partial Round Swap* e GR para *Game Rotation* e estão presentes em Ribeiro e Urrutia (2007a), com exceção da vizinhança THAS (elaboração própria).

O pseudo-código exibido a seguir resume o procedimento composto pela heurística construtiva e pelo processo iterativo que avalia as soluções vizinhas encontradas.

Algorithm 1 Geração da Tabela do Campeonato Brasileiro Série A - 2023

Entradas: T, C, Vizinhanças, MaxIter, MNII

Saída: S^*

```

 $S \leftarrow CriaçãoTabelaTimesFictícios(size(T))$ 
 $S \leftarrow DefiniçãoMandos(S)$ 
while Verdadeiro do
     $S' \leftarrow AlocaçãoTimesReais(S, C)$ 
    if  $EhFactive1(S')$  then
         $ObjetivoInicial \leftarrow S'.Objetivo$ 
        break
    end if
end while
 $MelhorObjetivo \leftarrow ObjetivoInicial$ 
 $S^* \leftarrow S'$ 
 $Iterações \leftarrow 0$ 
 $NII \leftarrow 0$ 
while  $Iterações < MaxIter$  do
     $Vizinhança \leftarrow random(Vizinhanças)$ 
     $S^P \leftarrow Vizinhança(S^*)$ 
    if  $EhFactive1(S^P) = False$  then
         $NII \leftarrow NII + 1$ 
    else if  $S^P.Objetivo < MelhorObjetivo$  then
         $S^* \leftarrow S^P$ 
         $MelhorObjetivo \leftarrow S^P.Objetivo$ 
         $NII \leftarrow 0$ 
    else
         $NII \leftarrow NII + 1$ 
    end if
    if  $NII = MNII$  then
        break
    end if
     $Iterações \leftarrow Iterações + 1$ 
end while

```

O pseudo-código recebe como entradas, quatro variáveis: uma lista T, contendo todos os times envolvidos no campeonato, uma lista C que armazena todas as cidades com os

respectivos times, uma lista *Vizinhanças* que guarda todas as funções que são responsáveis pela exploração das vizinhanças, e dois números inteiros *MaxIter*, que corresponde ao número máximo de buscas nas vizinhanças que serão executadas em cima da solução inicial e *MNII*, do inglês *Maximum Non Improvement Iterations*, que é o valor máximo de iterações consecutivas permitido sem que haja alguma melhoria na solução. Como saída, tem-se a tabela de “custo” mínimo, levando em consideração os objetivos existentes, ou seja, aquela com menor valor para a função objetivo encontrada.

O algoritmo é iniciado com a criação de uma tabela com times fictícios que recebe como parâmetro o número de times que disputarão a competição, através da chamada da função *CriacaoTabelaTimesFicticios*. A tabela gerada é passada como argumento para a função *DefinicaoMandos* que definirá quem serão os mandantes e visitantes de cada uma das partidas geradas. Em seguida, os times fictícios são substituídos na tabela corrente pelos times reais, através da função *AlocacaoTimesReais*, que, além de receber a tabela, recebe, também, a lista de cidades que impactará na ordem em que os times reais serão alocados. A alocação de times reais é executada até que uma solução factível que respeite todas as restrições do modelo seja encontrada.

Posteriormente, são definidas as variáveis *MelhorObjetivo*, S^* , *Iteracoes* e *NII* que armazenarão, respectivamente, o melhor valor da função objetivo encontrado, a melhor tabela (a que leva ao melhor valor de *MelhorObjetivo*), o número de tentativas de melhoria executadas e o número de tentativas de melhoria consecutivas executadas que não levaram a um ganho na função objetivo.

Enquanto o número de buscas locais executadas for inferior ao valor máximo definido, é definido de maneira aleatória qual será a vizinhança explorada utilizada naquela iteração (todas as vizinhanças possuem chances iguais de serem escolhidas). A exploração da vizinhança é realizada através da chamada da função *Vizinhança* e uma nova solução é retornada. Caso a solução não seja factível, o número de *NII* é incrementado em uma unidade. Caso contrário, é avaliado se a nova solução factível encontrada é melhor que a melhor solução atual. Em caso afirmativo, a nova solução encontrada é atribuída à melhor solução, bem como o valor da sua função objetivo é atribuído à variável *MelhorObjetivo*. Caso contrário, o número de *NII* é incrementado em uma unidade. Se o valor da variável *NII* for igual ao valor definido de *MNII*, o algoritmo é interrompido e tem-se a melhor solução encontrada para os parâmetros definidos. Ao final da iteração, o número de iterações executadas é incrementado em uma unidade.

6 EXPERIMENTOS COMPUTACIONAIS E RESULTADOS

Neste capítulo são apresentados os experimentos conduzidos. O capítulo é dividido em quatro seções. Na primeira delas, é analisada a eficiência do algoritmo desenvolvido, confrontando as soluções obtidas pelo modelo (exposto no Capítulo 4) com as geradas pelo algoritmo (exposto no Capítulo 5) em 9 diferentes instâncias, avaliando, também, os resultados gerados por 3 execuções do algoritmo para cada instância e o comportamento de cada uma das variáveis de interesse ao longo do procedimento iterativo proposto pelo algoritmo. Na segunda, cada uma das vizinhanças elaboradas é avaliada, visando encontrar quais são as vizinhanças que mais vezes melhoraram soluções, quanto melhoraram em média para cada instância e se existe algum momento do procedimento iterativo em que cada uma delas tem melhor desempenho. Em seguida, é avaliado o impacto da escolha das penalidades atribuídas às *breaks* e aos efeitos de *carry over* nos resultados de cada uma dessas variáveis, considerando o algoritmo. Por fim, todas as análises e *insights* são consolidados.

Para a realização dos testes do modelo, foi executado um código de programação na linguagem *Python 3.6.0*, utilizando o resolvidor *Gurobi 9.1.2* para a solução do problema. Já a execução do algoritmo proposto, foi também implementada na linguagem *Python*, mas não necessitou do resolvidor. Os testes foram realizados em um computador equipado com Microsoft Windows 11 e processador Intel(R) Core(TM) i7-12700 CPU @ 2.10GHz, 12 Cores, 1 Processador Lógico, com memória RAM instalada de 16 GB.

6.1 Eficiência do Algoritmo

Nesta seção, é avaliada a eficiência do algoritmo desenvolvido quando confrontado com a solução obtida puramente pelo modelo através do resolvidor. Para isso, são analisadas nove instâncias distintas. Para as oito primeiras, são consideradas apenas as restrições que não tenham qualquer influência do Campeonato Brasileiro de Futebol - Série A 2023,

ou seja, tem-se um modelo genérico que poderia ser adaptado para qualquer campeonato do tipo *MDDR*. Para cada uma dessas instâncias é considerado um número diferente de times, sendo eles: 6, 8, 10, 12, 14, 16, 18 e 20. Para a nona instância serão considerados a instância do Campeonato Brasileiro de Futebol - Série A 2023 e o modelo completo apresentado no Capítulo 4. Essa instância será identificada como “20”. Para todas as instâncias, os parâmetros ω_1 e ω_2 , referentes às penalidades de *breaks* e efeitos de *carry over* na função objetivo, foram fixados com o valor 1, ou seja ambos os objetivos possuem igual importância para o modelo.

Na Tabela 19, é possível observar os resultados obtidos tanto pelo modelo como pelo algoritmo. Na primeira coluna, tem-se o número de times n considerados na respectiva instância, sendo 20’ a instância que leva em consideração as restrições exclusivas para o Campeonato Brasileiro de Futebol - Série A 2023. As 4 colunas seguintes trazem os limites inferiores e superiores para cada um dos objetivos do problema.

Werra (1981) prova que o limite inferior para o número de *breaks* para um campeonato do tipo *MDDR* pode ser obtido através de $3n - 6$. O número máximo de *breaks* que uma equipe pode ter seria $2(n - 1) - 1 = 2n - 3$, se ela jogasse todas as partidas fora de casa ou todas as partidas em casa. No entanto, em um torneio *DRR*, cada equipe joga $n - 1$ partidas em casa e $n - 1$ partidas fora de casa. Para maximizar o número de *breaks*, uma equipe deve jogar as primeiras $n - 1$ partidas em casa e as últimas $n - 1$ partidas fora de casa, ou vice-versa. A rodada n é a única sem *break* para essa equipe. Apenas duas equipes podem ter $2n - 4$ *breaks*. As outras $n - 2$ equipes podem ter no máximo $2n - 5$ *breaks*. Portanto, o limite superior de *breaks* para um torneio *DDR* é igual a $2(2n - 4) + (n - 2)(2n - 5) = 2n^2 - 5n + 2$. Observe que as tabelas construídas com base nesta regra são necessariamente espelhadas (URRUTIA; RIBEIRO, 2006). Já os limites inferior e superior para os valores de efeitos de *carry over* foram adaptados de Çavdaroglu e Atan (2022) que consideram um campeonato do tipo *SRR*.

Dentro do bloco MIQP, existem três colunas. A primeira delas traz os valores de *breaks*, valores de efeitos de *carry over* e da função objetivo (soma de cada um dos termos normalizados) encontrados pelo resolvidor através da otimização dos modelos. A coluna seguinte traz o tempo de execução que gerou a respectiva solução, ressaltando-se que o tempo de resolução foi limitado a 12h (43200 segundos), ou seja, caso a solução ótima não tenha sido encontrada nesse tempo, a melhor solução encontrada foi retornada. A última coluna traz o *gap* de otimalidade fornecido pelo resolvidor.

Na sequência, no bloco Algoritmo, têm-se duas colunas. Na primeira delas, tem o

melhor resultado, em termos da função objetivo, encontrado pelo algoritmo dentre três execuções realizadas para aquela instância. A coluna seguinte traz o tempo de execução.

Por fim, o bloco Desvio traz o valor dos desvio entre as soluções obtidas através do modelo e do algoritmo.

Para a aplicação do algoritmo, o valor de MaxIter e MNII, mencionados na Seção 5.4 foram definidos como sendo iguais a n^3 e n^2 , respectivamente, visto que o aumento do número de times aumenta, consideravelmente, o número de possíveis soluções vizinhas a serem exploradas.

Tabela 19: Resultados obtidos pelo modelo e pelo algoritmo para diferentes instâncias.

n	Limitantes				MIQP			Algoritmo		Desvio
	breaks		carry over		<i>Best</i>	Tempo (sec.)	Gap	<i>Best</i>	Tempo (sec.)	
	LB	UB	LB	UB	<i>brk/coe/z</i>			<i>brk/coe/z</i>		
6*	12	44	120	240	12/240/1,000	51,5	0,00%	12/240/1,000	0,38	-0,00%
8	18	90	224	784	20/224/0,028	43200	1356,43%	24/320/0,255	4,67	810,71%
10	24	152	360	1872	40/632/0,305	43200	228,16%	34/584/0,226	15,69	-25,90%
12	30	230	528	3696	124/1080/0,644	43200	146,28%	64/800/0,256	45,44	-60,25%
14	36	324	728	6448	154/1360/0,520	43200	146,52%	86/1176/0,252	190,54	-51,54%
16	42	434	960	10320	408/4544/1,317	43200	115,93%	104/1752/0,243	421,93	-81,55%
18	48	560	1224	15504	276/2344/0,524	43200	133,50%	128/2184/0,223	1033,45	-57,44%
20	54	702	1520	22192	402/3344/0,625	43200	124,73%	158/2864/0,226	3922,31	-63,84%
20 ¹	54	702	1520	22192	324/3056/0,519	43200	130,24%	180/2416/0,238	4251,95	-54,14%

Fonte: Elaborado pelo autor.

Algumas análises podem ser feitas em cima desses resultados. A primeira delas que chama a atenção é o fato de que apenas para a instância $n = 6$ o modelo alcançou a solução ótima dentro do tempo de execução limite proposto. Para todas as outras, soluções sub-ótimas foram encontradas até o tempo limite definido.

Olhando para a melhor solução de cada instância encontrada pelo algoritmo, destaca-se que, para todas as instâncias, com exceção da instância $n = 8$, tal solução é igual ou melhor que a encontrada pelo modelo, inclusive para a solução ótima encontrada em $n = 6$. A solução do algoritmo não é apenas melhor, como é gerada em um tempo consideravelmente menor para todas as instâncias analisadas.

Olhando para as duas instâncias que consideram 20 times, é interessante observar como, apesar de a instância referente ao Campeonato Brasileiro de Futebol - Série A 2023

¹As tabelas geradas pelo modelo e pelo algoritmo, bem como as tabelas de *breaks* e efeitos de *carry over* se encontram no Apêndice A.

possuir mais restrições e, por consequência, analisar um espaço mais restrito de soluções, o tempo de execução foi maior do que o da instância com 20 times genéricos.

Comparando as tabelas obtidas pela otimização do modelo e pela implementação do algoritmo com a tabela do Campeonato Brasileiro de Futebol Série A - 2023 (presente no Capítulo 3), percebe-se que ambas as tabelas geradas possuem um número maior de *breaks* (existem 152 na tabela oficial), porém inferior em efeitos de *carry over* (3152 na tabela oficial). Ambas as tabelas também resultam em um número maior para a função objetivo (0,230 para a tabela oficial). Apesar dos valores piores no número de *breaks* e na função objetivo, o fato de os efeitos de *carry over* apresentar um valor inferior (no modelo e no algoritmo) é algo notável e importante para assegurar a justiça competitiva do campeonato.

Na Tabela 20, é possível observar o número de restrições e de variáveis para cada uma das instâncias previamente apresentadas.

Tabela 20: Número de restrições e variáveis por instância.

n	Restrições	Variáveis
6	2604	780
8	9792	1904
10	26540	3780
12	58892	6600
14	114828	10556
16	203264	15840
18	335052	22644
20	522480	31160
20'	523616	31160

Fonte: Elaborado pelo autor.

Como era imaginado, o número de restrições cresce significativamente com o aumento do número de times. É possível perceber, também, que a enorme parte das restrições para a instância do Campeonato Brasileiro de Futebol Série A vem do modelo com 20 times genéricos, visto que o número de restrições de ambas as instâncias não é muito diferente.

Na Tabela 21, pode-se observar as execuções através do algoritmo em maior nível de detalhe.

Tabela 21: Resultados gerados nas 3 execuções de cada instância.

n	Execuções						Valores médios	
	1		2		3		z	Tempo (s)
	brz/coe/z	Tempo (s)	brz/coe/z	Tempo (s)	brz/coe/z	Tempo (s)		
6*	14/240/1,063	0,51	12/240/1,000	0,84	12/240/1,000	0,38	1,021	0,58
8	26/320/0,283	3,32	26/320/0,283	3,01	24/320/0,255	4,67	0,273	3,67
10	34/584/0,226	15,69	42/648/0,331	10,29	40/576/0,268	15,57	0,275	13,85
12	68/936/0,319	48,41	68/920/0,314	34,38	64/800/0,256	45,44	0,296	42,74
14	86/1176/0,252	190,54	96/1120/0,277	142,49	92/1144/0,267	195,38	0,265	176,14
16	104/1752/0,243	421,93	118/1600/0,262	273,08	134/1720/0,316	370,06	0,274	355,02
18	140/2128/0,243	1430,39	146/2200/0,260	1273,92	128/2184/0,223	1033,45	0,242	1245,90
20	176/2824/0,251	2779,31	202/2984/0,299	1401,51	158/2864/0,226	3922,31	0,259	2701,04
20'	182/2800/0,259	2491,10	186/3248/0,287	3251,67	180/2416/0,238	4251,95	0,262	3331,57

Fonte: Elaborado pelo autor.

Algumas conclusões importantes podem ser obtidas. Primeiramente, os resultados médios também apresentam valores, na maioria das vezes, melhores que os encontrados pelo modelo no tempo limite de execução definido, comprovando a boa performance do algoritmo.

Em segundo lugar, chama a atenção o fato de que os resultados dentro de uma mesma instância não são, necessariamente, muito próximos, o que pode ser justificado pela escolha de soluções vizinhas diferentes durante as iterações, bem como soluções iniciais diferentes.

Por fim, um fato interessante diz respeito ao fato de que o resultado da função objetivo e o tempo decorrido numa execução parecem não ser variáveis correlacionadas. Existem exemplos de casos em que soluções piores foram obtidas em tempos menores de execução, mas também existem casos em que as melhores soluções foram encontradas de forma mais rápida. Isso também pode ser explicado por uma diferença na escolha das soluções vizinhas exploradas e pelo fato de cada execução poder partir de uma diferente solução inicial.

