

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BRUNA ALMEIDA OSTI

**OPERADOR DE CRUZAMENTO BASEADO EM PARTIÇÕES
APLICADO AO PROBLEMA DA ÁRVORE DE STEINER EM
GRAFOS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2020

BRUNA ALMEIDA OSTI

**OPERADOR DE CRUZAMENTO BASEADO EM PARTIÇÕES
APLICADO AO PROBLEMA DA ÁRVORE DE STEINER EM
GRAFOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Danilo Sipoli Sanches
Universidade Tecnológica Federal do Paraná

CORNÉLIO PROCÓPIO
2020



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Computação
Engenharia de Computação



TERMO DE APROVAÇÃO

Operador de cruzamento baseado em partições aplicado ao problema da árvore de Steiner em grafos

por

Bruna Almeida Osti

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Programa de Graduação em 2020 da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 14/10/2020

Prof. Dr. Danilo Sipoli Sanches

Prof. Dr. Lucas Dias Hiera Sampaio

Prof. Dr. Henrique Yoshikazu Shishido

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

À meu pai Vanderlei (in memoriam), que já se
foi, mas que se fez presente nos momentos mais
importantes da minha vida

AGRADECIMENTOS

Gostaria de agradecer à meus pais Lenilde e Vanderlei (in memorian) com quem compartilhei momentos de alegria, tristeza e ansiedade, com todo meu amor e gratidão pelo oportunidade dos estudos e apoio para concluí-los, sem o qual não teria sido possível.

Agradeço ao meu namorado Eduardo por toda a paciência, compreensão e apoio durante o curso e durante a realização deste trabalho.

A toda instituição de ensino da UTFPR, que me proporcionou a infraestrutura acadêmica e todo o ambiente de aprendizagem que foram fundamentais para o meu crescimento acadêmico e pessoal.

Agradeço ao meu orientador Prof. Danilo Sipoli Sanches que me proporcionou a oportunidade de participar da pesquisa acadêmica tanto na iniciação científica quanto para o trabalho de conclusão de curso, e me guiou durante as fases de dificuldade, sendo não só meu professor, mas meu amigo.

Também agradeço ao Prof. Josimar Silva Rocha, ao Prof. Renato Tinós e ao discente Giliard Godoi pelo auxílio durante o desenvolvimento do trabalho.

Por último, agradeço a todos que estiveram do meu lado durante toda a jornada acadêmica, me apoiando de diversas maneiras, sem vocês esse trabalho não seria possível.

“DO, OR DO NOT, THERE IS NO TRY.”

Frase atribuída ao personagem Mestre Yoda (Star Wars).

RESUMO

OSTI, Bruna. Operador de cruzamento baseado em partições aplicado ao problema da árvore de Steiner em grafos. 2020. 30 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Neste trabalho é proposto um método para aplicar o operador de cruzamento de partição generalizado (GPX) no problema da árvore de Steiner em grafos (STPG). Em geral, o operador de cruzamento de partição generalizado tem como princípio aproveitar as melhores partes de duas soluções, garantindo sempre que a melhor solução gerada seja sempre melhor ou que mantenha o custo das soluções iniciais, sem aumentar a complexidade computacional do operador. O modelo é funcional, portanto, é possível reutilizá-lo para outros problemas de otimização combinatória em grafos, com outros tipos de restrições apenas alterando algumas estruturas do algoritmo.

Palavras-chave: Árvore de Steiner em Grafos. Algoritmo Genético. GPX.

ABSTRACT

OSTI, Bruna. Partition-based crossover operator applied to Steiner tree problem in graphs. 2020. 30 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

This work proposes a method to apply the generalized partition crossing operator (GPX) to the Steiner tree graph problem (STPG). In general, the generalized partition crossover operator has the principle of taking advantage of the best parts of two solutions, always ensuring that the best generated solution is always better or that it keeps the cost of the initial solutions without increasing the computational complexity of the operator. The model is functional, so you can reuse it for other combinatorial optimization problems in graphs, with other constraints just by changing some algorithm structures.

Keywords: Steiner tree problem in graphs. Genetic Algorithm. GPX.

LISTA DE FIGURAS

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Adaptação da superfície de aptidão | 5 |
| Figura 2 – (a) Dataset Completo (b) Árvore de Steiner | 6 |
| Figura 3 – Heurística SPH | 6 |
| Figura 4 – Partições formadas pelo operador PX - (a) Retirada de arestas em comuns, (b) Formação das partições | 7 |
| Figura 5 – Partições formadas pelo GPX - (a) Retirada de arestas em comuns, (b) Formação das partições | 8 |
| Figura 6 – Operador de mutação de um ponto | 12 |
| Figura 7 – Operador de cruzamento de um ponto | 13 |
| Figura 8 – Aplicação do Algoritmo 3 com componente não recombinante (a) Soluções: vermelha e azul, (b) Partições encontradas, (c) Grafo Simplificado e (d) Solução Recombinada | 14 |
| Figura 9 – Função custo médio para cada abordagem | 20 |
| Figura 10 – Tempo médio para cada abordagem | 20 |
| Figura 11 – Função custo médio melhor para a abordagem Híbrido | 23 |
| Figura 12 – Função custo médio melhor para a abordagem GPX | 24 |
| Figura 13 – Datasets semelhantes | 25 |

LISTA DE QUADROS

| | |
|----------------------------------------------|----|
| Quadro 1 – Abordagens relacionadas | 8 |
| Quadro 2 – Base de dados | 16 |

LISTA DE TABELAS

| | |
|------------------------------------------------------------------------|----|
| Tabela 1 – Abordagens implementadas | 17 |
| Tabela 2 – Resultados obtidos para os datasets do conjunto B | 22 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|------------------------------------------------------------------------------------------------------|
| AG | Algoritmo genético (<i>Genetic algorithm</i>) |
| ADH | Heurística da distância média (<i>Average Distance Heuristic</i>) |
| BFS | Busca em largura (<i>Breadth-first search</i>) |
| GPX | Operador de cruzamento de partições generalizado (<i>Generalized partition crossover operator</i>) |
| MST | Árvore mínima (<i>Minimum spanning tree</i>) |
| PRIM | Algoritmo para calcular árvore mínimas (<i>Algorithm to calculate the minimum spanning tree</i>) |
| PSO | Enxame de partículas (<i>Particle swarm optimization</i>) |
| PX | Operador de cruzamento de partições (<i>Partition crossover operator</i>) |
| SDH | Heurística da menor distância (<i>Shortest Distance Heuristic</i>) |
| SPH | Heurística do caminho mínimo (<i>Shortest Path Heuristic</i>) |
| SPOH | Heurística do caminho mínimo com origem (<i>Shortest Path Heuristic with Origin</i>) |
| STPG | Problema da árvore de Steiner em grafos (<i>Steiner tree problem in graphs</i>) |
| TRIM | Árvore mínima podada (<i>Prunning minimum spanning tree</i>) |
| TSP | Problema do caixeiro-viajante (<i>Travelling salesman problem</i>) |

LISTA DE SÍMBOLOS

| | |
|-------------|---------------------------------------------------------------------|
| A | Arestas |
| T | Árvore de Steiner |
| $P(t)$ | Conjunto P indicando população principal |
| $P(t)'$ | Conjunto P indicando subpopulação |
| a | Constante correspondente a quantidade de vértices do dataset |
| L | Comprimento |
| \subseteq | Contido |
| \pm | Desvio padrão amostral |
| c | Distância entre cada vértice; |
| $f_c(S_i)$ | Função custo da solução dado um indivíduo “i” da população |
| $f(S_i)$ | Função <i>Fitness</i> da solução dado um indivíduo “i” da população |
| G | Grafo |
| min | Minimização |
| N | Par de vértices |
| \in | Pertence |
| T_f | Quantidade de vértices terminais faltantes na solução |
| \sum | Somatório |
| \cup | União |
| V | Vértices |

LISTA DE ALGORITMOS

| | |
|------------------------------------------------------------------|----|
| Algoritmo 1 – Algoritmo Genético | 4 |
| Algoritmo 2 – GPX para o Problema da Árvore de Steiner | 14 |
| Algoritmo 3 – GPX | 17 |
| Algoritmo 4 – Híbrido | 18 |

SUMÁRIO

| | |
|----------------------------------------------------------------|-----------|
| 1 – INTRODUÇÃO | 1 |
| 1.1 PROBLEMA | 1 |
| 1.2 JUSTIFICATIVA | 2 |
| 1.3 OBJETIVOS | 2 |
| 1.3.1 Objetivos específicos | 2 |
| 1.4 ORGANIZAÇÃO DO TEXTO | 2 |
| 2 – REVISÃO DE LITERATURA | 3 |
| 2.1 DEFINIÇÕES E CONCEITOS | 3 |
| 2.1.1 Algoritmos Evolucionários | 3 |
| 2.1.2 Adaptação do domínio do problema de otimização | 4 |
| 2.1.3 Formulação do Problema | 5 |
| 2.2 HEURÍSTICAS AUXILIARES | 6 |
| 2.2.1 SPH | 6 |
| 2.2.2 SPOH | 7 |
| 2.3 OPERADOR DE CRUZAMENTO DE PARTIÇÕES | 7 |
| 2.3.1 PX | 7 |
| 2.3.2 GPX | 8 |
| 2.4 TRABALHOS RELACIONADOS | 8 |
| 3 – MODELAGEM DO PROBLEMA UTILIZANDO ALGORITMO GENÉTICO | 10 |
| 3.1 RESOLUÇÃO DO STPG COM ALGORITMO GENÉTICO | 10 |
| 3.1.1 Representação da solução | 10 |
| 3.1.2 Cálculo de fitness das soluções | 11 |
| 3.1.3 Geração da população inicial | 11 |
| 3.1.4 Operadores básicos | 12 |
| 3.1.4.1 Operador de mutação de um ponto | 12 |
| 3.1.4.2 Operador de cruzamento de um ponto | 12 |
| 3.2 O OPERADOR <i>GENERALIZED PARTITION CROSSOVER</i> | 13 |
| 4 – METODOLOGIA | 16 |
| 4.1 TECNOLOGIAS E FERRAMENTAS | 16 |
| 4.2 BASE DE DADOS | 16 |
| 4.3 ABORDAGENS UTILIZADAS | 16 |
| 4.3.1 AG | 17 |
| 4.3.2 GPX | 17 |

| | | |
|----------|----------------------------------------------------------------------|---------------|
| 4.3.3 | Híbrido | 18 |
| 5 | – ANÁLISE E DISCUSSÃO DOS RESULTADOS | 19 |
| 5.1 | PARAMETRIZAÇÃO DOS ALGORITMOS | 19 |
| 5.2 | ANÁLISE GERAL | 19 |
| 5.2.1 | Datasets b01, b02, b04, b05, b07, b08, b09, b12, b16 e b17 | 23 |
| 5.2.2 | Datasets b03, b11, b15 e b18 | 23 |
| 5.2.3 | Datasets b06, b10, b13 e b14 | 24 |
| 6 | – CONSIDERAÇÕES FINAIS | 26 |
| | REFERÊNCIAS | 27 |
| | Apêndices | 29 |
| | APÊNDICE A–Datasets do Conjunto B detalhados | 30 |

1 INTRODUÇÃO

A otimização combinatória é constantemente utilizada para modelar e resolver grandes problemas de logística, tendo como exemplo um dos problemas mais conhecidos e estudados nessa área: o problema da árvore de Steiner, foco deste trabalho. No qual, apesar de tratar-se de um problema NP-Difícil é de ampla eficácia em modelagens de projetos de circuitos elétricos, redes de comunicação e de transporte (CHENG; DU, 2002).

