Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

Otimização do Planejamento Logístico de Campeonatos Esportivos

Bolsista: Paulo Henrique dos Santos
Orientador: Marco Antonio Moreira de Carvalho – DECOM/UFOP

Nota: Relatório Final referente ao período de 01/08/2017 a 31/07/2018, apresentado à Universidade Federal de Ouro Preto, como parte das exigências do Programa Institucional de Bolsas de Iniciação Científica (CNPq)

Local: Ouro Preto - Minas Gerais - Brasil
Data: 11 de julho de 2018

Título do Resumo

Relacionado aos problemas de planejamento de torneios esportivos, o *Mirrored Traveling Tournament Problem* (mTTP) é um problema de otimização $\mathcal{NP}$-Difícil que visa a produção de calendários de competições esportivas com o objetivo de minimizar a distância total viajada por todas as equipes do campeonato. Neste artigo é apresentada uma abordagem que aplica a metaheurística o Algoritmo Genético de Chaves Aleatórias Viciadas em conjunto com o método do polígono para a solução do mTTP. Os experimentos computacionais envolveram os dois principais conjuntos de instâncias reportados na literatura e os resultados gerados demonstram a competitividade do método proposto.

Assinatura do orientador(a): _____
Marco Antonio Moreira de Carvalho

Assinatura do bolsista: _____
Paulo Henrique dos Santos

Sumário

1 Introdução **1**
1.1 Organização do Texto 2

2 Objetivos **3**

3 Revisão da Literatura **4**

4 Fundamentação Teórica **7**
4.1 O Mirrored Traveling Tournament Problem 7
4.2 Algoritmo Genético de Chaves Aleatórias Viciadas 9
4.2.1 População 10
4.2.2 Codificação 10
4.2.3 Decodificação 10
4.2.4 Elitismo 10
4.2.5 Mutação 10
4.2.6 *Crossover* 11

5 Desenvolvimento **12**
5.1 BRKGA Aplicado ao mTTP 12
5.2 Método do Polígono 12
5.3 Buscas Locais 14
5.3.1 *Home-Away Swap* - HAS 14
5.3.2 *Round Swap* - RS 15
5.3.3 *Team Swap* - TS 15
5.3.4 *Partial Round Swap* - PRS 16
5.3.5 *Partial Team Swap* - PTS 17
5.4 Correção dos Cromossomos 18
5.5 Visão Geral 18

6 Resultados e Discussão **20**
6.1 Experimentos Computacionais 20
6.1.1 Ajuste de Parâmetros 20
6.1.2 Comparação com Resultados da Literatura 21

7 Conclusões **23**

Capítulo 1

Introdução

Desde a Grécia antiga até os dias atuais os esportes têm sido de grande importância para a humanidade. Seja promovendo saúde, lazer ou entretenimento, o esporte está presente no cotidiano das pessoas, surgindo nos espaços públicos, na TV, no rádio, nas redes sociais e em tantos outros lugares. As interações com os esportes fazem com que milhares de pessoas se tornem fãs e consumam o esporte como forma de entretenimento. Os fãs acompanham seus times ou ídolos indo aos estádios, assistindo TV, ou os acompanhando pela internet e redes sociais.

A atração do público por esportes leva a movimentação de grandes quantias de dinheiro, principalmente por esportes que são mais populares como o futebol, basquete, futebol americano entre outros. Competições como a Copa do Mundo de Futebol e Olímpiadas são competições que atraem a atenção do mundo todo e consequentemente aquecem a economia local de onde são sediadas. Devido à importância dessas competições, diferentes países estão sempre disputando o direito de sediá-las.

Vários esportes possuem campeonatos longos, divididos em temporadas de aproximadamende um ano e que são organizados pelas federações ou ligas. As temporadas ainda podem ser subdivididas em turnos e rodadas. Estes campeonatos são tipificados de acordo com a estrutura da tabela de jogos, ou calendário. Por exemplo, caso o campeonato possua uma estrutura em que todos as equipes jogam contra todos os outros um número fixo de vezes, caracteriza-se o tipo *Round Robin*. Caso este número seja exatamente um, o campeonato é do tipo *Single Round Robin* (SRR). Caso seja exatamente dois, o campeonato é do tipo *Double Round Robin* (DRR). Por exemplo, cita-se o Campeonato Brasileiro de Futebol, em que as equipes jogam duas vezes entre si, em dois turnos.

A organização de campeonatos, no entanto, vai muito além destas estruturas básicas. Existem diferentes variáveis que afetam diretamente os interesses das equipes, organizadores, patrocinadores e torcedores. Por exemplo, o direito de sediar um jogo é extremamente importante. O padrão de alternância de jogos em que que cada equipe joga dentro ou fora de casa (*Home-Away Pattern*, HAP) por exemplo, influencia diretamente a logística e arrecadação de bilheterias, além do benefício de jogar junto à torcida local.

A transmissão dos jogos é outro fator importante que pode ser levado em conta. Determinados jogos atraem uma maior atenção do público, seja por causa da rivalidade entre as equipes ou pela disputa de uma colocação no campeonato. Escalar esses jogos para datas e horários em que o público poderá assistir é de grande interesse para as emissoras de televisão que transmitem os jogos. Segundo uma análise feita no site UOL Esporte [1], o principal grupo de televisão do Brasil que detém os direitos de transmissão do Campeonato Brasileiro de Futebol arrecadou no

[1] Disponível em https://rodrigomattos.blogosfera.uol.com.br/2016/03/07/veja-quanto-dinheiro-a-globo-ganha-com-o-brasileiro/ Acessado em 10 de dezembro de 2017.

ano de 2014 cerca de 16 bilhões de reais somente com a transmissão deste campeonato.

Desta maneira, os organizadores destes se preocupam em produzir um calendário que atenda também diversos objetivos práticos e logísticos, eventualmente conflitantes. Um objetivo comum é a produção de um calendário justo em que todas as equipes possuam o mesmo número de jogos como mandante (i.e., em casa) ou o mesmo número de *breaks* (i.e., séries de jogos que são jogados sucessivamente em casa ou fora de casa). Outros objetivos incluem aumentar o número de jogos transmitidos pela TV, minimizar o número de *breaks*, criar calendários em que as preferências específicas das equipes são levadas em consideração, minimizar a distância total viajada por todos as equipes durante o campeonato, entre vários outros.

O último objetivo citado é um problema de otimização conhecido como *Traveling Tournament Problem* (TTP) definido por Easton et al. [2001]. No TTP, os campeonatos são normalmente do tipo DRR e duas equipes não podem se enfrentar duas vezes consecutivas. O TTP é um problema de otimização $\mathcal{NP}$-Difícil no sentido forte, conforme demonstrado em Thielen and Westphal [2011].

Em campeonatos onde as equipes têm suas sedes uma longe das outras faz-se necessário a realização de várias viagens ao longo do campeonato. Isso gera custos com o transporte da delegação, além de impactar fisicamente os atletas, influenciando o rendimento dos mesmos durante as partidas. Assim, o TTP se mostra extremamente importante para a minimização da distância viajada pelas equipes durante os campeonatos e para a equilíbrio das condições de disputa.

Uma variação do TTP é o *Mirrored Traveling Tournament Problem* (mTTP) definido por Ribeiro and Urrutia [2007]. No mTTP o campeonato é um DRR dividido em dois turnos SRR, em que a o os jogos do segundo turno obedecem a mesma sequência do primeiro turno, porém, com o direito de sediar os jogos invertido (i.e., os HAPs são invertidos). Assim como o TTP, o mTTP também é um problema de otimização $\mathcal{NP}$-Difícil e traz grandes desafios para otimização.

