

# Heuristics for the Mirrored Tournament Traveling Tournament Problem Based on the Home-Away Swap Neighborhood

Bruno Guilera, Italo Teixeira e Tiago Januario  
Computational Intelligence and Optimization Research Lab (CInO)  
Departamento de Ciência da Computação  
Universidade Federal da Bahia  
Avenida Adhemar de Barros, s/n, Salvador, BA 40170-110, Brazil  
Email: brunoguilera@hotmail.com, italo95teixeira@gmail.com, januario@ufba.br

**Abstract**—In this paper we present a detailed experimental analysis of the existing neighborhoods in the literature using a few quality criteria and propose new heuristics for the Mirrored Traveling Tournament Problem. Based on the Home-Away Swap, the proposed heuristics are able to search the solution space from an infeasible solution and obtaining a feasible one in the end. We also describe a strategy for reduce the amount of visited neighborhoods in the local search process without drastically affect the solution, consequently we managed to obtain feasible solutions with reduced computation time, as shown through empirical analysis.

**Index Terms**—Combinatorial optimization, Graph theory, Heuristics, Sports scheduling

## 1. Introdução

A organização de grandes eventos esportivos é uma atividade complexa e que envolve a participação de profissionais de diversas áreas. Tabelas esportivas mal planejadas podem afetar o desempenho e a arrecadação dos times envolvidos devido ao excesso de viagens, pois implica em custos elevados de deslocamento para os atletas, comissão técnica, imprensa, torcedores e outras pessoas envolvidas. Tendo em vista a importância da elaboração de tabelas esportivas que minimizem a distância total viajada, este problema vêm ganhando destaque como tema de pesquisa.

O Problema do Torneio com Viagens (TTP), do inglês *Traveling Tournament Problem*, é um dos problemas de programação de tabelas esportivas mais conhecidos na literatura [1]. Dado um número  $n$  (par) de times e uma matriz de distâncias  $D_{n \times n}$ , onde cada elemento  $d_{ij} \in D$  contém a distância entre as cidades sedes dos times  $i$  e  $j$ , tem-se como objetivo definir uma tabela de jogos tal que a distância total percorrida pelos times seja a menor possível. Variações do TTP podem ser encontradas em [2] e [3]. Torneios esportivos podem ser classificados como simples, também chamados de torneios *Single Round Robin* (SRR), em que cada time joga com cada um dos seus oponentes uma única vez em  $n - 1$  rodadas. Já nos torneios duplos, chamados de torneios *Double Round Robin* (DRR), cada time joga exatamente duas vezes contra cada um dos seus

oponentes em  $2(n - 1)$  rodadas. Quando um jogo é disputado na sede de uma das duas equipes que se enfrentam, uma das equipes joga em casa e a outra fora de casa. A equipe que joga em casa tem o mando de campo e é chamada de anfitrião, enquanto a outra equipe é chamada de visitante.

O Problema do Torneio com Viagens Espelhado (MTTP), do inglês *Mirrored Traveling Tournament Problem*, apresenta a estrutura de um torneio DRR dividido em dois turnos, sendo considerado um torneio SRR em suas  $n - 1$  primeiras rodadas, ou primeiro turno, seguido por outro torneio SRR nas  $n - 1$  rodadas seguintes, segundo turno, com os jogos na mesma ordem, porém com as sedes dos jogos invertidas, também chamadas de espelhadas, sendo um tipo de torneio muito comum na América Latina [2].

Em geral, abordagens heurísticas para problemas de torneios com viagens são aplicadas em duas etapas distintas: (i) definição dos adversários para cada time e em qual rodada os confrontos irão acontecer; e (ii) atribuição de sede para cada confronto definido na etapa anterior. A Tabela 1 apresenta uma tabela de jogos com as definições dos adversários e atribuições de sedes para um torneio com  $n = 4$  times composto por  $2 \times (4 - 1) = 6$  rodadas, onde a primeira coluna corresponde aos times pertencentes ao conjunto  $T = \{A, B, C, D\}$  e as demais colunas correspondem às rodadas, cada linha contém os adversários dos times indicados na primeira coluna. O símbolo “+” indica que o jogo será realizado em casa, enquanto o símbolo “-” indica que o jogo será realizado na sede do adversário. A distância total é dada pela soma das distâncias percorridas pelos times ao longo do torneio. A partir da Tabela 1 podemos observar que a distância percorrida pelo time  $D$  pode ser expressa por  $d_{DC} + d_{CB} + d_{BA} + d_{AD}$ .

Tabela 1. TABELA DE JOGOS PARA UM TORNEIO COM 4 TIMES.

	1º Turno			2º Turno		
	1	2	3	4	5	6
A	-D	+B	-C	+D	-B	+C
B	+C	-A	+D	-C	+A	-D
C	-B	+D	+A	+B	-D	-A
D	+A	-C	-B	-A	+C	+B

Este trabalho apresenta também um estudo da vizinhança

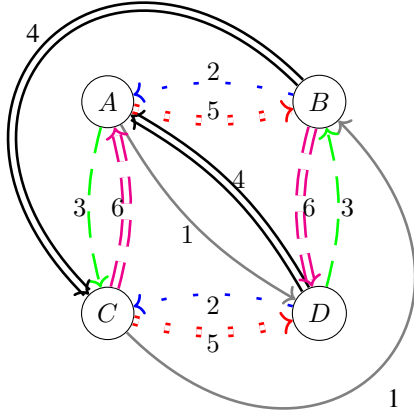


Figura 1. Representação em grafo da Tabela 1

Troca de Times e Rodadas (TARS), do inglês *Teams and Rounds Swap*, definida em [3]. A vizinhança TARS foi avaliada no Problema do Torneio com Viagens com Estádios Pré Definidos, proposto por [4] e [5], e para o Problema de Minimização do Efeito Carry-Over, investigado por [6] e [7], mas até o momento, nunca foi analisada no contexto do MTTP. Como contribuição deste trabalho, também são propostas duas novas heurísticas para o MTTP utilizando a combinação de vizinhanças existentes baseadas na vizinhança Troca de Anfitriões.

Este trabalho está organizado como segue: uma descrição do MTTP e sua modelagem como um problema de coloração de grafos é dada na Seção 2. Na Seção 3 são descritas as vizinhanças mais conhecidas na literatura. A estrutura TARS é definida e adaptada ao MTTP na Seção 4. Uma breve explicação da estratégia para percurso no espaço de soluções é apresentada na Seção 5. As novas heurísticas propostas são apresentadas na Seção 6. Na Seção 7 são elucidados alguns experimentos computacionais juntamente com algumas discussões. Por fim, apresentamos algumas considerações finais na Seção 8.