Segundo Gilbert e Pollak (1968), o problema da árvore de Steiner tem como objetivo encontrar um caminho mínimo que interconecte " n " pontos fixos dados. Para atingir tal objetivo é permitida a adição de pontos auxiliares que minimizem o comprimento do grafo final, que é a soma dos comprimentos de todas as arestas conectadas, estes pontos auxiliares são chamados de pontos de Steiner.

Entretanto, devido à complexidade do problema surgiram algumas outras variações do problema que buscam limitar o problema em função da aplicação, sendo eles: o problema da árvore de Steiner em grafos (STPG, Steiner Tree Problem in Graphs) (FOULDS; RAYWARD-SMITH, 1983), o problema da árvore de Steiner retilíneo (RSTP, Retilinear Steiner Tree Problem) (HANAN, 1966), e o problema da árvore de Steiner euclidiano (ESTP, Euclidean Steiner Tree Problem) (HWANG, 1976). Desta forma, devido à grande variedade de aplicações este trabalho terá como base o problema da árvore de Steiner em grafos (STPG).

O Problema de Steiner surgiu a partir do Problema de Fermat, que consiste em, dado um triângulo, encontrar o ponto cuja soma das distâncias aos vértices é mínima. Esse problema foi generalizado para qualquer figura em qualquer dimensão e hoje é conhecido como Problema de Steiner Euclidiano. Posteriormente, formulou-se uma versão em grafos que tem complexidade NP-Completo (PRöMEL; STEGER, 2002).

Com o intuito de encontrar soluções para o STPG, será utilizado um algoritmo genético desenvolvido de acordo com restrições e peculiaridades do problema, também será utilizado o operador de cruzamento GPX para buscar uma possível melhora na solução, sendo este operador baseado no *Partition Crossover* (PX), definido por Whitley, Hains e Howe (2009) para caminhar nos ótimos locais do TSP (Problema do Caixeiro-Viajante).

1.1 PROBLEMA

Um dos problemas mais comuns na implementação da solução para o problema da árvore de Steiner é a necessidade do poder computacional envolvido no processamento, pois todo o processo acaba sendo muito custoso devido a quantidade de iterações necessárias e o tamanho dos dados disponíveis. Deste modo, a utilização de um método heurístico que trouxesse um resultado aproximado mas muito preciso acaba tornando-se indispensável, neste caso optamos pela utilização de algoritmos evolucionários.

1.2 JUSTIFICATIVA

De acordo com [Whitley, Hains e Howe \(2010\)](#) o operador GPX tem como princípio caminhar entre os ótimos locais da fitness landscape como descrito na [Subseção 2.1.2](#), podendo combinar dois ótimos locais para encontrar outro ótimo local, portanto o algoritmo sempre mantém ou melhora a solução.

A partir disso, será investigada a eficiência do algoritmo genético para otimizar as soluções para o problema da árvore de Steiner em grafos, sendo aplicado o operador de cruzamento de partição generalizado (GPX) para caminhar entre os mínimos locais, baseado no operador de cruzamento de partição (PX) proposto por [Whitley, Hains e Howe \(2009\)](#).

1.3 OBJETIVOS

O principal objetivo deste trabalho consiste em desenvolver um algoritmo genético para o STPG e melhorá-lo através da implementação do operador de cruzamento de partições generalizado (GPX).

1.3.1 Objetivos específicos

Os objetivos específicos deste trabalho foram determinados para atingir o objetivo geral descrito na seção [1.3](#) e estão listados abaixo:

- Entender os conceitos e passos utilizados pelo GPX;
- Investigar informações e conceitos a cerca do STPG;
- Investigar e definir ideias para a representação das soluções;
- Investigar métodos de cálculo de *fitness* para o Algoritmo Genético;
- Investigar abordagens de implementação para o STPG;
- Propor uma função de custo e *fitness* para o problema.

1.4 ORGANIZAÇÃO DO TEXTO

- No [Capítulo 2](#) será apresentado todo o embasamento teórico do artigo, conceitos e definições além de explicações sobre o problema e o operador que será utilizado.
- No [Capítulo 3](#), será apresentada a proposta do trabalho, apresentando exemplos e como será estruturada a modelagem do problema dado o método.
- No [Capítulo 4](#) será apresentado a metodologia do trabalho, tanto no quesito das tecnologias e ferramentas utilizadas, quanto a base de dados e metodologia de validação dos algoritmos.
- No [Capítulo 5](#) serão analisados os resultados obtidos pelo algoritmo proposto.
- No [Capítulo 6](#) serão abordados as considerações finais, possíveis melhorias e aplicações.

2 REVISÃO DE LITERATURA

Neste capítulo, visando aprofundar os conceitos e definições essenciais para o entendimento do trabalho, serão abordados alguns conceitos sobre o funcionamento dos algoritmos evolucionários. Posteriormente, será apresentada a definição do STPG, além de alguns operadores de cruzamento específicos para o problemas de otimização combinatória em grafos. Por fim, serão mencionados alguns trabalhos relacionados relevantes para o problema STPG buscando outras visões a respeito do problema.

2.1 DEFINIÇÕES E CONCEITOS

Nesta seção serão apresentados conceitos necessários para o entendimento do trabalho, onde serão abordados alguns conceitos sobre algoritmos evolucionários, passando pela definição do problema STPG.

2.1.1 Algoritmos Evolucionários

Os algoritmos evolucionários utilizam modelos computacionais baseados nos processos naturais de evolução, independente da grande variedade de algoritmos propostos todos tem o mesmo conceito de evolução, ou seja, apresentando a simulação das espécies através de seleção, mutação e reprodução. Processos estes que dependem apenas da adaptação dos indivíduos da espécie dentro do ambiente em que vivem, baseando-se puramente na teoria da evolução e na hereditariedade, isto é, partem do pressuposto que indivíduos com boas características genéticas tem maiores chances de sobrevivência e de produzirem indivíduos cada vez mais aptos, de maneira que os indivíduos menos aptos tenderão a desaparecer ([LINDEN, 2012](#)). Podemos agrupar os algoritmos evolutivos em três categorias:

- Algoritmos Genéticos;
- Estratégias de Evolução;
- Programação Genética.

As técnicas de busca e otimização evolutivas operam sobre uma população de candidatos em paralelo, assim mantêm-se a diversidade nas áreas do espaço da solução, logo, tais técnicas têm maior probabilidade de atingir áreas promissoras do espaço de busca ([REZENDE, 2003](#)). Dentro desse aspecto, os algoritmos genéticos diferem dos métodos tradicionais de busca e otimização, pois combinam uma propensão para melhoria da solução com a exploração aleatória e com a troca de informações entre processos de busca paralelos. Os termos utilizados para simbolizar elementos do algoritmo genético estão descritos abaixo:

- **Cromossomo:** representação da solução;
- **Fitness/Aptidão:** função de avaliação da qualidade das soluções;
- **Mutação:** alteração aleatória nas características do cromossomo;
- **Cruzamento:** recombinação de características do cromossomo;
- **População:** conjunto de cromossomos;
- **Geração:** número de populações geradas a partir das anteriores;
- **Seleção:** escolha dos cromossomos que gerarão descendentes nas próximas populações.

O algoritmo [Algoritmo 1](#) apresenta o processo de execução de um algoritmo genético, primeiramente é gerado uma população aleatória, a partir dessa população o algoritmo é iniciado. A cada geração a população é avaliada a fim de satisfazer a condição de parada, enquanto não estiver satisfeita o algoritmo continua sua execução.

A partir da avaliação da população, os cromossomos são selecionados para gerar uma população auxiliar que será utilizada para procedimentos de evolução, como por exemplo: Mutação e Cruzamento. O método de seleção é escolhido de acordo com o algoritmo, entretanto, é utilizado para dar variedade na população, focando nos melhores indivíduos sem que exclua os menos aptos, pois podem conter genes que são partes de uma solução boa. Após a seleção, é feito os procedimentos de evolução, e então o algoritmo é repetido.

Algoritmo 1: Algoritmo Genético

Entrada: $t \leftarrow 0$

Saída: Solução esperada para o problema

início

Gerar população $P(t)$

Avaliar os indivíduos da população $P(t)$

repita

Selecionar os indivíduos de $P(t)$ para cruzamento criando-se $P(t)'$

Aplicar cruzamento em $P(t)'$

Aplicar mutação em $P(t)'$

$P(t) = P(t)'$

Avaliar os indivíduos da população $P(t)$

$t = t + 1$

até condição de parada alcançada;

fim

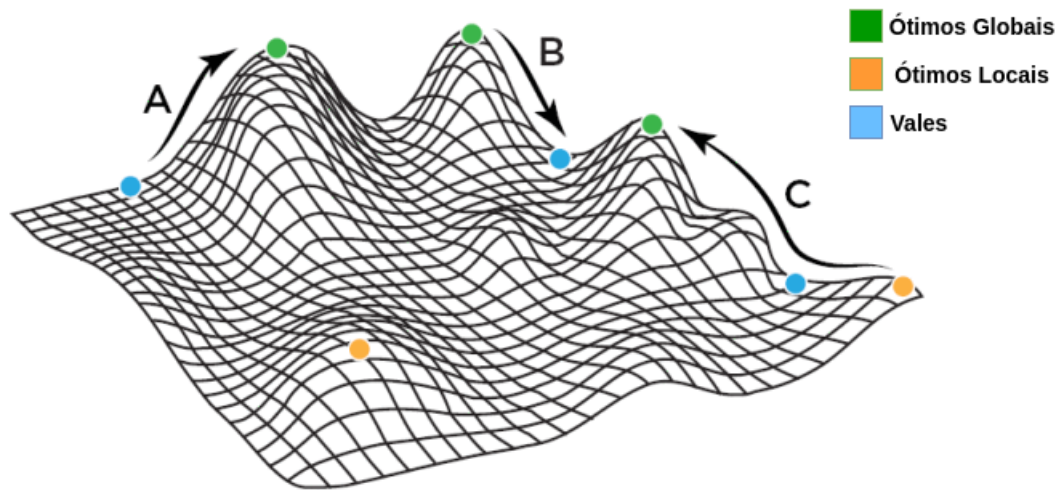
2.1.2 Adaptação do domínio do problema de otimização

A adaptação do domínio do problema é realizada através da modelagem da área de busca de um algoritmo como superfície de aptidão, o qual é utilizado para a visualização da relação entre a composição genética (genótipo) e aptidão evolutiva (capacidade de sobreviver e se reproduzir). A [Figura 1](#) descreve que cada quadrado da imagem representa um genótipo e

os picos representam a sua aptidão, ou seja, vales representam a baixa aptidão e picos (ótimos locais e globais) representam a alta aptidão.