A Figura 6 permite enxergar a evolução dos números de *breaks* e de valores de efeitos de *carry over* para cada uma das instâncias executadas pelo algoritmo. Para a construção de cada um dos gráficos foi utilizada a melhor solução gerada pelo algoritmo. É possível perceber que o comportamento das duas variáveis de interesse é bem parecido ao longo das iterações para todas as instâncias. Inicialmente, ocorre um aumento no número de *breaks* com uma redução significativa nos valores de efeitos de *carry over*. O número de *breaks* tem um pico logo nas primeiras iterações e começa a cair à medida que os valores de efeitos de *carry over* começam a se estabilizar. Esse é um comportamento, um tanto quanto esperado, visto que já era sabido que a solução inicial teria sempre o número

máximo de efeitos de *carry over*.

Figura 6: Evolução de *breaks* e valores de efeitos de *carry over* para a melhor execução de instância.



Fonte: Elaborado pelo autor.

6.2 Análise de Vizinhanças

Nessa seção, serão avaliadas todas as vizinhanças propostas na Seção 5.2, com o objetivo de encontrar aquelas que mais vezes melhoraram a solução corrente e aquelas que possuem maior impacto positivo nas soluções.

Para isso, serão analisadas as mesmas 3 execuções da seção anterior, porém buscando olhar de maneira mais detalhada e profunda o comportamento dos valores de número de *breaks* e de efeitos de *carry over*, bem como da função objetivo a cada iteração. No Apêndice B, estão todas as tabelas que permitiram as análises que serão realizadas.

Enfatiza-se que os valores escolhidos para os parâmetros ω_1 e ω_2 , bem como os limites inferiores e superiores para os valores de número de *breaks* e de efeitos de *carry over* são os mesmos utilizados na análise anterior.

Na Tabela 22, é possível ver para cada uma das vizinhanças, o número de vezes em que

aquela vizinhança foi explorada e conseguiu melhorar a solução corrente em cada instância (através da coluna Contagem) e qual foi o impacto médio daquela vizinhança nas vezes em que conseguiu melhorar a solução corrente (através da coluna Melhoria). Destaca-se que, para ambas as colunas citadas, foram consideradas as 3 execuções realizadas, ou seja, a coluna Contagem traz a soma do número de vezes em que a vizinhança atendeu as condições citadas anteriormente, considerando-se as 3 execuções, assim como a coluna Melhoria traz a média considerando as melhorias de todas as 3 execuções. A última linha da tabela, identificada por Total traz a soma das colunas identificadas com Contagem e a média dos valores de Melhoria, considerando todas as instâncias.

Tabela 22: Desempenho das vizinhanças consideradas nas 9 instâncias propostas.

n	Vizinhança Explorada									
	MHAS		THAS		RS		PRS		GR	
	Contagem	Melhoria	Contagem	Melhoria	Contagem	Melhoria	Contagem	Melhoria	Contagem	Melhoria
6*	3	-13,23%	2	-10,53%	0	-	0	-	1	-4,55%
8	11	-11,34%	3	-8,32%	8	-17,58%	4	-11,19%	8	-8,32%
10	14	-7,36%	2	-6,47%	8	-18,43%	19	-5,10%	8	-6,79%
12	20	-4,85%	6	-5,63%	16	-14,55%	0	-	3	-4,36%
14	37	-3,41%	7	-3,95%	19	-13,37%	0	-	2	-1,63%
16	33	-2,49%	1	-1,01%	14	-14,94%	26	-3,05%	8	-9,76%
18	49	-2,36%	3	-2,19%	21	-9,94%	42	-1,39%	22	-2,83%
20	75	-1,58%	1	-2,60%	27	-10,68%	0	-	3	-1,90%
20'	104	-1,36%	4	-1,31%	33	-7,88%	0	-	1	-0,00%
Total	346	-5,33%	29	-4,67%	146	-13,42%	91	-5,18%	56	-4,46%

Fonte: Elaborado pelo autor.

A partir desses dados, uma série de análises e conclusões podem ser tiradas. Com relação ao valor da coluna Contagem para cada uma das vizinhanças, percebe-se um padrão para as vizinhanças MHAS e RS. À medida que o número de times aumenta, o número de vezes em que essas vizinhanças foram capazes de melhorar a solução corrente aumenta também. Para as outras vizinhanças, não foi possível notar a presença de algum padrão, mas chama a atenção o fato de que, para as instâncias $n = 16$ e $n = 18$, a vizinhança PRS tenha conseguido várias vezes melhorar a solução corrente, enquanto que para as instâncias $n = 12$, $n = 14$, $n = 20$ e a instância do Campeonato Brasileiro de Futebol - Série A 2023, ela não tenha sido capaz uma única vez de ter um impacto positivo nas soluções. Uma situação semelhante ocorre para a vizinhança GR em $n = 18$, na qual a vizinhança melhorou as soluções correntes por 22 vezes, valor muito superior a todas as outras instâncias.

Com relação aos valores das colunas Melhoria, alguns fatos interessantes são percebidos. Para as primeiras 4 vizinhanças (MHAS, THAS, RS e PRS), observa-se que, à

medida que o número de times aumenta, o impacto médio da vizinhança (coluna Melhoria) tende a diminuir, enquanto que para a vizinhança GR isso não parece acontecer. Isso pode ser explicado pelo fato de que as 4 primeiras vizinhanças possuem um número fixo de “movimentos” que serão realizados em cada solução, o que não acontece para a vizinhança GR. Por exemplo, para a vizinhança MHAS, sabe-se de antemão que a solução vizinha que será gerada possui apenas 2 partidas diferentes à da solução que está sendo avaliada, que são as 2 partidas que terão seus mandos de campo invertido (no caso, ambas as partidas envolvem os mesmos dois times, uma no primeiro turno, a outra no segundo). Para a vizinhança THAS, sabe-se que a solução vizinha terá todas as partidas para um dado time com mando invertido, ou seja, 38 partidas serão alteradas. A vizinhança RS apenas inverte duas rodadas, ou seja, 40 partidas (20 no primeiro turno e 20 no segundo) serão realizadas em rodadas distintas às da solução que está sendo avaliada. Por fim, na vizinhança PRS, ocorre essa mesma inversão de rodadas, mas para um número inferior de partidas (8). Entretanto, a vizinhança GR explora a inserção de uma partida específica em uma rodada específica, ambas decididas de maneira aleatória e não é possível prever o número de partidas que deverão ser removidas e reinseridas durante a geração da solução vizinha.

Outro resultado que chama bastante a atenção é que, para todas as instâncias, com exceção de $n = 6$ em que não foi utilizada, a vizinhança RS foi a que mais teve impacto nas soluções correntes. Sabe-se que, das 5 vizinhanças, apenas as últimas 3 têm potencial de impactar tanto o número de *breaks* como de efeitos de *carry over* (as duas primeiras impactam apenas no número de *breaks*) e isso ajuda a explicar esse maior valor de melhoria. Dentre as 3 que impactam em ambas as variáveis de interesse, a vizinhança RS é aquela que tem o maior potencial de impactar positivamente na solução. Para o número de *breaks* pode-se pensar em um cenário em que um time dispute 3 partidas consecutivas em casa. Isso geraria duas *breaks* para esse time, nesse intervalo. Imaginando que uma rodada diferente seja inserida exatamente no lugar da segunda rodada desse conjunto de 3 rodadas, e que nessa rodada inserida o time em questão dispute sua partida fora de casa, o time passaria a não ter mais 2 ou mais partidas consecutivas em casa nesse intervalo e o número de *breaks* cairia de 2 para 0. Esse é um cenário que considera apenas uma sequência de 3 rodadas para um time. A vizinhança RS poderia, no melhor dos cenários, ter esse mesmo impacto para os n times em 4 sequências de 3 jogos, já que 2 rodadas são invertidas em cada turno. Com relação aos efeitos de *carry over*, já foi demonstrado na Seção 5.2.3 que essa vizinhança tem impacto sobre essas variáveis e é fácil entender o porquê. Com a inversão de rodadas, ocorrerá uma redistribuição dos efeitos de *carry*

over, visto que os times terão uma nova sequência de adversários.

Retomando as colunas de Melhoria, destaca-se mais uma vez que todos os valores apresentados são valores médios, o que acaba limitando outra análise interessante. É possível que o momento em que as diferentes vizinhanças são exploradas seja uma importante variável na melhoria resultante. Por exemplo, alguma vizinhança pode ter um comportamento mais agressivo e com maior potencial de melhoria quando utilizada logo nas primeiras iterações, enquanto outra pode funcionar melhor em momentos finais, de “ajustes finos”.

Sendo assim, buscou-se olhar, dentre todas as execuções presentes no Apêndice B, aquelas que por mais vezes apresentaram cada uma das vizinhanças consideradas para que, dessa forma, possa-se analisar com uma amostra maior de dados, se os momentos em que cada uma delas é utilizada para melhorar a solução corrente, de fato, tem impactos diferentes. Na Tabela 23 é possível visualizar qual instância e execução presente no Apêndice B foi considerada para cada uma das vizinhanças e quantas vezes a vizinhança melhorou a solução corrente na instância em questão (através da coluna Contagem).

Tabela 23: Performance das vizinhanças consideradas nas 9 instâncias propostas.

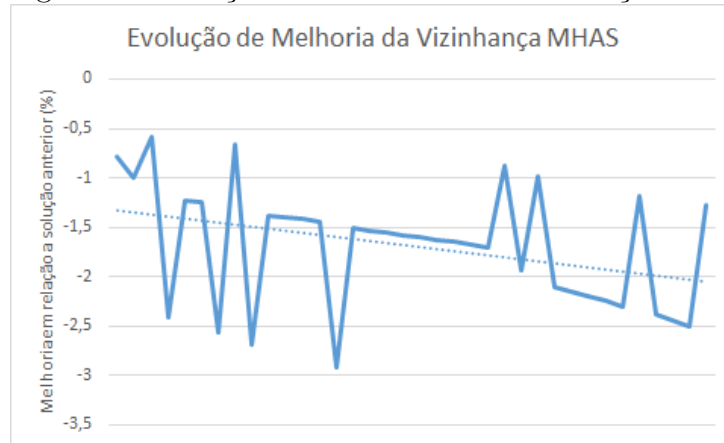
Vizinhança	Instância / Execução	Contagem
MHAS	Instância $n = 20'$ / Execução 3	36
THAS	Instância $n = 12$ / Execução 2	3
RS	Instância $n = 20'$ / Execução 1	13
PRS	Instância $n = 18$ / Execução 1	21
GR	Instância $n = 18$ / Execução 1	8

Fonte: Elaborado pelo autor.

Consideram-se as contagens para as vizinhanças THAS e GR muito pequenas e, portanto, insuficientes para realizar qualquer análise.

Na Figura 7, é possível observar os valores de Melhoria associados à vizinhança MHAS ao longo das iterações.

Figura 7: Evolução de Melhoria da Vizinhança MHAS



Fonte: Elaborado pelo autor.

Os resultados evidenciam uma faixa limitada de valores de melhoria, variando entre -3 e -0,5, para a vizinhança MHAS. Este padrão sugere uma consistência notável, indicando uma tendência moderada ao longo do processo. Era esperado que os valores de melhoria permanecessem muito próximos e constantes ao longo das iterações, visto que essa vizinhança influencia em apenas duas partidas (uma em cada turno) e tem potencial apenas de reduzir o número de *breaks*. Interessantemente, à medida que as iterações avançam, torna-se mais aparente o potencial crescente de melhoria dentro desta vizinhança. Embora essa mudança não se apresente de forma pronunciada, sua presença sutil pode fornecer informações valiosas, especialmente ao considerar a exploração de diferentes vizinhanças em momentos específicos. Esta percepção aponta para a possibilidade de que determinadas estratégias de otimização ou ajustes se tornem mais eficazes em estágios posteriores do processo, abrindo oportunidades para aprimoramentos graduais ao longo do tempo.

Na Figura 8, é possível observar os valores de Melhoria associados à vizinhança RS ao longo das iterações.

Para concluir, a análise da vizinhança PRS indica um potencial maior de melhoria nas fases iniciais do processo iterativo, mas de maneira não tão acentuada como na vizinhança anterior. Essa dinâmica é fundamentada na mesma lógica observada na vizinhança RS. A vizinhança PRS também tem a capacidade de influenciar significativamente nos efeitos de *carry over*, no entanto, a diferença crucial reside no número menor de partidas que são invertidas em comparação com a alternância total de rodadas oferecida pela vizinhança RS. Como resultado, embora essa abordagem também seja eficaz na mitigação dos efeitos de *carry over*, sua influência acaba sendo menos expressiva quando comparada à estratégia de inverter completamente as rodadas.

6.3 Trabalhando com Diferentes Penalidades

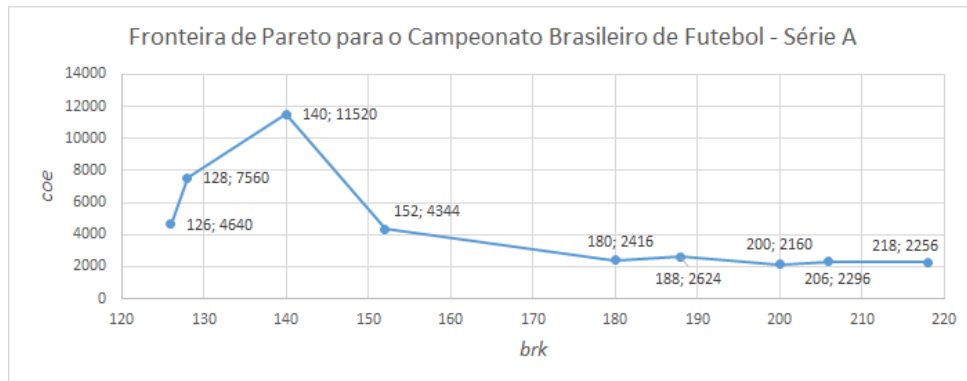
O problema abordado neste TF apresenta uma estrutura com dois objetivos principais: a redução do número de *breaks* e a redução dos efeitos de *carry over*. Este é um desafio complexo, uma vez que ambos os objetivos precisam ser otimizados simultaneamente. A busca pela solução ideal torna-se ainda mais desafiadora devido à natureza conflitante desses objetivos, já que, ao tentar minimizar um dos objetivos, pode-se inevitavelmente prejudicar a otimização do outro.

Uma vez que não é viável alcançar os valores mínimos de cada problema separadamente, existe o dilema de atribuir diferentes níveis de importância à minimização do número de *breaks* e de efeitos de *carry over*. Dependendo das preferências e prioridades específicas do Campeonato Brasileiro de Futebol - Série A, as escolhas feitas na criação da tabela podem variar. Por exemplo, pode-se optar por priorizar uma redução significativa no número de *breaks*, mesmo que isso resulte em um aumento nos efeitos de *carry over*. Em contrapartida, a minimização desses efeitos pode ser considerada mais crucial para a criação da tabela, mesmo que isso acabe acarretando em um aumento no número de *breaks*.

Essa flexibilidade na abordagem permite que sejam realizados diferentes *trade-offs* entre a redução do número de *breaks* e a minimização de efeitos de *carry over*, refletindo assim as preferências e objetivos específicos do Campeonato Brasileiro de Futebol - Série A.

Pensando nisso, a Figura 10 traz uma fronteira de Pareto construída para a instância do Campeonato Brasileiro de Futebol - Série A 2023 através do algoritmo proposto.

Figura 10: Fronteira de Pareto para a instância do Campeonato Brasileiro de Futebol - Série A



Fonte: Elaborado pelo autor.

A Tabela 24 identifica qual conjunto de penalidades escolhidos (ω_1 e ω_2) levou a cada solução exibida na fronteira de Pareto. Na primeira coluna, têm-se os dois valores escolhidos para os parâmetros ω_1 e ω_2 e, nas duas colunas seguintes, os valores do número de *breaks* e efeitos de *carry over* encontrados. Enfatiza-se que são resultados obtidos através de uma única execução do algoritmo e que existem muitas componentes aleatórias na geração de cada uma das soluções, o que faz com que diferentes resultados possam ser encontrados para uma mesma instância. Destaca-se, também, que os valores do número de *breaks* e de efeitos de *carry over* encontrados para o conjunto de penalidades $(\omega_1, \omega_2) = (1, 1)$ se referem aos resultados exibidos na última linha da Tabela 19.

Tabela 24: Valores encontrados de *breaks* e efeitos de *carry over* para diferentes penalidades.

ω_1 / ω_2	<i>brk</i>	<i>coe</i>
10 / 1	140	11520
5 / 1	128	7560
3 / 1	126	4640
2 / 1	152	4344
1 / 1	180	2416
1 / 2	188	2624
1 / 3	218	2256
1 / 5	200	2160
1 / 10	206	2296

Fonte: Elaborado pelo autor.

