



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

PAULO HENRIQUE SOUSA DE ARAUJO

**UMA ABORDAGEM HÍBRIDA EXATA-HEURÍSTICA APLICADA AO PROBLEMA
DO CICLO MEDIANO SEM RESTRIÇÕES DE CAPACIDADE**

CRATEÚS

2019

PAULO HENRIQUE SOUSA DE ARAUJO

UMA ABORDAGEM HÍBRIDA EXATA-HEURÍSTICA APLICADA AO PROBLEMA DO
CICLO MEDIANO SEM RESTRIÇÕES DE CAPACIDADE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Crateús da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientadora: Prof^a. Ma. Lisieux Marie
Marinho dos Santos Andrade

Coorientador: Prof. Me. Luiz Alberto
do Carmo Viana

CRATEÚS

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- A69a Araujo, Paulo Henrique Sousa de.
Uma abordagem híbrida exata-heurística aplicada ao problema do ciclo mediano sem restrições de capacidade / Paulo Henrique Sousa de Araujo. – 2019.
102 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Crateús, Curso de Ciência da Computação, Crateús, 2019.
Orientação: Profa. Ma. Lisieux Marie Marinho dos Santos Andrade.
Coorientação: Prof. Me. Luiz Alberto do Carmo Viana.
1. Problema do Ciclo Mediado sem Restrições de Capacidade. 2. Abordagem Híbrida. 3. Algoritmo Quasi-Exato. I. Título.

CDD 004

PAULO HENRIQUE SOUSA DE ARAUJO

UMA ABORDAGEM HÍBRIDA EXATA-HEURÍSTICA APLICADA AO PROBLEMA DO
CICLO MEDIANO SEM RESTRIÇÕES DE CAPACIDADE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Crateús da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em: 13 de Dezembro de 2019

BANCA EXAMINADORA

Prof^a. Ma. Lisieux Marie Marinho dos Santos
Andrade (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Me. Luiz Alberto do Carmo
Viana (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Renê Pereira Gusmão
Universidade Federal de Sergipe (UFSE)

Prof. Dr. Nicolas de Almeida Martins
Universidade da Integração Internacional da
Lusofonia Afro-Brasileira (Unilab)

Prof. Me. Mardson da Silva Ferreira
Universidade Federal do Ceará (UFC)

À minha mãe, minha sustentação.

AGRADECIMENTOS

A Deus, por ter sido fonte de resiliência em muitos momentos, apesar da relação conturbada.

À minha mãe, por ser minha base de sustentação em todos os sentidos e durante todo o processo.

À Prof^a. Ma. Lisieux Marie Marinho dos Santos Andrade, por ter aceito o desafio que foi me orientar. Nela encontrei a melhor mistura que poderia existir entre inteligência e doçura.

Ao Prof. Me. Luiz Alberto do Carmo Viana, por ter aceitado me coorientar. A mente mais inteligente que já conheci.

Ao Prof. Dr. Rennan Ferreira Dantas, coordenador do curso de Ciência da Computação no Campus Crateús. O melhor coordenador que este curso poderia ter.

Aos amigos que estão e estiveram comigo desde antes da graduação, e aos que fiz durante a mesma, que nunca duvidaram do que eu poderia alcançar e que tornaram essa experiência bem mais agradável do que poderia ter sido.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu mais sincero agradecimento.

“Death is so terribly final, while life is full of possibilities.”

(George R. R. Martin)

RESUMO

As telecomunicações são intrínsecas ao cotidiano da maioria das pessoas, sua importância e complexidade geram custos que empresas fornecedoras desse tipo de serviço visam sempre minimizar. Uma maneira de reduzir estes custos é definindo melhor a organização das conexões que constituem a rede pela qual os dados do serviço vão passar. O Problema do Ciclo Mediano sem Restrições de Capacidade (PCMRC) transforma em um problema computacional a necessidade de otimizar a implantação de uma topologia anel-estrela em uma rede. Para alcançar o propósito desta otimização, é necessário minimizar os custos para construir sua estrutura central (ciclo) e para atribuir externamente os demais elementos que não a compõem. Desta forma, o presente trabalho propõe três abordagens híbridas aplicadas ao PCMRC, algoritmos *quasi-exatos* compostos pelos procedimentos *Branch-and-Bound* (B&B), *Greedy Randomized Adaptive Search Procedure* (GRASP), *Iterated Local Search* (ILS) e *Generic Variable Neighborhood Search* (GVNS). O objetivo é encontrar uma região de soluções promissoras, usando o B&B de forma parcial, para a realização de uma busca suave pelas heurísticas, duas composições entre os procedimentos citados, GRASP-ILS e GGVNS. Experimentos, executados em 360 casos de teste para cada abordagem, apresentaram resultados relevantes, sobretudo quando comparados aos resultados exatos produzidos pelo B&B.

Palavras-chave: Problema do Ciclo Mediado sem Restrições de Capacidade. Abordagem Híbrida. Algoritmo *Quasi-Exato*.

ABSTRACT

The telecommunication area plays an important role in everyday life for most people, its importance and complexity produce outlays that providers for this kind of service always try to avoid. One way to reduce these costs is to better set the arrangements of the links that integrate the installed network whose service data will flow through. The Ring Star Problem (RSP) converts in a computational problem the need for optimizing the installation of a ring-star topology into a network, and make it by minimizing the costs of building its central structure (ring) and assigning the remaining elements to this structure. Therefore, this paper brings three hybrid approaches to RSP, quasi-exact algorithms constituted by Branch-and-Bound (B&B), Greedy Randomized Adaptative Search Procedure (GRASP), Iterated Local Search (ILS) and Generic Variable Neighborhood Search (GVNS) procedures. The aim is to find a region with good initial solutions, using B&B partially, so the heuristics, two compositions called GRASP-ILS and GGVNS, can accomplish a smoother search. Computational experiments, performed on 360 test cases for each hybrid approach, revealed expressive results, especially when compared to the exact results produced by the B&B.

Keywords: The Ring Star Problem. Hybrid Approach. Quasi-Exact Algorithm

LISTA DE FIGURAS

| | |
|---|-----|
| Figura 1 – Um grafo qualquer | 18 |
| Figura 2 – Grafo simples | 19 |
| Figura 3 – Grafo completo | 20 |
| Figura 4 – Subgrafo induzido do grafo da Figura 2 | 20 |
| Figura 5 – Grafo conexo | 21 |
| Figura 6 – Ciclo derivado da Figura 3 | 22 |
| Figura 7 – Digrafo simples | 23 |
| Figura 8 – Gráfico de uma FO qualquer | 27 |
| Figura 9 – Um exemplo de solução para PCMRC | 32 |
| Figura 10 – Os dois níveis da reformulação do problema | 43 |
| Figura 11 – Exemplo de <i>crossover</i> entre cromossomos | 46 |
| Figura 12 – Exemplo de solução estrela aceita por PCMRC métrico | 48 |
| Figura 13 – Exemplos de caminhos P e Q em um ciclo | 50 |
| Figura 14 – Desempenho dos procedimentos GRASP-ILS e Híbrido para os 360 casos de testes | 74 |
| Figura 15 – Desempenho dos procedimentos GGVNS e Híbrido para os 360 casos de testes | 82 |
| Figura 16 – Desempenho dos procedimentos GGVNS e Híbrido <i>Multistart</i> para os 360 casos de testes | 84 |
| Figura 17 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 95 |
| Figura 18 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 96 |
| Figura 19 – Comportamento das Funções Objetivos obtidas para instâncias de 100 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 97 |
| Figura 20 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 98 |
| Figura 21 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 99 |
| Figura 22 – Comportamento das Funções Objetivos obtidas para instâncias de 100 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 100 |

| | |
|---|-----|
| Figura 23 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices | |
| (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 101 |
| Figura 24 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices | |
| (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 102 |
| Figura 25 – Comportamento das Funções Objetivos obtidas para instâncias de 100 vértices | |
| (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$ | 103 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Parâmetros estabelecidos para o procedimento Heurístico GRASP-ILS | 71 |
| Tabela 2 – Resultados aproximativos das execuções exatas | 73 |
| Tabela 3 – Instâncias não executadas nos experimentos com o B&B | 73 |
| Tabela 4 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 50 nós. | 75 |
| Tabela 5 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 75 nós. | 76 |
| Tabela 6 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 100 nós. | 77 |
| Tabela 7 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 50 nós. | 79 |
| Tabela 8 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 75 nós. | 80 |
| Tabela 9 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 100 nós. | 81 |
| Tabela 10 – Desempenho dos métodos GGVNS e a proposta Híbrida com <i>Multistart</i> para instâncias de 50 nós. | 83 |
| Tabela 11 – Desempenho dos métodos GGVNS e a proposta Híbrida com <i>Multistart</i> para instâncias de 75 nós. | 85 |
| Tabela 12 – Desempenho dos métodos GGVNS e a proposta Híbrida com <i>Multistart</i> para instâncias de 100 nós. | 86 |
| Tabela 13 – Desempenho do método exato para instâncias de 10 e 20 nós. | 91 |
| Tabela 14 – Desempenho do método exato para instâncias de 30 e 40 nós. | 92 |
| Tabela 15 – Desempenho do método exato para instâncias de 50 e 75 nós. | 93 |
| Tabela 16 – Desempenho do método exato para instâncias de 100 e 125 nós. | 94 |

LISTA DE ALGORITMOS

| | |
|--|----|
| Algoritmo 1 – Pseudocódigo GRASP-ILS. | 65 |
| Algoritmo 2 – Pseudocódigo GVNS. | 66 |
| Algoritmo 3 – Pseudocódigo GGVNS. | 67 |
| Algoritmo 4 – Hibridização <i>Quasi-exata</i> com GRASP-ILS | 68 |
| Algoritmo 5 – Hibridização <i>Quasi-exata</i> com GGVNS | 68 |
| Algoritmo 6 – Hibridização <i>Quasi-exata Multistart</i> com GGVNS | 69 |

SUMÁRIO

| | | |
|----------------|--|----|
| 1 | INTRODUÇÃO | 16 |
| 1.1 | Objetivo Geral | 17 |
| 1.1.1 | <i>Objetivos Específicos</i> | 17 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 18 |
| 2.1 | Teoria dos Grafos | 18 |
| 2.1.1 | <i>Grafos</i> | 18 |
| 2.1.2 | <i>Digrafos</i> | 22 |
| 2.2 | Otimização Combinatória | 24 |
| 2.3 | Métodos Heurísticos | 25 |
| 2.4 | Métodos Exatos | 29 |
| 3 | O PROBLEMA DO CICLO MEDIANO SEM RESTRIÇÕES DE CAPACIDADE | 31 |
| 3.1 | Caracterização | 31 |
| 3.2 | Modelo | 33 |
| 4 | TRABALHOS RELACIONADOS | 36 |
| 4.1 | <i>The Ring Star Problem: Polyhedral analysis and exact algorithm</i> | 36 |
| 4.1.1 | <i>Algoritmo Branch-and-Cut</i> | 37 |
| 4.1.1.1 | <i>Separações de Restrições</i> | 38 |
| 4.1.2 | <i>Resultados</i> | 40 |
| 4.2 | Algoritmos Heurísticos e Meta-heurísticas Híbridas Aplicadas ao Planejamento de uma Rede de Telecomunicações com Topologia Anel-Estrela | 41 |
| 4.2.1 | <i>Resultados</i> | 42 |
| 4.3 | <i>An Efficient Evolutionary Algorithm for the Ring Star Problem</i> | 42 |
| 4.3.1 | <i>BBEA: Bilevel Based Evolutionary Algorithm</i> | 45 |
| 4.3.2 | <i>Resultados</i> | 47 |
| 4.4 | <i>Algorithms for the Ring Star Problem</i> | 47 |
| 4.4.1 | <i>Aproximações</i> | 49 |
| 4.4.2 | <i>Heurísticas</i> | 56 |
| 4.4.3 | <i>Resultados</i> | 57 |
| 4.5 | Síntese dos Trabalhos Correlatos | 58 |

| | | |
|-------|--|-----|
| 5 | QUASI-EXATO | 59 |
| 5.1 | Branch-and-Bound | 59 |
| 5.1.1 | Separação das Restrições (3.4) | 61 |
| 5.2 | GRASP-ILS | 64 |
| 5.3 | GGVNS | 65 |
| 5.4 | Híbrido | 67 |
| 5.4.1 | Composições Quasi-exatas | 67 |
| 5.4.2 | Composição Quasi-exata com Multistart | 68 |
| 6 | RESULTADOS COMPUTACIONAIS E ATIVIDADES FUTURAS . . . | 70 |
| 6.1 | Cenário | 70 |
| 6.2 | Resultados - Branch-and-Bound | 72 |
| 6.3 | Resultados Comparativos | 72 |
| 6.3.1 | Quasi-exato com GRASP-ILS e GRASP-ILS | 72 |
| 6.3.2 | Quasi-exato com GGVNS e GGVNS | 78 |
| 6.3.3 | Quasi-exato Multistart com GGVNS e GGVNS | 82 |
| 7 | CONCLUSÃO | 87 |
| | REFERÊNCIAS | 88 |
| | APÊNDICES | 91 |
| | APÊNDICE A – Resultados dos Experimentos Exatos | 91 |
| | APÊNDICE B – Comportamento dos resultados envolvendo GRASP-ILS simplex | 95 |
| | APÊNDICE C – Comportamento dos resultados envolvendo GGVNS simplex | 98 |
| | APÊNDICE D – Comportamento dos resultados envolvendo GGVNS e Multistart | 101 |

1 INTRODUÇÃO

O mundo moderno tem na comunicação virtual um dos seus pilares de sustentação. A cada ano torna-se mais difícil imaginar uma realidade sem o fluxo de informações que percorre as redes de telecomunicações conectando pessoas, desde pesquisadores, que compartilham informações para otimizar seus trabalhos, até familiares que tem nessa forma de comunicação um meio de aplacar a distância de entes queridos. Figurativamente as pessoas conectam-se, mas na realidade os dispositivos que elas usam é que proporcionam a distribuição desse fluxo de dados, que em muitos casos já não necessitam da intervenção humana direta.

Segundo dados do World Bank (2017), existem mais de 3 bilhões de usuários de internet ao redor do mundo e há mais linhas de celulares que habitantes no planeta Terra, tendo 104,5 linhas para cada 100 pessoas. Já o uso de telefones fixos vem caindo, mas ainda é significativo com 13,2 linhas a cada 100 pessoas. Com esse tamanho, o impacto causado pelas redes de telecomunicação passa por vários setores da sociedade, como bem explica Bertschek *et al.* (2015). O impacto econômico é geralmente a força-motriz que leva à inovação, mas as melhorias na vida das pessoas fazem das telecomunicações esse pilar.

Jorgenson e Vu (2016, p. 384) mostram a evolução e aumento no uso de internet e linhas telefônicas para celular a partir de 1990. Consequentemente, aumenta também o tamanho e a complexidade das estruturas que organizam a rede e planejá-las tem se tornado uma tarefa bastante difícil de realizar sem ajuda tecnológica já que, quando mal elaboradas, a disposição dessas estruturas influencia bastante no custo de mantê-las, pois a forma como os concentradores de rede se interconectam afeta os gastos relacionados à manutenção e funcionamento da mesma.

Dessa forma, o estudo de soluções para minimizar estes custos é estimulado e, geralmente, topologias são propostas para se encaixarem em cada situação. Uma dessas topologias é a *anel-estrela*, que foi estudada inicialmente por Labbé *et al.* (1999) em busca da melhor organização de uma rede com essa topologia, o que acabou por situar esse problema na área de Otimização Combinatória (OC).

A área da Otimização engloba os problemas que têm por objetivo encontrar uma configuração para os elementos do problema que seja considerada ótima ou perfeita segundo uma meta pré-definida (PAPADIMITRIOU; STEIGLITZ, 1998). Então, o estudo de Labbé *et al.* (1999) mostrou que o objetivo de encontrar a melhor configuração da topologia *anel-estrela* segundo uma função que determina para cada conexão entre nós um custo real é um problema de OC.

Alguns trabalhos propõem abordagens exatas, que fornecem a melhor organização possível das conexões, outros apresentam abordagens heurísticas e/ou aproximativas, que não resolvem o problema de forma ótima, mas oferecem respostas muito próximas da solução ótima ou das melhores configurações já encontradas para o problema. Isto exposto, uma observação da literatura revelou que não haviam trabalhos que misturassem estratégias heurísticas com técnicas exatas para resolver o problema *anel-estrela*, ou mais formalmente *Problema do Ciclo Mediano sem Restrições de Capacidade*. Sendo assim, o desenvolvimento de algo semelhante parece promissor quando se objetiva a união das principais vantagens de cada vertente desta hibridização. (PUCHINGER; RAIDL, 2005) (JOURDAN *et al.*, 2009)

1.1 Objetivo Geral

Propor novas abordagens de resolução para o *Problema do Ciclo Mediano Sem Restrições de Capacidade*.

1.1.1 Objetivos Específicos

- Analisar as abordagens de resolução já propostas para o problema.
- Desenvolver hibridizações entre um algoritmo exato e heurísticas já existentes para o problema.
- Realizar testes para as duas formas de abordar o problema, heurística e híbrida, sobre instâncias já utilizadas por trabalhos anteriores.
- Analisar desempenho e resultados em comparação com tais trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

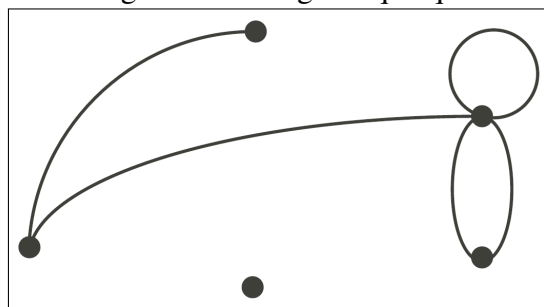
Com objetivo de habituar o leitor aos conceitos utilizados no decorrer do presente trabalho, este capítulo abordará temas que vão desde a estrutura utilizada para representar e estudar a problemática, passando pela área de estudo em que o trabalho se encaixa e culminando nos métodos utilizados na presente pesquisa. Sendo assim, a organização do capítulo se dará da seguinte forma: primeiro serão apresentados conceitos básicos sobre Teoria dos Grafos; logo após, conceitos sobre a área da Otimização Combinatória (OC); e, por fim, dando o tom do objetivo do presente trabalho, os conceitos dos Métodos Heurísticos e Métodos Exatos.

2.1 Teoria dos Grafos

2.1.1 Grafos

Um *grafo* é uma abstração matemática utilizada para representar entidades e as relações existentes entre elas. Essa representação é construída por meio de uma estrutura que define tais entidades como pontos, e as relações como linhas, ou ligações, entre esses pontos. O foco dessa abstração é descrever de forma sistemática as associações entre essas entidades e não as características que as definem. A Figura 1 fornece um exemplo de grafo qualquer (BONDY *et al.*, 1976).

Figura 1 – Um grafo qualquer



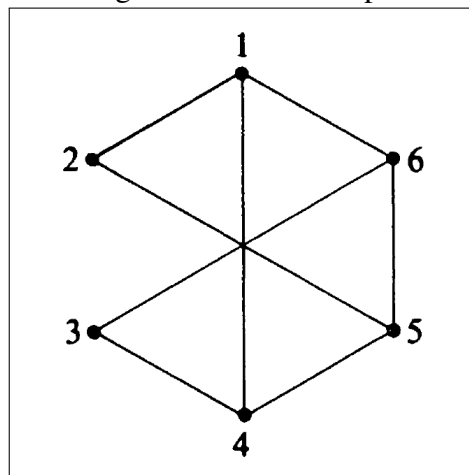
Fonte: (RUOHONEN, 2013)

Um grafo $G = (V, E)$ pode ser formalmente definido como um par de conjuntos que representam respectivamente os pontos e as ligações citadas anteriormente. G , então, é composto por um conjunto V de *vértices*, a nomenclatura formal definida para os pontos, e por outro conjunto E de *arestas*, que é a definição formal das ligações. V é representado por $\{1, 2, 3, \dots, n\}$, enquanto E é representado por pares de vértices pertencentes a V , matematicamente $E \subseteq \{(i, j) :$

$i, j \in V\}$, não importando a ordem em que aparecem no par, ou seja, $(i, j) = (j, i)$. O valor n , que simboliza o último vértice de V , também delimita o tamanho desse conjunto, $|V| = n$. De forma semelhante, estabeleceremos m como a cardinalidade $|E|$ do conjunto de arestas (RUOHONEN, 2013).

Neste trabalho, apenas conceitos de grafos simples serão considerados, tal qual definem Bondy *et al.* (1976) e Ruohonen (2013). Este tipo de grafo, como pode ser visto na Figura 2, possui nenhuma ou apenas uma aresta ligando cada par de vértices. Além disso, não existem *laços* em sua estrutura, isto é, arestas em que o par de vértices que a define é formado pelo mesmo vértice – tal como (i, i) .

Figura 2 – Grafo simples



Fonte: (BOLLOBÁS, 1979)

Ao longo do presente trabalho, as letras G , V e E serão utilizadas para representar genericamente um grafo e seus conjuntos formadores. Adicionalmente, é estabelecido o uso de $V(G)$ e $E(G)$ para determinar, respectivamente, os conjuntos de vértices e de arestas de um grafo G qualquer.

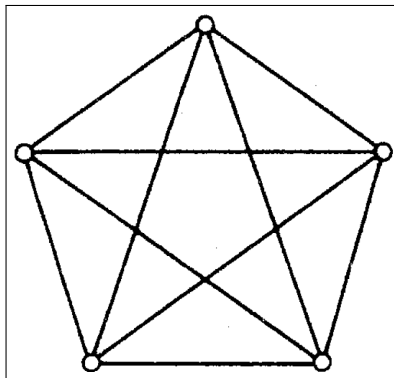
Se existe uma aresta $(i, j) \in E$ entre dois vértices i e j de V , diz-se que tal aresta *incide* sobre os dois vértices e que i e j são *adjacentes* entre si e também *extremidades* dessa aresta. As arestas de um grafo são ditas *não direcionadas*, pois cada uma delas permite acessar qualquer uma de suas extremidades a partir da outra.

Com o conceito de incidência e adjacência, podemos determinar o conjunto de arestas que incidem em um vértice. Este conjunto é denotado por $\Gamma(i) = \{(i, j) \in E : j \in V\}$ e sua cardinalidade, $|\Gamma(i)|$, nos diz o *grau* de tal vértice, isto é, a quantidade de arestas que incidem sobre i , ou a quantidade de vértices aos quais i é adjacente (BOLLOBÁS, 1979).

De acordo com Bondy *et al.* (1976), um grafo é dito *completo* quando todas as possibilidades de arestas estão presentes em E , ou seja, para todo possível par (i, j) , em que i e j pertencem a V , existe uma aresta em E que os interliga. De forma matemática, $E = \{(i, j) : i, j \in V; i \neq j\}$. A Figura 3 mostra um grafo completo com 5 vértices. Bollobás (1979), adicionalmente, define um *subgrafo* de G como um grafo formado por subconjuntos de $V(G)$ e de $E(G)$ com extremidades em $V(G)$. Portanto, H é um subgrafo de G , denotado por $H \subseteq G$, se, e somente se, $V(H) \subseteq V(G)$ e $E(H) \subseteq \{(i, j) : i, j \in V(H) \text{ e } (i, j) \in E(G)\}$.

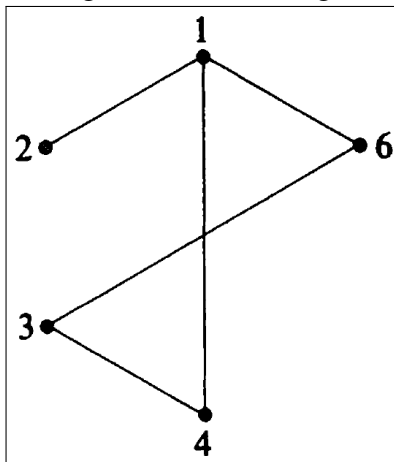
Um caso especial de subgrafos são aqueles que têm seu conjunto de arestas definido a partir do seu conjunto de vértices. Definimos como *subgrafo induzido* de G , um grafo H' que, dado $V(H')$, tem $E(H')$ formado, exatamente, por todas as arestas de G que possuem as duas extremidades em $V(H')$. Um subgrafo induzido do grafo mostrado anteriormente na Figura 2 é dado na Figura 4 (BOLLOBÁS, 1979).

Figura 3 – Grafo completo



Fonte: (BONDY *et al.*, 1976)

Figura 4 – Subgrafo induzido do grafo da Figura 2



Fonte: (BOLLOBÁS, 1979)

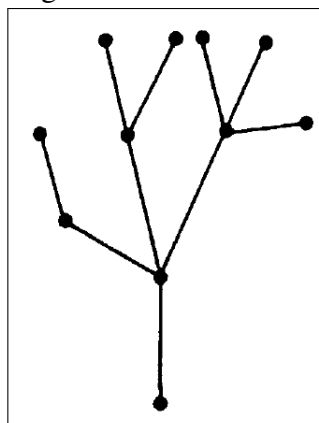
Em contrapartida, o *complemento* de um grafo G , como define Ruohonen (2013), é outro grafo \overline{G} que possui seu conjunto de arestas $E(\overline{G})$ definido como o complemento de $E(G)$, ou seja, $\overline{G} = (V(G), \overline{E(G)})$. Dessa forma, as arestas de \overline{G} são todas aquelas que não estiverem presentes em G . Por exemplo, o complemento de um grafo completo, como o mostrado na Figura 3, é um *grafo vazio*, sem aresta alguma.

Anteriormente foi determinado que uma aresta (i, j) permite que a partir do vértice i seja possível acessar j , e vice-versa. Isso possibilita definir estruturas mais abrangentes dentro de um grafo. Segundo Bollobás (1979), um *caminho* em um grafo é uma sequência $P = v_0v_1v_2v_3\dots v_k$ de vértices distintos, em que v_t e v_{t+1} , para $0 \leq t \leq k - 1$, devem ser ligados por uma aresta de G .

Bondy *et al.* (1976) define que dois vértices i e j são denominados *conectados* se existe pelo menos um caminho que comece em i e termine em j , tornando possível acessar j a partir de i , sendo a recíproca, acessar i a partir de j , também válida. Os vértice 2 e 3 do grafo simples apresentado na Figura 2 não são adjacentes, mas são conectados, pois, por exemplo, o caminho $P' = 3 - 6 - 5 - 2$ os interliga.

De forma complementar, Bollobás (1979) estabelece que um grafo é dito *conexo* quando todo possível par de vértices de V é conectado. Uma *componente* em G é um subgrafo conexo maximal de G . Ou seja, se G já é conexo, então possui apenas uma componente. Caso contrário, há partições do conjunto de vértices de G que não possuem arestas os interligando, dessa forma, G é *desconexo* e essas partições são denominadas componentes de G . A Figura 5 apresenta mais um exemplo de grafo conexo. O único grafo não conexo apresentado nesta seção é aquele mostrado na Figura 1 (BONDY *et al.*, 1976).

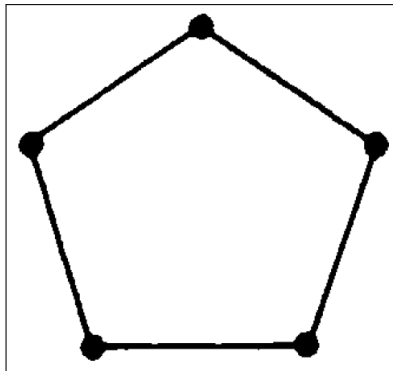
Figura 5 – Grafo conexo



Fonte: (BOLLOBÁS, 1979)

Ainda sobre conectividade, um *ciclo* pode ser definido como uma extensão de um caminho. Seja $C = v_0v_1v_2v_3\dots v_k$. C será um ciclo caso a sequência $v_0v_1v_2v_3\dots v_k$ seja um caminho e a aresta $(v_k, v_0) \in E(G)$. Um caso especial de ciclos são os chamados Ciclos Hamiltonianos. Estes são ciclos que perpassam todos os vértices de um grafo exatamente uma vez por cada um deles, respeitando, assim, a definição de ciclo. Um ciclo hamiltoniano contido no grafo da Figura 3 é mostrado na Figura 6.

Figura 6 – Ciclo derivado da Figura 3



Fonte: (BOLLOBÁS, 1979)

Estas estruturas, e outras não mencionadas aqui, podem ser induzidas por um conjunto de arestas. Por exemplo, caso as arestas em um conjunto E' possam ser organizadas de tal forma que um ciclo seja representado por elas, define-se que E' induz um ciclo.

Por serem estruturas simples e abrangentes, os grafos são uma abstração amplamente utilizada para representação de problemas em uma perspectiva matemática. Entretanto, nem todo problema pode ser reduzido a um grafo. Como esclarece Bondy *et al.* (1976), problemas de fluxo de tráfego ou de produção de rotas veiculares devem ser capazes de indicar vias de mão única e qual direção a circulação de veículos é realizada. Esta espécie de problemas leva ao surgimento de um novo tipo de grafos, os *grafos direcionados*.

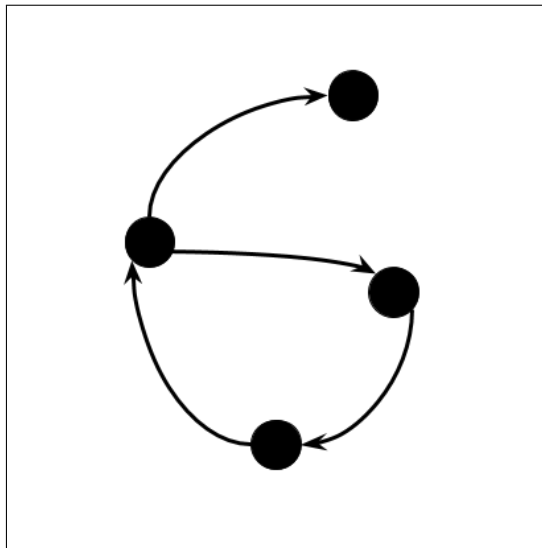
2.1.2 Digrafos

Bem como um grafo simples, um *grafo direcionado*, ou apenas *digrafo*, é uma abstração matemática utilizada para representar entidades e as relações existentes entre elas. Pontos e setas, ou ligações, formam a representação gráfica destas entidades e suas relações. Tais setas são o correspondente no digrafo para as linhas de um grafo, e definem uma *orientação* na relação existente entre duas entidades (BONDY *et al.*, 1976).

Formalmente, seja $D = (V, A)$ um digrafo simples, V é o conjunto de vértices de D ,

seguindo o que foi descrito para grafos, e A é o conjunto de *arcos*, ou *arestas direcionadas*, que representam as relações citadas anteriormente. Matematicamente, $A \subseteq \{[i, j] : i, j \in V\}$. Dado um arco $[i, j]$, é dito que i e j são unidos por ele e que i é a *cauda* da relação e j a *cabeça* (BONDY *et al.*, 1976), ou mais recentemente apenas vértices inicial e final respectivamente (RUOHONEN, 2013). Além disso, um digrafo simples, assim como um grafo simples, não possui laços, isto é, $\forall i \in V : [i, i] \notin A$, nem múltiplos arcos unindo os mesmos vértices, exceto para o caso dos arcos invertidos (BONDY *et al.*, 1976). A Figura 7 a seguir apresenta um digrafo simples.

Figura 7 – Digrafo simples



Fonte: Próprio autor

Diferentemente das arestas, não é possível acessar qualquer uma das extremidades de um arco, apenas acessar o vértice final a partir do inicial. Isso leva ao conceito de arcos invertidos. Dado um arco $[i, j]$, seu arco invertido é $[j, i]$, que pode ou não estar no digrafo. Isso define, por exemplo, que $[i, j] \neq [j, i]$ e que estes são pares ordenados, ou seja, a ordem em que aparecem na representação do arco os diferencia (RUOHONEN, 2013).

Grande parte dos conceitos apresentados na seção anterior podem ser trazidos para os digrafos, com a diferença de que estes conceitos devem considerar a orientação dos arcos neste novo cenário. Para os ciclos, por exemplo, o vértice final de um arco que o compõe deve ser sempre o inicial do arco seguinte da estrutura. Caminhos seguem uma regra semelhante, em que apenas seus primeiro e último vértices são exceção a ela. Outra diferença relacionada a essa orientação é que um vértice i está conectado a outro j se existe um caminho direcionado de i para j . A inversa nem sempre é verdadeira, j estará conectado a i apenas se o caminho inverso

existir (BONDY *et al.*, 1976).

Que nem para grafos, o foco de um digrafo é retratar os relacionamentos entre as entidades representadas. Esse fato faz com que os problemas reduzidos para essas estruturas matemáticas tenham sua resolução mais vinculada ao seu conjunto de ligações, sejam elas arestas ou arcos, que ao seu conjunto de vértices. A necessidade de, em muitos casos de resolução, enumerar permutações das ligações ou subconjuntos de $E(G)$ ou $A(D)$, dados o grafo G e o digrafo D , torna a tarefa de encontrar soluções exatas algo bastante difícil de se alcançar em tempo hábil. Para tratar dessa espécie de problemas faz-se o estudo da OC.

2.2 Otimização Combinatória

Muitos problemas práticos e reais, ou que são desenvolvidos no campo teórico para posterior aplicação, voltam sua atenção para encontro da melhor configuração dos elementos que os compõem. Essa configuração é comumente chamada de solução para o problema abordado. Esse objetivo o transforma em um problema de otimização (PAPADIMITRIOU; STEIGLITZ, 1998).