## 2. O Problema do Torneio com Viagens Espelhado

O MTTP trata-se de um problema de otimização combinatoria que visa obter uma programação de times em tabelas esportivas de forma a reduzir a distância total percorrida pelos times obedecendo às seguintes restrições:

- R1:** Nenhum time pode jogar mais de três jogos consecutivos em casa ou três jogos consecutivos fora de casa;
- R2:** As rodadas são igualmente divididas entre dois turnos e os times se enfrentam na mesma ordem em ambos os turnos;

No MTTP, uma tabela de jogos é viável quando satisfaz às restrições R1 e R2, caso contrário, tal solução é inviável. Para a restrição R1, a contagem de violações é atualizada a cada nova ocorrência de jogo consecutivo. Por exemplo, uma solução onde um time possui uma sequência de cinco

jogos consecutivos em casa é considerada uma solução com 2 violações da restrição R1. A solução é dita ótima quando for viável e o somatório das distâncias percorridas por todos os times durante o campeonato for o menor dentre todas as soluções do problema.

O MTTP pode ser modelado como um problema de coloração de arestas em um grafo completo orientado  $G(V, E)$ , combinado com o problema de atribuição de mandos de campo, tal que: (i) os vértices  $\{v_1, v_2, \dots, v_n\}$  representam os times; (ii) cada arco  $a_{ij} = (v_i, v_j)$  representa o confronto no qual o time  $v_i$  viaja até a sede do time  $v_j$ ; (iii) existe uma atribuição de cores  $\sigma: (v_i, v_j) \rightarrow c$ , que representa a rodada na qual este confronto irá acontecer; (iv) existe uma atribuição de pesos positivos  $(v_i, v_j) \rightarrow d_{ij}$ , que representa a distância entre as sedes  $v_i$  e  $v_j$ . O conjunto de cores  $C = \{c_1, c_2, \dots, c_{2n-3}, c_{2n-2}\}$  é mínimo e ordenado, onde os índices  $i = \{1, 2, \dots, 2n-3, 2n-2\}$  denotam a ordem em que as rodadas irão acontecer. Pela propriedade espelhada do torneio pode-se perceber que as rodadas  $i$  e  $i + (n-1)$  são espelhadas, isto é, representam os mesmos confrontos porém em turnos distintos. Denotaremos como  $\bar{c}$  a cor espelhada da cor  $c$ , onde as arestas associadas possuem as mesmas extremidades, porém com orientação invertida.

A Figura 1 apresenta, por meio de um grafo, o torneio envolvendo 4 times descrito na Tabela 1. Aqui, os vértices correspondem aos times, as cores representam às rodadas, os arcos indicam os confrontos e suas direções indicam os mandos de campo. Na rodada 3, por exemplo, podemos observar que os arcos são coloridos com a cor 3 e suas direções indicam os mandos de campo nos confrontos, ou seja, o time A joga na sede do time C enquanto o time B joga em casa contra o time D na mesma rodada.

A seguir descrevemos os procedimentos utilizados para a transformação de uma dada solução de modo a percorrer de forma sistemática o conjunto de soluções do problema, semelhante à descrição das vizinhanças encontrada no trabalho de [8]. Neste texto, iremos utilizar a modelagem em grafos para auxiliar na descrição das vizinhanças existentes.

## 3. Vizinhanças

Uma vizinhança é um mapeamento que atribui para cada solução  $s$  um conjunto de soluções vizinhas  $N(s)$ . Procedimentos de busca local usam o conceito de vizinhanças para mover-se de uma solução  $s$  para uma outra solução vizinha  $s' \in N(s)$  [3]. Nesta seção, todas as estruturas de vizinhanças foram modeladas e definidas com o auxílio de conceitos de teoria dos grafos e coloração de arestas.

Dado um torneio DRR espelhado com  $n$  times, uma vizinhança pode ser construída a partir do seguinte conjunto de movimentos:

- *Troca de Anfitriões* - HAS (do inglês *Home-Away Swap*)
  - Efeito: para quaisquer dois times distintos, inverte-se a atribuição dos mandos de campos nos confrontos entre eles.

- No grafo: para quaisquer dois vértices distintos  $v$  e  $w$ , invertem-se as orientações dos arcos  $(v, w)$  e  $(w, v)$ .
- **Troca de Times - TS** (do inglês *Team Swap*)
  - Efeito: para quaisquer dois times distintos, trocam-se todos os oponentes entre eles, assim como as atribuições dos mandos de campo, em todas as rodadas exceto a rodada na qual eles se enfrentam.
  - No grafo: para quaisquer dois vértices distintos  $v$  e  $w$ , todos os arcos  $(v, x)$  e  $(x, v)$  passam a ser coloridos pelas cores de  $(w, x)$  e  $(x, w)$  respectivamente, e vice-versa, onde  $x \in V - \{v, w\}$ .
- **Troca de Rodadas - RS** (do inglês *Round Swap*)
  - Efeito: para quaisquer duas rodadas distintas de um mesmo turno, trocam-se todas as atribuições de oponentes entre elas, assim como as atribuições dos mandos de campo, para todos os times em ambos os turnos.
  - No grafo: para quaisquer duas cores distintas  $c$  e  $k$ , com índices  $i, j$ , tal que,  $|i - j| < n - 1$ , todos os arcos coloridos pelas cores  $c$  e  $\bar{c}$  passam a ser coloridos pelas cores  $k$  e  $\bar{k}$ , e vice-versa.
- **Troca Parcial de Times - PTS** (do inglês *Partial Team Swap*)
  - Efeito: para quaisquer dois times distintos e uma rodada diferente daquela em que esses times se enfrentam, trocam-se o menor número de oponentes comuns destes dois times nesta rodada e nas demais rodadas que forem necessárias para que todas as trocas envolvidas sejam consistentes.
  - No grafo: para quaisquer dois vértices distintos  $v$  e  $w$  e uma cor  $c \neq \sigma(v, w)$  e  $c \neq \sigma(w, v)$ , seja  $C$  o conjunto mínimo de todas as cores de  $G(V, E)$ , existe um subconjunto  $\Omega \subseteq C$ , tal que,  $\bigcup \sigma(v, x) \cup \sigma(x, v) = \Omega = \bigcup \sigma(w, x) \cup \sigma(x, w)$  com  $x \in V \setminus \{v, w\}$  e a cardinalidade de  $\Omega$  é mínima. Todos os arcos  $(v, x)$  e  $(x, v)$  passam a ser coloridos pelas cores de  $(w, x)$  e  $(x, w)$ , e vice-versa.
- **Troca Parcial de Rodadas - PRS** (do inglês *Partial Round Swap*)
  - Efeito: para quaisquer duas rodadas distintas de um mesmo turno e um time qualquer, trocam-se os oponentes e as atribuições de mando de campo deste time entre estas duas rodadas e dos demais times que forem necessários para que todas as rodadas sejam consistentes.
  - No grafo: para quaisquer duas cores distintas  $c$  e  $d$ , com índices  $i, j$ , tal que,  $|i - j| < n - 1$ ,