Atualmente, diversas competições utilizam a otimização para a organização dos campeonatos. O exemplo mais recente é o que foi utilizado para as Eliminatórias da Copa do Mundo FIFA de 2018 da América do Sul [Durán et al., 2017]. Neste exemplo conseguiu-se produzir um calendário em que todos as equipes participantes tivessem o mesmo número de jogos fora e dentro de casa. Outras aplicações incluem o Campeonato Brasileiro de Futebol [Ribeiro and Urrutia, 2006, 2012], o Campeonato Italiano de Futebol [Della Croce and Oliveri, 2006], e o Campeonato Belga de Futebol [Goossens and Spieksma, 2009]. Uma outra aplicação recente e interessante trata do escalonamento de árbitros de futebol simultaneamente com o escalonamento dos jogos [Atan and Hüseyinoğlu, 2017].

1.1 Organização do Texto

O restante deste trabalho é composto por 6 capítulos. Os objetivos deste trabalho são apresentados no Capítulo 2. O Capítulo 3 apresenta a revisão da literatura de 1981 até 2017. A base conceitual sobre mTTP e dos métodos utilizados é apresentada no Capítulo 4. No Capítulo 5 é apresentada uma implementação inicial do método proposto. Os experimentos realizados são apresentados no Capítulo 6. As conclusões sobre esta primeira parte do trabalho são apresentadas no Capítulo 7.

Capítulo 2

Objetivos

De maneira geral, esse trabalho consiste em desenvolver um método de inteligência computacional para a resolução do mTTP, utilizando uma estratégia que combina Algoritmos Genéticos e o Método do Polígono. A partir do objetivo principal, temos os seguintes objetivos específicos:

- Realizar pesquisa para geração de embasamento teórico e revisão bibliográfica sobre o mTTP;
- Elaborar uma heurística consistente que possa ser utilizada no problema e que permita a obtenção rápida de soluções próximas da solução ótima sem que se perca a vantagem da busca sistemática;
- Avaliar o método implementado considerando dados reais e também com problemas teste publicamente disponíveis, realizando uma análise crítica considerando outros métodos da literatura;
- Buscar a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as competições nacionais.

Outros produtos deste projeto de pesquisa serão trabalhos publicados em eventos nacionais,os quais contribuem para a promoção da Universidade Federal de Ouro Preto e também do tema tratado.

Capítulo 3

Revisão da Literatura

Motivado pelo problema de se organizar o calendário da principal liga americana de beisebol (*Major League Baseball*), o *Traveling Tournament Problem* (TTP) foi concebido por Easton et al. [2001] com o objetivo de se minimizar a distância total viajada pelas equipes durante o campeonato. O TTP tem se mostrado um grande desafio para a pesquisa operacional por se tratar de um problema de otimização $\mathcal{NP}$-Difícil no sentido forte, conforme demonstrado em Thielen and Westphal [2011].

Os primeiros métodos para solucionar o TTP foram baseados apenas em Programação por Restrições (PR) e Programação Inteira (PI). Um desses métodos foi o de Easton et al. [2002] que apresenta uma metodologia de solução híbrida de PI/PR com uma implementação paralela do algorítimo de *branch-and-price* que usa PI para resolver o problema principal e PR para resolver o problema de *pricing*. Com essa metodologia também se conseguiu pela primeira vez a otimalidade para instância NL8.

Uma das primeiras meta-heurísticas de bom desempenho para a solução do TTP foi apresentada por Anagnostopoulos et al. [2003], denominada *Traveling Tournament Simulated Annealing* (TTSA). O TTSA foi a primeira meta-heurística a explorar vizinhanças cujos os movimentos incluíam a troca de equipes, troca de rodadas e a troca de jogos através da tabela. O TTSA ajusta dinamicamente a função objetivo para balancear o tempo gasto em regiões viáveis e inviáveis. O ajuste se assemelha a uma estratégia de oscilação em busca tabu que resolve problemas gerais de atribuição. O TTSA também utiliza um mecanismo de reaquecimento para escapar de mínimos locais em baixas temperaturas. Este mecanismo aumenta a temperatura novamente e divide a busca em várias fases.

Baseado no TTSA, Van Hentenryck and Vergados [2006] apresentaram extensões e melhorias para solucionar o caso em que o segundo turno possui a mesma sequência de jogos porém, com a sede dos jogos invertida. Foram considerados novamente os mesmos movimentos utilizados pelo TTSA e foram adicionadas pequenas modificações que levavam em consideração as restrições de espelhamento da tabela. Todos os melhores resultados até então foram igualados ou superados por este novo método, exceto por uma instância.

Outra aplicação do *simulated annealing* para solucionar o TTP foi desenvolvida em Lim et al. [2006]. O *simulated annealing* foi utilizado para gerar os *Home-Away Patterns* (HAPs) e o algorítimo de *hill-climbing* foi utilizado para atribuir as equipes aos HAPs gerados. Primeiramente as soluções iniciais eram geradas por *beam search*. Com o intuito de se reduzir o tempo computacional e gerar soluções viáveis para um grande número de equipes os padrões gerados nessa etapa inicial eram espelhados. Assim, o número de HAPs possíveis a serem gerados foi reduzido e deste modo a solução também era espelhada, embora esse não fosse o foco inicial.

Posteriormente, Ribeiro and Urrutia [2007] formalizaram a variante espelhada do TTP, de-

nominada *Mirrored Traveling Tournament Problem*, embora esta já fosse conhecida da comunidade acadêmica. O mTTP possui o mesmo objetivo do TTP, porém, conforme mencionado, na tabela gerada o segundo turno tem a mesma sequência de jogos do primeiro, invertendo-se a sede dos jogos. Considera-se este trabalho como o primeiro a se concentrar especificamente em instâncias espelhadas do TTP. Foi também apresentada uma heurística híbrida que combina os princípios do *Greedy Randomized Adaptive Search Procedures* (GRASP) e do *Iterated Local Search* (ILS), denominada GRILS-mTTP. Essa heurística obteve bons resultados tanto para instâncias espelhadas e quanto para não espelhadas. Adicionalmente, esta meta-heurística serviu como uma base para algumas das outras heurísticas que estariam por vir.

Di Gaspero and Schaerf [2007] utilizaram as vizinhanças propostas com TTSA e uma nova adicional para produzir diferentes combinações de vizinhanças. Nestas combinações, alguns movimentos que se sobrepunham, gerando redundâncias na busca por uma solução, foram descartados. Assim, o espaço de busca foi reduzido. As combinações de vizinhanças criadas foram analisadas e comparadas. Posteriormente, estas foram embutidas em uma busca tabu a fim de se solucionar o TTP.

Uma dessas vizinhanças é a *troca parcial de times*. A busca local por troca parcial de times é frequentemente utilizada para resolver problemas de *round robin sport scheduling* como o TTP e o mTTP. Foi realizada uma análise específica da conectividade desta vizinhança por Januario et al. [2016]. Concluiu-se que a conectividade é baixa, o que torna difícil se movimentar entre soluções vizinhas diferentes. Os métodos propostos, baseados nesta análise, incluem método do polígono e método *Faro Shuffle*. Adicionalmente, foram caracterizados os casos em que a busca local por troca parcial de times é particularmente útil.

O *simulated annealing* também foi utilizado nos métodos evolutivos aplicados para solucionar o mTTP. O primeiro método evolutivo foi o de Biajoli and Lorena [2006] em que foi utilizado um Algoritmo Genético em conjunto com o método do Polígono. O método do Polígono foi utilizado para gerar a tabela a partir de um cromossomo codificado como uma permutação. Após as mutações e *crossover* serem aplicadas o *simulated annealing* foi utilizado para se encontrar um ótimo local. Um outro método evolutivo foi utilizado por Gupta et al. [2013] em que o *Biogeography Based Optimization* (BBO) foi utilizado como processo evolutivo. Neste artigo o método do polígono e o *simulated annealing* também são utilizados de maneira similar. Ambos tiveram resultados parecidos e que não trouxeram grandes melhorias em relação aos melhores de suas respectivas épocas.