No caso estudado o algoritmo busca um mínimo global como solução pois é um problema de minimização da função de aptidão. Um algoritmo de busca completo sempre encontra um objetivo, caso ele exista, por outro lado, um algoritmo ótimo sempre encontra um mínimo global (NORVIG, 2013).

Figura 1 – Adaptação da superfície de aptidão



Fonte: (OLSON, 2013)

Entretanto, se o algoritmo encontra algum ótimo local como solução é muito difícil escapar da área de busca e alcançar o ótimo global. Torna-se necessário a aplicação de algum método auxiliar para causar uma leve perturbação na solução visando que esta alcance outras áreas próximas que ainda não foram exploradas.

2.1.3 Formulação do Problema

Considerando um grafo não direcionado conexo $G = (V, A)$, e um subconjunto não vazio de N (par de vértices) de V . O problema de Steiner consiste em encontrar um subgrafo(conexo) $G' = (V', A')$ de G que tenha um caminho em G' entre cada par de vértices N ; e que o comprimento $L(G') = \sum_{e \in A'} f(e)$ seja mínimo (CIESLIK, 1998). A Equação (1) demonstra a definição do problema.

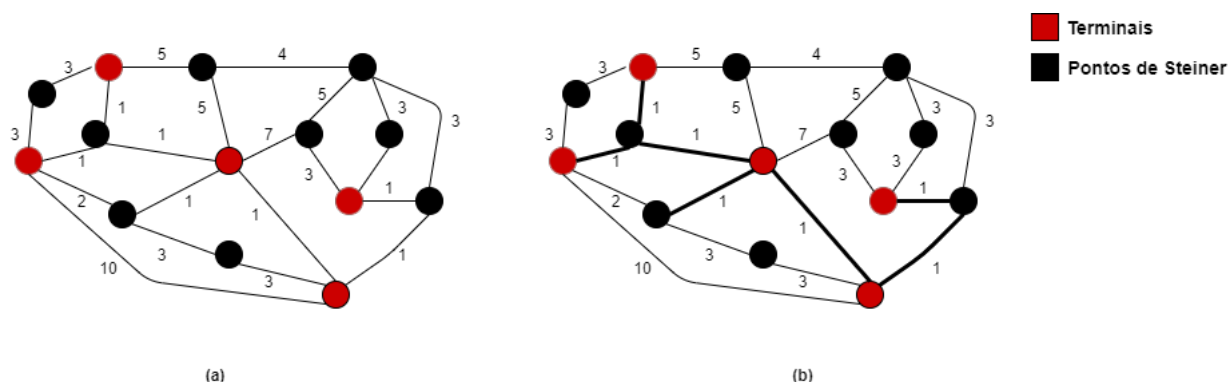
$$L(T(N \cup V_o)) = \min\{L(T(N \cup V')) : V' \subseteq V \setminus N\} \quad (1)$$

No qual, $T(N \cup V_o) = STPG$ para N em G .

A Figura 2 demonstra primeiramente a árvore completa com todos os pontos contidos no dataset, posteriormente a árvore de Steiner. Os vértices terminais são chamados de pontos

obrigatórios e todos os outros vértices do conjunto são opcionais, ou seja, os vértices opcionais que entram na solução são chamados de pontos de Steiner, portanto, a solução final abrange apenas um subconjunto que é denominada árvore de Steiner (HWANG, 1992).

Figura 2 – (a) Dataset Completo (b) Árvore de Steiner



Fonte: Autoria Própria

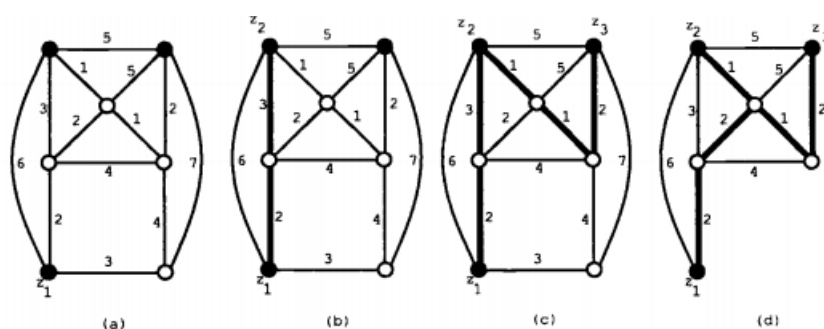
2.2 HEURÍSTICAS AUXILIARES

Nessa seção falaremos sobre heurísticas auxiliares utilizadas para a construção de árvores conexas que serão utilizadas para iniciar a solução em algumas das abordagens, foram utilizados as heurísticas SPH (Shortest Path Heuristic) e SPOH (Shortest Path Heuristic with origin).

2.2.1 SPH

Na heurística SPH (Shortest Path Heuristic) primeiramente é escolhido um vértice terminal arbitrário para dar início a solução, cada passo construído é calculado através do algoritmo de Dijkstra e o caminho contendo o vértice inicial é o caminho mínimo como descrito na Figura 3. Após a construção completa da árvore é opcional a aplicação do algoritmo de Prim para remover possíveis ciclos (HWANG, 1992).

Figura 3 – Heurística SPH



Fonte: (HWANG, 1992)

2.2.2 SPOH

A heurística SPOH (Shortest Path with Origin) consiste em encontrar o menor caminho de um vértice terminal arbitrário para todos os outros terminais de forma iterativa. Geralmente tende a ter resultados inferiores ao SPH (HWANG, 1992).

2.3 OPERADOR DE CRUZAMENTO DE PARTIÇÕES

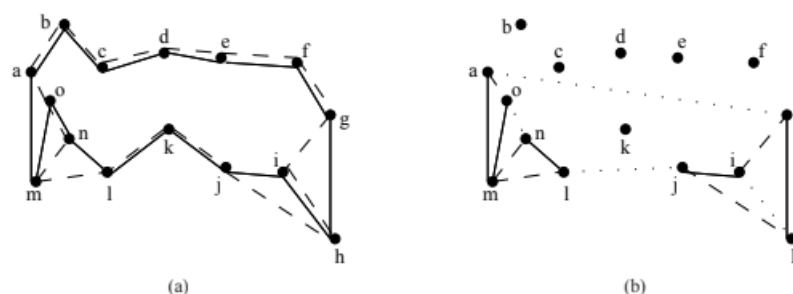
Nesta seção falaremos um pouco sobre as duas versões do operador de cruzamento de partições: PX e GPX. Todas as versões foram criadas especificamente para o problema de otimização combinatória TSP simétrico com o intuito de melhorar as soluções encontradas, tendo como princípio serem respeitadas e garantirem a transmissão dos alelos das soluções recombinantes, isto é, garantem que a solução encontrada será melhor ou equivalente as anteriores.

Além disso, por ser um operador de grafos é possível reutilizá-lo para outros problemas de otimização combinatória em grafos, com outros tipos de restrições apenas alterando algumas estruturas do algoritmo.

2.3.1 PX

O operador *Partition Crossover* (PX) utiliza a união de duas soluções no qual ambas são circuitos Hamiltonianos e procura por um tipo específico de partição no grafo construído como descrito na Figura 4, se a partição é encontrada, duas soluções únicas são construídas. Além disso, uma característica intrínseca do operador é a habilidade de encontrar ótimos locais a partir de outros ótimos locais (WHITLEY; HAINS; HOWE, 2009).

Figura 4 – Partições formadas pelo operador PX - (a) Retirada de arestas em comuns, (b) Formação das partições

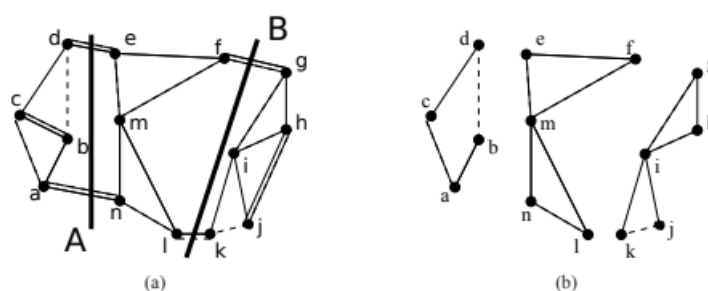


Fonte: (WHITLEY; HAINS; HOWE, 2009)

2.3.2 GPX

O *Generalized Partition Crossover* (GPX) é a versão generalizada do *Partition Crossover* (PX), pois utiliza a união de duas soluções para construir um grafo, entretanto o grafo construído contém múltiplas partições com as características que o *Partition Crossover* está procurando como mostra a [Figura 5](#). O GPX é capaz de encontrar essas partições e usá-las em uma única combinação sem aumentar o custo de processamento.

Figura 5 – Partições formadas pelo GPX - (a) Retirada de arestas em comuns, (b) Formação das partições



Fonte: ([WHITLEY; HAINS; HOWE, 2009](#))

2.4 TRABALHOS RELACIONADOS

Devido à estrutura do trabalho, uma proposição de método, não foi encontrado outros trabalhos que proponham a aplicação do GPX sobre o STPG. Deste modo serão apresentados trabalhos que propuseram outros métodos em geral, no qual o resumo geral está apresentado no [Quadro 1](#) e as demais informações estão apresentadas abaixo.