e um vértice  $v$ , existem dois ciclos, com sentidos inversos, que contém  $v$ . No primeiro os arcos são alternadamente coloridos pelas cores  $c$  e  $d$ , no segundo os arcos são alternadamente coloridos pelas cores  $\bar{c}$  e  $\bar{d}$ . Sendo assim inverte-se as atribuições de cores dos arcos presentes nestes ciclos.



Figura 2. Estrutura do PRS isomorfa ao RS para as cores 2 e 3. À esquerda a estrutura antes do movimento de troca, à direita após o movimento.

As Figuras 2 e 3 ilustram as representações das vizinhanças TS, PTS, RS e PRS em grafos e aplicadas ao torneio definido na Tabela 1 e modelado na Figura 1. A Figura 2 ilustra um movimento do PRS para as cores/rodadas 2 e 3 e o vértice/time  $x \in \{A, B, C, D\}$ . Como todos os times estão envolvidos, neste caso, o movimento será idêntico ao movimento da vizinhança RS.

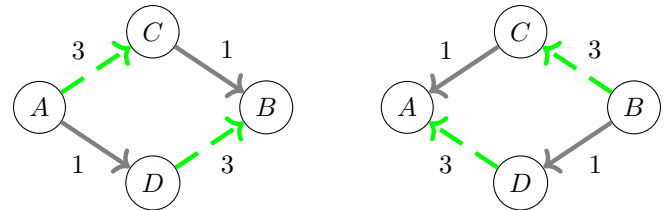


Figura 3. Estrutura do PTS isomorfa ao TS para os times A e B. À esquerda a estrutura antes do movimento de troca, à direita após o movimento.

A Figura 3 ilustra um movimento do PTS para os vértices/times A e B e a cor/rodada  $c \in \{1, 3\}$ . Como todos os times estão envolvidos, neste caso, o movimento será idêntico ao movimento da vizinhança TS.

#### 4. Troca de Times e Rodadas

Este trabalho apresenta a primeira aplicação da vizinhança Troca de Times e Rodadas - TARS (do inglês *Teams and Rounds Swap*), proposta por Januario e Urrutia [3], para o MTP. Nesta sessão descreveremos sua estrutura e sua modelagem para o problema. A vizinhança TARS combina características da PTS e da PRS e pode ser dividida em duas subestruturas, aqui chamadas de colar e diamante. Na modelagem da estrutura iremos usar a propriedade “espelho” do torneio e iremos considerar apenas as cores que representam as rodadas de um dos turnos. Sendo assim, para aplicar as trocas dos times e rodadas, basta espelhar as modificações para o outro turno.

Dados  $c, d \in C$  e  $v \in V$ , tal que,  $c \neq d$ , as subestruturas são definidas de forma recursiva como segue:

## 4.1. Colar

A estrutura colar, como ilustrada na Figura 4, pode ser vista como uma cadeia de Kempe [9], tendo  $v$  como vértice no centro da cadeia e as cores  $c$  e  $d$  alternadas ao longo da cadeia. Seja a função  $\Upsilon : V \times C \rightarrow V$ , onde  $\Upsilon(v, c)$  retorna o vértice adjacente a  $v$  pelo arco de cor  $c$ . A partir de  $v$  são determinados os vértices  $v_1, v_2 \in V$ , tais que,  $v_1 = \Upsilon(v, c)$  e  $v_2 = \Upsilon(v, d)$ , dessa forma é possível construir um grafo  $G_1(\{v, v_1, v_2\}, \{(v, v_1), (v, v_2)\})$  com atribuições de cores para os arcos,  $\{(v, v_1) \rightarrow c, (v, v_2) \rightarrow d\}$ . Dizemos o colar tem nível  $k$  se sua cadeia tem comprimento  $2 \cdot k$ , sendo  $k = n/2$  o nível máximo que o colar pode assumir.

A cada iteração são encontrados os times  $w_i, w_j$  tais que  $w_i = \Upsilon(v_i, c)$  e  $w_j = \Upsilon(v_j, d)$  com as cores  $c$  e  $d$  sendo invertidas a cada expansão. Caso  $w_i \neq w_j$  então o colar é aberto e  $w_i, w_j$  representam suas pontas; caso contrário então o colar é fechado e  $v_e = w_i = w_j$  representa o vértice fecho do colar. Quando o colar é fechado significa que todos os times estão envolvidos na vizinhança, por conta disso, sua estrutura é isomorfa à estrutura de vizinhança PRS, ou seja, nesta condição o efeito de PRS e TARS sobre uma solução será o mesmo.

Generalizando, temos que para cada tupla  $(v, c, d)$  existe um conjunto de colares  $C_o = \bigcup_{k=1}^{n/2-1} G_k$  onde

$$G_k = \begin{cases} G_1 & \text{se } k = 1 \\ G_{k-1} \cup G^* & \text{se } k > 1 \text{ e } G_{k-1} \text{ for aberto} \\ G_{k-1} & \text{se } k > 1 \text{ e } G_{k-1} \text{ for fechado} \end{cases}$$

tal que

$$G^* = \{V^*, E^*, C^*\}$$

$$V^* = \begin{cases} \{w_i, w_j\} & \text{se } w_i \neq w_j \\ \{v_e\} & \text{se } w_i = w_j \end{cases}$$

$$E^* = \{(v_i, w_i), (v_j, w_j)\}$$

$$C^* = \{(v_i, w_i) \rightarrow c, (v_j, w_j) \rightarrow d\}$$

## 4.2. Diamante

Para cada colar  $G_k \in C_o$  aberto, constrói-se um diamante, ilustrada pela Figura 5, como segue: A partir das extremidades do colar, com os vértices  $w_i$  e  $w_j$ , utilizando as cores  $c, d$ , são determinados os vértices  $v_1$  e  $v_2$  tais que  $v_1 = \Upsilon(w_i, d)$  e  $v_2 = \Upsilon(w_j, c)$ , em seguida são encontrados as cores  $e = \sigma(v_1, w_j)$  e  $f = \sigma(v_2, w_i)$ . Novamente serão encontrados vértices  $v_3 = \Upsilon(w_i, e)$  e  $v_4 = \Upsilon(w_j, f)$  e cores  $g = \sigma(v_3, w_j)$  e  $h = \sigma(v_4, w_i)$ , iterativamente, até  $v_i = v_j$ .