Uma solução baseada em Programação Linear Inteira foi apresentada por de Carvalho and Lorena [2012], que utilizaram o conceito de grafos de conflitos. Cada nó no grafo representava um jogo e se esses nós fossem adjacentes, os respectivos jogos não poderiam ocorrer na mesma rodada devido ao fato de serem conflitantes. Adicionalmente, foi introduzida uma nova versão do TPP, denominada MinMaxTTP cujo objetivo era a minimização da maior distância viajada pelas equipes.

Há uma classe de instâncias do TTP e do mTTP em que as distâncias entre quaisquer pares de sedes das equipes são constantes. Para essa classe, foi provado por Urrutia and Ribeiro [2006] que minimizar a distância total viajada é equivalente a maximizar o número de *breaks*. Essa concepção de distâncias constantes é mais interessante quando se deseja diminuir o número de viagens ao invés da distância total viajada. Distâncias constantes também foram levadas em consideração por Rasmussen and Trick [2007] que usou o *Pattern Generating Benders Approach* (PGBA) para resolver o problema e trouxe soluções nunca obtidas anteriormente para distâncias constantes. Ademais, foram definidos novos limitantes inferiores e superiores para instâncias com distâncias constantes por se tratar de um método exato.

Limitantes inferiores para o mTTP também foram encontrados por Cheung [2009] em que

foi aplicado uma decomposição de *Benders*. Este método não teve bom desempenho para se solucionar o mTTP mas gerou novos limitantes inferiores para algumas instâncias. Alguns métodos exatos baseados em *branching* foram utilizados para resolver o TTP e o mTTP e acabaram por definir limitantes inferiores para esses problemas. O primeiro, Irnich [2010] foi aplicado um algoritmo *branch-and-price* em que se alcançou resultados ótimos para algumas instâncias do mTTP além de alguns novos limitantes inferiores. O segundo, proposto por Uthus et al. [2009] e baseado em uma versão preliminar do trabalho anterior, utilizou um algoritmo DFS* para podar prováveis caminhos não ótimos da árvore de busca e trouxe bons resultados para instâncias do TTP além de alguns limitantes inferiores.

Uma aplicação real para o mTTP foi realizada por Atan and Hüseyinoğlu [2017] em que abordou a Liga de Futebol Turco (*Super Lig*) considerando o escalonamento de árbitros e jogos simultaneamente. A Liga Turca é um típico campeonato DDR espelhado, e os objetivos foram a minimização da distância total viajada pelas equipes, equilibrar o número de jogos em que os árbitros atuavam, atribuir árbitros mais experientes para os jogos mais importantes e difíceis, e evitar que um determinado árbitro atuasse em vários jogos de uma determinada equipe. Duas abordagens de solução são apresentadas. A primeira é uma abordagem simultânea em que as os árbitros são escalados ao mesmo tempo em que a tabela de jogos é criada. A segunda é uma abordagem sequencial em que primeiramente a tabela de jogos é criada e os árbitros são escalados posteriormente. As soluções simultâneas apresentaram melhores resultados em relação às sequenciais devido ao fato de levarem em consideração a dificuldade dos jogos e os níveis dos árbitros. Além disso, a quantidade de jogos sequenciais e o nível de dificuldade dos jogos que um determinado árbitro apita se apresentou mais equilibrado nas soluções simultâneas.

Capítulo 4

Fundamentação Teórica

Neste capítulo é apresentada a definição formal do *Mirrored Traveling Tournament Problem* utilizando exemplos da *National League* com quatro equipes. Também é apresentado o conceito do Algoritmo Genético de Chaves Aleatórias Viciadas, utilizado nesse trabalho para solucionar o mTTP.

4.1 O Mirrored Traveling Tournament Problem

O *Mirrored Traveling Tournament Problem* é aplicado em campeonatos DRR espelhados (ou *Mirrored Double Round Robin*, MDRR) com n equipes, em que n é um número par. Cada equipe deve enfrentar duas vezes cada uma das outras equipes: uma vez no primeiro turno (ou primeiro SRR) e a segunda vez no segundo turno (ou segundo SRR). Duas equipes não devem se enfrentar mais do que uma vez em cada turno. A sequência de adversários de cada equipe deve ser a mesma tanto no primeiro turno quanto no segundo turno, porém, os HAPs (sede dos confrontos) devem ser invertidos no segundo turno.

Dados n equipes, sendo n um número par, o total de rodadas de um campeonato MDRR é $r = 2(n-1)$, sendo $n-1$ rodadas no primeiro turno e outras $n-1$ rodadas no segundo turno. Em cada rodada devem existir $\frac{n}{2}$ confrontos, sendo que cada equipe deve possuir apenas um confronto por rodada. confrontos entre duas equipes E_i e E_j são definidos por pares ordenados (i, j). As distâncias d_{ij} entre as cidades das equipes i e j são conhecidas e cada equipe começa o campeonato em sua respectiva cidade. Quando uma equipe possui uma sequência de confrontos fora de casa essa equipe vai de uma cidade de um adversário para outra cidade de outro adversário sem retornar para casa, i.e., a equipe sai em turnê. A tabela gerada pelo mTTP deve obedecer às seguintes restrições:

1. Nenhuma equipe pode ter mais do que três confrontos em sequência em casa ou três confrontos em sequência fora de casa;

2. Um confronto entre uma equipe E_i e uma equipe E_j que aconteceu na casa de E_i não pode ser seguido de outro confronto entre E_i e E_j na casa de E_j. Em outras palavras, não pode existir dois confrontos consecutivos entre duas equipes; e

3. Os confrontos em uma rodada R são exatamente os mesmos confrontos em uma rodada $R + (n-1)$ para $R = 1, 2, 3, \ldots, n-1$, com o local dos confrontos invertidos.

A Tabela 4.1 apresenta a matriz de distâncias D para a instância NL4. Esta é uma instância derivada de dados reais da liga americana de beisebol profissional dos Estados Unidos e Ca-

nadá que contém quatro equipes: *Atlanta Braves* (ATL), *New York Mets* (NYM), *Philadelphia Phillies* (PHI) e *Montreal Expos* (MON).

Tabela 4.1: Exemplo de matriz de distâncias D da instância NL4.

Equipe	ATL	NYM	PHI	MON
ATL	0	745	665	929
NYM	745	0	80	337
PHI	665	80	0	380
MON	929	337	380	0

Um exemplo de HAPs para cada equipe da instância NL4 é mostrado na Tabela 4.2. Nos HAPs apresentados, C indica um confronto disputado em casa, e F indica um confronto disputado fora. No caso da equipe ATL, o HAP indica os três primeiros confrontos como mandante e os três últimos confrontos como visitante.

Tabela 4.2: Exemplo de HAPs para a instância NL4.

Equipe	Primeiro Turno			Segundo Turno		
ATL	C	C	C	F	F	F
NYM	C	F	F	F	C	C
PHI	F	C	C	C	F	F
MON	F	F	F	C	C	C

Uma solução para o mTTP consiste em uma tabela MDDR de tamanho $n \cdot 2(n-1)$ que representa quais equipes se enfrentam e o local do enfrentamento em cada rodada. A Tabela 4.3 apresenta uma possível solução para a instância NL4, com base nos HAPs apresentados anteriormente. O sinal positivo indica que a equipe da coluna será a visitante e o sinal negativo indica que o enfrentamento é fora de casa.

Tabela 4.3: Exemplo tabela de confrontos para instância NL4.