A partir da Tabela 24, podem ser realizadas algumas análises. Primeiramente, pode-se notar que, para nenhum conjunto de penalidades estipulado, a solução gerada apresenta valores que sejam simultaneamente melhores em *breaks* e em efeitos de *carry over* quando comparados aos da tabela do Campeonato Brasileiro de Futebol Série A - 2023 (152 e 3152, respectivamente, ambos calculados no Capítulo 3). Entretanto, é interessante notar como para todos os conjuntos de penalidades com $\omega_1 > \omega_2$, as soluções encontradas possuem menos *breaks* do que a tabela oficial do campeonato (152), e, como para todos os conjuntos de penalidades com $\omega_2 > \omega_1$, as soluções encontradas possuem menos efeitos de *carry over* do que a tabela oficial do campeonato (3152). Percebe-se, também, que não necessariamente atribuir uma maior penalidade para uma das variáveis fará com que os resultados para aquela variável sejam melhores. Isso fica evidente para ambas as variáveis, pois percebe-se que, em ambos os casos, quando uma penalidade 10 foi associada, os valores para a variável em questão não foram os melhores dentre todos os gerados. Para o caso das *breaks*, o melhor valor foi encontrado quando a penalidade ω_1 assumiu valor 3 (mantendo o valor de ω_2 em 1), enquanto que, para os efeitos de *carry over*, o melhor valor foi alcançado ao atribuir valor 5 para a penalidade ω_2 (mantendo o valor de ω_1 em 1).

Esse comportamento pode ser entendido analisando o comportamento das duas variáveis ao longo das execuções, como pôde ser visto na Figura 6. Percebe-se que a queda no número de *breaks* acontece após um aumento nesse número que, por sua vez, acontece simultaneamente a uma redução nos efeitos de *carry over*. Ao atribuir uma penalidade alta para o número de *breaks*, torna-se mais difícil que sejam geradas soluções em que aconteça a redução dos efeitos de *carry over* em detrimento a um aumento do número de *breaks*. O número de *breaks* não aumentando, por sua vez, impede que os efeitos de *carry over* caiam e se estabilizem em um patamar que permita a redução do número de *breaks*. Resumindo, da maneira como o algoritmo foi desenvolvido, é necessário que existam *trade-offs* ao longo das iterações que, por sua vez, só são possíveis se as penalidades atribuídas a cada uma das variáveis não sejam muito distantes.

Isso ajuda a explicar o ponto (140, 11520) na Figura 10 que, dentre todos, é o que chama mais atenção. O fato de a penalidade associada às *breaks* ser muito alta impede a ocorrência dos *trade-offs* e a solução encontrada acaba tendo o maior número de *breaks* e de efeitos de *carry over*, dentre aquelas cuja penalidade associada às *breaks* era maior.

6.4 Consolidação das Análises Realizadas e Principais *Insights*

De forma geral, nesse capítulo, puderam ser realizadas uma série de análises a respeito do modelo proposto e do algoritmo implementado. O capítulo apresentou, inicialmente, uma análise minuciosa da eficiência do algoritmo, exibindo os resultados para nove instâncias distintas do problema em questão. Dentro desse contexto, observou-se que o modelo alcançou a solução ótima em apenas uma das instâncias, cumprindo o tempo de execução estabelecido. De maneira surpreendente, o algoritmo também conseguiu atingir a solução ótima para essa instância específica. No entanto, nas demais instâncias, o modelo não conseguiu alcançar a solução ótima, e somente em uma delas superou o desempenho do algoritmo. Esses resultados destacam a complexidade do problema e a dificuldade de encontrar soluções ótimas em diferentes cenários.

Uma análise detalhada da evolução das variáveis de interesse ao longo das iterações proporcionou *insights* significativos. Foi percebido um padrão comum entre todas as instâncias, revelando a presença de *trade-offs* durante o procedimento iterativo. Esses *trade-offs* indicam a necessidade de equilibrar a minimização do número de *breaks* e dos efeitos de *carry over*, demonstrando a natureza desafiadora e multifacetada do problema abordado.

Além disso, foi conduzida uma análise profunda das diversas vizinhanças propostas. Nessa investigação, foram examinadas não apenas a frequência com que cada vizinhança melhorou a solução corrente, mas também a magnitude média das melhorias e os momentos em que cada vizinhança teve maior impacto nas soluções. Essa análise minuciosa ofereceu uma compreensão mais completa das estratégias de vizinhança, revelando abordagens que podem influenciar diretamente no processo iterativo e podem conduzir a soluções em um tempo consideravelmente menor, utilizando e explorando as vizinhanças nos momentos em que podem produzir maior impacto sobre as soluções.

No decorrer do estudo, foi também investigado o impacto do uso de diferentes penalidades para cada variável do problema, considerando a solução através do algoritmo. Intrigantemente, constatou-se que associar uma penalidade alta a uma das variáveis não necessariamente resultou em melhores resultados para essa variável. Esse achado sublinha a complexidade na definição de penalidades, evidenciando a importância de uma abordagem estratégica e flexível na configuração do algoritmo. Se a mesma análise fosse realizada através da otimização do modelo em si, sem a estratégia de construir e melhorar uma solução através do algoritmo proposto, provavelmente, constataria-se que as

penalidades estão fortemente correlacionadas com os resultados de cada variável, como é possível observar no trabalho de Çavdaroglu e Atan (2022), no qual a fronteira de Pareto exposta é melhor comportada.

Essas análises abrangentes oferecem uma visão holística do desempenho do algoritmo, das interações entre as variáveis essenciais do problema e da influência das estratégias de busca. Os *insights* obtidos enriquecem não apenas a compreensão do comportamento do algoritmo, mas também a dinâmica dos *trade-offs* e a sensibilidade do algoritmo às escolhas de penalidades, contribuindo assim para uma abordagem mais informada na resolução do mesmo problema.

Além disso, podem ser citados os benefícios de diferente pontos de vista. Do ponto de vista do Campeonato Brasileiro de Futebol - Série A como um “produto”, aplicação do modelo matemático e/ou do algoritmo propostos neste estudo oferece ao campeonato a possibilidade de alcançar uma estrutura mais justa na esfera esportiva. A busca pela justiça competitiva, fundamentada em critérios objetivos, representa um avanço significativo na criação da tabela de jogos. Essa abordagem pode impactar positivamente a experiência geral do campeonato, promovendo partidas mais equilibradas e um ambiente competitivo mais justo para todas as equipes envolvidas. A possibilidade de reduzir desequilíbrios e minimizar situações desfavoráveis para determinados times é um aspecto crucial para fortalecer a credibilidade e a atratividade do campeonato para os espectadores e mídia em geral.

A Confederação Brasileira de Futebol (CBF), organização responsável por organizar o Campeonato Brasileiro de Futebol - Série A, por sua vez, pode se beneficiar significativamente ao adotar um modelo matemático e algoritmo que visam assegurar a justiça competitiva na disputa. Essa abordagem oferece à organização uma ferramenta sólida e embasada para a elaboração das tabelas de jogos. Ao implementar critérios objetivos, a confederação demonstra um compromisso genuíno com a promoção de uma competição justa e balanceada. Além disso, a transparência e a objetividade proporcionadas por esses métodos podem fortalecer a confiança dos torcedores, patrocinadores e *stakeholders* no processo de organização do campeonato.

Os times envolvidos no campeonato são, no fim, os maiores interessados em uma tabela justa e balanceada. Encontrar situações desfavoráveis na mesma, como uma longa sequência de jogos difíceis é uma preocupação fundamental para os mesmos. A garantia de uma competição justa e equilibrada, com capacidade de nivelar as condições de disputa entre os times proporciona a todos uma oportunidade mais igualitária de competir. Re-

duzir a ocorrência de sequências desfavoráveis de jogos e minimizar possíveis vantagens ou desvantagens resultantes da distribuição irregular de partidas pode impactar diretamente o desempenho e a motivação das equipes. A previsibilidade e a justiça na distribuição de jogos oferecem às equipes a oportunidade de se prepararem de maneira mais eficaz, buscando maximizar suas chances de sucesso em um ambiente competitivo mais justo e balanceado. Além disso, ao participar de um torneio mais justo, os times têm a oportunidade de demonstrar verdadeiramente seu potencial e habilidades, elevando o nível de competitividade e valorizando a qualidade do espetáculo esportivo.

Esses *insights* destacam não apenas os potenciais benefícios tangíveis para o Campeonato Brasileiro de Futebol - Série A, a CBF e os times participantes, mas também ressaltam a importância de adotar abordagens mais científicas e objetivas na formulação de estratégias para as competições esportivas de alto nível.

7 CONCLUSÕES E PERSPECTIVAS FUTURAS

O presente Trabalho de Formatura abordou o Problema de Geração de Tabelas de Campeonatos, buscando dar mais atenção para uma instância específica: a do Campeonato Brasileiro de Futebol - Série A 2023. Os motivos que levaram à escolha desse tema foram a alta complexidade advinda desse problema combinatório, que permitiram que o autor colocasse em prática conhecimentos adquiridos durante o curso e se desafiasse a aprender sobre novas formas de resolução de problemas de Pesquisa Operacional, e a afinidade e familiaridade do autor com a instância escolhida. A automatização da geração da tabela desse campeonato é uma alternativa importante e que já é adotada pelas entidades que organizam o campeonato em questão.

Para a solução do problema proposto, foram elaboradas duas abordagens. A primeira delas consiste na otimização de um modelo matemático original que contempla uma série de restrições, sendo algumas delas observáveis nas tabelas oficiais e outras de autoria do aluno que acreditou que seriam relevantes para a geração de uma tabela considerada boa. Destaca-se que existem trabalhos e artigos científicos que tratam da mesma instância, mas percebeu-se que algumas informações estavam defasadas e algumas restrições que haviam sido consideradas no passado já não eram mais levadas em conta na geração das tabelas mais recentes.

A segunda abordagem consistiu na implementação de um algoritmo, composto de duas partes fundamentais. Uma heurística construtiva que, em 3 etapas, gera rapidamente uma solução inicial, e um processo iterativo que explora um conjunto de vizinhanças, buscando encontrar melhores resultados.

A partir dessas duas abordagens, foi possível construir tabelas que atendessem todas as restrições levantadas, buscando minimizar os dois grandes objetivos propostos: o número de *breaks* e os efeitos de *carry over*, que, por sua vez, são dois conceitos amplamente discutidos na literatura sobre o tema, tendo diversas abordagens de solução. Enfatiza-se o diferencial desse trabalho em abordar os dois objetivos de maneira conjunta, já que, comumente, esses dois objetivos são abordados de maneira isolada.

Como perspectivas futuras, o modelo, bem como o algoritmo desenvolvidos poderiam ser apresentados às entidades organizadoras como mais uma alternativa na geração de uma tabela que, sob a ótica dos objetivos levantados, pode ser considerada justa e propiciar condições iguais entre os times. Além disso, há espaço para expandir o modelo, incorporando mais restrições e necessidades pertinentes, bem como para avaliar a aplicabilidade desse mesmo modelo a outros campeonatos, potencialmente em diferentes modalidades esportivas, indo além do escopo do Campeonato Brasileiro de Futebol - Série A. Contudo, isso exigiria uma reavaliação criteriosa das restrições existentes e a identificação de novas que se adéquem à nova instância.

Convém, também, destacar que os experimentos computacionais realizados possuíam um número considerável de restrições e variáveis e foram solucionados por meio da aplicação de abordagens de PO através do otimizador *Gurobi*. As soluções geradas pela otimização do modelo apresentaram *gaps* de otimalidade consideráveis para os tempos de execução definidos, ou seja, para o limite de tempo proposto, o modelo não foi capaz de atingir a solução ótima para as maiores instâncias elaboradas. O algoritmo gera soluções, na maioria das vezes, melhores e em intervalos de tempo consideravelmente menores do que os propostos para a otimização do modelo, mas para a principal instância desse TF, a melhor solução obtida não apresentou valores melhores para ambos os objetivos, quando confrontada com a tabela real do campeonato.

Este cenário abre espaço para a exploração de novos métodos de solução, como o uso de técnicas de decomposição e relação, ou ainda o aprimoramento do algoritmo, explorando novas estratégias para a construção de soluções iniciais, a criação de novos conjuntos de vizinhanças a serem exploradas durante o processo de refinamento das soluções, a variação das penalidades associadas a cada um dos objetivos e a adoção de diferentes estratégias durante o processo iterativo, explorando as vizinhanças em momentos mais assertivos. A implementação de meta-heurísticas também poderia ser analisada.

REFERÊNCIAS

- ALARCÓN, F.; DURÁN, G.; GUAJARDO, M. Referee assignment in the chilean football league using integer programming and patterns. *International Transactions in Operational Research*, Wiley Online Library, v. 21, n. 3, p. 415–438, 2014.
- BALL, B. C.; WEBSTER, D. B. Optimal scheduling for even-numbered team athletic conferences. *AIIE Transactions*, Taylor & Francis, v. 9, n. 2, p. 161–169, 1977.
- BARTSCH, T.; DREXL, A.; KRÖGER, S. Scheduling the professional soccer leagues of austria and germany. *Computers & Operations Research*, Elsevier, v. 33, n. 7, p. 1907–1937, 2006.
- BEAN, J. C.; BIRGE, J. R. Reducing travelling costs and player fatigue in the national basketball association. *Interfaces*, INFORMS, v. 10, n. 3, p. 98–102, 1980.
- BONOMO, F. et al. An application of the traveling tournament problem: The argentine volleyball league. *Interfaces*, Informs, v. 42, n. 3, p. 245–259, 2012.
- BRISKORN, D.; DREXL, A. A branching scheme for finding cost-minimal round robin tournaments. *European Journal of Operational Research*, Elsevier, v. 197, n. 1, p. 68–76, 2009.
- BRISKORN, D.; DREXL, A. Ip models for round robin tournaments. *Computers & Operations Research*, Elsevier, v. 36, n. 3, p. 837–852, 2009.
- BRISKORN, D.; KNUST, S. Constructing fair sports league schedules with regard to strength groups. *Discrete Applied Mathematics*, Elsevier, v. 158, n. 2, p. 123–135, 2010.
- BULCK, D. V. et al. Robinx: A three-field classification and unified data format for round-robin sports timetabling. *European Journal of Operational Research*, Elsevier, v. 280, n. 2, p. 568–580, 2020.
- BULCK, D. V.; GOOSSENS, D. R.; SPIEKSMAN, F. C. Scheduling a non-professional indoor football league: a tabu search based approach. *Annals of Operations Research*, Springer, v. 275, n. 2, p. 715–730, 2019.
- BURAIMO, B.; FORREST, D.; SIMMONS, R. Insights for clubs from modelling match attendance in football. *Journal of the Operational Research Society*, Springer, v. 60, p. 147–155, 2009.
- ÇAVDAROĞLU, B.; ATAN, T. Integrated break and carryover effect minimization. *Journal of Scheduling*, Springer, v. 25, n. 6, p. 705–719, 2022.
- CBF, A. *CCBF divulga tabela da Série A do Campeonato Brasileiro 2023*. 2023. <https://www.cbf.com.br/futebol-brasileiro/noticias/campeonato-brasileiro-serie-a/cbf-divulga-tabela-da-serie-a-do-campeonato-brasileiro-2023> [Acesso em: (27/11/2023)].

