



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ERNANDO GOMES DE SOUSA

**MODELAGEM MULTIGRAFO, DECOMPOSIÇÃO E BRKGA PARA O PROBLEMA
DA ÁRVORE GERADORA GENERALIZADA MÍNIMA**

FORTALEZA

2019

ERNANDO GOMES DE SOUSA

MODELAGEM MULTIGRAFO, DECOMPOSIÇÃO E BRKGA PARA O PROBLEMA DA
ÁRVORE GERADORA GENERALIZADA MÍNIMA

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Algoritmos e Otimização

Orientador: Prof. Dr. Rafael Castro de Andrade

Coorientadora: Profa. Dra. Andréa Cynthia Santos Duhamel

FORTALEZA

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)

S696m Sousa, Ernando Gomes de.

Modelagem multigrafo, decomposição e BRKGA para o problema da árvore geradora generalizada mínima / Ernando Gomes de Sousa. – 2020.
75 f. : il.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2020.

Orientação: Prof. Dr. Rafael Castro de Andrade.

Coorientação: Profa. Dra. Andréa Cynthia Santos Duhamel.

1. Otimização. 2. Árvore geradora generalizada mínima. 3. Formulações de programação linear inteira mista. 4. Metaheurística BRKGA. 5. Decomposição de Benders. I. Título.

CDD 005

ERNANDO GOMES DE SOUSA

MODELAGEM MULTIGRAFO, DECOMPOSIÇÃO E BRKGA PARA O PROBLEMA DA
ÁRVORE GERADORA GENERALIZADA MÍNIMA

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Ciência da Computação. Área de Concentração: Algoritmos e Otimização

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Rafael Castro de Andrade (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Andréa Cynthia Santos Duhamel (Coorientadora)
Université du Havre

Prof. Dr. Manoel Bezerra Campêlo Neto
Universidade Federal do Ceará (UFC)

Abílio Pereira de Lucena Filho
Universidade Federal do Rio de Janeiro (UFRJ)

Philippe Yves Paul Michelin
Université D'Avignon (UNIV-AVIGNON)

À Deus, minha família, amigos...

AGRADECIMENTOS

Para produzir este trabalho precisei realizar uma longa caminhada, a qual teve diversas colaborações e por isso os meus agradecimentos.

Primeiramente a Deus por ter me dado força durante todo o caminho.

O apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 - e a Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - Brasil (FUNCAP).

Ao professor Rafael Castro de Andrade e à professora Andréa Cynthia Santos Duhamel pela paciente e sábia orientação e por ter acreditado em minha pessoa.

À minha família pelo incentivo, em especial meu pai Salomão Alves de Sousa, minha mãe Enoides Gomes de Sousa, meu irmão Davy Gomes de Sousa e minha prima Odaleia Alves da Costa.

Ao professor Dario José Aloise pelo incentivo para ingressar nesse programa de doutorado.

Ao IFMA - Campus São Raimundo das Mangabeiras - pelo incentivo à qualificação oportunizado, em especial aos meus amigos/diretores/professores Bruno Vicente, Jairo Ferraz, Jânio Fernandes, Sebastião Ricardo, Roberto Kennedy, Sigfran Santana, Tércio Santana e Thiago Reis.

Aos amigos do grupo de programação matemática, vulgo grupo dos orientandos do prof Rafael Andrade, Adriano Blue, Jefferson Gurguri, Mardson Ferreira, Rommel Dias e Sergio Filho; aos amigos do laboratório ParGO Fabio Dias, Joel Cruz, Rafael Teixeira e Tatiane Fernandes; aos amigos do laboratório Árida Arlino Magalhães, Antonio José [AJ], Bruno Leal, Gabriel Lopes, Narciso Arruda, Salomão Santos, Tiago Vinutto, Tulio Vidal e Wellington Franco; e aos amigos do grupo Ciroquers Daniel Ramos, Diego Farias, Henrique Bustamente, Madson Dias, Rafael, Renan Vieira e Thiago Alves. Com eles adquiri mais do que conhecimento acadêmico, adquiri conhecimento de vida.

À FUNCAP - Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico por ter me concedido uma bolsa de doutorado.

A todos os professores e demais servidores da UFC com os quais tive contanto durante esse período.

“Puru budu gudu budu, patiz quedi quedique-
edôôô!!!”

(Xand Avião)

RESUMO

Dado um grafo conexo, não direcionado e m -partido $G = (V_1 \cup V_2 \cup \dots \cup V_m, E)$, cujo conjunto de vértices é particionado em subconjuntos disjuntos V_1, V_2, \dots, V_m e cujas arestas de E conectam somente vértices de *clusters* distintos, o Problema da Árvore Geradora Generalizada Mínima (denotado por GMSTP, do inglês *Generalized Minimum Spanning Tree Problem*) consiste em encontrar uma árvore com exatamente $m - 1$ arestas, as quais conectam V_1, V_2, \dots, V_m através da seleção de um único vértice em cada *cluster*. O GMSTP encontra aplicações em problemas de modelagem de redes, agricultura irrigada, cidades inteligentes, *data science*, entre outros. Este trabalho apresenta para o GMSTP uma formulação multigrafo original, um procedimento para fixação de vértices que não possam pertencer a uma solução ótima, uma metaheurística Algoritmo Genético Baseado em Chave Aleatória (denotado por BRKGA, do inglês *Biased Random Key Genetic Algorithm*) e a aplicação do método de decomposição de Benders para uma formulação multifluxo da literatura. Resultados computacionais da formulação multigrafo superam os resultados das melhores formulações da literatura em termos de tempo de CPU para a obtenção de soluções comprovadamente ótimas para instâncias de teste da literatura e a decomposição de Benders aplicada neste trabalho apresenta os melhores resultados dentre todas as estratégias exatas analisadas. Além disso, o BRKGA apresenta melhor performance que metaheurísticas existentes para o problema em termos de qualidade de soluções viáveis obtidas para instâncias *benchmark* do GMSTP. Por fim, este trabalho abre novas direções de pesquisas para o desenvolvimento de algoritmos para problemas relacionados.

Palavras-chave: Otimização. Árvore geradora generalizada mínima. Formulações de programação linear inteira mista. Metaheurística BRKGA. Decomposição de Benders.

ABSTRACT

Given a connected, undirected, and m -partite graph $G = (V_1 \cup V_2 \cup \dots \cup V_m, E)$, with set of vertex partitioned into disjoint subsets V_1, V_2, \dots, V_m and whose edges of E connect only nodes of distinct clusters, the Generalized Minimum Spanning Tree Problem (GMSTP) looks for a minimum-cost tree with exactly $m - 1$ edges, connecting V_1, V_2, \dots, V_m through the selection of exactly one vertex of each cluster. The GMSTP finds applications in network design, irrigation agriculture, smart cities, data science, among others. This work presents, for the GMSTP, an original multigraph mathematical formulation, a new procedure for eliminating vertices proved to not belong to an optimal solution, a Biased Random Key Genetic Algorithm (BRKGA) metaheuristic, and a Benders decomposition method applied to an existing flow-based formulation. Computational results for the multigraph formulation has better results than for existing formulations for the problem and the Benders decomposition obtains the best results among all the exact strategies analyzed. Moreover, the BRKGA presents better performance than existing metaheuristics with respect to solution quality for benchmark instances of the problem. Finally, this work opens new directions for research and the development of algorithms for related problems.

Keywords: Optimization. Generalized minimum spanning tree. Mixed integer linear programming formulations. BRKGA-based Metaheuristic. Benders decomposition.

LISTA DE FIGURAS

Figura 1 – Uma instância em (A) e uma solução viável em (B) do GMSTP	14
Figura 2 – Uma instância em (A) e uma solução viável em (B) do L-GMSTP	15
Figura 3 – Grafo $G = (V, E)$ em (A), grafo global $G^* = (V^*, E^*)$ de G em (B), uma árvore geradora Z de G^* em (C) e uma árvore geradora generalizada de G com a estrutura Z em (D)	25
Figura 4 – Redução do problema de cobertura de vértices para o GMSTP. Em (A) tem-se uma instância do problema de cobertura de vértices de tamanho até $\omega = 2$ e, em (B), sua transformação (polinomial) em uma instância GST, com $\omega = 2$. As arestas tracejadas indicam uma GST	36
Figura 5 – Em (A) tem-se ilustração de arestas de conflito direto e em (B) tem-se a ilustração de arestas de conflito indireto	38
Figura 6 – Fixando vértices em uma instância do GMSTP	40
Figura 7 – Fixando arestas em uma instância do GMSTP	42
Figura 8 – Solução viável em (B), representada por um cromossomo em (C), para a instância do GMSTP em (A)	45
Figura 9 – Grafo de entrada $G = (V, E)$ em (A) e seu correspondente multigrafo $G^\circ = (V^\circ, E^\circ)$ em (B)	54
Figura 10 – Instância GMSTP em (A). Solução ótima em (E) de valor 7 para essa instância. Em (B), (C) e (D) soluções fracionárias da relaxação linear do modelo (\mathcal{F}_{MG}) fortalecido das restrições (5.10); $(5.10) \cap (2.32)$; e $(5.10) \cap (2.32) \cap (5.11)$, respectivamente	57

LISTA DE TABELAS

Tabela 1	– Resultados de instâncias <i>benchmark</i> usando clusterização em grade, com $\mu = 3$	51
Tabela 2	– Resultados de instâncias <i>benchmark</i> usando clusterização em grade, com $\mu = 5$	51
Tabela 3	– Resultados de instâncias <i>benchmark</i> usando clusterização em grade, com $\mu = 7$	52
Tabela 4	– Resultados de instâncias <i>benchmark</i> usando clusterização em grade, com $\mu = 10$	52
Tabela 5	– Características das 76 instâncias usadas nos experimentos computacionais e valores de solução viável obtidos com o BRKGA	68
Tabela 6	– Resultados computacionais para os modelos exatos considerando 40 instâncias de pequeno porte com até 300 vértices	69
Tabela 7	– Resultados computacionais para os modelos exatos considerando 36 instâncias de grande porte com mais de 300 e até 657 vértices	70
Tabela 8	– Resultados dos modelos exatos para o MIP do CPLEX considerando a solução do BRKGA como valor inicial para instâncias de pequeno porte	71
Tabela 9	– Resultados dos modelos exatos para o MIP do CPLEX considerando a solução do BRKGA como valor inicial para instâncias de grande porte.	71

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
B&B	<i>Branch and Bound</i>
B&C	<i>Branch and Cut</i>
GMSTP	Problema da Árvore Geradora Generalizada Mínima (do inglês <i>Generalized Minimum Spanning Tree Problem</i>)
GMSAP	Problema da Arborescência Geradora Generalizada de Custo Mínimo (do inglês <i>Generalized Minimum Spanning Arborescence Problem</i>)
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
L-GMSTP	Árvore Geradora Generalizada Mínima com Pelo Menos um Vértice Selecionado em Cada <i>Cluster</i> (do inglês <i>Least - Generalized Minimum Spanning Tree Problem</i>)
MST	Árvore Geradora Mínima (do inglês <i>Minimum Spanning Tree</i>)
MSTCCP	Problema da Árvore Geradora Mínima Sob Restrições de Conflitos (do inglês <i>Minimum Spanning Tree under Conflict Constraints Problem</i>)
PASD	Problema da Árvore de Steiner Direcionada
TS	<i>Tabu Search</i>
VND	<i>Variable Neighborhood Descent</i>
VNS	<i>Variable Neighborhood Search</i>

SUMÁRIO

1	INTRODUÇÃO	13
2	TRABALHOS RELACIONADOS	18
2.1	Formulações da literatura para o GMSTP	18
2.2	Algoritmo exato de programação dinâmica para o GMSTP	27
2.3	Metaheurísticas da literatura para o GMSTP	28
3	COMPLEXIDADE E PRÉ-PROCESSAMENTO DE INSTÂNCIAS DO GMSTP	34
3.1	Complexidade do problema	34
3.2	Redução polinomial em relação ao problema de conjunto independente de menor valor	37
3.3	Casos polinomiais do GMSTP	39
3.4	Propriedades para fixação de vértices e arestas em instâncias do GMSTP	39
3.4.1	<i>Fixando vértices em uma instância do GMSTP</i>	39
3.4.2	<i>Fixando arestas em uma instância do GMSTP</i>	41
4	UMA NOVA METAHEURÍSTICA BRKGA PARA O GMSTP	44
4.1	População inicial	44
4.2	Operadores genéticos	45
4.3	Operação global do BRKGA para o GMSTP	46
4.4	Busca local	47
4.5	Resultados computacionais obtidos com o BRKGA	48
5	NOVA FORMULAÇÃO MATEMÁTICA E UM ESQUEMA DE DE- COMPOSIÇÃO DE BENDERS PARA O GMSTP	53
5.1	Nova formulação polinomial baseada em multigrafo para o GMSTP . .	53
5.2	Decomposição de Benders	59
5.2.1	<i>Resultados computacionais para as principais formulações exatas</i>	61
6	CONSIDERAÇÕES FINAIS	72
	REFERÊNCIAS	74

1 INTRODUÇÃO

Um clássico problema de otimização combinatória definido em grafos é o problema da Árvore Geradora Mínima (denotado por MST, do inglês *Minimum Spanning Tree*), o qual busca uma árvore geradora de menor custo de um dado grafo. Uma árvore geradora T de um grafo conexo $G = (V, E)$ é um subgrafo conexo e acíclico, no qual o conjunto de arestas conecta todos os vértices de V . T^* é uma árvore geradora de custo mínimo de G se $\sum_{e \in T^*} c_e \leq \sum_{e \in T} c_e$, para toda árvore geradora T de G . Em geral, problemas de árvores geradoras modelam aplicações em redes e têm motivado diversos estudos teóricos e práticos, abordando complexidade, modelagem matemática e a construção de algoritmos eficientes para resolvê-los (SANTOS *et al.*, 2016).

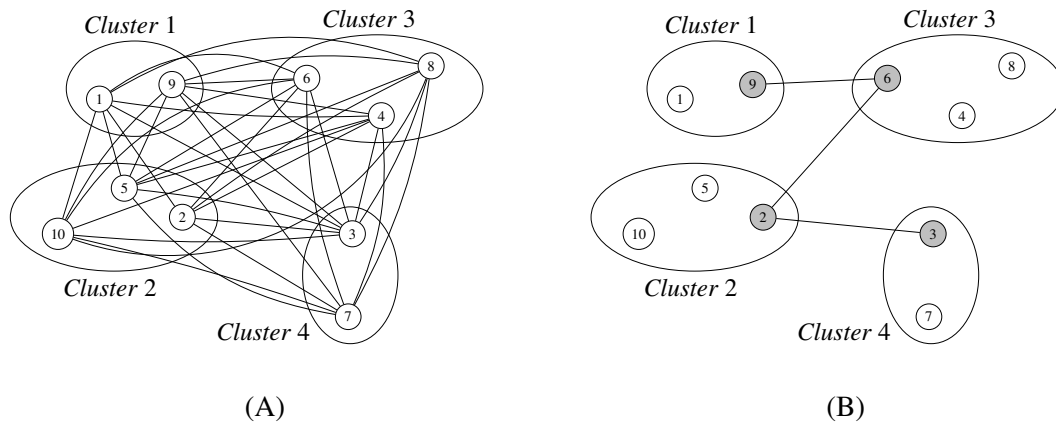
O problema de encontrar uma MST de um grafo G é especialmente simples por ser polinomial. Algoritmos bem conhecidos na literatura como o algoritmo de (KRUSKAL, 1956) e (PRIM, 1957) encontram em tempo polinomial uma MST de um grafo G . Todavia, quando restrições tais como conflito de arestas, diâmetro, grau dos vértices e outras restrições são incorporadas ao problema MST, frequentemente obtém-se um problema NP-Difícil, como é o caso do Problema da Árvore Geradora Mínima Sob Restrições de Conflito (DIAS; CAMPÊLO, 2016), Problema da Árvore Geradora Mínima com Restrição de Diâmetro (SANTOS, 2006) e o Problema da Árvore Geradora Mínima com Restrição de Grau (ANDRADE *et al.*, 2006). Além desses problemas, podemos citar também o Problema da Árvore Geradora Generalizada Mínima (MYUNG *et al.*, 1995), alvo de estudo deste trabalho, e o Problema da Árvore Geradora Generalizada Mínima com Pelo Menos um Vértice Selecionado em Cada *Cluster* (FEREMANS *et al.*, 2001).

O Problema da Árvore Geradora Generalizada Mínima (denotado por GMSTP, do inglês *Generalized Minimum Spanning Tree Problem*) é definido em um grafo conexo e m -partido $G = (V, E)$. O conjunto de vértices V é particionado em m *clusters* V_1, V_2, \dots, V_m , tal que $V = V_1 \cup V_2 \cup \dots \cup V_m$, com $V_r \cap V_q = \emptyset$, para todo $r \neq q$, com $r, q \in M = \{1, \dots, m\}$. O conjunto de arestas é tal que $E \subseteq \{\{i, j\} \mid i \in V_r, j \in V_q, \forall r \neq q, \text{ com } r, q \in M\}$. O termo $c_e = c_{ij} \in \mathbb{R}_+$ denota o custo de uma aresta $e = \{i, j\} \in E$. O GMSTP consiste em encontrar uma árvore de custo mínimo conectando os m *clusters*, sendo que o conjunto de extremidades dessas arestas incidem sobre um único vértice de cada *cluster*.

Na Figura 1 (A) tem-se a ilustração de uma instância do GMSTP. Existem arestas somente entre vértices de *clusters* diferentes. Não existe conexão entre os vértices de um mesmo *cluster*. Na Figura 1 (B) tem-se um exemplo de solução viável de Árvore Geradora Generalizada

(denotado por GST, do inglês *Generalized Spanning Tree*) para o grafo de entrada dado na Figura 1 (A). Nota-se na Figura 1 (B) que o número de arestas selecionadas é igual a $m - 1$, exatamente um vértice de cada *cluster* recebe incidência de arestas, existe um caminho entre qualquer par de *clusters* $r, q \in M$ e não existe ciclos na solução.

Figura 1 – Uma instância em (A) e uma solução viável em (B) do GMSTP

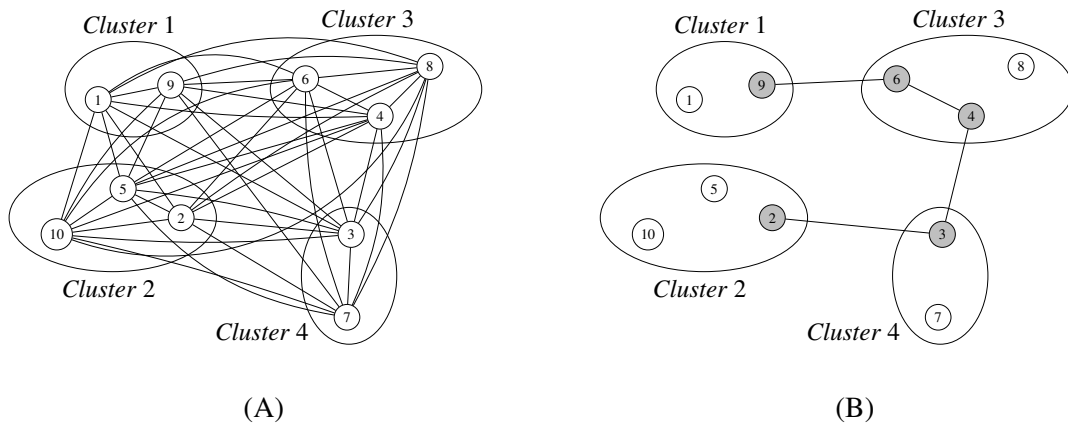


Fonte: autor.

No Problema da Árvore Geradora Generalizada Mínima com Pelo Menos um Vértice Selecionado em Cada *Cluster* (denotado por L-GMSTP, do inglês *Least - Generalized Minimum Spanning Tree Problem*) existem arestas entre vértices de *clusters* diferentes e vértices de um mesmo *cluster*. Na Figura 2 (A) tem-se a ilustração de uma instância do L-GMSTP. Na Figura 2 (B) tem-se um exemplo de solução viável do L-GMSTP para o grafo de entrada dado na Figura 2 (A). Segundo (HAOUARI; CHAOUACHI, 2006), o problema L-GMSTP é NP-Difícil.

O foco deste trabalho é o GMSTP, o qual tem aplicação em problemas de rede, agricultura de irrigação, cidades inteligentes, *data science*, entre outros. Em problemas de rede, por exemplo, o GMSTP pode aparecer quando um provedor de telecomunicações deseja construir links que interconectem redes locais já existentes em uma determinada região (provedor de longa distância que compra direitos de conexão de provedores locais). Em uma primeira estimativa, o custo de um centro de conexão em uma rede local é tão alto que o provedor de telecomunicações está restrito a pagar por exatamente um centro de conexão em cada rede local. Sendo assim,

Figura 2 – Uma instância em (A) e uma solução viável em (B) do L-GMSTP



Fonte: autor.

deve-se seleccionar centros de conexão, exatamente um em cada rede local, de modo a conectar todas as redes locais por uma infraestrutura de menor custo.

Na literatura podemos encontrar vários trabalhos relacionados ao GMSTP. Em (MYUNG *et al.*, 1995) são apresentadas quatro formulações matemáticas para o GMSTP. Duas dessas formulações são aplicadas a grafos não direcionados e possuem um número exponencial de restrições. As outras duas formulações são aplicadas a grafos direcionados, onde uma delas é exponencial no número de restrições e a outra possui um número polinomial de restrições e variáveis. Utilizando a relaxação linear desse último modelo, (MYUNG *et al.*, 1995) desenvolveu um algoritmo *Branch and Bound* (B&B) para encontrar soluções ótimas de instâncias do GMSTP.

Em (FEREMANS *et al.*, 2002) são apresentadas outras quatro formulações exponenciais no número de restrições para o GMSTP. Nesse trabalho, uma análise comparativa do politopo associado à relaxação linear dessas formulações com as de (MYUNG *et al.*, 1995) é realizada. Desses oito modelos matemáticos, demonstrou-se que quatro deles são equivalentes na relaxação linear e dominam os demais. Dessas quatro melhores formulações, uma delas é a formulação polinomial (em número de restrições e de variáveis) de (MYUNG *et al.*, 1995).

Um algoritmo aproximativo para o GMSTP foi desenvolvido por (POP, 2002). No pior caso ele tem uma solução limitada por $\omega(2 - 2/m) \times O^*$, onde ω é o número de vértices do maior *cluster* e O^* é o valor da solução ótima. Esse trabalho também apresentou uma versão bidirecional do modelo de multifluxo de (MYUNG *et al.*, 1995) e uma relaxação Lagrangeana utilizando esse modelo bidirecional.

Em (POP, 2002) também é apresentada uma formulação matemática que considera o grafo original $G = (V, E)$ de uma instância do GMSTP e um grafo simples denominado de

global $G^* = (V^*, E^*)$ de G , onde cada vértice de V^* é associado exclusivamente a um *cluster* de G e dois vértices de V^* são conectados por uma aresta em E^* se existir pelo menos uma aresta conectando seus respectivos *clusters* em G . Um modelo de árvore geradora é aplicado em G^* e restrições adicionais sobre G são acrescentadas nessa formulação para encontrar uma GST de G . Para um maior entendimento sobre o grafo global G^* de G , consultar a Figura 3 na página ??.

Um algoritmo *Branch and Cut* (B&C) e algumas desigualdades válidas para o GMSTP foram propostos por (FEREMANS *et al.*, 2004). Esse algoritmo B&C se utiliza de uma metaheurística *Tabu Search* para gerar uma solução viável inicial de boa qualidade, da formulação matemática de (FEREMANS *et al.*, 2002), e das desigualdades válidas propostas pela autora, utilizadas como planos de corte.

Além de métodos exatos, várias metaheurísticas foram desenvolvidas para o GMSTP, entre elas pode-se citar: *Tabu Search* (TS) (FEREMANS *et al.*, 2001), (GHOSH, 2003), (WANG *et al.*, 2006), (ÖNCAN *et al.*, 2008); *Variable Neighborhood Descent* (VND) e *Variable Neighborhood Search* (VNS) (GHOSH, 2003), (HU *et al.*, 2005); *Greedy Randomized Adaptive Search Procedure* (GRASP) (FERREIRA *et al.*, 2012); Algoritmo Genético (AG) (GOLDEN *et al.*, 2005), (CONTRERAS-BOLTON *et al.*, 2016) e um procedimento de duas fases (POP *et al.*, 2017).