	Primeiro Turno			Segundo Turno		
Equipe	Rodada 1	Rodada 2	Rodada 3	Rodada 4	Rodada 5	Rodada 6
ATL	PHI	NYM	MON	-PHI	-NYM	-MON
NYM	MON	-ATL	-PHI	-MON	ATL	PHI
PHI	-ATL	MON	NYM	ATL	-MON	-NYM
MON	-NYM	-PHI	-ATL	NYM	PHI	ATL

O custo produzido por cada equipe em uma solução é determinada pela distância percorrida por essa equipe durante o campeonato desde a primeira vez que essa equipe deixa sua cidade até a última vez que retorna. A partir da matriz de distâncias D apresentada na Tabela 4.1 e da tabela de confrontos apresentada na Tabela 4.3, pode-se calcular por exemplo, o custo total produzido pela equipe ATL, que é dado por:

$$d_{11} + d_{11} + d_{11} + d_{13} + d_{32} + d_{24} = 1082$$

Para o cálculo do custo total de todas as equipes na função objetivo uma variável binária x_{ijk} é utilizada:

$$x_{ijk} = \begin{cases} 1, & \text{se a equipe } E_i \text{ confronta a equipe } E_j \text{ na rodada } k \\ 0, & \text{caso contrário} \end{cases} \tag{4.1}$$

A função objetivo é dada pela Equação (4.2), que visa minimizar a total distância viajada por todas as equipes do campeonato:

$$\min Z_{mTTP} = \sum_{k=0}^{r} \sum_{j=0}^{n} \sum_{i=0}^{n} x_{ijk} d_{ij} \tag{4.2}$$

4.2 Algoritmo Genético de Chaves Aleatórias Viciadas

Algoritmos Genéticos (AG) são uma classe de metaheurísticas populacionais evolutivas, comumente usadas para solucionar problemas de otimização combinatória. São métodos inspirados na teoria da evolução de *Darwin*, de forma que cada *individuo* ou *cromossomo* representa uma solução para um problema de otimização específico, normalmente dividido em porções de informação denominadas *genes*. Uma população evolui ao longo de gerações através de mecanismos de intensificação e diversificação das soluções, representados por operadores de reprodução (ou *crossover*) e mutação, respectivamente. Cada indivíduo é avaliado de acordo com sua aptidão (ou *fitness*), de maneira que os mais aptos têm maior chance de perpetuarem seu material genético ao longo das gerações. Ao final da execução de um AG, o indivíduo mais apto é a melhor solução encontrada em todas as gerações.

Existem as mais diversas variações dos AGs. Dentre elas, há os Algoritmos Genéticos de Chaves Aleatórias (*Random-Key Genetic Algorithm*, RKGA) introduzido por Bean [1994]. Neste método, cada individuo é codificado como um vetor de chaves aleatórias, em que cada chave é um numero real pertencente ao intervalo [0,1). Novamente, os indivíduos representam soluções para um determinado problema, entretanto, é necessário um decodificador para transformar o vetor de chaves aleatórias em uma solução de fato para o problema tratado, além de determinar sua aptidão. No RKGA, uma porcentagem dos indivíduos mais aptos de cada geração pertencem a um conjunto denominado *elite*. A cada geração no RKGA, uma nova população é criada, composta pelo conjunto elite da geração anterior, por indivíduos gerados pela reprodução de indíviduos da geração anterior e por indivíduos mutantes. Particularmente, o RKGA utiliza a reprodução uniforme parametrizada de Spears and De Jong [1995], na qual o *i*-ésimo gene do cromossomo *filho* gerado pode receber a *i*-ésima chave de um dos dois cromossomos *pai*, selecionados aleatoriamente, com a mesma probabilidade. O operador de mutação consiste em introduzir novos indivíduos, cujos genes são gerados de maneira aleatória.

Uma variação do RKGA foi recentemente introduzida por Gonçalves and Resende [2011], denominada de Algoritmo Genético de Chaves Aleatórias Viciadas (*Biased Random-Key Genetic Algorithm*, BRKGA). O BRKGA segue os mesmos princípios do RKGA, diferindo principalmente quanto ao operador de reprodução. No BRKGA, os dois indivíduos pai que serão utilizados para gerar um cromossomo filho são novamente selecionados de maneira aleatória, entretanto, um dos indivíduos deve pertencer ao conjunto elite e o outro deve pertencer ao restante da população. Adicionalmente, embora seja utilizado a reprodução uniforme parametrizada, os genes advindos do indivíduo pai do conjunto elite possuem maior probabilidade de serem atribuídos ao individuo filho, em detrimento do individuo pai não pertencente ao conjunto elite. Os componentes e as fases de operação do BRKGA serão apresentados e exemplificados a seguir.

4.2.1 População

A população consiste em vários indivíduos ou cromossomos que são vetores de chaves aleatórias. A população inicial é gerada de maneira totalmente aleatória a partir da construção dos vetores. Cada vetor possui um valor de *fitness* associado de modo que ao longo das gerações novas populações são criadas utilizando o *fitness* de cada indivíduo como base.

4.2.2 Codificação

A codificação é responsável pela criação dos indivíduos ou cromossomos representados pelos vetores de chaves aleatórias. Cada vetor criado possui um número *n* de chaves aleatórias e cada chave tem um valor real entre 0 e 1 definido aleatoriamente. As chaves representam os genes do cromossomo e o fato de serem gerados aleatoriamente ajudam a promover uma diversidade maior das soluções. A codificação realizada dessa maneira proporciona um uso geral dessa metaheurística já que independe do problema. A dependência do problema se dá na fase da decodificação que será tratada a seguir. Um exemplo de cromossomo que possui quarto chaves pode ser visto na Figura 4.1.

0,13	0,59	0,33	0,88

Figura 4.1: Exemplo de cromossomo.

4.2.3 Decodificação

Na fase da decodificação um vetor de chaves aleatórias é transformado em uma solução para o problema específico que está sendo tratado. Assim, é possível a partir da solução obtida pela decodificação associar um valor de *fitness* com o cromossomo calculado através da função objetivo do problema.

4.2.4 Elitismo

O elitismo consiste em classificar os cromossomos com os melhores valores de *fitness* como sendo os melhores cromossomos da população corrente. Esses cromossomos classificados como elite são copiados para a próxima população sem sofrer alterações. O elitismo é importante para o algoritmo a medida que proporciona a convergência para uma melhor solução ao longo das gerações.

4.2.5 Mutação

A mutação é responsável por gerar novos cromossomos de forma totalmente aleatória para a próxima população de maneira a substituir parte dos cromossomos não elite da população corrente. Assim, os cromossomos com um desempenho baixo são descartados e novos cromossomos com um desempenho melhor podem surgir ajudando o algoritmo a escapar de ótimos locais.

4.2.6 *Crossover*

O restante dos cromossomos não elite que não foram substituídos pelos mutantes na próxima população são gerados com a aplicação do *crossover*. O *crossover* consiste em unir genes de um cromossomo elite com de um cromossomo não elite para produzir um novo cromossomo. É definida um probabilidade p que diz se determinado gene do novo cromossomo será herdado do cromossomo elite. A probabilidade p caracteriza as chaves como viciadas, pois fará com que a maioria dos genes do novo cromossomo serão provindos do cromossomo elite. No exemplo de *crossover* apresentado na Figura 4.2 é possível verificar que apenas o último gene do cromossomo descendente foi herdado do pai não elite, o restante foi provindo do pai elite.