- CLARKE, S. R.; NORMAN, J. M. Optimal challenges in tennis. *Journal of the Operational Research Society*, Taylor & Francis, v. 63, n. 12, p. 1765–1772, 2012.
- COCCHI, G. et al. Scheduling the italian national volleyball tournament. *Interfaces, INFORMS*, v. 48, n. 3, p. 271–284, 2018.
- COSTA, D. An evolutionary tabu search algorithm and the nhl scheduling problem. *INFOR: Information Systems and Operational Research*, Taylor & Francis, v. 33, n. 3, p. 161–178, 1995.
- CROCE, F. D.; OLIVERI, D. Scheduling the italian football league: An ilp-based approach. *Computers & Operations Research*, Elsevier, v. 33, n. 7, p. 1963–1974, 2006.
- DREXL, A.; KNUST, S. Sports league scheduling: Graph-and resource-based models. *Omega*, Elsevier, v. 35, n. 5, p. 465–471, 2007.
- DUARTE, A. R.; RIBEIRO, C. C.; URRUTIA, S. A hybrid ils heuristic to the referee assignment problem with an embedded mip strategy. *Lecture Notes in Computer Science*, Springer, v. 4771, p. 82–95, 2007.
- DURÁN, G. et al. Scheduling the chilean soccer league by integer programming. *Interfaces, INFORMS*, v. 37, n. 6, p. 539–552, 2007.
- DURÁN, G.; GUAJARDO, M.; SAURÉ, D. Scheduling the south american qualifiers to the 2018 fifa world cup by integer programming. *European Journal of Operational Research*, Elsevier, v. 262, n. 3, p. 1109–1115, 2017.
- DURÁN, G.; GUAJARDO, M.; WOLF-YADLIN, R. Operations research techniques for scheduling chile's second division soccer league. *Interfaces, INFORMS*, v. 42, n. 3, p. 273–285, 2012.
- DURÁN, G. A. et al. Scheduling multiple sports leagues with travel distance fairness: An application to argentinean youth football. *INFORMS Journal on Applied Analytics, Inform*, v. 51, n. 2, p. 136–149, 2021.
- EASTON, K.; NEMHAUSER, G.; TRICK, M. The traveling tournament problem description and benchmarks. In: SPRINGER. *Principles and Practice of Constraint Programming—CP 2001: 7th International Conference, CP 2001 Paphos, Cyprus, November 26–December 1, 2001 Proceedings 7*. [S.l.], 2001. p. 580–584.
- EASTON, K. K. *Using integer programming and constraint programming to solve sports scheduling problems*. [S.l.]: Georgia Institute of Technology, 2003.
- EGLESE, R. W. Simulated annealing: a tool for operational research. *European Journal of Operational Research*, Elsevier, v. 46, n. 3, p. 271–281, 1990.
- FERLAND, J. A.; FLEURENT, C. Computer aided scheduling for a sport league. *INFOR: Information Systems and Operational Research*, Taylor & Francis, v. 29, n. 1, p. 14–25, 1991.
- FIALLOS, J. et al. Scheduling soccer league of honduras using integer programming. In: INSTITUTE OF INDUSTRIAL AND SYSTEMS ENGINEERS (IIE). *IIE Annual Conference. Proceedings*. [S.l.], 2010. p. 1.

- FRY, M. J.; OHLMANN, J. W. *Introduction to the special issue on analytics in sports, part II: Sports scheduling applications*. [S.l.]: INFORMS, 2012. 229–231 p.
- GENDREAU, M.; POTVIN, J.-Y. Tabu search. *Search methodologies: introductory tutorials in optimization and decision support techniques*, Springer, p. 165–186, 2005.
- GERCHAK, Y. Operations research in sports. *Handbooks in operations research and management science*, Elsevier, v. 6, p. 507–527, 1994.
- GOOSSENS, D.; SPIEKSMA, F. Scheduling the belgian soccer league. *Interfaces*, INFORMS, v. 39, n. 2, p. 109–118, 2009.
- GOOSSENS, D. R.; SPIEKSMA, F. C. Soccer schedules in europe: an overview. *Journal of Scheduling*, Springer, v. 15, p. 641–651, 2012.
- GUEDES, A. C.; RIBEIRO, C. C. A heuristic for minimizing weighted carry-over effects in round robin tournaments. *Journal of Scheduling*, Springer, v. 14, p. 655–667, 2011.
- GÜNNEÇ, D.; DEMIR, E. Fair-fixtute: Minimizing carry-over effects in football leagues. *Journal of Industrial and Management Optimization*, v. 15, n. 4, p. 1565–1577, 2019.
- HOSHINO, R.; KAWARABAYASHI, K.-i. A multi-round generalization of the traveling tournament problem and its application to japanese baseball. *European journal of operational research*, Elsevier, v. 215, n. 2, p. 481–497, 2011.
- HOSHINO, R.; KAWARABAYASHI, K.-i. Scheduling bipartite tournaments to minimize total travel distance. *Journal of Artificial Intelligence Research*, v. 42, p. 91–124, 2011.
- KENDALL, G. et al. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, Elsevier, v. 37, n. 1, p. 1–19, 2010.
- KNUST, S. Scheduling non-professional table-tennis leagues. *European Journal of Operational Research*, Elsevier, v. 200, n. 2, p. 358–367, 2010.
- KNUST, S. *Classification of Literature on Sports Scheduling*. 2018. Disponível em: http://www2.inf.uos.de/knust/sportssched/sportlit_class/sportlit_class.html.
- KNUST, S.; THADEN, M. V. Balanced home-away assignments. *Discrete Optimization*, Elsevier, v. 3, n. 4, p. 354–365, 2006.
- KOSTUK, K. J.; WILLOUGHBY, K. A. A decision support system for scheduling the canadian football league. *Interfaces*, INFORMS, v. 42, n. 3, p. 286–295, 2012.
- KYNGAS, J.; NURMI, K. Scheduling the finnish major ice hockey league. In: IEEE. *2009 IEEE Symposium on Computational Intelligence in Scheduling*. [S.l.], 2009. p. 84–89.
- KYNGÄS, J. et al. Scheduling the australian football league. *Journal of the Operational Research Society*, Taylor & Francis, v. 68, n. 8, p. 973–982, 2017.
- LARSON, J.; JOHANSSON, M. Constructing schedules for sports leagues with divisional and round-robin tournaments. *Journal of Quantitative Analysis in Sports*, De Gruyter, v. 10, n. 2, p. 119–129, 2014.

- LEUNG, J. Y. *Handbook of scheduling: algorithms, models, and performance analysis*. [S.l.]: CRC press, 2004.
- LEWIS, R.; THOMPSON, J. On the application of graph colouring techniques in round-robin sports scheduling. *Computers & Operations Research*, Elsevier, v. 38, n. 1, p. 190–204, 2011.
- LOZANO, S. et al. Measuring the performance of nations at the summer olympics using data envelopment analysis. *Journal of the Operational Research Society*, Springer, v. 53, p. 501–511, 2002.
- MELO, R. A.; URRUTIA, S.; RIBEIRO, C. C. Scheduling single round robin tournaments with fixed venues. In: CITESEER. *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications, Paris*. [S.l.], 2007. p. 431–438.
- MIETTINEN, K. *Nonlinear multiobjective optimization*. [S.l.]: Springer Science & Business Media, 1999. v. 12.
- MIYASHIRO, R.; IWASAKI, H.; MATSUI, T. Characterizing feasible pattern sets with a minimum number of breaks. In: SPRINGER. *Practice and Theory of Automated Timetabling IV: 4th International Conference, PATAT 2002, Gent, Belgium, August 21-23, 2002. Selected Revised Papers 4*. [S.l.], 2003. p. 78–99.
- NEMHAUSER, G. L.; TRICK, M. A. Scheduling a major college basketball conference. *Operations research*, INFORMS, v. 46, n. 1, p. 1–8, 1998.
- NURMI, K. et al. A framework for scheduling professional sports leagues. In: AMERICAN INSTITUTE OF PHYSICS. *AIP Conference Proceedings*. [S.l.], 2010. v. 1285, n. 1, p. 14–28.
- NURMI, K.; GOOSSENS, D.; KYNGÄS, J. Scheduling a triple round robin tournament with minitournaments for the finnish national youth ice hockey league. *Journal of the Operational Research Society*, Taylor & Francis, v. 65, n. 11, p. 1770–1779, 2014.
- PERCY, D. F. A mathematical analysis of badminton scoring systems. *Journal of the Operational Research Society*, Taylor & Francis, v. 60, n. 1, p. 63–71, 2009.
- PINEDO, M. L. *Scheduling*. [S.l.]: Springer, 2012. v. 29.
- PIRLOT, M. General local search methods. *European Journal of Operational Research*, Elsevier, v. 92, n. 3, p. 493–511, 1996.
- RASMUSSEN, R. V. Scheduling a triple round robin tournament for the best danish soccer league. *European Journal of Operational Research*, Elsevier, v. 185, n. 2, p. 795–810, 2008.
- RASMUSSEN, R. V.; TRICK, M. A. Round robin scheduling—a survey. *European Journal of Operational Research*, Elsevier, v. 188, n. 3, p. 617–636, 2008.
- RASMUSSEN, R. V.; TRICK, M. A. The timetable constrained distance minimization problem. *Annals of Operations Research*, Springer, v. 171, p. 45–59, 2009.

RECALDE, D.; TORRES, R.; VACA, P. Scheduling the professional ecuadorian football league by integer programming. *Computers & Operations Research*, Elsevier, v. 40, n. 10, p. 2478–2484, 2013.

RIBEIRO, C. C. Sports scheduling: Problems and applications. *International Transactions in Operational Research*, Wiley Online Library, v. 19, n. 1-2, p. 201–226, 2012.

RIBEIRO, C. C.; URRUTIA, S. Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, Elsevier, v. 179, n. 3, p. 775–787, 2007.

RIBEIRO, C. C.; URRUTIA, S. Scheduling the brazilian soccer tournament with fairness and broadcast objectives. In: SPRINGER. *Practice and Theory of Automated Timetabling VI: 6th International Conference, PATAT 2006 Brno, Czech Republic, August 30–September 1, 2006 Revised Selected Papers 6*. [S.l.], 2007. p. 147–157.

RIBEIRO, C. C.; URRUTIA, S. Scheduling the brazilian soccer tournament: Solution approach and practice. *Interfaces*, INFORMS, v. 42, n. 3, p. 260–272, 2012.

ROSSI, F.; BEEK, P. V.; WALSH, T. Constraint programming. *Foundations of Artificial Intelligence*, Elsevier, v. 3, p. 181–211, 2008.

RUSSELL, R. A.; LEUNG, J. M. Devising a cost effective schedule for a baseball league. *Operations Research*, INFORMS, v. 42, n. 4, p. 614–625, 1994.

SAUR, M. C. et al. Scheduling softball series in the rocky mountain athletic conference. *Interfaces*, INFORMS, v. 42, n. 3, p. 296–309, 2012.

SCHONBERGER, J.; MATTFELD, D. C.; KOPFER, H. Automated timetable generation for rounds of a table-tennis league. In: IEEE. *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*. [S.l.], 2000. v. 1, p. 277–284.

SCHREUDER, J. A. Combinatorial aspects of construction of competition dutch professional football leagues. *Discrete Applied Mathematics*, Elsevier, v. 35, n. 3, p. 301–312, 1992.

TRICK, M. A. Integer and constraint programming approaches for round-robin tournament scheduling. In: SPRINGER. *Practice and Theory of Automated Timetabling IV: 4th International Conference, PATAT 2002, Gent, Belgium, August 21–23, 2002. Selected Revised Papers 4*. [S.l.], 2003. p. 63–77.

TRICK, M. A. Using sports scheduling to teach integer programming. *INFORMS Transactions on Education*, INFORMS, v. 5, n. 1, p. 10–17, 2004.

URRUTIA, S.; RIBEIRO, C. C. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Applied Mathematics*, Elsevier, v. 154, n. 13, p. 1932–1938, 2006.

VOORHIS, T. V. College basketball scheduling with travel swings. *Computers & Industrial Engineering*, Elsevier, v. 48, n. 2, p. 163–172, 2005.

- WERRA, D. D. Scheduling in sports. *Studies on graphs and discrete programming*, Amsterdam, v. 11, p. 381–395, 1981.
- WERRA, D. D. On the multiplication of divisions: The use of graphs for sports scheduling. *Networks*, Wiley Online Library, v. 15, n. 1, p. 125–136, 1985.
- WESTPHAL, S. Scheduling the german basketball league. *Interfaces*, INFORMS, v. 44, n. 5, p. 498–508, 2014.
- WINSTON, W. L. *Operations research: applications and algorithms*. [S.l.]: Cengage Learning, 2022.
- WRIGHT, M. Timetabling county cricket fixtures using a form of tabu search. *Journal of the operational research society*, Taylor & Francis, v. 45, n. 7, p. 758–770, 1994.
- WRIGHT, M. Scheduling fixtures for new zealand cricket. *IMA Journal of Management Mathematics*, Oxford University Press, v. 16, n. 2, p. 99–112, 2005.
- WRIGHT, M. Fifty years of operational research in sport. *Journal of the Operational Research Society*, v. 60, p. 161–168, 2009.
- WRIGHT, M. *Operational research applied to sports*. [S.l.]: Springer, 2016.
- WRIGHT, M.; HIROTSU, N. The professional foul in football: Tactics and deterrents. *Journal of the Operational Research Society*, Taylor & Francis, v. 54, n. 3, p. 213–221, 2003.
- WRIGHT, M. B. Scheduling fixtures for basketball new zealand. *Computers & Operations Research*, Elsevier, v. 33, n. 7, p. 1875–1893, 2006.
- YU, X.; GEN, M. *Introduction to evolutionary algorithms*. [S.l.]: Springer Science & Business Media, 2010.
- ZHANG, H. Generating college conference basketball schedules by a sat solver. In: *Proceedings of the fifth international symposium on the theory and applications of satisfiability testing*. [S.l.: s.n.], 2002. p. 281–291.

APÊNDICE A – TABELAS GERADAS ATRAVÉS DO MODELO E DO ALGORITMO

Neste apêndice, são expostos os resultados fundamentais obtidos através da aplicação dos métodos propostos neste estudo. Aqui, são exibidas as tabelas resultantes da resolução do modelo matemático proposto, bem como aquela que se destaca como a melhor dentre três execuções distintas do algoritmo desenvolvido.

As tabelas fornecem uma visão clara e detalhada das alocações de jogos entre as equipes participantes do campeonato, delineando a distribuição de partidas ao longo do período da competição. Elas representam não apenas o produto final dos métodos empregados, mas também refletem o compromisso em assegurar a justiça competitiva entre os times, conforme estabelecido pelos critérios estipulados no modelo.

A tabela gerada pela solução do modelo matemático demonstra a aplicação direta dos critérios definidos, buscando otimizar a distribuição de jogos de maneira a atender aos requisitos de equilíbrio e justiça competitiva. Por outro lado, a tabela resultante da melhor execução do algoritmo, após três iterações distintas, oferece uma visão do potencial do algoritmo desenvolvido para produzir soluções viáveis.

Esses registros são essenciais para compreender não apenas a aplicabilidade dos métodos propostos, mas também para fornecer um panorama claro das possíveis soluções viáveis para a elaboração das tabelas de jogos em campeonatos futuros. A exposição dessas saídas contribui significativamente para o embasamento teórico-prático deste estudo, demonstrando sua aplicabilidade e relevância no contexto dos Problemas de Geração de Tabelas de Campeonatos.

As Tabelas 25, 26 e 27 trazem, respectivamente, a tabela final gerada para o Campeonato Brasileiro de Futebol Série A - 2023, o cálculo do número de *breaks* e o cálculo dos efeitos de *carry over* obtidos através da solução do modelo. Já as Tabelas 28, 29 e 30

trazem, respectivamente, a tabela final gerada para o Campeonato Brasileiro de Futebol Série A - 2023, o cálculo do número de *breaks* e o cálculo dos efeitos de *carry over* obtidos através da melhor execução do algoritmo.

Tabela 25: Tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.

