Filipe Pádua Ribeiro

Algoritmo Híbrido *Iterated Local Search* e *Simulated Annealing* para o Problema de Tabela-Horário de Jogos da Superliga Brasileira de Voleibol

ALEGRE - ES

2025

Filipe Pádua Ribeiro

Algoritmo Híbrido *Iterated Local Search* e *Simulated Annealing* para o Problema de Tabela-Horário de Jogos da Superliga Brasileira de Voleibol

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo

Centro de Ciências Exatas, Naturais e da Saúde

Departamento de Computação

Orientador: Prof. Dr. Edmar Hell Kampke

ALEGRE - ES

2025

Filipe Pádua Ribeiro

Algoritmo Híbrido *Iterated Local Search* e *Simulated Annealing* para o Problema de Tabela-Horário de Jogos da Superliga Brasileira de Voleibol

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 19 de Março de 2025.

COMISSÃO EXAMINADORA

Prof. Dr. Edmar Hell Kampke
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Prof. Dr. Rodrigo Freitas Silva
Universidade Federal do Espírito Santo

Agradecimentos

Em primeiro lugar, agradeço a Deus por me dar saúde e guarnição durante toda a minha jornada.

Agradeço também aos meus pais, Ubirajara e Márcia, por me darem a educação necessária para, mesmo estando longe de casa, buscar cumprir o meu dever da maneira correta e respeitando meus princípios. Obrigado a toda a minha família, meus irmãos, minhas irmãs, minhas avós Terezinha e Ana Maria, e ao meu querido avô José Miwton, que nos deixou durante este período em que estive na faculdade. Obrigado a minha namorada Victória por, mesmo distante, sempre estar ao meu lado em todos os momentos, o que me fazia lembrar o motivo de me dedicar tanto durante esses anos de graduação.

Um agradecimento mais que especial a todos os meus amigos que estiveram comigo durante esses anos: Arthur, Christian, Daniel, João Celso, Matheus, Talles e Vanderson. Sei que fiz amizades que quero levar para a vida, e sem dúvida vocês fizeram a minha jornada mais rica e divertida. Um agradecimento também a todos os professores do DCOMP, pelos ensinamentos transmitidos. Por fim, mas não menos importante, um agradecimento especial ao professor Edmar, por ter me convidado a participar do projeto de iniciação científica, por ter aceitado ser meu orientador neste trabalho de conclusão de curso, mas acima de tudo, pela amizade.

Esses anos sem dúvida me fizeram crescer como pessoa e como profissional, e serei eternamente grato por tudo o que vivi. Muito obrigado.

Resumo

Este trabalho apresenta uma solução baseada no algoritmo híbrido *Iterated Local Search* com *Simulated Annealing* para resolver o Problema de Tabela-Horário de Jogos da Superliga Brasileira de Voleibol. Este problema consiste em encontrar a melhor organização dos jogos do campeonato para que a distância percorrida pelas equipes seja mínima, reduzindo os gastos com o deslocamento e aumentando o tempo disponível para o descanso e a preparação dos atletas. O método proposto para a resolução do problema se baseia na meta-heurística *Iterated Local Search*, que a partir de uma solução ótima local, iterativamente sofre perturbações. As soluções obtidas após a perturbação são enviadas para o *Simulated Annealing*, responsável por realizar a busca local. Neste trabalho, foram obtidos resultados consideravelmente melhores do que a solução atualmente utilizada pela Superliga Brasileira de Voleibol, apresentando, em média, soluções 48,2% melhores.

Palavras-chave: Superliga Brasileira de Voleibol, Tabela-Horário de Jogos, Meta-heurísticas, *Iterated Local Search*, *Simulated Annealing*.

Abstract

This work presents a solution based on the hybrid algorithm Iterated Local Search with Simulated Annealing to address the Schedule Timetabling Problem for Volleyball Games in the Brazilian League. This problem involves finding the best organization of championship games to minimize the distance traveled by teams, thereby reducing transportation costs and increasing the time available for rest and athlete preparation. The proposed method for solving the problem is based on the metaheuristic Iterated Local Search, which, starting from an optimal local solution, undergoes iterative perturbations. The solutions obtained after the disturbance are then sent to Simulated Annealing, which performs the local search. In this work, the proposed approach achieved significantly better results than the current solution used by the Brazilian League, yielding, on average, solutions that are 48.2% better.

Keywords: Brazilian Volleyball Super League, Game Scheduling, Metaheuristics, Iterated Local Search, Simulated Annealing.

Lista de Figuras

Figura 1 – Resumo ilustrativo de Lourenço, Martin e Stützle (2003) do espaço de soluções do ILS. 19
Figura 2 – Ilustração de Biajoli (2007) para uma geração de rodadas com $n = 6$. . 20
Figura 3 – Exemplo do movimento Troca de Ordem entre Rodadas. 21
Figura 4 – Exemplo do movimento Inversão de Mando de Campo. 22
Figura 5 – Exemplo do movimento Permutação de Times nas Rodadas. 23
Figura 6 – Exemplo de formato das instâncias. 27

Lista de Tabelas

Tabela 1 – Calibração dos Parâmetros. 27
Tabela 2 – Ambiente Computacional. 28
Tabela 3 – Resultados Obtidos pelo ILS+SA. 29
Tabela 4 – Comparação de Resultados. 30

Lista de Algoritmos

1 Algoritmo *ILS* proposto por Lourenço, Martin e Stützle (2003) 18

2 Algoritmo *SA* adaptado de Rocha (2013) 24

Lista de Abreviaturas e Siglas

CCENS	Centro de Ciências Exatas, Naturais e da Saúde
DCOMP	Departamento de Computação
FO	Função Objetivo
ILS	*Iterated Local Search*
PTHJ	Problema de Tabela-Horário de Jogos
SA	*Simulated Annealing*
SBV	Superliga Brasileira de Voleibol
UFES	Universidade Federal do Espírito Santo

Sumário

1 Introdução **11**
1.1 O Problema e sua Importância 11
1.2 Objetivos 12
1.2.1 Objetivo Geral 13
1.2.2 Objetivos Específicos 13

2 Revisão de Literatura **14**

3 Metodologia **17**
3.1 Modelagem do Problema 17
3.2 ILS 18
3.2.1 Construção da solução inicial 19
3.2.2 Perturbação 21
3.2.2.1 Troca de Ordem entre Rodadas 21
3.2.2.2 Inversão do Mando de Campo 22
3.2.2.3 Permutação de Times nas Rodadas 22
3.2.3 Busca local 23
3.2.3.1 *Simulated Annealing* 23
3.2.3.2 *Geração de Soluções Vizinhas* 25

4 Resultados Obtidos **26**
4.1 Experimentos Computacionais 26
4.1.1 Instâncias 26
4.1.2 Escolha dos Parâmetros 27
4.1.3 Detalhes de Implementação e Execução de Testes 28
4.2 Resultados Computacionais 28
4.3 Análise e Discussão 29

5 Conclusão **31**

Referências **32**

1 Introdução

A automatização da programação de ligas esportivas tornou-se uma importante vertente de aplicações voltadas para a otimização combinatória nos últimos anos. Enquanto as ligas esportivas representam significativas fontes de receitas para as redes de rádio, televisão e de entretenimento em todo o mundo, essa realidade também gera desafios extremamente complexos de otimização (ANAGNOSTOPOULOS et al., 2006).