Neste trabalho apresentamos uma nova estratégia exata de resolução do GMSTP, onde propomos uma formulação matemática que trabalha sobre um multigrafo $G^\circ = (V^\circ, E^\circ)$, obtido a partir do grafo original $G = (V, E)$ de uma instância GMSTP. Cada vértice de V° é associado exclusivamente a um *cluster* de G e cada aresta $e \in E^\circ$ é associada exclusivamente a uma aresta $e \in E$, com o mesmo custo c_e , onde $|E| = |E^\circ|$. Apresentamos também propriedades para as restrições dessa formulação multigrafo, mostrando inclusive uma comparação de resultados computacionais com e sem a remoção de algumas restrições. Além disso, comparamos essa formulação multigrafo com as formulações polinomiais de (MYUNG *et al.*, 1995) e (POP, 2002), as quais mostraram-se as melhores formulações em tempo de processamento que identificamos durante nossa pesquisa.

Desenvolvemos uma metaheurística do tipo Algoritmo Genético Baseado em Chave Aleatória (denotado por BRKGA, do inglês *Biased Random Key Genetic Algorithm*) para o GMSTP (vide (GONCALVES; RESENDE, 2011) para mais informações sobre o BRKGA), cujo principal objetivo foi o de tentar obter boas soluções de partida para as formulações testadas neste trabalho e verificar o desempenho das mesmas recebendo uma solução inicial.

No entanto, comparado a outras metaheurísticas desenvolvidas para o GMSTP, o BRKGA obteve um desempenho global superior para todas as instâncias utilizadas em nossa investigação, demonstrando ser competitivo e apresentando em alguns casos melhores resultados para algumas instâncias da literatura, como será relatado mais adiante.

Aplicamos também um método de decomposição de Benders à formulação polinomial multifluxo de (MYUNG *et al.*, 1995), a qual demonstrou ser uma boa estratégia para solucionar o GMSTP. Tanto quanto sabemos, essa decomposição supera os melhores resultados da literatura e a formulação que propomos. Apresentamos também um procedimento que procura identificar vértices que garantidamente não pertencem a uma solução ótima.

O restante deste trabalho é organizado como segue. No Capítulo 2 apresentamos algoritmos da literatura para solucionar o GMSTP, entre eles mostramos um algoritmo exato de programação dinâmica e metaheurísticas, além de modelos matemáticos para o problema. No Capítulo 3 discutimos a complexidade do GMSTP e particularidades do problema. No Capítulo 4 apresentamos a metaheurística BRKGA para o GMSTP. No Capítulo 5 apresentamos nossa formulação multigrafo, seguido de uma decomposição de Benders. No Capítulo 6 abordamos as considerações finais.

2 TRABALHOS RELACIONADOS

Neste capítulo são apresentados formulações matemáticas existentes na literatura, um algoritmo exato e metaheurísticas da literatura para o GMSTP. Estes não são aqui apresentados em maiores detalhes, pois os mesmos podem ser analisadas com mais exatidão nos seus trabalhos originais, os quais são devidamente referenciados.

Inicialmente, introduzimos alguns conceitos e notação usada nas formulações apresentadas neste capítulo. Para um dado grafo $G = (V, E)$ e um subconjunto de vértices $S \subseteq V$, considere o subconjunto de arestas do corte $(S, V \setminus S)$ como $\delta(S) = \{\{i, j\} \in E : i \in S, j \notin S\}$ e o subconjunto de arestas com ambas extremidades em S como $E(S) = \{\{i, j\} \in E : i \in S, j \in S\}$. Assuma agora um grafo orientado $D = (V, A)$ obtido a partir do grafo $G = (V, E)$, onde cada aresta $\{i, j\} \in E$ está associada a dois arcos (i, j) e (j, i) em A , ambos com o mesmo custo da aresta $\{i, j\}$. Para um dado digrafo $D = (V, A)$ e um subconjunto de vértices $S \subseteq V$, considere $\delta^-(S) = \{(i, j) \in A : i \notin S, j \in S\}$, $\delta^+(S) = \delta^-(V \setminus S)$ e $A(S) = \{(i, j) : i \in S, j \in S\}$ como sendo o conjunto de arcos entrando em S , saindo de S e com ambas extremidades em S , respectivamente. Por simplicidade, para um dado vértice $i \in V$, usamos a notação $\delta^+(i)$ ou $\delta^-(i)$ ao invés de $\delta^+(\{i\})$ ou $\delta^-(\{i\})$, respectivamente.

2.1 Formulações da literatura para o GMSTP

Nesta seção apresentamos as principais formulações matemáticas da literatura para o GMSTP. Em uma dada solução do GMSTP, $x \in \mathbb{B}^{|E|}$ denota um vetor de incidência referente às arestas de E na solução e $y \in \mathbb{B}^{|V|}$ denota um vetor de incidência referente aos vértices de V na solução. Nesse caso, uma aresta $e \in E$ (respectivamente um vértice $i \in V$) pertence à solução se $x_e = 1$ (respectivamente $y_i = 1$) ou, caso contrário, $x_e = 0$ (respectivamente $y_i = 0$). As variáveis x_{ij} e x_{ji} , para $e = \{i, j\} \in E$, definem uma mesma variável e o uso de x_{ij} ou x_{ji} é auto evidente, no contexto em que irão aparecer. Se $i \in V_k$ pertencer à solução, ele é dito ser o representante de V_k . Quando o problema for tratado com base no digrafo obtido do grafo original, $w \in \mathbb{B}^{|A|}$ denota um vetor de incidência referente aos arcos de A na solução. Nesse caso, um arco $(i, j) \in A$ pertence à solução se $w_{ij} = 1$ ou, caso contrário, $w_{ij} = 0$. Uma solução viável T do GMSTP pode ser vista como um subgrafo minimal em arestas satisfazendo pelo menos duas entre as três condições: (i) o número de arestas pertencentes à T é um a menos que o número de *clusters*; (ii) T contém um único caminho entre qualquer par de vértices representantes de cada *cluster* e (iii)

T não possui ciclos.

Uma formulação não direcionada proposta no trabalho pioneiro de (MYUNG *et al.*, 1995) é dada a seguir. Seja $G = (V_1 \cup \dots \cup V_m, E)$ um grafo não direcionado m -partido, sendo V_k o conjunto de vértices do *cluster* $k \in M$. Defina $x(S) = \sum_{e \in S} x_e$ e $y(S) = \sum_{i \in S} y_i$, com $S \subset V$ tal que $|S| \geq 2$. O custo de uma aresta $\{i, j\} \in E$ é denotado por $c_{ij} \geq 0$. Um modelo não direcionado baseado em cortes (\mathcal{F}_{cut}) é dado por:

$$(\mathcal{F}_{cut}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

$$\text{s.a: } x(E) = m - 1 \quad (2.1)$$

$$y(V_k) = 1, \quad \forall k \in M \quad (2.2)$$

$$x(\delta(S)) \geq y_i + y_j - 1, \quad \forall i \in S, \forall j \in V \setminus S, \forall S \subset V \quad (2.3)$$

$$x \in \mathbb{B}^{|E|}, y \in \mathbb{B}^{|V|} \quad (2.4)$$

A restrição (2.1) especifica que exatamente $m - 1$ arestas pertencem à solução. As restrições (2.2) determinam que exatamente um vértice de cada *cluster* $k \in M$ compõe a solução. As restrições de corte (2.3) garantem a existência de pelo menos uma aresta entre qualquer par de vértices $i \in S$ e $j \in V \setminus S$ que fazem parte da solução. É possível mostrar que os m *clusters* estarão conectados por seus respectivos vértices representantes.

(MYUNG *et al.*, 1995) propõem outra formulação baseada em restrições de quebra de subrotas substituindo as restrições (2.3) em (\mathcal{F}_{cut}), da seguinte forma:

$$(\mathcal{F}_{sub}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

$$\text{s.a: (2.1), (2.2), (2.4) e}$$

$$x(E(S)) \leq y(S \setminus \{i\}), \quad \forall i \in S, \forall S \subset V, \text{ com } 2 \leq |S| \leq |V| - 1 \quad (2.5)$$

(MYUNG *et al.*, 1995) provam que o conjunto de soluções da relaxação linear de (\mathcal{F}_{sub}) está estritamente contido no conjunto de soluções da relaxação linear de (\mathcal{F}_{cut}).

O problema GMSTP pode também ser considerado como o Problema da Arborescência Geradora Generalizada de Custo Mínimo (denotado por GMSAP, do inglês *Generalized Minimum Spanning Arborescence Problem*) (MYUNG *et al.*, 1995). Nesse caso, sem perda de generalidade, é construída uma árvore generalizada e direcionada sobre um digrafo $D = (V, A)$

obtido a partir de $G = (V, E)$, onde cada aresta $\{i, j\} \in E$ dá origem a dois arcos (i, j) e (j, i) em A , ambos com o mesmo custo de $\{i, j\}$. Na versão direcionada, a ideia é construir uma arborescência T enraizada em algum vértice de V_1 , o qual é o *cluster* escolhido para conter a raiz da arborescência. A partir do vértice representante de V_1 , a arborescência T é expandida sobre cada vértice representante dos demais *clusters* $q \in M \setminus \{1\}$. Uma formulação para o GMSAP é apresentada em (MYUNG *et al.*, 1995), a qual utiliza cortes para definir uma solução. Nessa formulação, uma variável x_{ij} , w_{ij} e y_i indica, respectivamente, se a aresta $\{i, j\} \in E$, um arco $(i, j) \in A$ e um vértice $i \in V$ pertence ($x_{ij} = 1, w_{ij} = 1, y_i = 1$) ou não ($x_{ij} = 0, w_{ij} = 0, y_i = 0$) à solução.

$$(\mathcal{F}_{dcut}^1) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

s.a: (2.1), (2.2) e

$$w(\delta^-(S)) \geq y_i, \quad \forall i \in S, \forall S \subseteq V \setminus V_1 \quad (2.6)$$

$$w_{ij} \leq y_i, \quad \forall i \in V_1, \forall j \in V \setminus V_1 \quad (2.7)$$

$$w_{ij} + w_{ji} = x_{ij}, \quad \forall \{i, j\} \in E \quad (2.8)$$

$$w \in \mathbb{B}^{|A|}, x \in \mathbb{B}^{|E|}, y \in \mathbb{B}^{|V|} \quad (2.9)$$

As restrições (2.6) garantem que em todo vértice i representante de um *cluster* não raiz deve existir pelo menos um arco entrando nele. As restrições (2.7) especificam que se um vértice do *cluster* raiz não for representante, então dele não deve existir nenhum arco saindo. As restrições (2.8) determinam que se o arco (i, j) ou (j, i) estiver na solução, então a aresta correspondente $\{i, j\}$ também estará.

(FEREMANS *et al.*, 2002) afirmam que (\mathcal{F}_{dcut}^1) pode ser reescrita de maneira mais compacta trocando (2.1) e (2.8) pelas seguintes desigualdades válidas, onde as variáveis x_{ij} da função objetivo são substituídas por w_{ij} :

$$w(\delta^-(V_1)) = 0, \quad (2.10)$$

$$w(\delta^-(V_k)) \leq 1, \quad \forall k \in M \setminus \{1\} \quad (2.11)$$

A restrição (2.10) especifica que nenhum arco entrará no *cluster* raiz V_1 , enquanto as restrições (2.11) determinam que no máximo um arco entrará em cada *cluster* não raiz de

$M \setminus \{1\}$. Note que esta última desigualdade é satisfeita na igualdade. A formulação resultante é denominada de (\mathcal{F}_{dcut}^2) .

Com base na formulação (\mathcal{F}_{usub}) , (FEREMANS *et al.*, 2002) apresentam uma formulação de empacotamento para o GMSAP. As variáveis para essa formulação são as mesmas usadas em (\mathcal{F}_{dcut}^1) .

$$(\mathcal{F}_{dsub}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

s.a: (2.1), (2.2), (2.5), (2.8) e

$$w(\delta^-(i)) = y_i, \quad \forall i \in V \setminus V_1 \quad (2.12)$$

$$x \in \mathbb{B}^{|E|}, w \in \mathbb{B}^{|A|}, y \in \mathbb{B}^{|V|} \quad (2.13)$$

As restrições (2.12) especificam que todo vértice que esteja na solução e que não pertença ao *cluster* raiz V_1 tem exatamente um arco de entrada.

(FEREMANS *et al.*, 2002) mostram que (\mathcal{F}_{dsub}) e (\mathcal{F}_{dcut}^1) tem o mesmo espaço de soluções na relaxação linear. Além disso, (FEREMANS *et al.*, 2002) mostram que (\mathcal{F}_{usub}) pode ser fortalecida trocando-se as restrições de eliminação de ciclos (2.5) pelas seguintes restrições

$$x(E(S)) \leq y(S \setminus V_k), \quad \forall S \subset V, 2 \leq |S| \leq n-1, \forall k \in M \quad (2.14)$$

sendo que as restrições (2.14) são dominadas por

$$x(E(S')) \leq y(S' \setminus V_k) = y(S') - 1, \quad (2.15)$$

onde $S' = S \cup V_k$, pois $x(E(S)) \leq x(E(S \cup V_k)) \leq y((S \cup V_k) \setminus V_k) = y(S \cup V_k) - 1$. Em outras palavras, qualquer restrição (2.14) para o qual S não inclui um *cluster* integralmente é dominada por (2.15), visto que S' inclui pelo menos um *cluster*. Logo, obtêm-se a seguinte formulação:

$$(\mathcal{F}_{ucsub}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

s.a: (2.1), (2.2), (2.4) e

$$x(E(S)) \leq y(S) - 1, \quad \forall S \subset V \mid \exists k \in M, \text{ com } V_k \subseteq S, \text{ e } 2 \leq |S| \leq n-1 \quad (2.16)$$

A versão direcionada da formulação (\mathcal{F}_{ucsub}) é dada a seguir.

$$(\mathcal{F}_{dsub}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

s.a: (2.1), (2.2), (2.8), (2.12), (2.13) e (2.16)

Todas as formulações abordadas até aqui são exponenciais no número de restrições. As formulações que serão consideradas a partir deste ponto apresentam um número polinomial de restrições e variáveis. Uma formulação polinomial usando fluxo simples é apresentada a seguir. Além das variáveis x indexadas pelas arestas, usaremos variáveis $f \in \mathbb{R}^{2|E|}$ em que f_{ij} é o fluxo na aresta $\{i, j\} \in E$ na direção de i para j . Na formulação apresentada em (POP, 2002), uma arborescência T é construída utilizando as variáveis f , enraizada em algum vértice de V_1 e expandida sobre exatamente um vértice de cada um dos demais *clusters* de $M \setminus \{1\}$. A ideia é um vértice do *cluster* raiz V_1 enviar $|M| - 1$ unidades de fluxo e todo *cluster* $k \in M \setminus \{1\}$ consumir exatamente uma unidade. Para cada aresta $\{i, j\} \in E$, tem-se associado a ela duas variáveis de fluxo f_{ij} e f_{ji} , um fluxo na direção de i para j e outro na direção de j para i , respectivamente. Se $f_{ij} > 0$ (resp. $f_{ji} > 0$), então um fluxo transita na direção de i para j (resp. de j para i) e nesse caso a aresta $\{i, j\}$ pertence à solução. A variável x_{ij} indica se a aresta $\{i, j\} \in E$ pertence ($x_{ij} = 1$) ou não ($x_{ij} = 0$) à solução. Segue o modelo.

$$(\mathcal{F}_{flow}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij}$$

s.a: (2.1), (2.2) e

$$\sum_{\{i,j\} \in E} f_{ij} - \sum_{\{i,j\} \in E} f_{ji} = \begin{cases} (m-1)y_i, & i \in V_1 \\ -y_i, & i \in V_k \end{cases} \quad \forall k \in M \setminus \{1\} \quad (2.17)$$

$$f_{ij} \leq (m-1)x_{ij} \quad \forall \{i, j\} \in E \quad (2.18)$$

$$f_{ji} \leq (m-1)x_{ij} \quad \forall \{i, j\} \in E \quad (2.19)$$

$$f \in \mathbb{R}^{2|E|}, x \in \mathbb{B}^{|E|}, y \in \mathbb{B}^{|V|} \quad (2.20)$$

As restrições de conservação de fluxo (2.17) estabelecem que a solução seja conexa, determinando que $m - 1$ unidades de fluxo sejam enviadas de algum vértice do *cluster* V_1 , enquanto exatamente uma unidade de fluxo será consumida por cada *cluster* $k \in M \setminus \{1\}$ por meio de seu vértice representante. Essas restrições também evitam a formação de ciclo por causa

da minimização de (\mathcal{F}_{flow}) e custos positivos das arestas de E . As restrições (2.18) e (2.19) asseguram que se a aresta $\{i, j\} \in E$ não fizer parte da solução, então nenhum fluxo passará por ela. A restrição (2.20) informa o domínio das variáveis.

Uma formulação polinomial com multifluxo é proposta em (MYUNG *et al.*, 1995). Essa formulação também assume a ideia de construir uma arborescência T enraizada em algum vértice de V_1 . Nessa formulação, o vértice representante do *cluster* raiz envia uma unidade de fluxo destinado a cada *cluster* $k \in M \setminus \{1\}$. A variável não negativa de fluxo f_{ij}^k , associada ao arco $(i, j) \in A$, transporta um fluxo originado em algum vértice de V_1 e destinado ao *cluster* $k \in M \setminus \{1\}$. Se $f_{ij}^k > 0$, então um fluxo saindo de algum vértice do *cluster* raiz V_1 transita de i para j pelo arco (i, j) com destino ao *cluster* k , sendo que nesse caso o arco (i, j) pertence à solução T . A variável w_{ij} indica se o arco $(i, j) \in A$ pertence ($w_{ij} = 1$) ou não ($w_{ij} = 0$) à solução. Segue o modelo.

$$(\mathcal{F}_{mflow}) \quad \min \sum_{(i,j) \in A} c_{ij} w_{ij}$$

s.a: (2.2), (2.11) e

$$\sum_{(i,j) \in A} f_{ij}^k - \sum_{(j,i) \in A} f_{ji}^k = \begin{cases} y_i, & \forall i \in V_1 \\ -y_i, & \forall i \in V_k \\ 0, & \forall i \notin V_1 \cup V_k \end{cases} \forall k \in M \setminus \{1\} \quad (2.21)$$

$$f_{ij}^k \leq w_{ij}, \quad \forall (i, j) \in A; \forall k \in M \setminus \{1\} \quad (2.22)$$

$$f \in \mathbb{R}_+^{|A|}, w \in \mathbb{B}^{|A|}, y \in \mathbb{B}^{|V|} \quad (2.23)$$

As restrições (2.21) determinam que uma unidade de fluxo origina-se no vértice representante do *cluster* raiz V_1 com destino a cada *cluster* $k \in M \setminus \{1\}$. Uma unidade de fluxo destinada ao *cluster* k entra no vértice representante de V_k e existe uma conservação de fluxo nos *clusters* intermediários de V_1 a V_k . Além disso, essas restrições evitam a formação de ciclo por causa da minimização de (\mathcal{F}_{mflow}) e dos custos positivos dos arcos. As restrições (2.22) estabelecem que se um arco não estiver na solução, então a variável de fluxo associada a ela será nula. A restrição (2.23) define o domínio das variáveis. Essa formulação foi implementada neste trabalho e comparada à formulação que estamos propondo.

Uma formulação de fluxo bidirecional pode ser obtida a partir de (\mathcal{F}_{mflow}) , de acordo com (POP, 2002), substituindo as restrições que possuem as variáveis w_{ij} pelas restrições:

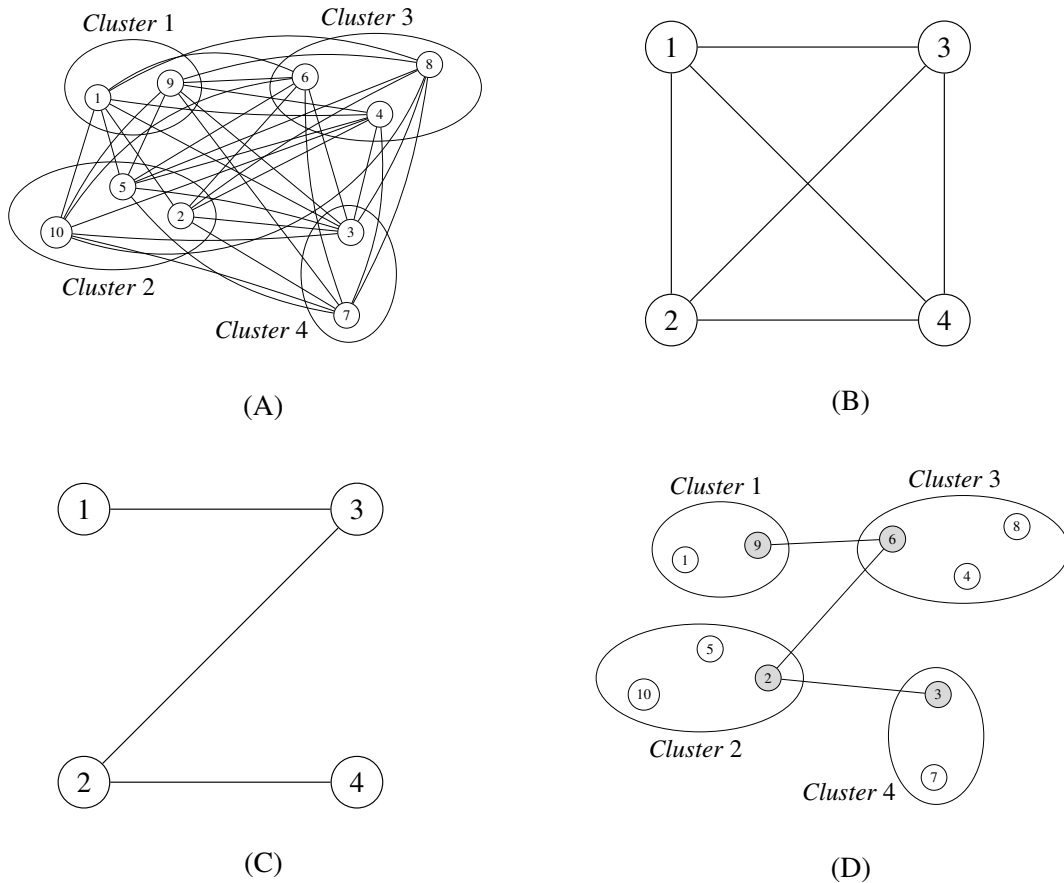
$$f_{ij}^r + f_{ji}^q \leq x_{ij}, \quad \forall \{i, j\} \in E, \quad \forall r, q \in M \setminus \{1\} \quad (2.24)$$

Denotamos essa formulação de fluxo bidirecional como (\mathcal{F}_{BDflow}) . Essa formulação não utiliza variáveis de arcos w_{ij} e unidades de fluxo são permitidas em qualquer direção, tanto de i para j como de j para i , considerando uma determinada aresta $\{i, j\} \in E$. Assuma o termo *commodity* como um fluxo destinado ao *cluster* $k \in M$. As restrições (2.24) unem diferentes *commodities* na aresta $\{i, j\} \in E$. Sempre que duas *commodities* r e q fluem na aresta $\{i, j\}$, pode-se mostrar que elas fluem na mesma direção porque somente uma das variáveis f_{ij}^r ou f_{ji}^q poderá ter valor positivo. Além disso, partindo do pressuposto que todo fluxo sai de uma única origem, não é possível que duas *commodities* se cruzem em direções diferentes associado a uma mesma aresta pelo fato de que não existe ciclo na solução, o que é garantido pelas restrições do modelo e sua função objetivo.