Seja  $V_i$  o conjunto de vértices encontrados a partir de  $w_i$  e  $V_j$  o conjunto de vértices encontrados a partir de  $w_j$ , caso exista  $\sigma(V_i, w_j) = c$  ou  $\sigma(V_j, w_i) = d$  então não é possível determinar um diamante para  $G_k$ , pois  $V_i \cap V_j = \emptyset$ .

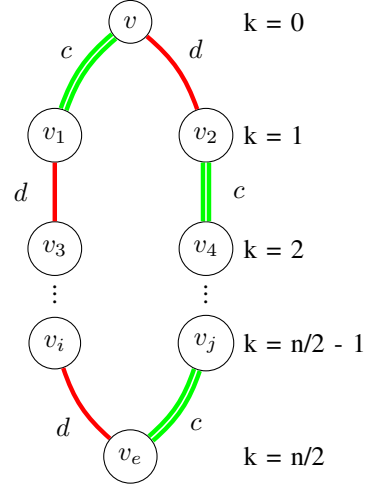


Figura 4. Estrutura do colar de  $k$  níveis, com os parâmetros  $(v, c, d)$ .

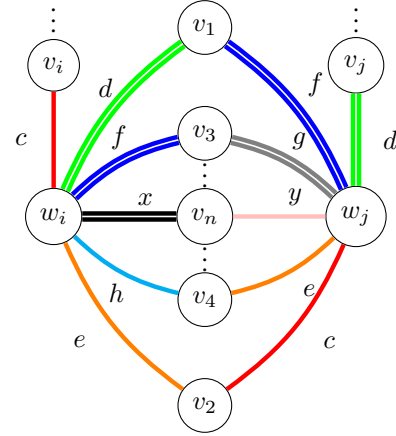


Figura 5. Estrutura do diamante.

A partir da estrutura diamante é encontrado o subconjunto  $V_d \in V$  de vértices, tal que,  $|V_d|$  é mínimo e definido recursivamente da seguinte forma:

$$V_d = \begin{cases} \Upsilon(w_j, c) \in V_d \\ \Upsilon(w_i, d) \in V_d \\ \text{se } u \in V_d \rightarrow \Upsilon(w_i, \sigma(u, w_j)) \in V_d \quad \forall \sigma(u, w_j) \neq c, d \end{cases}$$

Generalizando temos que para cada colar aberto  $G_k \in C_o$  é possível definir um diamante como um grafo bipartido completo  $G_{kd}$  com partição  $\{w_i, w_j\} \cup V_d$  com arestas coloridas de tal forma que exceto as cores  $c$  e  $d$ , todas as demais cores estejam presentes duas vezes, incidindo nos vértices  $\{w_i, w_j\}$ .

## 5. Espaço de Soluções

Para utilizarmos procedimentos de busca local para resolver o MTTP iremos primeiro definir o espaço das

soluções como uma relação de ordem entre as soluções obtidas e a solução inicial. Definiremos o espaço de soluções  $S$ , como o conjunto de soluções que podem ou não violar a restrição R1.

Definimos uma relação de ordem utilizando duas variáveis pertinentes a cada solução: a *distância total* e a *violação* e as representamos como:  $s.dist$  e  $s.vio$ , respectivamente. Podemos então definir o operador “<” (menor que) para o conjunto  $S$  como:

$\forall s', s'' \in S$ , dizemos que  $s' < s''$  se, somente, se:

- $s'.vio < s''.vio$ , ou
- $s'.vio = s''.vio \wedge s'.dist < s''.dist$

Com isso, damos prioridade a encontrar uma solução viável e em segundo plano encontrar uma solução que reduza a distância.

A solução inicial é gerada em duas etapas: primeiramente, definimos os confrontos entre os times usando o algoritmo de Vizing, descrito em [3], [10] e [11]. Em seguida, definimos os mandos de campo para cada confronto de forma aleatória. Neste trabalho foram consideradas as estratégias Primeiro Aprimorante (do inglês *First Improvement*) e Melhor aprimorante (do inglês *Best Improvement*), ambas definidas em [12].

## 6. Melhorias na Vizinhança Troca de Anfitriões

Supondo que dada uma solução com violações é possível realizar uma sequência de trocas de anfitriões e encontrar uma solução viável mantendo a mesma atribuição de oponentes. Uma maneira de encontrar tal solução é através da heurística de busca local usando a estratégia melhor aprimorante associada a vizinhança HAS, que daqui em diante chamaremos de HAS\*. Nesta seção propomos duas novas heurísticas: X-HAS e HAS++. O X-HAS é uma composição de heurísticas que combina a heurística primeiro aprimorante de umas das vizinhanças PTS, PRS ou TARS com o HAS\*. O HAS++ é uma derivação do HAS que tem como propósito reduzir o tempo computacional do procedimento de busca local.

Na estratégia melhor aprimorante utilizada no HAS\*, todos os  $n(n-1)/2$  pares de times são submetidos a uma troca do tipo HAS para descobrir qual, dentre estas trocas, é a melhor, se existir, para aplicar na solução corrente. Ou seja, mover-se da solução corrente até o melhor vizinho até que não haja vizinhos aprimorantes. A cada interação são analisados  $\Theta(n^2)$  pares de times.

X-HAS é definida como a composição de duas buscas locais, sendo correspondente à execução de uma busca local na vizinhança  $X \in \{PTS, PRS, TARS\}$ , seguida de uma busca local na vizinhança HAS\*. Desta maneira, a partir de uma solução  $s$  podemos realizar “longos percursos” dentro do espaço de soluções a partir de soluções inviáveis até soluções viáveis. É importante observar que o custo computacional desta heurística se torna elevado, uma vez que a busca encadeada pela vizinhança HAS\* implica num número maior de soluções avaliadas.