Segundo Ribeiro e Urrutia (2007), calendários com tempos mínimos de viagem e que ofereçam condições semelhantes a todos os times são de grande interesse a todos os envolvidos em uma liga desportiva: times, patrocinadores, torcedores e mídia. No caso do campeonato brasileiro de futebol, uma única viagem de Porto Alegre a Belém, por exemplo, possui um trajeto extenso e muitas paradas, devido à ausência de voos diretos para percorrer uma distância de aproximadamente 4.000 km. Com isso, a distância total percorrida torna-se uma variável importante a ser minimizada, de modo que se possa reduzir os custos de deslocamento e dar mais tempo aos jogadores para descanso e treinamento.

Ligas profissionais de esportes são disputadas durante a maior parte do ano, e requerem que seus atletas, equipes e trabalhadores envolvidos em todo o processo de uma partida, mantenham um alto nível de desempenho. Diante disso, uma chave para esses níveis de rendimento é o calendário que as equipes jogam. Nenhum detentor de direitos de transmissão quer pagar grandes quantias apenas para que equipes pouco atraentes joguem em datas nobres. As equipes não querem ver seus grandes investimentos em jogadores e infraestrutura prejudicada por uma má programação dos jogos. Os torcedores, que fornecem boa parte da renda e popularidade para as ligas, também são muito afetados pelos horários dos jogos (EASTON; NEMHAUSER; TRICK, 2001).

Portanto, a eficácia na programação de jogos não atende apenas às demandas dos envolvidos, como as equipes e detentores de direitos de transmissão, mas também o interesse e o apoio dos torcedores, elementos essenciais para a sustentabilidade e sucesso das ligas esportivas profissionais.

1.1 O Problema e sua Importância

O Brasil é um país com dimensões continentais e, portanto, os campeonatos esportivos disputados nacionalmente sofrem com questões que envolvem o deslocamento das equipes. A Superliga Brasileira de Voleibol (SBV) é um exemplo disso, pois conta atualmente com a participação de 12 equipes tanto para a modalidade Masculina quanto para a Feminina, e é composta por quatro fases: classificatória, quartas de final, semifinal e final. A fase

classificatória é organizada em 22 rodadas (turno e returno) de seis partidas cada, com todos os 12 times jogando entre si, sendo que no returno as equipes jogam invertendo o mando de campo dos jogos do turno, seguindo o modelo de tabela espelhada (se um time x joga em casa contra o time y na rodada de número r do turno, a partida de número r do returno deverá ser entre a equipe y e x, realizada na casa da equipe y). As demais fases são organizadas de acordo com o resultado das fases anteriores. Dado este volume de partidas e as restrições de mando de campo impostas pela organização das ligas, as equipes devem viajar constantemente, limitando o tempo disponível para a recuperação, o treino dos atletas e, consequentemente, reduzindo o seu desempenho.

O Problema de Tabela de Horário de Jogos (PTHJ), que é observado na criação das tabelas da SBV, é constituído por um conjunto de possibilidades de alocação das rodadas. As rodadas devem ser preenchidas de modo que todos os 12 times do campeonato joguem, e devem respeitar todas as restrições impostas pela organização da SBV.

De acordo com Easton, Nemhauser e Trick (2001), mesmo casos com apenas oito equipes são intratáveis em relação ao estado da arte. Isso torna o PTHJ atraente como referência, pois é relativamente fácil de declarar e os requisitos de dados são mínimos.

Segundo Ribeiro (2012), existem muitos aspectos relevantes a serem considerados na determinação da melhor agenda de um torneio. Em algumas situações, busca-se um cronograma que minimize a distância total percorrida pelas equipes que disputam o campeonato, como no caso do problema do torneio itinerante e em torneios de tabela espelhada (turno e returno com o mando de campo das equipes invertidos). Outros problemas tentam minimizar o número total de quebras, ou seja, o número de pares de jogos consecutivos em casa ou fora de casa disputados por uma mesma equipe. A distribuição uniforme da sequência de jogos é um outro fator que pode ser considerado na busca pela otimização do calendário do torneio, como descrito por Russell (1980).

Com isso, dada a importância da resolução do PTHJ e as dificuldades enfrentadas na disputa da SBV, observa-se a possibilidade e a oportunidade de se desenvolver uma maneira de otimizar a construção da tabela de jogos do campeonato, beneficiando, em primeira escala os atletas, disponibilizando mais tempo para sua preparação e recuperação, melhorando o nível de competitividade da liga e, consequentemente, atraindo mais público e patrocinadores para as partidas.

1.2 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

O objetivo primário deste trabalho é aplicar a meta-heurística *Iterated Local Search* (ILS), combinada com o *Simulated Annealing* (SA) para realizar a busca local no PTHJ da SBV, com o intuito de encontrar soluções melhores do que as utilizadas atualmente, minimizando assim as distâncias percorridas pelas equipes.

1.2.2 Objetivos Específicos

a) Estudar os dados necessários para resolver o PTHJ da SBV.

b) Implementar a meta-heurística ILS com SA para resolver o PTHJ da SBV.

c) Validar o algoritmo desenvolvido e calibrar os parâmetros.

d) Executar a aplicação utilizando as instâncias reais do problema.

e) Avaliar os resultados obtidos e compará-los com os resultados da tabela-horário usados atualmente na SBV.

2 Revisão de Literatura

Uma obra fundamental para compreender as abordagens do PTHJ foi desenvolvida por Ribeiro (2012), na qual é realizada uma revisão introdutória dos problemas fundamentais e da importância do PTHJ, tornando o artigo um importante guia para pesquisas relacionadas a esta área. Além disso, foram discutidas diferentes meta-heurísticas, como a Busca Tabu, Algoritmos Genéticos e *Simulated Annealing*, aplicadas por diversos autores para a resolução do PTHJ nos mais variados esportes, evidenciando a versatilidade de abordagens para o tratamento do problema.