Outra formulação compacta é apresentada em (POP, 2002). Essa formulação utiliza um grafo global $G^* = (V^*, E^*)$ para auxiliar a construção de uma solução GST de um grafo $G = (V, E)$. No grafo global G^* , cada vértice de V^* é associado exclusivamente a um *cluster* de G . Dois vértices de V^* são conectados por uma aresta em E^* , se existir pelo menos uma aresta em G conectando os respectivos *clusters* que esses vértices representam. A ideia de (POP, 2002) é construir uma estrutura de árvore geradora Z em G^* . Restrições adicionais são definidas sobre arestas de E , relacionadas aos *clusters* conectados pela estrutura de Z , para poder definir uma solução viável GST de G . A Figura 3 (B) ilustra o grafo global G^* obtido a partir da instância do GMSTP ilustrada na Figura 3 (A). A Figura 3 (C) apresenta uma árvore geradora do grafo global presente na Figura 3 (B). A Figura 3 (D) fornece uma solução GST da instância presente na Figura 3 (A), correspondente à estrutura de árvore geradora global definida na Figura 3 (C).

As restrições para o grafo global G^* na formulação de (POP, 2002) são oriundas da formulação de (MARTIN, 1991), a qual define uma árvore geradora. Se dois vértices $r, q \in V^*$ são conectados por uma aresta $\{r, q\} \in E^*$ presente na solução, então na solução final existirá uma aresta entre os *clusters* r e q de G . A ideia é construir simultaneamente $|V^*|$ arborescências distintas, uma para cada vértice $k \in V^*$ definido como (raiz) referencial, onde cada vértice referencial induz uma orientação independente para as arestas presentes na solução. A formulação para obter uma árvore geradora Z de G^* é definida a seguir. Uma variável z_{rq} indica se a aresta $\{r, q\} \in E^*$ pertence ($z_{rq} = 1$) ou não ($z_{rq} = 0$) à árvore geradora Z . Define-se uma variável λ_{krq} para cada tripla de vértices $k, r, q \in V^*$, com $r \neq q$. Para todo vértice referencial

Figura 3 – Grafo $G = (V, E)$ em (A), grafo global $G^* = (V^*, E^*)$ de G em (B), uma árvore geradora Z de G^* em (C) e uma árvore geradora generalizada de G com a estrutura Z em (D)



Fonte: autor.

$k \in V^*$ e toda aresta $\{r, q\} \in E^*$, se $\lambda_{krq} = 1$, então a aresta $\{r, q\}$ pertence a Z e, nesse caso, ela possui uma orientação de q para r com relação ao vértice referencial k ; caso contrário $\lambda_{krq} = 0$. Seja $N(r)$ o conjunto de vértices adjacentes ao vértice $r \in V^*$. O modelo de (MARTIN, 1991) para árvore geradora é dado por:

$$(\mathcal{F}_{Martin}) \quad \sum_{\{r,q\} \in E^*} z_{rq} = m - 1 \quad (2.25)$$

$$z_{rq} - \lambda_{krq} - \lambda_{kqr} = 0, \quad \forall k \in V^*, \forall \{r, q\} \in E^* \quad (2.26)$$

$$\sum_{q \in N(r)} \lambda_{krq} = 1, \quad \forall k \in V^*, \forall r \in V^* \setminus \{k\} \quad (2.27)$$

$$\lambda_{kkq} = 0, \quad \forall k \in V^*, \forall q \in N(k) \quad (2.28)$$

$$\lambda \geq 0, z \geq 0 \quad (2.29)$$

A restrição (2.25) garante que $m - 1$ arestas de E^* compõem uma árvore geradora Z

de G^* . As restrições (2.26) especificam que $\lambda_{krq} = 1$ ou $\lambda_{kqr} = 1$ quando a aresta $\{r, q\}$ pertencer à solução Z . Para todo vértice (raiz ou referencial) $k \in V^*$, as restrições (2.27) determinam que toda aresta $\{r, q\}$ na solução Z têm uma orientação, de r para q ou vice-versa, definida por λ em relação a k . As restrições (2.28) especificam que se k for o vértice referencial de uma arborescência, então k não recebe orientação de entrada. As restrições (2.29) definem o domínio das variáveis.

Na formulação de (POP, 2002) para o GMSTP, rotulada de (\mathcal{F}_{Pop}) , a variável x_{ij} indica se a aresta $\{i, j\} \in E$ pertence ($x_{ij} = 1$) ou não ($x_{ij} = 0$) à solução GST de G . A variável y_i indica se o vértice $i \in V$ pertence ($y_i = 1$) ou não ($y_i = 0$) à solução. A formulação (\mathcal{F}_{Pop}) é descrita a seguir.

$$(\mathcal{F}_{Pop}) \quad \min \sum_{\{i,j\} \in E} c_{ij} x_{ij} \quad (2.30)$$

s.a: (2.1), (2.2), (2.25)–(2.29) e

$$\sum_{i \in V_r} \sum_{j \in V_q} x_{ij} = z_{rq}, \quad \forall \{r, q\} \in E^* \quad (2.31)$$

$$\sum_{j \in V_q} x_{ij} \leq y_i, \quad \forall q \in M, \forall i \in V \setminus V_q \quad (2.32)$$

$$x \in \mathbb{B}^{|E|}, y \in \mathbb{B}^{|V|}, z \in \mathbb{B}^{|E^*|} \quad (2.33)$$

As igualdades (2.31) especificam que se uma aresta $\{r, q\}$ pertencer à árvore geradora Z de G^* , ou seja $z_{rq} = 1$, então deve existir uma aresta do grafo original conectando os *clusters* r e q na solução GST de G . As restrições (2.32) especificam que em um dado vértice $i \in V \setminus V_q$, com $q \in M$, a soma das arestas que incidem em diferentes vértices de V_q e sobre o próprio vértice i será no máximo y_i , se i pertencer à solução. O domínio das variáveis é definido em (2.33). A corretude desse modelo e algumas propriedades sobre ele podem ser encontradas em (POP, 2002). Pode-se mostrar que para esse modelo, as variáveis x e y podem ser declaradas como contínuas. Essa formulação foi implementada neste trabalho e comparada à formulação que estamos propondo.

Por fim, destacamos as formulações poliomiais (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) . A formulação (\mathcal{F}_{mflow}) tem um bom desempenho em qualidade de soluções para instâncias pequenas e médias, enquanto que para instâncias grandes ela apresenta muito estouro de memória. A formulação de (\mathcal{F}_{Pop}) não apresenta a mesma eficiência de qualidade de solução para instâncias pequenas e médias quanto (\mathcal{F}_{mflow}) , mas (\mathcal{F}_{Pop}) não apresenta estouro de memória para instâncias grandes.

Mais comparações entre as principais formulações da literatura para o GMSTP são encontradas em (POP, 2019).

2.2 Algoritmo exato de programação dinâmica para o GMSTP

Em (POP, 2002) é definido um algoritmo exato de programação dinâmica para encontrar uma solução ótima GST de G , dada uma estrutura de árvore geradora Z do grafo global $G^* = (V^*, E^*)$, obtido a partir de G como dito anteriormente.

A partir da árvore geradora Z de G^* , define-se um grafo acíclico dirigido (denotado por DAG, do inglês *Directed Acyclic Graph*) Z^{\rightarrow} assumindo algum vértice $k \in V^*$ como sua raiz e cujos arcos são orientados a partir de k em Z^{\rightarrow} . Obedecendo a estrutura de conexão de Z^{\rightarrow} , pode-se obter uma melhor solução GST de G que segue essa estrutura. Para um dado vértice $v \in V$, considere que $T(v)$ retorna uma “sub-árvore” enraizada em v , incluindo todos os vértices descendentes de v nessa “sub-árvore”, obedecendo a estrutura e orientação das conexões de Z^{\rightarrow} . Assuma que $N^+(u)$ denota todos os vértices $v \in V$ adjacentes a u , obedecendo a orientação (u, v) , e seja $W(T(u))$ o peso mínimo da “sub-árvore” generalizada enraizada em u . O algoritmo de programação dinâmica de (POP, 2002) computa

$$\min_{u \in V_k} W(T(u)) \quad (2.34)$$

sendo que o vértice raiz $k \in V^*$ de Z^{\rightarrow} corresponde ao *cluster* $k \in M$.

Nesse algoritmo de programação dinâmica, o processo recursivo para resolver o sub-problema $W(T(u))$ é dado a seguir. Sua inicialização consiste em atribuir $W(T(u)) = 0$, para todo $u \in V_q$, tal que q seja um vértice folha em Z^{\rightarrow} .

A recursão para um vértice $u \in V_q$, com *cluster* $q \in M$ representando um vértice interno de Z^{\rightarrow} , é dada por

$$W(T(u)) = \min_{v \in N^+(u)} \{c(u, v) + W(T(v))\}, \quad (2.35)$$

onde $v \in N^+(u) \neq \emptyset$ e $c(u, v)$ denota o custo do arco (u, v) .

Para calcular o valor da solução ótima para o sub-problema $W(T(u))$, faz-se necessário testar todos os vértices adjacentes de u , seguindo a orientação de Z^{\rightarrow} . Dessa forma,

para um determinado vértice u , devemos checar no máximo $|V| - 1$ vértices para identificar a melhor conexão partindo de u . Sendo assim, a complexidade desse algoritmo de programação dinâmica é da ordem de $\mathcal{O}(|V|^2)$. O número de árvores geradoras em G^* corresponde ao valor de m^{m-2} , segundo a fórmula de (CAYLEY, 1874). Logo, uma solução ótima do GMSTP para uma instância G deve considerar todas as possibilidades de árvore geradora em G^* , o que leva a uma complexidade computacional dessa estratégia da ordem de $\mathcal{O}(m^{m-2}|V|^2)$.

Esse algoritmo de programação dinâmica é utilizado na metaheurística VNS definida em (HU *et al.*, 2005) e no procedimento heurístico de duas fases definido em (POP *et al.*, 2017). No trabalho de (HU *et al.*, 2005) ele é utilizado num processo de busca local e no trabalho de (POP *et al.*, 2017) ele é utilizado para definir uma solução GST dadas as estruturas de árvore geradora criadas para o grafo global G^* .

2.3 Metaheurísticas da literatura para o GMSTP

No uso de metaheurísticas, frequentemente utiliza-se um algoritmo de busca local para melhorar a exploração do espaço de solução de uma instância. No GMSTP, tem-se uma estratégia de busca local utilizada em diversos trabalhos da literatura, a qual é definida a seguir. Essa estratégia utiliza a representação de uma solução GST num vetor de m posições. A i -ésima posição desse vetor é associada ao i -ésimo *cluster* de G e guarda a informação de qual vértice do i -ésimo *cluster* pertence à solução. Dada uma solução T , considere a estrutura de exploração de vizinhança $N^c(T)$, com $c \geq 1$, conforme segue. Para $c = 1$, tem-se a busca local $N^1(T)$. Nela, um *cluster* $k \in M$ é selecionado aleatoriamente e troca-se o vértice representante $i \in V_k \cap V(T)$ com outro vértice $j \in V_k \setminus V(T)$. Para o *cluster* V_k , essa estratégia considera todas as $|V_k| - 1$ soluções vizinhas de T . O vértice representante de V_k que fornece a solução de menor custo é mantido no vetor solução e esse processo é repetido até que todos os *clusters* sejam explorados. Para $c = 2$, tem-se a busca local $N^2(T)$. Nessa busca local, dois *clusters* $r, q \in M$ são selecionados aleatoriamente e trocam-se os vértices representantes $i \in V_r \cap V(T)$ e $j \in V_q \cap V(T)$ com outros vértices $s \in V_r \setminus V(T)$ e $y \in V_q \setminus V(T)$. Para os *clusters* V_r e V_q , essa estratégia considera todas as $(|V_r|-1) \times (|V_q|-1)$ soluções vizinhas de T . A combinação de vértices em V_r e V_q que fornece a solução de menor custo é mantida no vetor solução. Esse processo é repetido até que todos os *clusters* sejam visitados. A mesma lógica serve para a busca local $N^3(T)$.

Tabu Search (GLOVER, 1989) é uma metaheurística baseada em busca local e já foi utilizada para resolver o GMSTP. Partindo de uma solução inicial, a *Tabu Search* move-se em

cada iteração do algoritmo, da solução atual para uma solução vizinha através de um algoritmo de busca local, mesmo que essa tenha um custo de solução maior. Não é aceito movimentos que levem a soluções recém visitadas, pois nessa metaheurística existe uma memória chamada de lista tabu, a qual guarda soluções recém exploradas durante um determinado número de iterações. Soluções dessa lista são descartadas pela metaheurística na intenção de evitar ciclos de soluções e o algoritmo é encerrado quando um critério de parada é satisfeito.

Em (FEREMANS *et al.*, 2002), a metaheurística *Tabu Search* foi utilizada para gerar uma boa solução de partida para um algoritmo *Branch-and-Cut*. Nesse trabalho, a *Tabu Search* utiliza a busca local $N^1(T)$. Em (GHOSH, 2003) são apresentadas duas versões dessa metaheurística para o GMSTP. Na primeira versão, essa metaheurística explora sua vizinhança utilizando a estratégia de busca local $N^1(T)$. Na segunda versão é utilizada a estratégia $N^1(T)$ junto com a estratégia de desestimular a seleção de vértices que tenham sido utilizados muitas vezes, onde é utilizado um vetor para contabilizar o número de vezes de seleção de cada vértice. Em (WANG *et al.*, 2006), também foi desenvolvida uma *Tabu Search* para o GMSTP e utilizou-se a estratégia de busca local $N^1(T)$ para explorar a vizinhança de uma solução. Além disso, esse trabalho utilizou a estratégia de criar um conjunto elite das melhores soluções para intensificar a análise de vizinhança nessas soluções. Em (ÖNCAN *et al.*, 2008), a metaheurística *Tabu Search* explora a vizinhança de uma solução utilizando as buscas locais $N^1(T)$, $N^2(T)$ e $N^3(T)$.

Outras metaheurísticas já desenvolvidas para o GMSTP são a *Variable Neighborhood Descent* (VND) (MLADENOVIC; HANSEN, 1997) e a *Variable Neighborhood Search* (VNS) (MLADENOVIC; HANSEN, 1997), as quais exploram o espaço de soluções de uma instância através de algoritmos de busca local. A VND usa sucessivas estruturas de vizinhança. Primeiramente ela define um conjunto de estruturas de vizinhança R_l , com o $l = \{1, \dots, l_m\}$. Seja R_1 a primeira estrutura de vizinhança a ser usada na solução T . Se um melhoramento da solução T não é possível com a estrutura de vizinhança R_1 , então a estrutura de vizinhança é alterada para R_2 . Se um melhoramento na solução atual T é encontrada, então a estrutura de vizinhança a ser utilizada retorna para a primeira de R_l , a R_1 . O algoritmo termina após serem utilizadas todas as estruturas de vizinhança sem conseguir melhorar a solução corrente. Já a metaheurística VNS, a partir de uma solução inicial T , escolhe uma vizinhança R_k para explorar a solução atual. No início de cada iteração, a solução atual é perturbada na intenção de fugir de ótimos locais, sendo que nessa perturbação é aceitado uma degradação da solução. Utilizando a estrutura de vizinhança R_k , caso seja encontrada alguma solução melhor que a atual, o algoritmo vai para a

estrutura de vizinhança R_1 , caso contrário vai para a estrutura R_{k+1} . O algoritmo para quando um critério de parada é satisfeito.

Em (GHOSH, 2003) são utilizadas duas versões do VNS e duas versões do VND para o GMSTP, sempre utilizando como estrutura principal as buscas locais $N^1(T)$ e $N^2(T)$. (HU *et al.*, 2005) criaram uma metaheurística para o GMSTP que combina o VNS e o VND. Nesse trabalho, além das buscas locais $N^1(T)$ e $N^2(T)$, uma busca local baseada na estrutura do grafo global G^* de G também é utilizada. Dada uma solução T , essa busca local identifica qual a estrutura de árvore global de T e calcula a solução ótima GST para essa estrutura, utilizando o algoritmo de programação dinâmica definido em (POP, 2002). Em seguida, uma conexão da estrutura de árvore global é substituída por outra e novamente o algoritmo de programação dinâmica é utilizado para calcular a solução ótima GST dessa nova estrutura. Esse processo é repetido até que todas as conexões da estrutura de árvore global inicial sejam substituídas.

(FERREIRA *et al.*, 2012) propõem uma metaheurística GRASP (FEO; RESENDE, 1989) para o GMSTP. Essa metaheurística constrói uma nova solução em cada iteração e a melhora por meio de algoritmos de busca local. O trabalho de (FERREIRA *et al.*, 2012) utiliza cinco procedimentos diferentes para criar uma solução GMSTP, os quais são utilizados de forma intercalada no decorrer do algoritmo. O primeiro procedimento consiste em selecionar aleatoriamente um vértice para cada *cluster* e por meio do algoritmo de (KRUSKAL, 1956) construir uma solução. O segundo procedimento é uma adaptação do algoritmo de (KRUSKAL, 1956). Nessa adaptação, cada nova aresta inserida na solução é oriunda da seleção de uma lista restrita de candidatas. Essa lista é constituída de arestas que têm o custo menor ou igual a um determinado valor definido dinamicamente.

Ainda sobre a metaheurística GRASP de (FERREIRA *et al.*, 2012), o terceiro procedimento empregado para criar uma nova solução é uma adaptação do algoritmo de (PRIM, 1957). Inicialmente, a solução parcial T é constituída de um vértice de um *cluster* V_r . No passo seguinte, um *cluster* V_q se une aos *clusters* de T com uma aresta selecionada de uma lista restrita de candidatas. Essa lista também é constituída por arestas que têm um custo menor ou igual a um determinado valor definido dinamicamente. O quarto procedimento heurístico escolhe um vértice $i \in V_r$ para compor a solução, usando como base a distância média de cada vértice em V_r para um conjunto auxiliar $S_i \subseteq V \setminus V_r$, sendo que essa distância média do vértice i para o conjunto S_i é denotada por d_i . O vértice $i \in V_r$ a compor a solução é sorteado de uma lista composta por vértices que têm o valor de d_i menor ou igual a determinado valor definido dinamicamente.

O quinto procedimento consiste em escolher c *clusters* para compor a solução parcial T , cada *cluster* tendo seu vértice próprio. Em seguida, iterativamente, cada *cluster* fora de T é inserido na solução através da menor conexão que o une a um *cluster* de T .

O trabalho de (FERREIRA *et al.*, 2012) utiliza como busca local a estratégia $N^1(T)$. Além disso, esse trabalho utiliza a estratégia *path-relinking*. A partir da solução corrente T e uma determinada solução $T^\Delta \in T_E$, onde T_E é o conjunto elite de boas soluções encontradas até a iteração atual, o algoritmo percorre soluções intermediárias de T até T^Δ . Se uma solução intermediária tiver um custo menor do que a melhor solução corrente, essa é guardada. A estratégia *iterated local search* também foi utilizada. Nesse processo, uma perturbação é aplicada na solução corrente e posteriormente a busca local definida no início desse parágrafo é empregada. Essa metaheurística GRASP, usando um conjunto de instâncias padrão da literatura para o GMSTP, são comparados com o BRKGA desenvolvido neste trabalho na sub-seção 4.5.

O trabalho de (FERREIRA *et al.*, 2012) ainda define um procedimento para determinar limites inferiores de instâncias usando um algoritmo de geração de cortes para o Problema da Árvore de Steiner Direcionada (PASD). Inicialmente, uma dada instância GMSTP $G = (V, E)$ é considerada como uma instância PASD $G^\triangleright = (V^\triangleright, E^\triangleright)$. Nesse processo, um vértice terminal é criado em cada *cluster* de G^\triangleright . Seja S o conjunto de vértices terminais criados. O *cluster* de maior tamanho possuirá o vértice terminal r , o qual será raiz da arborescência a ser criada, sendo que o mesmo se conecta a todos os vértices do seu *cluster* através de arcos partindo de r . Nos demais *clusters*, os vértices não terminais se conectam ao vértice terminal do seu *cluster*, com arcos partindo dos vértices não terminais. Todas as arestas de G são substituídas por arcos em G^\triangleright , ambos com o mesmo custo da aresta que os originou. O objetivo é encontrar uma arborescência enraizada em r que cubra todos os vértices em S , de forma a minimizar a soma dos custos dos arcos selecionados. O algoritmo é baseado numa formulação de corte direcionado para o PASD. No início do algoritmo, resolve-se um problema linear relaxado P referente a essa formulação de corte. Em seguida, a partir da solução X_0 gerada por essa relaxação linear é resolvido um problema de separação, procurando-se restrições não presentes em P e violadas por X_0 . Tais restrições são incluídas em P que novamente é resolvido, gerando uma nova solução X_1 . O processo é repetido até que não haja mais restrições de corte violadas em X .

Em (GOLDEN *et al.*, 2005) foi apresentada a metaheurística Algoritmo Genético (AG) para o GMSTP. Essa metaheurística, proposta por (HOLLAND, 1962), consiste em criar novas populações até que uma condição de parada seja satisfeita. Sua população inicial é criada

aleatoriamente. Em cada nova geração, 10% de seus indivíduos são herdados do conjunto elite de indivíduos da geração predecessora. Os indivíduos de uma geração são classificados por seu valor de solução e por uma função de probabilidade. Os demais indivíduos de uma geração são criados a partir de quatro operadores: cruzamento, auto reprodução, mutação e busca local. O cruzamento é feito selecionando aleatoriamente duas soluções pais e escolhendo um mesmo ponto de corte sobre elas. São gerados dois filhos que recebem alternadamente partes das soluções pais, sendo que essas partes obedecem ao ponto de corte. A auto reprodução consiste em selecionar uma solução e dividi-la em duas partes. Cada parte se reproduz até se tornar uma solução viável. A mutação consiste em selecionar um *cluster* da solução atual e substituir seu vértice representante por outro vértice desse mesmo *cluster*. Além disso, a busca local $N^1(T)$ é aplicada nessa metaheurística como um segundo tipo de mutação.

Outro AG para o GMSTP é proposto em (CONTRERAS-BOLTON *et al.*, 2016). Ele é executado até que seja alcançado um número fixo de gerações ou até que o valor ótimo conhecido de uma instância seja alcançado. Cada nova geração recebe 10% dos indivíduos elite da sua geração predecessora. A classificação de indivíduos elite e não elite é avaliada segundo dois critérios. Dada uma solução T , o primeiro critério considera o erro relativo da solução T em relação ao valor de uma solução ótima ou do melhor valor conhecido para a instância tratada. O segundo critério considera uma penalização para indivíduos que tendem a gerar soluções inviáveis. Novos indivíduos são gerados por meio de um operador de cruzamento ou por um dos cinco operadores de mutação dessa metaheurística. No cruzamento, dois indivíduos pais são selecionados para gerar dois indivíduos filhos com base em dois pontos de corte. O primeiro operador de mutação escolhe aleatoriamente duas posições do vetor e troca seus valores. O segundo operador de mutação escolhe uma posição do vetor e altera o vértice representante do *cluster* associado a essa posição por outro vértice que gere uma solução viável.

O terceiro, quarto e quinto operadores de mutação são similares, com a diferença de que não exaustivas são as mutações. O terceiro operador corresponde à estratégia de busca local $N^1(T)$. O quarto operador é inspirado em $N^1(T)$, com a diferença de que a busca local é encerrada em um *cluster* assim que houver a primeira melhoria de custo da solução no indivíduo analisado. Esse quarto operador visita todos os *clusters* do indivíduo abordado. O quinto operador de mutação escolhe aleatoriamente um *cluster* e busca um vértice que melhore o custo de solução do indivíduo trabalhado. Se não houver melhoria alguma para o *cluster* escolhido, outro *cluster* é selecionado. Assim que houver a primeira melhoria, esse operador é encerrado. O AG de

(CONTRERAS-BOLTON *et al.*, 2016) também é comparado com o BRKGA desenvolvido neste trabalho, na sub-seção 4.5.