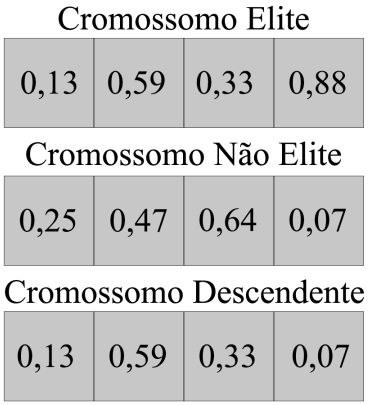

Figura 4.2: Exemplo de *crossover*.

Como dito anteriormente, ao longo das gerações as populações são construídas a partir da população anterior. A próxima população é dividida em três partes. A primeira parte é o conjunto dos cromossomos elites que são copiados da população corrente. A segunda parte são os cromossomos provindos dos *crossovers* entre os elementos elite e não elite da população corrente. A terceira parte possui o restante dos cromossomos da população e é constituída pelos mutantes gerados aleatoriamente. É possível observar a composição das populações na Figura 4.3.

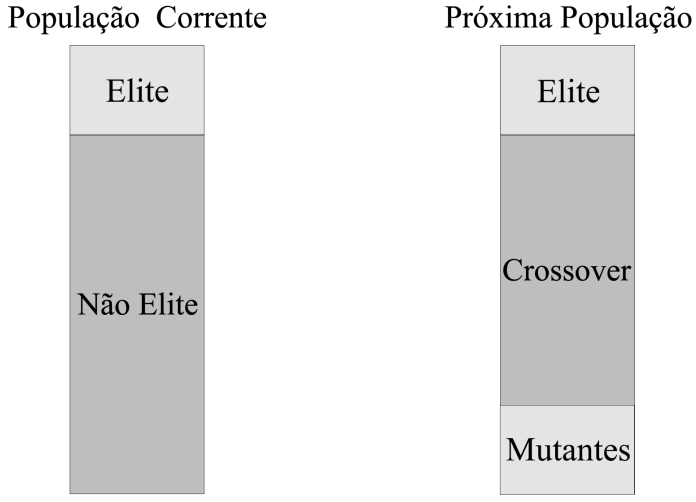

Figura 4.3: Composição das populações.

Capítulo 5

Desenvolvimento

Neste capítulo é apresentado o Método do Polígono que neste trabalho é utilizado para decodificar os cromossomos do BRKGA em possíveis soluções para o mTTP. Apresenta-se também a aplicação do BRKGA ao problema e as buscas locais que foram aplicadas em cada um dos cromossomos.

5.1 BRKGA Aplicado ao mTTP

Para aplicação ao mTTP, o BRKGA gera vetores de chaves aleatórias de tamanho equivalente ao número n de equipes que participam do torneio. O intuito é gerar permutações de tamanho n, que posteriormente são transformadas em tabelas SRR pelo método o polígono. Para obter tais permutações, a decodificação de cada indivíduo consiste em ordenar o vetor de chaves aleatórias de cada indivíduo de forma não decrescente de acordo com o valor das chaves, mantendo junto com a chave o seu índice original. Assim, obtém-se permutações dos n índices de cada indivíduo.

5.2 Método do Polígono

O método do polígono [Dinitz et al., 1995] é comumente utilizado na literatura, sendo utilizado por exemplo por Biajoli and Lorena [2006] e por Gupta et al. [2013]. Com o método do polígono é possível, a partir de uma permutação base dos índices das equipes, representar uma tabela SRR com todos os confrontos e todas as rodadas. A permutação base, corresponde à primeira rodada da tabela do campeonato. Conforme mencionado, neste trabalho a permutação base é obtida por meio da decodificação dos indivíduos do BRKGA. A Tabela 5.1 apresenta um exemplo de permutação base para uma instância que contém seis equipes.

Tabela 5.1: Permutação base para seis equipes.

5	4	6	2	3	1

Considerando os dados da Tabela 5.1 é possível criar uma tabela SRR que corresponde às cinco rodadas de um torneio. Inicialmente, a permutação base é expandida, e posteriormente, os confrontos são definidos baseados nesta expansão.

A expansão consiste em gerar n-1 permutações. A primeira consiste da própria permutação base. Para geração das demais permutações, fixa-se o primeiro elemento da permutação base na primeira posição de todas as demais. O segundo elemento de cada permutação é o terceiro

elemento da permutação anterior, o terceiro elemento é o quarto elemento da permutação anterior e assim consecutivamente até o último elemento, que é o segundo elemento da permutação anterior. Para o exemplo da permutação da Tabela 5.1 o resultado da expansão gerada pelo método do polígono é apresentado na Tabela 5.2.

Tabela 5.2: Expansão gerada pelo método do polígono.

5	4	6	2	3	1
5	6	2	3	1	4
5	2	3	1	4	6
5	3	1	4	6	2
5	1	4	6	2	3

A expansão gerada pelo método do polígono é utilizada como base para geração de uma tabela SRR. Considerando a expansão da Tabela 5.2, a tabela SRR apresentada na Tabela 5.3, na qual cada linha corresponde a uma equipe e cada coluna corresponde a uma rodada, é gerada como descrito a seguir. Para cada rodada $Ri = 1, ..., n-1$, a equipe da posição 1 da permutação i confronta a equipe na posição 2 da mesma permutação. De maneira análoga, as equipes das posições $i = 3, ..., (n/2)+1$ confrontam as equipes das posições $n-i+3$. Considerando a permutação base [5, 4, 6, 2, 3, 1], temos a definição da rodada 1 com os confrontos das equipes das posições 1 e 2 (5×4); 3 e 6 (6×1); e 4 e 2 (2×3), o que corresponde à primeira coluna da Tabela 5.3.

Tabela 5.3: Tabela SRR obtida relativa ao primeiro turno.

	R_1	R_2	R_3	R_4	R_5
E_1	6	3	4	2	5
E_2	3	4	5	1	6
E_3	2	1	6	5	4
E_4	5	2	1	6	3
E_5	4	6	2	3	1
E_6	1	5	3	4	2

Ainda é necessário associar os HAPs aos confrontos apresentados na Tabela 5.3. Para este fim, é utilizada a heurística gulosa proposta em Ribeiro and Urrutia [2007]. De acordo com este método, os HAPs da primeira rodada são gerados aleatoriamente. Para as rodadas seguintes, cada confronto é analisado e o mando de campo é determinado pela equipe envolvida com maior número de confrontos consecutivos anteriores em casa ou fora. Com base nesta informação, o padrão predominante é invertido, e.g., se uma equipe vem de confrontos consecutivos anteriores como mandante, o confronto atual será disputado como visitante. Caso as duas equipes tenham o mesmo número de confrontos consecutivos anteriores com mesmo padrão, tenta-se inverter o padrão da mesma maneira, ou, caso seja impossível, o mando de campo é gerado aleatoriamente. Determinados os HAPs, o resultado é a Tabela 5.4.

Tabela 5.4: Tabela SRR contendo os confrontos do primeiro turno.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	3	-4	2	-5
E_2	-3	4	-5	-1	6
E_3	2	-1	6	5	-4
E_4	5	-2	1	-6	3
E_5	-4	6	2	-3	1
E_6	1	-5	-3	4	-2

O segundo turno é criado espelhando-se o primeiro. O resultado final é uma tabela MDRR, como a apresentada na Tabela 5.5.

Tabela 5.5: Tabela MDRR contendo os confrontos dos dois turnos.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
E_1	-6	3	-4	2	-5	6	-3	4	-2	5
E_2	-3	4	-5	-1	6	3	-4	5	1	-6
E_3	2	-1	6	5	-4	-2	1	-6	-5	4
E_4	5	-2	1	-6	3	-5	2	-1	6	-3
E_5	-4	6	2	-3	1	4	-6	-2	3	-1
E_6	1	-5	-3	4	-2	-1	5	3	-4	2

5.3 Buscas Locais

Após a fase de decodificação, as tabelas SRR são submetidas a cinco buscas locais diferentes, propostas por Anagnostopoulos et al. [2003]. Cada tabela gerada é submetida a cada busca local com probabilidade $\rho \in [0,1)$. Todas as buscas locais são aplicadas com política de primeira melhoria, de maneira que movimentos de piora não são aceitos. Movimentos que levam a inviabilidade da solução espelhada também são descartados. A seguir, cada busca local é descrita detalhadamente.