Um dos trabalhos mais importantes acerca do PTHJ foi realizado por Russell e Leung (1994), aplicado na Liga de Beisebol do Texas (LBT) e resolvido pelos autores utilizando uma abordagem de construção da tabela de jogos por dois fatores, seguindo a geração de padrões casa-fora e atribuições de equipes a esses padrões. No artigo, o PTHJ é descrito como um problema notoriamente difícil de ser tratado. Além das restrições determinadas pela organização para a elaboração do calendário de jogos, existem diversos fatores adicionais que devem ser considerados, como a preferência de jogos em casa dos times em datas específicas e o equilíbrio da tabela para que os jogos entre dois times sejam igualmente espaçados durante toda a temporada. Portanto, além de encontrar um cronograma viável e que atenda a todas as restrições, o problema abordado pelos autores tem como objetivo minimizar os custos de viagem das equipes, tornando as soluções para o PTHJ extremamente interessantes aos envolvidos na LBT.

Bao (2009) realizou um estudo inspirado nas regras da Liga Americana de Basquete (*National Basketball Assossiacion* - NBA), na qual utiliza o formato de torneio itinerante, com as equipes viajando constantemente para realizar seus jogos. Para isso, o autor desenvolveu três algoritmos baseados no *Independent Lower Bound* (ILB) para resolver esse problema e apresentou todas as principais restrições presentes na temporada regular da NBA, separadas em três categorias: estruturais, externas e de justiça. Este estudo foi o primeiro a examinar as propriedades de torneios itinerantes aplicadas ao PTHJ, reforçando as possibilidades de exploração de soluções para o problema.

Assim, o PTHJ possui grande aplicação prática e muitos desafios a serem considerados. Esses desafios tornam o uso de métodos exatos, para encontrar soluções, computacionalmente inviável. Dentre as meta-heurísticas mais poderosas aplicadas para a solução de problemas de otimização está a ILS, que combina a busca local iterativa com uma perturbação controlada para tentar encontrar soluções de alta qualidade em problemas complexos.

Stützle (1998) abordou a meta-heurística ILS para a resolução do Problema de Fluxo

de Trabalho (PFT). Os resultados computacionais obtidos pelo estudo mostraram que a abordagem ILS se comparava muito favoravelmente a outras abordagens propostas para o PFT e que, apesar de sua simplicidade, o algoritmo foi capaz de encontrar melhores soluções para algumas instâncias em comparação a resultados encontrados por outros algoritmos.

Costa, Urrutia e Ribeiro (2012) propuseram a utilização da ILS em seu trabalho sobre o Problema do Torneio Itinerante com Locais Pré-Definidos (PTILP). Em seu artigo, os autores afirmam que o uso dessa meta-heurística apresenta um desempenho muito superior em comparação com heurísticas baseadas em programação inteira, e ainda aprimora as melhores soluções conhecidas para as instâncias utilizadas. Portanto, a meta-heurística ILS se mostra atrativa para solução de problemas de otimização combinatória.

Nos problemas de otimização, é de suma importância que o algoritmo utilizado para a resolução do problema seja capaz de escapar de mínimos locais para diversificar a busca por soluções. Diante disso, um algoritmo que atende este requisito é o SA, que busca soluções ótimas explorando o espaço de busca de forma probabilística.

Uma importante aplicação do SA foi realizada por Bouleimen e Lecocq (2003) para solucionar o Problema de Agendamento de Projetos com Restrições de Recursos (*Resource-Constrained Project Scheduling problem* - RCPSP). Os autores substituíram o algoritmo padrão do SA para que a busca se concentrasse em uma área mais específica do espaço de soluções ótimas do RCPSP, visando direcionar a busca local do SA para regiões mais promissoras.

Anagnostopoulos et al. (2006) propuseram uma aplicação do SA ao Problema do Torneio Itinerante (PTI) levando em consideração as características mais importantes da *Major League Baseball* (MLB) nos Estados Unidos. Nesse contexto, foi proposto um algoritmo do SA adaptado para o PTI, capaz de explorar tanto soluções viáveis quanto inviáveis, bem como suas vizinhanças. O resultado do uso da aplicação deste algoritmo ao PTI se mostrou robusto, uma vez que sua pior qualidade de solução obtida em mais de 50 execuções é consistentemente igual ou superior às melhores soluções conhecidas até aquele momento.

Além disso, Blum e Roli (2003) conduziram uma pesquisa acerca das diferentes meta-heurísticas existentes para a aplicação em problemas de otimização combinatória como a Busca Tabu, o SA, *Greed Randomized Adaptative Search Procedure* (GRASP), entre outros. Nesse estudo, foram levantadas as mais importantes meta-heurísticas utilizadas atualmente, bem como suas semelhanças, conceitos e diferenças. As principais meta-heurísticas estudadas foram: Otimização de Colônia de Formigas, Computação Evolutiva, Algoritmos Genéticos, ILS, SA e Busca Tabu. Com todas essas meta-heurísticas estudadas, os autores concluíram que a hibridização das meta-heurísticas apresentam, para muitos problemas, as melhores soluções conhecidas e, portanto, é uma área de pesquisa

extremamente promissora para a resolução de problemas de otimização combinatória.

Desse modo, a hibridização das meta-heurísticas ILS e SA é uma boa alternativa, já que ambas são consideradas simples e poderosas, o que pode resultar em uma solução mais refinada. Portanto, assim é justificada a hibridização das meta-heurísticas ILS e SA para a resolução do PTHJ da SBV, que é o objetivo deste trabalho. Além disso, não foi encontrado na literatura trabalhos que usam as meta-heurísticas ILS e SA para o PTHJ da SBV.

3 Metodologia

Neste trabalho foi implementada a meta-heurística ILS juntamente com o SA como busca local para a resolução do PTHJ da SBV.

3.1 Modelagem do Problema

A modelagem matemática do PTHJ da SBV segue a que foi proposta por Souza et al. (2005), tendo a solução s avaliada pela função objetivo f, com as seguintes definições:

- T: conjunto de times participantes da competição;
- D_i: deslocamento total do time $i \in T$;
- C: conjunto de restrições que descrevem o problema;
- R: conjunto de rodadas, sendo $|R| = (|T| - 1) * 2$;
- d_{ij}: distância percorrida pelo time $i \in T$ na rodada $j \in R$;
- τ_c: penalidade por desrespeitar a restrição $c \in C$;
- inv_c: número de vezes que a restrição $c \in C$ é desrespeitada;

Dessa forma, o modelo matemático é descrito da seguinte forma:

$$\text{Minimizar} \quad f(s) = \sum_{i \in T} D_i + \sum_{c \in C} \tau_c * inv_c \tag{3.1}$$

Sujeito a:

$$D_i = \sum_{j \in R} d_{ij}, \forall i \in T \tag{3.2}$$

A função objetivo 3.1, que deve ser minimizada, é calculada pela soma da distância total percorrida pelas equipes na solução s e baseada em penalidades descritas pelas seguintes restrições:

- Um time não pode jogar mais de uma vez na mesma rodada;
- Dois times jogarão entre si duas vezes, uma no turno e outra no returno, alternando o mando de campo entre os mesmos;

- Nenhum time deve jogar mais de seis vezes consecutivas dentro de casa;
- Nenhum time deve jogar mais de seis vezes consecutivas fora de casa;
- A diferença entre os jogos feitos em cada turno em casa e fora de casa de um time não pode ser maior que seis unidades.