Quadro 1 – Abordagens relacionadas

| Trabalho | Autor | Ano | Abordagem |
|----------------------------------------------------------------------------------|----------------------------------------|------|-------------------------------------------------|
| "The Steiner Problem in Graphs" | S. E. Dreyfus R. A. Wagner | 1971 | Programação Dinâmica - Algoritmo exato |
| "Solving the graphical steiner tree problem using genetic algorithms." | Kapsalis, A.; Rayward-Smith; Smith, G. | 1993 | AG |
| "A particle swarm optimization for steiner tree problem" | Ma e Liu | 2010 | PSO |
| "A hybrid ensemble approach for the steiner tree problem in large graphs" | Bouchachia e Prosegger | 2011 | Híbrido Clusterizado: ACO + Spectral Clustering |
| "A GRASP for the steiner tree problem in graphs to support multicasting routing" | Costa, Rosset e Nascimento | 2015 | Grasp distribuído |

Fonte: Autoria Própria

[Dreyfus e Wagner \(1971\)](#) propuseram um algoritmo exato com base na programação dinâmica, o método é exponencial pois propõe que as recursões sejam derivadas da quantidade de pares de vértices (arestas).

Kapsalis, Rayward-Smith e Smith (1993) propuseram um algoritmo genético que utilizava sementes de árvores construídas através de outras heurísticas como o MST e o TRIM para convergir a população e os comparava com a solução gerada através da população criada aleatoriamente. Além disso, os comparava com mais três abordagens: o SDH, SPH e ADH. O método conseguiu encontrar o mínimo do dataset em todas as vezes, entretanto no artigo apenas foi utilizado os resultados para o dataset B da biblioteca OR-LIBRARY (BEASLEY, 1990).

Ma e Liu (2010) propuseram um algoritmo de enxame de partículas (*Particle Swarm Optimization - PSO*). No algoritmo, uma estrutura em árvore representação é usada para codificar uma partícula. Além disso, é utilizado a r-topologia do anel de vizinhança de partículas para melhorar a capacidade de pesquisa local e global do algoritmo PSO e uma partícula método de mutação para manter a diversidade da população de partículas. Os resultados indicam que o algoritmo proposto tenha bom desempenho de pesquisa para encontrar a árvore ideal de Steiner.

Bouchachia e Prosegger (2011) propuseram a aplicação de um algoritmo híbrido clusterizado, utilizando a combinação dos métodos *Spectral clustering* e *ACO*. Consistindo-se de três etapas: (1) Agrupamento espectral para segmentar grafos grandes, (2) Aplicação de várias colônias em cada segmento do grafo para encontrar soluções locais, (3) Aplicação da colônia de formigas no hiper-grafo. O método produz resultados próximos ao ótimo conhecido mas é inversamente proporcional ao número de clusters, pois a medida que a quantidade de clusters aumenta, a performance diminui.

Costa, Rosset e Nascimento (2015) propuseram uma meta-heurística baseada em vizinhança (GRASP) para encontrar uma solução aproximada para a STPG, com a fase de construção adaptada da estratégia do Jarnik-Prim, para ser um algoritmo semi-guloso. O método apresentou bons resultados, considerados adequados para uma implementação distribuída.

3 MODELAGEM DO PROBLEMA UTILIZANDO ALGORITMO GENÉTICO

Neste capítulo será abordado o modo em que o problema será modelado, desde a representação das soluções à como será verificado a qualidade das soluções. Além disso, será descrito como o operador *Generalized Partition Crossover* (GPX) será adaptado para o funcionamento dentro do escopo do STPG.

3.1 RESOLUÇÃO DO STPG COM ALGORITMO GENÉTICO

Nesta seção será descrito como o problema será modelado para o funcionamento do algoritmo genético, salientando desde como será representado a solução, passando pelos algoritmos necessários para inserir diversidade para a população: mutação e cruzamento. E por fim, apresentando os métodos de seleção utilizados para gerar os indivíduos das próximas gerações.

3.1.1 Representação da solução

Os indivíduos da população são representados de forma binária e na forma de dicionário, onde, na forma binária os vértices são representados como “1” se contidos na solução S , e como “0” se não estão contidos, tendo o seu início de forma arbitrária com o intuito de abordar-se uma área de busca maior.

$$Cromossomo = [1, 0, 0, 1, 1, 0, 1]$$

$$S = [1, \mathbf{1}, 0, 0, \mathbf{1}, 1, 1, 0, \mathbf{1}, 1]$$

Entretanto, deve-se observar que os vértices terminais não estão contidos nos cromossomos inicialmente, pois eles devem estar na solução final obrigatoriamente, caso contrário a solução não é factível. Isto é, os terminais apenas são adicionados para o cálculo da árvore mínima e respectivamente para o cálculo do fitness das soluções que será aprofundado na [Subseção 3.1.2](#).

Por outro lado, na forma de dicionário são representados através da conexão dos vértices (V_1 e V_2) e sua respectiva distância, como demonstrado abaixo.

$$S = \{V'_1 : \{V'_2 : Distância\}\}$$

Foi necessário a utilização de dois tipos de representações pois os operadores utilizados em cada abordagem funcionam de formas diferentes. Por um lado os operadores cruzamento e mutação de um ponto são calculados utilizando a estrutura de lista, por outro lado, o GPX utiliza a estrutura do grafo para operar.

3.1.2 Cálculo de fitness das soluções

Como descrito no [Subseção 2.1.2](#), o *fitness* é o nível de aptidão que uma solução possui, como o problema é de minimização quanto menor o valor da aptidão melhor será a solução, como descrito nas equações 2, 3 e 4 proposta pelos autores.

Em cada solução gerada o algoritmo de Prim é aplicado para que este calcule a árvore mínima conexa (conectada) e retire da solução os outros vértices que não se encontram ligados a nenhum outro vértice e também retire os ciclos. Portanto, após a geração da árvore mínima a aptidão $f(S_i)$ é calculada através da soma das arestas contidas na árvore gerada, como mostra a equação [Equação \(2\)](#).

$$f_c(S_i) = \sum_{j=1}^A c_j \quad (2)$$

No qual,

c - distância entre cada vértice;

A - quantidade de arestas;

$f_c(S_i)$ - função custo da solução dado um indivíduo " i " da população.

A aptidão verifica apenas se as arestas contidas na árvore são mínimas, entretanto a árvore de Steiner é um subconjunto de vértices do conjunto original, sendo assim, seus vértices também variam. Por esse motivo, é necessário adicionar uma penalização caso o algoritmo não contenha todos os vértices terminais, caso contrário é adicionado uma bonificação. A penalidade é calculada através da [Equação \(3\)](#) e a bonificação através da [Equação \(4\)](#).

$$f(S_i) = f(S_i) + a(T_f) \quad (3)$$

Caso contrário,

$$f(S_i) = f(S_i) - a \quad (4)$$

No qual,

a - constante correspondente a quantidade de vértices do dataset;

T_f - quantidade de vértices terminais faltantes na solução;

$f(S_i)$ - *fitness* da solução dado um indivíduo " i " da população.

3.1.3 Geração da população inicial

A fim de aumentar a variabilidade dos dados comparativos, foi decidido a utilização de mais de um método de geração de população inicial. A partir dessa variação é possível encontrar particularidades ou padrões de melhoria do cruzamento, se existir.

1. **Método 1:** a geração é feita escolhendo-se aleatoriamente os bits([0,1]) que entram na solução e posteriormente são adaptadas através dos operadores de cruzamento e mutação;
2. **Método 2:** a população completa é criada através das heurísticas SPH e SPOH discutidas na [Seção 2.2](#) ;
3. **Método 3:** parte das população é criada aleatoriamente e parte da solução é criada com as heurísticas da [Seção 2.2](#).

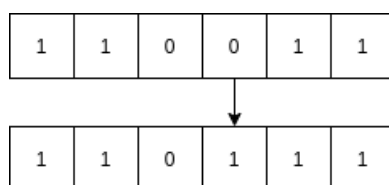
3.1.4 Operadores básicos

Os operadores são utilizados para transformar a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Os operadores genéticos são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores.

3.1.4.1 Operador de mutação de um ponto

O operador de mutação é responsável pela diversidade genética da população, é excelente para contornar problemas de mínimos locais, pois com esse mecanismo conseguimos alterar a direção da busca. Esse tipo de mutação apenas muda um bit da solução, como descrito na [Figura 6](#).

Figura 6 – Operador de mutação de um ponto

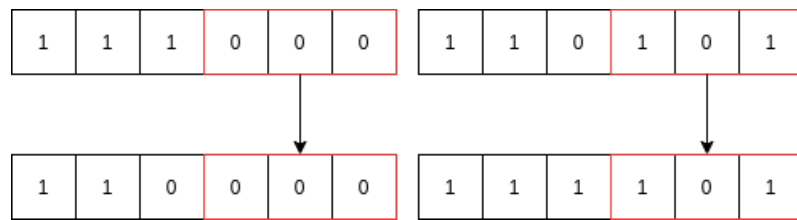


Fonte: Autoria própria

3.1.4.2 Operador de cruzamento de um ponto

O operador de cruzamento é responsável pela recombinação de características dos pais durante a reprodução, para que as próximas gerações herdarem características, portanto, sua taxa de aplicação deve ser maior que do operador de mutação. O operador de cruzamento de um ponto funciona fazendo uma combinação dos pais de forma que parte da solução seja de um pai e a outra parte seja do outro pai como descrito na [Figura 7](#).

Figura 7 – Operador de cruzamento de um ponto



Fonte: Autoria própria

3.2 O OPERADOR *GENERALIZED PARTITION Crossover*

Em geral, os operadores de partição tem como princípio aproveitar as melhores partes de duas soluções, garantindo sempre que a melhor solução gerada seja sempre melhor ou que mantenha o custo das soluções iniciais como descrito na [Seção 2.3](#).

Neste trabalho optou-se pela aplicação do operador de cruzamento de partição generalizado (GPX), pois, para o STP ([WHITLEY; HAINS; HOWE, 2009](#)) o grafo de união construído de grafos proveniente das soluções encontradas utilizando-se algoritmo genético contém várias partições do tipo específico que o cruzamento de partição (PX) está procurando. Desse modo, o cruzamento generalizado de partições é capaz de encontrar e usar todas essas partições em uma única recombinação sem aumentar a complexidade computacional do operador. O intuito do trabalho é criar uma versão adaptada do operador para o STPG, para que funcione de forma similar ao operador inicial.