5.3.1 *Home-Away Swap* - HAS

Nesta busca local um movimento consiste em, dado um confronto entre duas equipes E_i e E_j, inverter o mando de campo nos confrontos entre essas duas equipes. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local.

Considerando o exemplo da Tabela 5.4, um exemplo de movimento consiste em primeiramente escolher a equipe E_2 e a rodada R_1, multiplicar por -1 o respectivo elemento (-3) e o elemento correspondente a equipe do primeiro elemento (2) na mesma rodada. O resultado da aplicação do HAS é mostrado na Tabela 5.6.

Tabela 5.6: Tabela SRR após a aplicação do HAS.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	3	-4	2	-5
E_2	3	4	-5	-1	6
E_3	-2	-1	6	5	-4
E_4	5	-2	1	-6	3
E_5	-4	6	2	-3	1
E_6	1	-5	-3	4	-2

5.3.2 *Round Swap* - RS

Dadas duas rodadas R_i e R_j ($i \neq j$), esta busca local consiste em realizar a troca dos confrontos entre as duas rodadas na tabela. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local. Considerando o exemplo da Tabela 5.4, um possível movimento significa trocar as colunas R_1 e R_2, com isso, alterando a distância viajada pelas equipes. O resultado da aplicação do RS é mostrado na Tabela 5.7.

Tabela 5.7: Tabela SRR após a aplicação do RS.

	R_1	R_2	R_3	R_4	R_5
E_1	3	-6	-4	2	-5
E_2	4	-3	-5	-1	6
E_3	-1	2	6	5	-4
E_4	-2	5	1	-6	3
E_5	6	-4	2	-3	1
E_6	-5	1	-3	4	-2

5.3.3 *Team Swap* - TS

Dadas duas equipes E_i e E_j, um movimento desta busca local consiste em trocar todos os confrontos da equipe E_i com os confrontos da equipe E_j, exceto o confronto entre as duas. O mando de campo no confronto entre E_i e E_j é mantido o mesmo. Uma pequena modificação é necessária para manter a viabilidade da solução. Existe um total de $O(n^2)$ movimentos possíveis nesta busca local.

Considerando o exemplo da Tabela 5.4, um possível movimento significa trocar as linhas E_1 e E_6, exceto pela rodada R_1, correspondentes aos confrontos entre os times correspondentes.

Tabela 5.8: Tabela SRR após a troca de linhas.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	-5	-3	4	-2
E_2	-3	4	-5	-1	6
E_3	2	-1	6	5	-4
E_4	5	-2	1	-6	3
E_5	-4	6	2	-3	1
E_6	1	3	-4	2	-5

Faz-se necessário também a troca de todos os elementos 1 por 6 e os elementos 6 por 1, exceto pela rodada R_1, conforme a Tabela 5.9. Para a tabela resultante ser viável, os HAPs em

todas as modificações devem ser mantidos.

Tabela 5.9: Tabela SRR após a aplicação do TS.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	-5	-3	4	-2
E_2	-3	4	-5	-6	1
E_3	2	-6	1	5	-4
E_4	5	-2	6	-1	3
E_5	-4	1	2	-3	6
E_6	1	3	-4	2	-5

5.3.4 *Partial Round Swap* - PRS

Dadas duas rodadas R_i e R_j ($i \neq j$) e uma equipe E_k. Um movimento desta busca local consiste em trocar a equipe que E_k enfrenta na rodada R_i com a equipe que E_k enfrenta na rodada R_j. Este movimento pode tornar uma solução inviável, de maneira que é necessário criar uma cadeia de ejeção incluindo outros confrontos para manter a viabilidade da solução. Existe um total de $O(n^3)$ movimentos possíveis nesta busca local.

Considerando o exemplo da Tabela 5.4, um possível movimento significa trocar os elementos (E_1, R_1) e (E_1, R_2). Entretanto, isso implicaria na violação da restrição de apenas um jogo por rodada pela equipe E_3 na rodada R_1 e pela equipe E_6 na rodada R_2. Para tornar a tabela viável a primeira modificação equivale à troca de todos os elementos 3 por 6 e todos os elementos 6 por 3 que não pertencem a linha das equipes E_3 e E_6 conforme a Tabela 5.10.

Tabela 5.10: Tabela SRR após a primeira modificação para o PRS.

	R_1	R_2	R_3	R_4	R_5
E_1	-3	6	-4	2	-5
E_2	-6	4	-5	-1	3
E_3	2	-1	6	5	-4
E_4	5	-2	1	-3	6
E_5	-4	3	2	-6	1
E_6	1	-5	-3	4	-2

A segunda modificação necessária para tornar a tabela viável significa trocar a linha da equipe E_3 com a linha da equipe E_6, exceto pela coluna R_3, correspondentes aos confrontos entre os times correspondentes conforme a Tabela 5.11. É necessário manter os HAPs em todas as modificações para garantir a viabilidade da tabela resultante.

Tabela 5.11: Tabela SRR após a aplicação do PRS.

	R_1	R_2	R_3	R_4	R_5
E_1	-3	6	-4	2	-5
E_2	-6	4	-5	-1	3
E_3	1	-5	6	4	-2
E_4	5	-2	1	-3	6
E_5	-4	3	2	-6	1
E_6	2	-1	-3	5	-4

5.3.5 *Partial Team Swap* - PTS

Dadas duas equipes E_i e E_j ($i \neq j$) e uma rodada R_k, um movimento desta busca local consiste em trocar a equipe que E_i enfrenta na rodada R_k com a equipe que E_j enfrenta na rodada R_k. Assim como na busca local PRS, são necessárias outras modificações na tabela para que a solução se mantenha viável. Existe um total de $O(n^3)$ movimentos possíveis nesta busca local.

Considerando o exemplo da Tabela 5.4, um possível movimento significa trocar os elementos (E_1, R_5) e (E_6, R_5). Entretanto, isso implicaria na violação da restrição de apenas um jogo por rodada novamente, exigindo modificações adicionais para manter a viabilidade da solução. A primeira modificação equivale à trocar dos elementos da equipe E_2 e E_5 na rodada R_5 conforme a Tabela 5.12.

Tabela 5.12: Tabela SRR após a primeira modificação.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	3	-4	2	-2
E_2	-3	4	-5	-1	1
E_3	2	-1	6	5	-4
E_4	5	-2	1	-6	3
E_5	-4	6	2	-3	6
E_6	1	-5	-3	4	-5

A segunda modificação consiste em trocar todos os elementos 2 por 5 e 5 por 2, exceto nas linhas E_2 e E_5. A terceira modificação é similar à segunda e consiste em trocar todos os elementos 1 por 6 e 6 por 1, exceto nas linhas E_1 e E_6. A Tabela 5.13 mostra o resultado da segunda e da terceira modificação.

Tabela 5.13: Tabela SRR após a segunda e terceira modificação

	R_1	R_2	R_3	R_4	R_5
E_1	-6	3	-4	5	-2
E_2	-3	4	-5	-6	1
E_3	5	-6	1	2	-4
E_4	2	-5	6	-1	3
E_5	-4	1	2	-3	6
E_6	1	-2	-3	4	-5

A quarta modificação significa trocar as linhas E_1 e E_6, exceto pela rodada R_1, correspondentes aos confrontos entre os times correspondente. O resultado dessa modificação é mostrado na Tabela 5.14.