As duas primeiras restrições foram definidas de acordo com as regras da SBV disponibilizadas no *website* da Confederação Brasileira de Vôlei (CBV): `https://cbv.com.br`. Por fim, as três últimas restrições foram definidas com base no trabalho de Souza et al. (2005), sendo o valor seis definido com base na tabela atual dos jogos da SBV. O modelo matemático termina com a restrição 3.2, que determina o valor da distância percorrida por cada equipe durante o campeonato.

O valor de τ_c foi definido como 1.000.000 para inviabilizar a solução, caso essa viole alguma restrição $c \in C$.

3.2 ILS

A meta-heurística ILS foi definida formalmente por Lourenço, Martin e Stützle (2003), apesar de já ter sido usada anteriormente por Stützle (1998). O ILS já se mostrou uma heurística extremamente poderosa e que, apesar de sua simplicidade conceitual, levou a uma série de resultados de última geração sem a necessidade de conhecimentos muito específicos sobre o problema. No Algoritmo 1 é mostrado o pseudocódigo genérico da meta-heurística ILS.

Algoritmo 1: Algoritmo *ILS* proposto por Lourenço, Martin e Stützle (2003)

Entrada: Max_{Iter}
Saída: Solução S^{Melhor}

1 **início**
2 $S^{Inicial} \leftarrow GeraSolucaoInicial();$
3 $S^{Melhor} \leftarrow BuscaLocal(S^{Inicial});$
4 $FO^{Melhor} \leftarrow f(S^{Melhor});$
5 **para** $i \leftarrow 1$ **até** Max_{Iter} **faça**
6 $S^{Atual} \leftarrow GeraPerturbacao(S^{Melhor});$
7 $S^{Atual} \leftarrow BuscaLocal(S^{Atual});$
8 **se** $f(S^{Atual}) < FO^{Melhor}$ **então**
9 $S^{Melhor} \leftarrow S^{Atual};$
10 $FO^{Melhor} \leftarrow f(S^{Atual});$
11 **fim se**
12 **fim para**
13 **fim**

Basicamente o algoritmo ILS consiste em quatro fases: construção da solução inicial (linha 2), refinamento da solução (linha 3), perturbação da solução refinada (linha 6) e refinamento da solução após perturbação (linha 7), sendo que todos os refinamentos são realizados por uma busca local.

A iteração do algoritmo é feita enquanto o laço de repetição não atingir o Max_{Iter}, perturbando repetidamente a solução S^{Melhor} (linha 6). A solução perturbada é melhorada por um procedimento de busca local (linha 7). A solução que atende o critério de aceitação é armazenada em S^{Melhor} (linhas 9 e 10), que é retornada ao final da execução.

Lourenço, Martin e Stützle (2003) evidenciam em seu artigo que a versatilidade e a eficiência da meta-heurística ILS se dá pela sua característica de focar a procura de soluções em subespaços localmente ótimos, e não focar em todo o espaço de soluções. Os autores ainda enfatizam a vantagem do método em sua capacidade de escapar de mínimos locais, aumentando assim a probabilidade de encontrar mínimos globais, conforme apresentado na Figura 1.

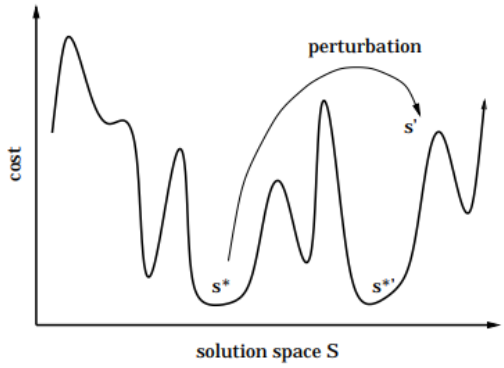

Figura 1 – Resumo ilustrativo de Lourenço, Martin e Stützle (2003) do espaço de soluções do ILS.

Fonte: Lourenço, Martin e Stützle (2003).

3.2.1 Construção da solução inicial

Uma das características do ILS é a liberdade de abordagens para a construção da solução inicial, podendo abordar métodos completamente aleatórios, completamente gulosos, parcialmente aleatórios, parcialmente gulosos ou até métodos híbridos, contendo ambas as abordagens. Para este trabalho, foi utilizada uma abordagem parcialmente aleatória, baseada na obra de Biajoli (2007), utilizando o método do polígono (COLBOURNE; DINITZ, 2007) para gerar a tabela dos jogos do primeiro turno e espelhá-la para gerar a tabela do segundo turno.

A inicialização do método do polígono é feita a partir de um vetor V de tamanho n (número de times) em que $V[i]$ representa um time. A execução começa com a escolha

aleatória de um time base, inserindo-o na primeira posição do vetor V. Os demais times também são dispostos aleatoriamente nas posições restantes ($i = 2, \ldots, n$).

Em cada rodada r_k, $k = 1, \ldots, n-1$, o time base $i = 1$ confronta o time da posição $i = 2$, definindo o jogo $T_{V[1]}, T_{V[2]}$. Os times da posições $i = 3, \ldots, (n/2) + 1$ confrontam os times das posições $j = n - i + 3$, definindo os demais jogos $T_{V[i]}, T_{V[j]}$. Após a definição de todos os jogos da rodada r_k, os times das posições $i = 2, \ldots, n$ são rotacionados no sentido horário dentro do vetor, onde os times das posições $i = 3, \ldots, n$ passam para a posição $i - 1$, e o time da posição 2 passa para a posição n.

A seguir, a Figura 2 ilustra a aplicação desse método com um vetor de times, em que $n = 6$, sendo o time 3 escolhido para ser o time base.