Por fim, um procedimento heurístico de duas fases para o GMSTP é definido em (POP *et al.*, 2017). Os autores obtêm uma solução GMSTP decompondo o problema original em dois sub-problemas: problema global e problema local. O objetivo do sub-problema global é determinar estruturas de árvore geradora Z sobre o grafo global $G^* = (V^*, E^*)$ de G utilizando um AG, enquanto que o objetivo do sub-problema local é determinar a solução ótima GST para uma dada estrutura de árvore global Z . O sub-problema local obtém uma solução ótima GST usando o algoritmo de programação dinâmica de (POP, 2002). No AG desse procedimento, uma solução de árvore global é representada através da codificação de Prüfer (MATOUSEK; NESETRIL, 2008). Além disso, cada indivíduo I é composto por um par de dois indivíduos $\{C_1, C_2\}$. O custo de um indivíduo I corresponde à soma de seus indivíduos C_1 e C_2 . A população inicial é criada aleatoriamente. Em uma dada iteração, para gerar uma nova população, selecionam-se φ indivíduos da geração predecessora e criam-se λ filhos. A nova população é composta por esses φ indivíduos da geração predecessora mais os novos λ indivíduos filhos gerados. É possível que um indivíduo sobreviva durante toda a execução do algoritmo. No cruzamento é utilizada a estratégia de um ponto de corte. Dois indivíduos pais geram dois indivíduos filhos, ambos contendo internamente dois indivíduos. A mutação corresponde a trocar o vértice representante de um *cluster* por um outro que gere uma solução viável, sendo aplicada sobre os cromossomos $\{C_1, C_2\} = I$, com 5% de probabilidade de acontecer para um *cluster* dos cromossomos de um novo indivíduo gerado. Esse procedimento heurístico também é comparado com o BRKGA desenvolvido neste trabalho, na sub-seção 4.5.

3 COMPLEXIDADE E PRÉ-PROCESSAMENTO DE INSTÂNCIAS DO GMSTP

Neste capítulo é abordada a complexidade do GMSTP, casos onde uma instância GMSTP pode ser resolvida em tempo polinomial, procedimentos de pré-processamento para fixação de vértices e arestas em instâncias GMSTP, onde vértices e arestas que seguramente não compõem uma solução ótima são fixados com o valor zero nos respectivos modelos matemáticos, e uma redução do GMSTP ao problema de conjunto independente de menor valor.

3.1 Complexidade do problema

Para provar que a versão de decisão do GMSTP pertence à classe de problemas NP-Completo, já assumindo que a mesma pertence à classe NP, é suficiente mostrar que existe um problema NP-Completo que pode ser reduzido polinomialmente para ele. Para tal demonstração, pode-se utilizar o problema de cobertura de vértices, o qual é NP-Completo (GAREY; JOHNSON, 1979). Para um determinado grafo $G = (V, E)$ e um valor $\omega < |V|$, o problema de cobertura de vértices consiste em determinar se existe um conjunto de vértices ζ com tamanho até ω , tal que toda aresta de E incida em pelo menos um vértice do conjunto ζ .

Teorema 3.1.1 *O GMSTP em sua versão de decisão é NP-Completo (MYUNG et al., 1995).*

Prova: Considere o problema de cobertura de vértices em um grafo $G = (V, E)$ e um valor inteiro positivo ω . Pode-se construir um grafo $G' = (V', E')$, a partir de G , onde G' contém uma cobertura de vértices ζ de tamanho até ω se, e somente se, existir uma árvore geradora generalizada em G' de custo ω .

O grafo G' contém $m = (|E| + \omega + 1)$ *clusters*. O conjunto de vértices $V' = \{V'_1, V'_2, \dots, V'_m\}$ é determinado da seguinte maneira:

- o conjunto V'_1 contém um único vértice, denotado por r ;
- os conjuntos $V'_2, \dots, V'_{\omega+1}$ são idênticos e cada um é composto por uma cópia de todos os vértices de V ;
- os $|E|$ conjuntos restantes, $V'_{\omega+2}, \dots, V'_m$, são *clusters* de 1 vértice, cada um contendo um vértice associado exclusivamente a uma aresta de E .

O conjunto de arestas E' é construído conforme descrito abaixo.

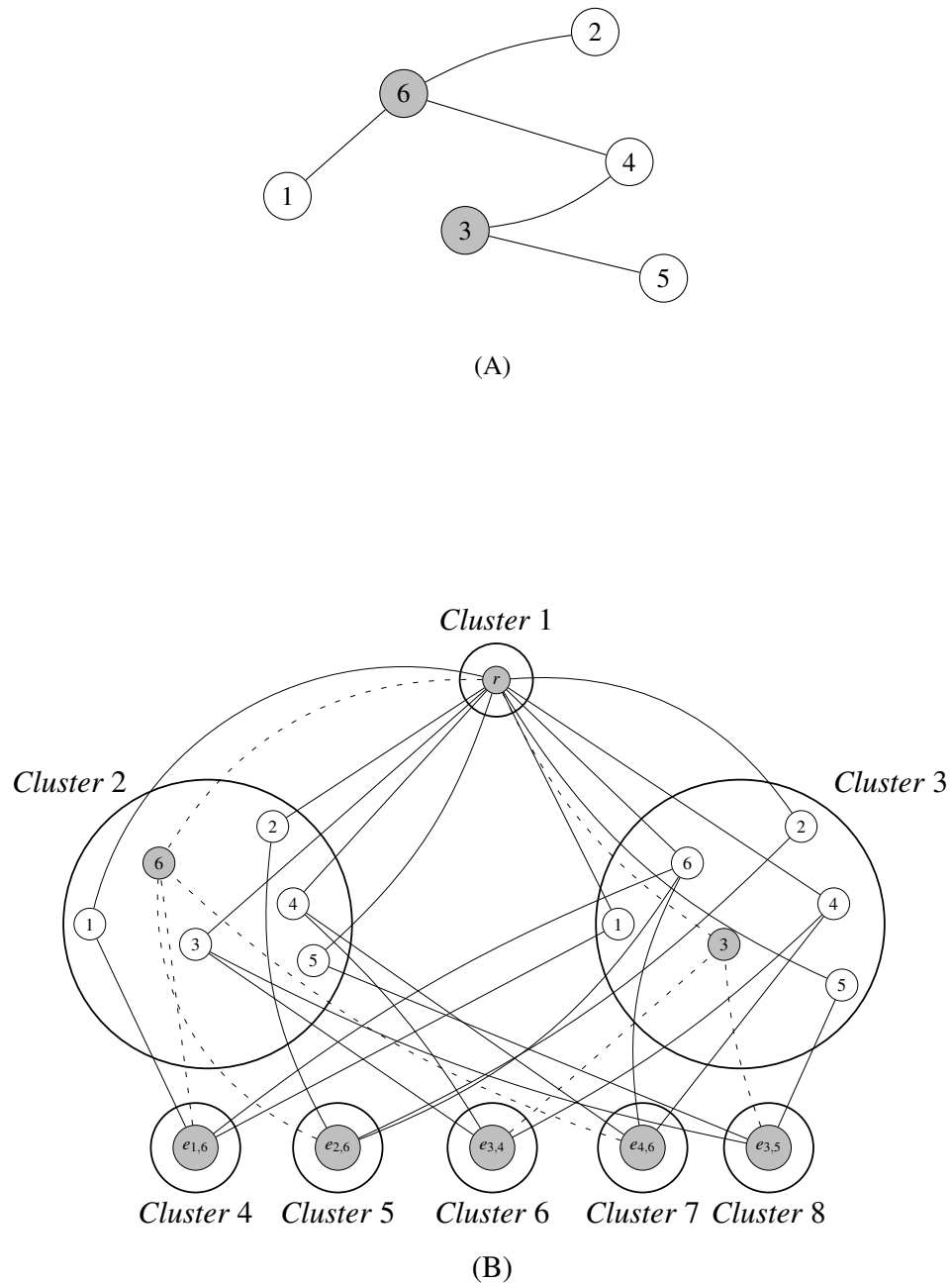
- cada vértice de $i \in V'_k$, para todo $k \in \{2, \dots, \omega + 1\}$, é conectado por uma aresta com o vértice $r \in V'_1$. Todas essas arestas tem custo 1 (um).

- considere um vértice $i \in V'_k$, para todo $k \in \{2, \dots, \omega + 1\}$, e um vértice $j \in V'_d$, para todo $d \in \{\omega + 2, \dots, m\}$. Existe uma aresta $\{i, j\} \in E'$, se a aresta de E associada ao vértice $j \in V'_d$ for incidente no vértice i em E . Todas essas arestas tem custo 0 (zero).

Suponha que exista uma árvore geradora generalizada em G' , contendo o vértice r , um vértice de cada *cluster* $\{2, \dots, \omega + 1\}$ e todo vértice de cada *cluster* $\{\omega + 2, \dots, m\}$. Se existir tal árvore geradora generalizada de custo ω em G' , então os vértices selecionados nos *clusters* $\{2, \dots, \omega + 1\}$ em G' correspondem a uma cobertura de vértices em G de tamanho ω . Por outro lado, se existir uma cobertura de vértices ζ de tamanho máximo ω em G , selecionando um vértice de ζ em cada *cluster* $\{2, \dots, \omega + 1\}$ em G' , pode-se montar uma árvore geradora generalizada em G' de custo ω . ■

Para exemplificar a redução do problema de cobertura de vértices para o GMSTP, considere o grafo $G = (V, E)$ da Figura 4 (A). Note que G tem uma cobertura de vértice de tamanho $\omega = 2$. A partir de G podemos construir o grafo $G' = (V', E')$ da Figura 4 (B), o qual possui um número de *clusters* $m = (1 + \omega + |E|) = 8$. Em V' , o *cluster* 1 contém o vértice r . Os *clusters* 2 e 3, os quais correspondem aos *clusters* $\{2, \dots, \omega + 1\}$ na construção, contém cada um cópias de todos os vértices de V . Os *clusters* 4, 5, 6, 7 e 8, os quais correspondem aos *clusters* $\{\omega + 2, \dots, m\}$, contém cada um exclusivamente uma aresta de E como vértice. Observe que é possível montar uma árvore geradora generalizada T em G' de custo $\omega = 2$. Nessa árvore, os vértices a serem selecionados nos *clusters* $\{2, \dots, \omega + 1\}$ devem conter uma ligação com todos os vértices dos *clusters* 4, 5, 6, 7 e 8. Suponha que os vértices selecionados nos *clusters* $\{2, \dots, \omega + 1\}$ sejam os vértices $\{3, 6\}$, um em cada *cluster*. Esses vértices selecionados correspondem a uma cobertura de vértices no grafo G da Figura 4 (A). Por outro lado, como existe uma cobertura de vértices ζ de tamanho até $\omega = 2$ em G , que no caso da Figura 4 (A) é $\zeta = \{3, 6\}$, pode-se montar uma árvore geradora generalizada de custo $\omega = 2$ em G' selecionando um vértice de ζ em cada *cluster* $\{2, \dots, \omega + 1\}$ de V' .

Figura 4 – Redução do problema de cobertura de vértices para o GMSTP. Em (A) tem-se uma instância do problema de cobertura de vértices de tamanho até $\omega = 2$ e, em (B), sua transformação (polinomial) em uma instância GST, com $\omega = 2$. As arestas tracejadas indicam uma GST



Fonte: autor.

3.2 Redução polinomial em relação ao problema de conjunto independente de menor valor

O GMSTP possui relação com o Problema da Árvore Geradora Mínima Sob Restrições de Conflitos (denotado por MSTCCP, do inglês *Minimum Spanning Tree under Conflict Constraints Problem*). Duas arestas são ditas conflitantes se elas não podem aparecer simultaneamente na solução.

Dado um grafo $G = (V, E)$ e um conjunto H formado de c subconjuntos de arestas conflitantes de E , onde $H = \{H_1, H_2, \dots, H_c\}$ e $H \subseteq E$, o problema MSTCCP consiste em encontrar uma árvore geradora mínima livre de conflitos, i.e., uma árvore geradora de T com custo mínimo e que inclua no máximo uma das arestas conflitantes de cada subconjunto $H_i \in H$, com $i \in \{1, 2, \dots, c\}$.

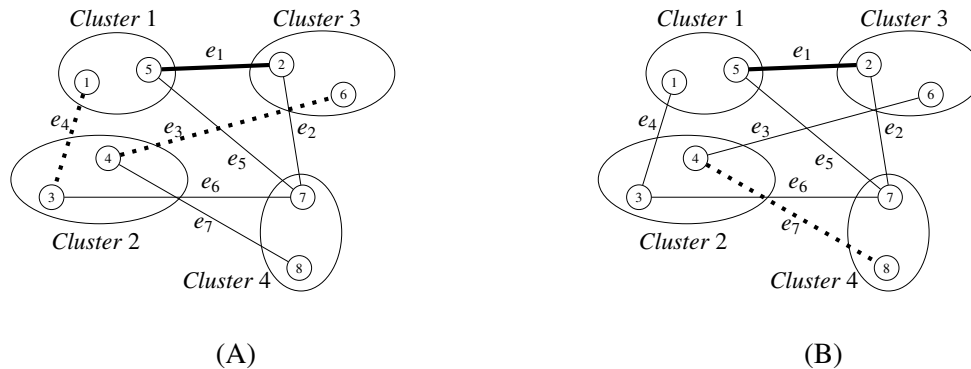
As arestas conflitantes podem ser divididas em dois grupos: arestas de conflito direto e indireto. Identificar as arestas de conflito direto é trivial. Essas arestas são aquelas incidentes em um mesmo *cluster*, porém em vértices distintos do mesmo. As arestas de conflito indireto são aquelas que incidem sobre *clusters* distintos e não podem aparecer juntas numa solução.

Para identificar conflito entre duas arestas quaisquer que não compartilham algum *cluster* em suas extremidades, pode-se utilizar um algoritmo de busca em profundidade. Dadas duas arestas $\{i, j\}, \{k, l\} \in E$, com $i \in V_r, j \in V_q, k \in V_u, l \in V_p$, com os *clusters* r, q, u, p todos distintos, elimine todos os vértices de $V_r \setminus \{i\}, V_q \setminus \{j\}, V_u \setminus \{k\}$ e de $V_p \setminus \{l\}$. Em seguida, inicie uma busca em profundidade a partir de uma das extremidades de $\{i, j\}$ a uma das extremidades de $\{k, l\}$. Se não houver um percurso que possibilite alcançar a segunda aresta partindo da primeira, é porque elas formam um conflito indireto.

As arestas conflitantes e não conflitantes de uma instância GMSTP podem ser analisadas por meio de um grafo de conflitos. Seja $G^\triangleright = (V^\triangleright, E^\triangleright)$ o grafo de conflitos de $G = (V, E)$. Cada vértice em V^\triangleright é associado exclusivamente a uma aresta de E . Dois vértices de V^\triangleright são conectados por uma aresta em E^\triangleright se esses vértices correspondem a duas arestas conflitantes em E . Como exemplo, podemos observar os grafos da Figura 5, onde a Figura 5 (B) apresenta o grafo de conflitos da instância GMSTP da Figura 5 (A).

Na configuração da Figura 5 (A), considerando a aresta e_1 , suas arestas de conflito direto são as arestas e_3 e e_4 . Ainda na Figura 5 (A), considerando a aresta e_1 , sua aresta de conflito indireto é a aresta e_7 , pois não existe um caminho de uma das extremidades de e_1 a uma extremidade e_7 que utilize apenas seus vértices como representantes de seus respectivos *clusters*.

Figura 5 – Em (A) tem-se ilustração de arestas de conflito direto e em (B) tem-se a ilustração de arestas de conflito indireto



Fonte: autor.

Através desse grafo de conflitos, podemos chegar a duas proposições diretas de se ver.

Proposição 3.2.1 *As arestas de toda GST em G fornecem um conjunto independente de tamanho $m - 1$ em G^\triangleright .*

Proposição 3.2.2 *Todo conjunto independente de tamanho $m - 1$ em G^\triangleright leva a uma GST em G , se as arestas correspondentes aos vértices desse conjunto independente não induzirem um ciclo em G .*

Na Figura 5 (B), o conjunto independente formado por vértices pontilhados corresponde à uma solução GST composta pelas arestas $\{e_1, e_2, e_6\}$ em G na Figura 5 (A).

Por meio do grafo de conflitos é possível explorar uma nova série de propriedades do GMSTP. Uma restrição de conjunto independente em G^\triangleright pode ser convertida para uma restrição de arestas conflitantes em G , se as arestas associadas aos vértices desse conjunto independente não formam um ciclo em G . Olhando para o problema de conjunto independente, sabe-se que ele tem uma relação com o problema de clique. Se um grafo $G^\delta = (V^\delta, E^\delta)$ possuir uma clique de tamanho h , o grafo complementar de G^δ possuirá um conjunto independente de tamanho h . O problema do conjunto independente também tem relação com o problema de cobertura de vértices. Se no grafo $G^\delta = (V^\delta, E^\delta)$ existir um conjunto independente de tamanho b , então em G^δ existirá uma cobertura de vértices de tamanho $|V^\delta| - b$. Vale lembrar que o trabalho de (MYUNG *et al.*, 1995) demonstra que o GMSTP é NP-Difícil através de uma redução do problema de cobertura de vértices de tamanho ω para o GMSTP.

As restrições abrangendo arestas de conflito direto são restrições redundantes em uma formulação para o GMSTP porque elas fazem parte da definição do problema, onde é garantido

que somente um vértice de cada cluster pertence à solução e não há arestas incidindo sobre vértices não selecionados. As restrições envolvendo arestas de conflitos indireto restringem o espaço de soluções em formulações para o GMSTP, uma vez que esse conflito não é estabelecido na definição do problema. Essas restrições não foram exploradas neste trabalho devido essa estratégia ter sido identificada muito recentemente.

3.3 Casos polinomiais do GMSTP

Nesta seção exibimos casos em que o GMSTP pode ser resolvido em tempo polinomial.

Caso 1: Dado uma instância GMSTP $G = (V, E)$ e seu conjunto de *clusters* M , se $|V_k| = 1$, para todo $k \in M$, o GMSTP pode ser reduzido ao clássico problema MST, o qual é resolvido em tempo polinomial por algoritmos como (KRUSKAL, 1956) e (PRIM, 1957).

Caso 2: Dada uma instância GMSTP $G = (V, E)$, se o grafo global $G^* = (V^*, E^*)$ de G tiver uma estrutura de árvore geradora, tem-se o caso onde uma melhor solução GST de G pode ser obtida em tempo polinomial, seguindo essa estrutura, usando o algoritmo de programação dinâmica de (POP, 2002), que nesse caso seria uma solução ótima de G .

Caso 3: Dado uma instância GMSTP $G = (V, E)$, onde as arestas do conjunto E formam componentes com no máximo $|M|$ vértices em cada uma, pode-se obter uma árvore geradora generalizada T usando o algoritmo de (PRIM, 1957), por exemplo. Por fim, a árvore generalizada geradora T de menor valor corresponde à solução ótima da instância $G = (V, E)$.

3.4 Propriedades para fixação de vértices e arestas em instâncias do GMSTP

Uma instância do GMSTP pode ser reduzida (em termos de eliminação ou fixação de vértices e arestas) para ter o seu tempo de cálculo de sua solução reduzido. Nesta seção, propomos regras para a fixação de vértices e explicamos as propriedades de fixação de arestas propostas por (FERREIRA *et al.*, 2012).

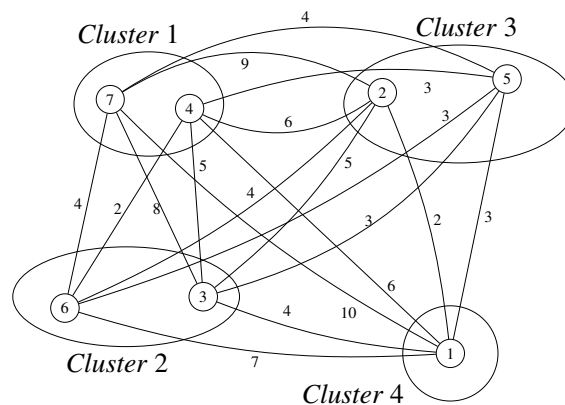
3.4.1 Fixando vértices em uma instância do GMSTP

Tão importante quanto eliminar arestas é eliminar vértices, pois para cada vértice eliminado existe a eliminação de todas as arestas incidentes nele. As condições impostas para a fixação de vértices aqui apresentadas são muito fortes e por isso às vezes é difícil pensar

em custos de arestas com significado teórico ou prático onde essa fixação possa vir a obter grande êxito. Considere uma instância $G = (V, E)$, do GMSTP, com G sendo um grafo m -partido completo. Se G não for completo, considera-se um custo bem grande nas arestas usadas para completar o grafo. Para um determinado *cluster* $r \in M$, com $|V_r| \geq 2$, considere um vértice $y \in V_r$. Se $c_{yv} > c_{zv}$, para todo $z \in V_r \setminus \{y\}$ e para todo $v \in V_q$, com $q \in M \setminus \{r\}$, ou seja, se y sempre tem a aresta de maior custo para conectar o *cluster* V_r com qualquer vértice v de V_q , então o vértice y pode ser descartado do grafo G porque inseri-lo em uma solução viável de G implicará em aumentar o custo dessa solução.

Considerando o *cluster* 1 da Figura 6, o vértice 7 sempre tem a conexão de maior custo entre os vértices do *cluster* 1 com qualquer vértice do *cluster* 2. Pode-se observar que o mesmo acontece quando é analisado o *cluster* 1 com os *clusters* 3 e 4. Nesse caso, o vértice 7 do *cluster* 1 pode ser eliminado. Considerando o vértice 2 do *Cluster* 3, pode-se observar que ele tem a maior conexão entre os vértices do *Cluster* 3 para qualquer vértice dos *Clusters* 1 e 2. No entanto, o vértice 5 do *Cluster* 3 é o mais distante desse *cluster* para qualquer vértice do *Cluster* 4. Nesse caso, nenhum vértice do *Cluster* 3 é sempre o mais distante de qualquer vértice dos demais *clusters* e, portanto, nenhum vértice do *cluster* 3 pode ser eliminado.

Figura 6 – Fixando vértices em uma instância do GMSTP



Fonte: autor.

A seguir, considere que a função $elm(V_r)$ retorna o vértice do *cluster* $r \in M$ que sempre é o mais distante entre os vértice de V_r de qualquer vértice dos demais *clusters*. Caso não exista esse vértice de V_r que sempre seja o mais distante, a função $elm(V_r)$ retorna um valor nulo.

Teorema 3.4.1 *Dado uma instância GMSTP associada a $G = (V, E)$, considere como solução ótima de G a árvore $T^> = (V^>, E^>)$. Se $elm(V_r)$ retorna um valor não nulo $y \in V_r$, com $|V_r| \geq 2$, então $y \notin T^>$.*

Prova: Seja $T^> = (V^>, E^>)$ uma GMST de G . Para um dado *cluster* V_r , com $r \in M$ e $|V_r| \geq 2$, suponha que a função $elm(V_r)$ retorna um vértice $y \in V^>$. Como $|V_r| \geq 2$, então existe pelo menos um vértice $z \in V_r \setminus \{y\}$. Trocando o vértice y de $V^>$ pelo vértice z , sendo que $c_{yv} > c_{zv}$ para todo $z \in V_r \setminus \{y\}$ e para todo $v \in V_q$, com $q \in M \setminus \{r\}$, tem-se uma nova solução de custo menor que $T^>$, o que é uma contradição. Portanto, $y \notin T^>$. ■

O processo de fixação de vértices também pode ser aplicado usando a expressão $c_{yv} \geq c_{zv}$ ao procurar o vértice y de V_r que é o mais distante para qualquer vértice de outro *cluster*. Nesse caso, a eliminação de y não afetaria o valor de uma solução ótima.

O processo de fixação de vértices pode ser aplicado em instâncias onde haja a fixação de arestas definida pelo processo polinomial descrito a seguir, na sub-seção 3.4.2. Isso é possível porque esse processo fixa arestas que garantidamente não compõem uma solução ótima. Logo, essas arestas podem ser consideradas como tendo um valor bem grande no processo de fixação de vértices. Por fim, observe que essa fixação de vértices tem efeito em cadeia, podendo ser reaplicada até que não seja mais possível fixar vértice algum.