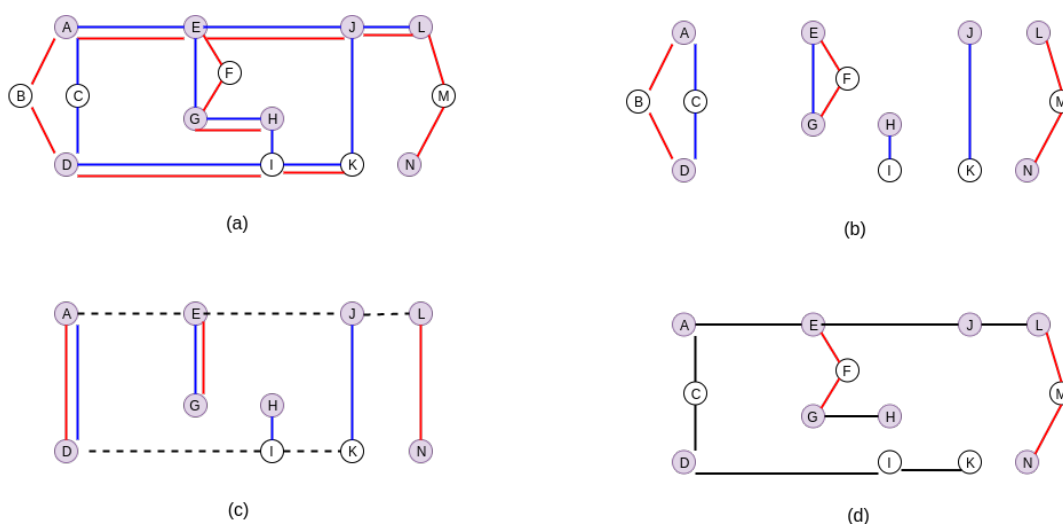
Portanto, na versão adaptada do operador ele trabalha criando um grafo de união $G = (V, E)$ utilizando a solução de dois parentes da mesma forma que o PX. Entretanto, o GPX cria um subgrafo G_u no qual é composto por múltiplos subgrafos desconexos, desta forma é necessário a utilização da busca em largura (BFS) para encontrar cada subgrafo (componente conectado) de G_u , o funcionamento completo do algoritmo está descrito no [Algoritmo 3](#).

Algoritmo 2: GPX para o Problema da Árvore de Steiner**Entrada:** G_1, G_2 **Saída:** Combinação de dois pais.

1. Crie um grafo de união $G_u = G_1 \cup G_2$ através dos grafos G_1 e G_2 (pais).
2. Remova todas as arestas em comum de G_u .
3. Encontre todos os componentes conexos de G_u . Os componentes conectados com mais de um nó são candidatos a recombinação.
4. Para cada componente candidato, crie 2 grafos simplificados, um para cada parente. Os grafos simplificados contêm apenas os vértices de entrada do componente candidato, vértices conectados com pelo menos uma aresta entre vértices localizados em diferentes componentes candidatos. As arestas dos grafos simplificados podem ser obtidas removendo os vértices que não são vértices de entrada (e ligando os vértices vizinhos).
5. Teste cada componente candidato. Se os dois grafos simplificados são iguais, então o componente candidato é um componente recombinante.
6. O grafo considerado também é considerado um componente recombinante.
7. Aplique o operador de cruzamento selecionando a menor solução parcial dentro de cada componente recombinante.

A Figura 8 descreve o processo de execução do Algoritmo 3 e da seleção dos componentes recombinantes, além disso, ilustra o formato de um grafo simplificado.

Figura 8 – Aplicação do Algoritmo 3 com componente não recombinante (a) Soluções: vermelha e azul, (b) Partições encontradas, (c) Grafo Simplificado e (d) Solução Recombinada



Fonte: Autoria Própria

No qual,

- (a) Grafo de União formado pelos pais vermelho e azul;
- (b) Depois de removido as arestas comuns, seis componentes candidatos são encontrados;
- (c) Os grafos simplificados são criados retirando-se os vértices internos, e vértices que não estão conectados a vértices em outros componentes candidatos. As arestas em comum são representadas pelas linhas tracejadas; Os grafos simplificados para dois componentes (formado pelos vértices A e D, e pelos vértices E e G) são respectivamente iguais, portanto, eles são componentes recombinantes. Os outros componentes candidatos não são componentes recombinantes porque seus grafos respectivos não são iguais. Depois de identificar os componentes recombinantes isolados, o resultado do grafo também é considerado um componente recombinante. Portanto, há 3 componentes recombinantes neste exemplo: componente 1 (Formado pelos vértices: A, B, C e D), 2 (Formado pelos vértices E, F, G), e o 3 (Formado pelos vértices H,I, J, K, L, M e N);
- (d) A solução formada pela seleção: azul pelo componente 1; vermelho pelo componente 2; e vermelho pelo componente 3.

4 METODOLOGIA

Neste capítulo será abordado detalhes de como o trabalho foi desenvolvido, tal como foi realizada a validação desses algoritmos e a análise dos resultados obtidos, visando obter a diferença entre os métodos avaliando-os por meio das métricas: custo, tempo e qualidade das soluções.

4.1 TECNOLOGIAS E FERRAMENTAS

Com o intuito de alcançar os objetivos descritos na [Seção 1.3](#) será utilizada a linguagem de programação *Python 3.7* para a implementação total do trabalho, desde a implementação do algoritmo genético, o GPX, outros métodos e outros elementos que serão utilizados obtenção dos dados, visualização, entre outros. A definição da linguagem de programação teve como principal motivo a velocidade e facilidade na implementação de scripts, além da disponibilidade de bibliotecas com recursos matemáticos. Para a execução dos testes será utilizado um processador Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz e 8GB de memória RAM.

4.2 BASE DE DADOS

Para os testes serão utilizados o conjunto B de datasets para a árvore de Steiner em grafos contidos na OR-LIBRARY, no qual são disponibilizados datasets de vários tamanhos como descrito pela [Quadro 2](#) que é mantida por JE Beasley pesquisador de otimização combinatória pela Brunel University – Londres ([BEASLEY, 1990](#)).

Quadro 2 – Base de dados

| Conjunto | Nós | Qt. Terminais | Gráfico e tipo de Peso |
|----------|--------|---------------|--------------------------------|
| B | 50-100 | 9-50 | Esparso e com pesos aleatórios |
| C | 500 | 5-250 | Esparso e com pesos aleatórios |
| D | 1000 | 5-500 | Esparso e com pesos aleatórios |
| E | 2500 | 5-1250 | Esparso e com pesos aleatórios |

Fonte: ([BEASLEY, 1990](#))

Podemos visualizar com mais clareza o conjunto B de datasets por meio do [Apêndice A](#), no qual estão descritos a quantidade de vértices, arestas e terminais e o ótimo global de cada dataset contido no conjunto.

4.3 ABORDAGENS UTILIZADAS

Foram utilizadas diversos tipos de abordagens com o intuito de verificar a eficácia de cada uma. Na [Tabela 1](#) está sumarizado as abordagens e cada heurística utilizada em cada

uma delas.

Tabela 1 – Abordagens implementadas

| Abordagens | Estrutura AG utilizada | População Inicial |
|------------|------------------------------------------------------------------------------|-------------------------------------------------------|
| AG | Cruzamento de um ponto, mutação de um ponto, seleção roleta. | Aleatória |
| GPX | Cruzamento gpx, seleção roleta | Árvores factíveis por meio de heurísticas |
| Híbrido | Cruzamento gpx, cruzamento de um ponto, mutação de um ponto, seleção roleta. | Aleatória + Árvores factíveis por meio de heurísticas |

Fonte: Autoria própria

4.3.1 AG

Na abordagem AG foi utilizado a estrutura do algoritmo genético simples, com população inicial randômica, operadores de cruzamento e mutação simples (1 ponto) e seleção roleta como descrito anteriormente no [Algoritmo 1](#).

4.3.2 GPX

Na abordagem GPX foi utilizado a estrutura do algoritmo genético, entretanto com população inicial conexa gerada pelas heurísticas da [Seção 2.2](#), usando uma probabilidade de 80% de ser gerado pela heurística SPH e 20% de ser gerado pela heurística SPOH, aplicando apenas o operador GPX, e utilizando a seleção roleta como descrito no [Algoritmo 3](#).

Algoritmo 3: GPX

Entrada: $t \leftarrow 0$

Saída: Solução esperada para o problema

início

Gerar população $P(t)$ conexa por meio de heurísticas

Avaliar os indivíduos da população $P(t)$

repita

Selecionar os indivíduos de $P(t)$ para cruzamento criando-se $P(t)'$

Aplicar o operador cruzamento GPX em $P(t)'$

$P(t) = P(t)'$

Avaliar os indivíduos da população $P(t)$

$t = t + 1$

até condição de parada alcançada;

fim

4.3.3 Híbrido

Na abordagem Híbrido foi utilizado um misto das duas primeiras abordagens, a população inicial é gerada tanto de forma randômica quanto de forma heurística usando uma probabilidade de 80% de ser gerado pela heurística SPH e 20% de ser gerado pela heurística SPOH, além disso, todos os operadores foram incluídos (cruzamento e mutação de um ponto para as soluções inactiváveis, GPX para as soluções factíveis). Além disso, foi utilizado a seleção roleta como descrito no [Algoritmo 4](#).

Algoritmo 4: Híbrido

Entrada: $t \leftarrow 0$

Saída: Solução esperada para o problema

início

Gerar população $P(t)$ parte aleatória e parte conexa por meio de heurísticas

Avaliar os indivíduos da população $P(t)$

repita

Selecionar os indivíduos de $P(t)$ para cruzamento criando-se $P(t)'$

if *Conexo* **then**

| Aplicar o operador cruzamento GPX em $P(t)'$

else

| Aplicar o operador de cruzamento de um ponto em $P(t)'$

| Aplicar o operador de mutação de um ponto em $P(t)'$

end

$P(t) = P(t)'$

Avaliar os indivíduos da população $P(t)$

$t = t + 1$

até *condição de parada alcançada*;

fim

5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Com a realização dos testes foi possível obter diversos dados sobre o comportamento e desempenho dos métodos STPG. A seguir serão descritos tabelas e gráficos apresentando informações e discussões sobre os testes, nos quais cada configuração foi executada dez vezes e foi calculada a média e o desvio padrão da amostra em análise, é importante observar que as bases de dados escolhidas estão contidas no conjunto B dos datasets OR-LIBRARY descritas na [Seção 4.2](#) anteriormente.

5.1 PARAMETRIZAÇÃO DOS ALGORITMOS

Para validação dos algoritmos e suas funções serão obtidos dados da execução através das métricas de avaliação: custo, tempo e qualidade das soluções. Visando obter os pontos fortes e fracos de cada algoritmo e sua eficiência. As configurações de entrada dos algoritmos foram:

- **População:** 10 indivíduos;
- **Critério de parada:** Estagnação (Média da população);
- **Total de execuções:** 10;
- **Taxa de crossover:** 80%;
- **Taxa de mutação:** 20%.