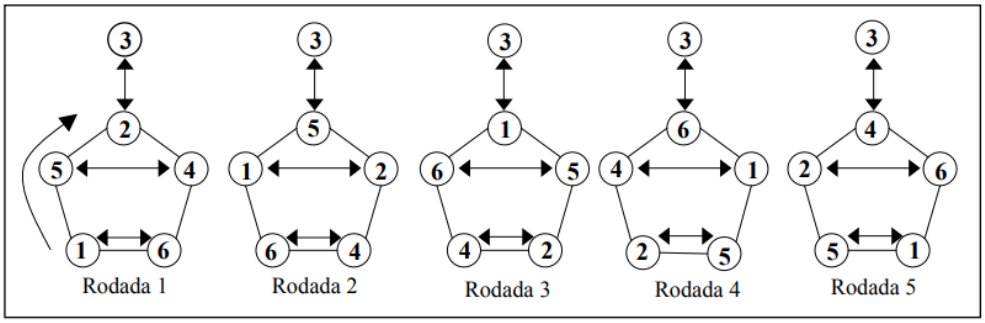

Figura 2 – Ilustração de Biajoli (2007) para uma geração de rodadas com $n = 6$.

Fonte: Biajoli (2007).

Para garantir a viabilidade da solução inicial, a distribuição dos mandos de campo das partidas é feita de maneira alternada. A definição dos mandos de campo é baseada na ordenação dos times no vetor V e suas posições representadas pelo método dos polígonos, alternando o mando de campo apenas para as posições base $i = 1$ e seu adversário $i = 2$. Em rodadas de numeração ímpar, o time base $i = 1$ realiza seu confronto em casa contra o time da posição $i = 2$. Já para as rodadas de numeração par, o mando de campo é invertido, com o time da posição $i = 2$ realizando o confronto em casa contra o time base de posição $i = 1$. Para os demais confrontos, fica definido que o time da posição ímpar sempre irá realizar seus confrontos em casa. Com isso, ao realizar o movimento de rotação no sentido horário, descrito pelo método do polígono, garante-se que a construção das partidas irá ser realizada com os mandos de campo alternados a cada rodada. Tomando como exemplo os confrontos definidos na Figura 2, temos as seguintes definições de mando de campo para as rodadas:

- Rodada 1: os times $3, 5, 6$ realizam seus confrontos em casa contra os times $2, 4, 1$, respectivamente;

- Rodada 2: os times $5, 1, 4$ realizam seus confrontos em casa contra os times $3, 2, 6$, respectivamente;
- Rodada 3: os times $3, 6, 2$ realizam seus confrontos em casa contra os times $1, 5, 4$, respectivamente;
- Rodada 4: os times $6, 4, 5$ realizam seus confrontos em casa contra os times $3, 1, 2$, respectivamente;
- Rodada 5: os times $3, 2, 1$ realizam seus confrontos em casa contra os times $4, 6, 5$, respectivamente;

Com isto, a solução inicial gerada pelo método ILS sempre apresenta uma solução viável passível de melhorias por meio da busca local.

3.2.2 Perturbação

A perturbação no ILS é a etapa na qual a solução atual é modificada, de alguma forma, antes de ser enviada à busca local novamente. Essas modificações da solução visam escapar de ótimos locais e explorar novas regiões do espaço de busca.

As implementações de métodos de perturbação podem variar de acordo com o problema abordado. Em alguns casos, a perturbação é feita através da destruição parcial e reconstrução de uma solução. Entretanto, neste trabalho, a perturbação é realizada por meio da modificação parcial de uma solução, sempre executando, a cada iteração e de forma sequencial, uma quantidade definida de cada um dos três tipos de movimentos apresentados a seguir:

3.2.2.1 Troca de Ordem entre Rodadas

Para realizar este tipo de movimento, são escolhidas aleatoriamente duas rodadas $i, j \in \{1, ..., |R|/2\}$ $(i \neq j)$. Em seguida, suas ordens de execução são invertidas no turno e no returno, conforme ilustrado na Figura 3, em um exemplo com $n = 4$ e seis rodadas no total, onde foi selecionado as rodadas 1 e 3 para serem permutadas.

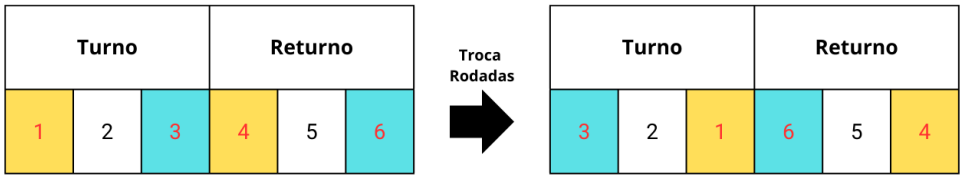

Figura 3 – Exemplo do movimento Troca de Ordem entre Rodadas.

A quantidade de execuções deste movimento é obtida pela multiplicação entre $|R|$ e o parâmetro de permutação de rodadas ρ.

3.2.2.2 Inversão do Mando de Campo

O segundo movimento consiste em inverter o mando de campo de uma partida. Para isso, sorteia-se uma rodada do primeiro turno $j \in \{1, ..., |R|/2\}$ e um time $i \in T$. A partir desse sorteio, a partida do time i tem seu mando de campo invertido tanto na rodada j no turno quanto na rodada $j + |R|/2$ no returno. No exemplo ilustrado pela Figura 4, temos o time B sendo sorteado dentro de uma rodada e a configuração de suas partidas sendo invertidas entre turno e returno após o movimento.

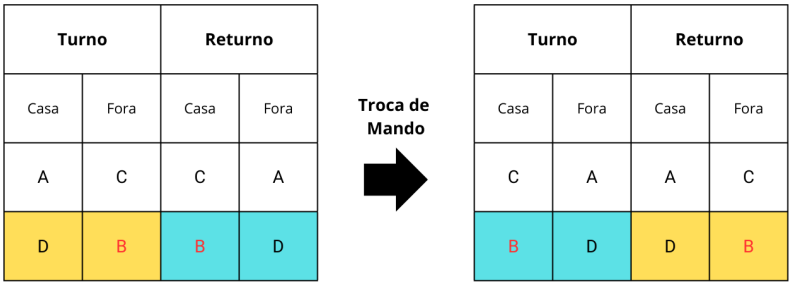

Figura 4 – Exemplo do movimento Inversão de Mando de Campo.

A quantidade de execuções deste movimento de troca é obtida pela multiplicação do parâmetro de permuta de mando de campo γ com o total de jogos realizados no turno $(n/2 \ * \ |R|/2)$.

3.2.2.3 Permutação de Times nas Rodadas

O terceiro e último movimento visa modificar as partidas dentro das rodadas do campeonato. Para isso, dois times $p, q \in T$ $(p \neq q)$ são selecionados aleatoriamente e em seguida completamente permutados ao longo de todas rodadas do turno. Isso significa que todas as ocorrências de partidas envolvendo p são substituídas por q, e vice-versa, mantendo o mando de campo original do time substituído. Nas rodadas em que p e q se enfrentam, o mando de campo da partida é invertido.

