



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

FELIPE ALBUQUERQUE BRITO DA SILVA

PROBLEMA DE FORMAÇÃO DE EQUIPES COM CAMINHOS POSITIVOS

FORTALEZA

2023

FELIPE ALBUQUERQUE BRITO DA SILVA

PROBLEMA DE FORMAÇÃO DE EQUIPES COM CAMINHOS POSITIVOS

Dissertação apresentada ao Curso de Mestrado em Ciência da Computação do 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 mestre em Ciência da Computação. Área de Concentração: Otimização Combinatória

Orientador: Prof. Dr. Manoel Bezerra Campêlo Neto

Coorientadora: Prof. Dra. Tatiane Fernandes Figueiredo

FORTALEZA

2023

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

- S58p Silva, Felipe Albuquerque Brito da.
Problema de formação de equipes com caminhos positivos / Felipe Albuquerque Brito da Silva. – 2023.
70 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2023.
Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.
Coorientação: Prof. Dr. Tatiane Fernandes Figueiredo.
1. Otimização combinatória. 2. Programação linear inteira. 3. Formação de equipes. 4. Grafo de sinais. 5. Caminho positivo mínimo. I. Título.

CDD 005

FELIPE ALBUQUERQUE BRITO DA SILVA

PROBLEMA DE FORMAÇÃO DE EQUIPES COM CAMINHOS POSITIVOS

Dissertação apresentada ao Curso de Mestrado em Ciência da Computação do 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 mestre em Ciência da Computação. Área de Concentração: Otimização Combinatória

Aprovada em: 23/10/2023

BANCA EXAMINADORA

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

Prof. Dra. Tatiane Fernandes Figueiredo (Coorientadora)
Universidade Federal do Ceará (UFC) - Campus Russas

Prof. Dr. Júlio César Silva Araújo
Universidade Federal do Ceará (UFC)

Prof. Dra. Rosa Maria Videira de Figueiredo
Avignon Université (AU)

À minha mãe Mônica e meus irmãos Robson e Lucas, pelo suporte, incentivo e apoio incondicional. Por estarem sempre presentes e pelo exemplo de coragem e determinação.

AGRADECIMENTOS

Primeiramente à minha família, minha mãe Mônica e meus irmãos Robson e Lucas, por todo apoio emocional, por sempre me motivarem na minha vida acadêmica e para ser uma pessoa melhor. À minha namorada Roberta, por todo o amor e paciência, seu apoio foi fundamental ao longo dessa jornada.

Ao meu orientador professor Manoel Campêlo, por todos os ensinamentos, orientações e experiências engrandecedoras na minha formação profissional e pessoal, além de ter acreditado em mim e me incentivado desde a minha graduação. Com certeza é uma importante referência de ser humano que irei carregar para minha vida.

À minha coorientadora professora Tatiane Figueiredo, agradeço sua orientação, incentivo e amizade que foram extremamente importantes para esse trabalho. Sou grato pelas inúmeras oportunidades de ampliar meus horizontes e por compartilhar experiências acadêmicas, profissionais e de vida durante o meu mestrado.

À professora Rosa Figueiredo, que me proporcionou uma experiência incrível de fazer um estágio de mestrado na França. Por acreditar em mim e me proporcionar, ainda mais, a felicidade de fazer meu doutorado com a orientação dela.

Aos professores Manoel Campêlo, Tatiane Figueiredo e Rosa Figueiredo, em conjunto, pois me considero sortudo pela oportunidade de estar perto dessas pessoas fantásticas que eu tanto admiro e que me fazem sentir que eu estou na direção certa.

Ao professor Julio Araújo, por aceitar o convite para participar da banca avaliadora deste trabalho.

À todos meus amigos, que me apoiaram e ajudaram bastante durante vários períodos do meu mestrado. Em especial, aos membros do grupo de pesquisa ParGO, que contribuíram substancialmente na minha formação e na construção desse trabalho.

À Universidade Federal do Ceará, em destaque aos professores, servidores e alunos do Departamento de Estatística e Matemática Aplicada e Departamento de Computação, por fomentarem um ambiente valoroso para a aprendizagem e a realização do meu mestrado.

Ao CNPq, pelo financiamento da bolsa de mestrado, durante dois anos do mestrado.