Times	Rodadas																																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	
América/MG	CRU	RBR	BAH	CUI	PAL	SAN	VAS	INT	SPO	GRE	CAM	FLA	CAP	BOT	COR	FLU	CTB	GOI	FOR	CRU	RBR	BAH	CUI	PAL	SAN	VAS	INT	SPO	GRE	CAM	FLA	CAP	BOT	COR	FLU	CTB	GOI	FOR	
Athletico/PR	GOI	CRU	CTB	SAN	COR	FLU	SPO	RBR	CUI	VAS	BAH	INT	CAM	GRE	BOT	PAL	FLA	FOR	CAM	GOI	CRU	CTB	SAN	COR	FLU	SPO	RBR	CUI	VAS	BAH	INT	CAM	GRE	BOT	PAL	FLA	FOR	CAM	
Atlético/MG	BOT	SPO	GOI	FLA	CRU	BAH	CUI	GRE	CTB	PAL	AMG	VAS	SAN	FOR	FLA	BOT	CRU	FOR	CAP	BOT	SPO	GOI	FLA	CRU	BAH	CUI	GRE	CTB	PAL	AMG	VAS	SAN	FLU	RBR	INT	FOR	CAP	CUI	
Bahia	COR	FLU	AMG	INT	GRE	CAM	SAN	FOR	FLA	BOT	CAP	PAL	RBR	SPO	CTB	GOI	VAS	CRU	CUI	COR	FLU	AMG	INT	GRE	CAM	SAN	FOR	FLA	BOT	CAP	PAL	RBR	SPO	CTB	GOI	VAS	CRU	CUI	
Borafogo	CAM	COR	INT	FLU	GOI	CUI	RBR	CRU	FOR	BAH	SPO	CTB	PAL	AMG	CAP	VAS	SAN	GRE	FLA	CAM	COR	INT	FLU	GOI	CUI	RBR	CRU	FOR	BAH	SPO	CTB	PAL	AMG	CAP	VAS	SAN	GRE	FLA	
Corinthians	BAH	BOT	CRU	RBR	CAP	GRE	FLU	SPO	GOI	SAN	CTB	FOR	FLA	VAS	AMG	CUI	INT	CAM	PAL	BAH	BOT	CRU	RBR	CAP	GRE	FLU	SPO	GOI	SAN	CTB	FOR	FLA	VAS	AMG	CUI	INT	CAM	PAL	
Cortiba	CUI	GRE	CAP	PAL	FLU	CRU	FLA	VAS	CAM	GOI	COR	BOT	FOR	RBR	BAH	SPO	AMG	INT	SAN	CUI	GRE	CAP	PAL	FLU	CRU	FLA	VAS	CAM	GOI	COR	BOT	FOR	RBR	BAH	SPO	AMG	INT	SAN	
Cruzeiro	AMG	CAP	COR	FOR	CAM	CTB	INT	BOT	SAN	FLU	VAS	SPO	CUI	PAL	GRE	FLA	RBR	BAH	GOI	AMG	CAP	COR	FOR	CAM	CTB	INT	BOT	SAN	CAP	FOR	FLU	VAS	SPO	CUI	PAL	GRE	FLA	RBR	BAH
Cuiabá	CTB	FLA	VAS	AMG	RBR	BOT	CAM	SAN	CAP	FOR	GRE	GOI	CRU	INT	FLU	COR	SPO	PAL	BAH	CTB	FLA	VAS	AMG	RBR	BOT	CAM	SAN	CAP	FOR	FLU	GOI	CRU	INT	FLU	COR	SPO	PAL	BAH	
Flamengo	VAS	CUI	GRE	CAM	SPO	GOI	CTB	PAL	BAH	INT	FLU	AMG	COR	FOR	SAN	CRU	CAP	RBR	BOT	VAS	CUI	GRE	CAM	SPO	GOI	RBR	BOT	CAM	SAN	CRU	FLA	AMG	COR	FOR	SAN	CRU	CAP	RBR	BOT
Fluminense	FOR	BAH	RBR	BOT	CTB	CAP	COR	GOI	VAS	CRU	FLA	GRE	INT	CAM	CUI	AMG	PAL	SAN	SPO	FOR	BAH	RBR	BOT	CTB	CAP	COR	GOI	VAS	CRU	FLA	GRE	INT	CAM	CUI	AMG	PAL	SAN	SPO	
Fortaleza	FLU	SAN	SPO	CRU	INT	VAS	GRE	BAH	BOT	CUI	PAL	COR	CTB	FLA	GOI	RBR	CAM	CAP	AMG	FLU	SAN	SPO	CRU	INT	VAS	GRE	BAH	BOT	CUI	PAL	COR	CTB	FLA	GOI	RBR	CAM	CAP	AMG	
Goiás	CAP	INT	CAM	VAS	BOT	FLA	PAL	FLU	COR	CTB	RBR	CUI	SPO	SAN	FOR	BAH	GRE	AMG	CRU	CAP	INT	CAM	VAS	BOT	FLA	PAL	FLU	COR	CTB	RBR	CUI	SPO	SAN	FOR	BAH	GRE	AMG	CRU	
Grêmio	RBR	CTB	FLA	SPO	BAH	COR	FOR	CAM	PAL	AMG	CUI	FLU	VAS	CAP	CRU	SAN	GOI	BOT	INT	RBR	CTB	FLA	SPO	BAH	COR	FOR	CAM	PAL	AMG	CUI	FLU	VAS	CAP	CRU	SAN	GOI	BOT	INT	
Internacional	PAL	GOI	BOT	BAH	FOR	SPO	CRU	AMG	RBR	FLA	SAN	CAP	FLU	CUI	VAS	CAM	COR	CTB	GRE	PAL	GOI	BOT	BAH	FOR	SPO	CRU	AMG	RBR	FLA	SAN	CAP	FLU	CUI	VAS	CAM	COR	CTB	GRE	
Palmeiras	INT	VAS	SAN	CTB	AMG	RBR	GOI	FLA	GRE	CAM	FOR	BAH	BOT	CRU	SPO	CAP	FLU	CUI	COR	INT	VAS	SAN	CTB	AMG	RBR	GOI	FLA	GRE	CAM	FOR	BAH	BOT	CRU	SPO	CAP	FLU	CUI	COR	
RB Bragantino	GRE	AMG	FLU	COR	CUI	PAL	BOT	CAP	INT	SPO	GOI	SAN	BAH	CTB	CAM	FOR	CRU	FLA	VAS	GRE	AMG	FLU	COR	CUI	PAL	BOT	CAP	INT	SPO	GOI	SAN	BAH	CTB	CAM	FOR	CRU	FLA	VAS	
Santos	SPO	FOR	PAL	CAP	VAS	AMG	BAH	CUI	CRU	COR	INT	RBR	CAM	GOI	FLA	GRE	BOT	FLU	CTB	SPO	FOR	PAL	CAP	VAS	AMG	BAH	CUI	CRU	COR	INT	RBR	CAM	GOI	FLA	GRE	BOT	FLU	CTB	
São Paulo	SAN	CAM	FOR	GRE	FLA	INT	CAP	COR	AMG	RBR	BOT	CRU	GOI	BAH	PAL	CTB	CUI	VAS	FLU	SAN	CAM	FOR	GRE	FLA	INT	CAP	COR	AMG	RBR	BOT	CRU	GOI	BAH	PAL	CTB	CUI	VAS	FLU	
Vasco	FLA	PAL	CUI	GOI	SAN	FOR	AMG	CTB	FLU	CAP	CRU	CAM	GRE	COR	INT	BOT	BAH	SPO	RBR	FLA	PAL	CUI	GOI	SAN	FOR	AMG	CTB	FLU	CAP	CRU	CAM	GRE	COR	INT	BOT	BAH	SPO	RBR	

Fonte: Elaborado pelo autor.

Tabela 26: Ocorrência de *breaks* para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.

Times		Rodadas																												Breaks											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		29	30	31	32	33	34	35	36	37	38	
América/MG	f	f	c	c	f	c	c	f	c	c	f	c	c	f	c	c	f	c	c	c	c	c	f	f	c	f	c	f	f	c	f	f	c	f	c	f	f	c	f	f	15
	c	c	c	f	c	c	c	c	c	f	c	c	f	f	c	c	c	c	c	f	c	f	f	c	c	f	f	f	c	f	f	c	f	f	c	f	f	c	f	20	
	f	c	f	f	f	c	f	c	c	f	c	c	c	c	c	f	c	c	c	f	c	f	c	c	c	c	f	c	c	f	c	c	f	c	f	f	c	f	c	16	
	f	c	f	f	c	c	f	c	f	f	c	f	c	c	c	f	f	c	f	f	f	f	f	f	f	f	f	c	f	c	f	c	c	f	c	f	c	c	c	16	
Botafogo	c	c	c	c	c	c	f	c	f	c	f	c	c	c	c	f	f	c	f	f	f	f	f	f	f	f	c	c	f	c	f	c	c	c	f	c	f	c	c	19	
	c	f	f	f	f	c	c	c	f	f	c	c	c	c	c	f	f	f	f	f	f	c	c	c	c	f	f	c	f	c	c	c	f	c	c	f	c	c	c	23	
Coritiba	f	c	f	f	f	c	f	c	f	f	f	f	f	f	c	c	f	c	c	c	c	c	c	c	c	f	c	c	f	c	c	f	c	f	c	f	c	f	f	19	
Cruzeiro	c	f	c	c	c	f	c	f	f	c	f	f	f	f	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	f	f	c	20		
	c	c	c	f	c	f	c	c	c	f	c	c	f	f	c	f	f	f	f	c	f	f	c	c	c	f	c	c	c	c	c	c	c	c	c	c	c	f	12		
Flamengo	c	f	c	c	f	c	c	f	c	c	f	f	f	c	c	f	c	f	f	c	c	f	c	c	c	f	c	c	f	c	c	c	c	c	c	c	c	c	f	12	
	f	f	c	f	c	f	f	c	f	f	c	c	f	c	c	f	f	f	f	c	c	c	f	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	f	19	
Fortaleza	c	c	f	f	c	f	c	f	c	c	f	f	c	c	f	c	f	f	c	f	f	f	c	c	c	f	c	c	f	c	c	c	c	c	f	c	c	c	c	11	
	f	c	c	f	f	f	c	f	c	c	f	c	f	f	f	f	f	f	f	f	c	f	f	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	20	
Grêmio	c	f	c	f	f	f	f	f	f	f	f	c	c	c	c	c	f	c	c	c	c	c	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	c	c	24	
	c	f	c	f	f	f	c	f	c	f	c	c	c	c	c	c	f	c	f	f	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	c	15		
Palmeiras	f	c	c	c	c	c	f	c	c	c	c	c	f	c	c	c	f	c	c	c	c	f	f	f	f	c	f	c	c	c	c	c	c	c	c	c	c	c	c	23	
RB Bragantino	f	c	f	c	f	f	c	f	c	f	c	f	f	c	c	c	c	f	c	c	c	c	c	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	c	11	
Santos	c	f	c	c	f	f	c	f	c	c	f	c	f	f	c	c	c	c	c	f	c	c	c	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	c	11	
São Paulo	f	f	c	f	c	c	f	f	c	c	c	c	c	f	c	c	f	c	f	f	c	f	c	f	f	c	c	f	c	c	c	c	c	c	f	f	c	c	c	20	
Vasco	f	f	c	c	c	c	f	f	c	f	c	f	f	f	c	f	f	c	f	c	c	c	c	c	f	c	c	c	c	c	c	c	c	c	c	c	c	c	c	16	
Total																																							342		

Fonte: Elaborado pelo autor.

Tabela 27: Valores de *carry-over* para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo modelo.

Times		Times																			SCOQ	
	América/MG	Athético/PR	Atlético/MG	Bahia	Botafogo	Corinthians	Coritiba	Cruzeiro	Cuiabá	Flamengo	Fluminense	Fortaleza	Goiás	Grêmio	Internacional	Palmeiras	RB Bragantino	Santos	São Paulo	Vasco		
América/MG	-	4	0	2	0	2	2	2	4	0	4	0	0	2	4	2	8	0	0	2	156	
Athético/PR	2	-	0	0	4	6	0	4	0	0	4	2	0	2	4	4	2	0	0	4	148	
Atlético/MG	0	2	-	0	0	4	2	0	2	2	0	6	6	2	0	4	0	4	2	2	148	
Bahia	0	0	0	-	6	2	4	0	6	0	0	2	2	2	4	2	2	0	6	0	164	
Botafogo	0	4	2	4	-	2	2	6	2	2	2	2	0	0	2	2	0	2	2	2	116	
Corinthians	2	2	0	0	2	-	6	0	2	0	6	6	2	0	8	0	0	0	2	0	196	
Coritiba	2	2	2	0	0	0	-	0	2	6	2	2	4	2	2	6	2	2	2	0	132	
Cruzeiro	2	4	2	2	0	2	2	-	2	6	0	2	2	0	4	0	4	2	2	0	124	
Cuiabá	2	0	0	0	0	4	0	2	-	0	2	0	2	6	2	8	2	0	2	6	180	
Flamengo	0	2	2	0	2	0	0	2	0	-	0	2	2	6	2	4	2	2	2	8	156	
Fluminense	4	2	0	0	0	6	4	2	4	0	-	0	2	0	0	0	2	4	4	4	148	
Fortaleza	2	0	6	8	0	2	0	4	0	4	0	-	0	4	0	2	2	2	2	0	172	
Goiás	2	0	0	2	4	2	2	4	2	6	0	2	-	0	0	0	2	6	0	4	148	
Grêmio	4	2	8	2	4	2	2	0	0	6	2	0	2	-	2	2	0	0	0	0	164	
Internacional	2	2	6	0	4	0	0	0	0	0	6	2	0	2	-	0	4	2	4	4	156	
Palmeiras	6	2	0	6	2	2	2	0	2	2	4	0	2	2	0	-	2	4	0	0	140	
RB Bragantino	0	2	4	6	8	0	2	2	4	4	0	0	2	0	2	0	-	0	2	0	172	
Santos	0	4	2	2	0	2	4	2	2	0	4	6	2	2	0	0	0	-	4	2	132	
São Paulo	2	2	0	2	0	0	4	4	2	0	0	4	8	2	0	2	4	2	-	0	156	
Vasco	6	2	4	2	2	0	0	4	2	0	2	0	0	4	2	0	0	6	2	-	148	
Total																					3056	

Fonte: Elaborado pelo autor.

Tabela 28: Tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.

Times	Rodadas																																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38		
América/MG	VAS	CRU	GRE	FLU	CTB	FLA	COR	BAH	CAP	INT	SPO	CUI	GOI	SAN	CAM	PAL	FOR	BOT	RBR	VAS	CRU	GRE	FLU	CTB	FLA	COR	BAH	CAP	INT	SPO	CUI	SAN	CAM	PAL	FOR	BOT	RBR			
Athletico/PR	CTB	GOI	RBR	CAM	FLU	BOT	BAH	CRU	AMG	GRE	SAN	VAS	INT	PAL	FOR	FLA	COR	CUI	SPO	CTB	GOI	RBR	CAM	FLU	BOT	BAH	CRU	AMG	GRE	SAN	VAS	INT	PAL	FOR	FLA	COR	CUI	SPO		
Atlético/MG	INT	FLA	VAS	CAP	GRE	BAH	SAN	PAL	CTB	CUI	RBR	GOI	BOT	FOR	AMG	COR	SPO	CRU	FLU	INT	FLA	VAS	CAP	GRE	BAH	SAN	PAL	CTB	CUI	RBR	GOI	BOT	FOR	AMG	COR	SPO	CRU	FLU		
Bahia	RBR	SPO	FLA	GOI	CRU	CAM	CAP	AMG	BOT	PAL	VAS	COR	SAN	CTB	INT	FLU	GRE	CUI	AMG	CRU	RBR	SPO	FLA	GOI	CAM	CAP	AMG	BOT	PAL	VAS	COR	SAN	CTB	INT	FLU	GRE	CUI	AMG	CRU	
Borafogo	SAN	FLU	COR	FLA	PAL	CAP	VAS	CTB	BAH	FOR	GOI	SPO	CAM	INT	RBR	GRE	CUI	AMG	CRU	SAN	FLU	COR	FLA	PAL	CAP	VAS	CTB	BAH	FOR	GOI	SPO	CAM	INT	RBR	GRE	CUI	AMG	CRU		
Corinthians	CRU	SAN	BOT	INT	GOI	FOR	AMG	SPO	CUI	FLA	CTB	BAH	PAL	FLU	GRE	CAM	CAP	RBR	VAS	CRU	SAN	BOT	INT	GOI	FOR	AMG	SPO	CUI	FLA	CTB	BAH	PAL	FLU	GRE	CAM	CAP	RBR	VAS		
Coritiba	CAP	CUI	FLU	SPO	AMG	GOI	FLA	BOT	CAM	RBR	COR	GRE	VAS	BAH	SAN	CRU	PAL	INT	FOR	CAP	CUI	FLU	SPO	AMG	GOI	FLA	BOT	CAM	RBR	COR	GRE	VAS	BAH	SAN	CRU	PAL	INT	FOR		
Cruzeiro	COR	AMG	PAL	RBR	BAH	FLU	GRE	CAP	FLA	SAN	CUI	FOR	SPO	VAS	GOI	CTB	INT	CAM	BOT	COR	AMG	PAL	RBR	BAH	FLU	GRE	CAP	FLA	SAN	CUI	FOR	SPO	VAS	GOI	CTB	INT	CAM	BOT		
Cuiabá	SPO	CTB	FOR	PAL	SAN	GRE	RBR	VAS	COR	CAM	CRU	AMG	FLU	GOI	FLA	INT	BOT	CAP	BAH	SPO	CTB	FOR	PAL	SAN	GRE	RBR	VAS	COR	CAM	CRU	AMG	FLU	GOI	FLA	INT	BOT	CAP	BAH		
Flamengo	PAL	CAM	BAH	BOT	RBR	AMG	CTB	FLU	CRU	COR	INT	SAN	FOR	GRE	CUI	CAP	VAS	SPO	GOI	PAL	CAM	BAH	BOT	RBR	AMG	CTB	FLU	CRU	COR	INT	SAN	FOR	GRE	CUI	CAP	VAS	SPO	GOI		
Fluminense	GRE	BOT	CTB	AMG	CAP	CRU	PAL	FLA	RBR	VAS	FOR	INT	CUI	COR	SPO	BAH	SAN	GOI	CAM	GRE	BOT	CTB	AMG	CAP	CRU	PAL	FLA	RBR	VAS	FOR	INT	CUI	COR	SPO	BAH	SAN	GOI	CAM		
Fortaleza	GOI	PAL	CUI	GRE	INT	COR	SPO	SAN	VAS	BOT	FLU	CRU	FLA	CAM	CAP	RBR	AMG	BAH	CTB	GOI	PAL	CUI	GRE	INT	COR	SPO	SAN	VAS	BOT	FLU	CRU	FLA	CAM	CAP	RBR	AMG	BAH			
Goiás	FOR	CAP	SAN	BAH	COR	CTB	INT	GRE	PAL	SPO	BOT	CAM	AMG	CUI	CRU	VAS	RBR	FLU	FLA	FOR	CAP	SAN	BAH	COR	CTB	INT	GRE	PAL	SPO	BOT	FLU	CRU	CAM	AMG	CUI	CRU	VAS	RBR		
Grêmio	FLU	INT	AMG	FOR	CAM	CUI	CRU	GOI	SPO	CAP	PAL	CTB	RBR	FLA	COR	BOT	BAH	VAS	SAN	FLU	INT	AMG	FOR	CAM	CUI	CRU	GOI	SPO	CAP	PAL	CTB	RBR	FLA	COR	BOT	BAH	VAS	SAN		
Internacional	CAM	GRE	SPO	COR	FOR	VAS	GOI	RBR	SAN	AMG	FLA	FLU	CAP	BOT	BAH	CUI	CTB	SAN	INT	FLA	FOR	GRE	SPO	COR	FOR	VAS	GOI	RBR	SAN	AMG	FLA	FLU	CAP	RBR	FLA	CUI	CRU	CTB	PAL	
Palmeiras	FLA	FOR	CRU	CUI	BOT	SPO	FLU	CAM	GOI	BAH	GRE	RBR	COR	CAP	VAS	AMG	CTB	SAN	INT	FLA	FOR	CRU	CUI	BOT	SPO	FLU	CAM	GOI	RBR	SAN	AMG	FLA	FLU	CAP	RBR	FLA	CUI	CRU	CTB	PAL
RB Bragantino	BAH	VAS	CAP	CRU	FLA	SAN	CUI	INT	FLU	CTB	CAM	PAL	GRE	SPO	BOT	FOR	GOI	COR	AMG	BAH	VAS	CAP	CRU	FLA	SAN	CUI	INT	FLU	CTB	CAM	PAL	GRE	SPO	BOT	FOR	GOI	COR	AMG		
Santos	BOT	COR	GOI	VAS	CUI	RBR	CAM	FOR	INT	CRU	CAP	FLA	BAH	AMG	CTB	SPO	FLU	PAL	GRE	BOT	COR	GOI	VAS	CUI	RBR	CAM	FOR	INT	CRU	CAP	FLA	BAH	AMG	CTB	SPO	FLU	PAL	GRE		
São Paulo	CUI	BAH	INT	CTB	VAS	PAL	FOR	COR	GRE	GOI	AMG	BOT	CRU	RBR	FLU	SAN	CAM	FLA	CAP	CUI	BAH	INT	CTB	VAS	PAL	FOR	COR	GRE	GOI	AMG	BOT	CRU	RBR	FLU	SAN	CAM	FLA	CAP		
Vasco	AMG	RBR	CAM	SAN	SPO	INT	BOT	CUI	FOR	FLU	BAH	CAP	CTB	CRU	PAL	GOI	FLA	GRE	COR	AMG	RBR	CAM	SAN	SPO	INT	BOT	CUI	FOR	FLU	BAH	CAP	CTB	CRU	PAL	GOI	FLA	CAP	COR		