Para definir formalmente este tipo de problema, é necessário primeiro estabelecer o que seria uma *instância* dele. Segundo Papadimitriou e Steiglitz (1998), uma instância pode ser definida como um par (F, c) em que F é um conjunto de soluções viáveis de um problema de otimização e c uma função tal que $c : F \rightarrow \mathbb{R}$. O objetivo é então encontrar $f \in F$ em que $c(f) \leq c(s) \forall s \in F$, caso o propósito seja achar a solução com valor mínimo em c , ou $c(f) \geq c(s) \forall s \in F$, caso deseje-se encontrar aquela de valor máximo em c . f é chamada solução ótima para a instância.

Estabelecido o conceito de instâncias, um problema de otimização é dito como o conjunto de todas as possíveis instâncias de um problema de otimização. Devido ao aspecto autorreferente da definição, alguns problemas podem ser dados como exemplo. O Problema do Caixeiro Viajante (PCV) consiste da tarefa de encontrar uma rota para um caixeiro dentre um conjunto de cidades, esta rota deve passar por todas as cidades do conjunto, uma única vez em cada e o custo dessa viagem tem de ser o menor possível. Uma instância de PCV, dado um conjunto C de cidades, se caracteriza pelo par (F, c) , onde F é o conjunto de todas as permutações possíveis em rotas com $|C|$ cidades, utilizando todas as cidades presentes em C , e c uma função que avalia o custo de cada uma dessas rotas. O problema de otimização chamado PCV é o conjunto de todas as possíveis instâncias que possuam essas características.

Resolver uma instância de PCV é encontrar a rota $r \in F$ que possui $c(r) \leq c(r') \forall r' \in F$ (PAPADIMITRIOU; STEIGLITZ, 1998).

O propósito de encontrar uma solução dentre aquelas que estão em F justifica que a maioria desses problemas sejam resolvidos com técnicas iterativas. Sendo assim, com a abordagem adequada, é preciso percorrer a lista de elementos desses conjuntos e uma característica importante a se saber é se o conjunto a ser percorrido é contável ou não. Um conjunto é contável quando pode ser estabelecida uma bijeção entre tal conjunto e o conjunto dos números naturais. Dessa forma, percorrê-lo pode ser feito como percorre-se os naturais. Portanto, caso F seja finito ou infinito contável, o problema ao qual a instância (F, c) está associada é chamado de *combinatório*, como é de otimização também, a nomenclatura final torna-se problema de otimização combinatória. Os problemas em que as instâncias possuem conjuntos F não contáveis são chamados de problemas *contínuos*, ou de problemas de otimização contínua (PAPADIMITRIOU; STEIGLITZ, 1998).

Devido à complexidade de muitos dos problemas de otimização combinatória, duas linhas de estudo são fortes na busca por soluções a estes casos: exata e não-exata. A exata sacrifica o tempo de execução, e outros altos custos que o processo de resolução pode exigir, para encontrar a melhor solução entre todas. Enquanto que a não-exata sacrifica a localização da melhor solução para encontrar aquelas com custo próximo ao ótimo, sem, muitas vezes, garantir que este objetivo será atingido.

2.3 Métodos Heurísticos

Resolver problemas de otimização, como mencionado na seção anterior, pode ser uma tarefa difícil e cara computacionalmente. Entretanto, nem sempre encontrar a melhor solução possível é necessário. Muitas vezes é suficiente apenas localizar uma solução próxima da ótima, ou que esteja a uma distância verificável dela ou uma que apenas respeite um certo limite pré-determinado.

Os métodos heurísticos são procedimentos computacionais que conseguem produzir "boas" soluções sem a garantia de estarem próximas do ótimo global. Sua principal vantagem é o reduzido custo computacional necessário para que esse tipo de procedimento atinja seu objetivo. Essa junção de boas soluções com baixo custo computacional é possível pois os procedimentos tentam simular alguns processos naturais, ou pelo menos mais inteligentes que enumerar todas as possíveis soluções, para conseguir melhores resultados iterativamente. Esses métodos se dividem

em heurísticas e meta-heurísticas, os primeiros são mais específicos e limitados, enquanto que os últimos são mais abrangentes e genéricos (OSMAN; KELLY, 1997).

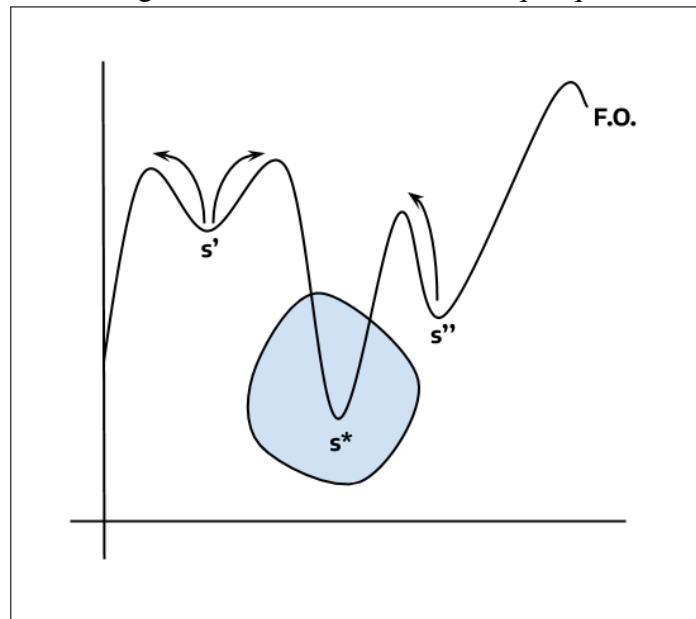
As heurísticas comuns têm funcionamento bastante específicos com objetivos limitados. Podem funcionar como procedimentos iniciais de algoritmos mais abrangentes, como as meta-heurísticas, ou podem ser utilizadas em conjunto para obter resultados mais promissores. Essa limitação existe pois suas performances, em sua maioria, baseiam-se em estruturas de vizinhança, que geralmente limitam a execução a um intervalo de soluções que, na maior parte dos casos, não contém o ótimo global. Essa característica pode ser melhor observada na Figura 8. Elas podem ser divididas em duas classes: construtivas e de refinamento (OSMAN; KELLY, 1997) (RIBEIRO, 1996).

As heurísticas construtivas são métodos que conseguem montar uma solução adicionando elementos a ela um a um de forma iterativa. São utilizados com frequência em procedimentos de atuação mais ampla realizando a produção de uma solução inicial que será refinada no decorrer da execução. Se dividem em heurísticas gulosas, que usam a função objetivo para ordenar os elementos de forma que o próximo a ser adicionado cause o melhor ganho para a solução sendo montada, e em heurísticas aleatórias que escolhem o próximo elemento aleatoriamente (RIBEIRO, 1996).

A segunda classe de heurísticas comuns é a de refinamento, elas usam o conceito de *vizinhanças* para realizar uma busca sobre o conjunto de soluções do problema a ser resolvido. Uma vizinhança N está atrelada a uma solução s e é obtida por meio de uma *movimentação* m na estrutura de s . Tal movimentação gera outras soluções $s' \in N(s)$ que podem ter valores melhores para sua função objetivo. Uma heurística de refinamento, portanto, utiliza uma solução inicial e a refina buscando em uma de suas vizinhanças por soluções vizinhas melhores. A Figura 8 exemplifica regiões de vizinhança de soluções com base no custo de cada uma.

Um desses métodos é o da Descida/Subida, que gera toda a vizinhança de s e escolhe seu melhor vizinho como nova solução corrente para o problema. Esse processo é repetido até que nenhum vizinho da solução corrente seja melhor, o que significa encontrar um ótimo local. Um segundo método, chamado de método da Primeira Melhor, evita buscar em toda a vizinhança ao produzir vizinhos apenas até que uma solução melhor que a corrente seja encontrada. Outra forma de evitar a busca em toda a vizinhança é gerar um vizinho de forma aleatória e substituir a solução corrente apenas se este novo vizinho tiver custo melhor, senão continuar procurando até que um limite de repetições seja atingido. Esta é a Descida/Subida

Figura 8 – Gráfico de uma FO qualquer



Fonte: Próprio autor

Randômica. Por fim, um último método, chamado Descida em Vizinhança Variável (VND, do inglês *Variable Neighborhood Descent*), tenta produzir melhores resultados ao mudar a vizinhança em que melhores soluções são procuradas. Ou seja, muda-se o movimento aplicado sobre a solução corrente para tentar encontrar um espaço de soluções diferente e mais favorável.

Dada a limitação no uso das heurísticas comuns, foram desenvolvidos novos métodos para se ter procedimentos que consigam atuar em uma classe maior de problemas e fugir dos vales de ótimos locais vistos na Figura 8. As Meta-heurísticas utilizam heurísticas simples para construir e refinar uma solução, elas geralmente repetem um desses processos mais simples, ou os dois, em diferentes áreas do espaço de soluções do problema. Com elas também é possível usar diferentes heurísticas para diferentes problemas, pois em sua estrutura não são fixados quais procedimentos devem ser usados, apenas de qual tipo eles devem ser (RIBEIRO, 1996).

As principais meta-heurísticas baseiam-se em conceitos biológicos, sejam evolucionários ou do sistema nervoso humano, ciências matemáticas e físicas, abordagens inteligentes na resolução dos problemas e mecânica estatística (OSMAN; KELLY, 1997). No presente trabalho elas serão separadas em três categorias: relacionadas a buscas, *multistart* e com base em processos naturais.

Entre aquelas relacionadas a buscas, as principais são Busca Tabu (BT), *Iterated Local Search* (ILS) e Busca em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*). BT é uma meta-heurística que usa os métodos de busca local para analisar a vizinhança de uma solução corrente e mover-se para a melhor solução vizinha desta, mesmo que essa nova

solução tenha um custo pior que a solução corrente. A melhor solução encontrada no decorrer da execução é sempre guardada e o processo de mudança da solução corrente se dá apenas para a exploração de diferentes vizinhanças na expectativa de sair de um vale de ótimos locais. Além disso, este procedimento usa uma lista tabu que guarda essas mudanças na solução corrente para que elas não sejam realizadas novamente (GENDREAU *et al.*, 2010).

O método ILS, em alternativa, tenta sair de ótimos locais ao realizar perturbações na solução corrente encontrada. Essa perturbação é uma mudança na estrutura dessa solução que pode ser grande, produzindo outra solução em um espaço distante de busca, ou pequena, que mantém a busca próxima da solução anterior. A última meta-heurística nesta classe, VNS, altera os espaços de busca mudando, em sequência, as vizinhanças relacionadas à solução corrente. Em cada uma delas, uma solução vizinha aleatória é gerada e, caso tenha custo pior que a corrente, o procedimento continua para próxima vizinhança, caso contrário, o procedimento se move para essa nova solução e gera para ela soluções vizinhas do primeiro tipo utilizado na sequência citada anteriormente (GENDREAU *et al.*, 2010).

A segunda categoria das meta-heurísticas, *multi-start*, engloba métodos que tentam escapar de ótimos locais por meio da produção de uma série de soluções iniciais diferentes. Um desses métodos nomeia a categoria e usa essas soluções iniciais para diversificar o espaço de busca. A cada solução gerada é aplicada uma busca local. Esse processo é repetido até que um critério de parada seja atendido. De forma similar, um outro procedimento, *Greedy Randomized Adaptive Search Procedures* (GRASP), produz essas soluções iniciais seguindo um grau de gulosidade/aleatoriedade definido previamente. A busca local também é aplicada no resultado (GENDREAU *et al.*, 2010).

A terceira e última classe aborda as meta-heurísticas que baseiam-se em processos naturais. A meta-heurística *Simulated Annealing* (SA), por exemplo, simula um processo termodinâmico de resfriamento de átomos aquecidos. O procedimento parte de uma solução inicial qualquer e gera soluções vizinhas aleatoriamente. A solução corrente muda quando a vizinha tem custo melhor, caso tenha custo pior, a solução corrente muda segundo fator probabilístico que se altera de forma a parecer com o resfriamento dos átomos (GENDREAU *et al.*, 2010).

Se baseando no processo biológico da evolução humana, os Algoritmos Genéticos por padrão transformam as soluções em cadeias de caracteres. O objetivo é mesclar soluções afim de produzir *descendentes* com melhor custo. Esses métodos avaliam populações de soluções

e aplica, além desse cruzamento, processos de mutação e seleção nelas. A última meta-heurística, denominada de Colônia de Formigas, imita o comportamento que esses insetos têm quando buscam por recursos. As formigas deixam feromônios durante a busca para indicar para as outras o caminho que levou ao recurso. A meta-heurística gera soluções e marca partes delas com um valor de feromônio, as novas soluções produzidas então consideram, em sua escolha de elementos para a solução que estão construindo, quanto de feromônio um candidato tem. Com isso, os elementos com esse valor mais acentuado são escolhidos com maior frequência, gerando uma cooperação na montagem da solução final (GENDREAU *et al.*, 2010).

Mesmo com o bom desempenho computacional, nenhum desses procedimentos garante certeza na localização da solução ótima. E para alguns casos reais apenas uma solução exata é interessante. Para esses casos, existe um tipo específico de procedimentos, os algoritmos exatos.

2.4 Métodos Exatos

Pode ser considerado um método exato para resolução de problemas de otimização toda e qualquer abordagem que encontre a melhor configuração possível de uma solução para um problema dessa natureza, desde que se mantenha dentro do contexto do problema e de sua instância sendo resolvida. Esse fato tornaria difícil categorizar os métodos que têm tal característica. Entretanto, muitos deles não abrangem apenas um problema.

O Algoritmo Simplex, por exemplo, é um procedimento genérico e eficiente para resolução de sistemas lineares. Dado que a Programação Linear (PL) modela um sistema real para um conjunto de equações e inequações lineares, este pode ser adaptado para resolver tais problemas. O Simplex, assim como outros métodos de natureza semelhante, tenta fugir de técnicas de resolução conhecidas como de "força bruta", isto é, que testam todas as possíveis respostas, neste caso soluções, até encontrar a melhor (GOLDBARG; LUNA, 2005).

Um outro algoritmo, desenvolvido posteriormente, para resolver problemas de programação linear em tempo polinomial é chamado o Método Elipsoide. Utilizando a ideia de espaço de soluções viáveis para um problema dessa natureza, este algoritmo, em termos gerais, particiona o espaço de soluções sucessivamente por meio do cálculo de elipsoides que englobam tal espaço de soluções. Devido sua complexidade, este método é pouco utilizado na prática (BERNHARD; VYGEN, 2008).

Saindo da PL, a Programação Inteira (PI) também possui métodos mais genéricos,

como, por exemplo, aqueles relacionados à "família" *Branch*. O método *Branch-and-Bound* usa um processo de ramificação do espaço de soluções para diminuir a quantidade delas que serão avaliadas. Além disso, estabelece limites de custo para evitar realizar essas ramificações de forma inútil, tais limites baseiam-se no fato de que ao tornar as variáveis do problema em contínuas, a resolução desse novo problema limita o custo do problema inteiro original (PAPADIMITRIOU; STEIGLITZ, 1998).

O procedimento conhecido por *Branch-and-Cut* é um método híbrido que mescla a ramificação do *Branch-and-Bound* com uma técnica de *planos de corte*. Essa técnica de cortes usa também a limitação contínua para o problema inteiro, no entanto, o diferencial é a ideia de que se forem adicionadas ao problema restrições que removam do espaço de soluções um conjunto delas que não contenha soluções inteiras, então o resultado final do algoritmo não é alterado. Com essas restrições, o algoritmo corta o conjunto de soluções até que a resolução linear produza uma solução inteira (PADBERG; RINALDI, 1991) (PAPADIMITRIOU; STEIGLITZ, 1998).

O terceiro procedimento que usa o *Branch* é chamado *Branch-and-Price* e também é um híbrido. Seu segundo processo diz respeito a uma técnica chamada *geração de colunas*. Na resolução linear utilizando Simplex, as colunas representam as variáveis do problema, a técnica mencionada retarda a inclusão dessas colunas através das várias resoluções lineares executadas nas ramificações. Um problema mestre resolve o problema linearizado com poucas variáveis, e um subproblema define qual a variável, que não foi considerada ainda na resolução, que mais impacta na função objetivo. O mestre então inclui essa variável e é executado novamente. Esse processo é repetido até que o subproblema não encontre variável com impacto na função (BARNHART *et al.*, 1998).

Assim como um problema pode ter muitas variáveis, muitas restrições também podem trazer dificuldades. Para esses casos, uma técnica parecida com a geração de colunas é comumente utilizada, a geração de linhas. Na resolução linear, as linhas representam as restrições, então este método retarda a inclusão dessas linhas, pondo-as no modelo apenas quando necessário. O modo de operação é similar ao da geração de colunas.

Após a apresentação desse arcabouço de conhecimentos, é possível entender melhor os aspectos que envolvem o objeto de estudo do presente trabalho. Com isso, a seguir é apresentada uma descrição mais completa sobre o Problema do Ciclo Mediano sem Restrições de Capacidade.

3 O PROBLEMA DO CICLO MEDIANO SEM RESTRIÇÕES DE CAPACIDADE

Neste capítulo será apresentada uma descrição para o Problema do Ciclo Mediano sem Restrições de Capacidade (PCMRC). Além disso, também será descrito um modelo matemático que o define como um problema de Otimização Combinatória (OC).

Esse problema foi introduzido na literatura por Labbé *et al.* (1999), sob o nome de Problema do Ciclo Mediano 1 (MCP1, do inglês *Median Cycle Problem 1*). Além de MCP1, o referido trabalho apresenta um MCP2, que possui algumas diferenças em relação a MCP1 referentes aos objetivos de cada problema. Em Labbé *et al.* (2004) os autores voltam a abordar o MCP1. Neste segundo trabalho, MCP1 passa a se chamar Problema Anel-Estrela (RSP, do inglês *Ring Star Problem*). A seção a seguir apresenta uma descrição mais detalhada do problema e de suas características matemáticas.

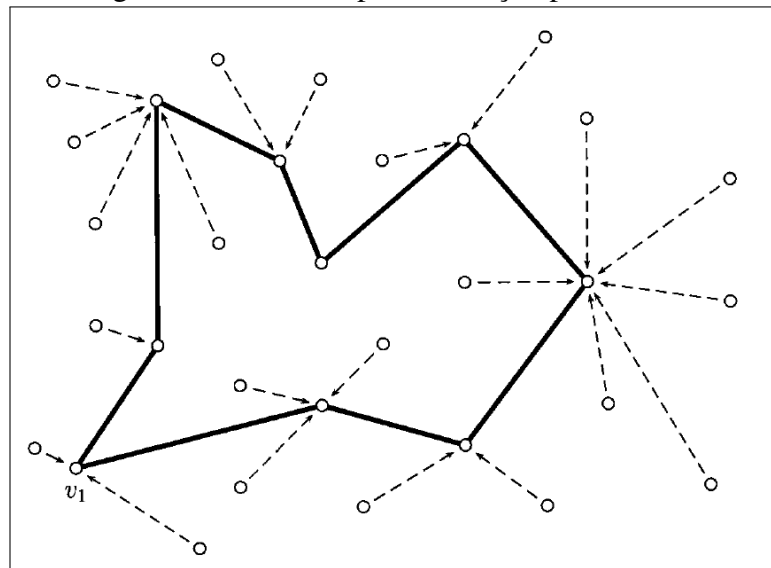
3.1 Caracterização

O PCMRC pode ser descrito como a tarefa de separar adequadamente dois subconjuntos disjuntos de vértices em um grafo: o primeiro subconjunto é composto pelos vértices que irão induzir um ciclo no grafo, e deve conter um vértice previamente fixado como *raiz* no problema; o segundo é formado por vértices clientes que devem ser atribuídos, ligados direta e individualmente, a vértices no ciclo induzido anteriormente. O processo de separação deve ter como objetivo a minimização dos custos de montagem do ciclo e de atribuição dos clientes. A Figura 9 nos mostra uma possível solução para uma instância de PCMRC. A *raiz* é representada na figura por v_1 .

De modo formal, PCMRC pode ser definido como a seguir. Seja $G = (V, E \cup A)$ um grafo completo misto composto por um conjunto de vértices $V = \{1, 2, 3, \dots, n\}$, um conjunto de arestas não direcionadas $E = \{(i, j) : i, j \in V\}$, e um conjunto de arcos direcionados $A = \{[i, j] : i, j \in V\}$. O vértice 1 é definido como a *raiz*. Assumimos $n \geq 6$ com objetivo de excluir instâncias com resolução trivial ou que sejam degeneradas.

As arestas são utilizadas para construir um ciclo mediano na solução, enquanto os arcos definem atribuições, que são usados para ligar vértices que não estão no ciclo a nós pertencentes a ele. É importante deixar claro que os arcos que possuem duas extremidades iguais – $[i, i]$, com $i \in V$ –, denominados laços, também estão presentes em A e têm um papel na modelagem descrita na próxima seção. Além disso, para cada aresta $(i, j) \in E$ é definido um

Figura 9 – Um exemplo de solução para PCMRC



Fonte: (LABBÉ *et al.*, 2004)

custo $c_{ij} \in \mathbb{R}_+$, e para cada arco $[i, j] \in A$ também é estabelecido um custo $d_{ij} \in \mathbb{R}_+$.

O objetivo central do PCMRC é minimizar, em instâncias que representem uma topologia anel-estrela, os custos de montagem do ciclo e de atribuição de nós. Estes são calculados a partir da soma dos pesos de cada aresta e cada arco presentes em uma solução. Definida a estrutura, a construção de uma solução deve então, de forma conjunta, localizar um subconjunto $V' \subseteq V$ que contenha o vértice estabelecido anteriormente como *raiz* e atribuir cada vértice $i \notin V'$ a exatamente um vértice $j \in V'$.

V' deve induzir um ciclo de custo mínimo, ou seja, é obrigatório haver pelo menos um subconjunto $E' \subseteq E(V')$ definindo um ciclo que perpassasse todos os vértices presentes em V' , e o custo deste ciclo deve ser o menor entre todas as possibilidades de E' que induzam este tipo de ciclo para esse subconjunto de vértices. O processo de atribuições consiste em encontrar para cada nó $i \notin V'$ o arco $[i, j]$ de menor custo tal que $j \in V'$. Todos os vértices devem ou compor o ciclo mediano ou ser atribuídos a ele, não podendo haver nenhum nó desconexo na solução. Essa restrição fica clara no exemplo da Figura 9.

PCMRC é classificado como NP-difícil (MARTÍN, 2000 apud DIAS, 2006, p. 15). Essa classificação se baseia no fato de que o Problema do Caixeiro Viajante (PCV), um dos mais conhecidos problemas de OC e também NP-difícil, é um caso especial de PCMRC. O PCV consiste na tarefa de encontrar um ciclo que inclua todos os vértices de um grafo, e que a soma dos custos de cada aresta presente na solução seja a menor possível. Dessa forma, caso os custos de atribuição dos arcos de G sejam impraticáveis em uma instância de PCMRC, ou seja, infinitos

ou altos o suficiente, uma solução ótima para tal instância deverá incluir todos os vértices de G no ciclo central V' , transformando a resolução de PCMRC em uma resolução de PCV, isto é, encontrar o ciclo hamiltoniano de menor custo em V' .

Definidos a estrutura do problema, seus objetivos e sua complexidade, a próxima seção aplica essas características fundamentais com o objetivo de descrever e delimitar o PCMRC a partir de um modelo matemático.

3.2 Modelo

O modelo apresentado em Labbé *et al.* (1999), consolidado em Labbé *et al.* (2004) e incrementado por Dias (2006) é construído com a adição de variáveis binárias para qualquer ligação no grafo, ou seja, uma variável de decisão para cada aresta e cada arco presente na estrutura definida na seção anterior. Essas variáveis terão o papel de definir quais dessas ligações estarão presentes na solução encontrada para uma dada instância do problema.

Sendo assim, dado um grafo misto G , para cada aresta (i, j) pertencente ao conjunto $E(G)$ é definida uma variável $x_{ij} \in \{0, 1\}$, que assumirá o valor 1 caso a aresta (i, j) componha o ciclo definido na solução final. Caso contrário, x_{ij} assume valor 0. De forma semelhante, para cada arco $[i, j]$ presente em A a variável $y_{ij} \in \{0, 1\}$ é estabelecida. y_{ij} terá valor igual a 1 apenas caso o vértice i esteja atribuído ao vértice j na solução. As variáveis y_{ii} , relacionadas aos laços $[i, i]$, terão valor 1 se o vértice i estiver no ciclo ao fim da resolução. Caso contrário, terão valor 0.

Por fim, com o objetivo de abreviar a descrição de algumas restrições do modelo, são estabelecidas duas notações. Estendemos o conceito de $\Gamma(i)$, conjunto de arestas incidentes ao vértice i . Essa definição passa a considerar também conjuntos de vértices. Portanto, dado um subconjunto $S \subset V(G)$, $\Gamma(S)$ representa o conjunto de arestas do grafo que possuem exatamente uma extremidade pertencente a S , isto é, $\Gamma(S) = \{(i, j) \in E(G) : i \in S \text{ e } j \notin S\}$. Nos casos em que $S = \{i\}$, usa-se $\Gamma(i)$ apenas. Essa definição estabelece, então, o conjunto de arestas que ligam S a seu complemento. A segunda notação proposta tem como objetivo contar, passado um conjunto de arestas, quantas delas fazem parte do ciclo na solução final. Formalmente, seja

$$E' \subseteq E(G), x(E') = \sum_{(i,j) \in E'} x_{ij}.$$

Após esse preâmbulo, pode-se definir o modelo matemático de PCMRC como sendo:

$$\min \sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{[i,j] \in A} d_{ij}y_{ij} \quad (3.1)$$

$$\text{sujeito a } x(\Gamma(i)) = 2y_{ii}, \quad \forall i \in V, \quad (3.2)$$

$$\sum_{j \in V} y_{ij} = 1, \quad \forall i \in V \setminus \{1\}, \quad (3.3)$$

$$x(\Gamma(S)) \geq 2 \sum_{j \in S} y_{ij}, \quad \forall S \subset V : 1 \notin S, i \in S, \quad (3.4)$$

$$y_{ij} \leq y_{jj}, \quad \forall [i, j] \in A, i \neq j, \quad (3.5)$$

$$y_{11} = 1, \quad (3.6)$$

$$y_{1j} = 0, \quad \forall j \in V \setminus \{1\}, \quad (3.7)$$

$$y_{ij} \geq 0, \quad \forall [i, j] \in A, \quad (3.8)$$

$$y_{jj} \in \{0, 1\}, \quad \forall j \in V \setminus \{1\}, \quad (3.9)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (3.10)$$

Os somatórios em (3.1) definem a função objetivo do problema, que visa minimizar os custos de roteamento e de atribuição das ligações do grafo. Observe que, mesmo a soma considerando todas as ligações existentes, apenas os custos de arestas e arcos presentes na solução final serão computados, pois cada um é multiplicado por sua variável de decisão correspondente.

As equações e inequações (3.2)-(3.10) determinam as limitações que desenham o problema sob uma perspectiva matemática. As restrições (3.2) estabelecem que o grau de um vértice deve ser 2 caso ele esteja no ciclo, $y_{ii} = 1$, ou 0 caso contrário. Isso porque $x(\Gamma(i))$ conta o número de arestas do ciclo que incidem sobre i .

As restrições (3.3) definem que todo vértice ou está no ciclo, $y_{ii} = 1$, ou deve estar atribuído a outro vértice, $y_{ij} = 1$. Note que não há possibilidade de que um vértice possua essas duas características simultaneamente.

As restrições (3.4) visam impor que qualquer subconjunto $S \subset V \setminus \{1\}$, caso contenha pelo menos um vértice pertencente ao ciclo, esteja conectado ao seu complemento por no mínimo duas arestas do ciclo da solução. Com isso, essas restrições também limitam o número de ciclos na solução a apenas um. Caso exista mais de um ciclo em um dos possíveis S , significa que ou todas as arestas de um, ou mais de um, desses ciclos estão contidas em $E(S)$, ou que todos esses ciclos fornecem, então, pelo menos um par de suas arestas que conectam S ao seu complemento. O segundo caso respeita (3.4). No entanto, o primeiro caso, apesar de também respeitar (3.4),

indica que existe pelo menos um subconjunto S que contém vértices do ciclo e não está conectado ao seu complemento por pelo menos duas arestas. Isso aponta que só pode haver um ciclo na solução.

A combinação de (3.2), (3.7), (3.6) e (3.10) garante que o ciclo terá o vértice *raiz* em sua composição. A restrição (3.7) delimita que o mesmo não deve ser atribuído a nenhum outro nó do grafo.

(3.3), (3.8) e (3.5) juntas garantem que vértices não presentes no ciclo só possam ser atribuídos a vértices presentes no ciclo. Essa garantia é alcançada pois a variável que define se uma atribuição pertencerá à solução está limitada pela variável do laço que determina se o vértice que recebe essa atribuição pertencerá ao ciclo. Dessa forma, caso y_{jj} tenha valor 0, denotando que j não está no ciclo, então y_{ij} só poderá ter valor 0 também, limitando que um vértice i só possa ser atribuído a um j que pertença ao ciclo. Isso só é possível porque (3.3) e (3.8) estabelecem $0 \leq y_{ij} \leq y_{jj} \leq 1$.

As restrições (3.10) e (3.9) definem a integralidade de parte das variáveis de decisão do problema. (3.8) define uma relaxação nas variáveis de decisão referentes às atribuições de vértices ao ciclo. Isso porque dado um ciclo ótimo, a minimização da função objetivo é alcançada apenas escolhendo o arco de menor custo que liga o vértice que não está no ciclo a um que esteja.

A modelagem apresentada e descrita acima enquadra PCMRC como um problema de programação linear inteira mista. Isso porque algumas de suas variáveis de decisão só podem assumir valores inteiros, como x_{ij} e y_{ii} em (3.10) e (3.9) respectivamente, enquanto outras podem assumir valor real como y_{ij} em (3.8).

Com o problema caracterizado e seu modelo definido, passamos a considerar os estudos que já foram realizados sobre o mesmo. A seguir apresentamos alguns trabalhos sobre o PCMRC que possuem diferentes perspectivas sobre como resolvê-lo.

4 TRABALHOS RELACIONADOS

Após sua primeira formulação, conforme apresentado no Capítulo 3, o Problema do Ciclo Mediano sem Restrições de Capacidade (PCMRC) foi abordado de diferentes formas. Reformulações na sua estrutura e em seu modelo, e diferentes técnicas de resolução, avançaram a pesquisa sobre o problema. Nas seções a seguir serão apresentados exemplos de abordagens ao PCMRC que compreendem vertentes importantes da Otimização Combinatória (OC).

Um bom exemplo é o trabalho de Dias (2006) que utiliza vários procedimentos heurísticos, meta-heurísticos e hibridizações entre estes em busca de produzir melhores soluções para instâncias do problema. Calvete *et al.* (2013), por sua vez, propõem uma nova formulação para o problema e uma nova forma de resolução utilizando uma heurística evolucionária. Enquanto que Chen *et al.* (2017) tratam do problema sob um aspecto aproximativo. Antes destes, será apresentado a seguir o trabalho de Labbé *et al.* (2004) que consolidou o modelo original do problema com uma análise poliédrica do mesmo e a proposta de um algoritmo exato para resolvê-lo.

4.1 *The Ring Star Problem: Polyhedral analysis and exact algorithm*

Labbé *et al.* (2004) consolidou o modelo apresentado na Seção 3.2, com exceção apenas da restrição (3.5), devidamente creditada ao trabalho de Dias (2006). Além disso, foi o primeiro trabalho a definir PCMRC como uma boa abstração para montagem de redes de telecomunicações, antes, sempre havia sido encaixado em uma classe mais ampla e genérica relacionada à localização-alocação de instalações.

Como visto antes, este modelo foi proposto em Labbé *et al.* (1999), onde é realizada uma análise poliédrica sobre algumas inequações derivadas da restrição (3.4). O trabalho de 2004, que reúne os mesmos autores, solidifica o modelo ao incluir uma série de novas restrições que induzem novas facetas poliédricas, importantes para a análise realizada. Por fim, os autores realizam experimentos em um número maior de instâncias, com quantidade elevada de nós, demonstrando a eficiência do algoritmo proposto sobre este espaço de testes.

Sua análise poliédrica serve, principalmente, ao processo de separação de restrições realizado comumente em algoritmos exatos. Esta técnica visa tornar viável a execução de um algoritmo deste tipo sobre problemas que consideram uma quantidade extensa de restrições, que levam a um número intratável de regras a serem processadas. As facetas encontradas e

demonstradas em tal análise indicam as melhores, entre as restrições, a serem utilizadas na separação, pois revelam aquelas que englobam, e restringem (cortam o conjunto de soluções), todo um conjunto de soluções que podem as violar. O processo mencionado está incluso na construção e execução do algoritmo exato descrito a seguir.