“L’intelligence ne vaut qu’au service de l’amour.”
(Antoine de Saint-Exupéry)

RESUMO

De maneira frequente informações obtidas de redes sociais têm sido utilizadas para resolver aplicações em Pesquisa Operacional, como o Problema de Formação de Equipes, cujo objetivo é encontrar um grupo de indivíduos que cobrem coletivamente um conjunto de habilidades e que podem trabalhar colaborativamente de forma eficaz. Na literatura, diferentes abordagens têm sido usadas para avaliar a sinergia entre membros de uma equipe, dando origem a várias versões desse problema. Neste trabalho, utilizamos a Teoria do Equilíbrio Estrutural para definir compatibilidade entre pares de indivíduos de uma rede social. Para tanto, as redes sociais são representadas por grafos de sinais, e a métrica de compatibilidade é definida a partir da análise de caminhos positivos (com número par de arestas negativas) entre pares de vértices distintos. Com isso, introduzimos e estudamos uma nova versão do problema, a ser chamada Problema de Formação de Equipes com Caminhos Positivos. Como trabalho preliminar, revisitamos o Problema de Caminho Positivo Mínimo em grafos direcionados e não direcionados de sinais, compilando e reescrevendo algoritmos, resultados de complexidade e formulações matemáticas. Para o problema principal, apresentamos duas formulações de Programação Linear Inteira e uma estratégia de decomposição das formulações, que resultou em ganho expressivo de desempenho computacional. Além disso, derivamos desigualdade válidas, algumas das quais contribuíram adicionalmente para a melhoria do desempenho do modelo decomposto. Os métodos, implementados utilizando a linguagem C++ e o solver *CPLEX*, foram aplicados em um conjunto de instâncias baseadas em redes sociais reais ou sintéticas. Uma análise dos experimentos computacionais realizados comprova a potencial eficiência da estratégia de decomposição e de desigualdades válidas propostas.

Palavras-chave: otimização combinatória; programação linear inteira; formação de equipes; grafo de sinais.

ABSTRACT

Information obtained from social networks has frequently been used to solve applications in Operations Research, such as the Team Formation Problem, whose goal is to find a group of the workers that collectively cover a set of skills and can work collaboratively and effectively. In the literature, different approaches have been used to assess the synergy between team members, giving rise to various versions of this problem. In this work, we used the Structural Balance Theory to define compatibility between pairs of individuals in a social network. Therefore, the social networks are represented by signed graphs, and the compatibility metric is calculated from the analysis of positive paths (with an even number of negative edges) between pairs of distinct vertices. With this, we introduced and studied a new version of the problem, to be called the Team Formation Problem with Positive Paths. As preliminary work, we revisited the Shortest Positive Path Problem in directed and undirected signed graphs, compiling and rewriting algorithms, complexity results and mathematical formulations. For the main problem, we presented two Integer Linear Programming formulations and a strategy for decomposing the formulations, which resulted in a significant gain in computational performance. In addition, we derived valid inequalities, some of which also contributed to improving the performance of the decomposed model. The methods, implemented using C++ language and *CPLEX* solver, were applied in a set of instances based on real or synthetic social networks. An analysis of the computational experiments carried out proves the potential efficiency of the proposed decomposition strategy and valid inequalities.

Keywords: combinatorial optimization; integer linear programming; team formation; signed graphs.

LISTA DE FIGURAS

Figura 1 – Exemplo de um grafo e um digrafo	23
Figura 2 – Exemplo do grafo par G_p de um grafo de sinais G	30
Figura 3 – Ilustração do grafo auxiliar H'	31
Figura 4 – Ilustração da redução de NP-Compleitude	33
Figura 5 – Exemplo da necessidade da restrição de quebra de ciclo	42
Figura 6 – Exemplo de obtenção do grafo de compatibilidade G' a partir do grafo de sinais G	44
Figura 7 – Médias dos tempos por quantidade de vértices para cada método do SPP . .	55
Figura 8 – Porcentagem de instâncias, por número de vértices, com soluções viáveis e ótimas encontradas por cada modelo para grafos não direcionados	60
Figura 9 – Porcentagem de instâncias, por número de vértices, com soluções viáveis e ótimas encontradas por cada modelo para grafos direcionados	63
Figura 10 – Médias dos tempos de computação, para cada quantidade de vértices, de cada subproblema da decomposição	66