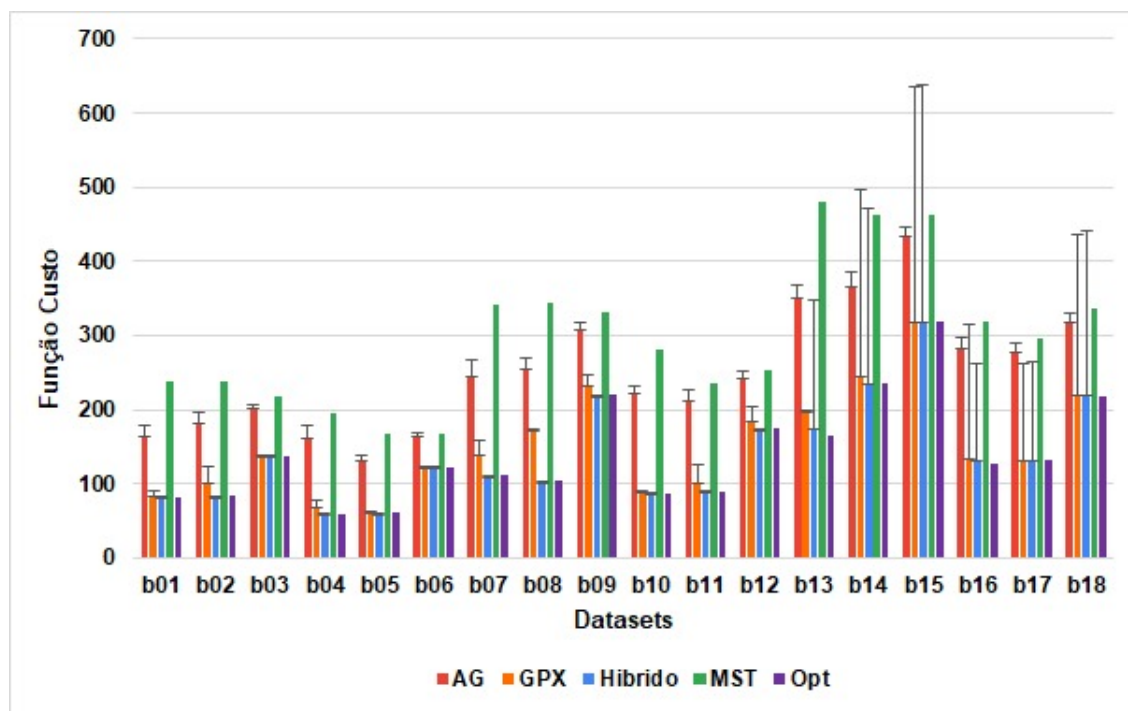
5.2 ANÁLISE GERAL

Conseguimos ter uma visão geral dos resultados obtidos através das Figuras 9 e 10, no qual a primeira descreve o custo obtido por cada abordagem e a segunda descreve o tempo gasto por cada abordagem na média das 10 execuções respectivamente. Em razão da quantidade de dados obtidos os resultados também estão expressos na [Tabela 2](#).

De modo geral, na [Figura 9](#) em 4 dos 18 datasets as abordagens GPX e Híbrido tiveram um desempenho equivalentemente bons, ambos encontraram os ótimos globais do dataset em tempo hábil. Em 4 dos 18 datasets, a abordagem GPX teve um custo menor que as demais abordagens. De mesmo modo, em 7 dos 18 datasets a abordagem Híbrida se destacou das demais abordagens pelo menor custo. Entretanto, nos datasets b09 e b17 a abordagem Híbrida encontrou certa dificuldade em convergir a população completa, portanto, aproximadamente 10% dos indivíduos na população final não eram factíveis.

A abordagem AG não conseguiu se destacar em nenhum dos datasets, isto acontece pois o algoritmo genético acaba ficando preso em ótimos locais, enquanto as outras abordagens utilizam sementes na sua população inicial que garantem a convergência da solução para o ótimo global. Apesar disso, pode-se verificar que em todos os casos a população convergiu

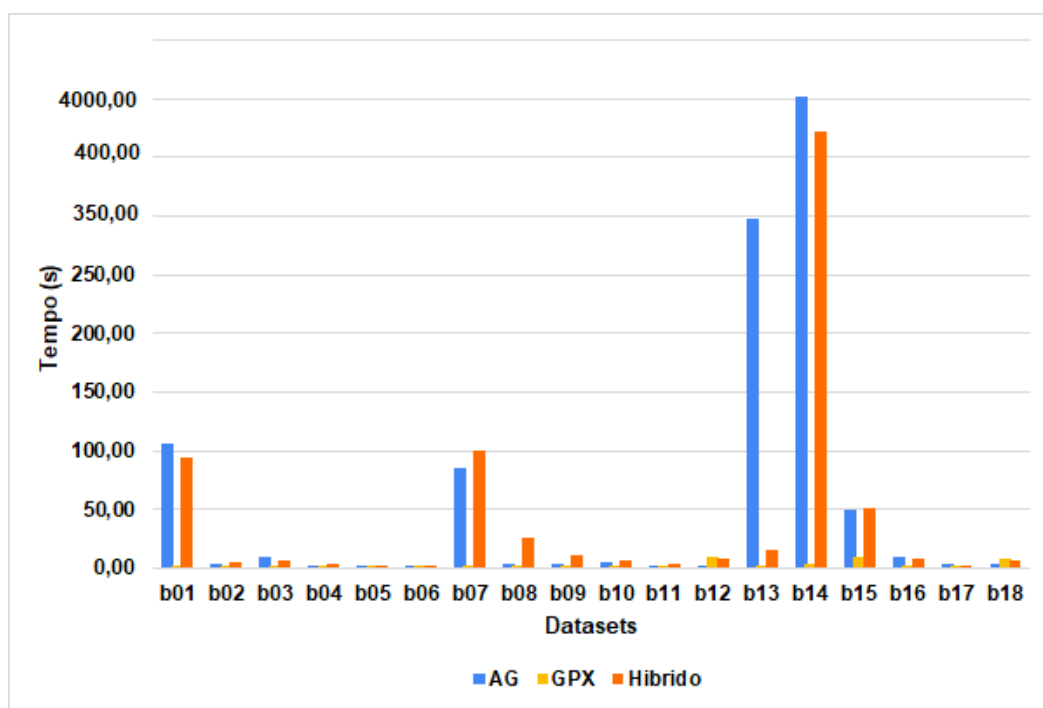
Figura 9 – Função custo médio para cada abordagem



Fonte: Autoria própria

completamente para a solução encontrada mantendo a população conexa. Além disso, sempre se manteve abaixo da árvore mínima do grafo completo (MST).

Figura 10 – Tempo médio para cada abordagem



Fonte: Autoria própria

Na [Figura 10](#), em termos de tempo, observa-se que a abordagem Híbrido em 61,11% dos casos tem o seu tempo de processamento maior que as outras abordagens, salvo pelo dataset b14 no qual é perceptível que o algoritmo genético acabou caindo nos piores casos. Entretanto, é possível verificar que a abordagem Híbrida apresenta média de apenas 67,55% de desvios dos custos das soluções, obtendo o ótimo global em 58,33% das vezes e em 44,00% das vezes obtendo ótimos locais bem próximos ao objetivo global.

Contudo, a melhor abordagem acaba sendo o GPX pois além de ter o tempo menor em 83,33% dos casos quando comparado ao Híbrido, apresenta média de apenas 67,66% nos desvios dos custos das soluções, obtendo o ótimo global em 41,11% das vezes e em 55,00% das vezes obtendo ótimos locais bem próximos ao objetivo global e que realizaram a convergência da população completa.