4.1.1 Algoritmo Branch-and-Cut

PASSO 1: Estabelece um limite superior para as soluções produzidas com base na instância recebida. Isso é feito construindo uma solução gulosa que vai incluindo novos vértices ao ciclo, definido por um conjunto que inicialmente contém apenas a *raiz*. O vértice i a ser inserido é escolhido considerando o valor mínimo de uma função L que calcula a relação entre o aumento no custo de ciclo causado pela inserção de um vértice e a redução no custo de atribuição.

$$L(i, \lambda) = \lambda \cdot inc(\bar{C}, i) + (1 - \lambda) \cdot (-dec(\bar{C}, i)),$$

onde $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, inc é o valor do incremento mínimo gerado no custo de C ao inserir o vértice i e dec o valor da redução no custo de atribuição causada pela realocação de i . As inserções são executadas até que a função não consiga mais reduzir o custo da solução construída. Além disso, essa fase monta uma lista \mathcal{L} de subproblemas (soluções) lineares apenas com as restrições (3.2), (3.3) e (3.6) unidas à regra abaixo:

$$x_{1i} \leq y_{ii} \quad \forall i \in V \setminus \{1\}.$$

PASSO 2: Escolhe a solução de menor custo em \mathcal{L} . Caso a lista esteja vazia a execução para.

PASSO 3: Se essa solução for viável para PCMRC e tiver custo menor que a melhor solução encontrada até então, essa passa a ser a melhor solução encontrada até o momento. Depois o Passo 2 é executado novamente. Na primeira comparação, obviamente, a solução testada será automaticamente a melhor encontrada até o momento.

PASSO 4: (Heurística baseada nas soluções lineares) Dado que os subproblemas na lista \mathcal{L} não consideram variáveis inteiras, suas soluções podem ser fracionárias. Caso sejam fracionárias, a cada 5 comparações do Passo 3, a seguinte heurística é aplicada: as variáveis x devem ser ordenadas de forma decrescente; as arestas atreladas a essas variáveis são selecionadas, uma por vez e nessa ordem, até que um ciclo seja formado; caso a *raiz* não esteja no ciclo, esta deve ser

inserida na melhor posição, definida utilizando a função L ; o ciclo é otimizado com um algoritmo 2-aproximativo. Se o custo dessa solução inteira for melhor que o custo da melhor encontrada até o momento, essa nova solução passa a ser a melhor.

PASSO 5: Separação de Restrições. Até 300 restrições são incluídas a modelagem dos subproblemas do Passo 1. As restrições adicionadas podem pertencer às restrições (3.4), aos 2 conjuntos de restrições abaixo, provenientes da análise poliédrica:

$$y_{ij} + y_{jk} + y_{ki} \leq 1, \quad \forall i, j, k \in V \setminus \{1\} : i \neq j \neq k \quad (4.1)$$

$$x(\Gamma(S)) \geq 2(y_{ij} + y_{jk} + y_{ki}), \quad \forall S \subseteq V \setminus \{1\}, \forall i, j, k \in V \setminus \{1\} : i \neq j \neq k \quad (4.2)$$

ou à restrição *2-matching* de ciclo: (BAUER, 1997 apud LABBÉ *et al.*, 2004, p. 180)

$$x(E(H)) + x(T) \leq \sum_{i \in H} y_{ii} + \frac{|T| - 1}{2} \quad \forall H \subset V; T \subset \Gamma(H) \quad (4.3)$$

satisfazendo:

- (i) $\{i, j\} \cap \{k, l\} = \emptyset$ para $[i, j], [k, l] \in T$ e $[i, j] \neq [k, l]$,
- (ii) $|T| \geq 3$ e ímpar.

Caso não haja mais restrições para adicionar, o próximo Passo deve ser o 6.

PASSO 6: Ramificações de subproblemas. Criam-se mais dois subproblemas com a ramificação baseada nas variáveis relacionadas ao ciclo, y_{ii} e x_{ij} . Dado um subproblema, são escolhidas as 5 variáveis y mais próximas de 0.5 e a ramificação ocorre arredondando seu valor para cima, 1, e para baixo, 0. Caso as 5 variáveis escolhidas sejam inteiras, o procedimento é executado com as variáveis x . Os novos subproblemas são incluídos em \mathcal{L} e o Passo 2 é novamente executado.

Todos os passos do algoritmo apresentado acima são compostos por procedimentos comuns à resolução de problemas no cenário da computação, um procedimento guloso, heurística, seleção das melhores soluções. No entanto, o Passo 5 é diferente, executa uma inclusão de restrições durante a resolução de uma instância e faz isso através de um processo chamado de separação dessas restrições.

4.1.1.1 Separações de Restrições

Essas estratégias de separação produzem como saída as restrições do problema que estão sendo mais violadas pela solução passada inicialmente. Portanto, seja $G = (V, E \cup A)$ o grafo

definindo na instância sendo resolvida pelo algoritmo, e (x^*, y^*) o conjunto dos valores de todas as variáveis x e y em uma solução para essa instância. É definido, então, $G^* = (V^*, E^* \cup A^*)$ onde $V^* = \{i \in V : 0 < y_{ii}^* < 1\}$, $E^* = \{(i, j) \in E : 0 < x_{ij}^* < 1\}$ e $A^* = \{[i, j] \in A : i \neq j, 0 < y_{ij}^* < 1\}$.

A separação das restrições (3.4) considera apenas as arestas em E^* e os arcos em A^* e possui dois processos que dependem da conectividade de G^* para serem acionados. Se G^* não é conexo, cada conjunto $V(C)$, onde C é uma das componentes de G^* que não contém a *raiz*, produz uma restrição violada com relação a qualquer vértice $i \notin V(C)$. Caso esse grafo seja conexo, fixando um vértice $j \in V \setminus \{1\}$, é possível encontrar a restrição do tipo (3.4) mais violada reduzindo essa tarefa de separação a um problema de fluxo máximo.

Esse problema é descrito a seguir: seja $G^{**} = (V^{**}, E^{**})$ um grafo com $V^{**} = V^* \cup \{n+1\}$ e $E^{**} = E^* \cup \{(n+1, k) : k \in V^*\}$. A capacidade de fluxo das arestas $e \in E^*$ que foram incluídas em E^{**} deve ser igual a x_e^* e a capacidade das novas arestas, criadas para ligar $n+1$ ao grafo, deve ser $2y_{jk}^*$, onde j é o vértice fixado antes do problema ser descrito. Por fim, é selecionado o conjunto $S' \subset V^{**}$ que não contenha *raiz*, contenha j e a capacidade Δ do corte $\Gamma(S')$ em G^{**} é mínima. Se $\Delta \geq 2$, não existe restrição violada pela solução (x^*, y^*) que envolva j . No entanto, caso $\Delta < 2$, $S = S' \setminus \{n+1\}$ define uma restrição (3.4) mais violada que envolve j . Resolvendo esse problema de fluxo para cada vértice diferente da *raiz* obtém-se as restrições mais violadas para cada vértice, e, assim, é possível decidir quais delas devem ser incluídas no modelo durante a execução do *Branch-and-Cut*.

Para encontrar as restrições (4.3) mais violadas, o trabalho usa uma heurística apresentada em (FISCHETTI *et al.*, 1997 apud LABBÉ *et al.*, 2004, p. 183). Essa heurística considera cada componente conexa C de G^* como um identificador de uma restrição *2-matching* generalizada que possivelmente será violada. Os chamados "dentes" de 2 nós de C são as arestas de ciclo que cortam ela, ou seja, $e \in \Gamma(C) : x_e^* = 1$.

As restrições (4.1), por sua vez, podem ser separadas com uma completa enumeração de todas as possíveis triplas $\{i, j, k\}$ em que as variáveis $y_{ij}^* > 0, y_{jk}^* > 0, y_{ki}^* > 0$. O processo de separação para as restrições (4.2) também realiza a mesma enumeração dos trios, mas, além disso, um *min-cut* é executado para apartar a *raiz* de $\{i, j, k\}$.

Com esse último procedimento de separação de restrições o algoritmo exato proposto se completa e encontra-se pronto para ter sua eficácia avaliada. O trabalho segue, portanto, ao seu último ato, testar o algoritmo sobre 3 classes de instâncias e analisar seu desempenho.

4.1.2 Resultados

A primeira classe de instâncias sobre a qual os testes são realizados é definida Classe I, baseada na classe TSPLIB (REINELT, 1991) para o Problema do Caixeiro Viajante, que possui instâncias de 50 a 200 vértices. A segunda é definida como Classe II e construída a partir da geração aleatória de pontos no plano cartesiano, $[0, 1000] \times [0, 1000]$. Essa segunda classe também possui instâncias com tamanhos de 50 a 200 vértices. A terceira e última classe, definida Classe III, é baseada nas instâncias definidas em (LEE *et al.*, 1998 apud LABBÉ *et al.*, 2004, p. 183) que possuem de 10 até 300 vértices e segue o mesmo procedimento de formação utilizado na Classe II, a diferença entre essas duas reside na definição dos custos.

Para a Classe I, os custos são calculados a partir das distâncias l_{ij} definidas originalmente em TSPLIB, essas distâncias são multiplicadas por um fator $\alpha \in \{3, 5, 7, 9\}$, para diferenciar os custos de ciclo e de atribuição da instância sendo construída. Dessa forma, pode-se determinar $c_{ij} = \lceil \alpha \cdot l_{ij} \rceil$, $d_{ij} = \lceil (10 - \alpha) \cdot l_{ij} \rceil$ e $d_{ii} = 0$. A Classe II usa distâncias euclidianas como l_{ij} e calcula os custos da mesma forma que a Classe I. Por fim, a Classe III define os custos de ciclo e atribuição iguais às distâncias euclidianas l_{ij} e os custos d_{ii} como um valor aleatório no intervalo $[0, 1000]$.

Algumas restrições incluídas na fase de separação do algoritmo se mostraram sem eficácia durante os testes. As restrições (4.1) e (4.2) não foram incluídas no modelo antes que todas as restrições de outros tipos fossem totalmente separadas. Assim, as duas foram desconsideradas para as instâncias utilizadas. As restrições (4.3) também não contribuíram na resolução da Classe III, foram, portanto, descartadas nessas instâncias para diminuir o tempo de execução do processo de separação.

Durante os experimentos foi estabelecido limite de 2 horas de execução para cada instância e, dentro desse limite, 96,13% das instâncias nas 3 classes foram resolvidas de forma ótima. A partir desses resultados, foi observada uma dificuldade maior de resolução das instâncias com poucos vértices no ciclo e as instâncias com todos os vértices no ciclo se revelaram mais simples de serem resolvidas. As instâncias mais difíceis foram aquelas em que o ciclo possui cerca de 20% do total de nós do grafo.

O trabalho de Labbé *et al.* (2004) aborda PCMRC com um algoritmo exato que usa de alguns procedimentos para otimizar não só as soluções, mas também as etapas internas do algoritmo. Um desses procedimentos é uma heurística, uma abordagem não exata usada para arredondar soluções fracionárias oriundas das etapas anteriores. A seguir será apresentado um

trabalho que propõe uma série desses procedimentos heurísticos para o problema.

4.2 Algoritmos Heurísticos e Meta-heurísticas Híbridas Aplicadas ao Planejamento de uma Rede de Telecomunicações com Topologia Anel-Estrela

Dias (2006) apresenta um trabalho que utiliza heurísticas e meta-heurísticas de forma bastante ampla para tratar do PCMRC. Ao todo, 6 procedimentos são usados em testes sobre o problema: 3 meta-heurísticas comuns e 3 procedimentos híbridos entre essas meta-heurísticas. A modelagem utilizada é a mesma apresentada em Labbé *et al.* (1999) com o acréscimo de mais uma restrição que foi citada na Seção 3.2. Para comparar e analisar a eficácia desses procedimentos, os resultados são confrontados com os de Pérez *et al.* (2003).

A dificuldade de resolução do problema, explicada na Seção 3.1, faz com que muitos detalhes sobre ele devam ser cuidadosamente selecionados. A escolha de uma estrutura para representar as soluções encontradas é um destes detalhes, pois caso ela se mostre ineficiente para avaliar seu custo, por exemplo, pode prejudicar o tempo final de execução e até a qualidade da solução. Dias (2006) decide usar um vetor em que cada posição i representa o vértice v_i na solução. Da posição 1 a posição m , delimitam-se os vértices no ciclo dessa solução. O restante, $n - m$, são os vértices que serão atribuídos, onde n é o número do total de vértices do grafo da instância.

Definida a estrutura de uma solução, o trabalho utiliza inicialmente as meta-heurísticas *Greedy Randomized Adaptive Search Procedure* (GRASP), Busca Tabu (BT) e *Simulated Annealing* (SA) buscando melhorar a qualidade das soluções de instâncias com até 100 vértices das classes I e II descritas em Labbé *et al.* (2004), abordadas também na Seção anterior. Além disso, tem o objetivo secundário de manter seu tempo de execução próximo daqueles observados no trabalho anterior a ser comparado. Os três procedimentos usam estruturas de vizinhanças de soluções. Elas são obtidas através de *movimentos*, alterações na estrutura original de uma solução. Tais movimentos são: *add*, adicionar um vértice ao ciclo; *remove*, remover um vértice do ciclo; *swap*, uma mistura do movimento *add* e do *remove*; e *2-opt*, uma troca entre pares de arestas no ciclo mantendo sua estrutura cíclica. Todos esses movimentos visam gerar soluções vizinhas melhores que a original ou pelo menos com o menor aumento de custo possível em relação a ela.

Além destas três abordagens básicas, foram desenvolvidas hibridizações entre elas. Inicialmente é feita uma alteração no procedimento BT, sua lista tabu passa a ter tamanho

dinâmico que varia entre 0.5 e 1.5 vezes o tamanho inicial. A primeira hibridização é SABB, SA misturada a BT, onde SA produz uma solução inicial para ser refinada por BT dinâmica. A segunda é GBT, GRASP com BT. Nessa composição, o procedimento GRASP usa BT dinâmica para refinar suas soluções iniciais. A última combinação se dá entre GRASP e a heurística de busca *Generic Variable Neighborhood Search* (GVNS) – GGVNS. GVNS é uma heurística de busca que, nesse caso, usa os movimentos de vizinhança para seu objetivo original e também para gerar perturbações em soluções locais encontradas no refinamento. Com essas perturbações, novas vizinhanças podem ser exploradas e ótimos locais de menor custo encontrados. A composição usa esse procedimento após a fase de busca do próprio GRASP para refinar as soluções fornecidas por este último. Com os 3 procedimentos definidos, a autora segue para comparar os resultados entre os 3 e com o trabalho de Pérez *et al.* (2003).

4.2.1 Resultados

Ao fim, testes foram realizados sobre as instâncias citadas anteriormente e produziram bons resultados. A meta-heurística híbrida com melhor desempenho foi GGVNS, seguida de GBT e por último SABB. Isso demonstra que melhorar o processo de busca do GRASP pode ser uma boa forma de encontrar bons resultados nesse problema. GGVNS teve uma piora considerável nos tempos de execução e a sugestão do trabalho é para diminuir o número de vezes que GVNS é ativado após a fase de busca. Em se tratando dos custos das soluções fornecidas, o melhor procedimento conseguiu encontrar soluções melhores que os trabalhos originais em 20% das instâncias sem ótimo global encontrado. A grande maioria dos outros 80% obteve resultados similares aos resultados dos trabalhos comparados. Para as instâncias com custo ótimo já encontrado, GGVNS alcançou tal valor em mais de 60% das instâncias, superando o trabalho anterior com os quais os resultados foram confrontados.

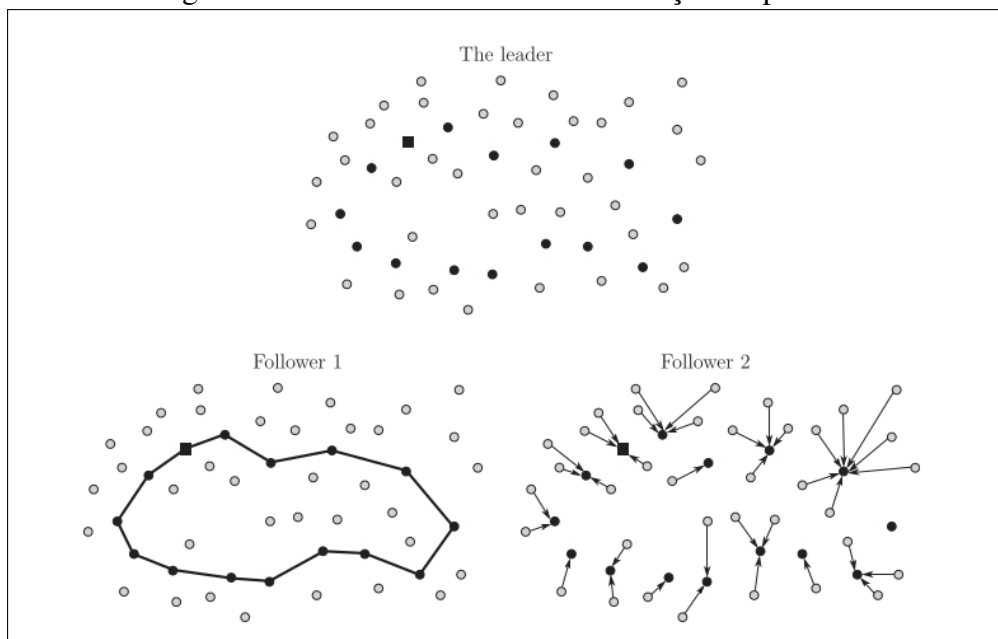
Continuando nessa seara, em seguida será apresentado um trabalho que propõe, além de uma nova abordagem heurística, uma nova formulação para o problema. Calvete *et al.* (2013) desenvolve uma heurística evolucionária e uma modelagem em dois níveis para PCMRC.

4.3 An Efficient Evolutionary Algorithm for the Ring Star Problem

Percebendo a peculiaridade em PCMRC, composto por duas tarefas distintas, a montagem do ciclo e o processo de atribuição, que devem minimizar os custos de seus resultados

em conjunto, Calvete *et al.* (2013) decidem, então, dividir PCMRC em dois níveis: um nível superior, chamado líder, que realiza um processo de separação dos vértices do grafo em dois subconjuntos; e um nível inferior com dois seguidores que resolvem um subproblema cada. O primeiro seguidor soluciona um Problema do Caixeiro Viajante (PCV) sobre o primeiro subconjunto fornecido pelo líder, enquanto o segundo seguidor tem a tarefa de atribuir de forma ótima os vértices do segundo subconjunto aos do primeiro.

Figura 10 – Os dois níveis da reformulação do problema



Fonte: (CALVETE *et al.*, 2013)

O problema deve portanto ser reformulado considerando dois níveis de atuação, o nível do líder e o dos seguidores. E para adequar o novo modelo a essa mudança, uma variável binária é adicionada, $z_i \in \{0, 1\} : \forall i \in V(G)$, onde G é o grafo de uma instância qualquer do problema. Esta variável decide quais vértices estarão no ciclo da solução, tendo valor 1, e quais não estarão, com valor final 0, representando assim o comportamento das variáveis y_{ij} na formulação original.

Cada vértice terá um custo $p_i \in \mathbb{R}_+$ que também será considerado no custo total da solução. Em um problema real significa o custo de se montar uma instalação naquele ponto. O problema ganha, além disso, uma alteração em seu nome, passa a se chamar Problema do Ciclo Mediano sem Restrições de Capacidade de Dois Níveis (PCMRC DN). O modelo dessa reformulação é apresentado a seguir:

$$\min \sum_{i \in V} p_i z_i + \sum_{(i,j) \in E} x_{ij} c_{ij} + \sum_{[i,j] \in A} y_{ij} d_{ij} \quad (4.4)$$

$$\text{sujeito a } z_0 = 1 \quad (4.5)$$

$$z_i \in \{0, 1\}, \quad \forall i \in V \quad (4.6)$$

o problema líder acima gera conjuntos solução $\{z_i\}_{i \in V}$ com os valores de z para cada vértice do grafo. Esses valores são usados nos problemas seguidores $L_1(z)$:

$$\min_x \sum_{(i,j) \in E} x_{ij} c_{ij} \quad (4.7)$$

$$\text{sujeito a } \sum_{(i,j) \in E} x_{ij} = 2z_i, \quad \forall i \in V \quad (4.8)$$

$$\sum_{(i,j) \in E(S)} x_{ij} \leq |S| - 1, \quad \forall S \subseteq \{i \in V \setminus \{0\} : z_i = 1\} \quad (4.9)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in E \quad (4.10)$$

e $L_2(z)$:

$$\min_y \sum_{[i,j] \in A} y_{ij} d_{ij} \quad (4.11)$$

$$\text{sujeito a } \sum_{[i,j] \in A} y_{ij} = 1 - z_i, \quad \forall i \in V \quad (4.12)$$

$$\sum_{[i,j] \in A} y_{ij} \leq (n-2)z_j, \quad \forall j \in V \setminus \{0\} \quad (4.13)$$

$$y_{ij} \in \{0, 1\}, \quad \forall [i,j] \in A \quad (4.14)$$

O problema líder é caracterizado pelas expressões (4.4)-(4.6). A soma (4.4) busca minimizar o custo total do problema assim como a função objetivo (3.1). Já as restrições (4.5) e (4.6) definem, respectivamente, que a *raiz*, estabelecida no trabalho como sendo o vértice 0, deve estar no ciclo e que z deve ser binário.

O líder, então, produz uma solução que consiste em um conjunto $\{z_i\}_{i \in V}$ valorado. Esse conjunto define quais vértices serão utilizados por cada um dos dois problemas seguidores. Estes subproblemas são independentes entre si, ou seja, cada um trata apenas das suas variáveis, sem interferência entre si.

O primeiro problema seguidor, chamado $L_1(z)$, deve solucionar um PCV e é definido pelas expressões (4.7)-(4.10). Este subproblema trata das variáveis x , que possuem o mesmo

intuito da formulação de Labbé *et al.* (1999), apresentada na Seção 3.2. Portanto, (4.7) busca minimizar o custo do ciclo. As restrições (4.8) definem que o grau dos vértices no ciclo deve ser igual a 2, assim como fazem as restrições (3.2). (4.10) estabelece x como binário.

As restrições (4.9) limitam que, dado o conjunto V' que contém os vértices i com $z_i = 1$ e que não inclui a *raiz*, para todo subconjunto próprio de V' , o número de arestas deve ser no máximo o número de vértices do subconjunto diminuído de um. Essa restrição se baseia no fato de que o número de vértices e de arestas de qualquer ciclo deve ser sempre o mesmo. Então, quando a restrição estabelece que nenhum desses subconjuntos poderá ter essa igualdade, é definido então que o único ciclo que poderá existir é aquele que contém todos os vértices com $z_i = 1$, inclusive a *raiz*.

O segundo problema seguidor, chamado $L_2(z)$, é caracterizado pelas expressões (4.11)-(4.14). Esse subproblema tem a tarefa de atribuir os vértices j que têm $z_j = 0$ aos vértices com $z_i = 1$, que estarão no ciclo. De forma similar ao primeiro subproblema, $L_2(z)$ também busca minimizar os custos relacionados às variáveis das quais trata, y . Essa minimização é obtida por meio de (4.11). (4.14) estabelece y como binário.

As restrições (4.12) definem que caso um vértice k tenha $z_k = 1$ a contagem dos nós aos quais está atribuído deve ser 0, ou seja, não deve estar atribuído a nenhum outro vértice. Caso $z_k = 0$, então k deve estar atribuído a um, e apenas um, vértice. Conjuntamente, as restrições (4.13) determinam que um vértice j qualquer poderá ter de 0 a $n - 2$ outros vértices atribuídos a ele. Será 0 no caso em que j não estiver no ciclo, $z_j = 0$, ou seja, nenhum vértice fora do ciclo poderá receber atribuições.

Este é, portanto, o novo modelo proposto ao PCMRC, que se aproveita dessa característica relacionada às tarefas que o problema possui em sua composição. Com isso definido, o trabalho avança para a elaboração de um algoritmo genético que se encaixa nessa nova forma de defini-lo.

4.3.1 **BBEA: Bilevel Based Evolutionary Algorithm**

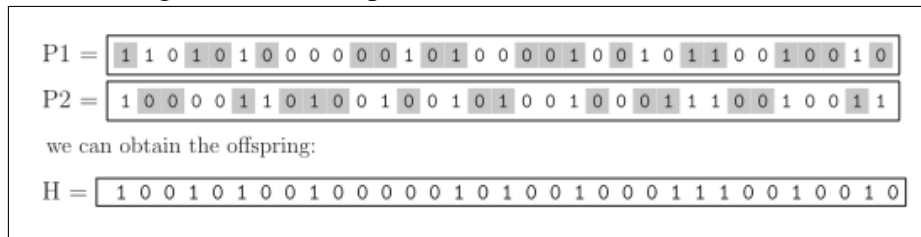
Para tentar fornecer soluções próximas da ótima, uma heurística é apresentada no trabalho: um algoritmo genético chamado *Bilevel Based Evolutionary Algorithm* (BBEA). Algoritmos genéticos codificam as prováveis soluções em cadeias de caracteres passíveis de combinação umas com as outras. Estas cadeias são chamadas *cromossomos*.

Os autores decidem então representar as soluções de PCMRCNDN como uma cadeia

binária de tamanho n , onde n é a quantidade de vértices no grafo da instância do problema. Um exemplo pode ser visto na Figura 11. Cada caractere, ou neste caso *bit*, é chamado *gene* e seu valor *alelo*. Cada gene na posição i define se o vértice i está ou não no ciclo da solução representada.

Essas cadeias são combinadas entre si em um processo chamado *crossover*. Duas delas, denominadas *pais*, tem seus caracteres selecionados para formar uma nova cadeia a partir deles. Esse processo pode ser observado na Figura 11.

Figura 11 – Exemplo de *crossover* entre cromossomos



Fonte: (CALVETE *et al.*, 2013)

Antes de aplicar *crossovers* entre os pais, o algoritmo genético produz, de forma aleatória, uma população inicial de cromossomos e no decorrer da execução divide tal população em grupos menores que trocam elementos entre si. É realizado em seguida um processo de *seleção* baseado no custo de cada solução representada por um cromossomo. Como é a função objetivo que define o custo de uma solução, esta é chamada de *fitness* no procedimento. É imposta uma regra elitista que prioriza a escolha das melhores soluções, ou seja, aqueles que tiverem os menores custos segundo seu *fitness*.

Dois cromossomos pais são finalmente escolhidos e o *crossover* é executado. Cada gene na posição i da nova cadeia, denominada *descendência*, é escolhido de forma probabilística entre os genes presentes na mesma posição em cada um dos pais. Após a descendência ser gerada, uma nova operação é aplicada sobre ela, a *mutação*, que altera o alelo de um gene escolhido aleatoriamente. A última forma de tentativa de melhora das soluções é empregada através de uma busca local, que produz um espaço de soluções vizinhas à descendência recém-gerada. Essa busca é realizada com o objetivo de encontrar cromossomos com melhor *fitness*. Todo esse processo para gerar descendências é repetido segundo um limite preestabelecido e pretende gerar populações melhores a cada nova iteração.

Dado que as cadeias determinam, por meio de seus genes, se determinado vértice estará ou não no ciclo da solução, é natural que sua aplicação deva se dar no problema líder de

BBEA. Com o cromossomo delimitando os subconjuntos que serão repassados aos problemas seguidores, BBEA segue, então, para a fase de resolução desses dois subproblemas. O primeiro seguidor é um PCV, um problema conhecido, para o qual os autores decidem usar apenas um procedimento que mescla um algoritmo guloso para arestas menos custosas e uma busca local simples com vizinhanças *2-opt* e *3-opt*. Fazem isso para não comprometer o tempo de resolução de PCMRCN, pois esse procedimento gera boas soluções em um tempo satisfatório. O segundo seguidor pode ser resolvido em tempo polinomial de forma ótima.

A junção destas 3 fases constitui a proposta de Calvete *et al.* (2013) para fornecer soluções que tenham boa qualidade, ou seja, estejam próximas da solução ótima, mas também gastem um tempo aceitável em sua execução. Esse objetivo é posto à prova e analisado na subseção a seguir.

4.3.2 Resultados

Depois de montar o procedimento para resolver essa nova formulação de PCMRC, o trabalho faz um estudo sobre os parâmetros da execução: tamanho das populações, se terá ou não busca local, entre outros. Com esses valores definidos, o algoritmo é executado sobre uma série de instâncias conhecidas em trabalhos anteriores, muitas delas já detentoras de soluções ótimas descobertas. Os resultados são satisfatórios quando comparados com os trabalhos originais. Em mais de 90% das instâncias, a maioria das quais a melhor solução já havia sido encontrada, o algoritmo apresentado atinge o ótimo em tempo significativamente menor que os outros trabalhos. Para uma das instâncias, sem solução ótima fornecida, o algoritmo obteve um custo melhor do que o registrado até o momento.

O trabalho de Calvete *et al.* (2013), assim como o de Dias (2006), usa procedimentos que não garantem localizar a melhor solução possível para uma instância, mas dadas suas técnicas conseguem montar soluções com qualidade considerável. O trabalho que será estudado a seguir também utiliza uma técnica que não garante encontrar uma solução ótima, mas nesse caso o grau de proximidade para ela é provado matematicamente.

4.4 Algorithms for the Ring Star Problem

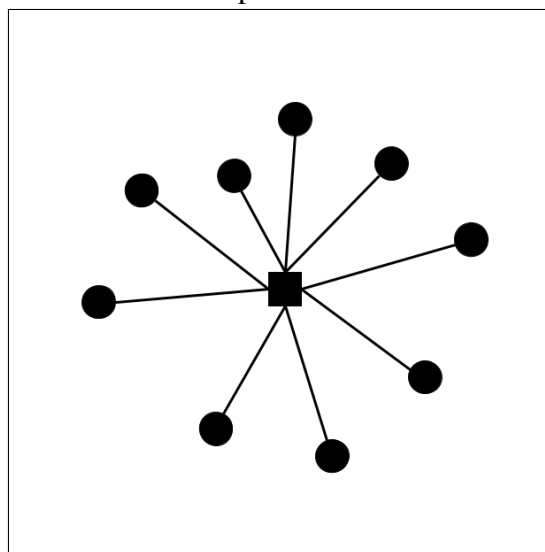
Chen *et al.* (2017) utiliza uma abordagem aproximativa para estudar o PCMRC. Além desse problema, o trabalho trata de outro chamado Problema do Ciclo Mediano Restrito,

que também se origina na topologia anel-estrela. No entanto, difere de PCMRC, pois possui uma restrição de capacidade: todo vértice no ciclo pode ter, no máximo, um valor fixo k de vértices atribuídos a ele. Como esse problema não é o foco do presente trabalho, este texto considera apenas as proposições e resultados relacionados ao PCMRC.

No trabalho de Chen *et al.* (2017) são feitas alterações em algumas características de PCMRC. É usado, por exemplo, apenas um grafo simples, $G = (V, E)$, sem arcos direcionados, isto é, a estrutura deixa de ser mista. Com isso, os custos de atribuição são postos também nas arestas. Isso altera, portanto, a possibilidade de que dois vértices possam ter custos de atribuição diferentes dependendo do sentido dos arcos, $[i, j]$ ou $[j, i]$, entre eles. Assim, as arestas passam a ter os dois tipos de custo, de ciclo e de atribuição.

Para além de alterações estruturais, na maior parte do trabalho um subproblema de PCMRC é abordado: o PCMRC métrico. Este estabelece uma proporção $M \geq 1$ entre os custos de ciclo, definidos pela função $r : E \rightarrow \mathbb{R}_+$, e o custo de atribuição de cada aresta, $a : E \rightarrow \mathbb{R}_+$. M é definido como r dividido por a , $M = \frac{r(e)}{a(e)}$ para $e \in E$, e assume-se que os valores para r e a satisfazem as desigualdades triangulares. Esse subproblema é usado para provar a primeira aproximação proposta. Mais uma diferença importante para a forma original do problema é o fato de que, para provar essa primeira aproximação, PCMRC métrico aceita uma solução degenerada além daquelas com organização já conhecida: uma solução estrela em que o único vértice no "ciclo" é a raiz.

Figura 12 – Exemplo de solução estrela aceita por PCMRC métrico



Fonte: Próprio autor

Nas características restantes o problema se mantém igual. O grafo continua sendo completo e os dois tipos de custos, r e a , são não negativos. Uma instância do problema é denotada por (G, r, a, M) , onde G é o grafo, r e a as funções de custo e M a proporção fixada previamente entre os custos. Uma solução para essa forma do problema terá um ciclo, ou apenas a *raiz*, e algumas, ou nenhuma, folhas ligadas ao ciclo. Assim como antes, nenhum vértice pode estar desconectado em uma solução. Seu custo é calculado como no original, mas se adaptando à nova definição dos custos de atribuição.

Inicialmente as conclusões matemáticas são tomadas sobre PCMRC métrico para que então haja um fechamento com uma aproximação relacionada ao próprio PCMRC. A seguir são apresentadas e esclarecidas as aproximações propostas por Chen *et al.* (2017).

4.4.1 Aproximações

Definida a configuração do problema, Chen *et al.* (2017) propõe o Teorema 1 abaixo. \mathcal{T} é definido como a solução em forma de estrela que tem apenas a *raiz* no ciclo e $n = |V(G)|$.