LISTA DE TABELAS

Tabela 1 – Porcentagem de arestas positivas e negativas nos grafos não direcionados . .	52
Tabela 2 – Porcentagem de arcos positivos e negativos nos grafos direcionados	53
Tabela 3 – Teste de Viabilidade de Instâncias com Grafos Não Direcionados	54
Tabela 4 – Teste de Viabilidade de Instâncias com Grafos Direcionados	54
Tabela 5 – Tempos de computação dos métodos para o SPP em grafos não direcionados	56
Tabela 6 – Tempos de computação dos métodos para o SPP em grafos direcionados . .	56
Tabela 7 – Número de restrições no Modelo-$\mathcal{F}_{MTZ}$ e no Modelo-$\mathcal{F}_{SIGN}$	57
Tabela 8 – Número de restrições nas três versões do Modelo-$T\mathcal{F}$	58
Tabela 9 – Desempenho dos modelos para o TFP_SPP em grafos não direcionados de 25 vértices	59
Tabela 10 – Desempenho dos modelos para o TFP_SPP em grafos não direcionados de 50 vértices	61
Tabela 11 – Desempenho dos modelos para o TFP_SPP com grafos não direcionados de 100 vértices	62
Tabela 12 – Desempenho dos modelos para o TFP_SPP em digrafos com 25 e 50 vértices	64
Tabela 13 – Desempenho dos modelos para o TFP_SPP em digrafos com 100 vértices .	65

LISTA DE ABREVIATURAS E SIGLAS

TFP	Team Formation Problem
MTFP	Multiple Team Formation Problem
STFP	Socio-Technical Team Formation Problem
CPM	Caminho Positivo Mínimo
PLI	Programação Linear-Inteira
PL	Programação Linear
PIM	Programação Inteira Mista
PI	Programação Inteira
SPP	Shortest Positive Path
TFP_SPP	Team Formation Problem with Shortest Positive Paths
MWPMP	Minimum Weight Perfect Matching Problem

LISTA DE SÍMBOLOS

A	Conjunto de arcos
A^+	Conjunto de arcos com sinal positivo
A^-	Conjunto de arcos com sinal negativo
E	Conjunto de arestas
E_j	Pares de vértices em E onde ambos possuem alguma habilidade requerida na equipe $j \in T$
$\bar{E}$	Conjunto de arestas complementar ao conjunto E
$\bar{E}_j$	Pares de vértices em $\bar{E}$ onde ambos possuem alguma habilidade requerida na equipe $j \in T$
E^+	Conjunto de arestas com sinal positivo
E^-	Conjunto de arestas com sinal negativo
E'	Conjunto de arestas do grafo de compatibilidade G'
E'_j	Pares de vértices em E' onde ambos possuem alguma habilidade requerida na equipe $j \in T$
G	Grafo de sinais
G'	Grafo de compatibilidade, com arestas ponderadas pelos caminhos positivos mínimos
P_{uv}	Caminho entre os vértices u e v
$\mathcal{P}_{uv}^+$	Conjunto dos caminhos positivos entre os vértices u e v
$\mathcal{P}_{uv}^-$	Conjunto dos caminhos negativos entre os vértices u e v
S	Conjunto de habilidades
$\bar{s}$	Função que retorna as habilidades de cada indivíduo ou habilidades necessárias na equipe
T	Conjunto de equipes
t	Função que retorna a demanda de trabalhadores para cada tipo de habilidade em cada equipe
$V(G)$	Conjunto de vértices do grafo G

V_j	Conjunto de indivíduos em V que possuem alguma habilidade requerida na equipe $j \in T$
V_T	Conjunto de indivíduos em V que possuem alguma habilidade requerida em alguma equipe
$\mathcal{F}_{MTZ}$	Formulação para resolver o TFP_SPP utilizando as restrições MTZ
$\mathcal{F}_{SIGN}$	Formulação para resolver o TFP_SPP utilizando as restrições de sinais
$\mathcal{SP}_{MTZ}$	Formulação para resolver o SPP utilizando as restrições MTZ
$\mathcal{SP}_{SIGN}$	Formulação para resolver o SPP utilizando as restrições de sinais
$T\mathcal{F}$	Formulação simplificada para resolver o TFP_SPP
$\delta^+(u, v)$	Menor peso de um u, v -caminho positivo
$\delta^-(u, v)$	Menor peso de um u, v -caminho negativo