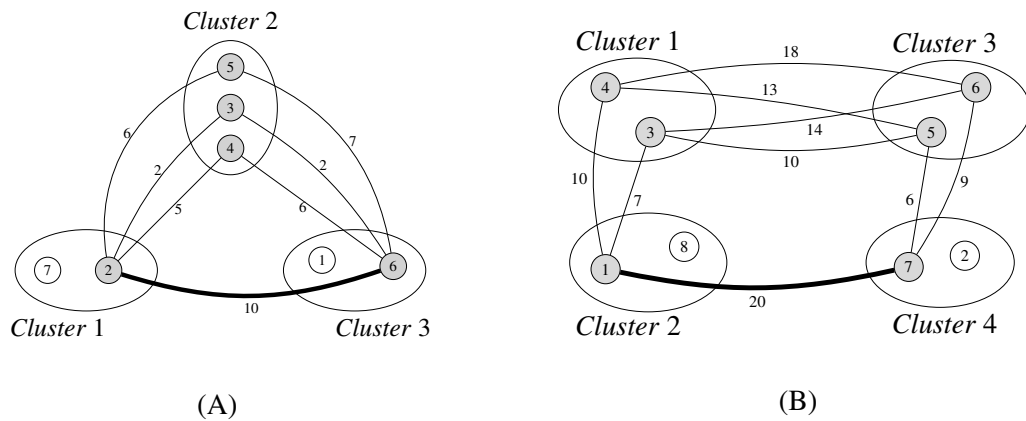
Esse processo de fixação de vértices não foi empregado durante os testes deste trabalho devido o mesmo não funcionar de maneira eficiente em instância onde o custo de suas arestas é Euclidiano, que é o caso da grande maioria das instâncias testadas. Nesse tipo de instância, há uma tendência de acontecer fixação somente dos vértices que estão nos extremos da grade de *clusters*, pois certamente esses são os mais distantes para quaisquer outros vértices da instância abordada. Acredita-se que em instância que contém um custo aleatório em suas arestas esse processo possa ser eficiente.

3.4.2 Fixando arestas em uma instância do GMSTP

Podemos realizar um pré-processamento em uma instância do GMSTP $G = (V, E)$ para eliminar arestas que garantidamente não compõem uma solução ótima. Segundo (FERREIRA *et al.*, 2012), uma aresta $\{i, j\}$ é tida como redundante se existe pelo menos um caminho ligando os vértices i e j , excluindo a aresta $\{i, j\}$, onde os vértices desse caminho são compostos por vértices da solução ótima e o custo de toda aresta desse caminho é menor ou igual ao custo de $\{i, j\}$. Como todo *cluster* de G possui um vértice na solução ótima, se existir um *cluster* V_r

intermediário entre os vértices i e j no qual o custo de $\{i, j\}$ é maior ou igual ao custo de cada aresta $\{i, w\}$ e $\{w, j\}$, para todo $w \in V_r$, então existe pelo menos um caminho que internamente contém um vértice de uma solução ótima, conectando i a j , e esse caminho contém todas as suas arestas de custo menor ou igual a aresta $\{i, j\}$, o que implica que $\{i, j\}$ pode ser eliminada. No caso da Figura 7 (A), tomando a aresta $\{2, 6\}$ e o *Cluster 2* como intermediário entre os vértices 2 e 6, tem-se que o custo 10 da aresta $\{2, 6\}$ é maior ou igual ao custo de qualquer aresta entre o vértice 2 e algum vértice do *cluster 2*, e maior ou igual ao custo de qualquer aresta entre o vértice 6 e algum vértice do *cluster 2*. Portanto, a aresta $\{2, 6\}$ pode ser eliminada.

Figura 7 – Fixando arestas em uma instância do GMSTP



Fonte: autor.

O mesmo teste de eliminação para uma aresta pode ser feito considerando dois *clusters* intermediários. Como todo *cluster* de G possui um vértice na solução ótima, se existirem dois *clusters* $r, q \in M$ intermediários que conectam os vértices i e j , tal que o custo de $\{i, j\}$ seja maior ou igual ao custo de cada aresta contida em todo caminho $\{\{i, w\}, \{w, z\}, \{z, j\}\}$, com $w \in V_r$ e $z \in V_q$ ou $w \in V_q$ e $z \in V_r$, então existe pelo menos um caminho que internamente contém dois vértices de uma solução ótima, conectando i a j , e esse caminho contém todas as suas arestas de custo menor ou igual a aresta $\{i, j\}$. No caso da Figura 7 (B), tomando a aresta $\{1, 7\}$ e os *Clusters 1* e 3 como intermediários, tem-se que o custo 20 da aresta $\{1, 7\}$ é maior ou igual ao custo de toda aresta de qualquer caminho entre os vértices 1 e 7 passando pelos *clusters* intermediários 1 e 3. Em (FERREIRA *et al.*, 2012) existe uma generalização do processo de eliminação de uma aresta para 3 ou mais *clusters* intermediários entre as suas extremidades. Entretanto, o tempo computacional para identificação desses caminhos entre os vértices i e j aumenta exponencialmente conforme cresce o número de *clusters* intermediários

entre os vértices i e j . Dessa forma, neste trabalho é feita a eliminação de arestas somente com um e dois *clusters* intermediários.

4 UMA NOVA METAHEURÍSTICA BRKGA PARA O GMSTP

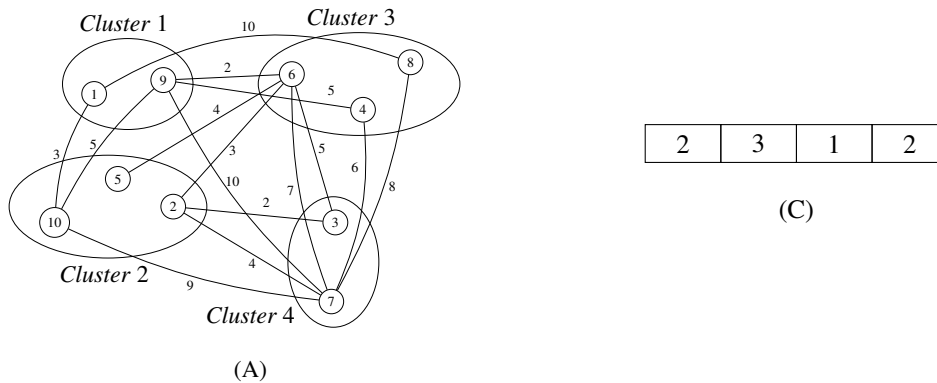
Nesta seção propomos um novo procedimento para determinar soluções viáveis para o GMSTP, o qual será usado para fornecer soluções de partida para as formulações matemáticas testadas neste trabalho. Metaheurísticas são algoritmos que buscam boas soluções a um custo computacional razoável, geralmente sem serem capazes de garantir a otimalidade, ou mesmo afirmar quão próxima a solução obtida está perto de uma solução ótima, se esta informação não estiver disponível.

Neste trabalho desenvolvemos um algoritmo genético baseado em chaves aleatórias, (denotado por BRKGA, do inglês *Biased Random-Key Genetic Algorithm*) para o GMSTP. Seguindo o trabalho sobre BRKGA (GONCALVES; RESENDE, 2011), uma solução é chamada de cromossomo e, neste trabalho, será representada por um vetor de tamanho m . A i -ésima posição do cromossomo, chamada de gene, é associada ao i -ésimo *cluster* da instância abordada e guarda qual é o vértice representante desse i -ésimo *cluster* na solução. Um algoritmo determinístico, chamado de *decoder*, recebe um cromossomo com os vértices representantes de cada *cluster* e obtém uma solução do problema baseado no grafo induzido pelos vértices representantes. Uma solução é um conjunto de arestas formando uma árvore e conectando ao menor custo os vértices representantes de cada *cluster*. O valor da solução de um cromossomo é chamado de *fitness* e é obtido usando-se um algoritmo para MST que, em nosso caso, é o algoritmo de (KRUSKAL, 1956).

4.1 População inicial

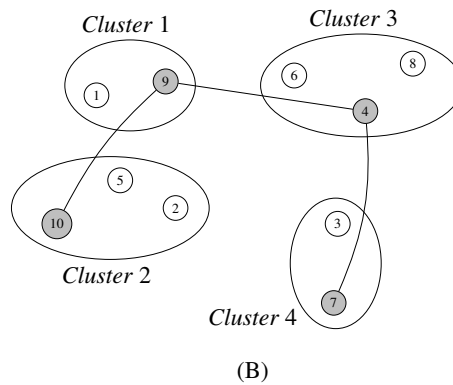
Uma população do BRKGA possui p cromossomos, cada um com m genes. Um gene i de um cromossomo recebe um número inteiro randômico sorteado do intervalo $[1, |V_i|]$, para todo *cluster* $i \in M$. No intervalo $[1, |V_i|]$, o número 1 representa o vértice de menor índice do *cluster* i , o número 2 representa o vértice de segundo menor índice do *cluster* i e assim sucessivamente. Essa representação é útil para resolver problemas de otimização combinatória que requerem permutação de números e frequentemente obtém soluções inviáveis usando o tradicional cruzamento dos AG's, com um ou dois pontos de corte. Para exemplificar essa representação, podemos observar uma solução do GMSTP presente na Figura 8 (B), obtida a partir da instância da Figura 8 (A), a qual está representada pelo cromossomo da Figura 8 (C) no BRKGA.

Figura 8 – Solução viável em (B), representada por um cromossomo em (C), para a instância do GMSTP em (A)



2	3	1	2
---	---	---	---

(C)



Fonte: autor.

Neste trabalho, o BRKGA opera com um conjunto W de populações para serem trabalhadas de forma independente. Todas as populações, usando o mesmo processo, são criadas no início da execução do algoritmo. Testes de calibração sugerem que $|W| = 4$ é um parâmetro razoável para o GMSTP, tendo cada população $p = 10$ cromossomos.

Em cada nova geração, 15% da população $w \in W$ é composta por cromossomos vindos da última geração via reprodução, 65% dela é construída via cruzamento e, por fim, o restante de 20% obtém-se via imigração. Esses três operadores genéticos são detalhados a seguir.

4.2 Operadores genéticos

Reprodução: O BRKGA usa a estratégia elitista de passar os 15% melhores cromossomos da geração corrente para a sua sucessora. A intenção dessa estratégia é garantir a não degradação da geração atual para a próxima, fornecendo e garantindo bons indivíduos para

a geração sucessora.

Cruzamento: Para se criar um cromossomo filho na g -ésima geração do BRKGA numa dada população, dois cromossomos dessa população da geração $g - 1$ são escolhidos aleatoriamente, sendo um do conjunto elite e outro do conjunto não elite. O conjunto elite de uma população numa dada geração é formado pelos cromossomos com os menores *fitness* e tem tamanho $\lfloor p/2 \rfloor$. O conjunto não elite é formado pelos demais cromossomos e tem tamanho $\lceil p/2 \rceil$. Repetição na seleção de cromossomos é permitida e, portanto, um cromossomo da geração anterior pode ter mais de um filho na geração atual. Um filho é criado como segue. Para definir o i -ésimo gene do cromossomo filho, um número aleatório α é gerado do intervalo $[0, 1)$. Se α for menor que 0.7, o filho herda o i -ésimo gene do cromossomo pai elite; caso contrário herda o i -ésimo gene do cromossomo pai não elite. Esta operação sempre gera indivíduos viáveis.

Imigração: Em uma dada geração de uma população, novos cromossomos são criados aleatoriamente e inseridos na mesma, funcionando como o típico operador de mutação dos AG's. Esse processo previne uma convergência prematura do algoritmo para um ótimo local, garantindo diversidade em cada geração.

O processo de cruzamento e imigração usa um mecanismo de interdição de vértice, descrito a seguir, para proibir por um tempo específico a seleção de um vértice que tenha sido selecionado muitas vezes. Essa proibição é baseada na frequência de seleção de cada vértice e foi baseada no mecanismo de interdição de vértices da metaheurística *Tabu Search*. Após o BRKGA executar 40% das iterações, vértices com uma porcentagem de seleção maior ou igual a 35% são proibidos de compor uma nova solução criada pelos operadores de cruzamento ou imigração. É usado um vetor para contabilizar quantas vezes cada vértice foi selecionado. A porcentagem de seleção de um vértice decresce conforme outros vértices do seu *cluster* são selecionados. Esse mecanismo ajuda o BRKGA a explorar melhor o espaço de soluções viáveis de uma instância, sendo que esse mecanismo é usado em *clusters* com pelo menos 4 vértices. Todos os valores de parâmetros do BRKGA deste trabalho foram definidos por meio de testes de calibração com um grupo de instâncias selecionadas aleatoriamente.

4.3 Operação global do BRKGA para o GMSTP

O BRKGA evolui até o número máximo de 100 iterações ou atingir 30 iterações sem atualizar o *fitness* do melhor cromossomo. Em uma dada geração g , para uma dada população $w \in$

W , exceto na fase de inicialização, a população w é formada por cromossomos vindos do conjunto elite de sua geração predecessora e por cromossomos gerados pelos operadores de cruzamento e imigração. Dessa maneira, a geração g da população w é composta de p_e cromossomos do conjunto elite de sua geração $g - 1$, p_m novos cromossomos gerados randomicamente via imigração e $p_o = p - p_m - p_e$ novos cromossomos gerados via cruzamento. As populações são trabalhadas sequencialmente e de forma independente. Após executar 10 iterações, o BRKGA copia e insere os 2 melhores cromossomos do conjunto elite de cada população w nas demais populações de W , inserindo-os no lugar dos piores indivíduos das demais populações.

O custo de cada solução é calculado pelo algoritmo adaptado de (KRUSKAL, 1956). Computar o valor de todas as novas soluções de uma dada geração é um processo que requer muito tempo. Para acelerar esse processo, o cálculo do custo de todos os novos cromossomos de uma geração é distribuído entre 8 *threads* executadas concorrentemente. Seja J a lista de novos cromossomos da geração g que precisam ter o seu *fitness* calculado. Um cromossomo é removido de J e atribuído para alguma *thread* livre. Conforme uma *thread* termina o cálculo do *fitness* de um cromossomo, ela recebe um novo cromossomo que ainda está em J . Esse processo é realizado enquanto J não for vazio. Neste trabalho, o BRKGA teve o seu tempo de execução reduzido numa média de 42% ao usar *threads*.

Todos os novos cromossomos de uma dada geração são submetidos à aplicação de uma busca local. Essa busca local é aplicada se o *fitness* do cromossomo for no máximo 25% pior do que o *fitness* do melhor cromossomo da sua geração, na sua população. Além disso, *threads* também são usadas durante a aplicação de busca local em todas as populações.

4.4 Busca local

A busca local utilizada no BRKGA é inspirada na estratégia $N^1(T)$, a qual foi definida na sub-seção 2.3. Inicialmente, dada uma solução T , o algoritmo percorre aleatoriamente todos as posições do vetor de T , sempre procurando um vértice representante de cada *cluster* que forneça uma solução de menor custo. Em seguida, seja Z a árvore geradora do grafo global que correspondente a estrutura da solução atual T . Para cada nó folha em Z , seu *cluster* correspondente é desconectado de T . Logo depois, esses *clusters* são reconectados a T pela aresta de menor custo. Nesse processo, esses *clusters* folhas podem ter seus vértices representantes atualizados por um que forneça uma solução de menor custo. Isso não altera a estrutura de T e tem um custo barato de execução, pois para reconectar esses *clusters* a T basta verificar as arestas

de menores custos que incidem sobre os *clusters* desconectados e sobre os vértices presentes em T .

4.5 Resultados computacionais obtidos com o BRKGA

O BRKGA foi implementado com a linguagem de programação C++ e executado em uma máquina com processador Intel Core i7-3770 3.40 GHz, 8 núcleos, memória RAM de 16 GB e o sistema operacional Linux Ubuntu 14.04 LTS. A partir de 20 instâncias da biblioteca TSPLIB, (ÖNCAN *et al.*, 2008) criaram um conjunto padrão de instâncias GMSTP frequentemente utilizadas na literatura. Este trabalho utilizou 80 instâncias desse conjunto padrão para realizar experimentos computacionais, contendo de 229 a 783 vértices. Essas instâncias são de 3 tipos: *euc_2d*, *geo* e *att*. As instâncias do tipo *euc_2d* têm o valor de cada aresta igual à distância Euclidiana entre seus vértices. As instâncias do tipo *att* e *geo* têm o custo de suas arestas definidos por meio de funções pseudo-Euclidianas e funções de distâncias geográficas, respectivamente, ambas definidas em (REINELT, 1991).

Essas instâncias do GMSTP foram definidas aplicando um processo de clusterização em grade em instâncias da biblioteca TSPLIB. Esse processo de clusterização foi especificado em (FISCHETTI *et al.*, 1997), o qual define uma divisão do plano cartesiano sobre qual os vértices de uma instância se encontram, sendo que cada célula corresponde a um *cluster*. Cada *cluster* de uma instância do GMSTP contém pelo menos um vértice e em média $|V|/\mu$ vértices, onde $\mu \in \{3, 5, 7, 10\}$. Cada valor de μ define um conjunto de *clusters* para uma instância GMSTP.

Da Tabela 1 a 4, os resultados obtidos com o BRKGA são comparados com os resultados obtidos pelo GRASP de (FERREIRA *et al.*, 2012), no AG de (CONTRERAS-BOLTON *et al.*, 2016) e no procedimento heurístico de duas fases de (POP *et al.*, 2017). Nessas tabelas, a primeira coluna informa o nome da **Instância** testada. A parte numérica do nome de cada instância informa quantos vértices ela possui. Na segunda coluna é informado a quantidade $|M|$ de *clusters* em cada instância. Na terceira coluna é informado os melhores *bounds* (**MB**) da literatura de uma solução viável para cada instância. Nas três colunas seguintes, são apresentados o **Custo** de solução, **GAP** e **Tempo** de execução (em segundos) encontrados pelo BRKGA na resolução de cada instância. Os tempos de execução das metaheurísticas da literatura não são relatados aqui por eles não estarem disponíveis nas respectivas referências e por não termos obtido os executáveis das mesmas.

Os valores nas colunas GAP's dessas tabelas correspondem à diferença relativa em

porcentagem entre o valor de solução obtido pela metaheurística correspondente e o melhor valor de solução conhecido na literatura para aquela instância, já considerando eventualmente melhores valores obtidos por nossa metaheurística. Na coluna **Custo** relacionada ao BRKGA, os campos em sublinhado sinalizam onde o BRKGA alcançou os valores ótimos de solução provados pelos modelos conforme relatado nas Tabelas de 6 a 9. Nas colunas **Custo** e **GAP** relacionadas ao BRKGA, os campos em **negrito** sinalizam onde o valor de solução encontrado pelo BRKGA superou os melhores resultados da literatura. Nas colunas GRASP, GA e 2F-GA são apresentados os **GAP's** das metaheurísticas GRASP de (FERREIRA *et al.*, 2012), do AG de (CONTRERAS-BOLTON *et al.*, 2016) e do procedimento heurístico de duas fases de (POP *et al.*, 2017), respectivamente. O GAP informado de cada instância, para cada metaheurística, foi obtido da mesma forma que para o BRKGA. Todas essas metaheurísticas, bem como o BRKGA desenvolvido neste trabalho, utilizaram o processo de fixação de arestas descrito na sub-seção 3.4.2 (a fixação de vértices não foi utilizada). Esse mecanismo é muito útil no processo de resolução de uma instância porque ele diminui muito o tamanho da mesma. As duas últimas linhas das Tabelas de 1 a 4 mostram a **Média** dos GAP's e o **Número de sucessos** de cada metaheurística, sendo que o **Número de sucessos** corresponde ao número de instâncias onde cada metaheurística alcançou ou obteve os melhores valores de solução viável da literatura.

Na Tabela 1 analisamos os resultados computacionais de 20 instâncias criadas com o parâmetro $\mu = 3$. Desse total, observamos que o BRKGA supera os melhores resultados da literatura em 1 instâncias. Considerando a média dos GAP's, o BRKGA consegue superar apenas a metaheurística GA. A média dos GAP's em porcentagem das metaheurísticas BRKGA, GRASP, GA e 2F-GA são 0.0075%, 0.0060%, 0.0163% e 0.0058%, respectivamente. De 20 instâncias, a quantidade de instâncias onde o BRKGA, o GRASP, o GA e o 2F-GA alcançou os melhores valores de solução da literatura são de 13, 16, 15 e 17, respectivamente. Para 6 instâncias o BRKGA alcançou os valores ótimos conhecidos. A metaheurística que teve o melhor desempenho computacional para o grupo de instâncias da Tabela 1 foi a 2F-GA em termos de GAP médio e número de acertos.

Na Tabela 2 analisamos os resultados computacionais de 20 instâncias criadas com o parâmetro $\mu = 5$. Desse total, observamos que o BRKGA supera os melhores resultados de custo da literatura em 2 instâncias. Analisando a média dos GAP's, o BRKGA supera as demais metaheurísticas. A média dos GAP's em porcentagem das metaheurísticas BRKGA, GRASP, GA e 2F-GA são 0.0202%, 0.0242%, 0.0411% e 0.0242%, respectivamente. De 20 instâncias,

a quantidade de instâncias onde o BRKGA, o GRASP, o GA e o 2F-G alcançou os melhores valores de solução da literatura são de 14, 16, 15 e 16, respectivamente. Para 8 instâncias o BRKGA alcançou os valores ótimos conhecidos. Para o grupo de instâncias da Tabela 2, a metaheurística BRKGA obteve melhor desempenho computacional em relação ao GAP médio.

Na Tabela 3 analisamos os resultados computacionais de 20 instâncias criadas com o parâmetro $\mu = 7$. Desse total, observamos que o BRKGA supera os melhores resultados da literatura em 3 instâncias. Para o grupo de instâncias dessa tabela, o BRKGA supera todas as metaheurísticas da literatura tanto em termos de quantidade de melhores soluções obtidas quanto em relação ao GAP médio. A média dos GAP's em porcentagem das metaheurísticas BRKGA, GRASP, GA e 2F-GA são 0.0134%, 0.0471%, 0.0382% e 0.0353%, respectivamente. De 20 instâncias, a quantidade de instâncias onde o BRKGA, o GRASP, o GA e o 2F-G alcançou ou superou os melhores valores de solução da literatura são de 16, 11, 13 e 14, respectivamente. Para 9 instâncias o BRKGA alcançou os valores ótimos conhecidos.

Na Tabela 4 analisamos os resultados computacionais de 20 instâncias criadas com o parâmetro $\mu = 10$. Desse total, observamos que o BRKGA supera os melhores resultados da literatura para 3 instâncias. Para o grupo de instâncias dessa tabela, o BRKGA supera todas as metaheurísticas da literatura tanto em termos de quantidade de melhores soluções obtidas quanto em relação ao GAP médio. A média dos GAP's em porcentagem das metaheurísticas BRKGA, GRASP, GA e 2F-GA são 0.0005%, 0.0237%, 0.0237% e 0.0237%, respectivamente. De 20 instâncias, a quantidade de instâncias onde o BRKGA, o GRASP, o GA e o 2F-G alcançou os melhores valores de solução da literatura são de 19, 17, 17 e 17, respectivamente. Para 8 instâncias o BRKGA alcançou os valores ótimos conhecidos. As metaheurísticas GRASP, GA e 2F-GA tiveram desempenho idênticos e apresentam pior desempenho que o BRKGA quanto ao GAP médio e número de acertos.

Levando em conta os resultados apresentados nas Tabelas de 1 a 4, concluímos que o BRKGA mostrou melhor performance competitiva e algumas vezes superior em termos de qualidade das soluções e GAP médio. Comparado às melhores metaheurísticas da literatura, nossa metaheurística conseguiu superar os melhores valores de solução para 9 instâncias, de um conjunto padrão de testes com 80 instâncias.