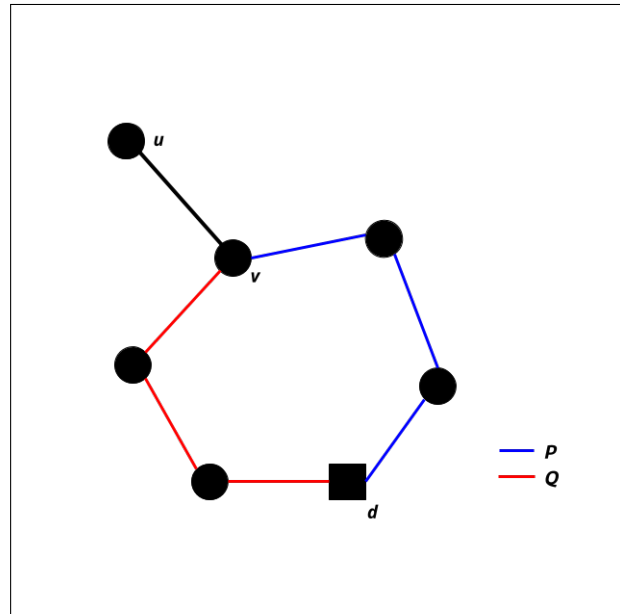
Teorema 1. *Se $M \geq \frac{n-1}{2}$, então \mathcal{T} é uma solução ótima para a instância (G, r, a, M) do PCMRC métrico.*

Demonstração. Seja $S = R \oplus A$ uma solução qualquer para a instância (G, r, a, M) . Pode-se definir S como a junção do conjunto de arestas que compõem o ciclo, R , com o conjunto de arestas que representam as atribuições, A . Para todo vértice $u \in V \setminus V(R)$, ou seja, que não está no ciclo, temos uma única aresta $(u, v) \in A$ que conecta u ao ciclo por meio de $v \in V$. São considerados também os caminhos internamente disjuntos P e Q que, quando $v \neq d$, ligam v à raiz d no ciclo como mostra a Figura 13. Além disso, é definida a função a para conjuntos de arestas. Formalmente, dado o conjunto E' de arestas, $a(E') = \sum_{(i,j) \in E'} a(i, j)$. As desigualdades triangulares permitem estabelecer:

$$a(u, d) \leq \frac{1}{2}(a(u, v) + a(P)) + \frac{1}{2}(a(u, v) + a(Q)) = a(u, v) + \frac{1}{2}a(R) \quad (4.15)$$

Essas desigualdades triangulares determinam que um lado de qualquer triângulo deve ser menor ou igual a soma dos outros dois lados do mesmo. Tendo em conta que os custos de atribuição são todos não negativos, as somas destes custos para três caminhos diferentes podem representar lados de um triângulo.

Figura 13 – Exemplos de caminhos P e Q em um ciclo



Fonte: Próprio autor

O primeiro caminho é formado pela ligação direta entre u e d , que não necessariamente faz parte de S , mas será usado para alcançar o objetivo da demonstração. É garantida a existência dessa ligação pois, como mencionado anteriormente, o grafo é completo. O segundo caminho é representado pelo passeio de u a d por meio de P enquanto que o terceiro pelo passeio de u a d utilizando Q .

A consequência matemática é de que $a(u, d)$ deve ser limitado superiormente por $a(u, v) + \frac{1}{2} a(R)$. Esse resultado é obtido ao se somar dois meios de $a(u, v)$ e, sendo a união de P e Q igual a R , como pode ser observado no exemplo da Figura 13, a soma de seus custos de atribuição resulta no custo de atribuição de R .

Essas desigualdades tratam apenas dos vértices que são atribuídos ao ciclo da solução S . Com respeito aos vértices do ciclo R em si, o resultado é $a(v, d) \leq \frac{1}{2} a(R)$ para todo $v \in V(R) \setminus \{d\}$. Elimina-se o custo da atribuição de v ao ciclo em comparação com a inequação (4.15), pois não há aresta para fazer essa ligação uma vez que o vértice já pertence ao ciclo.

Com as desigualdades definidas dessa forma, a soma dos lados esquerdos de cada uma resulta em algo bastante próximo do custo de \mathcal{T} . Como o lado esquerdo considera todas as ligações diretas a d para formar um dos lados da inequação (4.15), somar todos os custos de atribuição dessas ligações resulta no custo de \mathcal{T} . No entanto, para manter a relação de desigualdade triangular, é necessário remover dessa soma os custos referentes às atribuições $(u, d) \in A$, ou seja, as ligações (u, v) definidas como atribuições em S que já são diretas à raiz,

pois não permitem a existência, em R , dos caminhos P e Q , uma vez que v já é a raiz d . Dessa forma, $E(\mathcal{T}) \setminus A$ resultam em um conjunto com as atribuições diretas à raiz não presentes em S . E o contrário, $A \setminus E(\mathcal{T})$, representa as atribuições de S que não são feitas diretamente à raiz.

$$a(E(\mathcal{T}) \setminus A) \leq a(A \setminus E(\mathcal{T})) + \frac{|A \setminus E(\mathcal{T})| + |V(R) \setminus \{d\}|}{2} \cdot a(R) \quad (4.16)$$

Condensado o lado esquerdo das inequações (4.15) referentes a cada vértice de $V(G) \setminus \{d\}$, considerando as exceções mencionadas, deve-se então somar o lado direito delas. O somatório de $a(u, v)$ resulta em $a(A \setminus E(\mathcal{T}))$ pois apenas os vértices que são atribuídos ao ciclo possuem esse termo do lado direito. E são retiradas as atribuições diretas a d por meio da subtração das atribuições de \mathcal{T} que possui todas as atribuições diretas a d possíveis. Essa retirada tem a mesma motivação anterior com os papéis dos conjuntos invertidos.

O segundo termo do lado direito das inequações (4.15) está relacionado a $\frac{1}{2}a(R)$. É trivial perceber que, como $a(R)$ continua intacto, $\frac{|A \setminus E(\mathcal{T})| + |V(R) \setminus \{d\}|}{2}$ conta quantas vezes $\frac{1}{2}a(R)$ foi somado. Esse valor é obtido pela quantidade de arestas em $A \setminus E(\mathcal{T})$, que já foi explicado, somada com a quantidade de vértices em $V(R) \setminus \{d\}$, que define o número de inequações (4.15) para vértices do ciclo, também consideradas nessa inequação (4.16). O valor máximo para $|A \setminus E(\mathcal{T})| + |V(R) \setminus \{d\}|$ pode ser observado na situação em que A não possua nenhuma ligação direta a d , assim nenhuma aresta será removida desse conjunto e a soma da quantidade de atribuições com a quantidade de nós no ciclo, exceto a raiz d , seria igual a $n - 1$, delimitando nova inequação derivada:

$$a(E(\mathcal{T}) \setminus A) \leq a(A \setminus E(\mathcal{T})) + \frac{n-1}{2} \cdot a(R) \quad (4.17)$$

com esse resultado envolvendo \mathcal{T} , é possível fazer as afirmações a seguir:

$$c(\mathcal{T}) = a(\mathcal{T}) \leq a(A) + \frac{n-1}{2} \cdot a(R) \leq a(A) + M \cdot a(R) = a(A) + r(R) = c(S). \quad (4.18)$$

Considerando o cenário de maximização, o custo total de \mathcal{T} é igual ao seu custo de atribuição apenas, porque há somente atribuições nessa solução. Esse custo, passando pelo que definiu a inequação (4.17), é menor ou igual ao custo de atribuição em A somado a M vezes o custo de atribuição em R . É menor pois o teorema considera M maior ou igual a $\frac{n-1}{2}$, então qualquer valor assumido por M no escopo do teorema fará com que $a(A) + M \cdot a(R)$ seja maior ou pelo menos igual a $a(A) + \frac{n-1}{2} \cdot a(R)$. Consequentemente, sendo M uma proporção entre r e a , com $M = \frac{r(e)}{a(e)}$ para $e \in E(G)$, $M \cdot a(R)$ é igual a $r(R)$. Por fim, a soma de $a(A)$ com $r(R)$ resulta no custo total de S , a solução qualquer considerada no início da demonstração. Tem-se

então que $c(\mathcal{T}) \leq c(S)$, como S pode ser qualquer solução, logo \mathcal{T} é uma das, senão a única, soluções com menor custo para (G, r, a, M) , tornando-a uma solução ótima dessa instância. \square

Adicionalmente os autores derivam um corolário a partir do Teorema 1. Ele define que, para $M \leq \frac{n-1}{2}$, \mathcal{T} também pode ser usado como uma aproximação da ordem de $\frac{n-1}{2M}$ para solução ótima de uma instância. Sejam opt e OPT , respectivamente, a solução ótima e custo desta solução ótima de uma instância genérica (G, r, a, M) .

Esse corolário deve corroborar o Teorema 1 para $M = \frac{n-1}{2}$, ou seja, tendo M esse valor, a solução estrela deve ser ótima. Ao substituir M em $\frac{n-1}{2M}$ por $\frac{n-1}{2}$ obtém-se como resultado da divisão o valor 1, fazendo a inequação se tornar $c(\mathcal{T}) \leq OPT$. Como nenhum custo pode ser melhor que OPT , então \mathcal{T} é a solução ótima ou pelo menos uma delas. O corolário é definido a seguir.

Corolário 1. Se $M \leq \frac{n-1}{2}$, então $c(\mathcal{T}) \leq \frac{n-1}{2M} \cdot OPT$.

Demonstração. Com base em (G, r, a, M) é construída uma nova instância onde os custos de todas as arestas de ciclo são multiplicados por $\frac{n-1}{2M}$. Sendo assim, define-se $r'(u, v) = \frac{n-1}{2M} \cdot r(u, v)$, onde $(u, v) \in E(G)$. A nova instância é denotada por (G, r', a) e o custo de uma solução S qualquer para ela é representado por $c'(S)$. O seguinte resultado pode ser posto:

$$c(\mathcal{T}) \leq c'(\mathcal{T}) \leq c'(opt) \leq \frac{n-1}{2M} \cdot c(opt) \quad (4.19)$$

A primeira parte de (4.19) determina que o custo de \mathcal{T} para a instância original é menor ou igual ao custo de \mathcal{T} para a nova instância. Como \mathcal{T} tem apenas arestas de atribuição em sua estrutura, o seu custo total não é alterado pela multiplicação feita para obter (G, r', a) . Para produzir (G, r', a) , multiplicou-se cada $r(e)$ por $\frac{n-1}{2M}$, gerando então um cálculo diferente para definir r' , $r'(e) = \frac{n-1}{2M} \cdot M \cdot a(e)$. Com desenvolvimento adequado dessa equação chega-se a $r(e) = \frac{n-1}{2} \cdot a(e)$.

O Teorema 1 define que se $M \geq \frac{n-1}{2}$, então \mathcal{T} é solução ótima. Se todo custo de ciclo é calculado pela multiplicação de $\frac{n-1}{2}$ por $a(e)$, então existe M , com o valor $\frac{n-1}{2}$, para (G, r', a) e ele se encaixa no conjunto de valores que M pode assumir no Teorema 1. Sendo assim, $c'(\mathcal{T})$ para (G, r', a) deve ser menor ou igual ao custo de uma solução ótima dessa instância, o que justifica a segunda inequação em (4.19).

Por fim, $c'(opt)$ deve ser menor ou igual a $\frac{n-1}{2M} \cdot c(opt)$. Para calcular c e c' , são somados os custos adequados das arestas, escolhidas na solução, para compor ciclo ou para

estabelecer as atribuições. Desta forma, multiplicar $c(opt)$ por $\frac{n-1}{2M}$ é, na verdade, multiplicar cada um dos custos que compõem esse total. Isso se torna importante pois $c'(opt)$ tem apenas parte dos custos de (G, r, a, M) alterados quando da construção de (G, r', a) . Isto posto, em $\frac{n-1}{2M} \cdot c(opt)$ todos os custos de $c(opt)$ são multiplicados por esse valor, enquanto que em $c'(opt)$ apenas os custos de ciclo sofrem alteração, portanto conclui-se que $c'(opt)$ é realmente menor ou igual a $\frac{n-1}{2M} \cdot c(opt)$. Com isso, para $M \leq \frac{n-1}{2}$, tem-se que $c(\mathcal{T}) \leq \frac{n-1}{2M} \cdot c(opt)$, o que faz de \mathcal{T} uma $\frac{n-1}{2M}$ -aproximação para a solução ótima de (G, r, a, M) . \square

Estabelecidos o Teorema 1 e o Corolário 1, os autores prosseguem para o estudo do impacto desses resultados sobre o PCMRC genérico. Utilizando um algoritmo 2-aproximativo para o Problema do Caixeiro Viajante (VAZIRANI, 2013 apud CHEN *et al.*, 2017, p. 6), que produz um ciclo hamiltoniano baseando-se numa Árvore Geradora Mínima (AGM), é possível encontrar um ciclo \mathcal{C} para G usando como peso das arestas os custos $r(\cdot)$ de uma instância qualquer (G, r, a) de PCMRC métrico. Esse ciclo respeitará a restrição $r(\mathcal{C}) \leq 2r(T)$, onde T é qualquer árvore em G . O algoritmo 2-aproximativo produz Árvores Geradoras de no máximo duas vezes o custo de qualquer AGM de G . Dessa forma, o custo do ciclo hamiltoniano \mathcal{C} será sempre menor que duas vezes o custo de qualquer árvore em G , pois se é menor que duas vezes o custo de uma árvore geradora de custo mínimo, será menor que duas vezes o custo de qualquer outra.

Este ciclo \mathcal{C} se torna importante para PCMRC métrico, pois também pode ser usado como uma aproximação de opt para instâncias (G, r, a, M) quaisquer. Ao remover uma aresta de ciclo em opt , obtém-se uma árvore geradora T' de G uma vez que, sem ciclo algum e continuando conexo, o que é o caso, qualquer grafo é árvore geradora. Com isso, é possível afirmar que $r(T') \leq r(opt) \leq M \cdot c(opt)$. O custo de ciclo de T' é menor ou igual ao custo de ciclo de opt , pois foi eliminada uma aresta de ciclo de opt para se obter T' . A segunda inequação é válida, pois ao multiplicar $c(opt)$ por M , são multiplicados de fato cada custo de aresta, seja de ciclo ou de atribuição, que compõem o custo total. Relembrando que M é uma proporção entre r e a , onde $r(e) = M \cdot a(e)$ para $e \in E(G)$, é certo que os custos de atribuição de opt se transformam em custos de ciclo para ela ao serem multiplicados por esse M , o que transformaria $c(opt)$ em $r(opt)$. No entanto, os custos de ciclo também são multiplicados, fazendo de $M \cdot c(opt)$ maior que $r(opt)$, ou no mínimo igual a $r(opt)$ nos casos em que M assumir valor 1.

Unindo as observações dos dois parágrafos anteriores, chega-se a seguinte conclusão: $c(\mathcal{C}) \leq 2M \cdot OPT$. A primeira observação traz $c(\mathcal{C}) \leq 2r(T)$, sendo \mathcal{C} um ciclo, $c(\mathcal{C}) = r(\mathcal{C})$.

Essa desigualdade permite usar a segunda observação para estabelecer $c(\mathcal{C}) \leq 2r(T) \leq 2r(opt) \leq 2M \cdot c(opt) = 2M \cdot OPT$.

Com o Teorema 1, o Corolário 1 e a observação do parágrafo anterior foi possível aos autores demonstrarem um novo fato sobre o próprio PCMRC. O Corolário a seguir define uma aproximação sobre o problema genérico utilizando os intervalos de M para defini-la.

Corolário 2. *PCMRC pode ser aproximado em $\sqrt{n-1}$ vezes em um tempo de $O(n^2)$.*

Demonstração. Como solução S para uma instância (G, r, a, M) , escolha \mathcal{T} ou \mathcal{C} , decidindo por aquela que tiver menor custo. A seguinte conclusão surge quando combinados o Teorema 1, o Corolário 1 e a observação do último parágrafo:

$$c(S) \leq \min \left\{ \max \left\{ 1, \frac{n-1}{2M} \right\}, 2M \right\} \cdot OPT \leq \sqrt{n-1} \cdot OPT \quad (4.20)$$

$c(S)$ é menor ou igual ao resultado da função de mínimo vezes OPT . Como S é uma escolha entre \mathcal{T} e \mathcal{C} , se \mathcal{T} tiver o menor custo entre essas duas estruturas e, segundo o Teorema 1, $M \geq \frac{n-1}{2}$, \mathcal{T} é a solução ótima de (G, r, a, M) . Isso coloca a opção de valor 1 na função de máximo, pois caso essas condições se cumpram, a função selecionaria 1 em detrimento de $\frac{n-1}{2M}$, que assumiria valores menores que 1, e a função de mínimo escolheria 1 também pois esses valores para M fazem de $2 \cdot M$ maior que 1 em qualquer circunstância em que $n \geq 3$. Com isso, $c(\mathcal{T})$ seria menor que $1 \cdot OPT$.

M não assumindo valores nesse intervalo, o Corolário 1 define que \mathcal{T} é pelo menos uma boa aproximação da ordem de $\frac{n-1}{2M}$, a segunda opção da função de máximo. Caso esse valor seja escolhido, o que acontecerá apenas caso $M \leq \frac{n-1}{2}$, produziria o mesmo resultado do corolário, $c(\mathcal{T}) \leq \frac{n-1}{2M} \cdot OPT$.

No caso em que \mathcal{C} tivesse o menor custo, a função de mínimo sempre escolheria $2 \cdot M$, pois para que \mathcal{C} , com custo definido apenas pelo ciclo, tenha custo menor que \mathcal{T} , é necessário que M tenha valor 1, sendo assim, a função de máximo escolhe sua segunda opção e a função de mínimo escolheria $2 \cdot M$.

Com relação à segunda parte em (4.20), para mostrar que qualquer valor escolhido pela função de mínimo que multiplica OPT será sempre menor ou igual a $\sqrt{n-1}$, é preciso observar os valores assumidos por M em relação n . Para facilitar essa argumentação, será útil escolher um marco para dividir intervalos de valores para M com comportamentos iguais. Esse marco será mais uma vez $\frac{n-1}{2}$. Serão trabalhados portanto os intervalos a seguir:

$$\left\{ \begin{array}{l} 1 \leq M < \frac{n-1}{2}, \\ M = \frac{n-1}{2}, \\ M > \frac{n-1}{2} \end{array} \right.$$

É possível também terceirizar essa divisão de intervalos sem usar diretamente M . Com uma variável x multiplicando $\frac{n-1}{2}$, pode-se determinar os intervalos anteriores apenas pelas grandezas assumidas por essa nova variável.

$$\left\{ \begin{array}{l} \text{se } 1 \leq M < \frac{n-1}{2}, \text{ então } \frac{2}{n-1} \leq x < 1 \\ \text{se } M = \frac{n-1}{2}, \text{ então } x = 1 \\ \text{se } M > \frac{n-1}{2}, \text{ então } x > 1 \end{array} \right.$$

Com x , M é representado usando apenas $x \cdot \frac{n-1}{2}$. Quando x for menor que 1, M assume valores menores que $\frac{n-1}{2}$ e assim por diante. É preciso definir $\frac{2}{n-1} \leq x$, pois M só pode apresentar valores maiores ou iguais a 1. Desenvolvendo $x \cdot \frac{n-1}{2} \geq 1$, tem-se o resultado lógico de que o valor mínimo de x será $\frac{2}{n-1}$. A seguinte configuração surge quando substitui-se M por $x \cdot \frac{n-1}{2}$ nas funções de máximo e mínimo:

$$c(S) \leq \min \left\{ \max \left\{ 1, \frac{1}{x} \right\}, x \cdot (n-1) \right\} \cdot \text{OPT} \leq \sqrt{n-1} \cdot \text{OPT} \quad (4.21)$$

Para $x = 1$, a função de máximo escolhe sempre 1 pois os dois parâmetros são iguais. A função de mínimo escolhe 1 quando considera-se que o grafo deve ter pelo menos 3 vértices para que alguma decisão entre ciclo ou estrela seja tomada. Com $x > 1$, a função de máximo selecionará 1, pois $\frac{1}{x}$ será sempre menor que 1. Entre 1 e $x \cdot (n-1)$, a função de mínimo escolhe a primeira opção, pois $x > 1$ só aumentará o resultado da multiplicação e o mínimo que ela pode resultar é em $x \cdot 2$ já que considera-se $n \geq 3$.

Em relação ao último intervalo, $\frac{2}{n-1} \leq x < 1$, a função de máximo sempre escolherá o resultado de $\frac{1}{x}$, pois, sendo x menor que 1, o resultado dessa divisão resultará em algo maior que 1. Após isso, a função de mínimo deve escolher entre $\frac{1}{x}$ e $x \cdot (n-1)$. Uma pequena análise torna-se necessária. Caso x apresente, por exemplo, o valor 0,5, a fração $\frac{1}{x}$ resultará em 2 e $x \cdot (n-1)$ em no mínimo 1. A função de mínimo escolheria então $x \cdot (n-1)$. No entanto, caso x tenha valor 0,8, $\frac{1}{x}$ seria igual a 1,25, enquanto $x \cdot (n-1)$ assumiria o valor 1,6 no mínimo. A escolha seria inversa.

Esses exemplos mostram que existe um ponto, no intervalo considerado para x , em que $\frac{1}{x}$ e $x \cdot (n-1)$ deverão ter valores iguais variando junto com o valor de n . Ao igualar as duas expressões obtém-se $x = \frac{1}{\sqrt{n-1}}$. Então, para $\frac{2}{n-1} \leq x < \frac{1}{\sqrt{n-1}}$, o valor escolhido será sempre o resultado de $x \cdot (n-1)$, em $x = \frac{1}{\sqrt{n-1}}$ a escolha pode ser qualquer um dos dois pois se igualam, e para $\frac{1}{\sqrt{n-1}} < x < 1$ a escolha será sempre $\frac{1}{x}$.

Ao substituir x em $x \cdot (n-1)$ por $\frac{1}{\sqrt{n-1}}$, o resultado é $\sqrt{n-1}$. Isto é, quando $x \cdot (n-1)$ é escolhido pela função de mínimo, o valor máximo que a multiplicação resulta é $\sqrt{n-1}$, respeitando a inequação (4.21). Substituindo também em $\frac{1}{x}$, o resultado é o mesmo, entretanto, dessa vez, à medida que x decresce $\frac{1}{x}$ cresce, e o máximo valor que a divisão pode assumir é exatamente $\sqrt{n-1}$.

Com isso, demonstra-se que pra qualquer valor escolhido pelas funções de máximo e de mínimo que multiplique OPT, este será sempre menor ou igual a $\sqrt{n-1}$. Portanto, qualquer que seja a escolha para a solução S , está representará uma aproximação dessa ordem para a solução ótima da instância. Com isso, todo valor possível de M permite que \mathcal{T} e \mathcal{C} sejam aproximações da ordem de $\sqrt{n-1}$ para o PCMRC métrico. E para generalizar essa aproximação sobre PCMRC genérico basta eliminar a barreira de valores assumidos por M , e conseqüentemente por x , permitindo $M > 0$, para que instâncias com custos de atribuição maiores que custos de ciclo sejam consideradas. \square

As estruturas propostas acima como solução não se limitam a serem utilizadas apenas por algoritmos aproximativos. Elas podem, por exemplo, ser usadas em algum procedimento heurístico que possui nitidamente pontos de semelhança com a técnica aproximativa. Chen *et al.* (2017) propõem a seguir alguns desses procedimentos.

4.4.2 Heurísticas

Além desse estudo sob um contexto aproximativo demonstrado na subseção anterior, o trabalho prossegue apresentando 3 heurísticas para PCMRC. A primeira, chamada H_B utiliza os resultados desse estudo para produzir uma solução para o problema. O resultado final fornecido é escolhido entre três opções: a solução estrela \mathcal{T} e a solução em ciclo hamiltoniano \mathcal{C} propostas anteriormente, e uma solução baseada na Árvore Geradora Mínima (AGM) dos grafos das instâncias do problema. Esta última utiliza os custos de atribuição como peso das arestas e, após definir a AGM da instância a ser resolvida, transforma os vértices que não são folhas na AGM em um ciclo, que tem seu custo otimizado pelo algoritmo apresentado em Croes (1958), que

produz boas soluções para o Problema do Caixeiro Viajante (PCV) em tempo polinomial. Desse modo, a estrutura passa de árvore para anel-estrela já que as folhas tornam-se automaticamente vértices atribuídos ao ciclo. A solução escolhida será aquela que obtiver o menor custo total.

Uma segunda heurística, denominada H_{LS} , executa uma busca local em um conjunto vizinhança de uma solução de PCMRC. Essa primeira solução é obtida através de uma escolha aleatória de n vértices que formarão o ciclo central. As vizinhas das soluções que passam pela busca local são definidas com a remoção, também aleatória, de um vértice do ciclo que passa a ser um vértice atribuído. Os nós do ciclo adjacentes aquele que foi removido são conectados diretamente para que a estrutura volte a ser um ciclo fechado. Uma solução vizinha com um custo de melhor valor é buscada até um limite de repetições. Por fim o ciclo é otimizado pelo mesmo algoritmo de Croes (1958) visto no parágrafo anterior.

A terceira e última heurística apoia-se em observações experimentais para valores muito grandes de M (proporção entre os custos de ciclo e de atribuição). No procedimento proposto, intitulado H_{CT} , é realizada uma construção gulosa do ciclo central. Inicialmente um vértice qualquer é adicionado ao ciclo. Essa adição inicial é testada para todos os vértices em um ciclo inicial contendo apenas a *raiz*. Em seguida, os vértices que possuem a menor distância (euclidiana) para algum nó já no ciclo são adicionados em sequência, mas apenas caso tal ação implique em diminuição no custo da melhor solução já encontrada. É possível haver melhora pois a solução inicial na execução é a estrela \mathcal{T} . Essas adições posteriores são executadas até que todos os vértices tenham sido testados em busca de melhorias.

Com esses três algoritmos, o trabalho consegue, além de demonstrar aproximações para o problema, ultrapassar a barreira da vertente de otimização e juntar as soluções que desenvolveram técnicas externas aos algoritmos aproximativos. Em seguida, os autores realizam testes para demonstrar a qualidade do que produziram neste trabalho.

4.4.3 Resultados

Por fim, são executados testes sobre instâncias de até 50 nós com os custos baseados nas distâncias euclidianas entres eles. A diferença entre os custos de ciclo e de atribuição é dada pela proporção M citada anteriormente. Os testes compreendem valores para M de 1 a 13. Os resultados destes testes demonstraram que os custos das soluções propostas eram menos de 2 vezes o custo ótimo. Essas soluções foram apresentadas ao longo da Subseção 4.4.1. Com relação às heurísticas, os resultados apresentaram valores próximos do ótimo para instâncias

também de 50 nós presentes em Reinelt (1991).

Com isso, a contribuição do trabalho de Chen *et al.* (2017) consiste na demonstração de aproximações para uma variação do PCMRC, em uma aproximação da ordem de $\sqrt{n-1}$ para o próprio PCMRC e na apresentação de 3 heurísticas para o problema.

4.5 Síntese dos Trabalhos Correlatos

Como explicado na introdução do capítulo, os quatro trabalhos detalhados no decorrer das seções englobam três dimensões importantes da OC. Labbé *et al.* (2004), que representa a vertente exata da área, consolida o modelo original de PCMRC e propõe o primeiro algoritmo exato para o problema, uma abordagem *Branch-and-Cut* que demonstra boa performance. No entanto, o trabalho não divulga as soluções ótimas encontradas para as instâncias novas resolvidas. Apesar de sua vertente não ter tal foco, dado que estas soluções são garantidamente ótimas, estes resultados seriam de grande importância a outras das linhas da OC.

Uma destas linhas é a heurística, uma parte da Otimização com foco em soluções não exatas. Dois trabalhos com abordagens desta natureza foram detalhados nas Seções 4.2 e 4.3. Dias (2006) desenvolve seis procedimentos heurísticos para o problema, dentre eles três hibridizações, e os compara entre si e com um trabalho anterior de caráter semelhante. As três hibridizações se destacaram nos experimentos computacionais. Uma delas, GGVNS, detém o melhor desempenho heurístico com relação à qualidade das soluções encontradas. O segundo trabalho nesta linha é Calvete *et al.* (2013) que apresenta uma nova modelagem em dois níveis e um algoritmo genético para PCMRC. O algoritmo, uma heurística evolucionária, isto é, que se baseia em processos biológicos para definir seu comportamento, tem resultados bastante promissores com relação ao tempo decorrido para encontrar as melhores soluções conhecidas até então, uma espécie de complemento aos bons resultados de Dias (2006).

Outra linha que foca em soluções não exatas é a aproximativa, que é representada neste capítulo pelo trabalho de Chen *et al.* (2017). As soluções produzidas neste trabalho, e principalmente as aproximações, acabam por corroborar alguns comportamentos das instâncias que são comumente utilizadas para demonstrar a eficácia das abordagens propostas. Dado que, em algumas das classes de instâncias, os valores base dos custos de ciclo e atribuição são distâncias euclidianas entre os vértices, a utilização de fatores para diferenciar esses custos estabelece a proporção fixa descrita em Chen *et al.* (2017). Então, essas instâncias acabam se comportando como apontado no trabalho.

5 QUASI-EXATO

Como visto no capítulo 4, o PCMRC foi abordado de diferentes formas, algumas delas utilizam algoritmos exatos para resolver o problema e localizar a solução com melhor custo. Outras abordagens fazem uso de procedimentos heurísticos que optam por não garantir o encontro de uma solução ótima para diminuir consideravelmente o tempo gasto na busca por soluções viáveis ao problema.

Desta forma, o presente trabalho adota um método exato em conjunto a dois métodos heurísticos. O método exato consiste em um *Branch-and-Bound* (B&B) com sua execução limitada por um GAP percentual. A primeira heurística é denominada GRASP-ILS, uma composição dos métodos, *Greedy Randomized Adaptive Search Procedure* e *Iterated Local Search*. E, por fim, o método GGVNS que consiste da união do GRASP com a heurística de busca *Generic Variable Neighborhood Search* (GVNS).

5.1 *Branch-and-Bound*

O método exato de resolução *Branch-and-Bound* (B&B) (LAND; DOIG, 1960) busca a solução ótima para um problema de programação inteira, ou de programação inteira linear mista, executando uma enumeração inteligente das soluções viáveis do mesmo, tentando se afastar da necessidade de enumerar todas as possíveis candidatas. Papadimitriou e Steiglitz (1998) o descrevem como uma tentativa de provar que a solução encontrada é ótima baseando tal prova no particionamento sucessivo do espaço de soluções. Seu nome representa as duas técnicas que o método aplica a este espaço durante sua execução.

Branch é o processo de ramificação feito no espaço de soluções. O B&B aborda o problema inteiro P , por exemplo, considerando suas variáveis como contínuas, isto é, não obrigando que seu valor final seja inteiro. Este recurso é chamado de *relaxação*. O novo problema P^0 é resolvido como sendo de programação linear e, no geral, algumas de suas variáveis, ou muitas, serão fracionárias. A ramificação é feita, portanto, criando dois subproblemas que terão uma nova restrição cada. A base dessa nova restrição deve ser uma das variáveis com valor final não inteiro em P^0 . (PAPADIMITRIOU; STEIGLITZ, 1998)

Existem algumas técnicas para escolha da variável que realizará a divisão do problema, tais quais a Variante de Dakin (DAKIN, 1965 apud GOLDBARG; LUNA, 2005, p.184), de Land e Doig (LAND; DOIG, 1960 apud GOLDBARG; LUNA, 2005, p. 184), de Spielberg

(GUIGNARD; SPIELBERG, 1968 apud GOLDBARG; LUNA, 2005, p. 184), entre outras. Escolhida a variável x_i com valor fracionário x_i^0 , dois subproblemas de P^0 , denominados P^1 e P^2 , são criados com toda a sua configuração (restrições) e o acréscimo da restrição citada anteriormente. Para P^1 a restrição será $x_i \leq \lfloor x_i^0 \rfloor$. Para P^2 a restrição será $x_i \geq \lceil x_i^0 \rceil$. Isso é feito pois, sendo x_i fracionária, a solução ótima para P não pode conter valores dessa natureza, então seu valor só pode estar no piso, ou antes dele, ou no teto, ou após este. O processo descrito é repetido em cada subproblema resolvido, montando, assim, uma árvore de busca, em que cada nó possui dois nós filhos, que são seus subproblemas. (PAPADIMITRIOU; STEIGLITZ, 1998)

A garantia de que o procedimento não se ramificará indefinidamente reside no fato de que caso P tenha uma região limitada, finita, de soluções viáveis, então as restrições que vão sendo acumuladas em cada subproblema, em algum momento, devem obrigar a resolução contínua a escolher apenas variáveis inteiras. Neste momento, tal nó não mais se ramificará, tornando-se uma folha da árvore de busca. Quando não houver mais ramificações, a folha com menor custo será a solução ótima para o problema original P . (PAPADIMITRIOU; STEIGLITZ, 1998)

Bound, por sua vez, representa o uso dos valores das soluções finais, encontradas nas resoluções lineares dos subproblemas, como limites para evitar ramificações inúteis durante o procedimento como um todo. Esses limites podem ser superiores ou inferiores, dependendo do objetivo de otimização do problema a ser resolvido, de maximização ou de minimização. Como o PCMRC é um problema de minimização, para facilitar a descrição do processo de *Bound*, vão ser feitas afirmações baseadas apenas neste objetivo.