Fonte: Elaborado pelo autor.

Tabela 29: Ocorrência de *breaks* para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.

Times	Rodadas																																Breaks								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		33	34	35	36	37	38		
América/MG	c	f	c	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	f	f	c	f	c	c	c	f	c	c	f	c	f	c	f	c	f	c	f	c	f	c	7
Athletico/PR	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	f	c	c	f	f	c	f	c	c	f	c	f	c	c	f	c	f	f	c	f	c	f	f	c	8	
Atlético/MG	f	c	f	c	c	f	c	f	c	f	c	f	c	f	f	c	f	c	c	c	c	f	f	f	f	c	f	c	c	f	c	f	f	c	c	f	c	f	f	11	
Bahia	f	f	f	c	f	c	f	c	f	f	c	c	c	f	f	c	f	c	f	c	c	c	f	f	c	f	c	f	c	f	f	f	c	f	c	f	c	f	c	12	
Botafogo	c	f	f	c	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	c	f	c	f	c	f	c	f	c	f	c	f	c	f	c	f	c	f	c	f	4	
Corinthians	c	f	c	f	f	c	c	f	c	f	c	f	c	f	c	f	f	c	c	c	f	c	f	c	f	c	f	f	c	f	c	f	c	f	c	c	f	f	8		
Coritiba	c	f	c	f	c	c	f	c	f	f	c	c	f	c	f	c	f	f	c	c	f	c	f	c	f	c	f	c	f	c	c	f	c	f	c	f	c	c	f	8	
Cruzeiro	f	c	f	c	c	f	f	c	f	c	f	c	f	c	f	c	f	f	c	f	c	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	f	c	f	8	
Cuiabá	c	c	f	c	c	c	f	c	f	f	c	c	c	f	c	c	c	f	f	c	f	f	c	f	f	f	c	f	c	f	f	c	f	f	c	f	c	c	f	16	
Flamengo	f	f	c	f	f	c	f	c	c	c	c	f	c	c	c	f	c	f	f	c	c	f	c	c	c	f	c	f	c	f	f	c	f	f	c	f	c	f	c	12	
Fluminense	f	c	f	c	f	c	f	c	c	f	c	f	f	c	c	f	f	c	f	c	f	c	f	f	c	f	c	f	f	c	c	f	c	c	f	c	f	c	c	8	
Fortaleza	f	c	c	f	c	f	c	c	f	c	f	f	f	c	c	f	f	c	f	f	c	f	c	f	c	f	c	f	f	c	c	c	f	c	f	c	f	c	c	12	
Goiás	c	f	c	f	c	f	c	f	c	f	f	f	f	c	c	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	c	c	c	c	f	c	c	f	c	8	
Grêmio	c	c	f	c	f	f	c	c	f	c	f	c	f	c	f	f	c	c	f	c	f	f	c	f	c	c	f	f	c	c	f	c	f	c	c	f	f	c	f	12	
Internacional	c	f	c	c	f	c	f	f	c	f	c	f	c	f	c	c	f	c	f	f	c	f	f	c	f	f	c	c	f	c	c	f	c	f	c	f	c	f	c	11	
Palmeiras	c	f	c	f	c	f	c	c	f	c	f	c	f	c	c	f	c	f	c	f	c	f	c	f	c	f	f	c	f	c	f	c	f	c	f	c	f	c	f	4	
RB Bragantino	c	f	c	f	c	f	c	c	f	c	c	f	f	c	f	c	c	f	c	f	c	f	c	f	c	f	f	c	f	c	c	f	c	c	f	c	f	f	c	8	
Santos	f	c	f	c	f	c	f	f	c	f	c	c	f	c	c	c	f	c	c	f	c	f	c	f	c	f	c	f	c	f	c	f	f	c	f	c	f	f	c	8	
São Paulo	f	c	f	c	f	c	f	c	f	c	f	c	c	f	f	c	c	f	c	c	c	f	c	f	c	f	c	f	c	f	c	f	f	c	c	f	c	f	c	f	7
Vasco	f	c	c	f	c	f	c	f	c	c	f	f	c	f	c	f	f	c	c	f	c	f	c	f	c	f	c	f	f	c	f	c	c	f	c	f	c	f	c	8	
Total																																						180			

Fonte: Elaborado pelo autor.

Tabela 30: Valores de *carry-over* para a tabela do Campeonato Brasileiro de Futebol - Série A 2023 gerada pelo algoritmo.

Times		Times																		SECOQ	
	América/MG	Athletico/PR	Atlético/MG	Bahia	Botafogo	Corinthians	Coritiba	Cruzeiro	Cuiabá	Flamengo	Fluminense	Fortaleza	Goiás	Grêmio	Internacional	Palmeiras	RB Bragantino	Santos	São Paulo	Vasco	
América/MG	-	2	0	4	4	2	6	2	2	2	2	2	2	2	0	2	2	0	2	0	
Athletico/PR		-	0	2	2	0	2	4	4	4	0	0	0	2	2	2	4	2	0	6	
Atlético/MG	2	6	-	2	2	0	0	2	2	2	2	2	2	4	2	4	2	2	0	0	
Bahia	2	4	0	-	2	2	2	2	2	0	2	2	0	2	2	2	0	6	2	4	
Botafogo	0	2	4	6	-	4	2	2	2	0	2	4	0	0	2	2	4	0	2	0	
Corinthians	6	2	2	2	2	-	2	0	2	2	0	2	2	4	2	0	2	2	6	0	
Coritiba	2	0	2	4	0	0	-	2	2	2	2	2	4	0	6	2	2	2	2	2	
Cruzeiro	4	2	2	0	0	2	2	-	2	4	2	0	2	2	0	6	2	4	0	2	
Cuiabá	2	2	0	2	2	2	0	6	-	2	2	4	2	2	2	0	6	0	2	0	
Flamengo	0	2	2	2	2	6	2	0	0	-	2	4	2	2	2	2	2	4	0	2	
Fluminense	0	2	2	2	2	2	4	4	0	2	-	0	2	6	4	2	0	2	2	0	
Fortaleza	4	4	2	0	2	2	0	2	2	2	2	-	4	2	4	2	0	0	2	2	
Goiás	2	0	2	2	2	2	2	2	0	6	0	2	-	0	0	4	4	2	4	2	
Grêmio	0	2	2	2	4	2	0	0	4	0	2	2	2	-	2	2	4	2	4	2	
Internacional	2	0	2	0	4	2	2	2	2	4	4	2	2	2	-	2	2	2	2	0	
Palmeiras	0	2	4	0	0	0	4	0	2	2	2	6	2	4	2	-	2	2	2	2	
RB Bragantino	4	0	6	2	0	4	0	0	0	2	4	0	2	2	0	0	-	2	2	8	
Santos	2	0	4	2	2	0	2	2	4	0	4	2	2	2	2	2	0	-	2	4	
São Paulo	2	2	2	2	4	2	4	2	4	2	4	0	2	0	2	0	0	2	-	2	
Vasco	2	4	0	2	2	4	2	4	2	0	0	2	4	0	2	2	2	2	2	-	
																			Total	2416	

Fonte: Elaborado pelo autor.

APÊNDICE B – EXECUÇÕES DO ALGORITMO

Neste apêndice, é fornecido um detalhamento minucioso das três execuções do algoritmo aplicadas a cada uma das instâncias criadas para este estudo. A tabela apresentada contém informações cruciais que descrevem o processo de aprimoramento das soluções ao longo das iterações do algoritmo.

Cada linha da tabela corresponde a uma iteração do algoritmo e oferece dados específicos sobre a melhoria alcançada em relação à solução anterior. A coluna Iter indica o número da iteração em que ocorreu a melhoria, enquanto a coluna VE representa a Vizinhaça Explorada que propiciou aprimoramento na solução durante essa iteração.

As colunas *brk*, *coe* e *z* fornecem o número de *breaks* identificados, os efeitos de *carry over* e os valores da função objetivo em cada iteração, respectivamente. Esses parâmetros são fundamentais para compreender a evolução e o desempenho do algoritmo ao longo do processo de iterativo.

Além disso, a coluna Melhoria descreve a melhoria relativa em comparação com a solução obtida na iteração anterior. Essa informação quantifica e demonstra de forma clara o progresso alcançado a cada passo do algoritmo, proporcionando uma visão detalhada do processo de refinamento das soluções ao longo das iterações.

Tabela 31: Análise de vizinhanças para $n = 6$ - Execução 1.

Iter	VE ¹	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	24	240	1,375	-
2	GR	22	240	1,3125	-4,54%
33	MHAS	14	240	1,0625	-19,04%

Fonte: Elaborado pelo autor.

¹MHAS: Alternância de mando de um apartida; THAS: Alternância de mandos de campo para um time; RS: Alternância de rodadas; PRS: Alternância parcial de rodadas; GR: Rotação de jogos.

Tabela 32: Análise de vizinhanças para $n = 6$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	22	240	1,3125	-
1	MHAS	18	240	1,1875	-9,52%
14	THAS	16	240	1,125	-5,26%
47	MHAS	12	240	1	-11,11%

Fonte: Elaborado pelo autor.

Tabela 33: Análise de vizinhanças para $n = 6$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	18	240	1,1875	-
14	THAS	12	240	1	-15,79%

Fonte: Elaborado pelo autor.

Tabela 34: Análise de vizinhanças para $n = 8$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	32	784	1,194	-	26	PRS	44	320	0,533	-9,93%
4	RS	40	480	0,762	-36,15%	27	PRS	36	320	0,421	-20,86%
6	THAS	36	480	0,707	-7,28%	31	MHAS	32	320	0,366	-13,18%
10	MHAS	32	480	0,652	-7,86%	91	MHAS	30	320	0,338	-7,59%
21	GR	40	384	0,591	-9,26%	96	MHAS	26	240	0,283	-16,43%

Fonte: Elaborado pelo autor.

Tabela 35: Análise de vizinhanças para $n = 8$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	30	784	1,167	-	36	PRS	40	336	0,506	-5,21%
13	RS	38	512	0,792	-32,10%	38	GR	38	320	0,449	-11,15%
23	MHAS	34	512	0,737	-7,01%	41	RS	36	320	0,421	-6,16%
25	RS	46	384	0,675	-8,41%	51	MHAS	34	320	0,393	-6,59%
27	RS	38	384	0,563	-16,47%	57	MHAS	30	320	0,338	-14,11%
31	GR	38	384	0,563	-0,00%	69	MHAS	26	320	0,282	-16,43%
33	RS	38	336	0,533	-5,35%						

Fonte: Elaborado pelo autor.

Tabela 36: Análise de vizinhanças para $n = 8$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	28	784	1,139	-	43	RS	44	320	0,533	-0,29%
1	RS	44	432	0,733	-35,67%	54	GR	40	320	0,477	-10,43%
7	GR	36	448	0,650	-11,27%	65	GR	36	320	0,421	-11,65%
9	GR	44	384	0,647	-0,49%	68	MHAS	36	320	0,421	-0,00%
12	PRS	46	352	0,617	-4,54%	79	THAS	32	320	0,366	-13,18%
20	THAS	44	352	0,590	-4,50%	102	GR	32	320	0,366	-0,00%
36	PRS	40	352	0,534	-9,42%	155	MHAS	24	320	0,255	-30,37%

Fonte: Elaborado pelo autor.

Tabela 37: Análise de vizinhanças para $n = 10$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	54	1872	1,234	-	23	MHAS	54	648	0,425	-12,82%
2	PRS	64	1552	1,101	-10,81%	40	RS	50	608	0,367	-13,58%
3	RS	72	824	0,682	-38,06%	57	PRS	50	568	0,341	-7,21%
9	GR	68	720	0,582	-14,67%	61	MHAS	46	568	0,309	-9,17%
11	MHAS	66	720	0,566	-2,69%	70	GR	44	568	0,294	-5,05%
15	GR	66	672	0,534	-5,60%	109	MHAS	40	568	0,263	-10,64%
17	MHAS	62	672	0,503	-5,85%	181	GR	34	600	0,237	-9,79%
19	PRS	62	648	0,487	-3,15%	184	PRS	34	584	0,227	-4,47%

Fonte: Elaborado pelo autor.

Tabela 38: Análise de vizinhanças para $n = 10$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	50	1872	1,203	-	78	RS	54	656	0,430	-3,34%
1	RS	70	960	0,756	-37,15%	79	PRS	54	640	0,420	-2,46%
3	RS	86	616	0,654	-13,56%	95	THAS	52	640	0,404	-3,72%
7	RS	74	624	0,565	-13,53%	99	MHAS	48	640	0,373	-7,74%
10	RS	66	608	0,492	-0,10%	139	MHAS	44	640	0,341	-8,39%
11	PRS	70	560	0,492	-0,10%	166	GR	46	608	0,336	-1,62%
12	PRS	66	584	0,476	-3,13%	187	MHAS	44	608	0,320	-4,65%
34	MHAS	62	584	0,445	-6,56%	198	PRS	40	576	0,268	-16,37%

Fonte: Elaborado pelo autor.