Tabela 2 – Resultados obtidos para os datasets do conjunto B

| Instância | Opt | MST | AG | | | GPX | | | Híbrido | | |
|--------------|-----|-----|--------------|----------------|-------------------|--------------|----------------|--------------|--------------|----------------|-------------------|
| | | | Acertos Opt. | Custo | Tempo (s) | Acertos Opt. | Custo | Tempo (s) | Acertos Opt. | Custo | Tempo (s) |
| b01 | 82 | 238 | 0,00 % | 164,30 ± 14,87 | 105,36 ± 98,26 | 80,00 % | 84,10 ± 6,29 | 0,09 ± 0,03 | 100,00 % | 82,00 ± 0,00 | 93,91 ± 52,61 |
| b02 | 83 | 238 | 0,00 % | 182,70 ± 16,87 | 2,23 ± 3,97 | 40,00 % | 101,00 ± 21,57 | 0,15 ± 0,02 | 100,00 % | 82,90 ± 0,31 | 4,23 ± 5,14 |
| b03 | 138 | 217 | 0,00 % | 202,10 ± 3,60 | 9,54 ± 10,02 | 100,00 % | 138,00 ± 0,00 | 0,67 ± 0,12 | 0,00 % | 138,00 ± 0,00 | 6,57 ± 6,89 |
| b04 | 59 | 196 | 0,00 % | 162,90 ± 16,48 | 1,06 ± 0,77 | 0,00 % | 68,60 ± 8,81 | 0,11 ± 0,05 | 70,00 % | 59,90 ± 1,45 | 3,16 ± 3,23 |
| b05 | 61 | 167 | 0,00 % | 132,40 ± 5,64 | 0,42 ± 0,42 | 90,00 % | 61,10 ± 0,32 | 0,34 ± 0,15 | 100,00 % | 61,00 ± 0,00 | 1,85 ± 1,44 |
| b06 | 122 | 168 | 0,00 % | 165,10 ± 3,76 | 0,53 ± 0,34 | 0,00 % | 124,00 ± 0,00 | 0,76 ± 0,17 | 0,00 % | 124,00 ± 0,00 | 1,75 ± 1,59 |
| b07 | 111 | 341 | 0,00 % | 245,20 ± 21,79 | 84,96 ± 89,86 | 30,00 % | 138,70 ± 19,14 | 0,15 ± 0,03 | 100,00 % | 111,00 ± 0,00 | 99,83 ± 104,78 |
| b08 | 104 | 343 | 0,00 % | 255,80 ± 14,50 | 3,34 ± 4,30 | 0,00 % | 174,00 ± 0,00 | 0,40 ± 0,03 | 100,00 % | 104,00 ± 0,00 | 26,24 ± 34,81 |
| b09 | 220 | 331 | 0,00 % | 308,00 ± 9,82 | 4,08 ± 3,60 | 60,00 % | 231,60 ± 14,98 | 2,36 ± 0,40 | 100,00 % | 220,00 ± 0,00 | 10,23 ± 9,08 |
| b10 | 86 | 282 | 0,00 % | 221,90 ± 10,40 | 4,72 ± 3,89 | 70,00 % | 90,80 ± 11,83 | 0,41 ± 0,23 | 70,00 % | 86,70 ± 1,25 | 6,62 ± 6,10 |
| b11 | 88 | 236 | 0,00 % | 212,90 ± 14,22 | 0,78 ± 0,65 | 10,00 % | 101,20 ± 25,27 | 0,90 ± 0,41 | 0,00 % | 90,00 ± 0,00 | 2,72 ± 2,97 |
| b12 | 174 | 253 | 0,00 % | 243,10 ± 8,20 | 1,05 ± 0,74 | 60,00 % | 184,40 ± 20,21 | 10,05 ± 8,37 | 100,00 % | 174,00 ± 0,00 | 7,53 ± 5,35 |
| b13 | 165 | 480 | 0,00 % | 350,30 ± 17,40 | 319,02 ± 345,35 | 0,00 % | 198,00 ± 20,67 | 1,05 ± 0,37 | 0,00 % | 174,00 ± 0,00 | 282,16 ± 188,73 |
| b14 | 235 | 463 | 0,00 % | 365,90 ± 20,93 | 4010,85 ± 6465,71 | 0,00 % | 251,56 ± 16,45 | 3,91 ± 4,76 | 0,00 % | 236,00 ± 0,00 | 1961,52 ± 3768,15 |
| b15 | 318 | 462 | 0,00 % | 434,50 ± 10,56 | 49,33 ± 41,24 | 80,00 % | 318,44 ± 0,88 | 8,72 ± 1,18 | 10,00 % | 318,90 ± 0,31 | 51,42 ± 46,76 |
| b16 | 127 | 319 | 0,00 % | 283,20 ± 13,10 | 9,18 ± 12,46 | 20,00 % | 165,44 ± 27,18 | 0,74 ± 0,38 | 100,00 % | 131,00 ± 0,00 | 12,34 ± 14,21 |
| b17 | 131 | 296 | 0,00 % | 278,22 ± 11,40 | 2,97 ± 4,07 | 70,00 % | 131,44 ± 0,72 | 4,81 ± 6,27 | 100,00 % | 132,00 ± 0,00 | 9,94 ± 8,19 |
| b18 | 218 | 337 | 0,00 % | 318,00 ± 12,79 | 2,95 ± 2,48 | 30,00 % | 218,78 ± 0,67 | 10,73 ± 3,49 | 0,00 % | 220,60 ± 0,51 | 6,72 ± 3,57 |
| Média | - | - | 0,00 % | 251,47 ± 81,33 | 257,96 ± 940,32 | 41,11 % | 152,42 ± 69,16 | 2,22 ± 3,19 | 58,33 % | 141,42 ± 70,08 | 42,10 ± 97,05 |

Fonte: Autoria própria

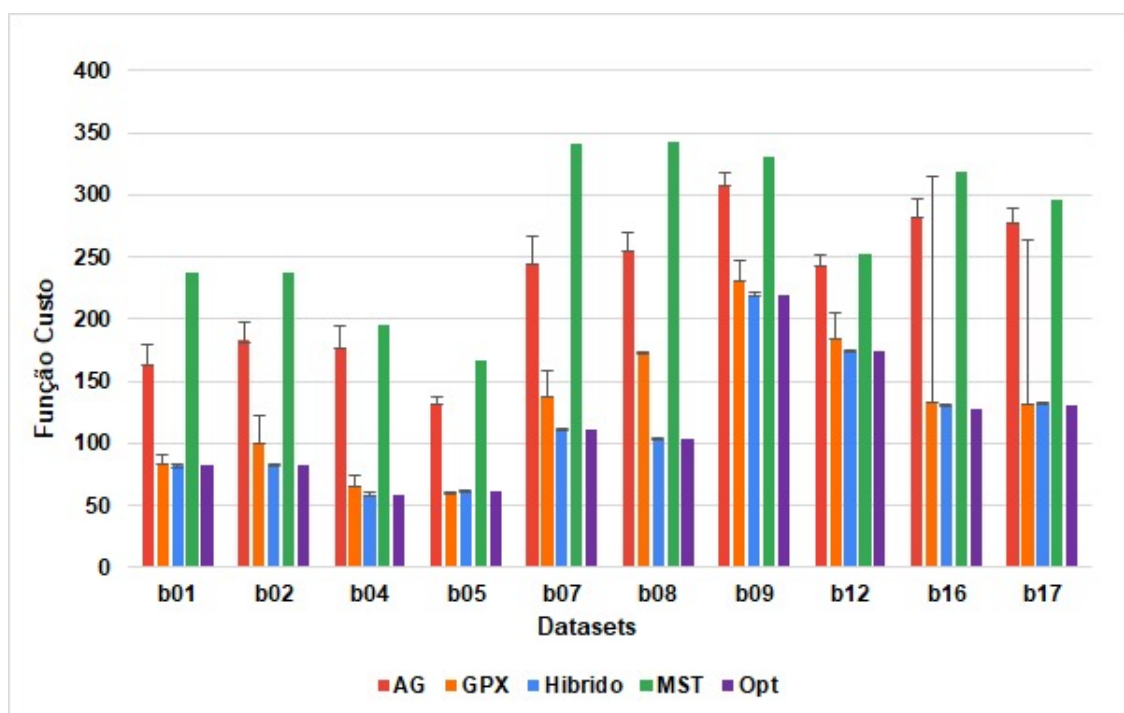
5.2.1 Datasets b01, b02, b04, b05, b07, b08, b09, b12, b16 e b17

Nos datasets em questão a abordagem Híbrido se saiu melhor do que as outras abordagens em quesito custo pois obteve em todas as execuções 100,00% de acerto do ótimo global na maioria dos datasets, salvo pelo dataset b04 no qual obteve 70,00% de acertos. Isso se deve a parte da população ser aleatória, pois garantiu a exploração de outras regiões do fitness e garantiu a melhoria das soluções geradas pelas heurísticas inicialmente.

Isso garantiu que em todos os datasets em questão a abordagem Híbrido se aproximasse muito ou encontrasse o ótimo global do dataset, nestes casos, o algoritmo se saiu melhor do que a abordagem GPX em termos de custo da solução, entretanto, o tempo utilizado para o processamento foi muito maior.

Além disso, a abordagem Híbrido se mostrou muito precisa pois o desvio padrão encontrado para abordagem foi muito pequeno, chegando a desvios nulos em alguns casos, apesar de que em alguns raros casos a população não convergiu completamente para o ótimo global.

Figura 11 – Função custo médio melhor para a abordagem Híbrido



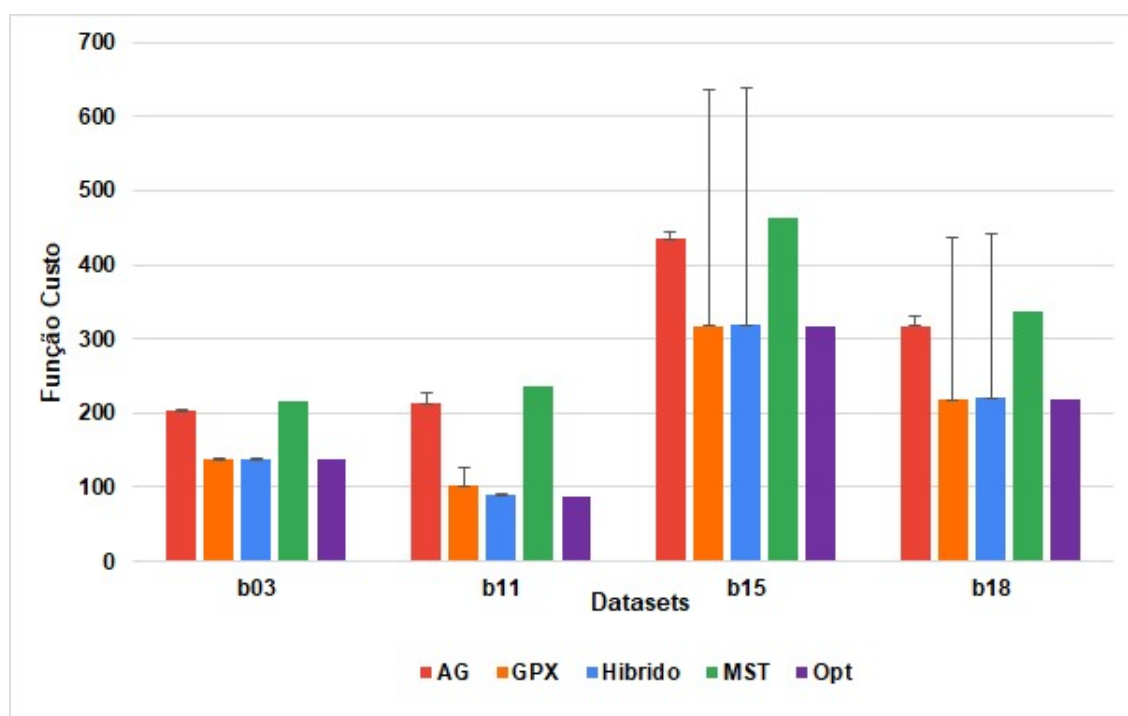
Fonte: Autoria Própria

5.2.2 Datasets b03, b11, b15 e b18

Nos datasets em questão a abordagem GPX se saiu melhor em todos os casos, apesar da diferença ser pequena em questão dos custos das soluções em comparação com a abordagem Híbrido como descrito na [Figura 12](#), a abordagem também se saiu melhor no quesito tempo.

Além disso, o algoritmo se mostrou muito preciso pois o desvio padrão encontrado para a abordagem foi muito pequeno, mostrando que toda a população convergia para o ótimo global do dataset.