Esses limites se tornam possíveis pois o custo da solução ótima s^0 , obtida por resolução linear, para P^0 é um limite inferior para as soluções de P . Caso tal afirmação não fosse verdadeira, significaria que existe uma solução s' para P , inteira, que tem custo menor que s^0 . Com isso, seria impossível s^0 ser uma solução ótima para P^0 , pois como P^0 possui todas as restrições de P , com exceção das relacionadas à integralidade das variáveis, qualquer solução do problema original é também uma solução do seu derivado linear. Matematicamente $\mathbb{Z} \subset \mathbb{R}$. Portanto, se s' é também uma solução para P^0 e tem custo menor que s^0 , então s' deveria ser a solução ótima, como, por hipótese, s^0 é ótimo, não existe solução como s' . Sendo assim, o custo de s^0 é um limite inferior para as soluções de P . (PAPADIMITRIOU; STEIGLITZ, 1998)

As definições anteriores permitem que as soluções inteiras, as folhas, encontradas durante as ramificações sejam usadas como limites inferiores. Por exemplo, caso um subproblema

P^k seja resolvido e a solução ótima s^k tenha custo maior que o limite inferior auferido até o momento, não há necessidade então de ramificar P^k , pois como s^k também limita inferiormente qualquer solução proveniente dos possíveis subproblemas de P^k , então nenhum desses possíveis resultados será melhor que a solução inteira já encontrada. (GOLDBARG; LUNA, 2005) (PAPADIMITRIOU; STEIGLITZ, 1998)

É nesses dois processos que o *Branch-and-Bound* se sustenta principalmente. Algumas outras decisões podem se fazer necessárias, como, por exemplo, a forma de desdobramento da árvore de busca, em largura ou em profundidade. Ainda assim, aquilo que define seu nome é aquilo que define seu método. Apesar da forma inteligente de abordar essa classe de problemas, este método não elimina uma dificuldade comum aos problemas que envolvem variáveis inteiras, a quantidade intratável de restrições que existe em muitos deles.

5.1.1 Separação das Restrições (3.4)

Conforme visto no capítulo 2 na Seção 2.4 sobre métodos exatos, incluir uma quantidade intratável de restrições durante a resolução do problema pode torná-lo inviável sob alguns aspectos. Normalmente, é realizado um processo chamado de *Separação* das que mais impactam no processo de resolução de um problema.

Assim como muitos outros problemas de natureza semelhante, o PCMRC possui um conjunto grande de restrições. A maior parte delas pertencentes ao conjunto definido por (3.4), que evita que exista mais de um ciclo na solução final. Estas restrições são definidas pelos subconjuntos S do conjunto V de vértices de uma instância de PCMRC, e por isso existem em quantidade exponencial.

Assim sendo, a abordagem exata proposta usa um problema subordinado, ou escravo, para realizar a separação dessas restrições. Tal problema é também de otimização e visa minimizar sua função objetivo que é baseada nas restrições (3.4). A seguir é mostrada uma variação desse conjunto de restrições.

$$x(\Gamma(S)) - 2 \sum_{j \in S} y_{ij} \geq 0, \quad \forall S \subset V : \text{raiz} \notin S, i \in S, \quad (5.1)$$

Com o problema escravo é definido também um problema mestre, que realiza a resolução do problema com B&B, mas, inicialmente, sem as restrições (3.4). O objetivo do escravo, portanto, é encontrar a restrição de (3.4) que minimize o valor do lado esquerdo de (5.1).

Os valores das variáveis x e y serão fixos, \bar{x} e \bar{y} , oriundos da execução do mestre, que produzirá uma solução com tais valores. O escravo então deve decidir qual o subconjunto S e qual vértice i que representam a configuração que mais viola (3.4), isto é, que tem valor mínimo na expressão contida no lado esquerdo de (5.1) baseando-se na solução fornecida pelo mestre.

Caso o problema escravo forneça um par, S e i , que possua um custo negativo para (5.1), esse par representa uma restrição violada, então o problema mestre é resolvido novamente considerando a restrição representada pelo par. Quando o par devolvido produz custo maior ou igual a 0, é sinal de que nenhum outro par poderá quebrar uma restrição desse conjunto.

Como dito anteriormente, o problema escravo descrito é também um problema de otimização, e por isso sua modelagem contém variáveis de decisão. A primeira delas é p_i que determina se o vértice $i \in V$ pertence a S . A segunda será q_i , que decide qual vértice $i \in V$ será aquele que é fixado em (3.4). Além dessas duas variáveis, que terão utilidade para o mestre, duas outras devem ser definidas para estabelecer as relações existentes entre o vértice fixo e aqueles pertencentes a S .

A primeira relação que necessita de uma variável determina que as únicas atribuições que podem ser contabilizadas no somatório da restrição são do vértice fixo i para vértices pertencentes a S , sendo que o próprio i deve pertencer a S também. A variável t_{ij} representará essa relação ao simular um E *booleano* entre p_j e q_i , ou seja, t_{ij} só terá valor 1 se p_j e q_i também o tiverem. Isto quer dizer que serão contabilizadas os valores \bar{y}_{ij} que tiverem o i sendo o vértice fixo escolhido e j pertencente a S , $\sum_{[i,j] \in A} \bar{y}_{ij} t_{ij}$.

A segunda relação que precisa ser levada em conta na resolução do escravo diz que as arestas contabilizadas por $x(\Gamma(S))$ devem ser do tipo (u, v) em que $u \in S$ e $v \notin S$, ou vice-versa, e tal aresta deve pertencer ao ciclo central na solução. Com isso, a variável a_{ij} é incluída no modelo e simulará a relação de um OU EXCLUSIVO *booleano* entre as variáveis p_i e p_j . Isto significa que esta variável nova só receberá valor 1 caso i ou j , exclusivamente, esteja em S . $\sum_{(i,j) \in E} \bar{x}_{ij} a_{ij}$. A seguir o modelo inteiro do problema escravo é apresentado.

$$\min \sum_{(u,v) \in E} a_{uv} \overline{x_{uv}} - 2 \sum_{[i,j] \in A} \overline{y_{ij}} t_{ij} \quad (5.2)$$

$$\text{sujeito a } \sum_{i \in V} q_i = 1 \quad (5.3)$$

$$p_{raiz} = 0 \quad (5.4)$$

$$q_i \leq p_i, \quad \forall i \in V \quad (5.5)$$

$$t_{ij} \leq p_j, \quad \forall [i, j] \in A \quad (5.6)$$

$$t_{ij} \leq q_i, \quad \forall [i, j] \in A \quad (5.7)$$

$$t_{ij} \geq q_i + p_j - 1, \quad \forall [i, j] \in A \quad (5.8)$$

$$a_{uv} \geq p_u - p_v, \quad \forall (u, v) \in E \quad (5.9)$$

$$a_{uv} \geq p_v - p_u, \quad \forall (u, v) \in E \quad (5.10)$$

$$a_{uv} \leq p_u + p_v, \quad \forall (u, v) \in E \quad (5.11)$$

$$a_{uv} \leq 2 - p_u - p_v, \quad \forall (u, v) \in E \quad (5.12)$$

$$p, q \in \{0, 1\}^{|V|} \quad (5.13)$$

$$a, t \in \{0, 1\}^{|E|} \quad (5.14)$$

A função objetivo em (5.2) é similar ao lado esquerdo de (5.1), com a diferença das variáveis a e t que possibilitam a contabilização correta das arestas e atribuições nos termos. Isso é necessário pois a função considera todas as arestas e atribuições e as variáveis de decisão realizam a filtragem para que sejam contadas apenas as que importam ao problema. Outra diferença é a função x estar representada por seu somatório.

A restrição (5.3) define que apenas um vértice deve ser o fixo, enquanto que (5.4) proíbe que a *raiz* pertença a S . Em seguida, (5.5) estabelece uma relação da restrição (3.4), o vértice fixo i deve pertencer a S , isto é incluído no modelo ao se permitir que q_i possa ter valor 1 caso p_i também o tenha. Caso p_i termine com valor 1, não necessariamente q_i o terá.

As restrições (5.6), (5.7) e (5.8) transformam a relação *booleana* E , representada pela variável t , para uma forma linear. Os modelos são construídos sobre sistemas lineares e só aceitam este de tipo de expressão. Então, se p_j ou q_i tiverem valor 0, (5.6) e (5.7) impedem t_{ij} de ter valor 1. Caso os dois tenham valor 1, as três restrições obrigam t_{ij} a ser menor ou igual a 1 e maior ou igual a 1 simultaneamente, restando para t_{ij} apenas possuir 1 como valor final. Esse comportamento representa um *E booleano*.

(5.9), (5.10), (5.11) e (5.12) transformam o OU EXCLUSIVO, descrito anteriormente, em expressões lineares que garantem que o comportamento será o mesmo dessa relação *booleana*. Caso p_u e p_v possuam valor 0, (5.9), (5.10), (5.11) obrigam a_{uv} a ter o mesmo valor. Se p_u e p_v tiverem valor 1, (5.12) proíbe a_{uv} de ter valor maior que 0, como esta variável é binária, então seu valor final será 0. Para o caso de apenas uma delas ter valor 1, relação de OU EXCLUSIVO, (5.12) fixa que a_{uv} será maior ou igual a 1, e pelo mesmo motivo anterior só resta a a_{uv} ter valor final 1.

Por fim, (5.13) e (5.14) determinam que as variáveis p , q , t e a são binárias. Com a descrição do *Branch-and-Bound* e do processo de separação das restrições em maior quantidade no modelo original, estabeleceu-se a abordagem exata proposta pelo presente trabalho. A seguir é apresentada a abordagem heurística, que será utilizada em conjunto com o algoritmo exato no procedimento híbrido a ser apresentado posteriormente.

5.2 GRASP-ILS

Procedimento aplicado por ANDRADE *et al.* (2016) ao PCMRC, realiza sua execução em duas fases: construção e refinamento. Na primeira, uma solução inicial é produzida pelo procedimento GRASP, a partir de um processo *guloso* que por meio de um parâmetro $\alpha \in [0, 1]$ define a proporção de aleatoriedade da solução esperada; na segunda fase, um processo de iteração, no espaço de vizinhança da solução construída, é realizado por meio do mecanismo de perturbação da heurística ILS. O procedimento torna-se interessante por mesclar a iteratividade do método GRASP com a potencialidade do procedimento de perturbação do ILS. Conforme Algoritmo 1.

O procedimento GRASP adotado possui um parâmetro extra *MaxIterFiltro*, que promove a geração de n soluções diversificadas em seu processo de construção local, diferentemente do processo convencional, com apenas uma solução para cada iteração GRASP. Após conclusão do procedimento GRASP com suas fases de construção e busca local, uma solução de boa qualidade é passada para o procedimento ILS.

O procedimento ILS possui um mecanismo de perturbação composto por três movimentos: adição, remoção e troca. O processo da adição inclui ao ciclo(V) um nó antes pertencente ao conjunto das atribuições(V'). O movimento de remoção corresponde ao processo contrário da adição, um elemento é removido do ciclo(V) e adicionado ao conjunto das atribuições(V'). A troca corresponde a combinação do procedimento de adição e remoção, ou seja, os conjuntos dos

nós pertencentes ao ciclo(V) e as atribuições(V') trocarão um elemento. Todos os movimentos, quando aplicados realizam a análise de viabilidade para suas ações.

Algoritmo 1: Pseudocódigo GRASP-ILS.

```

1: procedimento GRASP-ILS(MaxIterGRASP, MaxIterFiltro, NumMaxTentativaILS,
   MaxIterILS)
2:    $so \leftarrow$  GRASP(MaxIterGRASP, MaxIterFiltro,  $s$ );
3:    $s \leftarrow$  BuscaLocal( $so$ );
4:   enquanto  $i < \text{MaxIterILS}$  faça
5:      $s' \leftarrow$  Perturba( $s$ );
6:      $s'' \leftarrow$  BuscaLocal( $s'$ );
7:     se ( $f(s'') < f(s')$ ) faça
8:        $s \leftarrow s''$ ;
9:        $f(s) \leftarrow f(s'')$ ;
10:    fim-se
11:  fim-enquanto
12:  retorna( $s$ );
13: fim do procedimento

```

Fonte: (ANDRADE *et al.*, 2016)

5.3 GGVNS

O procedimento GGVNS, GRASP unido a GVNS, foi proposto por Dias (2006) junto a 5 outras meta-heurísticas e hibridizações. No referido trabalho, esta abordagem teve o melhor desempenho, conservando os melhores resultados de uma heurística para o PCMRC. Sua execução realiza dois processos, um de refinamento e outro de perturbação. O primeiro é realizado pelo procedimento GRASP, da mesma forma como foi descrito na seção anterior. O segundo procedimento realiza perturbações na solução produzida pelo GRASP. O Algoritmo 2 descreve a heurística GVNS utilizada na segunda etapa.

Algoritmo 2: Pseudocódigo GVNS.

```

1: procedimento GVNS(MaxIterGVNS, s)
2:   Inicializa estruturas de vizinhanças;
3:    $s^* \leftarrow s$ ;
4:    $Iter \leftarrow 1$ ;
5:   enquanto  $Iter < MaxIterGVNS$  faça
6:     // Perturbação
7:      $t \leftarrow random(2, 5)$ ;
8:     Aplicar  $t$  vezes, escolhendo de forma aleatória, um dos movimentos que definem
       as estruturas de vizinhança, obtendo um solução  $s'$  melhor;
9:     // Busca Local
10:    Aplicar a busca local sobre  $s'$  obtendo  $s''$ ;
11:    se ( $f(s'') < f(s^*)$ ) faça
12:       $s^* \leftarrow s''$ ;
13:       $f(s) \leftarrow f(s'')$ ;
14:    fim-se
15:     $Iter \leftarrow Iter + 1$ ;
16:  fim-enquanto
17:  retorna( $s^*$ );
18: fim do procedimento

```

Fonte: Adaptado de (DIAS, 2006)

O GVNS recebe dois parâmetros, um limitante de iterações, *MaxIterGVNS*, e a solução a ser perturbada e refinada. O processo de perturbação utiliza estruturas de vizinhanças para adicionar um vértice ao ciclo, *add*, remover um vértice do ciclo, *drop*, trocar um vértice do ciclo por um vértice atribuído, *swap*, e as operações de *2-opt* e *3-opt*. Essas 5 estruturas definem os limites para a quantidade t de perturbações realizadas, pelo menos 2 e no máximo 5, podendo ser aplicados 5 movimentos diferentes. A composição do GRASP com o GVNS é detalhada pelo Algoritmo 3.

Algoritmo 3: Pseudocódigo GGVNS.

- 1: **procedimento** GGVNS($MaxIterGRASP$, $MaxIterFiltro$, $MaxIterGVNS$, α)
 - 2: $so \leftarrow GRASP(MaxIterGRASP, MaxIterFiltro, \alpha)$;
 - 3: $s^* \leftarrow GVNS(MaxIterGVNS, so)$;
 - 4: retorna(s^*);
 - 5: **fim do procedimento**
-

Fonte: Adaptado de (DIAS, 2006)

O GGVNS recebe os limitantes de iterações para o GRASP e para o GVNS, e também um α para o processo de construção que o primeiro procedimento realiza. Com apresentação do funcionamento do GRASP-ILS, do GGVNS e do B&B, estão detalhados os três algoritmos que compõem as hibridizações que serão descritas a seguir.

5.4 Híbrido

Nesta seção serão apresentadas as hibridizações *quasi-exatas*. Os primeiros procedimentos, especificados na Subseção 5.4.1, consideram apenas a melhor solução encontrada pelo procedimento exato, enquanto que o último, na Subseção 5.4.2 utiliza todas as soluções inteiras produzidas durante a busca realizada pelo B&B.

5.4.1 Composições *Quasi-exatas*

Os métodos propostos por Dias (2006) e ANDRADE *et al.* (2016) produziram os melhores resultados heurísticos para o Problema do Ciclo Mediano sem Restrições de Capacidade. O primeiro deles, com o algoritmo GGVNS, realizou experimentos com duas classes de instâncias e foram comparados com o principal trabalho até então, Pérez *et al.* (2003). Já ANDRADE *et al.* (2016), em suas pesquisas, revelaram 31 novos resultados para o PCMRC, em experimentos com instâncias de até 75 vértices, demonstrando que os resultados anteriormente encontrados por Dias (2006) poderiam ser melhorados. Partindo desta perspectiva, a presente pesquisa propõe o uso de um processo exato como meio de determinar uma região promissora para a busca heurística realizada por estes procedimentos.

Algoritmo 4: Hibridização *Quasi-exata* com GRASP-ILS

- 1: **procedimento** HIBRIDO-GRASP-ILS (*instancia, gap, alfa*)
 - 2: $solucaoInicial \leftarrow B\&B(instancia, gap)$
 - 3: $solucaoFinal \leftarrow GRASP-ILS(instancia, solucaoInicial, alfa)$
 - 4: **retorna** $solucaoFinal$
 - 5: **fim do procedimento**
-

Fonte: Próprio autor.

Algoritmo 5: Hibridização *Quasi-exata* com GGVNS

- 1: **procedimento** HIBRIDO-GGVNS (*instancia, gap, alfa*)
 - 2: $solucaoInicial \leftarrow B\&B(instancia, gap)$
 - 3: $solucaoFinal \leftarrow GGVNS(instancia, solucaoInicial, alfa)$
 - 4: **retorna** $solucaoFinal$
 - 5: **fim do procedimento**
-

Fonte: Próprio autor.

A abordagem proposta, descrita nos Algoritmos 4 e 5, para GRASP-ILS e GGVNS respectivamente, é composta por duas etapas. A primeira executa o B&B sobre a entrada até o processo de execução ser interrompido, respeitando o limite aproximativo entre a solução inteira e a solução ótima para a instância recebida. A segunda etapa promove a continuidade do processo, utilizando como solução inicial para heurística aquela fornecida pela etapa anterior. O objetivo é explorar a solução obtida no processo realizado pelo B&B.

Assim como o objetivo de usar uma técnica exata é encontrar um espaço promissor de soluções para as heurísticas operarem, o próprio procedimento exato também realiza uma busca no espaço de soluções, e essa busca produz algumas outras soluções inteiras que não aquela fornecida ao atingir o limite aproximativo citado anteriormente. Com esse entendimento, este trabalho também propõe métodos que usem essas soluções adicionais.

5.4.2 Composição *Quasi-exata* com *Multistart*

Em sua execução, o procedimento *Branch-and-Bound* realiza buscas sobre o espaço de soluções de uma instância do PCMRC. Tal busca produz algumas soluções inteiras durante as ramificações do processo descrito na Subseção 5.1. Considerando o objetivo das hibridizações

como entregar um espaço promissor para as heurísticas, utilizar todas essas soluções geradas se revela uma estratégia pertinente.

Com isso, é proposta também uma hibridização que simula um procedimento *multistart* com essas diversas soluções. Nesse novo procedimento, o GGVNS é executado para cada uma das soluções iniciais oferecidas pelo algoritmo exato. A solução que obtiver o melhor custo é definida como a solução final a ser entregue, conforme descreve o Algoritmo 6.

Algoritmo 6: Hibridização *Quasi-exata Multistart* com GGVNS

- 1: **procedimento** HIBRIDO-GGVNS-MULTISTART (*instancia, gap, alfa*)
 - 2: *solucoesIniciais* \leftarrow *B&B*(*instancia, gap*)
 - 3: *solucaoFinal* \leftarrow *GGVNS*(*instancia, solucoesIniciais, alfa*)
 - 4: **retorna** *solucaoFinal*
 - 5: **fim do procedimento**
-

Fonte: Próprio autor.

6 RESULTADOS COMPUTACIONAIS E ATIVIDADES FUTURAS

Após conhecimento das abordagens adotadas, este capítulo apresenta os resultados computacionais obtidos para cada hibridização e a análise crítica frente aos procedimentos heurísticos desenvolvidos por ANDRADE *et al.* (2016) e Dias (2006).

6.1 Cenário

O algoritmo exato proposto foi implementado em linguagem Python® versão 2.7. Os experimentos foram executados com o *solver* da IBM – *International Business Machines* –, CPLEX® V12.4, dentro do *software* voltado para matemática SageMath versão 8.1. As heurísticas utilizadas na segunda fase da hibridização proposta foram implementadas em C++.

Apenas uma *thread* foi utilizada em uma máquina com processador Intel® Pentium G4560 de 3.50GHz e 2 núcleos de processamento, 16GB de memória RAM com frequência de 2409Mhz e sistema operacional Linux Ubuntu.

O algoritmo exato foi executado sem interrupção sobre instâncias de 10, 20, 30, 40, 50, 75, 100 e 125 vértices pertencentes à Classe II definida (de forma aleatória) em Labbé *et al.* (1999) e Labbé *et al.* (2004). Tais instâncias têm como formato **EUC2D** (distâncias euclidianas) para definir os pesos das ligações do grafo misto. Os testes com os procedimentos híbridos e com as heurísticas originais foram executados para instâncias de tamanho 50, 75 e 100 da mesma classe supracitada.

Os custos de ciclo e de atribuição foram definidos utilizando essas distâncias e um valor $\beta \in \{3, 5, 7, 9\}$ que diferencia os valores atribuídos para cada tipo de custo. Sendo G o grafo misto completo de uma instância qualquer de PCMRC e l_{ij} a distância euclidiana entre os vértices i e j pertencentes a $V(G)$, o custo de ciclo c_{ij} , para a aresta $(i, j) \in E(G)$, é definido por $c_{ij} = \beta \cdot l_{ij}$, ao passo que o custo de atribuição d_{ij} , para o arco $[i, j] \in A(G)$, é indicado por $d_{ij} = (10 - \beta) \cdot l_{ij}$.

Com os custos determinados, a primeira etapa da hibridização consiste de um algoritmo exato que se encerra antes da resolução completa da instância do problema. Para que isso ocorra, foi necessário estabelecer um valor de *GAP* padrão em sua execução. Este valor foi obtido durante uma bateria de testes com o GRASP-ILS simples, implementado por ANDRADE *et al.* (2016), e calculado como a média aritmética, em porcentagem, das diferenças entre os custos das soluções produzidas por cada construção e por cada refinamento após tal

construção. O valor auferido foi de 34.66%. Então, quando a solução inteira corrente atinge, em determinado momento do B&B, uma diferença de 34.66%, ou menos, para o ótimo, a execução se encerra e a solução atual é passada como solução inicial para a segunda etapa da hibridização, a execução do procedimento heurístico a ela relacionado. Este *GAP* foi utilizado nos testes de todas as instâncias, com exceção da instância f100_4 para $\beta = 9$, que provocou um estouro de memória. Foi, então, escolhido um *GAP* diferente para sua execução, 40%, adotado a partir de uma observação dos *logs* de execução do algoritmo exato.

Em relação à etapa da abordagem que utiliza uma das heurísticas descritas no Capítulo 5, os valores de $\alpha \in [0, 1]$ utilizados nos experimentos foram $\{0.1, 0.5, 0.9\}$, definindo uma execução mais gulosa (0.1), equilibrada entre gulosa e aleatória (0.5), e mais aleatória (0.9). Para cada instância foi aplicado a combinação dos valores de β e os 3 valores de α .

Para evitar execuções muito demoradas nesta etapa, é necessário delimitar o número máximo de iterações que os procedimentos heurísticos realizarão. Para o GRASP-ILS, esse limite foi definido como 70, 55 e 80 iterações para as instâncias de 50, 75 e 100 vértices respectivamente, sendo a relação com o tamanho das instâncias $50 + 20$, $75 - 20$ e $100 - 20$. Já o número máximo de iterações do filtro na fase de construção foi determinado em (*MaxIterFiltro*) 100. Outros dois valores máximos a estabelecer estão relacionados particularmente ao ILS. O primeiro define o número máximo de iterações que a busca local executará: 20 para instâncias de 50 vértices e 15 para o restante. O segundo valor limitante diz respeito à quantidade máxima permitida de tentativas de melhora em uma vizinhança, restringindo-se a 15 para instâncias de tamanho 50 e 10 para as instâncias restantes, conforme estabelecido na Tabela 1. Com relação ao GGVNS, os limitantes foram demarcados como o tamanho da instância para o número máximo de iterações do GRASP e 10 iterações para a heurística GVNS.

Tabela 1 – Parâmetros estabelecidos para o procedimento Heurístico GRASP-ILS

| Instância | Máximo de Iterações | | |
|-----------|---------------------|-----------------|------------------|
| | GRASP-ILS | Busca Local ILS | Perturbações ILS |
| 50 nós | 70 | 20 | 15 |
| 75 nós | 55 | 15 | 10 |
| 100 nós | 80 | 15 | 10 |

Fonte: Adaptado de ANDRADE *et al.* (2016)

Os procedimentos mistos adotados nesta pesquisa tiveram seus resultados comparados com experimentos realizados com os algoritmos propostos em ANDRADE *et al.* (2016) e Dias (2006). Desta forma, para a execução dos testes, foram mantidos os parâmetros originais dos

autores e cada caso de teste foi executado apenas uma vez para cada abordagem, diferentemente dos originais que executaram 5 vezes cada um dos casos e escolheram o melhor resultado entre os obtidos como resultado final. A presente pesquisa executa apenas uma vez cada caso pois realizou experimentos com todos os algoritmos no mesmo ambiente de execução.

6.2 Resultados - *Branch-and-Bound*

Foram realizados experimentos computacionais apenas com o algoritmo exato B&B pois, após análise bibliográfica, foi verificada a necessidade de conhecer o comportamento das instâncias devido a ausência de valores estabelecidos para suas Funções Objetivo (FO). O apêndice A apresenta os resultados brutos obtidos para cada instância com $\beta \in \{3, 5, 7, 9\}$ cuja as execuções foram concluídas com sucesso.

Observou-se que dos 320 casos de testes, 25 não concluíram sua execução devido ao consumo de memória, chegando a média de 7,8% de aproximação do valor ótimo para a FO, conforme apresenta a Tabela 2. Dada esta característica, não foram executados os experimentos para as instâncias apresentadas na Tabela 3, visto que tais instâncias possuíam um comportamento com consumo elevado de memória, obtendo valores aproximativos ainda para a FO com $\beta = 7$.

Conhecidos os valores estabelecidos da maioria das instâncias pertencentes a classe de testes definida em Labbé *et al.* (1999) e Labbé *et al.* (2004), foram realizados experimentos comparativos, para instâncias com 50, 75 e 100 vértices, entre os procedimentos puros GRASP-ILS e GGVNS e as soluções mistas utilizadas nesta pesquisa.

6.3 Resultados Comparativos

A seguir serão discutidos os resultados dos testes realizados para as 3 abordagens híbridas em comparação com os algoritmos originais dos trabalhos citados anteriormente. A análise dos resultados obtidos centra-se na qualidade das soluções produzidas por cada um dos métodos testados.

6.3.1 *Quasi-exato com GRASP-ILS e GRASP-ILS*

Os resultados da execução dos casos de teste para a abordagem híbrida com o GRASP-ILS e para o próprio GRASP-ILS são apresentados nas Tabelas 4, 5 e 6. Nelas estão

Tabela 2 – Resultados aproximativos das execuções exatas

| Instância | β | FO | Tempo | Proporção de Aproximação |
|------------------|---------------------------|-----------|--------------|---------------------------------|
| f75_2 | 9 | 27209 | 28572.50 | 12.60% |
| f75_3 | 9 | 27759 | 20764.87 | 10.63% |
| f75_5 | 9 | 28820 | 23485.91 | 16.07% |
| f75_7 | 9 | 28283 | 27364.41 | 11.88% |
| f75_10 | 9 | 28551 | 26342.25 | 13.12% |
| f100_1 | 7 | 41429 | 25202.57 | 3.07% |
| f100_2 | 7 | 41997 | 20361.72 | 3.49% |
| f100_3 | 9 | 28662 | 29798.96 | 19.79% |
| f100_4 | 7 | 41878 | 38206.65 | 3.85% |
| f100_5 | 7 | 40390 | 29021.08 | 4.33% |
| f100_6 | 7 | 41619 | 30076.95 | 6.60% |
| f100_7 | 9 | 19486 | 17500.36 | 10.72% |
| f100_8 | 7 | 42395 | 21243.60 | 2.81% |
| f100_9 | 7 | 40440 | 18663.72 | 5.52% |
| f100_10 | 7 | 43262 | 24663.28 | 5.20% |
| f125_1 | 7 | 44186 | 29265.33 | 7.09% |
| f125_2 | 7 | 45547 | 47902.02 | 7.95% |
| f125_3 | 7 | 47640 | 37576.48 | 8.77% |
| f125_4 | 7 | 44309 | 42731.69 | 5.24% |
| f125_5 | 7 | 46437 | 32973.29 | 7.47% |
| f125_6 | 7 | 46562 | 41501.58 | 6.78% |
| f125_7 | 7 | 47599 | 34893.45 | 5.58% |
| f125_8 | 7 | 47679 | 51820.78 | 6.22% |
| f125_9 | 7 | 45971 | 31058.85 | 6.17% |
| f125_10 | 7 | 45210 | 43181.49 | 3.88% |

Fonte: Próprio autor

Tabela 3 – Instâncias não executadas nos experimentos com o B&B

| Instância | β | Instância | β |
|------------------|---------------------------|------------------|---------------------------|
| f100_1 | 9 | f125_2 | 9 |
| f100_2 | 9 | f125_3 | 9 |
| f100_4 | 9 | f125_4 | 9 |
| f100_5 | 9 | f125_5 | 9 |
| f100_6 | 9 | f125_6 | 9 |
| f100_8 | 9 | f125_7 | 9 |
| f100_9 | 9 | f125_8 | 9 |
| f100_10 | 9 | f125_9 | 9 |
| f125_1 | 9 | f125_10 | 9 |

Fonte: Próprio autor

dispostos os valores das FO para os experimentos realizados com os parâmetros β e α aplicados às instâncias de 50, 75 e 100 nós respectivamente.

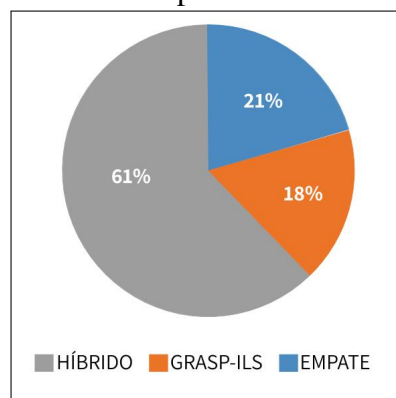
Os resultados dos experimentos para as instâncias de 50 nós, conforme destacado na Tabela 4, revelam ganho em 53 dos 120 casos de testes, com percentual de 47,5% para os experimentos com o uso do parâmetro $\alpha = 0.1$, contra 42,5% para os demais α aplicados. Esses ganhos foram destacados em negrito para melhor visualização. Os mesmos experimentos revelam a paridade das FO para 32 casos de testes. Contudo, o desempenho da hibridização *quasi-exata* proposta neste trabalho ganha mais destaque com os resultados obtidos para as instâncias seguintes.

A Tabela 5 expõe o desempenho superior da hibridização proposta em 76 dos 120 casos de testes para instâncias de 75 nós, correspondendo a 63,3% de ganho, ou seja, melhoria na FO obtida quando comparada aos valores do procedimento GRASP-ILS. Em confronto com as instâncias de 50 nós, os resultados presentes na Tabela 5 demonstram uma diminuição de paridade das FO para 21 casos, assim como revela a potencialidade do método a medida em que eleva-se o número de nós trabalhados. Esta característica é melhor destacada na análise dos experimentos para instâncias de 100 nós.

Os experimentos para instâncias de 100 nós revelam um melhor desempenho do procedimento híbrido, conforme destacado na Tabela 6, com ganho em 91 dos 120 casos de testes, tendo igualdade de desempenho em apenas 2 casos. Os ganhos computacionais são equilibrados quando analisada a aplicação do parâmetro α : 31, 31 e 29, respectivamente, para os α 0,1, 0,5 e 0,9.

Um conjunto de 360 casos de testes foram executados para instâncias de 50, 75 e 100 nós. Experimentos computacionais revelaram, conforme Figura 14, que para 220 casos (61%) o desempenho da hibridização *quasi-exata* apresentou resultados melhores para a FO.