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Formação de Equipes	16
1.2	Revisão Bibliográfica	17
1.3	Problema Estudado e Organização do Texto	19
2	FUNDAMENTOS	21
2.1	Programação Matemática	21
2.2	Teoria dos Grafos	22
2.3	Algoritmos e Classes de Problemas	25
3	CAMINHO POSITIVO MÍNIMO	26
3.1	Definição do Problema	26
3.2	Grafos de Sinais Balanceados	27
3.3	Grafos de Sinais Não Direcionados	29
3.4	Digrafos de Sinais	32
3.5	Formulações de Programação Inteira	33
4	FORMAÇÃO DE EQUIPES COM CAMINHOS POSITIVOS MÍNIMOS	37
4.1	Definição do Problema	37
4.2	Métodos para o TFP_SPP Não Direcionado	39
4.2.1	<i>Formulação Matemática</i>	40
4.2.2	<i>Decompondo a Formulação</i>	43
4.2.3	<i>Cálculo dos Caminhos Positivos Mínimos</i>	44
4.2.4	<i>Modelo Simplificado</i>	44
4.3	Métodos para o TFP_SPP Direcionado	45
4.3.1	<i>Cálculo dos Caminhos Direcionados Positivos Mínimos</i>	46
4.3.2	<i>Modelo Simplificado</i>	46
4.4	Desigualdades Válidas para o Modelo-TF	47
5	EXPERIMENTOS E RESULTADOS COMPUTACIONAIS	51
5.1	Ambiente Computacional	51
5.2	Instâncias	51
5.3	Resultados Computacionais para o SPP	54
5.4	Resultados Computacionais para o TFP_SPP Não Direcionado	57

5.4.1	<i>Dimensões dos Modelos</i>	57
5.4.2	<i>Comparação do Desempenho dos Modelos</i>	58
5.5	Resultados Computacionais para o TFP_SPP Direcionado	63
6	CONCLUSÕES E TRABALHOS FUTUROS	67
	REFERÊNCIAS	69

1 INTRODUÇÃO

Um dos grandes desafios para empresas e organizações é a formação de equipes eficientes na realização de determinados projetos. Em geral, consideram-se indivíduos que possuam um conjunto de habilidades (expertises ou perfis) e deseja-se criar, com esses indivíduos, equipes que satisfaçam demandas de habilidades requeridas. Podemos tratar esse problema de formar equipes com uma abordagem matemática. Além disso, é importante incorporarmos, nessa perspectiva matemática, conceitos de relações interpessoais presentes na área das ciências humanas para buscar que indivíduos na mesma equipe se relacionem positivamente e consigam bons resultados.

1.1 Formação de Equipes

No ramo das ciências humanas, dado um projeto que necessita de uma equipe de trabalho cooperativo, é convergente a ideia de que a escolha dos indivíduos para comporem a equipe está diretamente ligada com os resultados que serão obtidos. Assim, podemos afirmar que escolha certa de equipes é essencial para o sucesso das empresas (COHEN; BAILEY, 1997; BARRICK *et al.*, 1998). Uma das estratégias para conseguir se sobressair em um mercado competitivo e tecnológico é criar equipes que tenham como foco a motivação dos funcionários e metas de produtividade coletivas (NEUMAN *et al.*, 1999; BASFORD; OFFERMANN, 2012). A comunicação eficaz e o trabalho em equipe são essenciais para entregar alta qualidade de serviço (LEONARD *et al.*, 2004). A atuação coletiva é amplamente reconhecida como uma força positiva para o trabalho em equipe em qualquer organização ou instituição a ser bem sucedida (AGARWAL; ADJIRACKOR, 2016). Além disso, a importância de formarmos boas equipes cresce junto com o aumento da necessidade de criarmos serviços e produtos cada vez mais complexos (GUMMER, 1996).

As pessoas passam aproximadamente um terço de suas vidas no trabalho, e suas relações sociais afetam significativamente a sua saúde física e mental (UMBERSON; MONTEZ, 2010). Assim, torna-se evidente o benefício individual quando formamos equipes com bom relacionamento. García-Campayo *et al.* (2016) afirmam que relações positivas em equipes colaboram para diminuir o estresse e combater outros problemas psicológicos. Em equipes com boas relações, é perceptível um maior compartilhamento de conhecimento e produtividade entre os membros, levando a um ambiente de maior crescimento profissional (CORNELISSEN, 2016).