Esse movimento pode ser ilustrado na Figura 5, onde os times A e D foram sorteados. No cenário original, o time A enfrenta o time C na primeira rodada e o time D na segunda, ambas como mandante (coloração branca) e encerra a terceira rodada jogando contra o time B fora de casa (coloração cinza). Por sua vez, o time D inicia a competição jogando em casa contra o time B, depois viaja para enfrentar o time A e, na terceira rodada, retorna ao seu mando de campo para enfrentar o time C. Após o movimento de permutação, o time A herda a configuração original da tabela do time D, realizando seus

confrontos contra os times B, D e C, respectivamente. De maneira análoga, o time D assume a tabela original de A, enfrentando os times C, A e B, nesta ordem. Além disso, na rodada em que os times sorteados se enfrentam, apenas o mando de campo da partida é invertida.

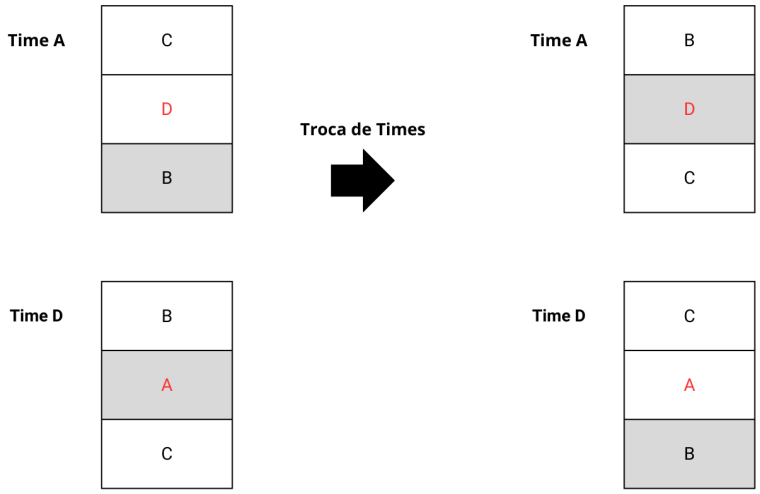

Figura 5 – Exemplo do movimento Permutação de Times nas Rodadas.

De modo análogo aos movimentos anteriores, a quantidade de execuções dessa troca é obtida multiplicando n pelo parâmetro de perturbação de times λ.

3.2.3 Busca local

A estratégia de busca local que foi utilizada neste trabalho é a meta-heurística *Simulated Annealing*, que segundo Delahaye, Chaimatanan e Mongeau (2019), é uma das meta-heurísticas mais simples e mais utilizadas em aplicações reais até o momento.

3.2.3.1 *Simulated Annealing*

O *Annealing* é um processo técnico que visa alcançar uma estrutura homogênea em um metal através do aquecimento a uma temperatura elevada, seguido de um resfriamento lento e controlado (HAESER; RUGGIERO, 2008).

O SA foi proposto por Kirkpatrick, Gelatt e Vecchi (1983), e tem como propósito ser um algoritmo que segue as técnicas do processo de *annealing*, e é utilizado de maneira geral na solução de problemas de otimização combinatória. No Algoritmo 2 é mostrado o pseudocódigo genérico da meta-heurística SA, com os seguintes parâmetros:

- Solução inicial ($S_{inicial}$).
- Temperatura inicial ($T_{inicial}$).

- Temperatura final (T_{final}).
- Taxa de resfriamento (β).
- Número de soluções vizinhas (N_v) que são geradas a cada iteração.

Algoritmo 2: Algoritmo SA adaptado de Rocha (2013)

Entrada: Solução $S_{inicial}$, $T_{inicial}$, $T_{final} = 0.01$, β, N_v
Saída: Solução S^{Melhor}

```
1  início
2      T ← T_inicial;
3      S^Melhor ← S_inicial;
4      S^Atual ← S_inicial;
5      enquanto T > T_final faça
6          para i ← 1 até N_v faça
7              S^Vizinho ← GeraVizinho(S^Atual);
8              Δf ← f(S^Vizinho) − f(S^Atual);
9              se Δf < 0 então
10                 S^Atual ← S^Vizinho;
11                 se f(S^Vizinho) < f(S^Melhor) então
12                     S^Melhor ← S^Vizinho;
13                 fim se
14             senão
15                 Gera um número aleatório ρ ∈ [0, 1];
16                 se ρ < e^(−Δf/T) então
17                     S^Atual ← S^Vizinho;
18                 fim se
19             fim se
20         fim para
21         T ← T * β;
22     fim enquanto
23 fim
```

O algoritmo inicia com a $T_{inicial}$ alta e é gradualmente resfriado até alcançar a T_{final}. A cada iteração, para cada temperatura T, são geradas N_v soluções vizinhas. Se a solução vizinha gerada apresentar valor de f melhor do que a solução atual, ela é atualizada. No entanto, se a solução vizinha apresentar valor de f pior, ela ainda pode ser aceita com uma probabilidade calculada como $e^{-\Delta f/T}$, sendo que Δf representa a diferença entre os valores de f da solução vizinha e da solução atual, e T é a temperatura atual. Quanto maior for a diferença entre Δf e menor a temperatura T, menor será a possibilidade de aceitar uma solução vizinha. No início, quando a temperatura está alta, o algoritmo

aceita uma ampla gama de soluções. À medida que a temperatura diminui, apenas poucas soluções piores são aceitas, intensificando a busca em uma vizinhança específica. A cada iteração, a temperatura T é multiplicada pelo valor β, onde β é uma taxa com valor entre 0 e 1. O algoritmo termina quando a temperatura T for menor que T_{final}.

3.2.3.2 *Geração de Soluções Vizinhas*

A etapa de geração de vizinhos é essencial para o funcionamento do SA. Nela é definida como o espaço de soluções atual é explorado, o que determina o desempenho da busca local e a possibilidade de encontrar soluções de qualidade para o problema.

Para isso, foram utilizados os movimentos descritos nas seções 3.2.2.1, 3.2.2.2 e 3.2.2.3. A cada iteração do SA, um desses três movimentos é sorteado aleatoriamente para ser executado uma única vez, gerando assim uma nova solução vizinha (linha 7 do Algoritmo 2).

4 Resultados Obtidos

Neste capítulo são apresentados os resultados obtidos, através de experimentos computacionais, pelo método proposto. As seções seguintes detalham, respectivamente, os experimentos computacionais realizados, os resultados obtidos, e por fim a análise e a discussão desses resultados.

4.1 Experimentos Computacionais

Foram realizados experimentos computacionais para avaliar a qualidade das soluções geradas pelo método proposto. Nas subseções seguintes, são detalhados aspectos importantes desta etapa, como as instâncias utilizadas e a calibração dos parâmetros.