Figura 14 – Desempenho dos procedimentos GRASP-ILS e Híbrido para os 360 casos de testes



Fonte: Próprio autor

Tabela 4 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 50 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório ($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|------------------------------|--------------|
| | | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido |
| f50_1 | 3 | 18090 | 18090 | 18021 | 18216 | 18021 | 18216 |
| | 5 | 28270 | 28210 | 28360 | 28100 | 28200 | 28225 |
| | 7 | 30387 | 30020 | 30410 | 30020 | 30425 | 30020 |
| | 9 | 22942 | 22942 | 22942 | 22942 | 22942 | 22942 |
| f50_2 | 3 | 18703 | 18591 | 18703 | 18591 | 18703 | 18591 |
| | 5 | 28580 | 28580 | 28545 | 28580 | 28580 | 28580 |
| | 7 | 30655 | 31089 | 31093 | 31133 | 31188 | 31194 |
| | 9 | 22081 | 22081 | 22081 | 22081 | 22081 | 22081 |
| f50_3 | 3 | 17788 | 17788 | 17784 | 17788 | 17691 | 17788 |
| | 5 | 28560 | 28535 | 28535 | 28535 | 28565 | 28535 |
| | 7 | 32404 | 32390 | 32585 | 32441 | 32717 | 32370 |
| | 9 | 22295 | 22295 | 22295 | 22295 | 22295 | 22295 |
| f50_4 | 3 | 15507 | 15507 | 15495 | 15507 | 15453 | 15507 |
| | 5 | 24870 | 24765 | 24800 | 24765 | 24785 | 24735 |
| | 7 | 29038 | 28830 | 28941 | 28893 | 28932 | 28959 |
| | 9 | 20778 | 20778 | 20778 | 20778 | 20778 | 20778 |
| f50_5 | 3 | 17561 | 17561 | 17544 | 17561 | 17529 | 17561 |
| | 5 | 27210 | 26950 | 26970 | 26950 | 27025 | 26950 |
| | 7 | 29835 | 29365 | 29513 | 29268 | 29539 | 29268 |
| | 9 | 23585 | 23585 | 23585 | 23585 | 23585 | 23585 |
| f50_6 | 3 | 16646 | 16626 | 16646 | 16626 | 16626 | 16626 |
| | 5 | 26225 | 26225 | 26225 | 26350 | 26255 | 26270 |
| | 7 | 30179 | 29958 | 29996 | 29926 | 29996 | 29994 |
| | 9 | 20709 | 20709 | 20709 | 20709 | 20709 | 20709 |
| f50_7 | 3 | 17256 | 17217 | 17217 | 17217 | 17241 | 17217 |
| | 5 | 26250 | 26260 | 26350 | 26170 | 26360 | 26095 |
| | 7 | 30481 | 29901 | 30265 | 29901 | 30074 | 29901 |
| | 9 | 22127 | 22127 | 22127 | 22127 | 22127 | 22127 |
| f50_8 | 3 | 16762 | 16762 | 16762 | 16762 | 16740 | 16762 |
| | 5 | 27215 | 27665 | 27325 | 27535 | 27215 | 27745 |
| | 7 | 32910 | 32996 | 32847 | 32754 | 33066 | 32847 |
| | 9 | 26564 | 26564 | 26564 | 26564 | 26564 | 26564 |
| f50_9 | 3 | 17326 | 16903 | 16903 | 16903 | 16893 | 16903 |
| | 5 | 27100 | 26900 | 27145 | 26900 | 26990 | 26900 |
| | 7 | 29710 | 29261 | 29443 | 29375 | 29753 | 29261 |
| | 9 | 22282 | 22282 | 22282 | 22282 | 22282 | 22352 |
| f50_10 | 3 | 18041 | 17934 | 18062 | 17934 | 17967 | 17934 |
| | 5 | 27920 | 27890 | 27915 | 27890 | 27925 | 27890 |
| | 7 | 28847 | 28651 | 28651 | 28651 | 28841 | 28651 |
| | 9 | 21715 | 21825 | 21715 | 21715 | 21715 | 21715 |

Fonte: Próprio autor

Tabela 5 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 75 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório ($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|------------------------------|--------------|
| | | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido |
| f75_1 | 3 | 21346 | 21019 | 21019 | 21019 | 21012 | 21019 |
| | 5 | 34030 | 33610 | 34050 | 33560 | 34165 | 33520 |
| | 7 | 39003 | 38988 | 38958 | 38712 | 38541 | 38863 |
| | 9 | 27479 | 27479 | 27479 | 27479 | 27479 | 27479 |
| f75_2 | 3 | 19468 | 19486 | 19468 | 19486 | 19437 | 19486 |
| | 5 | 31125 | 31145 | 31220 | 30810 | 31560 | 30900 |
| | 7 | 36816 | 35432 | 36288 | 35088 | 36025 | 35224 |
| | 9 | 26974 | 26972 | 26972 | 27006 | 26974 | 26972 |
| f75_3 | 3 | 21086 | 20948 | 20948 | 20948 | 20976 | 20948 |
| | 5 | 34100 | 33395 | 34090 | 33480 | 33940 | 33370 |
| | 7 | 38034 | 37278 | 37547 | 37167 | 37548 | 37255 |
| | 9 | 27443 | 27443 | 27506 | 27388 | 27443 | 27388 |
| f75_4 | 3 | 19509 | 19509 | 19509 | 19509 | 19509 | 19509 |
| | 5 | 34420 | 31425 | 32000 | 31360 | 32060 | 31285 |
| | 7 | 36306 | 34746 | 35482 | 34851 | 35029 | 34776 |
| | 9 | 25855 | 25855 | 25855 | 25855 | 25855 | 25855 |
| f75_5 | 3 | 19288 | 19299 | 19050 | 19299 | 19050 | 19299 |
| | 5 | 31260 | 30040 | 30035 | 30000 | 30295 | 30085 |
| | 7 | 35342 | 34135 | 34730 | 33889 | 34453 | 34006 |
| | 9 | 28917 | 28772 | 28845 | 28772 | 29031 | 28772 |
| f75_6 | 3 | 20447 | 20660 | 20447 | 20660 | 20447 | 20660 |
| | 5 | 33855 | 33105 | 32935 | 33255 | 33295 | 33475 |
| | 7 | 37971 | 37096 | 37988 | 37209 | 37669 | 37226 |
| | 9 | 25962 | 25962 | 25965 | 25962 | 25962 | 25962 |
| f75_7 | 3 | 20659 | 20686 | 20623 | 20686 | 20623 | 20686 |
| | 5 | 32595 | 32220 | 32870 | 32050 | 32300 | 32135 |
| | 7 | 36882 | 35967 | 36790 | 35668 | 36881 | 35671 |
| | 9 | 28264 | 28223 | 28279 | 28178 | 28383 | 28178 |
| f75_8 | 3 | 19161 | 19161 | 19161 | 19161 | 19161 | 19161 |
| | 5 | 31470 | 31060 | 31150 | 31050 | 31195 | 30995 |
| | 7 | 37743 | 35801 | 36951 | 35782 | 36166 | 35615 |
| | 9 | 26559 | 26598 | 26543 | 26578 | 26543 | 26543 |
| f75_9 | 3 | 19650 | 19650 | 19827 | 19650 | 19650 | 19650 |
| | 5 | 32000 | 31165 | 31670 | 30825 | 31745 | 30970 |
| | 7 | 35800 | 34197 | 34856 | 34491 | 34992 | 34395 |
| | 9 | 25025 | 25012 | 24957 | 24957 | 25025 | 24962 |
| f75_10 | 3 | 20098 | 20356 | 19945 | 20356 | 19926 | 20356 |
| | 5 | 32690 | 31445 | 32120 | 31330 | 31740 | 31345 |
| | 7 | 35383 | 35057 | 35803 | 35339 | 35758 | 35158 |
| | 9 | 28422 | 28355 | 28531 | 28356 | 28480 | 28424 |

Fonte: Próprio autor

Tabela 6 – Desempenho dos métodos GRASP-ILS e a proposta Híbrida para instâncias de 100 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório ($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|------------------------------|--------------|
| | | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido | GRASP-ILS | Híbrido |
| f100_1 | 3 | 22951 | 22969 | 22951 | 22969 | 22951 | 22969 |
| | 5 | 38975 | 36920 | 38435 | 36880 | 37795 | 36880 |
| | 7 | 43982 | 42483 | 43480 | 42199 | 43470 | 42263 |
| | 9 | 29843 | 29552 | 29842 | 29561 | 29976 | 29546 |
| f100_2 | 3 | 23611 | 23566 | 23566 | 23566 | 23566 | 23566 |
| | 5 | 39275 | 38290 | 39000 | 38050 | 39235 | 38010 |
| | 7 | 44589 | 42530 | 44248 | 42516 | 43728 | 42557 |
| | 9 | 31530 | 31268 | 31306 | 31193 | 31274 | 31280 |
| f100_3 | 3 | 21998 | 22622 | 21998 | 22622 | 21998 | 22622 |
| | 5 | 39240 | 35685 | 36995 | 36045 | 36505 | 35845 |
| | 7 | 41760 | 38907 | 40530 | 38969 | 39557 | 38549 |
| | 9 | 28391 | 28244 | 28410 | 28271 | 28427 | 28256 |
| f100_4 | 3 | 23917 | 24013 | 23863 | 23935 | 23863 | 23935 |
| | 5 | 39780 | 37440 | 39475 | 37625 | 39050 | 37440 |
| | 7 | 44648 | 42991 | 43606 | 42652 | 43558 | 42609 |
| | 9 | 33938 | 33629 | 33767 | 33729 | 33742 | 33641 |
| f100_5 | 3 | 22817 | 23107 | 22817 | 23107 | 22817 | 23107 |
| | 5 | 37635 | 36785 | 37745 | 36820 | 37350 | 36860 |
| | 7 | 42647 | 41506 | 42106 | 41475 | 41366 | 41385 |
| | 9 | 33803 | 33882 | 33908 | 33679 | 33850 | 33833 |
| f100_6 | 3 | 23403 | 23487 | 23400 | 23487 | 23358 | 23457 |
| | 5 | 37415 | 36440 | 37615 | 36380 | 38360 | 36405 |
| | 7 | 44466 | 43172 | 43951 | 42879 | 44001 | 43124 |
| | 9 | 32257 | 31687 | 32271 | 31672 | 32329 | 31720 |
| f100_7 | 3 | 22750 | 22865 | 22745 | 22865 | 22745 | 22865 |
| | 5 | 38815 | 35885 | 37090 | 35880 | 37100 | 35865 |
| | 7 | 41363 | 40075 | 40748 | 39843 | 40583 | 39943 |
| | 9 | 30041 | 29926 | 29943 | 29922 | 29982 | 29926 |
| f100_8 | 3 | 24213 | 24321 | 24111 | 24321 | 24000 | 24270 |
| | 5 | 39900 | 37905 | 39000 | 37815 | 38685 | 37720 |
| | 7 | 44782 | 43212 | 44744 | 43164 | 44822 | 43194 |
| | 9 | 31240 | 30930 | 31090 | 30933 | 31227 | 30927 |
| f100_9 | 3 | 24069 | 24130 | 23815 | 24130 | 23845 | 24130 |
| | 5 | 40790 | 37295 | 38700 | 37295 | 38720 | 36990 |
| | 7 | 43038 | 40793 | 41420 | 40982 | 41792 | 40833 |
| | 9 | 32148 | 32043 | 32225 | 32014 | 32316 | 31822 |
| f100_10 | 3 | 24480 | 24045 | 24120 | 24045 | 24051 | 24045 |
| | 5 | 40470 | 38350 | 40185 | 38350 | 40495 | 38340 |
| | 7 | 45015 | 44046 | 45091 | 43980 | 44873 | 43748 |
| | 9 | 31737 | 31555 | 31732 | 31534 | 31744 | 31536 |

Fonte: Próprio autor

O comportamento da hibridização *quasi-exata* com GRASP-ILS é destacado em instâncias de maior porte, visto que para instâncias de 50 nós obteve 53 ganhos, para instâncias de 75 nós 76 ganhos e para instâncias de 100 nós 91 ganhos. Reflexo deste desempenho é apresentado no Apêndice B, que descreve a análise comparativa dos resultados coletados nos experimentos computacionais. Desta forma, a presente pesquisa verifica que independente do fator de gulosidade do procedimento GRASP-ILS, o uso da estratégia exata promove o encontro de um espaço de busca promissor para este procedimento heurístico.

6.3.2 *Quasi-exato com GGVNS e GGVNS*

Os resultados dos testes executados para a abordagem híbrida com GGVNS e para o GGVNS em si estão dispostos nas Tabelas 7, 8 e 9. Elas apresentam os valores das FO para cada configuração de execução com uma instância de 50, 75 ou 100 vértices, respectivamente, um valor de β e outro de α .

Os testes para instâncias de 50 nós, com resultados organizados na Tabela 7, apresentaram um equilíbrio entre as abordagens, com uma pequena vantagem para o GGVNS. Dos 120 casos testes para esse tamanho de instância, foram 49 ganhos do GGVNS, 43 do Híbrido e 28 empates. Tal equilíbrio se sustenta quando analisados os resultados por valor de β ou de α . Assim como na Subseção anterior os ganhos do Híbrido estão destacados em negrito. No entanto, a medida que cresce o tamanho das instâncias, o híbrido se sobressai em relação ao GGVNS.

Na Tabela 8 pode-se observar a melhora nos resultados para o procedimento híbrido. Foram 83 ganhos contra apenas 36 para o GGVNS. Dos 120 casos, então, o híbrido se saiu melhor em aproximadamente 69% deles. Para este tamanho de instância, apenas um empate ocorreu, para o caso de teste com a instância f75_10, $\beta = 9$ e $\alpha = 0.9$. Com relação à análise por parâmetro, pode ser destacado que para $\alpha = 0.1$, o híbrido tem ganhos de 90%. Enquanto que para o parâmetro β , os testes com valores 5, 7 e 9 se destacaram, com cerca de 84% de ganhos. Contudo, o melhor resultado aparece em instâncias de 100 vértices.

Na Tabela 9 são apresentados os resultados para instâncias com 100 nós. Em ganhos gerais, o híbrido ganhou em 97 casos, à medida que o GGVNS teve 23. Nesta faixa não houveram empates. Um resultado a se destacar refere-se aos casos com $\beta = 5$, em que o Híbrido ganhou todos os 30 realizados.

No total foram realizados 360 casos, de forma semelhante ao que foi executado para o GRASP-ILS. Considerando todos eles de forma conjunta, a Figura 15 demonstra a relação de

Tabela 7 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 50 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório ($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|------------------------------|--------------|
| | | GGVNS | Híbrido | GGVNS | Híbrido | GGVNS | Híbrido |
| f50_1 | 3 | 19062 | 25599 | 18078 | 25599 | 18021 | 25599 |
| | 5 | 28330 | 28320 | 28145 | 28320 | 28145 | 28275 |
| | 7 | 31300 | 30286 | 30707 | 30037 | 30180 | 30073 |
| | 9 | 22942 | 22942 | 22942 | 22942 | 22942 | 22942 |
| f50_2 | 3 | 19590 | 18591 | 18591 | 18591 | 18723 | 18591 |
| | 5 | 28660 | 28580 | 28580 | 28580 | 28570 | 28580 |
| | 7 | 31224 | 31793 | 30808 | 32069 | 30860 | 31721 |
| | 9 | 22081 | 22081 | 22081 | 22081 | 22081 | 22081 |
| f50_3 | 3 | 19116 | 19941 | 18201 | 19941 | 17799 | 19941 |
| | 5 | 29245 | 28535 | 28610 | 28535 | 28610 | 28535 |
| | 7 | 32695 | 33490 | 32643 | 32912 | 32390 | 33246 |
| | 9 | 22300 | 22300 | 22300 | 22300 | 22300 | 22300 |
| f50_4 | 3 | 16965 | 18675 | 15597 | 18675 | 15453 | 18675 |
| | 5 | 24955 | 24735 | 24835 | 24735 | 24735 | 24735 |
| | 7 | 29462 | 29387 | 29232 | 29324 | 28868 | 29332 |
| | 9 | 20778 | 20778 | 20778 | 20778 | 20778 | 20778 |
| f50_5 | 3 | 18540 | 21369 | 17610 | 21369 | 17544 | 21369 |
| | 5 | 27430 | 26950 | 26970 | 26950 | 27080 | 26950 |
| | 7 | 31563 | 29717 | 30187 | 29838 | 30101 | 29814 |
| | 9 | 23585 | 23594 | 23585 | 23594 | 23585 | 23585 |
| f50_6 | 3 | 17457 | 16626 | 16683 | 16626 | 16626 | 16626 |
| | 5 | 26295 | 26695 | 26385 | 26695 | 26295 | 26695 |
| | 7 | 30416 | 30054 | 30310 | 29881 | 30493 | 29945 |
| | 9 | 20709 | 20709 | 20709 | 20709 | 20709 | 20709 |
| f50_7 | 3 | 19776 | 17217 | 17406 | 17217 | 17241 | 17217 |
| | 5 | 27755 | 26095 | 26405 | 26095 | 26095 | 26095 |
| | 7 | 31968 | 29901 | 30585 | 29901 | 30605 | 29901 |
| | 9 | 22127 | 22127 | 22127 | 22127 | 22127 | 22127 |
| f50_8 | 3 | 17175 | 22551 | 16839 | 22551 | 16746 | 22551 |
| | 5 | 27510 | 27745 | 27310 | 27745 | 27220 | 27750 |
| | 7 | 34362 | 33137 | 33268 | 32858 | 33131 | 32854 |
| | 9 | 26867 | 26672 | 26564 | 26752 | 26574 | 26676 |
| f50_9 | 3 | 17802 | 18819 | 17124 | 18819 | 16968 | 18819 |
| | 5 | 27215 | 26965 | 27030 | 26965 | 26920 | 26965 |
| | 7 | 29617 | 29669 | 29466 | 29603 | 29613 | 29757 |
| | 9 | 22282 | 22486 | 22282 | 22486 | 22282 | 22282 |
| f50_10 | 3 | 19425 | 17949 | 18309 | 17949 | 18045 | 17949 |
| | 5 | 27900 | 27970 | 27925 | 27970 | 27890 | 27980 |
| | 7 | 29677 | 28759 | 28996 | 28852 | 28651 | 28810 |
| | 9 | 21715 | 21715 | 21715 | 21715 | 21715 | 21715 |

Fonte: Próprio autor

Tabela 8 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 75 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|-----------------------------|--------------|
| | | GGVNS | Hibrido | GGVNS | Hibrido | GGVNS | Hibrido |
| f75_1 | 3 | 22995 | 22470 | 21516 | 22470 | 21204 | 22470 |
| | 5 | 35215 | 33550 | 34795 | 33520 | 33775 | 33520 |
| | 7 | 40253 | 39231 | 39461 | 38918 | 38800 | 38983 |
| | 9 | 27479 | 27849 | 27628 | 27809 | 27567 | 27809 |
| f75_2 | 3 | 21456 | 19611 | 19548 | 19611 | 19470 | 19611 |
| | 5 | 32790 | 30945 | 31640 | 30910 | 31185 | 30895 |
| | 7 | 37099 | 36171 | 36186 | 35851 | 35778 | 35667 |
| | 9 | 27354 | 27179 | 27220 | 27179 | 27193 | 27190 |
| f75_3 | 3 | 23385 | 21288 | 21510 | 21288 | 21222 | 21288 |
| | 5 | 34600 | 33285 | 34560 | 33610 | 33270 | 33395 |
| | 7 | 40233 | 37323 | 38725 | 37314 | 37558 | 37388 |
| | 9 | 27841 | 27857 | 27545 | 27710 | 27697 | 27841 |
| f75_4 | 3 | 20085 | 22626 | 19755 | 22626 | 19719 | 22626 |
| | 5 | 40455 | 31420 | 32950 | 31410 | 31815 | 31410 |
| | 7 | 38922 | 35399 | 35613 | 35425 | 35294 | 34924 |
| | 9 | 26238 | 25855 | 26148 | 25855 | 26281 | 25855 |
| f75_5 | 3 | 20100 | 20382 | 19299 | 20382 | 19212 | 20382 |
| | 5 | 31190 | 30240 | 30320 | 30005 | 30020 | 30030 |
| | 7 | 37769 | 33813 | 36185 | 33857 | 35155 | 33902 |
| | 9 | 29801 | 28812 | 29436 | 28776 | 29530 | 28783 |
| f75_6 | 3 | 22146 | 22851 | 21108 | 22851 | 20691 | 22851 |
| | 5 | 35010 | 33695 | 33640 | 34070 | 33025 | 34070 |
| | 7 | 39547 | 38286 | 38451 | 37730 | 37218 | 38159 |
| | 9 | 26088 | 26040 | 26091 | 26041 | 26144 | 26041 |
| f75_7 | 3 | 21588 | 22353 | 20832 | 22353 | 20805 | 22353 |
| | 5 | 34150 | 32235 | 33025 | 32195 | 32400 | 32090 |
| | 7 | 37271 | 36187 | 37392 | 35845 | 36872 | 36138 |
| | 9 | 28841 | 28192 | 28617 | 28287 | 28440 | 28287 |
| f75_8 | 3 | 20286 | 19812 | 19491 | 19812 | 19359 | 19812 |
| | 5 | 33510 | 31160 | 31895 | 31025 | 31070 | 31040 |
| | 7 | 38100 | 36164 | 36803 | 36193 | 36808 | 36108 |
| | 9 | 26860 | 26741 | 26815 | 26741 | 26799 | 26741 |
| f75_9 | 3 | 21996 | 20394 | 20325 | 20394 | 19815 | 20394 |
| | 5 | 34235 | 31105 | 32055 | 31030 | 31460 | 30845 |
| | 7 | 37147 | 34647 | 36086 | 34704 | 34720 | 34747 |
| | 9 | 25221 | 25052 | 25177 | 24957 | 25281 | 25012 |
| f75_10 | 3 | 21630 | 20595 | 20559 | 20595 | 20046 | 20595 |
| | 5 | 33380 | 31055 | 32090 | 31055 | 31325 | 31025 |
| | 7 | 37717 | 35474 | 36969 | 35393 | 35559 | 35398 |
| | 9 | 28488 | 28471 | 28529 | 28471 | 28471 | 28471 |

Fonte: Próprio autor

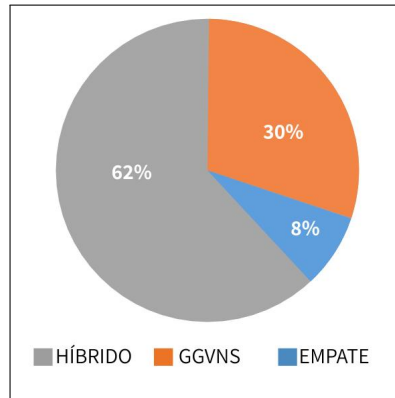
Tabela 9 – Desempenho dos métodos GGVNS e a proposta Híbrida para instâncias de 100 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório ($\alpha = 0.9$) | |
|-----------|---------|---------------------------|--------------|----------------------------|--------------|------------------------------|--------------|
| | | GGVNS | Híbrido | GGVNS | Híbrido | GGVNS | Híbrido |
| f100_1 | 3 | 24630 | 23241 | 23907 | 23241 | 23400 | 23241 |
| | 5 | 42400 | 37290 | 39755 | 37060 | 39115 | 37100 |
| | 7 | 47538 | 44557 | 44175 | 44251 | 42951 | 44381 |
| | 9 | 30490 | 29704 | 30212 | 29738 | 29949 | 29704 |
| f100_2 | 3 | 26064 | 24225 | 24156 | 24225 | 23874 | 24225 |
| | 5 | 46390 | 37995 | 40875 | 38075 | 39090 | 37855 |
| | 7 | 50041 | 42623 | 45484 | 42768 | 44250 | 42838 |
| | 9 | 31580 | 31391 | 31453 | 31244 | 31557 | 31253 |
| f100_3 | 3 | 24972 | 25395 | 22914 | 25395 | 22584 | 25395 |
| | 5 | 42085 | 36110 | 37645 | 35920 | 36490 | 36155 |
| | 7 | 44862 | 38666 | 41099 | 38551 | 40888 | 38686 |
| | 9 | 29810 | 28188 | 28513 | 28198 | 28463 | 28198 |
| f100_4 | 3 | 26010 | 27084 | 24510 | 27084 | 24306 | 27084 |
| | 5 | 40325 | 37675 | 38515 | 37855 | 39225 | 37595 |
| | 7 | 47935 | 43292 | 45242 | 43178 | 44723 | 43089 |
| | 9 | 34639 | 33706 | 34077 | 33674 | 34539 | 33690 |
| f100_5 | 3 | 24111 | 25398 | 23235 | 25398 | 23208 | 25398 |
| | 5 | 37900 | 36815 | 37880 | 36875 | 36930 | 36835 |
| | 7 | 47298 | 42564 | 44549 | 42480 | 42323 | 42243 |
| | 9 | 34656 | 34212 | 33998 | 34088 | 34206 | 34077 |
| f100_6 | 3 | 25845 | 24276 | 24000 | 24276 | 23853 | 24276 |
| | 5 | 38110 | 36355 | 37950 | 36355 | 37805 | 36355 |
| | 7 | 47147 | 43628 | 44283 | 43472 | 44018 | 43315 |
| | 9 | 33283 | 31612 | 32405 | 31661 | 32584 | 31612 |
| f100_7 | 3 | 24654 | 23160 | 23313 | 23160 | 23250 | 23160 |
| | 5 | 41755 | 35955 | 38660 | 35885 | 37215 | 35955 |
| | 7 | 45498 | 41410 | 40790 | 41534 | 41110 | 41420 |
| | 9 | 31346 | 29980 | 30341 | 29960 | 30523 | 29926 |
| f100_8 | 3 | 25851 | 25134 | 24624 | 25134 | 24303 | 25134 |
| | 5 | 44255 | 37830 | 39600 | 37795 | 38855 | 37795 |
| | 7 | 47100 | 44524 | 45133 | 44379 | 44489 | 44378 |
| | 9 | 31755 | 30950 | 31142 | 30851 | 31361 | 30950 |
| f100_9 | 3 | 25650 | 24723 | 24663 | 24723 | 24243 | 24723 |
| | 5 | 41340 | 37240 | 39420 | 37125 | 38870 | 37085 |
| | 7 | 46612 | 41613 | 43566 | 41991 | 42362 | 41552 |
| | 9 | 32966 | 32043 | 32617 | 32011 | 32820 | 32004 |
| f100_10 | 3 | 27003 | 24045 | 25563 | 24045 | 24864 | 24045 |
| | 5 | 44090 | 38345 | 40345 | 38435 | 39870 | 38405 |
| | 7 | 48742 | 46404 | 46234 | 46033 | 45444 | 46244 |
| | 9 | 32288 | 31853 | 31669 | 31482 | 31853 | 31711 |

Fonte: Próprio autor

ganhos, em que o híbrido mostrou melhor desempenho em 223 casos (62%).

Figura 15 – Desempenho dos procedimentos GGVNS e Híbrido para os 360 casos de testes



Fonte: Próprio autor

A hibridização com GGVNS demonstrou um resultado mais contundente em relação ao tamanho das instâncias. Nesta bateria de testes, o algoritmo original conseguiu se sobressair para instâncias de menor porte, mas com o aumento no número de vértices esse desempenho não se repetiu. Esse fato pode ser melhor observado no Apêndice C. Pode-se perceber também um comportamento um pouco pior do híbrido quando o caso de teste envolveu $\beta = 3$. Contudo, para configurações incluindo $\alpha = 0.1$ tiveram o melhor desempenho do híbrido. Portanto, a análise destes resultados demonstra que o algoritmo exato também proporciona à heurística GGVNS um espaço promissor de soluções a refinar. Buscando melhorar esse espaço, foram realizados testes considerando mais de uma solução inicial.

6.3.3 Quasi-exato Multistart com GGVNS e GGVNS

Nas Tabelas 10, 11 e 12 estão descritos os resultados dos testes realizados com mais de uma solução inicial produzida pelo B&B e com o procedimento GGVNS. Ao todo foram 433 soluções iniciais, 133 a mais que as 360 utilizadas nas hibridizações anteriores.

A primeira dessas tabelas, Tabela 10, apresenta os resultados para instâncias com 50 nós, e estes revelam um equilíbrio entre as abordagens. No entanto, neste caso, a vantagem pertence ao procedimento híbrido com *multistart*. Foram 49 ganhos contra 35 do GGVNS simples, e 36 empates. Contudo, assim como nos anteriores, o tamanho das instâncias fez diferença no comparativo.

A Tabela 11 expõe os resultados relacionados à instâncias de tamanho 75. Para este tamanho de instância, o híbrido *multistart* se sobressai com 86 ganhos, aproximadamente 71%

Tabela 10 – Desempenho dos métodos GGVNS e a proposta Híbrida com *Multistart* para instâncias de 50 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório($\alpha = 0.9$) | |
|-----------|---------|---------------------------|-------------------|----------------------------|-------------------|-----------------------------|-------------------|
| | | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> |
| f50_1 | 3 | 19062 | 18189 | 18078 | 18189 | 18021 | 18189 |
| | 5 | 28330 | 28320 | 28145 | 28100 | 28145 | 28200 |
| | 7 | 31300 | 30020 | 30707 | 30037 | 30180 | 30037 |
| | 9 | 22942 | 22942 | 22942 | 22942 | 22942 | 22942 |
| f50_2 | 3 | 19590 | 18591 | 18591 | 18591 | 18723 | 18591 |
| | 5 | 28660 | 28580 | 28580 | 28580 | 28570 | 28580 |
| | 7 | 31224 | 31288 | 30808 | 30629 | 30860 | 31060 |
| | 9 | 22081 | 22081 | 22081 | 22081 | 22081 | 22081 |
| f50_3 | 3 | 19116 | 19941 | 18201 | 19941 | 17799 | 19941 |
| | 5 | 29245 | 29420 | 28610 | 28535 | 28610 | 28535 |
| | 7 | 32695 | 33793 | 32643 | 33226 | 32390 | 33485 |
| | 9 | 22300 | 22300 | 22300 | 22300 | 22300 | 22300 |
| f50_4 | 3 | 16965 | 16179 | 15597 | 15555 | 15453 | 15453 |
| | 5 | 24955 | 24735 | 24835 | 24735 | 24735 | 24735 |
| | 7 | 29462 | 34004 | 29232 | 33677 | 28868 | 33984 |
| | 9 | 20778 | 20778 | 20778 | 20778 | 20778 | 20778 |
| f50_5 | 3 | 18540 | 17652 | 17610 | 17583 | 17544 | 17544 |
| | 5 | 27430 | 26950 | 26970 | 26950 | 27080 | 26970 |
| | 7 | 31563 | 29453 | 30187 | 30144 | 30101 | 29540 |
| | 9 | 23585 | 23585 | 23585 | 23585 | 23585 | 23585 |
| f50_6 | 3 | 17457 | 17928 | 16683 | 16914 | 16626 | 16641 |
| | 5 | 26295 | 26695 | 26385 | 26695 | 26295 | 26695 |
| | 7 | 30416 | 30196 | 30310 | 29992 | 30493 | 29920 |
| | 9 | 20709 | 20709 | 20709 | 20709 | 20709 | 20709 |
| f50_7 | 3 | 19776 | 17217 | 17406 | 17217 | 17241 | 17217 |
| | 5 | 27755 | 27035 | 26405 | 26095 | 26095 | 26095 |
| | 7 | 31968 | 29901 | 30585 | 29901 | 30605 | 29901 |
| | 9 | 22127 | 22127 | 22127 | 22127 | 22127 | 22127 |
| f50_8 | 3 | 17175 | 22551 | 16839 | 19698 | 16746 | 19527 |
| | 5 | 27510 | 27195 | 27310 | 27305 | 27220 | 27310 |
| | 7 | 34362 | 33913 | 33268 | 33339 | 33131 | 32978 |
| | 9 | 26867 | 26564 | 26564 | 26564 | 26574 | 26564 |
| f50_9 | 3 | 17802 | 18819 | 17124 | 18819 | 16968 | 18819 |
| | 5 | 27215 | 26965 | 27030 | 27000 | 26920 | 26960 |
| | 7 | 29617 | 29799 | 29466 | 29624 | 29613 | 29459 |
| | 9 | 22282 | 22282 | 22282 | 22282 | 22282 | 22282 |
| f50_10 | 3 | 19425 | 17949 | 18309 | 17949 | 18045 | 17949 |
| | 5 | 27900 | 27980 | 27925 | 27925 | 27890 | 27890 |
| | 7 | 29677 | 28819 | 28996 | 28832 | 28651 | 28819 |
| | 9 | 21715 | 21715 | 21715 | 21715 | 21715 | 21715 |

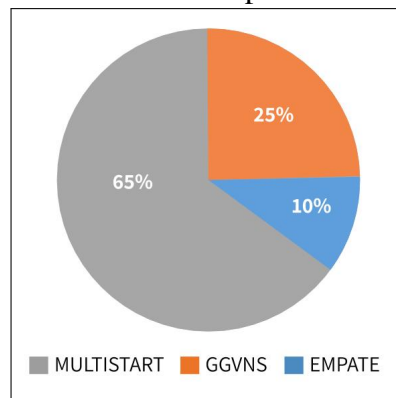
Fonte: Próprio autor

dos casos, enquanto o GGVNS conseguiu ser melhor em 33 casos. Apenas 1 empate ocorreu. Outra observação importante é o comportamento do híbrido para $\beta = 3$ destoando do resultado geral, com 21 ganhos do GGVNS contra 9 do híbrido *multistart* em um espaço de 30 casos de teste.

Na Tabela 12 são apresentados os resultados para instâncias com 100 vértices. Essa faixa produziu os melhores resultados, em que o híbrido *multistart* obteve 99 ganhos, correspondendo a cerca de 82% do total de casos nesta região. O procedimento GGVNS teve 21 ganhos. No entanto, assim como nos resultados para instâncias de 75 nós, $\beta = 3$ divergiu do resultado mais abrangente. Com este parâmetro, o GGVNS ganhou em 19 casos e o híbrido em 11.

Foram realizados 360 casos de teste, a mesma quantidade anterior. Entretanto, a característica *multistart* desta abordagem específica fez com fossem realizadas 1.299 execuções para o GGVNS. Considerando apenas os casos de teste, a Figura 16 apresenta o resultado total de ganhos. Mais uma vez a hibridização se sobressaiu e atingiu 234 ganhos, 65% dos casos.