1.2 Revisão Bibliográfica

Nos últimos anos, podemos perceber um aumento do uso de métodos matemáticos e computacionais para encarar o desafio de formação de equipes. Em (LAPPAS *et al.*, 2009), o Problema de Formação de Equipe (do inglês, Team Formation Problem (TFP)) foi formalmente definido considerando a visão de que o sucesso da equipe depende não apenas das habilidades dos indivíduos, mas também da eficiência com que eles colaboram, se comunicam e trabalham juntos como um time. Além disso, considerando a disseminação das redes sociais, os autores propõem a utilização dessas redes como uma forma mais prática para mensurar as relações entre os indivíduos.

De forma geral, Lappas *et al.* (2009) definem o TFP como o problema de encontrar, a partir de uma rede social, um subconjunto de indivíduos que possuam as habilidades necessárias para realizar uma tarefa e que consigam se comunicar efetivamente entre si. No trabalho, a “qualidade” da comunicação é definida em função da distância entre os membros da equipe no grafo representativo da rede social. Várias métricas são consideradas, entre elas o peso da árvore geradora mínima entre os membros da equipe. As diferentes versões do TFP estudadas em (LAPPAS *et al.*, 2009) são mostradas constituírem problemas NP-Difíceis, para os quais são apresentadas heurísticas construtivas.

A partir desse trabalho, o TFP ganhou espaço na literatura de Otimização Combinatória, e diversas variações do problema vêm sendo estudadas. Entre elas, destaca-se o Problema de Formação de Múltiplas Equipes (do inglês, Multiple Team Formation Problem (MTFP)), introduzida por Gutiérrez-Bahamondes *et al.* (2016). O MTFP admite que cada indivíduo possui apenas uma habilidade, considera demandas possivelmente distintas de trabalhadores por habilidade por equipe e permite divisão do tempo de dedicação do trabalhador entre mais de uma equipe. Os autores apresentam um modelo de otimização quadrática e outro de programação por restrições, além de uma heurística. Em sequência, Campêlo e Figueiredo (2021) apresentam uma formulação de programação linear-inteira e desigualdades válidas para o MTFP, com resultados computacionais mostrando que esta abordagem consegue superar a formulação quadrática anterior.

Já no trabalho apresentado em (CAMPÊLO *et al.*, 2020), é estudada outra variação, chamada Problema de Formação de Equipes Socio-Técnicas (do inglês, Socio-Technical Team Formation Problem (STFP)). No STFP, cada indivíduo pode possuir múltiplas habilidades, mas será atribuído a no máximo uma equipe, usando no máximo uma habilidade. Esse problema

busca encontrar equipes disjuntas, que trabalhem de forma mais harmônica possível, em que cada time requer uma quantidade mínima de habilidades. Uma formulação de programação linear-inteira, um estudo do poliedro associado e uma meta-heurística foram apresentados para o problema.

Recentemente, Kouvatís *et al.* (2020) definiram as primeiras versões do TFP utilizando *grafos de sinais* para representar redes sociais online. Em um grafo de sinais, as arestas (ou arcos) são rotuladas com um dos sinais $\{-, +\}$, para representar sentimentos opostos, como por exemplo gostar/não gostar ou confiança/desconfiança. Este tipo de representação possui uma vasta aplicação em ambientes colaborativos versus conflitantes, onde a competitividade está constantemente presente e tem grande influência na produtividade dos seus trabalhadores (INOHARA, 2003; LI *et al.*, 2015; SHI *et al.*, 2016; ARINIK *et al.*, 2017). Buscando criar equipes de trabalho com estas características, Kouvatís *et al.* (2020) se basearam na *Teoria do Equilíbrio Estrutural* (HEIDER, 1946) para definir formas de mensurar a eficiência de comunicação em um grupo, utilizando análises de *compatibilidade* entre pares de vértices em um grafo de sinal.