4.1.1 Instâncias

As instâncias utilizadas para realizar os experimentos computacionais são as instâncias reais da SBV Masculina e Feminina das temporadas 21-22, 22-23, 23-24 e 24-25, resultando em oito instâncias. Os dados destas temporadas foram extraídos e formatados conforme o modelo de leitura de instâncias apresentado na Figura 6. Portanto, para o funcionamento do algoritmo, devem ser informados: a quantidade de times, seguida do nome de cada um dos times, bem como uma matriz de distâncias com dimensão n x n, obtida calculando as distâncias rodoviárias entre as cidades sedes dos clubes através do *Google Maps*. Por fim, devem ser informados os parâmetros das restrições para a construção da tabela de jogos da SBV, composto pela quantidade máxima permitida de jogos consecutivos dentro de casa, a quantidade máxima permitida de jogos consecutivos fora de casa, e a diferença máxima entre os jogos realizados dentro e fora de casa por turno.

```
12
sada_cruzeiro
vedacit_guarulhos
itambe_minas
volei_renata
farma_conde_volei
sesi
suzano
apan_blumenau
araguari_volei
america
brasilia
rede_cuca
0     560   16    570   594   573   597   1163  546   419   729   2355
560   0     574   117   78    52    39    620   628   972   1013  3097
16    574   0     583   607   587   611   1175  560   423   734   2358
570   117   583   0     151   102   151   680   532   979   916   3001
594   78    607   151   0     93    71    695   677   1004  1062  3146
573   52    587   102   93    0     55    599   624   987   1007  3092
597   39    611   151   71    55    0     664   664   1011  1049  3133
1163  620   1175  680   695   599   664   0     1199  1588  1583  3668
546   628   560   532   677   624   664   1199  0     626   391   2476
419   972   423   979   1004  987   1011  1588  626   0     700   1942
729   1013  734   916   1062  1007  1049  1583  391   700   0     2135
2355  3097  2358  3001  3146  3092  3133  3668  2476  1942  2135  0
6
6
6
```

Figura 6 – Exemplo de formato das instâncias.

A Figura 6 apresenta, portanto, o modelo de instância para a edição masculina da temporada 22-23 da SBV.

4.1.2 Escolha dos Parâmetros

Os parâmetros das meta-heurísticas foram calibrados utilizando o *software* de configuração automática *Iterated racing package* (*iRace*) (LÓPEZ-IBÁÑEZ et al., 2016). Para isso, são definidos os intervalos possíveis para os parâmetros e fornecidas as instâncias utilizadas no problema. Com essas informações o *iRace* aplica uma série de iterações testando configurações candidatas diante do algoritmo de ajuste, e armazena as melhores configurações de parâmetros. Ao final da execução exibe os valores obtidos pela calibração, como pode ser observado na Tabela 1.

Tabela 1 – Calibração dos Parâmetros.

Parâmetro	Tipo	Intervalo	Melhor
β	Real	[0.9, 0.999]	0.9362
N_v	Categórico	{10, 100, 300, 500}	500
$T_{inicial}$	Categórico	{10, 50, 100, 200}	100
ρ	Real	[0.3, 0.9]	0.3802
γ	Real	[0.3, 0.9]	0.8537
λ	Real	[0.3, 0.9]	0.3939

O parâmetro T_{final} foi definido com o valor 0,01, conforme apresentado no Algoritmo 2.

4.1.3 Detalhes de Implementação e Execução de Testes

O algoritmo foi desenvolvido na linguagem de programação C++, compilado com *GCC Compiler* 6.3.0 e todos os testes foram executados em um computador com as configurações descritas na Tabela 2.

Tabela 2 – Ambiente Computacional.

Item	Valor
Processador	Intel Core i5 1.80 GHz
Memória ram	16GB
Sistema operacional	Windows 11 Pro

O algoritmo implementado foi executado 10 vezes para cada instância com os parâmetros calibrados descritos acima, sendo que o número de iterações do ILS (Max_{Iter}) foi substituído pelo tempo computacional de cinco minutos (300 segundos).

O código-fonte do algoritmo desenvolvido e os demais arquivos se encontram disponíveis em `https://github.com/filipepr27/tcc2`.

4.2 Resultados Computacionais

Nesta seção são apresentados os resultados computacionais obtidos a partir dos experimentos computacionais.

Os resultados obtidos para cada instância estão descritos na Tabela 3, que é dividida entre as categorias masculina e feminina. Para cada instância, tem-se o menor valor de f obtido (f_{best}) dentre as 10 execuções, a média (f_{avg}) dos resultados, o coeficiente de variação $\Delta = (((f_{agv} - f_{best}) \ / \ f_{avg}) \ * \ 100)$, calculado com base na diferença percentual entre f_{best} e f_{avg}, e por fim, o tempo (ω), em segundos, necessário para que o algoritmo encontrasse a solução f_{best}.

Tabela 3 – Resultados Obtidos pelo ILS+SA.

	Instâncias	f_{best}	f_{avg}	$\Delta(\%)$	$\omega(seg.)$
M	21-22	72.140	75.287,4	4,18	167
	22-23	125.071	127.670,1	2,04	248
	23-24	77.837	80.994,1	3,90	16
	24-25	75.932	80.882,8	6,12	227
F	21-22	70.537	72.461,9	2,66	32
	22-23	77.695	79.278,7	2,00	158
	23-24	70.436	75.764,5	7,03	96
	24-25	76.198	80.755,0	5,64	46

Diante disso, é possível observar na Tabela 3 que, para as oito instâncias analisadas, em três delas o algoritmo foi capaz de encontrar a melhor solução em menos de um minuto. Além disso, em apenas duas instâncias a melhor solução foi encontrada nos últimos 100 segundos de execução, sugerindo que, na maioria dos casos, a convergência para uma boa solução ocorreu em um estágio inicial ou intermediário da execução.

Além disso, a coluna Δ demonstra que o algoritmo foi robusto e conseguiu convergir rapidamente para um espaço de busca promissor, uma vez que a maior variação entre a média e a melhor solução encontrada foi de apenas 7,03%. Isso indica uma estabilidade na exploração de soluções, reduzindo grandes oscilações entre os valores encontrados de f_{best} e f_{avg}.

4.3 Análise e Discussão

Afim de validar o método proposto, é importante comparar a eficácia de seus resultados com os resultados apresentados atualmente pela SBV, como evidenciado pela Tabela 4. Na coluna f_{sbv} temos o valor de f calculado para a tabela (solução) proposta pela SBV em cada instância do campeonato, enquanto a coluna f_{best} apresenta os valores obtidos pelo método ILS+SA proposto neste trabalho. Por fim, a coluna $\Delta = (((f_{sbv} - f_{best}) \ / \ f_{sbv}) \ * \ 100)$ apresenta a diferença percentual entre os resultados de f_{best} e f_{sbv}.