Figura 12 – Função custo médio melhor para a abordagem GPX

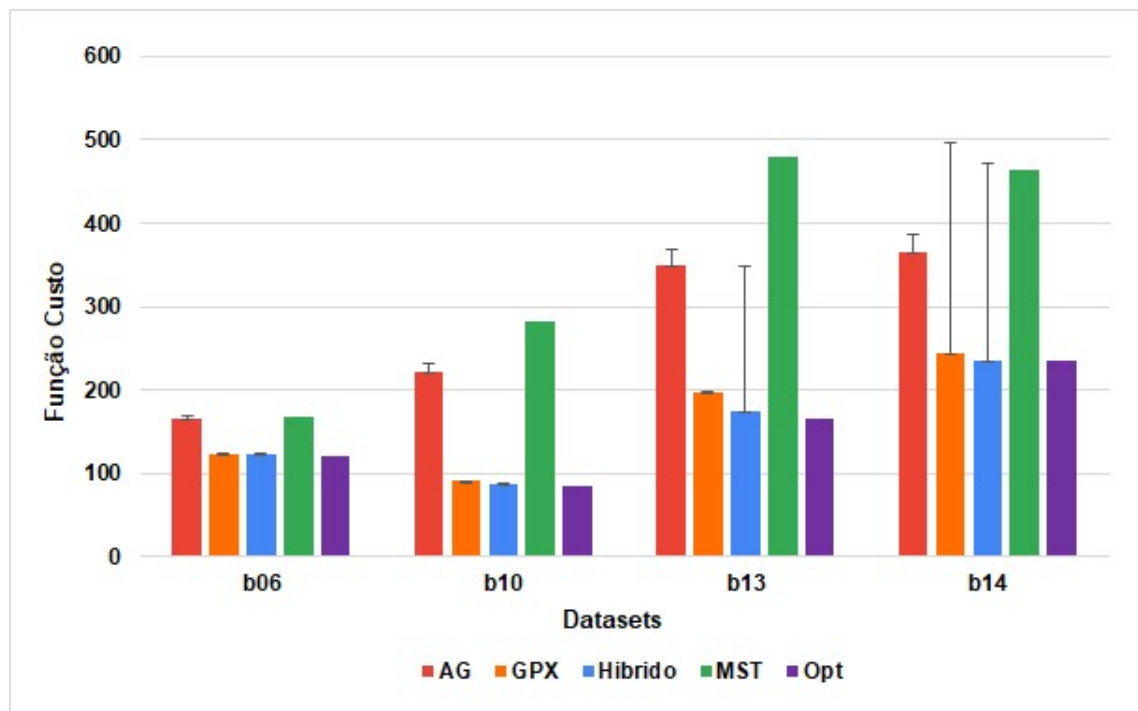


Fonte: Autoria Própria

5.2.3 Datasets b06, b10, b13 e b14

Nos datasets em questão ambas abordagens GPX e Híbrido tiveram problemas em alcançar o ótimo global como descrito na [Figura 13](#), com exceção do dataset b10 no qual ambas abordagens se aproximaram muito do ótimo global, entretanto, não o alcançaram. Em todas as vezes o GPX se saiu superior no quesito tempo como podemos observar pela [Tabela 2](#).

Figura 13 – Datasets semelhantes



Fonte: Autoria Própria

6 CONSIDERAÇÕES FINAIS

A princípio buscava-se comparar a eficácia de métodos para a resolução do problema da árvore de Steiner, posteriormente o enfoque do trabalho passou a ser a adaptação do operador GPX devido a sua versatilidade e potencialidade de desenvolvimento.

As etapas de desenvolvimento do trabalho foram desde a modelagem, representação e leitura das soluções; a proposta do método de cálculo da aptidão de cada indivíduo da população mais adequada; o estudo de métodos possíveis para a implementação do STPG, até o desenvolvimento de uma adaptação do operador de cruzamento GPX.

Embora diversos problemas tenham sido enfrentados durante o desenvolvimento, como encontrar meios e abordagens que pudessem garantir que as soluções geradas fossem factíveis (conexas), além da necessidade de utilização de dois tipos de representações das soluções, o objetivo geral do trabalho foi atingido.

Os dados e as análises mostraram a melhoria que o GPX traz para as soluções do STPG. Além disso, mostra-se que o modelo é funcional e que por meio do aprofundamento das pesquisas o algoritmo poderá se tornar competitivo em comparação a abordagens já consolidadas. Alguns pontos podem ser melhorados, como por exemplo a necessidade de as entradas do operador serem árvores já factíveis, dependendo então de outros métodos para a geração das árvores.

Nesta pesquisa, todas as abordagens se mostraram funcionais e a escolha do uso de cada uma depende da aplicação. Para o conjunto de datasets considerado, na maioria dos casos o GPX obteve resultados gerais superiores aos demais em quesitos de tempo, qualidade da solução e alcance dos ótimos globais de cada dataset. Desta forma, o operador consegue desenvolver e otimizar as soluções com melhor aproveitamento, além de não gerar soluções infactíveis.

REFERÊNCIAS

- BEASLEY, J. E. **OR-LIBRARY**. 1990. Disponível em: <<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>>. Citado 2 vezes nas páginas 9 e 16.
- BOUCHACHIA, A.; PROSSEGGER, M. A hybrid ensemble approach for the steiner tree problem in large graphs: A geographical application. **Applied Soft Computing**, Elsevier BV, v. 11, n. 8, p. 5745–5754, dez. 2011. Disponível em: <<https://doi.org/10.1016/j.asoc.2011.03.005>>. Citado na página 9.
- CHENG, X. Z.; DU, D.-Z. (Ed.). **Steiner Trees in Industry**. Springer US, 2002. Disponível em: <<https://doi.org/10.1007/978-1-4613-0255-1>>. Citado na página 1.
- CIESLIK, D. **Steiner minimal trees**. Dordrecht Boston: Kluwer Academic, 1998. ISBN 0792349830. Citado na página 5.
- COSTA, C. R.; ROSSET, V.; NASCIMENTO, M. C. V. A GRASP for the steiner tree problem in graphs to support multicast routing. **IEEE Latin America Transactions**, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 12, p. 3873–3878, dez. 2015. Disponível em: <<https://doi.org/10.1109/tla.2015.7404921>>. Citado na página 9.
- DREYFUS, S. E.; WAGNER, R. A. The steiner problem in graphs. **Networks**, Wiley, v. 1, n. 3, p. 195–207, 1971. Disponível em: <<https://doi.org/10.1002/net.3230010302>>. Citado na página 8.
- FOULDS, L. R.; RAYWARD-SMITH, V. J. STEINER PROBLEMS IN GRAPHS: ALGORITHMS AND APPLICATIONS. **Engineering Optimization**, Informa UK Limited, v. 7, n. 1, p. 7–16, jan. 1983. Disponível em: <<https://doi.org/10.1080/03052158308960625>>. Citado na página 1.
- GILBERT, E. N.; POLLAK, H. O. Steiner minimal trees. **SIAM Journal on Applied Mathematics**, Society for Industrial and Applied Mathematics, v. 16, n. 1, p. 1–29, 1968. ISSN 00361399. Disponível em: <<http://www.jstor.org/stable/2099400>>. Citado na página 1.
- HANAN, M. On steiner's problem with rectilinear distance. **SIAM Journal on Applied Mathematics**, Society for Industrial & Applied Mathematics (SIAM), v. 14, n. 2, p. 255–265, mar. 1966. Disponível em: <<https://doi.org/10.1137/0114025>>. Citado na página 1.
- HWANG, F. **The Steiner tree problem**. Amsterdam New York: North-Holland, 1992. ISBN 9780080867939. Citado 2 vezes nas páginas 6 e 7.
- HWANG, F. K. On steiner minimal trees with rectilinear distance. **SIAM Journal on Applied Mathematics**, Society for Industrial & Applied Mathematics (SIAM), v. 30, n. 1, p. 104–114, jan. 1976. Disponível em: <<https://doi.org/10.1137/0130013>>. Citado na página 1.
- KAPSALIS, A.; RAYWARD-SMITH, V. J.; SMITH, G. D. Solving the graphical steiner tree problem using genetic algorithms. **The Journal of the Operational Research Society**, JSTOR, v. 44, n. 4, p. 397, abr. 1993. Disponível em: <<https://doi.org/10.2307/2584417>>. Citado na página 9.
- LINDEN, R. **Algoritmos Genéticos**. Rio de Janeiro: Ciência Moderna, 2012. ISBN 8539901951. Citado na página 3.

- MA, X.; LIU, Q. A particle swarm optimization for steiner tree problem. In: **2010 Sixth International Conference on Natural Computation**. IEEE, 2010. Disponível em: <<https://doi.org/10.1109/icnc.2010.5583217>>. Citado na página 9.
- NORVIG, P. **Inteligência Artificial**. New Jersey: Elsevier, 2013. ISBN 8535237011. Citado na página 5.
- OLSON, M. **Modelo mental: paisagens de fitness**. 2013. Disponível em: <<https://futureblind.com/2013/05/08/mental-model-fitness-landscapes/>>. Citado na página 5.
- PRÖMEL, H. J.; STEGER, A. **The Steiner Tree Problem**. ViewegTeubner Verlag, 2002. Disponível em: <<https://doi.org/10.1007/978-3-322-80291-0>>. Citado na página 1.
- REZENDE, S. **Sistemas inteligentes : fundamentos e aplicacoes**. Barueri: Monole, 2003. ISBN 85-204-1683-7. Citado na página 3.
- WHITLEY, D.; HAINS, D.; HOWE, A. Tunneling between optima. In: **Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO 09**. ACM Press, 2009. Disponível em: <<https://doi.org/10.1145/1569901.1570026>>. Citado 5 vezes nas páginas 1, 2, 7, 8 e 13.
- WHITLEY, D.; HAINS, D.; HOWE, A. A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In: **Parallel Problem Solving from Nature, PPSN XI**. Springer Berlin Heidelberg, 2010. p. 566–575. Disponível em: <https://doi.org/10.1007/978-3-642-15844-5_57>. Citado na página 2.

Apêndices

APÊNDICE A – Datasets do Conjunto B detalhados

| Dataset | B01 | B02 | B03 | B04 | B05 | B06 | B07 | B08 | B09 | B10 | B11 | B12 | B13 | B14 | B15 | B16 | B17 | B18 |
|---------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Vértices | 50 | 50 | 50 | 50 | 50 | 50 | 75 | 75 | 75 | 75 | 75 | 75 | 100 | 100 | 100 | 100 | 100 | 100 |
| Arestas | 63 | 63 | 63 | 100 | 100 | 100 | 94 | 94 | 94 | 150 | 150 | 150 | 125 | 125 | 125 | 200 | 200 | 200 |
| Terminais | 9 | 13 | 25 | 9 | 13 | 25 | 13 | 19 | 38 | 13 | 19 | 38 | 17 | 25 | 50 | 17 | 25 | 50 |
| Ótimo global | 82 | 83 | 138 | 59 | 61 | 122 | 111 | 104 | 220 | 86 | 88 | 174 | 165 | 235 | 318 | 127 | 131 | 218 |