Os grafos de sinais foram introduzidos por Harary (1953) exatamente para formalizar a teoria (psicológica) do balanço social ou equilíbrio estrutural. Superficialmente falando, tal teoria se baseia na noção de consistência cognitiva entre amizade (+) e hostilidade (-). Isso sugere que um amigo de meu amigo é provavelmente meu amigo ou pode se tornar um, um inimigo do meu amigo é provavelmente meu inimigo também, e um inimigo do meu inimigo é um amigo em potencial. Em termos do grafo, estamos interessados na paridade do número de arestas negativas de um caminho entre um par de indivíduos para inferir uma potencial ou real relação amistosa entre eles.

Formalmente, um grafo de sinais pode ser representado por um par $G = (V, E^+ \cup E^-)$, onde V é o conjunto de vértices, E^+ o conjunto de arestas (ou arcos) positivas e E^- o conjunto de arestas (ou arcos) negativas. Considerando o princípio do *equilíbrio estrutural*, Kouvatís *et al.* (2020) chamam dois vértices adjacentes como *compatíveis* (resp. *incompatíveis*) se estão conectados (diretamente) por uma aresta positiva (resp. negativa). Quando não adjacentes, a compatibilidade do par pode ser definida de diversas formas alternativas. Uma delas é considerar dois vértices *compatíveis* se e somente se existe ao menos um *caminho positivo* entre eles, ou seja, um caminho com um número par de arestas negativas; caso contrário, o par de vértices é *incompatível*. Formas mais restritivas de compatibilidade consideram apenas caminhos mínimos. Para defini-las, seja SP_{uv} o conjunto de todos os caminhos mínimos entre os vértices u e v em G ,

com SP_{uv}^+ o subconjunto desses caminhos que são positivos. Kouvatis *et al.* (2020) definem três possíveis conjuntos de pares compatíveis:

- $C_1 = \{(u, v) : SP_{uv} = SP_{uv}^+\}$: todos os caminhos mínimos são positivos,
- $C_2 = \{(u, v) : |SP_{uv}^+| > |SP_{uv} \setminus SP_{uv}^+|\}$: a maioria dos caminhos mínimos são positivos,
- $C_3 = \{(u, v) : SP_{uv}^+ \neq \emptyset\}$: existe um caminho mínimo positivo.

É fácil vermos que $C_1 \subseteq C_2 \subseteq C_3$. Além dessas, outras definições de compatibilidade podem ser encontradas em (KOUVATIS *et al.*, 2020). Utilizando estas premissas, foram definidas novas versões do TFP que buscam encontrar equipes (subgrafos), onde cada par de trabalhadores (vértices) pertencentes a uma mesma equipe deve ser compatível.

1.3 Problema Estudado e Organização do Texto

Estudaremos, neste trabalho, um novo problema de formação de equipes, que consiste em determinar um número pré-estabelecido de grupos disjuntos de indivíduos, cada um deles cobrindo habilidades requeridas, de modo a minimizar a distância de compatibilidade entre seus membros. As relações entre os indivíduos serão representadas por um grafo de sinais, e o conceito de compatibilidade será importado de (KOUVATIS *et al.*, 2020). De maneira geral, o problema introduzido pode ser visto como aquele estudado por Kouvatis *et al.* (2020) agora com requisitos de habilidades em cada equipe.

Iremos considerar duas versões do problema: uma em grafos não direcionados de sinais e outra em digrafos de sinais. Nossa abordagem será via programação matemática. Apresentaremos formulações de programação inteira, aprimoradas por estratégia de decomposição e inclusão de desigualdades válidas. Experimentos computacionais serão conduzidos e analisados para avaliar o desempenho dos métodos de solução propostos, tanto em instâncias geradas de forma pseudoaleatória quando derivadas de dados de redes sociais reais. Com mais detalhes, os próximos capítulos da dissertação estão organizados da seguinte forma.

No Capítulo 2 apresentaremos noções básicas de teoria dos grafos, programação matemática e complexidade computacional, necessários para melhor compreensão do texto.

No Capítulo 3 apresentaremos formalmente o problema do Caminho Positivo Mínimo (CPM) em um grafo de sinal, cuja resolução é parte preliminar de métodos propostos para o problema principal. Revisitaremos o CPM, compilando e reescrevendo algoritmos, resultados de complexidade e formulações matemáticas. Em particular, veremos que o CPM é polinomial em grafos não direcionados, mas NP-Difícil em digrafos. Além disso, apresentaremos

diferentes estratégias de solução para o CPM em diferentes casos do grafo de sinais de entrada. Esses resultados foram apresentados no Encontro de Teoria da Computação em agosto de 2023 (ALBUQUERQUE *et al.*, 2023).