Tabela 5.14: Tabela SRR após a quarta modificação

	R_1	R_2	R_3	R_4	R_5
E_1	-6	-2	-3	4	-5
E_2	-3	4	-5	-6	1
E_3	5	-6	1	2	-4
E_4	2	-5	6	-1	3
E_5	-4	1	2	-3	6
E_6	1	3	-4	5	-2

A última modificação consiste em trocar as linhas E_2 e E_5, exceto pela rodada R_3, correspondentes aos confrontos entre os times correspondente. Para garantir a viabilidade da solução faz-se necessário manter os HAPs para todas as modificações realizadas. O resultado da aplicação do PTS é mostrado na Tabela 5.15

Tabela 5.15: Tabela SRR após aplicação do PTS.

	R_1	R_2	R_3	R_4	R_5
E_1	-6	-2	-3	4	-5
E_2	-4	1	-5	-3	6
E_3	5	-6	1	2	-4
E_4	2	-5	6	-1	3
E_5	-3	4	2	-6	1
E_6	1	3	-4	5	-2

5.4 Correção dos Cromossomos

Cada busca local altera de alguma forma a tabela gerada pelo método do polígono a partir da decodificação do cromossomo. Após a aplicação das buscas locais, torna-se necessário atualizar o cromossomo para que a representação da solução seja fidedigna.

Entretanto, nem todas as alterações realizadas podem ser codificadas como uma única permutação. Por este motivo, optou-se por uma representação indireta parcial, de maneira que a permutação que mais se aproxima da tabela gerada pelas buscas locais é utilizada para atualizar as chaves do cromossomo após a aplicação das buscas locais. As informações eventualmente perdidas neste processo podem ser alcançadas novamente por meio de uma nova aplicação das buscas locais.

5.5 Visão Geral

O esquema geral do BRKGA é apresentado no fluxograma apresentado na Figura 5.1. Nela é possível verificar o fluxo de execução e o relacionamento entre os diferentes componentes utilizados.

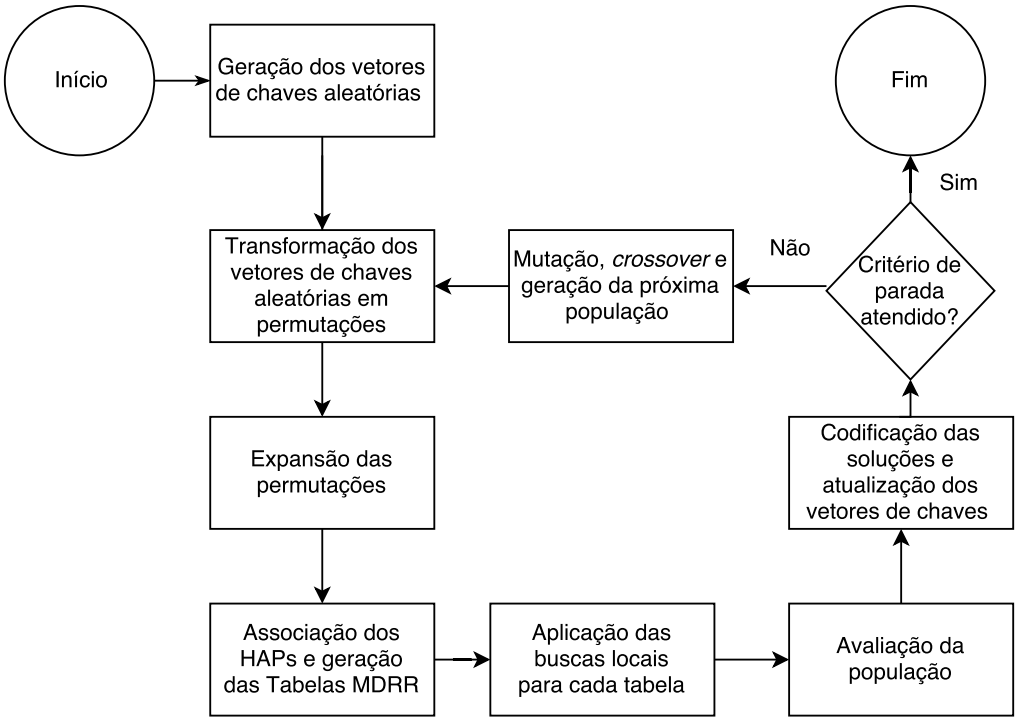

Figura 5.1: Fluxograma da abordagem proposta.

Capítulo 6

Resultados e Discussão

Neste capítulo é apresentado o ambiente computacional em que os experimentos foram realizados, como foi feita a calibragem dos parâmetros do BRKGA e os resultados obtidos para as principais instâncias reportadas na literatura. É apresentada também uma análise a respeito dos resultados obtidos e uma comparação com os melhores resultados da literatura até o tempo de escrita deste trabalho.

6.1 Experimentos Computacionais

Os experimentos computacionais foram realizados em um computador Intel Core i5 Quad Core de 3.0 GHz com 8 GB de RAM sob o sistema operacional Ubuntu 15.10. Foi utilizada a *brkgaAPI*, desenvolvida em C++ por Toso and Resende [2015], contendo as implementações de todas as fases do BRKGA. O código original foi alterado para adaptação ao mTTP, compilado com g++ 5.2.1 e opção de otimização -O3 foi utilizada.

6.1.1 Ajuste de Parâmetros

Para o ajuste da melhor configuração dos parâmetros, foi utilizado o pacote *irace* proposto por López-Ibáñez et al. [2016]. O *irace* é uma ferramenta que possui procedimentos de configuração automática utilizando métodos estatísticos e de corrida iterada. Além dos parâmetros do BRKGA também foi ajustada a ordem em que as buscas locais são aplicadas. O número máximo de gerações, tamanho da população e o ρ utilizado nas buscas locais foram ajustados manualmente verificando o tempo de execução médio para cada instância. A melhor configuração das buscas locais encontrada pelo *irace* foi a sequência RS, TS, PRS, PTS e por último, HAS. Os demais parâmetros são apresentados na Tabela 6.1.

Tabela 6.1: Parâmetros definidos para o BRKGA.

Parâmetro	Valor Definido
Tamanho da população	$3\times n$
Tamanho do conjunto elite	25%
Tamanho do conjunto mutante	20%
Probabilidade de herdar gene elite na reprodução	85%
Número de gerações	800
ρ	80%

6.1.2 Comparação com Resultados da Literatura

As instâncias utilizadas e as referências dos valores das melhores soluções estão disponíveis na seção de instâncias espelhadas no *site* oficial do pesquisador Michael Trick `http://mat.gsia.cmu.edu/TOURN/`. Para esse trabalho foram consideradas as instâncias da *National League* (NL) e as instâncias circulares (*circ*). Devido ao uso de componentes de aleatoriedade, o método proposto foi executado independentemente 10 vezes para cada instância.

As Tabelas 6.2 e 6.3 apresentam os resultados para cada instância considerada. São apresentados o valor da melhor solução conhecida na literatura (B^*), o melhor valor da solução obtida pelo BRKGA (S^*), o valor médio das soluções (S), o tempo de execução médio em segundos (T), a geração média de convergência das soluções (*conv*), a distância percentual entre os resultados (*gap*) obtidos e os melhores valores das soluções para cada instância e o desvio padrão (σ) em relação ao valor médio das soluções. Os melhores resultados da literatura são devidos a diferentes métodos, não havendo um método específico que reúna todos os melhores resultados. A distância percentual é calculada como $gap = \frac{S^* - B^*}{B^*} \times 100$.

Tabela 6.2: Resultados para instâncias NL.