Figura 16 – Desempenho dos procedimentos GGVNS e Híbrido *Multistart* para os 360 casos de testes



Fonte: Próprio autor

O uso de múltiplas soluções iniciais se mostrou eficaz, aumentando a vantagem do híbrido sobre o procedimento original. Como descrito anteriormente, o parâmetro β com valor 3 demonstrou comportamento diferente do observado no geral, um resultado, no total, de 54 a 33 ganhos para o GGVNS. O valor de $\alpha = 0.1$ novamente se mostrou aquele que rendeu maiores ganhos para o híbrido, 91 a 20. Tais considerações podem ser melhor observadas no Apêndice D.

Com esses resultados, fica mais uma vez constatada a capacidade do algoritmo exato de entregar à heurística um bom espaço de soluções. Além disso, foi demonstrado que usar o espaço investigado pelo B&B produz bons resultados.

Tabela 11 – Desempenho dos métodos GGVNS e a proposta Híbrida com *Multistart* para instâncias de 75 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório($\alpha = 0.9$) | |
|-----------|---------|---------------------------|-------------------|----------------------------|-------------------|-----------------------------|-------------------|
| | | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> |
| f75_1 | 3 | 22995 | 22470 | 21516 | 22470 | 21204 | 22470 |
| | 5 | 35215 | 33550 | 34795 | 33550 | 33775 | 33555 |
| | 7 | 40253 | 38966 | 39461 | 39115 | 38800 | 38933 |
| | 9 | 27479 | 27671 | 27628 | 27809 | 27567 | 27841 |
| f75_2 | 3 | 21456 | 19611 | 19548 | 19611 | 19470 | 19611 |
| | 5 | 32790 | 31435 | 31640 | 31050 | 31185 | 31655 |
| | 7 | 37099 | 35904 | 36186 | 35577 | 35778 | 35437 |
| | 9 | 27354 | 26974 | 27220 | 26972 | 27193 | 26998 |
| f75_3 | 3 | 23385 | 21288 | 21510 | 21288 | 21222 | 21057 |
| | 5 | 34600 | 33295 | 34560 | 33565 | 33270 | 33565 |
| | 7 | 40233 | 37435 | 38725 | 37310 | 37558 | 37369 |
| | 9 | 27841 | 27483 | 27545 | 27443 | 27697 | 27425 |
| f75_4 | 3 | 20085 | 22626 | 19755 | 22626 | 19719 | 22626 |
| | 5 | 40455 | 31420 | 32950 | 31410 | 31815 | 31410 |
| | 7 | 38922 | 35267 | 35613 | 35244 | 35294 | 35261 |
| | 9 | 26238 | 26213 | 26148 | 26138 | 26281 | 26101 |
| f75_5 | 3 | 20100 | 20382 | 19299 | 20382 | 19212 | 20382 |
| | 5 | 31190 | 29920 | 30320 | 29880 | 30020 | 29860 |
| | 7 | 37769 | 34109 | 36185 | 33979 | 35155 | 33883 |
| | 9 | 29801 | 28877 | 29436 | 28770 | 29530 | 28783 |
| f75_6 | 3 | 22146 | 22545 | 21108 | 21237 | 20691 | 20625 |
| | 5 | 35010 | 33650 | 33640 | 33715 | 33025 | 33240 |
| | 7 | 39547 | 38079 | 38451 | 38434 | 37218 | 37933 |
| | 9 | 26088 | 26121 | 26091 | 26121 | 26144 | 26040 |
| f75_7 | 3 | 21588 | 22353 | 20832 | 22353 | 20805 | 22353 |
| | 5 | 34150 | 32330 | 33025 | 32040 | 32400 | 32325 |
| | 7 | 37271 | 36068 | 37392 | 36191 | 36872 | 36281 |
| | 9 | 28841 | 28417 | 28617 | 28178 | 28440 | 28178 |
| f75_8 | 3 | 20286 | 19812 | 19491 | 19812 | 19359 | 19812 |
| | 5 | 33510 | 31130 | 31895 | 31055 | 31070 | 31020 |
| | 7 | 38100 | 36040 | 36803 | 36076 | 36808 | 36236 |
| | 9 | 26860 | 26543 | 26815 | 26543 | 26799 | 26543 |
| f75_9 | 3 | 21996 | 20394 | 20325 | 20394 | 19815 | 20394 |
| | 5 | 34235 | 30965 | 32055 | 31090 | 31460 | 31090 |
| | 7 | 37147 | 34831 | 36086 | 35002 | 34720 | 34791 |
| | 9 | 25221 | 24957 | 25177 | 24957 | 25281 | 24957 |
| f75_10 | 3 | 21630 | 20595 | 20559 | 20595 | 20046 | 20595 |
| | 5 | 33380 | 31120 | 32090 | 30975 | 31325 | 31055 |
| | 7 | 37717 | 35596 | 36969 | 35392 | 35559 | 35184 |
| | 9 | 28488 | 28471 | 28529 | 28382 | 28471 | 28471 |

Fonte: Próprio autor

Tabela 12 – Desempenho dos métodos GGVNS e a proposta Híbrida com *Multistart* para instâncias de 100 nós.

| Instância | β | Guloso ($\alpha = 0.1$) | | Mediano ($\alpha = 0.5$) | | Aleatório($\alpha = 0.9$) | |
|-----------|---------|---------------------------|-------------------|----------------------------|-------------------|-----------------------------|-------------------|
| | | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> | GGVNS | <i>Multistart</i> |
| f100_1 | 3 | 24630 | 23865 | 23907 | 23865 | 23400 | 23865 |
| | 5 | 42400 | 37345 | 39755 | 37150 | 39115 | 37225 |
| | 7 | 47538 | 43931 | 44175 | 43909 | 42951 | 43915 |
| | 9 | 30490 | 29672 | 30212 | 29566 | 29949 | 29577 |
| f100_2 | 3 | 26064 | 24225 | 24156 | 24225 | 23874 | 23604 |
| | 5 | 46390 | 37985 | 40875 | 38045 | 39090 | 37985 |
| | 7 | 50041 | 42851 | 45484 | 45425 | 44250 | 42917 |
| | 9 | 31580 | 31267 | 31453 | 31238 | 31557 | 31240 |
| f100_3 | 3 | 24972 | 24402 | 22914 | 23043 | 22584 | 23031 |
| | 5 | 42085 | 36120 | 37645 | 36105 | 36490 | 36215 |
| | 7 | 44862 | 38963 | 41099 | 38533 | 40888 | 38368 |
| | 9 | 29810 | 28227 | 28513 | 28190 | 28463 | 28201 |
| f100_4 | 3 | 26010 | 27084 | 24510 | 27084 | 24306 | 27084 |
| | 5 | 40325 | 37675 | 38515 | 37775 | 39225 | 37670 |
| | 7 | 47935 | 43400 | 45242 | 43144 | 44723 | 43006 |
| | 9 | 34639 | 33909 | 34077 | 33550 | 34539 | 33612 |
| f100_5 | 3 | 24111 | 24288 | 23235 | 24288 | 23208 | 24288 |
| | 5 | 37900 | 36900 | 37880 | 36735 | 36930 | 36825 |
| | 7 | 47298 | 41441 | 44549 | 41647 | 42323 | 41665 |
| | 9 | 34656 | 33979 | 33998 | 33979 | 34206 | 33979 |
| f100_6 | 3 | 25845 | 26487 | 24000 | 25644 | 23853 | 25173 |
| | 5 | 38110 | 36355 | 37950 | 36355 | 37805 | 36355 |
| | 7 | 47147 | 43606 | 44283 | 43920 | 44018 | 43749 |
| | 9 | 33283 | 31629 | 32405 | 31633 | 32584 | 31628 |
| f100_7 | 3 | 24654 | 23988 | 23313 | 23415 | 23250 | 23322 |
| | 5 | 41755 | 35985 | 38660 | 35845 | 37215 | 35860 |
| | 7 | 45498 | 41382 | 40790 | 40314 | 41110 | 40498 |
| | 9 | 31346 | 29911 | 30341 | 29911 | 30523 | 29926 |
| f100_8 | 3 | 25851 | 25071 | 24624 | 24789 | 24303 | 24672 |
| | 5 | 44255 | 37830 | 39600 | 37810 | 38855 | 37750 |
| | 7 | 47100 | 44321 | 45133 | 43973 | 44489 | 44356 |
| | 9 | 31755 | 31114 | 31142 | 30910 | 31361 | 30976 |
| f100_9 | 3 | 25650 | 25518 | 24663 | 24684 | 24243 | 24525 |
| | 5 | 41340 | 36955 | 39420 | 37095 | 38870 | 37130 |
| | 7 | 46612 | 41722 | 43566 | 41787 | 42362 | 41620 |
| | 9 | 32966 | 31873 | 32617 | 31828 | 32820 | 31849 |
| f100_10 | 3 | 27003 | 24045 | 25563 | 24045 | 24864 | 24045 |
| | 5 | 44090 | 38855 | 40345 | 38725 | 39870 | 39675 |
| | 7 | 48742 | 45395 | 46234 | 45341 | 45444 | 45319 |
| | 9 | 32288 | 31775 | 31669 | 31835 | 31853 | 31425 |

Fonte: Próprio autor

7 CONCLUSÃO

A presente pesquisa apresentou em seu decorrer três novas abordagens ao Problema do Ciclo Mediano sem Restrições de Capacidade. Tais abordagens envolveram 3 procedimentos: um exato, *Branch-and-Bound* (B&B), com execução limitada; e dois heurísticos, GRASP-ILS, de ANDRADE *et al.* (2016), e GGVNS, de Dias (2006).

As duas primeiras abordagens realizaram cada uma hibridização entre o exato e uma das heurísticas, onde o primeiro fornecia uma solução inicial para a última. O procedimento exato não foi executado completamente durante as hibridizações, sendo interrompido ao atingir um *GAP* percentual entre alguns de seus resultados internos. A segunda fase então refinou as soluções que provinham dessa interrupção.

A terceira abordagem, por sua vez, efetuou uma hibridização do método exato com a heurística GGVNS. Neste caso, contudo, o exato fornece mais de uma solução inicial para a heurística, estabelecendo um comportamento *multistart* para a mesma. Estas soluções foram obtidas durante toda a execução do B&B, diferentemente das anteriores que usavam apenas a solução produzida ao fim da execução do mesmo, quando este era interrompido.

Todas as três abordagens se sobressaíram quando comparadas aos algoritmos originais, obtendo mais de 60% de ganhos nos 360 casos de teste executados para cada uma delas e para as heurísticas originais. O comportamento padrão, que pode ser observado na comparação de desempenhos, demonstrou uma melhor performance das hibridizações a medida que o tamanho das instâncias aumentavam nos casos de teste. Explicitando, portanto, a partir dessas e de outras observações realizadas no capítulo anterior, que utilizar um procedimento *quasi-exato* pode produzir, e produz de fato, melhores soluções que usar apenas os métodos heurísticos.

Com objetivo de expandir o estudo sobre o PCMRC, uma possibilidade básica de trabalho futuro seria realizar testes mais abrangentes, com instâncias maiores e de diferentes classes, como, por exemplo, aquelas propostas em Reinelt (1991). Na fase heurística do *quasi-exato*, poderia ser usado paralelismo, principalmente na abordagem que possui um comportamento *multistart*. Além disso, repetir a forma de abordagem utilizando outros procedimentos heurísticos pode render resultados semelhantes ou melhores que os apresentados neste trabalho.

Na fase que usa o método exato, visando melhorar o tempo de execução do processo como um todo, um estudo mais detalhado da execução do B&B poderia sugerir um melhor *GAP* de interrupção e com isso reduzir esse tempo. Além disso, limitar o número de nós possíveis na árvore de busca também pode ajudar neste quesito.

REFERÊNCIAS

- ANDRADE, L.; CABRAL, L. d. A. F.; DIAS, T. C. S.; PINTO, E. R. Uma abordagem híbrida grasp-ils para o problema de projeto de redes com topologia anel-estrela. **TEMA (São Carlos)**, SciELO Brasil, v. 17, n. 1, p. 21–34, 2016.
- BARNHART, C.; JOHNSON, E. L.; NEMHAUSER, G. L.; SAVELSBERGH, M. W.; VANCE, P. H. Branch-and-price: Column generation for solving huge integer programs. **Operations research**, INFORMS, v. 46, n. 3, p. 316–329, 1998.
- BAUER, P. The circuit polytope: Facets. **Mathematics of Operations Research**, INFORMS, v. 22, n. 1, p. 110–145, 1997.
- BERNHARD, K.; VYGEN, J. Combinatorial optimization: Theory and algorithms. **Springer, Third Edition, 2005.**, 2008.
- BERTSCHEK, I.; BRIGLAUER, W.; HÜSCHEL RATH, K.; KAUF, B.; NIEBEL, T. The economic impacts of broadband internet: A survey. **Review of Network Economics**, De Gruyter, v. 14, n. 4, p. 201–227, 2015.
- BOLLOBÁS, B. Graph theory: an introductory course. New York, 1979.
- BONDY, J. A.; MURTY, U. S. R. *et al.* **Graph theory with applications**. [S.l.]: Citeseer, 1976. v. 290.
- CALVETE, H. I.; GALÉ, C.; IRANZO, J. A. An efficient evolutionary algorithm for the ring star problem. **European Journal of Operational Research**, Elsevier, v. 231, n. 1, p. 22–33, 2013.
- CHEN, X.; HU, X.; TANG, Z.; WANG, C.; ZHANG, Y. Algorithms for the ring star problem. In: SPRINGER. **International Conference on Combinatorial Optimization and Applications**. [S.l.], 2017. p. 3–16.
- CROES, G. A. A method for solving traveling-salesman problems. **Operations research**, INFORMS, v. 6, n. 6, p. 791–812, 1958.
- DAKIN, R. J. A tree-search algorithm for mixed integer programming problems. **The computer journal**, Oxford University Press, v. 8, n. 3, p. 250–255, 1965.
- DIAS, T. C. S. **ALGORITMOS HEURÍSTICOS E META-HEURÍSTICAS HÍBRIDAS APLICADAS AO PLANEJAMENTO DE UMA REDE DE TELECOMUNICAÇÕES COM TOPOLOGIA ANEL-ESTRELA**. Dissertação (Mestrado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2006.
- FISCHETTI, M.; GONZÁLEZ, J. J. S.; TOTH, P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. **Operations Research**, INFORMS, v. 45, n. 3, p. 378–394, 1997.
- GENDREAU, M.; POTVIN, J.-Y. *et al.* **Handbook of metaheuristics**. [S.l.]: Springer, 2010. v. 2.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. [S.l.]: Elsevier, 2005.

- GUIGNARD, M.; SPIELBERG, K. Search techniques with adaptive features for certain integer and mixed integer programming problems. In: **IFIP Congress (1)**. [S.l.: s.n.], 1968. p. 238–244.
- JORGENSEN, D. W.; VU, K. M. The ict revolution, world economic growth, and policy issues. **Telecommunications Policy**, Elsevier, v. 40, n. 5, p. 383–397, 2016.
- JOURDAN, L.; BASSEUR, M.; TALBI, E.-G. Hybridizing exact methods and metaheuristics: A taxonomy. **European Journal of Operational Research**, Elsevier, v. 199, n. 3, p. 620–629, 2009.
- LABBÉ, M.; LAPORTE, G.; MARTÍN, I. R.; GONZÁLEZ, J. J. S. The median cycle problem. Citeseer, 1999.
- LABBÉ, M.; LAPORTE, G.; MARTÍN, I. R.; GONZALEZ, J. J. S. The ring star problem: Polyhedral analysis and exact algorithm. **Networks**, Wiley Online Library, v. 43, n. 3, p. 177–189, 2004.
- LAND, A.; DOIG, A. Branch and bound algorithm. **discrete programming**, 1960.
- LEE, Y.; CHIU, S. Y.; SANCHEZ, J. A branch and cut algorithm for the steiner ring star problem. **Management Science and Financial Engineering**, v. 4, n. 1, p. 21–34, 1998.
- MARTÍN, I. R. **Cycle location problems**. Tese (Doutorado) — Universidad de La Laguna, 2000.
- OSMAN, I. H.; KELLY, J. P. Meta-heuristics theory and applications. **Journal of the Operational Research Society**, Taylor & Francis, v. 48, n. 6, p. 657–657, 1997.
- PADBERG, M.; RINALDI, G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. **SIAM review**, SIAM, v. 33, n. 1, p. 60–100, 1991.
- PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization: algorithms and complexity**. [S.l.]: Courier Corporation, 1998.
- PÉREZ, J. A. M.; MORENO-VEGA, J. M.; MARTIN, I. R. Variable neighborhood tabu search and its application to the median cycle problem. **European Journal of Operational Research**, Elsevier, v. 151, n. 2, p. 365–378, 2003.
- PUCHINGER, J.; RAIDL, G. R. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: SPRINGER. **International Work-Conference on the Interplay Between Natural and Artificial Computation**. [S.l.], 2005. p. 41–53.
- REINELT, G. TSPLIB—a traveling salesman problem library. **ORSA journal on computing, INFORMS**, v. 3, n. 4, p. 376–384, 1991.
- RIBEIRO, C. C. Metaheuristics and applications. **Advanced School on Artificial Intelligence, Estoril, Portugal**, 1996.
- RUOHONEN, K. Graph theory. tampereen teknillinen yliopisto. originally titled graafiteoria, lecture notes translated by tamminen, j., lee, k. **C. and Piché, R**, 2013.
- VAZIRANI, V. V. **Approximation algorithms**. [S.l.]: Springer Science & Business Media, 2013.

World Bank. **World Development Indicators: Power and communications**. 2017. Disponível em: <<http://wdi.worldbank.org/table/5.11>>. Acessado em: 04 de abril de 2019.

APÊNDICE A – RESULTADOS DOS EXPERIMENTOS EXATOS

As Tabelas 13 a 16 expõem os resultados completos dos experimentos exatos.

Tabela 13 – Desempenho do método exato para instâncias de 10 e 20 nós.

| Instância | β | F.O. | Tempo | Instância | β | F.O. | Tempo |
|------------------|---------|-------------|--------------|------------------|---------|-------------|--------------|
| f10_1 | 3 | 7590 | 0.265674 | f20_1 | 3 | 10191 | 1.073547 |
| | 5 | 12295 | 0.575765 | | 5 | 16195 | 5.407932 |
| | 7 | 12212 | 0.218831 | | 7 | 16295 | 1.526829 |
| | 9 | 8364 | 0.085475 | | 9 | 9100 | 0.547018 |
| f10_2 | 3 | 8985 | 0.065407 | f20_2 | 3 | 10800 | 0.496155 |
| | 5 | 13600 | 0.365525 | | 5 | 17075 | 3.805379 |
| | 7 | 14239 | 0.208303 | | 7 | 19946 | 4.753756 |
| | 9 | 8083 | 0.078815 | | 9 | 12085 | 0.853816 |
| f10_3 | 3 | 8241 | 0.067926 | f20_3 | 3 | 10125 | 0.248569 |
| | 5 | 12315 | 0.179517 | | 5 | 16270 | 3.373756 |
| | 7 | 12307 | 0.209320 | | 7 | 19444 | 3.357496 |
| | 9 | 8459 | 0.060974 | | 9 | 15967 | 1.522541 |
| f10_4 | 3 | 9423 | 0.065582 | f20_4 | 3 | 11814 | 0.874593 |
| | 5 | 14485 | 0.343298 | | 5 | 19315 | 4.371568 |
| | 7 | 15247 | 0.291596 | | 7 | 20517 | 4.014866 |
| | 9 | 11029 | 0.187159 | | 9 | 11540 | 0.409176 |
| f10_5 | 3 | 8823 | 0.060276 | f20_5 | 3 | 10614 | 0.962490 |
| | 5 | 14705 | 0.519822 | | 5 | 16140 | 8.198733 |
| | 7 | 16754 | 0.249003 | | 7 | 16298 | 2.485282 |
| | 9 | 11338 | 0.059160 | | 9 | 8835 | 0.544075 |
| f10_6 | 3 | 9321 | 0.067392 | f20_6 | 3 | 11361 | 0.683849 |
| | 5 | 15495 | 0.397567 | | 5 | 18045 | 6.388882 |
| | 7 | 15521 | 0.282005 | | 7 | 19818 | 3.737311 |
| | 9 | 9420 | 0.035929 | | 9 | 12721 | 0.437894 |
| f10_7 | 3 | 8916 | 0.208124 | f20_7 | 3 | 10422 | 0.342391 |
| | 5 | 14455 | 0.591224 | | 5 | 17305 | 1.428608 |
| | 7 | 15282 | 0.352879 | | 7 | 20520 | 5.695849 |
| | 9 | 10926 | 0.059024 | | 9 | 11581 | 0.854021 |
| f10_8 | 3 | 9210 | 0.185576 | f20_8 | 3 | 10131 | 0.462542 |
| | 5 | 13525 | 0.414185 | | 5 | 16045 | 4.462613 |
| | 7 | 13929 | 0.239565 | | 7 | 18615 | 4.612503 |
| | 9 | 10323 | 0.064742 | | 9 | 10860 | 0.798571 |
| f10_9 | 3 | 10698 | 0.123052 | f20_9 | 3 | 10659 | 0.423610 |
| | 5 | 16475 | 0.742183 | | 5 | 16085 | 4.632155 |
| | 7 | 14684 | 0.104126 | | 7 | 17202 | 4.461471 |
| | 9 | 10888 | 0.080329 | | 9 | 11352 | 0.426782 |
| f10_10 | 3 | 8199 | 0.108896 | f20_10 | 3 | 12801 | 0.422678 |
| | 5 | 13050 | 0.401911 | | 5 | 18420 | 3.359611 |
| | 7 | 13574 | 0.117621 | | 7 | 19688 | 1.736104 |
| | 9 | 8898 | 0.078614 | | 9 | 12535 | 0.344136 |

Tabela 14 – Desempenho do método exato para instâncias de 30 e 40 nós.

| Instância | β | F.O. | Tempo | Instância | β | F.O. | Tempo |
|------------------|---------------------------|-------------|--------------|------------------|---------------------------|-------------|--------------|
| f30_1 | 3 | 12375 | 1.593022 | f40_1 | 3 | 15573 | 15.089334 |
| | 5 | 19605 | 10.670385 | | 5 | 24360 | 123.931765 |
| | 7 | 22578 | 27.742368 | | 7 | 27431 | 70.176755 |
| | 9 | 14052 | 6.554136 | | 9 | 17792 | 22.263454 |
| f30_2 | 3 | 12729 | 4.154102 | f40_2 | 3 | 14505 | 6.504449 |
| | 5 | 19635 | 34.635393 | | 5 | 22940 | 107.824289 |
| | 7 | 22458 | 22.398818 | | 7 | 25255 | 59.988719 |
| | 9 | 16292 | 7.798474 | | 9 | 17590 | 56.821558 |
| f30_3 | 3 | 13545 | 7.390425 | f40_3 | 3 | 15315 | 15.406288 |
| | 5 | 21335 | 29.487093 | | 5 | 23545 | 78.128692 |
| | 7 | 24793 | 19.298541 | | 7 | 25789 | 85.394611 |
| | 9 | 13435 | 3.712943 | | 9 | 17530 | 81.522609 |
| f30_4 | 3 | 11856 | 4.973851 | f40_4 | 3 | 15459 | 3.481477 |
| | 5 | 18040 | 48.384607 | | 5 | 24585 | 63.775716 |
| | 7 | 20460 | 27.824332 | | 7 | 27517 | 90.998585 |
| | 9 | 12574 | 9.895532 | | 9 | 17145 | 34.278218 |
| f30_5 | 3 | 13419 | 6.657691 | f40_5 | 3 | 15195 | 29.677099 |
| | 5 | 21520 | 49.067244 | | 5 | 24345 | 98.956625 |
| | 7 | 24825 | 40.894121 | | 7 | 28268 | 319.736658 |
| | 9 | 16701 | 9.068120 | | 9 | 21159 | 115.300600 |
| f30_6 | 3 | 13833 | 2.296137 | f40_6 | 3 | 15297 | 5.135858 |
| | 5 | 21905 | 3.448789 | | 5 | 24380 | 55.924539 |
| | 7 | 24807 | 17.819100 | | 7 | 27480 | 71.922674 |
| | 9 | 16708 | 10.511764 | | 9 | 19143 | 51.626864 |
| f30_7 | 3 | 12945 | 5.192374 | f40_7 | 3 | 13674 | 26.548479 |
| | 5 | 19655 | 27.494741 | | 5 | 21905 | 87.030584 |
| | 7 | 22383 | 23.243347 | | 7 | 25973 | 142.461488 |
| | 9 | 15368 | 33.676252 | | 9 | 21584 | 340.112923 |
| f30_8 | 3 | 14013 | 5.813029 | f40_8 | 3 | 15105 | 19.194415 |
| | 5 | 22220 | 42.066401 | | 5 | 23665 | 90.954541 |
| | 7 | 22440 | 18.096988 | | 7 | 26830 | 148.413137 |
| | 9 | 13401 | 5.473640 | | 9 | 18461 | 200.348946 |
| f30_9 | 3 | 12924 | 2.679813 | f40_9 | 3 | 15573 | 11.863960 |
| | 5 | 19845 | 17.816602 | | 5 | 24980 | 95.544278 |
| | 7 | 22718 | 22.199563 | | 7 | 29393 | 53.145487 |
| | 9 | 15026 | 4.234421 | | 9 | 24120 | 171.048726 |
| f30_10 | 3 | 12954 | 1.286414 | f40_10 | 3 | 16632 | 13.983984 |
| | 5 | 19675 | 5.586316 | | 5 | 24395 | 87.846917 |
| | 7 | 22791 | 18.294899 | | 7 | 26323 | 108.623991 |
| | 9 | 15279 | 7.859862 | | 9 | 17400 | 128.210596 |

Fonte: Próprio autor

Tabela 15 – Desempenho do método exato para instâncias de 50 e 75 nós.

| Instância | β | F.O. | Tempo | Instância | β | F.O. | Tempo |
|------------------|---------------------------|-------------|--------------|------------------|---------------------------|-------------|--------------|
| f50_1 | 3 | 18021 | 82.017094 | f75_1 | 3 | 20985 | 620.371956 |
| | 5 | 28100 | 437.789128 | | 5 | 33520 | 1075.485207 |
| | 7 | 30020 | 206.648440 | | 7 | 37919 | 10751.857189 |
| | 9 | 22942 | 406.258934 | | 9 | 27479 | 37716.540680 |
| f50_2 | 3 | 18591 | 30.928134 | f75_2 | 3 | 19428 | 445.668224 |
| | 5 | 28510 | 335.077003 | | 5 | 30745 | 1537.959312 |
| | 7 | 30629 | 178.795412 | | 7 | 35032 | 14277.346404 |
| | 9 | 22081 | 709.940326 | | | | |
| f50_3 | 3 | 17691 | 65.391909 | f75_3 | 3 | 20898 | 439.124839 |
| | 5 | 28535 | 284.942209 | | 5 | 33090 | 1340.176749 |
| | 7 | 32129 | 233.293151 | | 7 | 37096 | 12336.294373 |
| | 9 | 22295 | 475.038211 | | | | |
| f50_4 | 3 | 15453 | 70.876548 | f75_4 | 3 | 19479 | 788.530082 |
| | 5 | 24735 | 681.047185 | | 5 | 31285 | 2677.351209 |
| | 7 | 28788 | 697.672562 | | 7 | 34583 | 9380.996015 |
| | 9 | 20778 | 680.231587 | | 9 | 25855 | 47157.999434 |
| f50_5 | 3 | 17529 | 71.794471 | f75_5 | 3 | 19014 | 463.148034 |
| | 5 | 26950 | 292.191834 | | 5 | 29840 | 2960.861342 |
| | 7 | 29268 | 1053.556419 | | 7 | 33726 | 28005.014868 |
| | 9 | 23585 | 1032.653910 | | | | |
| f50_6 | 3 | 16626 | 28.568385 | f75_6 | 3 | 20373 | 959.277420 |
| | 5 | 26225 | 587.788702 | | 5 | 32420 | 1508.246961 |
| | 7 | 29841 | 671.839345 | | 7 | 36949 | 4858.429086 |
| | 9 | 20709 | 521.459816 | | 9 | 25962 | 36119.603312 |
| f50_7 | 3 | 17217 | 21.383882 | f75_7 | 3 | 20442 | 1362.213623 |
| | 5 | 26095 | 171.409931 | | 5 | 31810 | 1969.616459 |
| | 7 | 29901 | 147.365426 | | 7 | 35381 | 19155.455277 |
| | 9 | 22127 | 352.322556 | | | | |
| f50_8 | 3 | 16740 | 59.181188 | f75_8 | 3 | 19143 | 563.795209 |
| | 5 | 27195 | 682.136786 | | 5 | 30875 | 2056.365897 |
| | 7 | 32603 | 800.427762 | | 7 | 35356 | 22380.863456 |
| | 9 | 26564 | 847.486344 | | 9 | 26543 | 59807.869334 |
| f50_9 | 3 | 16893 | 84.049048 | f75_9 | 3 | 19590 | 187.049366 |
| | 5 | 26900 | 480.158416 | | 5 | 30825 | 1289.205318 |
| | 7 | 29261 | 230.433367 | | 7 | 34028 | 8550.180487 |
| | 9 | 22282 | 770.243504 | | 9 | 24957 | 58786.828578 |
| f50_10 | 3 | 17934 | 27.524283 | f75_10 | 3 | 19914 | 841.710023 |
| | 5 | 27880 | 117.922206 | | 5 | 30930 | 3233.341745 |
| | 7 | 28651 | 193.507376 | | 7 | 34728 | 37463.663068 |
| | 9 | 21715 | 601.695965 | | | | |

Fonte: Próprio autor

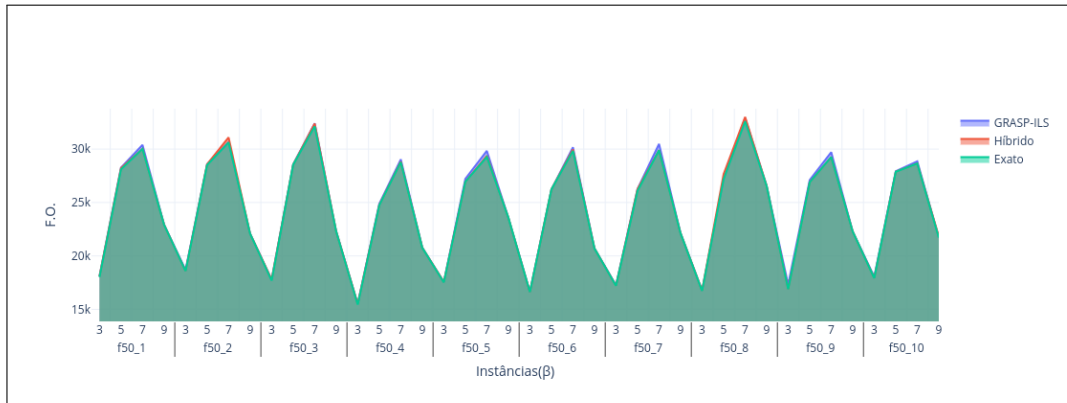
Tabela 16 – Desempenho do método exato para instâncias de 100 e 125 nós.

| Instância | β | F.O. | Tempo | Instância | β | F.O. | Tempo |
|------------------|---------|-------------|--------------|------------------|---------|-------------|--------------|
| f100_1 | 3 | 22845 | 4350.048117 | f125_1 | 3 | 25473 | 17612.621328 |
| | 5 | 36525 | 10648.814026 | | 5 | 40260 | 41628.667908 |
| f100_2 | 3 | 23523 | 1676.457763 | f125_2 | 3 | 25857 | 26480.807597 |
| | 5 | 37465 | 7960.270833 | | 5 | 40135 | 26056.939175 |
| f100_3 | 3 | 21987 | 3030.052436 | f125_3 | 3 | 25023 | 8969.558062 |
| | 5 | 34750 | 19613.934838 | | 5 | 40275 | 26488.863427 |
| | 7 | 37900 | 47136.138583 | | | | |
| f100_4 | 3 | 23856 | 4461.055003 | f125_4 | 3 | 24789 | 21087.317018 |
| | 5 | 37345 | 11644.579823 | | 5 | 39430 | 41605.744025 |
| f100_5 | 3 | 22767 | 3767.165776 | f125_5 | 3 | 25884 | 5547.137255 |
| | 5 | 35730 | 24410.360812 | | 5 | 41675 | 34373.232402 |
| f100_6 | 3 | 23346 | 3931.128146 | f125_6 | 3 | 26343 | 6892.075201 |
| | 5 | 36355 | 8120.770617 | | 5 | 41740 | 30679.506044 |
| f100_7 | 3 | 22713 | 5009.252419 | f125_7 | 3 | 26421 | 19342.094829 |
| | 5 | 35780 | 7668.390983 | | 5 | 42220 | 33126.142481 |
| | 7 | 38910 | 65692.999630 | | | | |
| f100_8 | 3 | 23976 | 3774.484653 | f125_8 | 3 | 25926 | 15698.902683 |
| | 5 | 37570 | 4766.523712 | | 5 | 41415 | 33238.078893 |
| f100_9 | 3 | 23784 | 2814.172766 | f125_9 | 3 | 25197 | 10800.288673 |
| | 5 | 36710 | 12376.781108 | | 5 | 40250 | 23142.517498 |
| f100_10 | 3 | 24012 | 2481.130174 | f125_10 | 3 | 26631 | 14162.925939 |
| | 5 | 38330 | 13641.648046 | | 5 | 42185 | 35269.1147 |