Tabela 39: Análise de vizinhanças para $n = 10$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	28	784	1,139	-	43	RS	44	320	0,533	-0,29%
1	RS	44	432	0,733	-35,67%	54	GR	40	320	0,477	-10,43%
7	GR	36	448	0,650	-11,27%	65	GR	36	320	0,421	-11,65%
9	GR	44	384	0,647	-0,49%	68	MHAS	36	320	0,421	-0,00%
12	PRS	46	352	0,617	-4,54%	79	THAS	32	320	0,366	-13,18%
20	THAS	44	352	0,590	-4,50%	102	GR	32	320	0,366	-0,00%
36	PRS	40	352	0,534	-9,42%	155	MHAS	24	320	0,255	-30,37%

Fonte: Elaborado pelo autor.

Tabela 40: Análise de vizinhanças para $n = 12$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	60	3696	1,150	-	42	MHAS	92	960	0,446	-8,22%
1	RS	84	1968	0,725	-37,00%	44	RS	92	936	0,439	-1,70%
3	RS	100	1232	0,572	-21,02%	78	MHAS	88	936	0,419	-4,56%
4	GR	100	1232	0,572	-0,00%	122	MHAS	84	936	0,399	-4,78%
6	RS	104	1144	0,564	-1,36%	148	MHAS	80	936	0,379	-5,02%
7	RS	104	960	0,506	-10,29%	275	MHAS	72	936	0,339	-10,56%
17	THAS	100	960	0,486	-3,95%	278	MHAS	68	936	0,319	-5,90%
20	MHAS	100	960	0,486	-0,00%						

Fonte: Elaborado pelo autor.

Tabela 41: Análise de vizinhanças para $n = 12$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	66	3696	1,180	-	42	THAS	102	1000	0,509	-5,57%
9	RS	90	2000	0,765	-35,20%	48	MHAS	94	1000	0,469	-7,86%
11	RS	102	1488	0,663	-13,29%	54	THAS	90	1000	0,449	-4,26%
12	RS	114	1016	0,574	-13,42%	109	RS	74	920	0,344	-23,44%
30	RS	112	100	0,559	-2,62%	142	MHAS	74	920	0,344	-0,00%
33	THAS	112	100	0,559	-0,00%	149	MHAS	70	920	0,324	-5,82%
36	MHAS	108	100	0,539	-3,58%	151	THAS	68	920	0,314	-3,09%

Fonte: Elaborado pelo autor.

Tabela 42: Análise de vizinhanças para $n = 12$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	74	3696	1,220	-	81	MHAS	82	1040	0,422	-4,53%
17	MHAS	70	3696	1,200	-1,64%	82	GR	76	1040	0,392	-7,12%
22	RS	106	1968	0,835	-30,45%	98	MHAS	72	1040	0,372	-5,11%
25	RS	114	1680	0,784	-6,10%	183	RS	72	1008	0,362	-2,72%
38	MHAS	112	1680	0,774	-1,28%	198	RS	80	800	0,336	-7,10%
43	MHAS	108	1680	0,754	-2,59%	215	GR	76	800	0,316	-5,95%
45	MHAS	100	1680	0,714	-5,31%	252	MHAS	72	800	0,296	-6,33%
55	RS	104	1192	0,580	-18,78%	257	MHAS	68	800	0,276	-6,76%
67	RS	104	1040	0,532	-8,28%	258	MHAS	64	800	0,256	-7,25%
72	THAS	86	1040	0,442	-16,93%						

Fonte: Elaborado pelo autor.

Tabela 43: Análise de vizinhanças para $n = 14$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	82	6448	1,160	-	91	MHAS	126	1176	0,391	-3,43%
4	RS	106	4304	0,868	-25,13%	98	MHAS	122	1176	0,377	-3,55%
12	RS	146	2256	0,649	-25,24%	164	THAS	114	1176	0,349	-7,37%
19	RS	156	1968	0,633	-2,41%	178	MHAS	110	1176	0,335	-3,98%
24	RS	138	2304	0,630	-0,59%	213	MHAS	106	1176	0,321	-4,14%
25	RS	142	1888	0,571	-9,34%	270	MHAS	102	1176	0,307	-4,32%
27	RS	148	1392	0,505	-11,54%	287	MHAS	98	1176	0,294	-4,52%
32	RS	138	1176	0,432	-14,35%	405	MHAS	94	1176	0,280	-4,73%
36	THAS	134	1176	0,419	-3,21%	482	MHAS	90	1176	0,266	-4,97%
48	THAS	130	1176	0,405	-3,32%	663	MHAS	86	1176	0,252	-5,22%

Fonte: Elaborado pelo autor.

Tabela 44: Análise de vizinhanças para $n = 14$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	104	6448	1,236	-	159	MHAS	132	1120	0,402	-3,34%
1	RS	110	3584	0,756	-38,82%	201	THAS	128	1120	0,388	-3,46%
4	RS	146	1848	0,578	-23,60%	235	MHAS	124	1120	0,374	-3,58%
9	RS	150	1352	0,505	-12,60%	258	THAS	124	1120	0,374	-0,00%
10	MHAS	146	1352	0,491	-2,75%	283	MHAS	116	1120	0,346	-7,43%
12	MHAS	146	1352	0,491	-0,00%	307	MHAS	112	1120	0,332	-4,01%
25	THAS	142	1352	0,477	-2,83%	312	MHAS	112	1120	0,332	-0,00%
43	MHAS	138	1352	0,463	-2,91%	347	MHAS	108	1120	0,319	-4,18%
46	GR	138	1352	0,463	-0,00%	387	MHAS	104	1120	0,305	-4,36%
54	THAS	128	1352	0,429	-7,50%	440	MHAS	100	1120	0,291	-4,56%
64	RS	124	1384	0,420	-1,94%	443	MHAS	100	1120	0,291	-0,00%
120	RS	136	1120	0,416	-1,07%	451	MHAS	96	1120	0,277	-4,78%

Fonte: Elaborado pelo autor.

Tabela 45: Análise de vizinhanças para $n = 14$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	z	Melhoria
0	-	88	6448	1,181	-	248	RS	118	1296	0,384	-2,14%
5	RS	112	4304	0,889	-24,69%	264	MHAS	114	1296	0,370	-3,62%
7	RS	136	2296	0,621	-30,11%	267	RS	120	1144	0,364	-1,55%
17	RS	152	1360	0,513	-17,39%	280	MHAS	116	1144	0,351	-3,81%
20	RS	136	1368	0,459	-10,55%	284	MHAS	112	1144	0,337	-3,96%
35	MHAS	132	1368	0,445	-3,03%	298	MHAS	112	1144	0,337	-0,00%
37	MHAS	132	1368	0,445	-0,00%	357	MHAS	108	1144	0,323	-4,13%
65	MHAS	128	1368	0,431	-3,12%	373	MHAS	106	1144	0,316	-2,15%
77	RS	128	1344	0,427	-0,97%	459	MHAS	102	1144	0,302	-4,40%
82	GR	124	1344	0,413	-3,25%	512	MHAS	98	1144	0,288	-4,60%
107	MHAS	122	1344	0,406	-1,68%	551	MHAS	94	1144	0,274	-4,82%
247	MHAS	118	1344	0,392	-3,42%	652	MHAS	92	1144	0,267	-2,53%

Fonte: Elaborado pelo autor.

Tabela 46: Análise de vizinhanças para $n = 16$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	94	10320	1,133	-	124	GR	136	1896	0,340	-0,28%
1	RS	118	6112	0,744	-34,29%	131	MHAS	132	1840	0,324	-4,76%
3	PRS	118	5688	0,699	-6,09%	166	MHAS	128	1840	0,313	-3,15%
4	GR	150	4336	0,636	-8,99%	185	MHAS	124	1840	0,303	-3,26%
6	GR	142	4336	0,616	-3,21%	206	MHAS	120	1904	0,300	-1,11%
7	PRS	142	4056	0,586	-4,86%	208	MHAS	116	1904	0,290	-3,40%
8	RS	146	2920	0,475	-18,97%	212	MHAS	112	1904	0,279	-3,52%
9	MHAS	142	2920	0,465	-2,15%	222	MHAS	112	1904	0,279	-0,00%
12	RS	146	2264	0,405	-12,89%	241	PRS	108	1864	0,265	-5,18%
13	PRS	138	2216	0,379	-6,31%	391	PRS	104	1944	0,263	-0,63%
20	PRS	138	2160	0,373	-1,58%	409	PRS	104	1904	0,259	-1,62%
23	MHAS	134	2160	0,363	-2,73%	418	MHAS	100	1904	0,249	-3,94%
28	MHAS	130	2160	0,353	-2,81%	458	PRS	104	1792	0,247	-0,71%
54	PRS	132	2096	0,351	-0,49%	521	PRS	104	1776	0,245	-0,69%
63	MHAS	130	2096	0,346	-1,45%	582	MHAS	104	1776	0,245	-0,00%
81	MHAS	128	2096	0,341	-1,48%	774	GR	104	1752	0,243	-1,05%

Fonte: Elaborado pelo autor.

Tabela 47: Análise de vizinhanças para $n = 16$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	104	10320	1,158	-	51	RS	158	1776	0,383	-4,28%
1	PRS	112	9619	1,103	-4,73%	72	PRS	154	1720	0,367	-4,23%
2	PRS	120	8912	1,049	-4,97%	81	MHAS	150	1720	0,357	-2,78%
4	PRS	136	8272	1,021	-2,63%	91	MHAS	146	1720	0,347	-2,86%
6	RS	152	5312	0,746	-26,98%	102	PRS	146	1640	0,338	-2,47%
8	RS	156	4200	0,637	-14,57%	128	MHAS	142	1640	0,328	-3,02%
12	RS	164	2936	0,522	-18,00%	138	PRS	142	1632	0,327	-0,26%
15	RS	164	2648	0,492	-5,89%	157	MHAS	138	1632	0,317	-3,12%
17	MHAS	160	2648	0,481	-2,08%	179	PRS	134	1688	0,312	-1,33%
18	GR	160	2336	0,448	-6,92%	186	PRS	126	1600	0,283	-9,54%
33	RS	164	2104	0,433	-3,25%	242	GR	122	1600	0,272	-3,61%
34	PRS	164	1984	0,421	-2,96%	336	MHAS	118	1600	0,262	-3,75%
36	MHAS	156	1984	0,400	-4,85%	401	MHAS	118	1600	0,262	-0,00%

Fonte: Elaborado pelo autor.

Tabela 48: Análise de vizinhanças para $n = 16$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	104	10320	1,158	-	64	MHAS	174	1792	0,426	-2,34%
3	RS	124	6144	0,763	-34,12%	80	PRS	174	1760	0,422	-0,80%
5	PRS	124	5792	0,725	-4,93%	106	MHAS	170	1760	0,412	-2,42%
6	RS	164	3184	0,549	-24,34%	129	MHAS	166	1760	0,402	-2,48%
8	RS	188	2352	0,521	-5,04%	134	MHAS	164	1760	0,397	-1,27%
13	PRS	184	2312	0,507	-2,78%	154	PRS	164	1728	0,393	-0,86%
17	THAS	182	2312	0,502	-1,01%	178	MHAS	160	1728	0,383	-2,59%
23	PRS	174	2328	0,483	-3,73%	195	PRS	160	1720	0,382	-0,22%
37	RS	174	2320	0,482	-0,18%	212	MHAS	156	1720	0,372	-2,67%
40	GR	182	2072	0,476	-1,26%	343	MHAS	152	1720	0,362	-2,74%
42	MHAS	182	2072	0,476	-0,00%	381	MHAS	146	1720	0,347	-4,23%
44	MHAS	178	2072	0,466	-2,14%	439	MHAS	138	1720	0,326	-5,89%
51	RS	178	1792	0,436	-6,42%	633	MHAS	134	1720	0,316	-3,13%
60	MHAS	174	1792	0,436	-0,00%						

Fonte: Elaborado pelo autor.

Tabela 49: Análise de vizinhanças para $n = 18$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	114	15504	1,129	-	99	RS	204	2320	0,381	-11,29%
8	MHAS	110	15504	1,121	-0,69%	116	MHAS	202	2320	0,378	-1,02%
16	GR	158	13840	1,098	-2,03%	124	GR	194	2304	0,361	-4,44%
17	GR	182	12472	1,049	-4,45%	133	GR	190	2304	0,353	-2,17%
18	RS	194	7720	0,740	-29,48%	146	MHAS	186	2304	0,345	-2,21%
20	GR	194	5856	0,610	-17,64%	167	PRS	186	2272	0,343	-0,65%
21	PRS	198	5856	0,603	-1,02%	170	PRS	188	2216	0,343	-0,00%
22	PRS	202	5384	0,592	-1,86%	197	MHAS	184	2216	0,335	-2,28%
24	RS	206	4752	0,556	-6,16%	255	PRS	188	2104	0,335	-0,01%
25	PRS	214	4456	0,551	-0,92%	336	PRS	180	2160	0,323	-3,49%
27	PRS	214	4320	0,541	-1,73%	371	PRS	180	2144	0,322	-0,35%
28	PRS	226	3976	0,540	-0,12%	387	PRS	176	2224	0,320	-0,69%
31	RS	238	3592	0,537	-0,64%	401	PRS	176	2208	0,319	-0,35%
33	MHAS	236	3592	0,533	-0,73%	431	PRS	172	2184	0,309	-2,98%
37	MHAS	234	3592	0,529	-0,73%	434	MHAS	168	2184	0,302	-2,52%
38	PRS	226	3560	0,527	-0,42%	438	PRS	168	2160	0,300	-0,56%
40	PRS	222	3504	0,507	-3,71%	503	PRS	168	2144	0,299	-0,37%
41	MHAS	246	3504	0,500	-1,54%	553	PRS	166	2168	0,297	-0,74%
50	RS	246	2752	0,494	-1,16%	576	MHAS	162	2168	0,289	-2,63%
58	RS	246	2608	0,484	-2,04%	773	PRS	162	2160	0,288	-0,19%
59	GR	246	2560	0,480	-0,70%	984	MHAS	158	2160	0,280	-2,71%
60	THAS	244	2560	0,476	-0,81%	1156	MHAS	154	2160	0,273	-2,79%
61	RS	236	2640	0,466	-2,10%	1190	MHAS	150	2160	0,265	-2,87%
63	PRS	236	2636	0,466	-0,06%	1310	MHAS	146	2160	0,257	-2,95%
66	PRS	236	2624	0,465	-0,18%	1323	MHAS	142	2160	0,249	-3,04%
77	RS	232	2344	0,438	-5,89%	1334	GR	142	2144	0,248	-0,45%
83	THAS	230	2344	0,434	-0,89%	1337	PRS	142	2128	0,247	-0,45%
91	GR	228	2344	0,430	-0,90%	1605	MHAS	140	2128	0,243	-1,58%

Fonte: Elaborado pelo autor.

Tabela 50: Análise de vizinhanças para $n = 18$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	130	15504	1,160	-	137	PRS	204	2232	0,375	-2,89%
2	RS	166	11152	0,926	-20,21%	155	MHAS	200	2232	0,367	-2,08%
6	GR	206	10016	0,924	-0,15%	181	MHAS	196	2232	0,360	-2,13%
7	PRS	210	9264	0,879	-4,85%	197	MHAS	192	2232	0,352	-2,17%
12	RS	230	5632	0,664	-24,48%	213	GR	192	2208	0,350	-0,48%
15	RS	230	4200	0,564	-15,10%	252	MHAS	188	2208	0,342	-2,23%
17	MHAS	226	4200	0,556	-1,39%	289	PRS	188	2192	0,341	-0,33%
20	RS	235	3408	0,518	-6,81%	298	MHAS	184	2192	0,333	-2,29%
23	RS	220	2864	0,451	-13,01%	348	MHAS	180	2192	0,326	-2,34%
29	GR	224	2592	0,440	-2,49%	362	MHAS	172	2192	0,310	-4,80%
33	PRS	222	2592	0,436	-0,84%	421	MHAS	164	2192	0,294	-5,04%
37	PRS	222	2520	0,431	-1,21%	593	MHAS	160	2192	0,287	-2,65%
40	GR	220	2456	0,422	-1,95%	692	GR	156	2192	0,279	-2,73%
49	PRS	220	2408	0,419	-0,80%	920	MHAS	152	2192	0,271	-2,80%
59	RS	220	2288	0,410	-2,01%	961	PRS	150	2208	0,268	-1,03%
75	MHAS	216	2288	0,403	-1,90%	1038	PRS	146	2216	0,261	-2,70%
79	GR	212	2280	0,394	-2,08%	1135	PRS	146	2200	0,260	-0,43%
91	MHAS	208	2280	0,386	-1,98%						

Fonte: Elaborado pelo autor.