Com o intuito de mitigar o incremento de tempo de execução do X-HAS, analisou-se uma maneira de reduzir a quantidade de vizinhos avaliados pelo HAS\* e consequentemente o seu tempo de execução, sem afetar drasticamente a qualidade da busca local. A chamada do HAS\* tem como objetivo reduzir as possíveis violações geradas pelo o movimento da vizinhança X, portanto apenas os times que foram envolvidos na troca podem gerar novas violações. O HAS++ se propõe a avaliar apenas as trocas de anfitrião que envolvem os times que foram afetados nas trocas realizadas nas buscas locais, dessa forma descartamos avaliações desnecessárias.

Para que o HAS++ seja utilizado, é necessário que este sempre seja precedido de um movimento do tipo PTS, PRS ou TARS. Somente desta forma é possível detectar os times que foram envolvidos nos movimentos das vizinhanças e registrar as possíveis formações de violações.

O funcionamento da heurística se dá da seguinte maneira:

- Na execução do movimento prévio é definido o conjunto:  $repair = \{(x, y) \mid x \neq y \in T\}$ ; onde  $T$  é o conjunto de todos os times e o par  $(x, y)$  representa o jogo entre os times  $x$  e  $y$  envolvidos no movimento;
- A partir do conjunto  $repair$  será construído o conjunto:  $filter = \{p \mid \forall r \in repair; p \in seq(r)\}$ ; onde  $seq$  representa o conjunto dos jogos com atribuições de mando de campo que violam a restrição R1;
- A vizinhança HAS++ então é avaliada tendo o conjunto  $filter$  como parâmetro, restringindo assim o número de jogos avaliados. Todos os confrontos de  $filter$  serão analisados usando a estratégia melhor aprimorante.

A principal diferença entre as vizinhanças HAS++ e HAS\* está na quantidade de pares de times que precisam ser analisados. A vizinhança HAS++ avalia um número  $O(n)$  de soluções por iteração, já que a quantidade de pares de times envolvidos pelos movimentos PTS, PRS e TARS são no máximo  $2(n-1)$ ,  $2(n/2)$  e  $2(n-1) + 2(n/2)$ , respectivamente.

Se após a avaliação da vizinhança HAS++ a solução resultante ainda for violada é possível então criar um novo conjunto:  $filter' = \{p' \mid \forall p \in filter; p' \in seq(p)\}$  e fazer uma invocação recursiva do HAS++ considerando o conjunto  $filter'$  como parâmetro. Esse procedimento pode ser repetido enquanto for possível reduzir as violações.

## 7. Resultados

Nesta seção apresentaremos resultados de experimentos realizados a fim de avaliar as estruturas de vizinhanças assim como as heurísticas propostas neste trabalho.

As instâncias utilizadas para a realização dos experimentos se encontram disponíveis na página web mantida por Michael Trick [13]. As classes das instâncias utilizadas foram:

- Circulares (*circN*);
- Baseadas na US National Football League (*nflN*);
- Baseadas no Campeonato Brasileiro de Futebol de 2013 (*cbN*).

Com  $N$  sendo o número de times. Os valores médios foram obtidos a partir de 100 execuções de cada uma das heurísticas de busca local com soluções iniciais geradas como descrito na Seção 5.

### 7.1. Análise Inicial das Vizinhanças

As tabelas 2-5 exibem resultados de testes que traçam um comparativo entre as estruturas de vizinhanças quando combinadas com uma estratégia de busca local utilizando o critério primeiro aprimorante ou melhor aprimorante, denotadas por *first* e *best* respectivamente.

Dadas duas estruturas de vizinhanças  $A$  e  $B$ , os seguintes critérios foram definidos, baseados nos critérios propostos em [14], a fim de avaliar as vizinhanças de acordo com:

- 1) **Eficiência:**  $A$  é mais eficiente que  $B$  se  $A$  consegue encontrar soluções ótimas locais melhores que  $B$ .
- 2) **Convergência:**  $A$  é mais convergente que  $B$  se  $A$  consegue encontrar soluções ótimas locais com menos passos que  $B$ , ou seja, com menor número de vizinhos visitados.
- 3) **Força:**  $A$  é mais forte que  $B$  se a variação média dos valores de *violações* e *distância total* da solução corrente para a melhor solução vizinha definida pela vizinhança  $A$  for maior que a variação média definida por  $B$ .
- 4) **Aprimoramento:**  $A$  é mais aprimorável que  $B$  se a quantidade média de vizinhos aprimorantes, ou seja, soluções vizinhas melhores que a solução corrente, encontrados por  $A$  a cada passo da busca, for maior que a quantidade média de vizinhos aprimorantes encontrados por  $B$ .

Em cada tabela os melhores resultados encontrados estão representados em negrito.

A Tabela 2 apresenta os resultados de *Convergência* e *Eficiência* das vizinhanças combinadas com uma estratégia de busca local. A primeira coluna apresenta os nomes das instâncias consideradas na avaliação. A segunda coluna apresenta as diversas vizinhanças exploradas, juntamente com sua estratégia de avaliação indicada em subscrito. A terceira coluna denotada por *Passos* mostra o valor médio de passos necessários pelo algoritmo de busca local para encontrar a solução ótima local. As duas colunas seguintes, *Violações* e *Distância*, exibem o valor médio de violações e distância total das soluções ótimas locais. Os valores entre parênteses representam o melhor resultado encontrado. Podemos constatar que a vizinhança PRS converge com menos passos que as outras vizinhanças, no entanto, não há a garantia de que a solução ótima encontrada seja viável. As vizinhanças PTS e TARS garantem uma solução viável, ao final da busca, sendo a última capaz de encontrar os melhores resultados com menos passos que a PTS.