Tabela 1 – Resultados de instâncias *benchmark* usando clusterização em grade, com $\mu = 3$

Instância	M	MB	BRKGA			GRASP	GA	2F-GA
			Custo	GAP	Tempo	GAP	GAP	GAP
ali535	181	134306	134354	0.0357	4631	0.0417	0.0417	0.0417
att532	182	15652	15652	0.0000	54	0.0000	0.0000	0.0000
d493	171	20269	20269	0.0000	36	0.0000	0.0000	0.0000
d657	221	25703	<u>25704</u>	0.0039	5193	0.0039	0.0000	0.0000
fl417	142	8353	<u>8353</u>	0.0000	66	0.0000	0.0000	0.0000
gil262	95	1255	1255	0.0000	4	0.0000	0.0000	0.0000
gr229	81	74792	<u>74819</u>	0.0361	262	0.0000	0.0000	0.0000
gr431	149	103805	103805	0.0000	3009	0.0376	0.0376	0.0376
gr666	224	174655	<u>174655</u>	0.0000	4838	0.0000	0.0000	0.0000
lin318	108	24092	<u>24092</u>	0.0000	3	0.0000	0.0000	0.0000
p654	230	23547	23548	0.0042	3556	0.0000	0.0000	0.0000
pcb442	156	27513	<u>27513</u>	0.0000	100	0.0000	0.0000	0.0000
pr264	101	29199	29199	0.0000	1	0.0000	0.0000	0.0000
pr299	102	23096	23096	0.0000	6	0.0000	0.0000	0.0000
pr439	163	64953	64953	0.0000	16	0.0000	0.0000	0.0000
rat575	196	2880	2880	0.0000	4101	0.0000	0.1389	0.0000
rat783	285	4203	4204	0.0238	5036	0.0000	0.0714	0.0000
rd400	135	7632	7632	0.0000	44	0.0000	0.0000	0.0000
u574	198	19696	19698	0.0102	617	0.0000	0.0000	0.0000
u724	266	21739	21747	0.0368	3508	0.0368	0.0368	0.0368
Média				0.0075		0.0060	0.0163	0.0058
Número de sucessos				13		16	15	17

Fonte: autor.

Tabela 2 – Resultados de instâncias *benchmark* usando clusterização em grade, com $\mu = 5$

Instância	M	MB	BRKGA			GRASP	GA	2F-GA
			Custo	GAP	Tempo	GAP	GAP	GAP
ali535	108	107936	107936	0.0000	4109	0.2455	0.2455	0.2455
att532	110	11896	<u>11896</u>	0.0000	493	0.0000	0.0000	0.0000
d493	102	16132	<u>16132</u>	0.0000	510	0.0000	0.0000	0.0000
d657	137	19826	19827	0.0050	1390	0.0000	0.0000	0.0000
fl417	93	7952	<u>7952</u>	0.0000	3	0.0000	0.0000	0.0000
gil262	63	984	984	0.0000	6	0.0000	0.0000	0.0000
gr229	47	54236	54236	0.0000	55	0.1272	0.1272	0.1272
gr431	87	81856	<u>81963</u>	0.1307	305	0.0000	0.0000	0.0000
gr666	139	144077	144225	0.1027	861	0.0000	0.0000	0.0000
lin318	64	17667	<u>17667</u>	0.0000	2	0.0000	0.0000	0.0000
p654	134	22377	<u>22377</u>	0.0000	121	0.0000	0.0000	0.0000
pcb442	95	19383	<u>19383</u>	0.0000	537	0.0000	0.0000	0.0000
pr264	55	21351	21351	0.0000	2	0.0000	0.0000	0.0000
pr299	69	18582	18582	0.0000	7	0.0000	0.0000	0.0000
pr439	96	54230	54230	0.0000	10	0.0000	0.0000	0.0000
rat575	121	2014	2015	0.0497	1130	0.0000	0.0000	0.0000
rat783	169	2823	2823	0.0000	1699	0.0000	0.0708	0.0000
rd400	81	5433	5434	0.0184	481	0.0331	0.0184	0.0331
u574	127	15240	15255	0.0984	974	0.0787	0.3609	0.0787
u724	166	15435	15435	0.0000	419	0.0000	0.0000	0.0000
Média				0.0202		0.0242	0.0411	0.0242
Número de sucessos				14		16	15	16

Fonte: autor.

Tabela 3 – Resultados de instâncias *benchmark* usando clusterização em grade, com $\mu = 7$

Instância	M	MB	BRKGA			GRASP		GA	2F-GA
			Custo	GAP	Tempo	GAP	GAP	GAP	
ali535	83	93905	93905	0.0000	1149	0.2598	0.2598	0.2598	
att532	80	10204	<u>10204</u>	0.0000	83	0.0000	0.0000	0.0000	
d493	78	14152	<u>14152</u>	0.0000	29	0.0000	0.0000	0.0000	
d657	96	16542	<u>16542</u>	0.0000	3236	0.0000	0.0000	0.0000	
fl417	61	7446	<u>7446</u>	0.0000	1	0.0000	0.0000	0.0000	
gil262	49	802	802	0.0000	212	0.0748	0.0000	0.0000	
gr229	34	46030	<u>46030</u>	0.0000	101	0.0413	0.0413	0.0413	
gr431	64	71318	71318	0.0000	666	0.1360	0.1360	0.1360	
gr666	96	119271	<u>119284</u>	0.0109	4208	0.0000	0.0000	0.0000	
lin318	49	14909	<u>14909</u>	0.0000	3	0.0000	0.0000	0.0000	
p654	100	21770	<u>21770</u>	0.0000	51	0.0000	0.0000	0.0000	
pcb442	64	14644	<u>14644</u>	0.0000	758	0.0082	0.0000	0.0000	
pr264	43	20438	<u>20438</u>	0.0000	128	0.0000	0.0000	0.0000	
pr299	47	15238	15238	0.0000	5	0.0000	0.0000	0.0000	
pr439	74	47101	47101	0.0000	6	0.0000	0.0000	0.0000	
rat575	100	1734	1735	0.0577	1885	0.0923	0.0577	0.0000	
rat783	121	2228	2230	0.0898	3281	0.0449	0.0449	0.0449	
rd400	64	4576	4581	0.1093	864	0.1093	0.1093	0.1093	
u574	92	12186	12186	0.0000	433	0.0000	0.0000	0.0000	
u724	119	13086	13086	0.0000	2847	0.1758	0.1146	0.1146	
Média				0.0134		0.0471	0.0382	0.0353	
Número de sucessos				16		11	13	14	

Fonte: autor.

Tabela 4 – Resultados de instâncias *benchmark* usando clusterização em grade, com $\mu = 10$

Instância	M	MB	BRKGA			GRASP		GA	2F-GA
			Custo	GAP	Tempo	GAP	GAP	GAP	
ali535	57	73207	73207	0.0000	305	0.2076	0.2076	0.2076	
att532	57	8497	<u>8497</u>	0.0000	388	0.0000	0.0000	0.0000	
d493	52	11121	<u>11121</u>	0.0000	5	0.0000	0.0000	0.0000	
d657	73	13495	<u>13495</u>	0.0000	305	0.0000	0.0000	0.0000	
fl417	42	6986	<u>6986</u>	0.0000	2	0.0000	0.0000	0.0000	
gil262	36	639	639	0.0000	2	0.0000	0.0000	0.0000	
gr229	23	39793	<u>39797</u>	0.0101	31	0.0000	0.0000	0.0000	
gr431	52	62688	62688	0.0000	150	0.0542	0.0542	0.0542	
gr666	70	96992	96992	0.0000	493	0.2124	0.2124	0.2124	
lin318	36	10119	<u>10119</u>	0.0000	2	0.0000	0.0000	0.0000	
p654	69	20739	<u>20739</u>	0.0000	235	0.0000	0.0000	0.0000	
pcb442	48	11941	11941	0.0000	6	0.0000	0.0000	0.0000	
pr264	27	16546	16546	0.0000	2	0.0000	0.0000	0.0000	
pr299	35	11624	11624	0.0000	2	0.0000	0.0000	0.0000	
pr439	48	40518	40518	0.0000	4	0.0000	0.0000	0.0000	
rat575	64	1235	1235	0.0000	10	0.0000	0.0000	0.0000	
rat783	81	1682	1682	0.0000	230	0.0000	0.0000	0.0000	
rd400	49	3825	3825	0.0000	175	0.0000	0.0000	0.0000	
u574	64	9755	9755	0.0000	97	0.0000	0.0000	0.0000	
u724	80	9608	9608	0.0000	483	0.0000	0.0000	0.0000	
Média				0.0005		0.0237	0.0237	0.0237	
Número de sucessos				19		17	17	17	

Fonte: autor.

5 NOVA FORMULAÇÃO MATEMÁTICA E UM ESQUEMA DE DECOMPOSIÇÃO DE BENDERS PARA O GMSTP

Apresentamos a seguir uma formulação matemática multigrafo inédita e um procedimento de decomposição de Benders baseado em uma formulação da literatura para o GMSTP. Tanto quanto sabemos, essa decomposição ainda não havia sido aplicada para o GMSTP.

5.1 Nova formulação polinomial baseada em multigrafo para o GMSTP

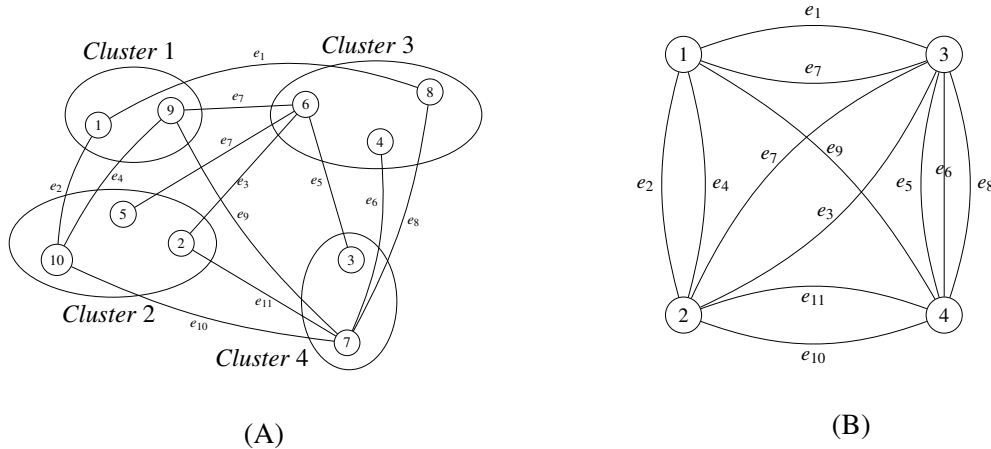
Apresentamos agora outra contribuição original desta tese. Uma nova formulação polinomial para o GMSTP com base em multigrafo inspirada na formulação de (ANDRADE, 2014), a qual determina uma floresta de tamanho conhecido em qualquer tipo de grafo: completo ou não, direcionado ou não, e tendo uma ou mais conexões duplicadas (arestas ou arcos ou ambas) entre dois vértices. Essa formulação faz uso de um multigrafo $G^\circ = (V^\circ, E^\circ)$, obtido a partir de uma instância $G = (V, E)$ do GMSTP. Cada vértice de V° é associado exclusivamente a um *cluster* de G . Cada aresta $e \in E^\circ$ é associada exclusivamente a uma aresta $e \in E$, de maneira que essas arestas possuem a mesma identificação e custo em G e G° , com $|E| = |E^\circ|$. Se r e q são as extremidades de uma aresta $\{r, q\} \in E^\circ$ numa solução de G° , então sua correspondente aresta $\{i, j\} \in E$ pertence a uma solução GST de G , cujas extremidades r e q correspondem aos respectivos *clusters* aos quais i e j pertencem. A Figura 9 (B) ilustra o multigrafo G° obtido a partir da instância GMSTP ilustrada na Figura 9 (A). A ideia geral dessa formulação é definir $|V^\circ|$ arborescências simultaneamente, uma para cada vértice (raiz) referencial $k \in V^\circ$. Cada vértice referencial define uma orientação independente para toda aresta presente na solução de G° , de acordo com o teorema de caracterização de florestas em um dado grafo presente no trabalho de (ADASME *et al.*, 2015) e reproduzido abaixo. Esse teorema garante que não haverá ciclos numa solução de G° .

Teorema 5.1.1 (ADASME *et al.*, 2015) *Dado um subgrafo F de um grafo $G^\circ = (V^\circ, E^\circ)$, não existe ciclo em F se para cada vértice $k \in V^\circ$ tido como (raiz) referencial existe uma orientação das arestas de F , todas independentes e satisfazendo simultaneamente as seguintes condições:*

- (a) *Nenhuma aresta de F incidente a um vértice referencial k está orientada em direção a k ;*
- (b) *Para todo vértice $u \in V^\circ \setminus \{k\}$, no máximo uma aresta de F incidente a u está orientada em direção a u .*

Prova: Considere um subgrafo F de $G^\circ = (V^\circ, E^\circ)$, onde exista um ciclo $C = [V^\circ(C), E^\circ(C)]$,

Figura 9 – Grafo de entrada $G = (V, E)$ em (A) e seu correspondente multigrafo $G^\circ = (V^\circ, E^\circ)$ em (B)



Fonte: autor.

com $V^\circ(C)$ e $E^\circ(C)$ sendo o conjunto de vértices e arestas de C , respectivamente. O subgrafo F não satisfaz uma das condições acima para todo vértice referencial $k \in V^\circ(C)$. Em qualquer orientação das arestas de $E^\circ(C)$ existirá pelo menos dois arcos incidentes em qualquer vértice de $V^\circ(C)$: dois arcos saindo e dois arcos entrando, ou um arco saindo e outro entrando em cada vértice desse conjunto. Distinguímos duas possibilidades. Primeiramente, se existir mais de um arco saindo de um vértice referencial k de C , então existirá mais de um arco entrando em algum outro vértice de C , o que viola a condição (b). Segundo, se os arcos de C forem simbolicamente orientados no sentido horário (ou anti-horário), então a condição (a) será violada porque existe uma orientação de aresta apontando para todo vértice de C , considerando um vértice de C como referencial. Dessa forma, se F contém ciclo, não existe a possibilidade de a orientação das arestas de F satisfazer simultaneamente as condições (a) e (b) do Teorema 5.1.1. Portanto, não existe ciclo em F . ■

Na formulação multigrafo a seguir, a notação $\rho(s)$ representa o *cluster* ao qual pertence o vértice $s \in V$. Cada variável y_s indica se o vértice $s \in V$ pertence ($y_s = 1$) ou não ($y_s = 0$) à solução. Uma variável x_e indica se a aresta $e \in E$ (ou equivalente se $e \in E^\circ$) pertence ($x_e = 1$) ou não ($x_e = 0$) à solução, sendo que essa aresta é representada por $e = \{i(e), j(e)\}$, onde $i(e)$ e $j(e)$ denotam os vértices i e j dessa aresta. Para cada vértice referencial $k \in V^\circ$, uma arborescência enraizada em k é induzida por variáveis $w_e^{k,0}$ e $w_e^{k,1}$, as quais estabelecem uma orientação para cada aresta $e \in E^\circ$ presente na solução. Para todo vértice referencial $k \in V^\circ$, se $w_e^{k,0} = 1$, então a aresta $e = \{\rho(i(e)), \rho(j(e))\} \in E$ pertence à solução e ela é orientada do *cluster* $\rho(j(e))$ para o *cluster* $\rho(i(e))$ com relação ao vértice referencial k ; caso contrário, $w_e^{k,0} = 0$. De maneira similar, para todo vértice referencial $k \in V^\circ$, se $w_e^{k,1} = 1$, então a aresta

$e = \{\rho(i(e)), \rho(j(e))\} \in E$ pertence à solução e ela é orientada do *cluster* $\rho(i(e))$ para o *cluster* $\rho(j(e))$ com relação ao vértice referencial k ; caso contrário, $w_e^{k,1} = 0$. Esta formulação multigrafo é similar à formulação (\mathcal{F}_{Martin}) da página 25, onde as variáveis w e λ tem conceitos parecidos. Nossa formulação multigrafo é descrita a seguir.

$$(\mathcal{F}_{MG}) \quad \min \sum_{e \in E^\circ} c_e x_e$$

$$\text{s.a: } \sum_{s \in V_k} y_s = 1 \quad \forall k \in M \quad (5.1)$$

$$\sum_{e \in E^\circ} x_e = m - 1 \quad (5.2)$$

$$x_e - w_e^{k,0} - w_e^{k,1} = 0, \quad \forall e \in E^\circ, \forall k \in V^\circ \quad (5.3)$$

$$\sum_{e \in E^\circ | i(e)=u} w_e^{k,0} + \sum_{e \in E^\circ | j(e)=u} w_e^{k,1} = 1, \quad \forall k \in V^\circ, \forall u \in V^\circ \setminus \{k\} \quad (5.4)$$

$$w_e^{k,0} = 0, \quad \forall k \in V^\circ : k = i(e), \forall e \in E^\circ \quad (5.5)$$

$$w_e^{k,1} = 0, \quad \forall k \in V^\circ : k = j(e), \forall e \in E^\circ \quad (5.6)$$

$$x_{ij} \leq y_i, \quad \forall \{i, j\} \in E \quad (5.7)$$

$$x_{ij} \leq y_j, \quad \forall \{i, j\} \in E \quad (5.8)$$

$$0 \leq w \leq 1, \quad 0 \leq x \leq 1, \quad y \in \mathbb{B}^{|V|} \quad (5.9)$$

A restrição (5.1) impõe que apenas um vértice de cada *cluster* faz parte da solução. A restrição (5.2) define que exatamente $m - 1$ arestas pertencem à solução. As restrições (5.3) determinam que se $w_e^{k,0} = 1$ ou $w_e^{k,1} = 1$, então a aresta $e \in E^\circ$ pertence à solução. As restrições (5.4) garantem que para um dado vértice referencial $k \in V^\circ$, em todo vértice $u \in V^\circ \setminus \{k\}$, as variáveis $w_e^{k,*}$ definem que apenas uma das arestas incidentes em u será orientada na direção de u . As restrições (5.5) e (5.6) estabelecem que não existe arestas orientadas na direção do vértice referencial $k \in V^\circ$. As restrições (5.7) e (5.8) determinam que se um vértice de V não fizer parte da solução, então nenhuma aresta de E presente na solução incidirá nele. O domínio das variáveis é definido em (5.9).

A relaxação linear de (\mathcal{F}_{MG}) pode encontrar uma solução fracionária do GMSTP. Um exemplo disso é mostrado na Figura 10 (B), na qual é exibida uma solução fracionária de custo 6.5 para o grafo da Figura 10 (A). Para fortalecer a relaxação linear da formulação (\mathcal{F}_{MG}) e resolver o problema do vértice isolado da Figura 10 (B), o vértice 1, pode-se acrescentar uma

restrição especificando que, na solução do problema, a soma das variáveis de arestas incidentes em um vértice é pelo menos o valor da variável correspondente a esse vértice, conforme segue.

$$\sum_{j \in N(i)} x_{ij} \geq y_i \quad \forall i \in V, \forall \{i, j\} \in E \quad (5.10)$$

Acrescentando as restrições (5.10) em (\mathcal{F}_{MG}) , é possível que na relaxação linear dessa formulação fortalecida $(\mathcal{F}_{MG}) \cap (5.10)$, haja ainda o problema do aparecimento de arestas conflitantes numa solução, com arestas incidindo sobre vértices distintos de um mesmo *cluster*, como pode ser observado Figura 10 (C), na qual é exibida uma solução fracionária de custo 6.5 de $(\mathcal{F}_{MG}) \cap (5.10)$ para o grafo da Figura 10 (A). Nessa figura, pode-se observar a aresta $\{4, 5\}$ conflitante com o conjunto de arestas $\{\{2, 3\}, \{2, 7\}, \{6, 9\}\}$; a aresta $\{6, 9\}$ conflitante com o conjunto de arestas $\{\{3, 4\}, \{4, 5\}, \{4, 9\}\}$, entre outros conflitos. As restrições (2.32) do trabalho de (POP, 2002) evitam o aparecimento do par de arestas conflitantes $\{\{2, 3\}, \{2, 7\}\}$.

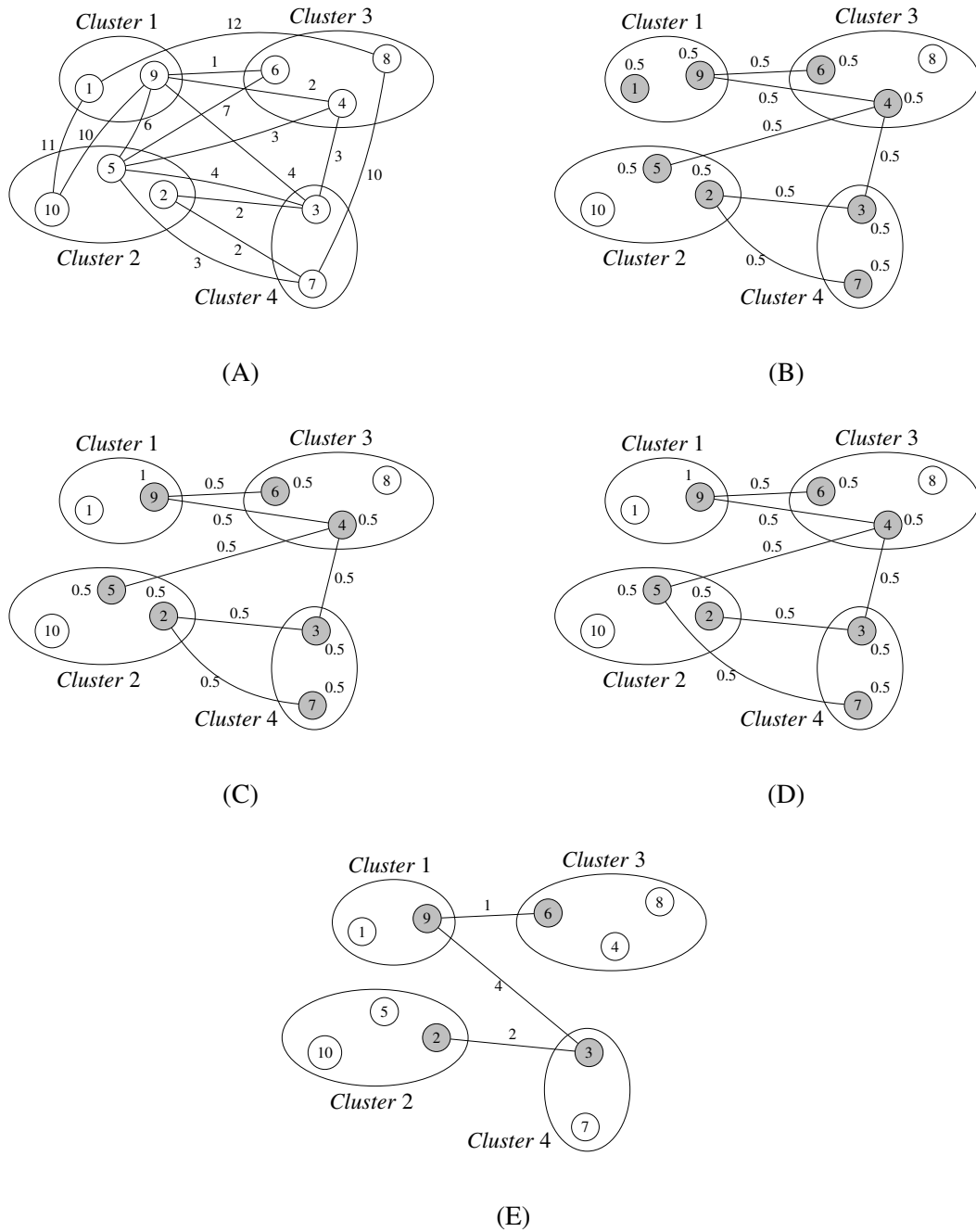
As restrições (2.32) impedem que arestas incidam em um vértice não pertencente à solução, além de limitar a no máximo y_i a soma das arestas conectando um vértice i a diferentes vértices de outro *cluster* distinto do *cluster* de i . A formulação (\mathcal{F}_{MG}) fortalecida com as restrições (5.10) e (2.32) ainda pode originar soluções com arestas conflitantes na relaxação linear, conforme descrito na Figura 10 (D), a qual é uma solução fracionária de custo 7 do grafo da Figura 10 (A) e tem as arestas conflitantes $\{2, 3\}$ e $\{5, 7\}$, por exemplo.