Tabela 51: Análise de vizinhanças para $n = 18$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	176	15504	1,250	-	215	MHAS	200	2272	0,370	-2,07%
1	RS	224	9696	0,937	-25,04%	238	PRS	200	2232	0,367	-0,76%
10	RS	248	6296	0,746	-20,41%	295	PRS	200	2200	0,365	-0,61%
11	GR	256	5776	0,725	-2,79%	315	MHAS	196	2200	0,357	-2,14%
16	PRS	256	5368	0,696	-3,94%	317	PRS	192	2232	0,352	-1,56%
18	RS	252	3920	0,587	-15,68%	318	GR	192	2208	0,350	-0,48%
19	RS	248	3552	0,554	-5,72%	331	PRS	188	2232	0,344	-1,75%
21	MHAS	240	3552	0,538	-2,82%	335	MHAS	184	2232	0,336	-2,27%
23	MHAS	232	3552	0,522	-2,90%	364	MHAS	180	2232	0,328	-2,32%
25	PRS	226	3368	0,498	-4,71%	367	PRS	178	2272	0,327	-0,34%
32	GR	220	3144	0,470	-5,51%	403	MHAS	174	2272	0,319	-2,39%
44	GR	222	2904	0,457	-2,74%	435	GR	170	2272	0,312	-2,45%
48	RS	234	2544	0,456	-0,39%	488	MHAS	166	2272	0,304	-2,51%
59	GR	230	2464	0,442	-2,94%	521	MHAS	162	2272	0,296	-2,57%
63	THAS	228	2464	0,438	-0,88%	527	MHAS	160	2272	0,292	-1,32%
77	PRS	228	2456	0,438	-0,13%	540	PRS	156	2248	0,283	-3,25%
81	RS	228	2400	0,434	-0,90%	544	MHAS	152	2248	0,275	-2,76%
94	PRS	228	2336	0,429	-1,03%	699	PRS	152	2184	0,270	-1,63%
103	GR	222	2360	0,419	-2,34%	793	MHAS	148	2184	0,263	-2,89%
137	PRS	218	2288	0,407	-3,07%	979	MHAS	144	2184	0,255	-2,98%
145	MHAS	216	2288	0,403	-0,96%	1139	MHAS	140	2184	0,247	-3,07%
147	MHAS	212	2288	0,395	-1,94%	1395	MHAS	136	2184	0,239	-3,16%
154	GR	212	2272	0,394	-0,28%	1435	MHAS	132	2184	0,231	-3,27%
162	MHAS	208	2272	0,386	-1,98%	1456	MHAS	128	2184	0,223	-3,38%
186	MHAS	204	2272	0,378	-2,02%						

Fonte: Elaborado pelo autor.

Tabela 52: Análise de vizinhanças para $n = 20$ - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	140	22192	1,133	-	211	MHAS	240	2752	0,347	-2,60%
7	RS	152	14544	0,781	-31,03%	230	RS	244	2584	0,345	-0,56%
13	RS	212	11280	0,716	-8,36%	268	RS	228	2824	0,332	-3,80%
15	MHAS	210	11280	0,713	-0,43%	276	MHAS	222	2824	0,322	-2,79%
19	RS	226	6424	0,503	-29,49%	335	MHAS	218	2824	0,316	-1,92%
20	RS	294	3928	0,487	-3,14%	343	MHAS	214	2824	0,310	-1,95%—
34	RS	290	3464	0,458	-5,88%	362	MHAS	210	2824	0,304	-1,99%
35	MHAS	286	3464	0,452	-1,35%	416	MHAS	206	2824	0,298	-2,03%
56	RS	254	3472	0,403	-10,84%	797	MHAS	206	2824	0,298	-0,00%
58	RS	274	2752	0,399	-0,98%	807	MHAS	202	2824	0,291	-2,07%
59	MHAS	270	2752	0,393	-1,55%	916	MHAS	202	2824	0,291	-0,00%
66	MHAS	266	2752	0,387	-1,57%	921	MHAS	198	2824	0,285	-2,12%
94	MHAS	262	2752	0,381	-1,60%	939	MHAS	198	2824	0,285	-0,00%
96	MHAS	262	2752	0,381	-0,00%	1006	MHAS	196	2824	0,282	-1,08%
101	MHAS	260	2572	0,369	-3,10%	1008	MHAS	192	2824	0,276	-2,19%
116	MHAS	256	2752	0,371	0,69%	1123	MHAS	188	2824	0,270	-2,24%
138	MHAS	254	2752	0,368	-0,83%	1150	MHAS	184	2824	0,264	-2,29%
169	MHAS	250	2752	0,362	-1,68%	1231	MHAS	184	2824	0,264	-0,00%
178	MHAS	246	2752	0,356	-1,70%	1559	MHAS	180	2824	0,258	-2,34%
210	RS	246	2752	0,356	-0,00%	1847	MHAS	176	2824	0,251	-2,40%

Fonte: Elaborado pelo autor.

Tabela 53: Análise de vizinhanças para $n = 20$ - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	190	22192	1,210	-	232	MHAS	244	3080	0,369	-1,65%
1	RS	222	14528	0,889	-26,56%	265	MHAS	240	3080	0,363	-1,67%
2	RS	234	11424	0,757	-14,82%	292	MHAS	232	3080	0,350	-3,41%
6	RS	250	7736	0,603	-20,31%	314	GR	228	3080	0,344	-1,76%
17	RS	294	5168	0,547	-9,34%	353	MHAS	224	3080	0,338	-1,79%
26	MHAS	272	5168	0,513	-6,21%	364	MHAS	222	3080	0,335	-0,91%
28	RS	274	3504	0,435	-15,09%	420	MHAS	218	3080	0,329	-1,84%
61	RS	262	3272	0,406	-6,83%	466	MHAS	218	3080	0,329	-0,00%
85	MHAS	258	3272	0,400	-1,52%	525	MHAS	214	3080	0,322	-1,88%
86	RS	256	3080	0,387	-3,10%	655	MHAS	210	3080	0,316	-1,91%
99	MHAS	252	3080	0,381	-1,59%	659	MHAS	206	3080	0,310	-1,95%
110	MHAS	252	3080	0,381	-0,00%	713	RS	202	2984	0,299	-3,49%
111	MHAS	248	3080	0,375	-1,62%	715	MHAS	202	2984	0,299	-0,00%
117	MHAS	248	3080	0,375	-0,00%						

Fonte: Elaborado pelo autor.

Tabela 54: Análise de vizinhanças para $n = 20$ - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	140	22192	1,133	-	235	MHAS	212	3144	0,322	-1,88%
1	RS	152	14544	0,781	-31,03%	300	MHAS	208	3144	0,316	-1,91%
9	RS	200	11088	0,688	-11,92%	333	MHAS	208	3144	0,316	-0,00%
16	RS	230	6752	0,525	-23,75%	401	MHAS	202	3144	0,307	-2,93%
17	RS	238	5416	0,472	-9,96%	533	MHAS	195	3144	0,296	-3,52%
18	MHAS	238	5416	0,472	-0,00%	601	MHAS	194	3144	0,295	-0,52%
23	MHAS	234	5416	0,466	-1,31%	609	RS	202	2864	0,293	-0,41%
24	RS	274	4040	0,461	-1,04%	717	MHAS	202	2864	0,293	-0,00%
45	RS	262	3256	0,405	-12,23%	754	MHAS	198	2864	0,287	-2,10%
58	RS	262	3080	0,396	-2,10%	787	MHAS	198	2864	0,287	-0,00%
60	RS	254	3144	0,387	-2,33%	947	MHAS	190	2864	0,275	-4,30%
66	MHAS	250	3144	0,381	-1,59%	1125	GR	186	2864	0,269	-2,25%
79	MHAS	246	3144	0,375	-1,62%	1171	MHAS	186	2864	0,269	-0,00%
97	MHAS	242	3144	0,369	-1,65%	1542	MHAS	182	2864	0,263	-2,30%
117	MHAS	238	3144	0,363	-1,67%	1542	MHAS	182	2864	0,263	-0,00%
148	GR	234	3144	0,356	-1,70%	1696	MHAS	178	2864	0,256	-2,35%
154	THAS	228	3144	0,347	-2,60%	1731	MHAS	174	2864	0,250	-2,41%
166	MHAS	224	3144	0,341	-1,78%	2014	MHAS	170	2864	0,244	-2,47%
176	MHAS	222	3144	0,338	-0,91%	2357	MHAS	166	2864	0,238	-2,53%
221	MHAS	220	3144	0,335	-0,91%	2641	MHAS	162	2864	0,232	-2,60%
222	MHAS	216	3144	0,329	-1,84%	2856	MHAS	158	2864	0,226	-2,66%

Fonte: Elaborado pelo autor.

Tabela 55: Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 1.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	100	22192	1,071	-	292	MHAS	252	2536	0,355	-1,71%
2	RS	160	16448	0,886	-17,30%	366	MHAS	248	2536	0,349	-1,74%
14	RS	182	10144	0,615	-30,60%	407	RS	240	2632	0,341	-2,21%
15	RS	186	9936	0,611	-0,63%	423	MHAS	236	2632	0,335	-1,81%
20	RS	250	6832	0,559	-8,41%	424	RS	228	2800	0,330	-1,26%
30	RS	310	4656	0,547	-2,26%	451	MHAS	224	2800	0,324	-1,87%
52	RS	274	4024	0,461	-15,75%	487	MHAS	222	2800	0,321	-0,95%
57	MHAS	270	4024	0,454	-1,34%	542	MHAS	218	2800	0,315	-1,92%
65	RS	286	3328	0,445	-1,98%	618	MHAS	214	2800	0,309	-1,96%
92	MHAS	282	3328	0,439	-1,39%	657	MHAS	210	2800	0,303	-2,00%
107	MHAS	278	3328	0,433	-1,41%	765	MHAS	210	2800	0,303	-0,00%
117	MHAS	274	3328	0,427	-1,43%	881	MHAS	206	2800	0,296	-2,04%
157	RS	270	2944	0,402	-5,80%	1143	MHAS	202	2800	0,290	-2,08%
159	RS	266	2712	0,385	-4,32%	1191	MHAS	202	2800	0,290	-0,00%
170	RS	258	2808	0,377	-2,00%	1303	MHAS	198	2800	0,284	-2,13%
176	RS	266	2536	0,376	-0,22%	1409	MHAS	198	2800	0,284	-0,00%
196	MHAS	264	2536	0,373	-0,82%	1416	MHAS	194	2800	0,278	-2,17%
228	MHAS	264	2536	0,373	-0,00%	1546	MHAS	190	2800	0,272	-2,22%
260	MHAS	260	2536	0,367	-1,65%	1551	MHAS	186	2800	0,266	-2,27%
267	MHAS	256	2536	0,361	-1,68%	1579	MHAS	182	2800	0,259	-2,32%
286	MHAS	256	2536	0,361	-0,00%						

Fonte: Elaborado pelo autor.

Tabela 56: Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 2.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	126	22192	1,111	-	553	MHAS	244	3544	0,391	-1,55%
4	RS	154	14528	0,784	-29,48%	623	RS	244	3248	0,377	-3,66%
48	MHAS	148	14528	0,774	-1,18%	633	MHAS	244	3248	0,377	-0,00%
69	RS	196	11280	0,691	-10,73%	637	MHAS	236	3248	0,364	-3,28%
70	RS	220	7368	0,539	-22,02%	682	MHAS	232	3248	0,358	-1,69%
71	RS	244	5424	0,482	-10,57%	727	MHAS	228	3248	0,352	-1,72%
91	MHAS	244	5424	0,482	-0,00%	763	MHAS	228	3248	0,352	-0,00%
94	MHAS	240	5424	0,476	-1,28%	798	MHAS	224	3248	0,346	-1,75%
105	MHAS	238	5424	0,473	-0,65%	823	MHAS	220	3248	0,340	-1,78%
108	RS	254	4448	0,450	-4,76%	864	MHAS	216	3248	0,334	-1,82%
154	RS	246	4584	0,445	-1,28%	872	MHAS	212	3248	0,327	-1,85%
200	MHAS	242	4584	0,438	-1,39%	1042	MHAS	210	3248	0,324	-0,94%
212	RS	274	3544	0,437	-0,21%	1085	MHAS	206	3248	0,318	-1,90%
216	MHAS	270	3544	0,431	-1,41%	1329	MHAS	202	3248	0,312	-1,94%
317	MHAS	264	3544	0,422	-2,15%	1682	MHAS	198	3248	0,306	-1,98%
326	MHAS	264	3544	0,422	-0,00%	1893	MHAS	194	3248	0,300	-2,02%
351	MHAS	260	3544	0,416	-1,46%	2245	MHAS	190	3248	0,293	-2,06%
366	MHAS	252	3544	0,403	-2,97%	2358	MHAS	186	3248	0,287	-2,10
399	MHAS	248	3544	0,397	-1,53%						

Fonte: Elaborado pelo autor.

Tabela 57: Análise de vizinhanças para a instância do Campeonato Brasileiro de Futebol Série A 2023 - Execução 3.

Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria	Iter	VE	<i>brk</i>	<i>coe</i>	<i>z</i>	Melhoria
0	-	158	22192	1,160	-	821	MHAS	252	3560	0,404	-0,00%
12	RS	182	16448	0,920	-20,75%	849	MHAS	248	3560	0,398	-1,53%
43	RS	222	12288	0,780	-15,17%	873	MHAS	248	3560	0,398	-0,00%
59	MHAS	218	12288	0,774	-0,79%	892	MHAS	244	3560	0,392	-1,55%
66	MHAS	218	12288	0,774	-0,00%	899	MHAS	240	3560	0,386	-1,58%
120	RS	234	11728	0,772	-0,31%	902	MHAS	236	3560	0,380	-1,60%
126	RS	266	7552	0,619	-19,78%	1157	MHAS	232	3560	0,373	-1,63%
144	MHAS	262	7552	0,613	-1,00%	1217	MHAS	228	3560	0,367	-1,65%
148	THAS	258	7552	0,607	-1,01%	1249	MHAS	224	3560	0,361	-1,68%
197	RS	242	7928	0,600	-1,07%	1278	MHAS	220	3560	0,355	-1,71%
240	RS	254	7512	0,599	-0,27%	1288	MHAS	218	3560	0,352	-0,87%
246	RS	278	5464	0,536	-10,36%	1322	RS	206	3256	0,319	-9,44%
284	MHAS	276	5464	0,533	-0,58%	1384	MHAS	202	3256	0,312	-1,94%
292	RS	268	5616	0,528	-0,94%	1472	MHAS	200	3256	0,309	-0,99%
351	THAS	260	5616	0,516	-2,34%	1503	RS	216	2416	0,293	-5,15%
378	THAS	258	5616	0,513	-0,60%	1529	MHAS	216	2416	0,293	-0,00%
432	MHAS	250	5616	0,501	-2,41%	1569	MHAS	212	2416	0,287	-2,10%
450	MHAS	246	5616	0,494	-1,23%	1581	MHAS	212	2416	0,287	-0,00%
480	MHAS	242	5616	0,488	-1,25%	1638	MHAS	208	2416	0,281	-2,15%
481	RS	274	4552	0,486	-0,43%	1649	MHAS	208	2416	0,281	-0,00%
485	RS	302	3560	0,481	-0,98%	1692	MHAS	204	2416	0,275	-2,20%
493	MHAS	294	3560	0,469	-2,56%	1698	MHAS	200	2416	0,269	-2,25%
499	MHAS	294	3560	0,469	-0,00%	1940	MHAS	196	2416	0,262	-2,30%
518	MHAS	292	3560	0,466	-0,66%	1968	MHAS	196	2416	0,262	-0,00%
520	THAS	288	3560	0,460	-1,32%	2301	MHAS	194	2416	0,259	-1,18%
527	MHAS	280	3560	0,447	-2,69%	2329	MHAS	194	2416	0,259	-0,00%
554	MHAS	276	3560	0,441	-1,38%	2631	MHAS	190	2416	0,253	-2,38%
570	MHAS	276	3560	0,441	-0,00%	2681	MHAS	190	2416	0,253	-0,00%
583	MHAS	272	3560	0,435	-1,40%	2850	MHAS	186	2416	0,247	-2,44%
613	MHAS	272	3560	0,435	-0,00%	2872	GR	186	2416	0,247	-0,00%
641	MHAS	268	3560	0,429	-1,42%	3003	MHAS	182	2416	0,241	-2,50%
660	MHAS	264	3560	0,423	-1,44%	3012	MHAS	180	2416	0,238	-1,28%
664	MHAS	256	3560	0,410	-2,92%	3054	MHAS	180	2416	0,238	-0,00%
684	MHAS	252	3560	0,404	-1,50%						

Fonte: Elaborado pelo autor.