Tabela 2. TESTE DE EFICIÊNCIA E CONVERGÊNCIA

Instância	Vizinhança	Passos	Violações	Distância
circ10	PTS_best	9,71	<b>0 (0)</b>	347,14 (318)
	PTS_first	22,46	0,01 (0)	347,36 (320)
	PRS_best	<b>6,77</b>	1,91 (0)	373,62 (336)
	PRS_first	13,05	2,08 (0)	374,46 (332)
	TARS_best	8,91	<b>0 (0)</b>	340,18 ( <b>314</b> )
	TARS_first	21,83	<b>0 (0)</b>	<b>339,64 (314)</b>
circ12	PTS_best	14,83	<b>0 (0)</b>	588,16 (552)
	PTS_first	36,13	<b>0 (0)</b>	588,30 (542)
	PRS_best	<b>10,02</b>	2,60 (0)	642,86 (594)
	PRS_first	20,69	2,62 (0)	641,42 (578)
	TARS_best	13,40	<b>0 (0)</b>	<b>571,98 (526)</b>
	TARS_first	34,87	<b>0 (0)</b>	573,52 (536)
circ14	PTS_best	20,33	<b>0 (0)</b>	928,32 (858)
	PTS_first	53,48	<b>0 (0)</b>	929,88 (876)
	PRS_best	<b>13,43</b>	3,04 (0)	1.021,12 (948)
	PRS_first	29,43	3,14 (0)	1.016,86 (956)
	TARS_best	17,65	<b>0 (0)</b>	906,82 ( <b>842</b> )
	TARS_first	48,68	<b>0 (0)</b>	<b>905,06 (848)</b>
circ16	PTS_best	25,39	<b>0 (0)</b>	1.378,06 (1.312)
	PTS_first	71,06	<b>0 (0)</b>	1.376,50 (1.302)
	PRS_best	<b>18,11</b>	2,78 (0)	1.513,82 (1.394)
	PRS_first	38,36	3,24 (0)	1.515,52 (1.410)
	TARS_best	23,58	<b>0 (0)</b>	1.339,36 ( <b>1.246</b> )
	TARS_first	65,23	<b>0 (0)</b>	<b>1.338,44 (1.254)</b>
circ18	PTS_best	34,43	<b>0 (0)</b>	1.945,02 (1.854)
	PTS_first	91,74	<b>0 (0)</b>	1.943,50 ( <b>1.812</b> )
	PRS_best	<b>22,78</b>	2,44 (0)	2.165,74 (2.024)
	PRS_first	49,50	2,63 (0)	2.156,04 (2.030)
	TARS_best	29,65	<b>0 (0)</b>	1.899,64 (1.820)
	TARS_first	84,21	<b>0 (0)</b>	<b>1.895,20 (1.812)</b>
circ20	PTS_best	41,81	<b>0 (0)</b>	2.657,60 (2482)
	PTS_first	118,16	<b>0 (0)</b>	2.650,16 (2470)
	PRS_best	<b>29,03</b>	2,65 (0)	2.963,00 (2776)
	PRS_first	61,79	2,94 (0)	2.962,14 (2770)
	TARS_best	35,63	<b>0 (0)</b>	2.601,98 (2460)
	TARS_first	104,4	<b>0 (0)</b>	<b>2.591,32 (2448)</b>
nfl22	PTS_best	60,85	<b>0 (0)</b>	587.102,00 (561.198)
	PTS_first	180,51	<b>0 (0)</b>	586.309,00 (562.714)
	PRS_best	<b>37,09</b>	2,17 (0)	656.261,00 (627.005)
	PRS_first	82,63	2,60 (0)	656.119,00 (618.885)
	TARS_best	56,02	<b>0 (0)</b>	573.527,00 ( <b>546.892</b> )
	TARS_first	161,26	<b>0 (0)</b>	<b>573.337,00 (548.705)</b>

A Tabela 3 representa o teste de *Força* que demonstra os valores médios da redução de *Violações* e *Distância* a cada passo da busca. A primeira coluna apresenta os nomes das instâncias, ao passo que a segunda coluna apresenta as vizinhanças exploradas juntamente com sua estratégia de avaliação. Nas duas últimas colunas são apresentados os valores médios da redução de *Violações* e de *Distância*, respectivamente. Podemos observar que a vizinhança PRS reduz com uma taxa maior as violações, o que era esperado, uma vez que o teste de *Convergência* demonstrou que sua convergência também é maior. Porém, sua convergência se dá de forma não eficiente, pois não há a garantia de uma solução viável ao final da busca. Constatamos também que a vizinhança TARS possui uma força superior às outras.

A Tabela 4 apresenta a porcentagem de vizinhos aprimorantes a cada passo da busca, ou seja, a porcentagem

Tabela 3. TESTE DE FORÇA

Instância	Vizinhança	$\Delta$ Violações/passos	$\Delta$ Distância/passos
circ10	PTS_best	1,70	2,86
	PTS_first	0,39	1,05
	PRS_best	<b>2,56</b>	0,46
	PRS_first	0,96	0,58
	TARS_best	2,00	<b>4,46</b>
	TARS_first	0,41	1,31
circ12	PTS_best	1,38	4,50
	PTS_first	0,19	1,57
	PRS_best	<b>2,12</b>	1,20
	PRS_first	0,77	0,92
	TARS_best	1,60	<b>6,26</b>
	TARS_first	0,22	2,12
circ14	PTS_best	1,76	5,18
	PTS_first	0,20	1,67
	PRS_best	<b>2,48</b>	0,92
	PRS_first	0,91	0,57
	TARS_best	1,89	<b>7,05</b>
	TARS_first	0,27	2,30
circ16	PTS_best	1,63	6,51
	PTS_first	0,06	2,04
	PRS_best	<b>2,51</b>	1,36
	PRS_first	0,90	0,66
	TARS_best	1,86	<b>8,76</b>
	TARS_first	0,18	2,89
circ18	PTS_best	1,65	7,41
	PTS_first	0,07	2,35
	PRS_best	<b>2,56</b>	1,35
	PRS_first	0,92	0,68
	TARS_best	1,80	<b>9,88</b>
	TARS_first	0,20	3,23
circ20	PTS_best	1,68	8,38
	PTS_first	0,10	2,75
	PRS_best	<b>2,66</b>	2,11
	PRS_first	0,93	0,82
	TARS_best	1,85	<b>11,43</b>
	TARS_first	0,19	3,78

Tabela 4. TESTE DE APRIMORAMENTO

Instância	Vizinhos Aprimorantes(%) / passos		
	PTS	PRS	TARS
circ10	8,25	<b>8,37</b>	5,51
circ12	<b>7,72</b>	6,70	4,12
circ14	<b>7,70</b>	5,71	2,64
circ16	<b>6,69</b>	4,62	2,44
circ18	<b>6,58</b>	4,53	1,67

de soluções melhores que a solução corrente a cada passo. A primeira coluna apresenta os nomes das instâncias. A segunda, terceira e quarta colunas mostram o percentual de vizinhos aprimorantes para as vizinhanças PTS, PRS e TARS, respectivamente. A vizinhança TARS apresenta a menor quantidade de vizinhos aprimorantes em relação as outras duas. Isso se deve ao fato da estrutura TARS ser uma combinação das outras duas, consequentemente suas trocas envolvem uma quantidade maior de times, o que favorece a geração de violações.