Tabela 4 – Comparação de Resultados.

	Instâncias	f_{sbv}	f_{best}	$\Delta(\%)$
M	21-22	137.788	72.140	47,64
	22-23	207.342	125.071	39,68
	23-24	156.566	77.837	50,28
	24-25	164.309	75.932	53,79
F	21-22	137.503	70.537	48,70
	22-23	147.048	77.695	47,16
	23-24	139.873	70.436	49,64
	24-25	149.542	76.198	49,05
Média		154.996,3	71.715,4	48,24

Dentre as instâncias, destaca-se a 22-23 da edição masculina. O valor de f_{sbv} elevado se deve ao fato de ser o único campeonato que contava com a participação de um time (Rede Cuca) da região Nordeste, com sede em Fortaleza/CE. Com isso, os demais times constantemente realizavam viagens mais longas do que as habituais, saindo do eixo majoritariamente composto por times das regiões Sul e Sudeste, elevando a distância total percorrida pelas equipes. Já para a Rede Cuca, um cronograma mal otimizado ocasionou em um gasto de tempo e recurso superior aos seus adversários, que dispuseram de uma janela maior para treino, preparação e recuperação dos atletas. Com o método proposto apresentando um valor de f quase 40% melhor, a SBV poderia implementar uma tabela mais justa e ponderada para todas as equipes, minimizando os possíveis impactos causados pela distância geográfica. Vale ressaltar que a Rede Cuca amargou a última colocação da SBV naquela edição, e consequentemente foi rebaixada, não retornando ao campeonato desde então.

Além disso, todas as instâncias apresentaram uma melhoria significativa de f quando comparadas à solução atualmente utilizada pela SBV. Mesmo na edição em que f_{best} mais se aproximou de f_{sbv}, a solução obtida ainda foi aproximadamente 40% melhor. Além disso, no melhor dos casos, o ILS+SA gerou uma solução quase 54% superior à adotada pela SBV, sendo que, em média, os resultados do algoritmo se mostraram 48,4% mais eficientes que a solução usada pela SBV.

Portanto, é possível afirmar que o algoritmo proposto para a resolução do PTHJ da SBV apresentou-se extremamente satisfatório, obtendo em todos os casos soluções melhores do que as dispostas atualmente. Além da qualidade das soluções, o tempo de execução demonstrou-se prático e viável, tornando o método adequado para aplicações reais.

5 Conclusão

Neste trabalho foi abordado o Problema de Tabela-Horário de Jogos (PTHJ) da Superliga Brasileira de Voleibol (SBV). Foi proposto um método que utiliza a meta-heurística ILS juntamente com a meta-heurística *Simulated Annealing* na etapa de busca local. Aplicando três movimentos de perturbação baseados na troca de ordem entre as rodadas, a inversão do mando de campo das partidas e a permutação de times nas rodadas, buscou-se uma ampla exploração do espaço de soluções para que o algoritmo conseguisse encontrar soluções de qualidade.

O método proposto gerou resultados consistentemente superiores em comparação com os atualmente utilizados pela SBV, alcançando uma melhoria média de 48,24% e no melhor cenário, de 53,70%. Estes ganhos refletem uma melhoria significativa para todas as equipes do campeonato, reduzindo as distâncias e o tempo gasto com o deslocamento para as partidas, permitindo uma alocação mais eficiente dos recursos e promovendo uma melhoria no cenário competitivo da liga.

Deste modo, os resultados obtidos reforçam a relevância e a aplicabilidade de métodos para otimizar a tabela de jogos em campeonatos, destacando seu potencial para futuras implementações e aprimoramentos.

Referências

ANAGNOSTOPOULOS, A. et al. A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, v. 9, n. 2, p. 177–193, 2006.

BAO, R. *Time relaxed round robin tournament and the NBA scheduling problem*. Tese (Doutorado em Engenharia Industrial) — Cleveland State University, 2009.

BIAJOLI, F. L. *Novas heurísticas para o problema de geração de escalas de jogos para torneios esportivos*. Dissertação (Mestrado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais, 2007.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, v. 35, n. 3, p. 268—308, 2003.

BOULEIMEN, K.; LECOCQ, H. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European journal of operational research*, v. 149, n. 2, p. 268–281, 2003.

COLBOURNE, C.; DINITZ, J. *Handbook of combinatorial designs*. Boca Raton, FL, USA: CRC press, 2007.

COSTA, F. N.; URRUTIA, S.; RIBEIRO, C. C. An ILS heuristic for the traveling tournament problem with predefined venues. *Annals of Operations Research*, v. 194, n. 1, p. 137–150, 2012.

DELAHAYE, D.; CHAIMATANAN, S.; MONGEAU, M. Simulated annealing: From basics to applications. In: GENDREAU, M.; POTVIN, J.-Y. (Ed.). *Handbook of Metaheuristics*. Cham, China: Springer International Publishing, 2019. p. 1–35.

EASTON, K.; NEMHAUSER, G.; TRICK, M. The traveling tournament problem description and benchmarks. In: *Principles and Practice of Constraint Programming — CP 2001*. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2001. p. 580–584.

HAESER, G.; RUGGIERO, M. G. Aspectos teóricos de simulated annealing e um algoritmo duas fases em otimização global. *Trends in Computational and Applied Mathematics*, v. 9, n. 3, p. 395–404, 2008.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.

LÓPEZ-IBÁÑEZ, M. et al. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, v. 3, p. 43–58, 2016.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of metaheuristics*. Boston, MA, USA: Springer, 2003. p. 320–353.

RIBEIRO, C. C. Sports scheduling: Problems and applications. *International Transactions in Operational Research*, 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*, v. 179, n. 3, p. 775–787, 2007.

ROCHA, W. S. *Algoritmo GRASP para o Problema de Tabela-horário de Universidades*. Dissertação (Mestrado em Informática) — Universidade Federal do Espírito Santo, 2013.

RUSSELL, K. Balancing carry-over effects in round robin tournaments. *Biometrika*, v. 67, n. 1, p. 127–131, 1980.

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

SOUZA, M. J. F. et al. Programação de jogos da primeira divisão do campeonato brasileiro de futebol por meio da metaheurística iterated local search. In: *Anais do Simpósio de Pesquisa Operacional e Logística da Marinha*. Rio de Janeiro, RJ, Brasil: CASNAV, 2005. v. 8.

STÜTZLE, T. *Applying iterated local search to the permutation flow shop problem*. [S.l.], 1998.