No Capítulo 4, definiremos e proporemos abordagens para resolução exata do problema introduzido nesta dissertação, o Problema de Formação de Equipes com Caminhos Positivos. Primeiro, apresentaremos duas formulações matemáticas e, em seguida, veremos como essas formulações podem ser decompostas, levando à resolução desacoplada de dois subproblemas: (i) o cálculo de caminhos positivos mínimos entre pares de vértices e (ii) a alocação de indivíduos compatíveis nas equipes, de modo a cobrir os requisitos de habilidades. Tal decomposição gera modelos com menor número de variáveis e restrições, possibilitando resolução mais rápida que a formulação original. Também neste capítulo derivamos famílias de desigualdades válidas para fortalecer o modelo do problema (ii), algumas delas inspiradas no trabalho de Campêlo *et al.* (2020).

No Capítulo 5, iremos destacar alguns dos experimentos computacionais realizados. Descreveremos as instâncias utilizadas nos testes, tanto baseadas em grafos não direcionados de sinais como em digrafos de sinais. Utilizaremos o solver *CPLEX* para resolver as formulações de Programação Linear-Inteira (PLI). Primeiro, compararemos as estratégias de solução para o CPM. Depois, focaremos em comparar os métodos de solução do problema de formação de equipes, apresentado no capítulo anterior. Veremos que a formulação inicial proposta para o problema possui desempenho pior que a segunda estratégia de solução, baseada na decomposição do modelo. Além disso, observaremos que o uso de algumas desigualdades válidas apresentadas geraram uma melhoria adicional no desempenho computacional do modelo decomposto. Resultados computacionais preliminares com o problema em grafos não direcionados foram apresentados no Simpósio Brasileiro de Pesquisa Operacional de 2022 (ALBUQUERQUE *et al.*, 2022).

No Capítulo 6, mostraremos as conclusões e trabalhos futuros que julgamos interessantes para o problema.

2 FUNDAMENTOS

Neste capítulo, apresentaremos alguns conceitos das áreas de programação matemática, teoria dos grafos e, em particular, grafos de sinais, que serão aplicados nos capítulos seguintes. Também falaremos brevemente sobre classes de complexidade computacional.

2.1 Programação Matemática

Na sociedade, diversos problemas práticos podem ser estudados sob uma ótica matemática. O processo de tradução do problema para linguagem matemática é chamado de modelagem, e a abstração resultante é denominada de modelo matemático. Dentro desse grupos de problemas existem os problemas que são chamados de problemas de otimização e, para sua resolução, geramos um modelo matemático com o objetivo de encontrar a melhor solução dentro de um conjunto de soluções possíveis.

A programação matemática se refere a uma grande área de estudo cujo propósito básico é a construção e a resolução de modelos matemáticos que consistem na maximização ou minimização de uma função definida sobre o conjunto das possíveis soluções viáveis. Esses modelos se prestam à resolução de problemas de otimização em áreas diversas, como da saúde, de transportes, de energia, de sustentabilidade, entre outras. Um dos ramos da programação matemática é a Programação Linear (PL). Em (DANTZIG, 1963), de maneira inovadora, foi publicado o método *Simplex* para solucionar modelos (ou formulações) de PL, sendo um marco importante para a área e que influencia na modelagem e solução de diversos problemas.

Um modelo de PL consiste em minimizar (ou maximizar) uma função linear sobre as variáveis, denominada função objetivo, sujeito a um número finito de restrições lineares sobre essas variáveis, denominadas restrições. Genericamente, uma formulação de PL é da forma:

$$\text{minimize } c_1x_1 + \dots + c_nx_n \quad \min \sum_{j=1}^n c_jx_j \quad (2.1)$$

$$\text{sujeito a } \left\{ \begin{array}{c} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \end{array} \right\} \Leftrightarrow \text{s.a: } \sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i \in \{1, 2, \dots, m\} \quad (2.2)$$

$$x_1 \geq 0, \dots, x_n \geq 0 \quad x_j \geq 0, \quad j \in \{1, 2, \dots, n\} \quad (2.3)$$