A Tabela 5 sumariza o resultado dos testes. Para cada

Tabela 5. CLASSIFICAÇÃO DA PERFORMANCE DAS VIZINHANÇAS

Critério	PTS	PRS	TARS
Eficiência	2º	3º	<b>1º</b>
Convergência	3º	<b>1º</b>	2º
Força	3º	2º	<b>1º</b>
Aprimoramento	<b>1º</b>	2º	3º

critério os resultados das vizinhanças foram classificados em primeiro (1º), segundo (2º) e terceiro lugar (3º). A primeira coluna representa o critérios definidos e as demais colunas as vizinhanças avaliadas. Podemos perceber que a vizinhança TARS foi classificada duas vezes em primeiro lugar e uma vez em segundo lugar. A vizinhança PRS ficou predominantemente em segundo lugar e a vizinhança PTS, predominantemente, em terceiro. A partir dos resultados obtidos podemos formular as seguintes conclusões:

- 1) A vizinhança TARS, possui o pior *Aprimoramento* e a segunda melhor *Convergência*, porém é a melhor em termos de *Eficiência* e *Força*.
- 2) A vizinhança PRS possui a melhor *Convergência* e a melhor *Força* em relação as *Violações*, no entanto possui a pior medida de *Eficiência*. Consequentemente, as soluções encontradas possuem maiores valores de *Distância* e podem ser inviáveis.
- 3) A vizinhança PTS de destaca no critério *Aprimoramento* e a sua *Eficiência* se assemelha a *Eficiência* da TARS, mesmo sendo inferior. No entanto possui a pior *Convergência* e a pior *Força*. Isto implica que a busca consegue resultados satisfatórios, porém com baixa redução dos valores de *Violações* e *Distância* a cada iteração, fazendo com que a busca se prolongue mais.

Concluímos que a vizinhança TARS, apesar dos pontos negativos, é superior às outras vizinhanças. A baixa quantidade de vizinhos aprimorantes, menos do que 10% para todas as vizinhanças, motivou a posposta da da heurística X-HAS.

## 7.2. Análise de Melhoria da Heurística X-HAS

A Tabela 6 ilustra a *Eficiência* e a *Convergência* da heurística X-HAS. Comparando com a Tabela 2, podemos constatar que X-HAS atingiu o objetivo proposto de ampliar o espaço de busca das vizinhanças, uma vez que todas as soluções encontradas são viáveis e melhores. No entanto, como era esperado, a quantidade média de passos necessários para encontrar a solução ótima local é, em média, 69% maior. Esse incremento se deve a quantidade de passos adicionais realizados pela aplicação de buscas locais em sequência.

## 7.3. Análise das Melhorias na Vizinhança Troca de Anfitriões

Apresentamos a seguir os resultados dos experimentos que foram conduzidos no intuito de comparar a performance

Tabela 6. TESTE DE EFICIÊNCIA E CONVERGÊNCIA DO X-HAS

Instância	Vizinhança	Passos	Violações	Distância
circ10	PTS	<b>49,06</b>	0 (0)	311,10 (288)
	PRS	52,71	0 (0)	321,04 (296)
	TARS	64,86	0 (0)	<b>303,62 (284)</b>
circ12	PTS	<b>80,10</b>	0 (0)	526,24 (496)
	PRS	85,94	0 (0)	538,96 (504)
	TARS	141,25	0 (0)	<b>511,06 (486)</b>
circ14	PTS	<b>117,84</b>	0 (0)	830,42 (778)
	PRS	130,74	0 (0)	854,98 (818)
	TARS	179,96	0 (0)	<b>808,30 (774)</b>
circ16	PTS	<b>167,84</b>	0 (0)	1.227,96 (1.172)
	PRS	180,39	0 (0)	1.262,42 (1.206)
	TARS	250,04	0 (0)	<b>1.184,74 (1.094)</b>
circ18	PTS	<b>226,54</b>	0 (0)	1.731,30 (1.628)
	PRS	247,94	0 (0)	1.784,76 (1.700)
	TARS	336,72	0 (0)	<b>1.689,90 (1.632)</b>
circ20	PTS	<b>291,35</b>	0 (0)	2.361,90 (2.284)
	PRS	330,54	0 (0)	2.427,12 (2.314)
	TARS	439,94	0 (0)	<b>2.297,52 (2.208)</b>

entre as heurísticas HAS\* e o HAS++. Para efeito de comparação só foram consideradas as soluções viáveis, ou seja, aquelas que não violam a restrição R1.

Na Tabela 7 a primeira coluna contém o nome da instância, a segunda coluna lista as vizinhanças consideradas. As demais colunas são divididas em três grupos de três colunas cada, com os valores das métricas estabelecidas para efeito de comparação entre as heurísticas HAS\* e HAS++. A primeira coluna do grupo representa o número médio de vizinhos avaliados a cada passo da busca, a segunda coluna mostra a distância total média encontrada e a terceira o tempo médio de execução da busca. Os valores medidos para HAS\* e HAS++ estão representados no primeiro e segundo grupos, respectivamente. O terceiro grupo compara os valores dos dois primeiros grupos mostrando a porcentagem da diferença relativa entre eles.

A partir da análise comparativa dos dados obtidos constatamos que a heurística HAS++ comparada com a heurística HAS\*, consegue reduzir, em média, 87,6% da quantidade de vizinhos que precisam ser avaliados a cada iteração da busca. Consequentemente, isso implica em uma busca, em média, 93,9% mais rápida. Porém ao restringir os vizinhos avaliados para somente aqueles que, majoritariamente, reduzem a quantidade de violações implica em soluções finais com valores de distância total viajada, em média, 2,7% piores do que aqueles obtidos avaliando todos os vizinhos. No entanto essa piora não é tão significativa se levarmos em conta a redução de tempo. Portanto o HAS++ cumpre seu objetivo de encontrar soluções viáveis boas a partir de soluções inviáveis em um intervalo de tempo menor.

## 8. Conclusão

Neste trabalho apresentamos duas heurísticas para o MTTP. Avaliamos a efetividade de TARS para encontrar soluções boas em uma busca local no espaço de soluções.

Comparado com as outras vizinhanças, PTS e PRS, constatamos que o TARS é capaz de gerar soluções melhores, porém num espaço de tempo maior.

A fim de melhor aproveitar a capacidade de exploração das vizinhanças no espaço de soluções, foi proposta uma nova heurística, X-HAS, que combina um movimento de uma das vizinhanças seguido da vizinhança HAS\*. Desta forma é possível considerar caminhos através de soluções inviáveis, já que uma única execução do HAS\* é capaz de obter uma solução viável a partir de uma solução inviável. Entretanto, a aplicação do X-HAS provocou um considerável aumento no tempo de execução do algoritmo.