Instância	B^*	S^*	S	T (s)	*conv*	*gap* (%)	σ
nl4	8276	8276	8276,00	0,55	5,4	0,00	0,00
nl6	26588	26588	26588,00	3,30	28,3	0,00	0,00
nl8	41928	43620	43743,10	12,04	447,7	4,04	139,43
nl10	63832	70386	71101,20	35,27	399,5	10,23	447,05
nl12	119608	130013	131540,70	86,84	363,6	8,70	962,41
nl14	199363	236461	239885,10	188,62	500,8	18,61	2212,51
nl16	278305	313555	317536,70	375,91	442,8	12,67	2776,91

De acordo com os dados reportados para as instâncias NL, o método proposto demonstra ser competitivo ao gerar soluções próximas àquelas geradas pelos melhores métodos presentes na literatura em baixo tempo computacional. Adicionalmente, o método demonstra consistência ao apresentar desvio padrão baixo, menor que 1% em execuções independentes. Por fim, o método também demonstra boa convergência, atingindo as melhores soluções próximo à metade do número máximo de iterações.

Tabela 6.3: Resultados para instâncias circulares.

Instância	B^*	S^*	S	T (s)	*conv*	*gap* (%)	σ
circ4	20	20	20,00	0,55	0,00	0,00	0,00
circ6	72	72	72,00	3,21	0,56	0,00	0,00
circ8	140	150	151,80	11,86	168,4	7,14	0,60
circ10	272	300	302,80	34,62	370,10	10,29	2,71
circ12	432	496	506,60	85,52	387,10	14,81	4,10
circ14	672	756	784,20	187,87	377,00	12,50	11,81
circ16	968	1102	1107,60	374,26	299,2	13,84	4,63
circ18	1306	1570	1582,89	846,08	373,89	20,21	9,10
circ20	1852	2134	2163,33	1461,20	305,77	15,23	18,26

Os resultados gerados para as instâncias circulares possuem qualidade similar aos obtidos

para as instâncias NL. Como este conjunto possui maiores instâncias é possível verificar um aumento no tempo de execução para as instâncias com 18 e 20 equipes. A convergência e o desvio padrão são menores para este conjunto de instâncias, ao passo que o *gap* médio apresentou leve aumento.

Capítulo 7

Conclusões

O *Mirrored Traveling Tournament Problem* (mTTP) é um problema $\mathcal{NP}$-Difícil que tem por objetivo otimizar a geração de tabelas de confrontos de torneios esportivos, minimizando a total distância viajada por todas as equipes durante o campeonato. Há ampla aplicação prática para métodos de solução deste problema, nos mais diferentes esportes praticados. Tabelas com distâncias viajadas reduzidas são imprescindíveis para minimizar os custos com transportes das equipes e aumentar o bem estar dos atletas.

Neste trabalho foi apresentada a abordagem ao mTTP aplicando-se a recente metaheurística Algoritmo Genético de Chaves Aleatórias Viciadas em conjunto com o método do polígono e cinco diferentes buscas locais. Os experimentos computacionais consideraram instâncias *benchmark* da literatura baseados em casos reais e compararam o método proposto às melhores soluções já obtidas por diferentes métodos da literatura. Os resultados obtidos demonstram que o método é competitivo, obtendo baixa divergência em relação às melhores soluções disponíveis para a maioria das instâncias em baixo tempo computacional. Adicionalmente, o método proposto apresenta rápida convergência e variação da qualidade das soluções abaixo de 1% em diferentes execuções independentes.

Os trabalhos futuros incluem testar diferentes esquemas de codificação do BRKGA e diferentes operadores de reprodução e mutação. Adicionalmente, serão consideradas modificações como o uso de operadores adaptativos e controle de diversidade do conjunto elite.

Referências Bibliográficas

A Anagnostopoulos, L Michel, P Van Hentenryck, and Y Vergados. A simulated annealing approach to the traveling tournament problem. In *Journal of Scheduling*. Citeseer, 2003.

Tankut Atan and Olgu Pelin Hüseyinoğlu. Simultaneous scheduling of football games and referees using turkish league data. *International Transactions in Operational Research*, 24 (3):465–484, 2017.

James C Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160, 1994.

Fabrício Biajoli and Luiz Lorena. Mirrored traveling tournament problem: An evolutionary approach. *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, pages 208–217, 2006.

Kevin KH Cheung. A benders approach for computing lower bounds for the mirrored traveling tournament problem. *Discrete Optimization*, 6(2):189–196, 2009.

Marco Antonio Moreira de Carvalho and Luiz Antonio Nogueira Lorena. New models for the mirrored traveling tournament problem. *Computers & Industrial Engineering*, 63(4):1089–1095, 2012.

Federico Della Croce and Dario Oliveri. Scheduling the italian football league: An ilp-based approach. *Computers & Operations Research*, 33(7):1963–1974, 2006.

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

JH Dinitz, E Lamken, and WD Wallis. Scheduling a tournament. *Handbook of Combinatorial Designs*, pages 578–584, 1995.

Guillermo Durán, Mario Guajardo, and Denis Sauré. Scheduling the south american qualifiers to the 2018 fifa world cup by integer programming. *European Journal of Operational Research*, 2017.

Kelly Easton, George Nemhauser, and Michael Trick. The traveling tournament problem description and benchmarks. In *International Conference on Principles and Practice of Constraint Programming*, pages 580–584. Springer, 2001.

Kelly Easton, George Nemhauser, and Michael Trick. Solving the travelling tournament problem: A combined integer programming and constraint programming approach. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 100–109. Springer, 2002.

José Fernando Gonçalves and Mauricio GC Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

Dries Goossens and Frits Spieksma. Scheduling the belgian soccer league. *Interfaces*, 39(2): 109–118, 2009.

Daya Gupta, Lavika Goel, and Vipin Aggarwal. A hybrid biogeography based heuristic for the mirrored traveling tournament problem. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, pages 325–330. IEEE, 2013.

Stefan Irnich. A new branch-and-price algorithm for the traveling tournament problem. *European Journal of Operational Research*, 204(2):218–228, 2010.

Tiago Januario, Sebastián Urrutia, and Dominique de Werra. Sports scheduling search space connectivity: A riffle shuffle driven approach. *Discrete Applied Mathematics*, 211:113–120, 2016.

Andrew Lim, Brian Rodrigues, and Xingwen Zhang. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3):1459–1478, 2006.

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

Rasmus V Rasmussen and Michael A Trick. A benders approach for the constrained minimum break problem. *European Journal of Operational Research*, 177(1):198–213, 2007.

Celso C Ribeiro and Sebastián Urrutia. Scheduling the brazilian soccer tournament with fairness and broadcast objectives. *Lecture notes in computer science*, 3867:147–157, 2006.

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

Celso C Ribeiro and Sebastián Urrutia. Scheduling the brazilian soccer tournament: Solution approach and practice. *Interfaces*, 42(3):260–272, 2012.

William M Spears and Kenneth D De Jong. On the virtues of parameterized uniform crossover. Technical report, Naval Research Lab, Washington D.C., 1995.

Clemens Thielen and Stephan Westphal. Complexity of the traveling tournament problem. *Theoretical Computer Science*, 412(4-5):345–351, 2011.

Rodrigo F Toso and Mauricio GC Resende. A c++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 30(1):81–93, 2015.

Sebastián Urrutia and Celso C Ribeiro. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Applied Mathematics*, 154(13):1932–1938, 2006.

David C Uthus, Patricia J Riddle, and Hans W Guesgen. Dfs* and the traveling tournament problem. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, pages 279–293. Springer, 2009.

Pascal Van Hentenryck and Yannis Vergados. Traveling tournament scheduling: a systematic evaluation of simulated annealling. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 228–243. Springer, 2006.