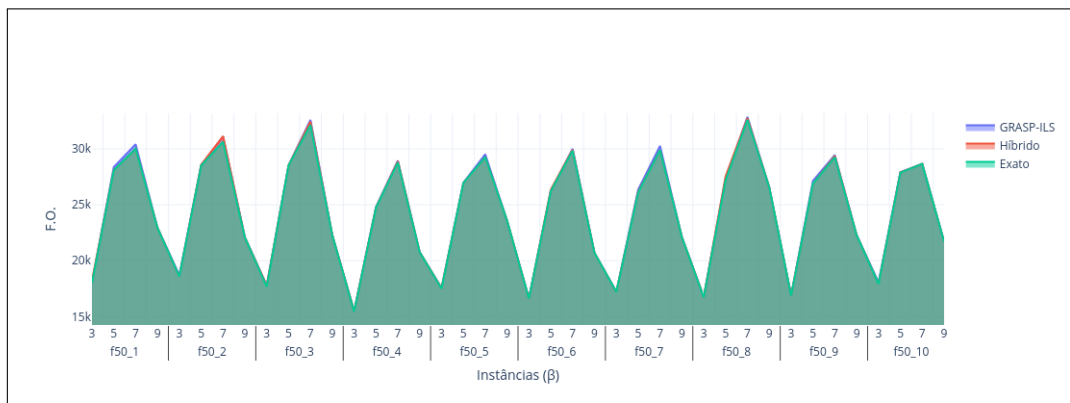
Fonte: Próprio autor

APÊNDICE B – COMPORTAMENTO DOS RESULTADOS ENVOLVENDO GRASP-ILS SIMPLES

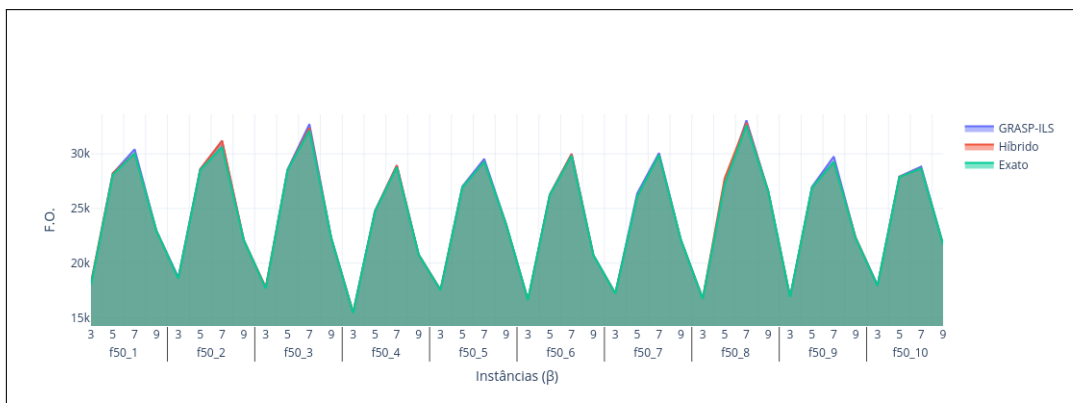
Figura 17 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



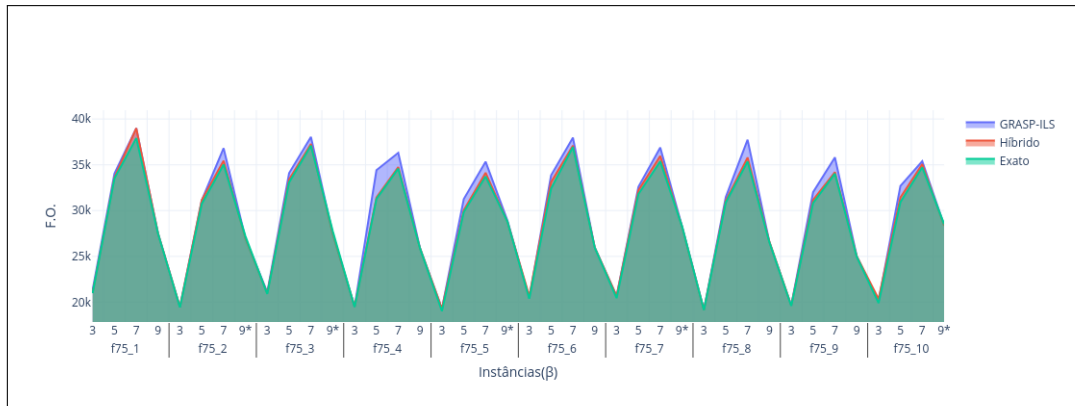
(b)



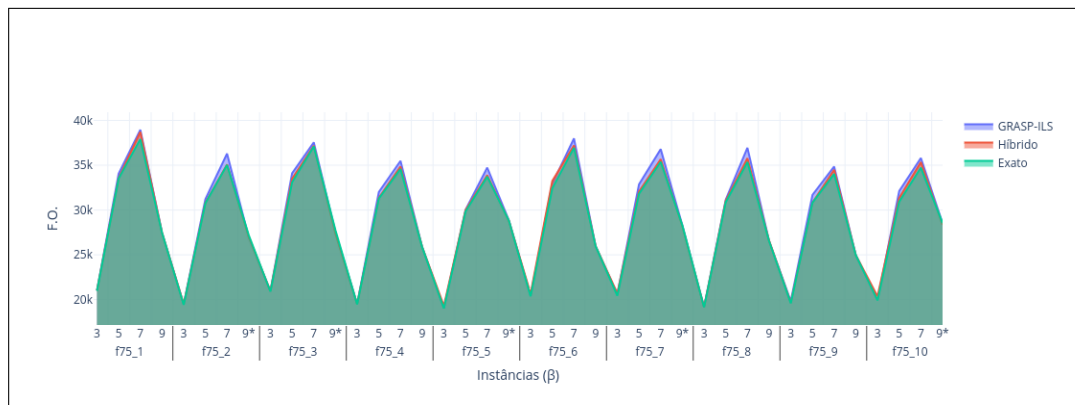
(c)

Fonte: Próprio autor.

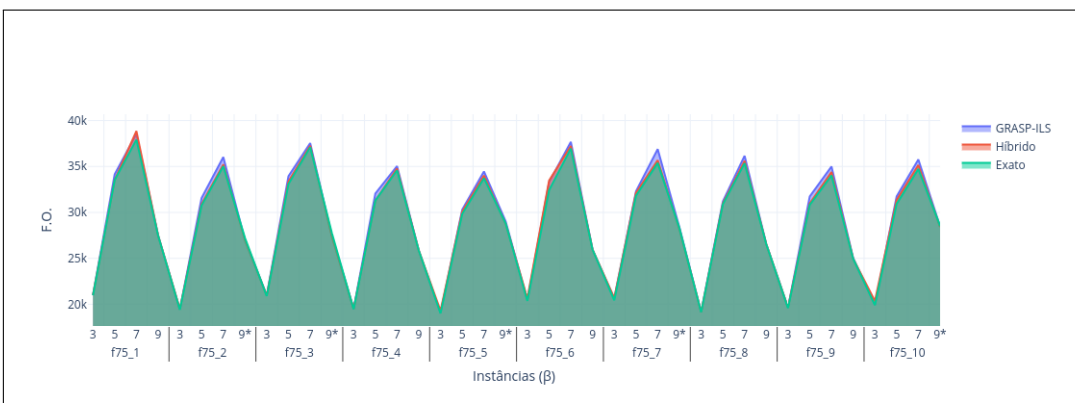
Figura 18 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



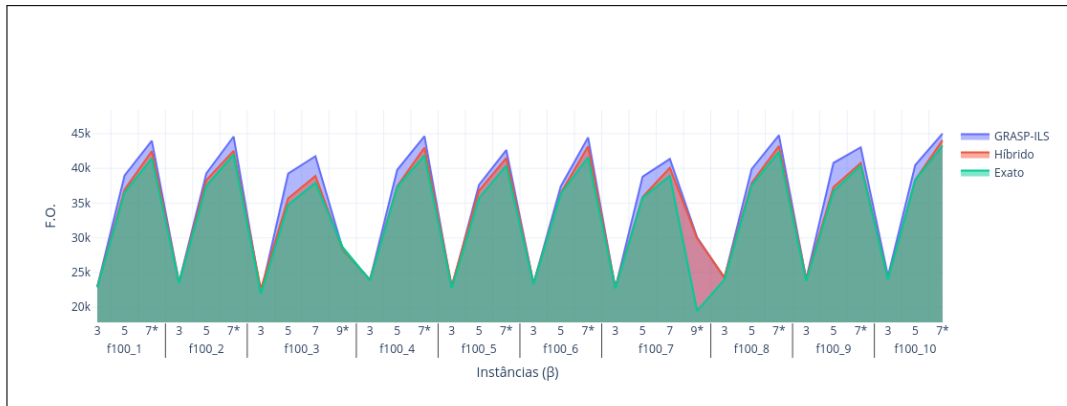
(b)



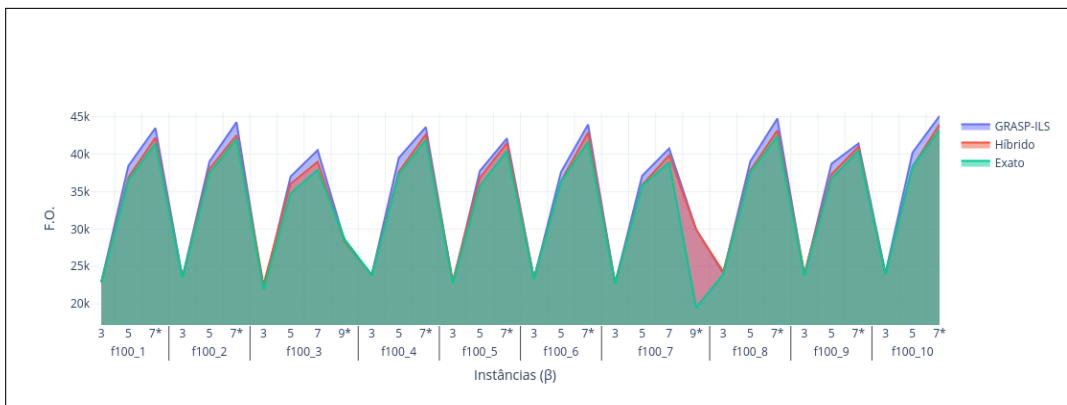
(c)

Fonte: Próprio autor.

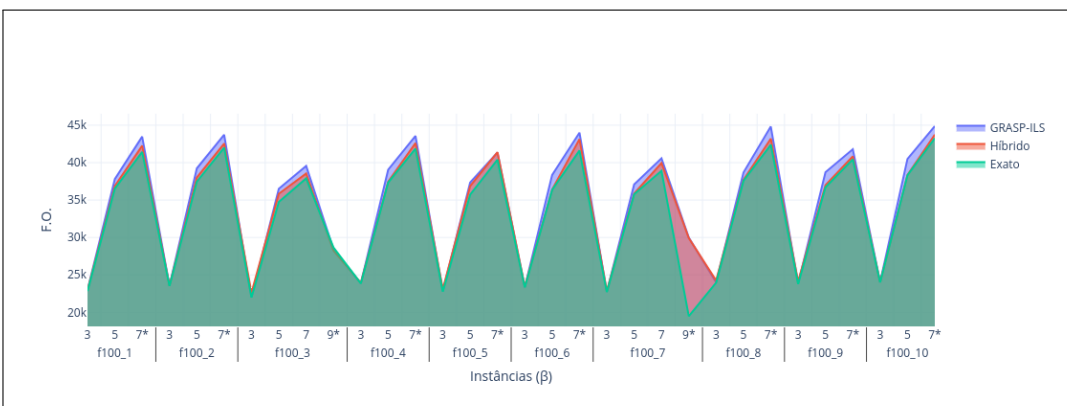
Figura 19 – Comportamento das Funções Objetivas obtidas para instâncias de 100 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



(b)

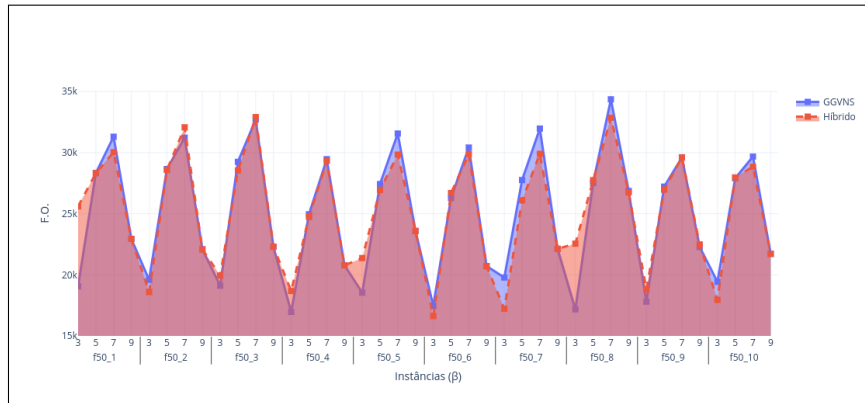


(c)

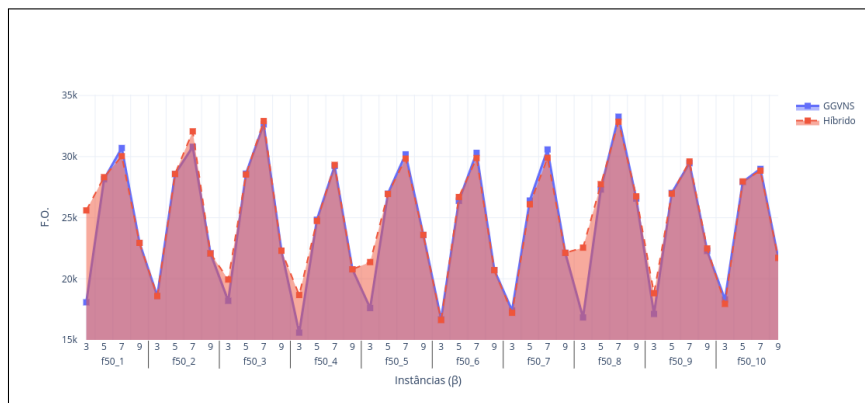
Fonte: Próprio autor.

APÊNDICE C – COMPORTAMENTO DOS RESULTADOS ENVOLVENDO GGVNS SIMPLES

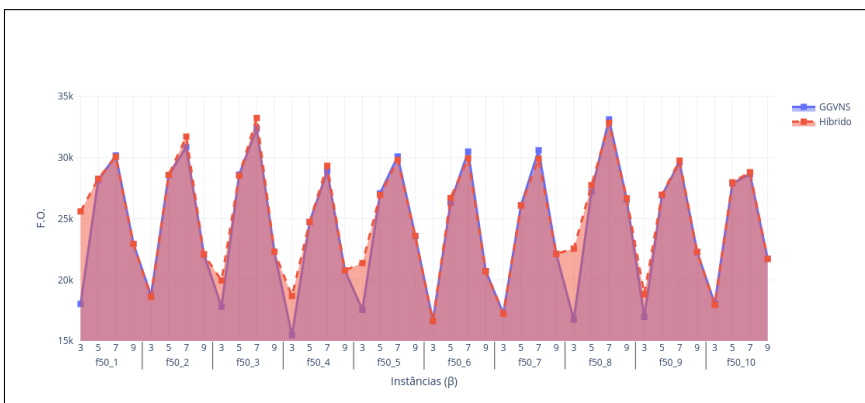
Figura 20 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



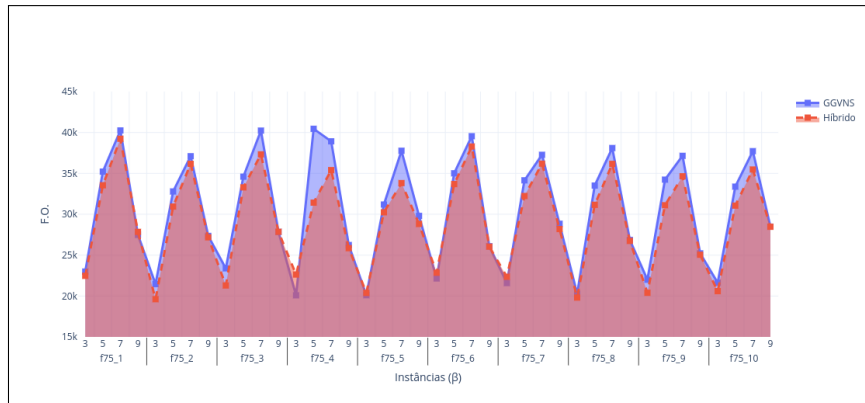
(b)



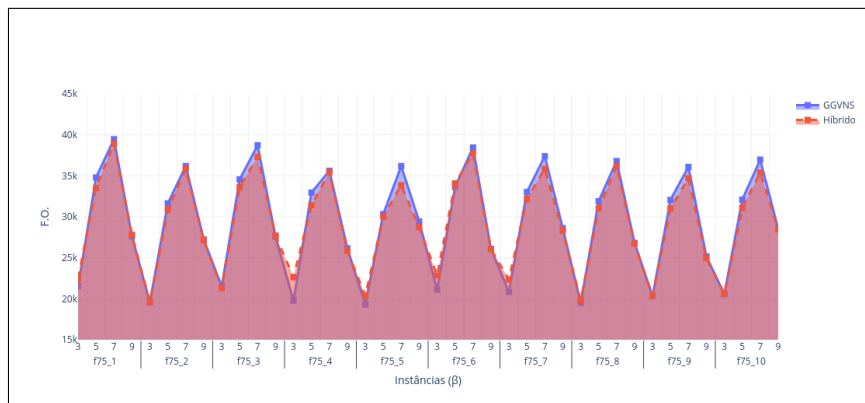
(c)

Fonte: Próprio autor.

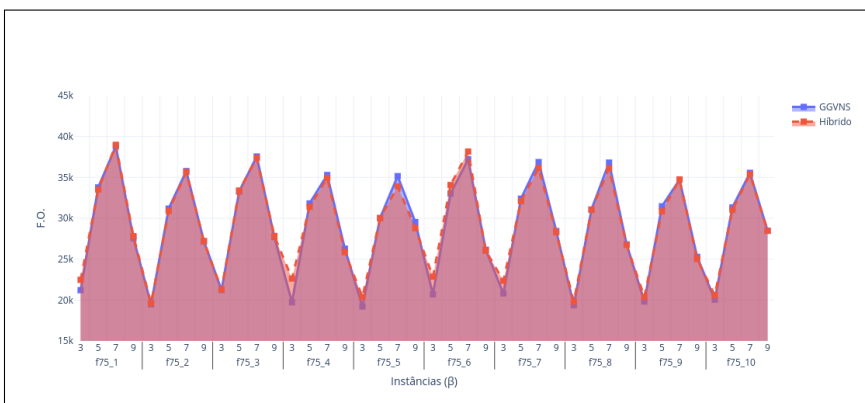
Figura 21 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



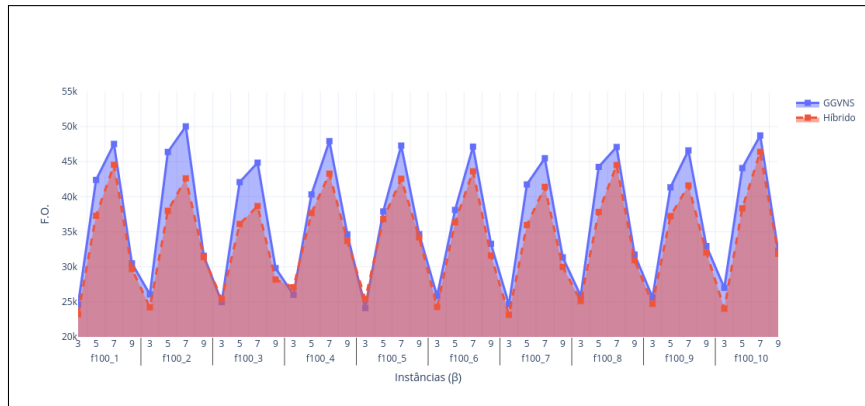
(b)



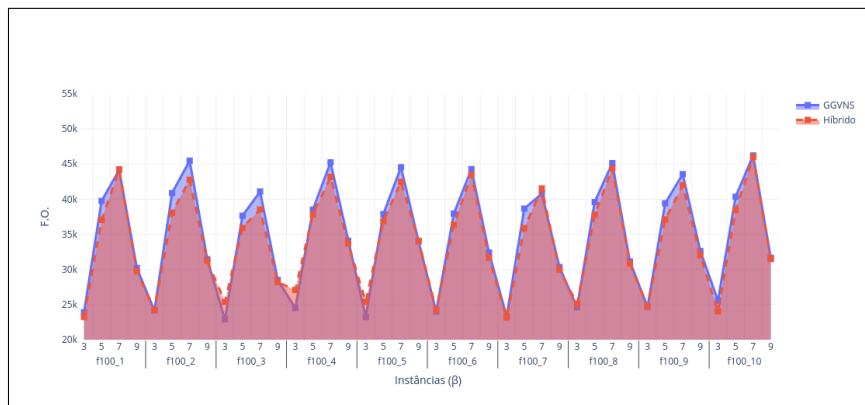
(c)

Fonte: Próprio autor.

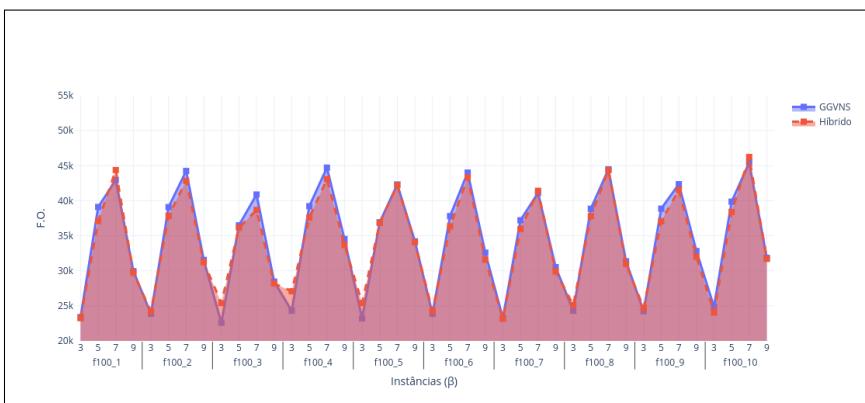
Figura 22 – Comportamento das Funções Objetivas obtidas para instâncias de 100 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



(b)

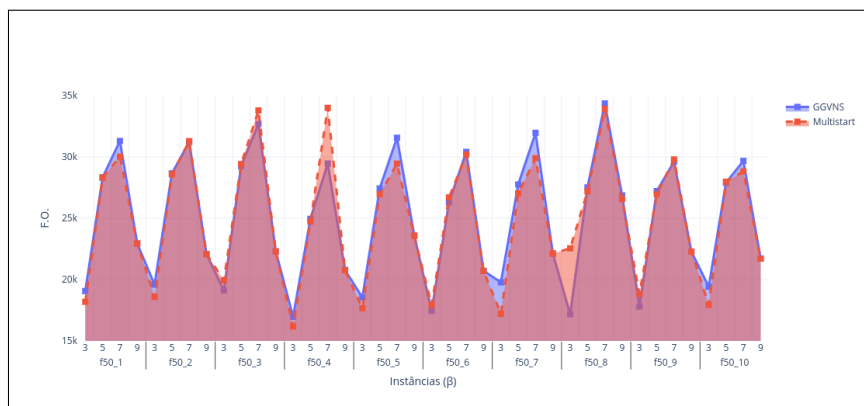


(c)

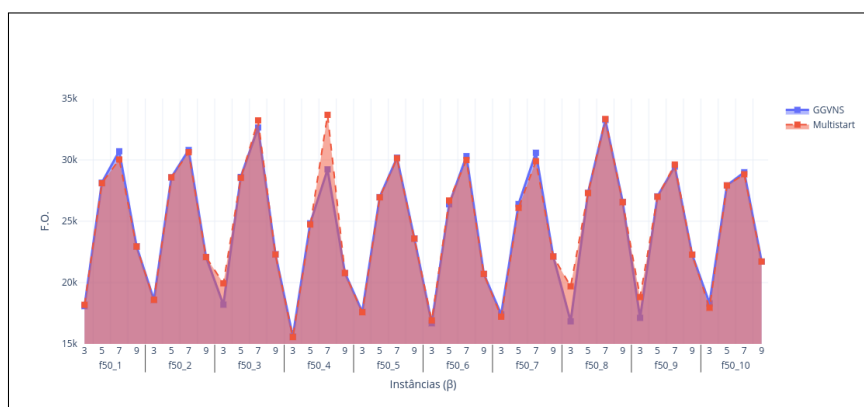
Fonte: Próprio autor.

**APÊNDICE D – COMPORTAMENTO DOS RESULTADOS ENVOLVENDO GGVNS E
MULTISTART**

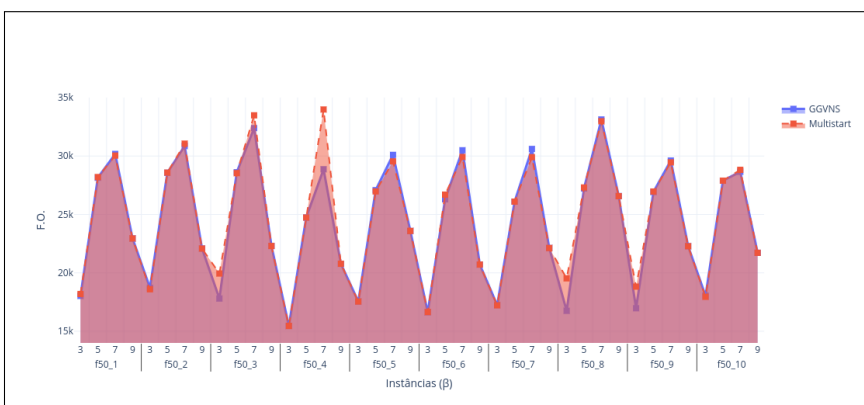
Figura 23 – Comportamento das Funções Objetivos obtidas para instâncias de 50 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



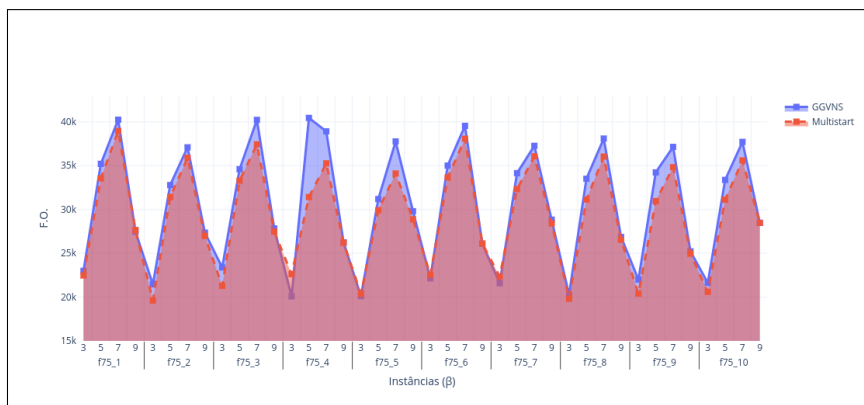
(b)



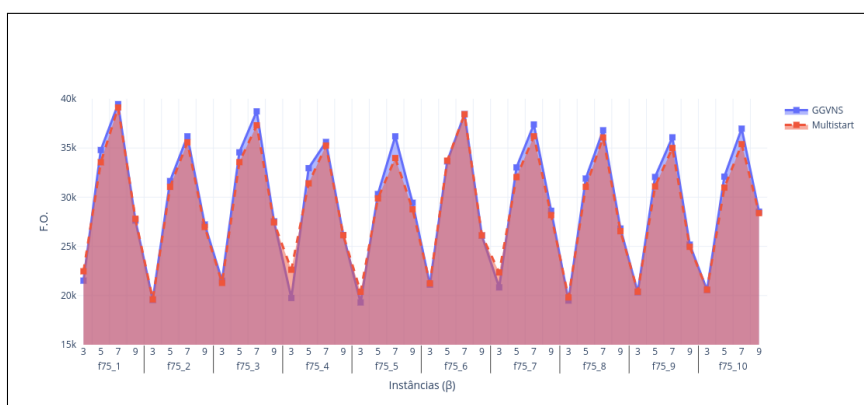
(c)

Fonte: Próprio autor.

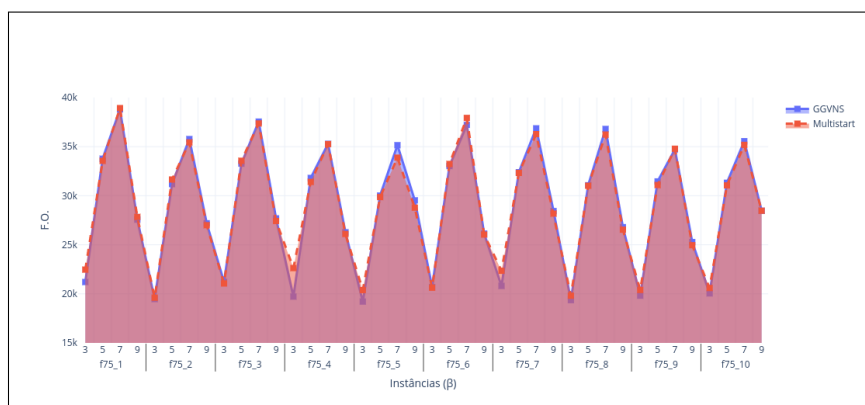
Figura 24 – Comportamento das Funções Objetivos obtidas para instâncias de 75 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



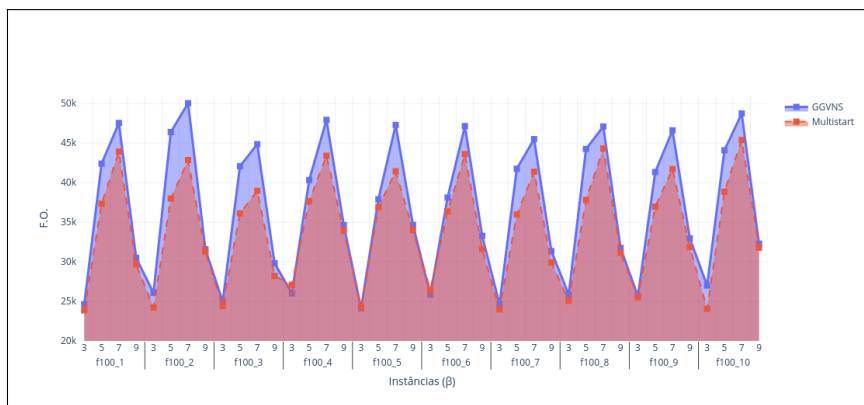
(b)



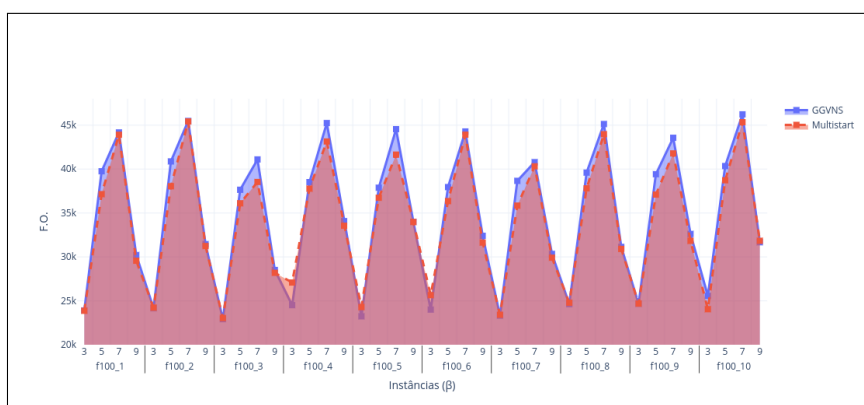
(c)

Fonte: Próprio autor.

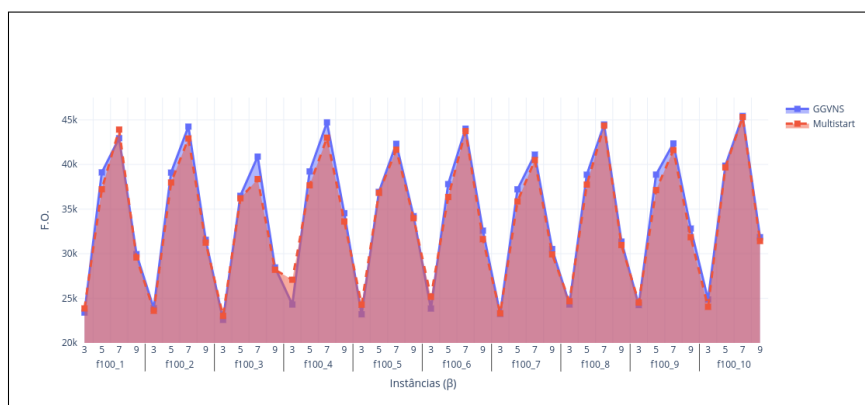
Figura 25 – Comportamento das Funções Objetivas obtidas para instâncias de 100 vértices (a) $\alpha = 0.1$. (b) $\alpha = 0.5$. (c) $\alpha = 0.9$.



(a)



(b)



(c)

Fonte: Próprio autor.