Propomos então uma nova heurística, HAS++, para minimizar o tempo computacional gasto pelo HAS\*. O HAS++ utiliza o conjunto *repair* composto pelos confrontos envolvidos num único movimento prévio das vizinhanças PTS, PRS ou TARS. A partir de cada elemento do conjunto *repair* é formado o conjunto *filter* composto de todos os confrontos que fazem parte de uma sequência de confrontos que desrespeitem a restrição R1. Ao final é realizada uma busca local melhor aprimorante com os elementos de *filter*.

Experimentos constataram que a heurística HAS++ implica numa redução, em média, de 93,9% do tempo computacional gasto comparado ao HAS\*.



Tabela 7. COMPARAÇÃO ENTRE AS HEURÍSTICAS HAS\* E HAS++

Instâncias	Vizinhança	HAS*			HAS++			Diferença(%)		
		Vizinhos	Distância	Tempo(ms)	Vizinhos	Distância	Tempo(ms)	Vizinhos	Distância	Tempo
circ10	PTS	45	365,04	0,92	5,66	377,00	0,05	87,4	3,3	94,5
	PRS	45	365,10	1,35	11,50	379,24	0,20	74,4	3,9	85,1
	TARS	45	363,32	1,09	13,52	377,00	0,21	70,0	3,8	80,9
circ12	PTS	66	626,46	1,57	6,90	643,22	0,07	89,6	2,7	95,3
	PRS	66	624,38	1,86	11,70	649,52	0,16	82,3	4,0	91,7
	TARS	66	622,10	1,81	12,92	644,48	0,19	80,4	3,6	89,3
circ14	PTS	91	992,69	2,21	7,00	1.014,40	0,06	92,3	2,2	97,2
	PRS	91	988,97	3,62	18,64	1.019,80	0,40	79,5	3,1	88,9
	TARS	91	984,77	3,21	18,20	1.015,71	0,37	80,0	3,1	88,6
circ16	PTS	120	1.470,82	4,07	7,90	1.502,74	0,08	93,4	2,2	97,9
	PRS	120	1.466,40	4,86	17,02	1.506,98	0,32	85,8	2,8	93,5
	TARS	120	1.469,58	4,91	20,10	1.506,20	0,42	83,3	2,5	91,4
circ18	PTS	153	2.099,41	4,58	7,96	2.142,62	0,08	94,8	2,1	98,3
	PRS	153	2.086,53	8,10	20,98	2.128,06	0,54	86,3	2,0	93,3
	TARS	153	2.088,78	7,25	22,04	2.134,23	0,55	85,6	2,2	92,4
circ20	PTS	190	2.867,14	6,82	8,68	2.919,60	0,10	95,4	1,8	98,6
	PRS	190	2.863,76	10,18	22,96	2.922,32	0,57	87,9	2,0	94,4
	TARS	190	2.857,41	9,15	24,98	2.912,68	0,58	86,9	1,9	93,7
nfl22	PTS	231	641.107,33	11,70	8,88	648.824,60	0,12	96,2	1,2	99,0
	PRS	231	640.305,29	14,70	27,22	650.993,75	0,76	88,2	1,7	94,9
	TARS	231	639.640,92	14,21	23,94	651.318,93	0,71	89,6	1,8	95,0
cb24	PTS	276	808.374,94	14,55	9,94	817.771,36	0,13	96,4	1,2	99,1
	PRS	276	808.380,36	18,48	26,24	820.304,92	0,71	90,5	1,5	96,1
	TARS	276	805.537,94	19,90	30,04	818.250,28	1,05	89,1	1,6	94,7
nfl30	PTS	435	1.204.489,78	29,95	11,22	1.213.386,71	0,19	97,4	0,7	99,4
	PRS	435	1.200.779,76	40,22	37,68	1.215.302,43	1,55	91,3	1,2	96,2
	TARS	435	1.200.475,74	39,90	34,38	1.214.256,30	1,44	92,1	1,1	96,4

## Referências

- [1] K. Easton, G. Nemhauser, and M. Trick, "The traveling tournament problem description and benchmarks," *Journal of Heuristics*, 2001.
- [2] C. C. Ribeiro and S. Urrutia, "Heuristics for the mirrored traveling tournament problem," *European Journal of Operational Research*, 2007.
- [3] T. Januario and S. Urrutia, "A new neighborhood structure for round robin scheduling problems," *Computers e Operations Research*, 2016.
- [4] R. A. Melo, S. Urrutia, and C. C. Ribeiro, "The traveling tournament problem with predefined venues," *Journal of Scheduling*, vol. 12, no. 6, pp. 607 – 622, Dec. 2009.
- [5] F. N. Costa, S. Urrutia, and C. C. Ribeiro, "An ILS heuristic for the traveling tournament problem with predefined venues," *Annals of Operations Research*, vol. 194, no. 1, pp. 137–150, 2012.
- [6] A. C. Guedes and C. C. Ribeiro, "A heuristic for minimizing weighted carry-over effects in round robin tournaments," *Journal of Scheduling*, vol. 14, no. 6, pp. 655–667, 2011.
- [7] M. Kidd, "A tabu-search for minimising the carry-over effects value of a round-robin tournament," *ORiON*, vol. 26, no. 2, 2010.
- [8] L. Di Gaspero and A. Schaerf, "A composite-neighborhood tabu search approach to the traveling tournament problem," *Journal of Heuristics*, 2007.
- [9] S. Belcastro and R. Haas, "Edge-Kempe-equivalence graphs of class-1 regular graphs," vol. 69, pp. 197–214, 01 2017.
- [10] V. G. Vizing, "On an estimate of the chromatic class of a p-graph," *Discrete Analysis*, vol. 3, pp. 25 – 30, 1964, in Russian.
- [11] J. Misra and D. Gries, "A constructive proof of Vizing's theorem," *Information Processing Letters*, vol. 41, no. 3, pp. 131 – 133, 1992.
- [12] P. Hansen and N. Mladenović, "First vs. best improvement: An empirical study," *Discrete Applied Mathematics*, pp. 802 – 817, 2006.
- [13] M. Trick. (2005) Challenge traveling tournament instances. [Online]. Available: <http://mat.tepper.cmu.edu/TOURN>
- [14] Z. Lü, J.-K. Hao, and F. Glover, "Neighborhood analysis: a case study on curriculum-based course timetabling," *Journal of Heuristics*, p. 97–118, 2010.