Uma alternativa para resolver o problema de arestas conflitantes da solução da Figura 10 (D) é fazer uso da estrutura das arborescências definidas em (\mathcal{F}_{MG}) , especificando que para cada arborescência enraizada em algum vértice de um *cluster* raiz V_k , com $k \in M$, em todo vértice s pertencente a um *cluster* V_u , com $u \neq k$, a quantidade de arestas orientadas para s é exatamente y_s . Note que, nesse caso, se um vértice não pertencer à solução, então todas as variáveis w associadas a esse vértice serão fixadas em zero.

$$\sum_{\substack{e \in E \\ i(e)=s}} w_{\{\rho(i(e)), \rho(j(e))\}}^{k,0} + \sum_{\substack{e \in E \\ j(e)=s}} w_{\{\rho(i(e)), \rho(j(e))\}}^{k,1} = y_s, \quad \forall k \in V^\circ, \forall u \in V^\circ \setminus \{k\}, \forall s \in V_u \quad (5.11)$$

As variáveis w em (5.11) são as mesmas variáveis da formulação (\mathcal{F}_{MG}) . Essas restrições fortalecem e seguem a ideia geral das restrições (5.4). No caso da solução relaxada da Figura 10 (D), a mesma viola uma restrição (5.11). Além disso, as restrições (5.11) solucionam também os problemas relacionados na Figura 10 (B) e (C), sendo que elas dominam as restrições

Figura 10 – Instância GMSTP em (A). Solução ótima em (E) de valor 7 para essa instância. Em (B), (C) e (D) soluções fracionárias da relaxação linear do modelo (\mathcal{F}_{MG}) fortalecido das restrições (5.10); (5.10) \cap (2.32); e (5.10) \cap (2.32) \cap (5.11), respectivamente



Fonte: autor.

(2.32) e (5.10). Na Figura 10 (E) é exibida uma solução inteira de custo 7 na relaxação linear da formulação $(\mathcal{F}_{MG}) \cap (5.10) \cap (2.32) \cap (5.11)$ para o grafo da Figura 10 (A). Como essa solução relaxada é inteira, então ela é ótima para a instância da Figura 10 (A).

Feito as observações anteriores, denominaremos a formulação em que as restrições (5.7) e (5.8) são substituídas pelas restrições (5.11) por (\mathcal{F}_{MG}^*) . A formulação (\mathcal{F}_{MG}^*) contém $(|V| + |E^\circ| + 2|V^\circ| \times |E^\circ|)$ variáveis (somente as $|V|$ variáveis y precisam ser binárias) e $(1 + |V^\circ| + |V^\circ||E^\circ| + |V^\circ|(|V^\circ| - 1)(|V| + 1))$ restrições.

As colunas das variáveis x e w formam uma matriz total dual inteira, conforme é afirmado em (ANDRADE, 2014). Para um dado y satisfazendo (5.1), colunas x ou w associadas a arestas com mesmas extremidades (*clusters*) compartilham os mesmos coeficientes de suas respectivas colunas. Sabe-se que, eliminadas colunas repetidas no sistema resultante, o que fica é uma matriz TDI conforme resultado de (MARTIN, 1991). Note que, exceto pelos limites das variáveis x , w e y , todas as demais restrições do novo modelo são satisfeitas na igualdade. Além disso, podemos estabelecer algumas propriedades em relação a (\mathcal{F}_{MG}^*) .

Proposição 5.1.2 *A restrição (5.2) não é necessária para a formulação (\mathcal{F}_{MG}^*) com a presença das restrições (5.3) - (5.6).*

Prova: As restrições (5.3) - (5.6) garantem que todos os *clusters* tem arestas incidindo neles, sendo que para cada arborescência não existe orientação de arestas entrando no *cluster* referencial e existe exatamente uma orientação de aresta entrando em cada *cluster* não referencial, os quais são em número de $m - 1$. Pelo teorema de caracterização de floresta, não há ciclos. Logo, (5.2) é redundante. ■

Proposição 5.1.3 *As restrições (5.4) são dominadas pelas restrições (5.11).*

Prova: Para qualquer *cluster* referencial $k \in V^\circ$, basta considerar individualmente todo vértice selecionado em cada *cluster* não referencial $u \in V^\circ \setminus \{k\}$. Como existirá uma orientação entrando no vértice representante de $u \in V^\circ \setminus \{k\}$ garantido por (5.11), automaticamente tem-se uma orientação entrando em cada *cluster* $u \in V^\circ \setminus \{k\}$, o que torna (5.4) uma restrição dominada. ■

As Proposições (5.1.2) e (5.1.3) nos permitem testar diferentes implementações de (\mathcal{F}_{MG}^*) . A seguir apresentamos a última abordagem de resolução que desenvolvemos para o GMSTP.

5.2 Decomposição de Benders

Nesta seção exploramos a formulação multifluxo (\mathcal{F}_{mflow}) de (MYUNG *et al.*, 1995) (ver página 23) e a aplicamos de forma automática no método de decomposição de Benders disponível no *solver* CPLEX. Essa decomposição divide o problema original em um problema mestre e um sub-problema escravo. As variáveis inteiras complicadoras do problema original são associadas ao problema mestre e, as variáveis restantes, ao sub-problema escravo. No caso de (\mathcal{F}_{mflow}), as variáveis inteiras complicadoras são as relacionadas às conexões w e aos representantes y dos *clusters*. Resolvido o problema mestre, tem-se uma solução \bar{w} e \bar{y} a qual é usada para resolver o sub-problema escravo, da seguinte maneira:

$$\begin{aligned}
 (\mathcal{F}_{sub-prob}) \quad & \min \sum_{(i,j) \in A} c_{ij} \bar{w}_{ij} \\
 \text{s.a:} \quad & \sum_{(i,j) \in A} f_{ij}^k - \sum_{(j,i) \in A} f_{ij}^k = \begin{cases} \bar{y}_i, & \forall i \in V_1 \\ -\bar{y}_i, & \forall i \in V_k \\ 0, & \forall i \notin V_1 \cup V_k \end{cases} \forall k \in M \setminus \{1\}
 \end{aligned} \tag{5.12}$$

$$f_{ij}^k \leq \bar{w}_{ij}, \quad \forall (i,j) \in A; \forall k \in M \setminus \{1\} \tag{5.13}$$

$$f_{ij}^k \geq 0, \quad \forall (i,j) \in A; \forall k \in M \setminus \{1\} \tag{5.14}$$

($\mathcal{F}_{sub-prob}$) é um problema contínuo linear cuja função objetivo é uma constante e pode ser dualizado como segue. Associamos um vetor de variáveis duais $\mu \in \mathbb{R}^{|V||M|}$ de componentes μ_i^k , para todo vértice $i \in V$ e todo *cluster* $k \in M$; e um vetor de variáveis duais $\tau \in \mathbb{R}^{|A||M|}$ de componentes τ_{ij}^k , para todo arco $(i,j) \in A$ e todo $k \in M$, às restrições (5.12) e (5.13), respectivamente, obtendo o problema equivalente.

$$\begin{aligned}
 (\mathcal{F}_{dual_subp}) \quad & \sum_{c_{ij} \in A} c_{ij} \bar{w}_{ij} + \max_{k \in M \setminus \{1\}} \sum \left[\sum_{i \in V_1} \bar{y}_i \mu_i^k - \sum_{i \in V_k} \bar{y}_i \mu_i^k + \sum_{(i,j) \in A} \bar{w}_{ij} \tau_{ij}^k \right] \\
 \text{s.a:} \quad & \mu_i^k - \mu_j^k + \tau_{ij}^k \leq 0 \quad \forall (i,j) \in A, \forall k \in M \setminus \{1\}
 \end{aligned} \tag{5.15}$$

$$\tau \leq 0 \tag{5.16}$$

$$\mu \text{ irrestrito} \tag{5.17}$$

A região viável de (\mathcal{F}_{dual_subp}) não depende dos valores de \bar{w} de \bar{y} . Se essa região for vazia, então ou ($\mathcal{F}_{sub-prob}$) é ilimitado, o que implica (\mathcal{F}_{mflow}) ser ilimitado; ou a região

viável de $(\mathcal{F}_{sub-prob})$ é vazia, o que implica que (\mathcal{F}_{mflow}) é inviável. Considerando que a região viável definida pelo dual do sub-problema não seja vazia, podemos enumerar um conjunto de raios extremos Q e um conjunto de pontos extremos D do poliedro definido pelo sub-problema $(\mathcal{F}_{dual_subp})$.

Dado o conjunto de raios extremos Q não vazio da região viável de $(\mathcal{F}_{dual_subp})$, existe uma direção ilimitada $(\hat{\mu}, \hat{\tau}) \in Q$ associada a um par ordenado (\hat{y}, \hat{w}) referente a uma solução inviável de (\mathcal{F}_{mflow}) . Assim, estamos interessados apenas em soluções (y, w) de (\mathcal{F}_{mflow}) para as quais:

$$\sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} y_i \hat{\mu}_i^k - \sum_{i \in V_k} y_i \hat{\mu}_i^k + \sum_{(i,j) \in A} w_{ij} \hat{\tau}_{ij}^k \right] \leq 0, \quad \forall (\hat{\mu}, \hat{\tau}) \in Q \quad (5.18)$$

As restrições (5.18) são denominadas de cortes de viabilidade em um esquema de decomposição de Benders para (\mathcal{F}_{mflow}) . Por outro lado, dado o conjunto de pontos extremos D não vazio da região de $(\mathcal{F}_{dual_subp})$, a cada solução viável limitada $(\hat{\mu}, \hat{\tau}) \in D$ associada a um par ordenado (\hat{y}, \hat{w}) referente a uma solução viável de (\mathcal{F}_{mflow}) , existe associado um valor $\hat{\eta}$ para a função objetivo desse problema de maximização; ou seja:

$$\hat{\eta} = \sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} \hat{y}_i \hat{\mu}_i^k - \sum_{i \in V_k} \hat{y}_i \hat{\mu}_i^k + \sum_{(i,j) \in A} \hat{w}_{ij} \hat{\tau}_{ij}^k \right] \quad (5.19)$$

Em particular, para uma solução viável genérica (w, y) de (\mathcal{F}_{mflow}) cuja solução de $(\mathcal{F}_{dual_subp})$ seja $(\mu, \tau) \in D$ de valor η , pela maximalidade de (μ, τ) em relação às demais soluções viáveis vale que:

$$\eta = \sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} y_i \mu_i^k - \sum_{i \in V_k} y_i \mu_i^k + \sum_{(i,j) \in A} w_{ij} \tau_{ij}^k \right] \geq \sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} (y_i - \hat{y}_i + \hat{y}_i) \hat{\mu}_i^k - \sum_{i \in V_k} (y_i - \hat{y}_i + \hat{y}_i) \hat{\mu}_i^k + \sum_{(i,j) \in A} (w_{ij} - \hat{w}_{ij} + \hat{w}_{ij}) \hat{\tau}_{ij}^k \right] \quad (5.20)$$

Logo obtemos

$$\eta \geq \sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} (y_i - \hat{y}_i) \hat{\mu}_i^k - \sum_{i \in V_k} (y_i - \hat{y}_i) \hat{\mu}_i^k + \sum_{(i,j) \in A} (w_{ij} - \hat{w}_{ij}) \hat{\tau}_{ij}^k \right] + \hat{\eta} \quad (5.21)$$

para todo ponto extremo $(\hat{\mu}, \hat{\tau}) \in D$; ou de forma simplificada, sem relacionar o valor genérico η com $\hat{\eta}$ associado a um par $(\hat{\mu}, \hat{\tau}) \in D$, temos

$$\eta \geq \sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} y_i \hat{\mu}_i^k - \sum_{i \in V_k} y_i \hat{\mu}_i^k + \sum_{(i,j) \in A} w_{ij} \hat{\tau}_{ij}^k \right], \quad \forall (\hat{\mu}, \hat{\tau}) \in D \quad (5.22)$$

A expressão (5.22) é denominada de corte de otimalidade. Considerando todos os raios e pontos extremos de $(\mathcal{F}_{dual_subp})$, e os cortes de viabilidade e otimalidade acima, o problema (\mathcal{F}_{mflow}) é equivalente ao problema (\mathcal{F}_{mestre}) abaixo, conhecido como problema mestre da decomposição de Benders. Como usual, não se conhece a priori todos os elementos de D e Q . Assim, o esquema de decomposição os enumera de modo iterativo. Parte-se de uma relaxação de (\mathcal{F}_{mestre}) considerando-se apenas as restrições (5.23) e (5.24). Se a solução relaxada for viável de (\mathcal{F}_{mflow}) adicionamos um corte de otimalidade (5.22) ao mestre relaxado. Caso contrário, adiciona-se um corte de viabilidade (5.18). Soluções viáveis de (\mathcal{F}_{mflow}) fornecem limites superiores, enquanto a solução de cada iteração do mestre relaxado fornece um limite inferior. Esse processo pára quando o valor ótimo for atingido ou algum outro critério de parada for verificado (número de iterações ou tempo de execução).

$$(\mathcal{F}_{mestre}) \quad \min \sum_{(i,j) \in A} c_{ij} w_{ij} + \eta$$

$$\text{s.a: } \sum_{i \in V_k} y_i = 1 \quad k \in M \quad (5.23)$$

$$\sum_{(i,j) \in A} w_{ij} = y_i \quad \forall i \in V \setminus \{V_1\} \quad (5.24)$$

$$\sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} y_i \hat{\mu}_i^k - \sum_{i \in V_k} y_i \hat{\mu}_i^k + \sum_{(i,j) \in A} w_{ij} \hat{\tau}_{ij}^k \right] \leq \eta, \quad \forall (\hat{\mu}, \hat{\tau}) \in D \quad (5.25)$$

$$\sum_{k \in M \setminus \{1\}} \left[\sum_{i \in V_1} y_i \hat{\mu}_i^k - \sum_{i \in V_k} y_i \hat{\mu}_i^k + \sum_{(i,j) \in A} w_{ij} \hat{\tau}_{ij}^k \right] \leq 0, \quad \forall (\hat{\mu}, \hat{\tau}) \in Q \quad (5.26)$$

$$w \in \mathbb{B}^{|A|}, y \in \mathbb{B}^{|V|} \quad (5.27)$$

5.2.1 Resultados computacionais para as principais formulações exatas

Experimentos computacionais com formulações polinomiais foram realizados em um computador equipado com processador Intel Core i7-7700 3.6 GHz, 8 núcleos, 32 GB

RAM e sistema operacional Linux Ubuntu 18.04 LTS. Os modelos exatos usam o *solver* IBM CPLEX 12.7.1, implementados com Concert Technology em C++. Para resolver as instâncias testadas, impomos às formulações matemáticas um limite de tempo de execução para o CPLEX de 10800 segundos, usando apenas uma *thread*. As abordagens de resolução implementadas cujos resultados computacionais merecem destaque são:

1. o modelo multifluxo (\mathcal{F}_{mflow}) de (MYUNG *et al.*, 1995);
2. o modelo (\mathcal{F}_{Pop}) de (POP, 2002) que utiliza a estratégia do grafo global;
3. o novo modelo multigrafo (\mathcal{F}_{MG}) com as seguintes variações:
 - a) (F_{MG})¹, onde são usadas as restrições (5.1) - (5.6), mais a restrição (5.11);
 - b) (F_{MG})², onde são usadas as restrições (5.1) - (5.3), junto com (5.5) - (5.6), mais a restrição (5.11).

Testes com a formulação (\mathcal{F}_{Pop}) usaram todos os *clusters* como referenciais, pois foi observado em nossos experimentos numéricos que adicionar iterativamente *clusters* como referenciais, um a um conforme descrito em (POP, 2002), apresenta uma performance de tempo de execução inferior do que quando se usa inicialmente todos os *clusters* como referenciais.

Em nossos experimentos computacionais usamos um conjunto de 76 instâncias, entre as quais 56 instâncias são originais de (ÖNCAN *et al.*, 2008) e as outras 20 foram criadas a partir de instâncias de grande porte desse trabalho, considerando um subconjunto de vértices das mesmas. Características das instâncias e valores de soluções viáveis obtidas com o BRKGA para serem usadas como valor de corte nos experimentos computacionais com os modelos exatos são apresentados na Tabela 5. Nessa tabela, a primeira coluna apresenta o nome da Instância (**Inst**) testada. Na segunda coluna é exibido o nome da instância original (**Orig**) de (ÖNCAN *et al.*, 2008) que gerou a instância cujo nome está na primeira coluna. Se uma linha contém o mesmo nome na primeira e segunda coluna, então essa é uma instância original de (ÖNCAN *et al.*, 2008). Cada nova instância foi criada selecionando os primeiros γ vértices da instância que a originou e essa quantidade é a parte numérica do nome da nova instância. Na terceira coluna da Tabela 5 é apresentado o parâmetro μ utilizado na instância TSPLIB para a criação de uma instância do GMSTP conforme descrito em (FISCHETTI *et al.*, 1997). Para uma mesma instância TSPLIB, valores diferentes de μ geram instâncias diferentes do GMSTP. A quarta coluna apresenta o número m de *clusters* de cada instância. A quinta coluna apresenta o número original inicial $|E|^I$ de arestas de cada instância e a sexta coluna apresenta o número final $|E|^F$ de arestas após a fixação de arestas. Vale destacar que todas as abordagens de resolução beneficiaram-se do

procedimento de eliminação de arestas e que isso tornou possível tratar instâncias de grande porte do problema. Nas duas últimas colunas são apresentados o valor de solução viável e tempo de execução em segundos processados pela metaheurística BRKGA para cada instância. Cada valor em negrito indica onde o BRKGA encontrou uma solução com valor ótimo provado para a respectiva instância. Essas soluções encontradas pelo BRKGA foram utilizadas como solução de partida para os modelos exatos testados nesta seção.

Nas Tabelas 6 e 7 são exibidos os resultados computacionais dos modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) , (\mathcal{F}_{flow}) , (\mathcal{F}_{Pop}) e da decomposição de Benders para (\mathcal{F}_{flow}) . Nas Tabelas 8 e 9 repetimos os mesmos experimentos para os modelos matemáticos considerando o valor obtido pelo BRKGA como valor de corte para o MIP do CPLEX. Na execução desses modelos foi utilizada somente uma *thread* e nenhum parâmetro foi passado para o algoritmo de resolução do CPLEX, exceto o limite de tempo de execução de 10800 segundos. Nessas tabelas, a primeira coluna informa o nome das instâncias testadas. Em cada nome, a primeira sequência de números (antes do hífen) informa o rótulo junto da quantidade de vértices da instância e a segunda parte (após o hífen) informa o parâmetro μ utilizado para gerar os *clusters* de cada instância do GMSTP, permitindo dessa forma identificar seus dados na Tabela 5. Os resultados obtidos para cada formulação são dados nas demais colunas. Para cada experimento executado com o MIP do CPLEX para os modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) , (\mathcal{F}_{flow}) e (\mathcal{F}_{Pop}) são fornecidos: o valor da solução da relaxação linear do modelo na coluna **Custo** (valores ótimos são destacados em negrito) e seu tempo de execução em segundos na coluna **Tempo**; o valor de solução do MIP do CPLEX para cada modelo na coluna **Custo** (valores ótimos são destacados em negrito) e seu tempo de execução em segundos na coluna **Tempo**; a diferença relativa em porcentagem entre o melhor limite inferior (LI) e o melhor limite superior (LS) inteiro encontrados pelo CPLEX, dado por $(1 - LI/(LS+10^{-10})) * 100$, na coluna **GAP**; e o número de nós gerados na árvore B&B do MIP do CPLEX, na coluna **Nós**.

Já para a decomposição de Benders do CPLEX executada com parâmetro *Benders-Full*, ainda nas Tabelas 6 e 7, reportamos: o valor da melhor solução viável na coluna **Custo** (valores ótimos são destacados em negrito) e seu tempo de execução em segundos na coluna **Tempo**, juntamente com a diferença relativa em porcentagem entre o valor da melhor solução viável dentre todas associadas aos cortes de otimalidade e a solução da última iteração do problema mestre na coluna **GAP** (valor esse fornecido pelo CPLEX). Nas Tabelas 6, 7, 8 e 9, a sigla OM significa estouro de memória, do inglês *out of memory*, para a instância testada.

Note que nas formulações derivadas de (\mathcal{F}_{MG}) , somente as variáveis representando os vértices são binárias. Na formulação (\mathcal{F}_{mflow}) , bem como no problema mestre da decomposição de Benders a ela associada, as variáveis representando os vértices e as variáveis representando os arcos são binárias. Na formulação (\mathcal{F}_{Pop}) , as variáveis representando as conexões do grafo global G^* são binárias. Em todos os testes com formulações usamos o processo de fixação de arestas descrito na sub-seção 3.4.2 (a fixação de vértices não foi utilizada). Sem a utilização desse mecanismo muitas das instâncias não poderiam ser resolvidas, pois esse processo reduz muito o tamanho de uma instância, onde em média são fixadas 85% das arestas.

Na Tabela 6, consideramos um conjunto de 40 instâncias de pequeno porte com até 300 vértices. Para esse conjunto, o CPLEX resolve a relaxação linear dos modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) para todas as instâncias, respectivamente, num tempo médio de 980,9 segundos, sendo que 39 são soluções inteiras; 635,8 segundos, sendo que 39 são soluções inteiras; 879,8 segundos, sendo que dessas 38 são soluções inteiras; e 2,7 segundos, sendo todas as soluções fracionárias. Ainda em relação à relaxação linear desses modelos, houve estouro de memória (OM) para uma instância na formulação (\mathcal{F}_{mflow}) . Nesses casos, os dados da instância não são usados para cálculo da média de tempo na penúltima linha da tabela. Considerando a versão MIP do CPLEX, essas instâncias foram solucionadas todas a otimalidade num tempo médio de 909,5 segundos com (\mathcal{F}_{MG}^1) e de 479,8 segundos com (\mathcal{F}_{MG}^2) . Com o modelo (\mathcal{F}_{mflow}) foi provada a otimalidade para 38 instâncias num tempo médio de 1094,4 segundos, sendo que houve problema de memória para uma instância, e com o modelo (\mathcal{F}_{Pop}) foi provada a otimalidade para 20 instâncias num tempo médio de 6384,6 segundos. Ainda na versão MIP, a média de GAP e de número de nós no *Branch-and-Bound* (B&B) para os modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) , na Tabela 6, são de 0% e 0; 0% e 0,35; 0% e 0; e 3,5% e 112393, respectivamente. A decomposição de Benders demonstrou ser a melhor abordagem de resolução para esse conjunto de instâncias resolvendo todas a otimalidade num tempo médio de 149,4 segundos. Dentre todos os modelos, (\mathcal{F}_{Pop}) consegue provar a otimalidade para 20 instâncias com a maior média de tempo, enquanto que (\mathcal{F}_{MG}^2) prova a otimalidade para 40 instâncias com segunda melhor média de tempo. Não houve estouro de memória na decomposição de Benders. Vale destacar que para a instância gr229-7, somente (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) e a decomposição de Benders resolveram essa instância. Observa-se uma diferença que foge a padrões normais entre o tempo de execução da relaxação linear e da versão MIP do CPLEX para os modelos (\mathcal{F}_{MG}^1) e (\mathcal{F}_{MG}^2) , onde a versão MIP tem média de tempo de execução menor que a da relaxação linear.

Acreditamos que isso ocorra devido ao pré-processamento de variáveis do MIP do CPLEX.

Considere agora o conjunto de 36 instâncias de grande porte com até 657 vértices apresentados na Tabela 7. Para esse conjunto, a relaxação linear dos modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) consegue encontrar 9, 9, 11 e 0 (zero) soluções inteiras, respectivamente. As soluções relaxadas para as instâncias resolvidas sem estouro de memória pelos modelos (\mathcal{F}_{MG}^1) , (\mathcal{F}_{MG}^2) e (\mathcal{F}_{mflow}) são inteiras, enquanto que todas as 36 soluções relaxadas obtidas pelo modelo (\mathcal{F}_{Pop}) são fracionárias. Essas instâncias resolvidas sem estouro de memória em suas relaxações lineares levaram, em média, 3567,3s com (\mathcal{F}_{MG}^1) , 2588,3s com (\mathcal{F}_{MG}^2) , 3025,6s com (\mathcal{F}_{mflow}) e 145,5s com (\mathcal{F}_{Pop}) . Note que há grande quantidade de estouro de memória para resolver essas instâncias na relaxação linear desses modelos, onde foram 27 para (\mathcal{F}_{MG}^1) , 27 para (\mathcal{F}_{MG}^2) , 25 para (\mathcal{F}_{mflow}) e 0 para (\mathcal{F}_{Pop}) . Considerando a versão MIP, (\mathcal{F}_{MG}^1) resolve sem estouro de memória 8 instâncias em tempo médio de 3409,1s (todas as 8 ótimas), (\mathcal{F}_{MG}^2) resolve 14 instâncias em tempo médio de 3600,1s (13 ótimas), (\mathcal{F}_{mflow}) resolve 10 instâncias em tempo médio de 3635,2s (todas as 10 ótimas) e (\mathcal{F}_{Pop}) resolve 36 instâncias em tempo médio de 8706,4s (com apenas 7 ótimas). A média de GAP para cada modelo é de 0%, 63,0%, 0%, e 14,3% seguindo a ordem acima. Quanto ao número de nós no B&B do MIP do CPLEX, apenas (\mathcal{F}_{Pop}) não é resolvido na raiz, apresentando, em média, 8697 nós para todas as instâncias. Já a decomposição de Benders demonstra ter a melhor estrutura para resolver o GMSTP considerando esse conjunto de instâncias de grande porte, conseguindo executar 27 instâncias em tempo médio de 4023,2s e provando a otimalidade para 24 delas, com uma média de GAP de 0,4%. Considerando somente os modelos sem a decomposição, (\mathcal{F}_{MG}^2) foi mais eficiente que (\mathcal{F}_{MG}^1) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) , resolvendo mais instâncias à otimalidade e em menos tempo de execução.

Nas Tabelas 8 e 9 são exibidos os resultados computacionais dos modelos (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) , (\mathcal{F}_{Pop}) e da decomposição de Benders para (\mathcal{F}_{mflow}) em sua versão MIP, quando esses recebem uma solução de partida obtida pelo BRKGA, as quais tem seu valor de solução relatados na Tabela 5. O modelo (\mathcal{F}_{MG}^1) não foi testado para esse caso porque nas Tabelas 6 e 7 o mesmo apresenta um desempenho inferior a (\mathcal{F}_{MG}^2) .

Na Tabela 8 são exibidos resultados considerando o conjunto de instâncias de pequeno porte e, na Tabela 9, as de grande porte. Os modelos (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) , (\mathcal{F}_{Pop}) e a decomposição de Benders, para cada instância, receberam a mesma solução viável de partida. Percebemos na Tabela 8, considerando o conjunto de instâncias de pequeno porte,

com a contribuição de uma solução heurística fornecida inicialmente pelo BRKGA ao MIP do CPLEX, os modelos (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) provaram a otimalidade de soluções para as mesmas instâncias de quando esses modelos não receberam uma solução inicial, ou seja, não houve variação nesse quesito. Considerando o tempo médio de execução desse grupo, houve uma redução de 479,8s para 406,0s (15,4% de redução) para (\mathcal{F}_{MG}^2) , um aumento de 1094,4s para 1295,2s (18,35% de aumento) para (\mathcal{F}_{mflow}) e uma redução de 6384,6s para 5255,4s (redução de 2%) para (\mathcal{F}_{Pop}) . Assim, o modelo que mais se beneficiou de uma solução inicial de partida no quesito tempo médio de execução das instâncias foi (\mathcal{F}_{MG}^2) , além de manter a melhor performance de tempo em relação aos outros dois modelos para essas instâncias. Considerando a média de GAP para esses modelos, na mesma ordem, tivemos 0%; 0% e 3,5%. O número médio de nós do algoritmo B&B do MIP do CPLEX para esses modelos foi de 0; 0 e 101852,1 nós, seguindo a ordem acima. Houve estouro de memória somente para o modelo (\mathcal{F}_{mflow}) , para uma instância, a qual foi a mesma instância em que ocorreu estouro de memória quando esse modelo não recebeu uma solução de partida. Considerando a decomposição de Benders, houve a prova de otimalidade de todas as soluções, assim como ocorreu quando não lhe fora passada uma solução inicial de partida. Houve pouca diminuição no tempo médio de execução das instâncias, onde houve uma redução de 149.4s para 146s (0,02% de redução). A decomposição de Benders continuou sendo a estratégia que tem a menor média de tempo para esse conjunto de instâncias. A média de GAP continuou em 0%.

Na Tabela 9, para as instâncias de grande porte, houve estouro de memória para 0, 8, 0 instâncias para os modelos (\mathcal{F}_{MG}^2) , (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) quando foi passada uma solução de partida, respectivamente, enquanto que houve estouro de memória para 27, 25 e 0 instâncias quando não foi fornecida uma solução de partida, nessa ordem. Considerando a prova de otimalidade das soluções, houve um aumento de 7 para 13 instâncias para (\mathcal{F}_{MG}^2) , o modelo (\mathcal{F}_{mflow}) manteve o número de 10 instâncias, e houve um aumento de 0 para 7 instâncias para o modelo (\mathcal{F}_{Pop}) . Seguindo essa ordem, o tempo médio de execução dos modelos foram de 7933,0s, 8247,8s e 8706,7s. Comparando esses três modelos, (\mathcal{F}_{MG}^2) continua com melhor performance computacional assim como relatado na Tabela 7. Para esse conjunto de instância, conforme a ordem acima, a média de GAP para esses modelos foi de 62,2%; 61,3% e 10,4%, respectivamente. O número médio de nós no algoritmo B&B do MIP do CPLEX para esses modelos foi de 0, 0 e 8956, respectivamente. Vale ressaltar que nenhum dos modelos testados evoluiu a solução viável fornecida pelo BRKGA a ponto de encontrar outra solução de menor

custo. De certa forma, isso pode indicar que o BRKGA fornece uma solução viável de boa qualidade ou até mesmo ótima, mas que os modelos não conseguem provar sua otimalidade ou melhorá-la devido a limitação de memória da máquina ou limite de tempo imposto, o qual é bem expressivo. Foi o caso das instâncias gr431-5 e gr431-7, onde mesmo os modelos recebendo uma solução ótima (com otimalidade provada pela decomposição de Benders), nenhum deles conseguiu provar sua otimalidade. Considerando a decomposição de Benders quando recebe uma solução de partida, para o conjunto de instâncias de grande porte, houve uma diminuição de 9 para 5 instâncias em que aconteceu estouro de memória, foi repetida as mesmas 24 instâncias com prova de otimalidade e, devido um número maior de instâncias executadas, houve aumento no tempo médio de execução de 4023,2s para 4634,8s. Vale ressaltar que mesmo fornecendo uma solução de partida, nenhum modelo obteve melhor tempo de execução que a decomposição de Benders.

Tabela 5 – Características das 76 instâncias usadas nos experimentos computacionais e valores de solução viável obtidos com o BRKGA

Inst	Orig	u	m	$ E ^l$	$ E ^r$	BRKGA	Tempo
gr159	gr229	3	48	12141	1523	36335	22.1
gil140	gil262	3	78	9642	448	1220	24.0
pr164	pr264	3	73	13175	733	22692	22.9
pr199	pr299	3	72	19418	1153	17241	34.3
lin218	lin318	3	76	23239	1457	17739	26.7
gr159	gr229	5	28	11836	2906	28081	11.4
gil140	gil262	5	59	9589	641	1059	12.2
pr164	pr264	5	41	12970	1343	18045	11.9
pr199	pr299	5	49	19271	2071	14431	22.8
lin218	lin318	5	50	22919	1886	14407	23.0
gr159	gr229	7	21	11445	3633	24198	11.2
gil140	gil262	7	45	9529	823	872	11.9
pr164	pr264	7	35	12782	1511	17109	11.5
pr199	pr299	7	37	19014	2111	12585	9.0
lin218	lin318	7	36	22838	2920	11870	17.5
gr159	gr229	10	14	10825	4908	18813	9.1
gil140	gil262	10	35	9455	1243	726	10.5
pr164	pr264	10	21	12143	2153	15889	10.9
pr199	pr299	10	27	18780	3075	9826	9.5
lin218	lin318	10	26	22543	4982	8223	13.0
gr229	gr229	3	81	25584	2640	74819	55.4
gil262	gil262	3	95	33845	1472	1255	69.2
pr264	pr264	3	101	34357	1378	29199	27.4
pr299	pr299	3	102	44114	1927	23096	19.9
lin300	lin318	3	103	44304	2132	23083	20.5
gr229	gr229	5	47	25129	4340	54236	23.2
gil262	gil262	5	63	33643	2986	984	28.5
pr264	pr264	5	55	34016	2593	21351	29.3
pr299	pr299	5	69	43893	3542	18582	26.8
lin300	lin318	5	62	43839	3086	17416	30.5
gr229	gr229	7	34	24595	5752	46030	52.6
gil262	gil262	7	49	33487	3413	802	39.0
pr264	pr264	7	43	33481	3574	20438	17.8
pr299	pr299	7	47	43463	4758	15238	20.6
lin300	lin318	7	46	43651	4458	14234	24.6
gr229	gr229	10	23	23761	9567	39797	12.3
gil262	gil262	10	36	33208	5533	639	23.1
pr264	pr264	10	27	32814	4289	16546	17.3
pr299	pr299	10	35	43054	6361	11624	23.4
lin300	lin318	10	36	43221	6902	10119	27.5
ali535	ali535	3	181	140717	9916	134306	536.9
att532	att532	3	182	140049	5463	15652	236.7
d493	d493	3	171	119945	7605	20269	133.2
d657	d657	3	221	214467	5912	25704	677.3
fl417	fl417	3	142	85821	3453	8353	117.8
gr431	gr431	3	149	90783	10993	103805	422.1
lin318	lin318	3	108	49821	2257	24092	110.8
p654	p654	3	230	211794	6258	23548	255.3
pcb442	pcb442	3	156	96904	2616	27513	130.9
ali535	ali535	5	108	137198	16993	109995	525.4
att532	att532	5	110	138952	11289	12369	418.0
d493	d493	5	102	118143	15922	16496	306.1
d657	d657	5	137	213715	11339	19836	438.7
fl417	fl417	5	93	84944	4900	7952	99.3
gr431	gr431	5	87	88598	18591	81963	119.9
lin318	lin318	5	64	49320	3385	17667	96.1
p654	p654	5	134	209256	15180	22377	124.1
pcb442	pcb442	5	95	96383	5321	19390	415.1
ali535	ali535	7	83	137048	22525	95308	489.8
att532	att532	7	80	137788	16093	10459	479.0
d493	d493	7	78	116770	21828	14381	374.2
d657	d657	7	96	212843	17168	17879	664.3
fl417	fl417	7	61	83476	8872	7446	245.3
gr431	gr431	7	64	86817	22485	71378	139.2
lin318	lin318	7	49	49106	4808	14909	104.9
p654	p654	7	100	207125	19048	21789	432.0
pcb442	pcb442	7	64	95900	8693	15896	349.9
ali535	ali535	10	57	129989	20532	74401	256.3
att532	att532	10	57	136796	23567	8659	253.3
d493	d493	10	52	114177	30338	11336	148.2
d657	d657	10	73	212014	23612	14935	160.8
fl417	fl417	10	42	82007	9745	6986	123.8
gr431	gr431	10	52	84391	23342	62688	168.3
lin318	lin318	10	36	48664	7723	10119	152.9
p654	p654	10	69	203619	18791	20749	277.0
pcb442	pcb442	10	48	95084	9571	12222	133.8

Fonte: autor.

Tabela 7 – Resultados computacionais para os modelos exatos considerando 36 instâncias de grande porte com mais de 300 e até 657 vértices

Instância	Relax. Linear			MIP			Relax. Linear			MIP			Relax. Linear			MIP			Relax. Linear			MIP			Relax. Linear			MIP		
	Valor	Tempo	Nes	Valor	Tempo	GAP(%)	Valor	Tempo	Nes	Valor	Tempo	GAP(%)	Valor	Tempo	Nes	Valor	Tempo	Nes	Valor	Tempo	Nes	Valor	Tempo	Nes	Valor	Tempo	Nes	Valor	Tempo	Nes
id1-v-t	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
alt32-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d493-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
f417-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
gr431-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
lin318-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
p554-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
pcb442-3	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
alt535-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d493-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d657-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
gr431-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
lin318-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
p554-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
pcb442-5	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
alt535-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d493-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d657-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
gr431-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
lin318-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
p554-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
pcb442-7	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
alt535-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d493-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
d657-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
gr431-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
lin318-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
p554-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
pcb442-10	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
Qtd de ótimos	9	3567.3	8	3409.1	8	3600.1	13	2588.3	9	2588.3	9	3600.1	63.0	0	0	11	3025.6	10	10	10	10	10	10	10	10	10	10	10	10	10
Valores médios	9	3567.3	8	3409.1	8	3600.1	13	2588.3	9	2588.3	9	3600.1	63.0	0	0	11	3025.6	10	10	10	10	10	10	10	10	10	10	10	10	10
Qtd de ótimos	9	3567.3	8	3409.1	8	3600.1	13	2588.3	9	2588.3	9	3600.1	63.0	0	0	11	3025.6	10	10	10	10	10	10	10	10	10	10	10	10	10
Qtd de ótimos	9	3567.3	8	3409.1	8	3600.1	13	2588.3	9	2588.3	9	3600.1	63.0	0	0	11	3025.6	10	10	10	10	10	10	10	10	10	10	10	10	10

OM - instância onde houve estouro de memória para o referido modelo
 * - instância onde o referido modelo não conseguiu encontrar uma solução viável dentro do limite de tempo
 Fonte: autor.

6 CONSIDERAÇÕES FINAIS

Neste trabalho desenvolvemos algoritmos heurísticos, exatos e de decomposição para o GMSTP. Esse problema consiste em encontrar um conjunto de arestas que expande todos os m clusters de vértices V_1, V_2, \dots, V_m de um grafo $G = (V_1 \cup V_2 \cup \dots \cup V_m, E)$, incidido sobre exatamente um mesmo vértice de cada cluster.

Apresentamos uma revisão da literatura sobre a análise da complexidade do problema e destacamos alguns casos em que uma solução ótima pode ser encontrada em tempo polinomial. Desenvolvemos uma metaheurística BRKGA para o GMSTP que se mostrou mais eficiente que as melhores metaheurísticas da literatura para o problema, em termos de qualidade de solução, conseguindo superar os melhores resultados da literatura para 10 instâncias. Esses resultados foram apresentados no XLIX Simpósio Brasileiro de Pesquisa Operacional (SOBRAPO) 2017 (SOUSA *et al.*, 2017).

Também conduzimos um estudo sobre modelos matemáticos exatos para o GMSTP. Os principais modelos exatos da literatura foram implementados para que pudéssemos compará-los com o novo modelo (\mathcal{F}_{MG}) proposto, baseado em uma abordagem inédita de modelagem para multigrafo. As principais formulações da literatura são o modelo multifluxo (\mathcal{F}_{mflow}) de (MYUNG *et al.*, 1995) e o modelo (\mathcal{F}_{Pop}) de (POP, 2002) que utiliza a estratégia do grafo global. Uma comparação entre os resultados computacionais preliminares para os modelos (\mathcal{F}_{MG}), (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) foi apresentada no *International Symposium on Combinatorial Optimization* (ISCO) 2018 (SOUSA *et al.*, 2018). Apresentamos também propriedades estruturais para o modelo (\mathcal{F}_{MG}), onde conseguimos fortalecê-lo considerando restrições de igualdade que dominam algumas outras do modelo original, o que nos possibilitou duas de suas versões, (\mathcal{F}_{MG}^1) e (\mathcal{F}_{MG}^2). Em geral, a versão (\mathcal{F}_{MG}^2) apresentou tempo médio de processamento menor do que para (\mathcal{F}_{MG}^1). Em relação ao conjunto de instâncias de pequeno porte, a relaxação linear de (\mathcal{F}_{MG}^2) encontrou mais soluções ótimas inteiras e em tempo médio de processamento menor do que para (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}). Na versão MIP dos modelos implementados, para o conjunto de instâncias de pequeno porte, (\mathcal{F}_{MG}^2) encontrou mais soluções ótimas e em tempo médio de processamento menor do que para (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}). Considerando o conjunto de instâncias de grande porte, (\mathcal{F}_{MG}^2) encontrou 3 soluções ótimas a mais do que (\mathcal{F}_{mflow}) e 6 a mais do que (\mathcal{F}_{Pop}), tendo tomado menor tempo médio de execução que (\mathcal{F}_{mflow}) e (\mathcal{F}_{Pop}) com o uso da solução heurística.

Também investigamos uma eficiente estratégia para solucionar o GMSTP, a qual

superou todos os resultados dos modelos exatos da literatura e do novo modelo (\mathcal{F}_{MG}^2). Essa estratégia é o algoritmo de decomposição de Benders aplicado ao modelo (\mathcal{F}_{mflow}), cuja implementação é direta no CPLEX. Tanto para instâncias de pequeno quanto de grande porte, passando ou não solução inicial, a decomposição de Benders obteve mais soluções ótimas e em bem menos tempo de execução que as demais abordagens de resolução exata.

Por fim, concluímos que os procedimentos aqui apresentados contribuíram significativamente para resolver o problema em questão, apresentando melhor performance computacional que o estado da arte para o GMSTP.

Como trabalhos futuros, o novo conjunto de restrições com orientação de uma única entrada em cada vértice de cada *cluster*, o qual fortaleceu o modelo (\mathcal{F}_{MG}), pode ser explorado como plano de cortes. Outra opção interessante é verificar uma estratégia de definição da árvore global na formulação (\mathcal{F}_{Pop}) diferente da formulação (\mathcal{F}_{Martin}). Pretende-se testar uma nova fixação de arestas ou vértices baseada em custos reduzidos com base na formulação empregada (sendo que ambas podem ser realizadas de forma iterativa), bem como testar as abordagens de resolução do modelo multigrafo e a decomposição de Benders com uma formulação multifluxo em problemas relacionados, como é o caso do L-GMSTP (HAOUARI; CHAOUACHI, 2006). Outra alternativa é desenvolver um algoritmo de Relaxação Lagrangeana baseado nas formulações multifluxo e multigrafo. Além disso, planeja-se explorar restrições de arestas conflitantes para a resolução do GMSTP fazendo uso de um grafo de conflitos.

REFERÊNCIAS

- ADASME, P.; ANDRADE, R.; LETOURNEL, M.; LISSER, A. Stochastic maximum weight forest problem. **Networks**, v. 65, p. 289–305, 2015.
- ANDRADE, R. C. Appointments on the spanning tree polytope. **Proceedings of the ALIO/EURO - Montevideo**, Uruguay, p. 1–3, 2014.
- ANDRADE, R. C.; LUCENA, A. P.; MACULAN, N. Using lagrangean dual information to generate degree constrained minimum spanning trees. **Discrete Applied Mathematics**, v. 154, p. 703–717, 2006.
- CAYLEY, A. On the mathematical theory of isomers. **Philosophical Magazine**, v. 67, p. 444–446, 1874.
- CONTRERAS-BOLTON, C.; GATICA, G.; BARRA, C. R.; PARADA, V. A multi-operator genetic algorithm for the generalized minimum spanning tree problem. **Expert Systems With Applications**, v. 50, p. 1–8, 2016.
- DIAS, F. C. S.; CAMPÊLO, M. Formulação MTZ para o problema da árvore geradora mínima sob restrições de conflito. **XLVIII Simpósio Brasileiro de Pesquisa Operacional**, Brasil, Anais do XLVIII SBPO, p. 2441–2448, 2016.
- FEO, T. A.; RESENDE, M. G. C. A probabilistic heuristic for a computationally difficult set covering problem. **Operations Research Letters**, v. 8, p. 67–71, 1989.
- FEREMANS, C.; LABBE, M.; LAPORTE, G. On generalized minimum spanning trees. **European Journal of Operational Research**, v. 134(2), p. 457–458, 2001.
- FEREMANS, C.; LABBE, M.; LAPORTE, G. A comparative analysis of several formulations for the generalized minimum spanning tree problem. **Networks**, v. 39(1), p. 29–34, 2002.
- FEREMANS, C.; LABBE, M.; LAPORTE, G. The generalized minimum spanning tree problem: Polyhedral analysis and branch-and-cut algorithm. **Networks**, v. 43(2), p. 71–86, 2004.
- FERREIRA, C. S.; OCHI, L. S.; PARADA, V.; UCHOA, E. A GRASP-based approach to the generalized minimum spanning tree problem. **Expert Systems with Applications**, v. 39(3), p. 3526–3536, 2012.
- FISCHETTI, M.; SALAZAR, J. J.; TOTH, P. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. **Operations Research**, v. 45, p. 378–394, 1997.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-completeness**. [S. l.]: W. H. Freeman, 1979.
- GHOSH, D. Solving medium to large sized Euclidean generalized minimum spanning tree problems. **Technical report NEP-CMP-2003-09-28 Indian Institute of Management**, 2003.
- GLOVER, F. Tabu search: Part i. **ORSA Journal on Computing**, v. 1(3), p. 190–206, 1989.
- GOLDEN, B.; RAGHAVAN, S.; STANOJEVIC, D. Heuristic search for the generalized minimum spanning tree problem. **INFORMS Journal on Computing**, v. 17(3), p. 290–304, 2005.

- GONCALVES, J. F.; RESENDE, M. G. C. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, p. 487 – 525, 2011.
- HAOUARI, M.; CHAOUACHI, J. S. Upper and lower bounding strategies for the generalized minimum spanning tree problem. **European Journal of Operational Research**, v. 172, p. 632–647, 2006.
- HOLLAND, J. H. Outline for a logical theory of adaptive systems. **J. Association Comput**, v. 3, p. 297–314, 1962.
- HU, B.; LEITNER, M.; RAIDL, G. R. Computing generalized minimum spanning trees with variable neighborhood search. **Proceedings of the 18th Mini-Euro Conference on Variable Neighborhood Search**, Austria, 2005.
- KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. **Proceedings of the American Mathematical Society**, v. 7(1), p. 48–50, 1956.
- MARTIN, K. Using separation algorithms to generate mixed integer model reformulations. **Operations Research Letters**, v. 10(3), p. 119–128, 1991.
- MATOUSEK, J.; NESETRIL, J. **Invitation to discrete mathematics**. [S. l.]: Oxford University Press, 2008.
- MLADENOVIC, M.; HANSEN, P. Variable neighborhood search. **Computers and Operations Research**, v. 24, p. 1097–1100, 1997.
- MYUNG, Y. S.; LEE, C. H.; TCHA, D. W. On the generalized minimum spanning tree problem. **Networks**, v. 26(4), p. 231–241, 1995.
- POP, P. C. **The generalized minimum spanning tree problem**. 147 f. Tese (Doutorado) — University of Twente, The Netherlands, 2002.
- POP, P. C. The generalized minimum spanning tree problem: an overview of formulations, solution procedures and latest advances. **European Journal of Operational Research**, v. 170, p. 1–10, 2019.
- POP, P. C.; MATEI, O.; SABO, C.; PETROVAN, A. A two-level solution approach for solving the generalized minimum spanning tree problem. **European Journal of Operational Research**, v. 265, p. 478–487, 2017.
- PRIM, R. C. Shortest connection networks and some generalizations. **Bell System Technical Journal**, v. 36(6), p. 1389–1401, 1957.
- REINELT, G. TSPLIB – A traveling salesman problem library. **ORSA Journal on Computing**, v. 3, p. 267–384, 1991.
- SANTOS, A. C. **Modelos e algoritmos para o problema da árvore geradora de custo mínimo com restrição de diâmetro**. 133 f. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, Brasil, 2006.
- SANTOS, A. C.; DUHAMEL, C.; ANDRADE, R. Trees and forests. **Springer International Publishing**, p. 1–27, 2016.

SOUSA, E. G. d.; ANDRADE, R. C. d.; SANTOS, A. C. Algoritmo genético para o problema da árvore geradora generalizada de custo mínimo. **XLIX Simpósio Brasileiro de Pesquisa Operacional**, Brasil, v. 4099, p. 918–922, 2017.

SOUSA, E. G. d.; ANDRADE, R. C. d.; SANTOS, A. C. A multigraph formulation for the generalized minimum spanning tree problem. **Proceedings of the 5th International Symposium on Combinatorial Optimization**, Morocco, v. 10856, p. 133–143, 2018.

WANG, Z.; CHE, C. H.; LIM, A. Tabu search for generalized minimum spanning tree problem. **PRICAI 2006, LNAI**, v. 4099, p. 918–922, 2006.

ÖNCAN, T.; CORDEAU, J.; LAPORTE, G. A tabu search heuristic for the generalized minimum spanning tree problem. **European Journal of Operational Research**, v. 191(2), p. 306–319, 2008.