As expressões (2.1) e (2.2) representam, respectivamente, a função objetivo e o

conjunto de restrições, enquanto (2.3), também chamadas restrições de não negatividade, definem o domínio das variáveis de decisão do problema. Quando no problema algumas variáveis precisam ter valores inteiros, chamamos de problema de Programação Inteira Mista (PIM). Genericamente um problema de PIM é da forma

$$(\mathcal{P}) \quad \min \{c^T x + h^T y : Ax + Dy \leq b, y \geq 0, x \in \mathbb{Z}_+^n\}$$

onde $A_{m \times n}$ e $D_{m \times p}$ são, respectivamente, as matrizes de coeficientes racionais das m restrições para as n variáveis x , que são inteiras, e p variáveis y , que são reais; os vetores $c \in \mathbb{Q}^n$ e $h \in \mathbb{Q}^p$ são coeficientes ou custos, que determinam a função objetivo. Defina $f(x, y) = c^T x + h^T y$ e $P = \{(x, y) \in \mathbb{Z}^n \times \mathbb{R}^p : Ax + Dy \leq b, x \geq 0, y \geq 0\}$, denominado conjunto ou região viável. Cada ponto $(x, y) \in P$ é dito *solução viável* para o problema. Nosso objetivo no problema de minimização é encontrar uma solução $(x^*, y^*) \in P$ tal que $f(x^*, y^*) \leq f(x, y)$ para todo $(x, y) \in P$, chamada de *solução ótima*.

A relaxação linear de $\mathcal{P}$ é o problema de PL

$$\min \{c^T x + h^T y : Ax + Dy \leq b, y \geq 0, x \geq 0\},$$

obtido com a remoção das restrições de integralidade de $\mathcal{P}$. Similarmente, o conjunto $\bar{P} = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : Ax + Dy \leq b, x \geq 0, y \geq 0\}$ é uma relaxação do conjunto viável P .

Considerando ainda o problema $\mathcal{P}$, no caso em que todas as variáveis precisam ser inteiras, ou seja, considerando $p = 0$, temos que o conjunto viável é descrito por $P = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$. Esse novo problema é dito de PLI, ou simplesmente, Programação Inteira (PI).

Uma desigualdade da forma $\pi^T x \leq \pi_0$ é chamada de *desigualdade válida* para um conjunto C , se $\pi^T x \leq \pi_0$ é satisfeito para todos os pontos $x \in C$. Em particular, se $\pi^T x \leq \pi_0$ é desigualdade válida para o conjunto viável $P = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ de um problema de PIM, então $P = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0, \pi^T x \leq \pi_0\}$ e $\bar{P} \supseteq \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0, \pi^T x \leq \pi_0\}$. Em outros termos, a inclusão da desigualdade não muda o conjunto viável do problema, mas pode reduzir o da relaxação linear.

Mais detalhes sobre PL são apresentados em (BREGALDA *et al.*, 1983; CHVATAL *et al.*, 1983; BAZARAA *et al.*, 2011) e sobre PIM/PI em (SCHRIJVER, 1998; WOLSEY, 2020).

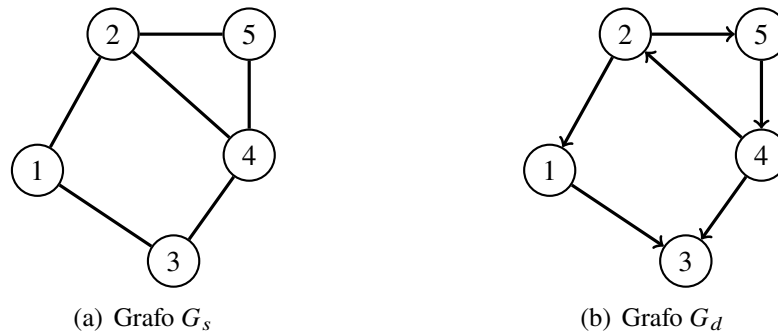
2.2 Teoria dos Grafos

Um *grafo simples* é um par ordenado de conjuntos representados por $G = (V, E)$, onde $E \subseteq V \times V$. Os elementos do conjunto V são chamados de *vértices* e os elementos do

conjunto E são chamados de *arestas*, que definem uma relação entre pares de vértices distintos. Os vértices que definem uma aresta são seus extremos. No caso em que as arestas são pares ordenados, elas são usualmente chamadas *arcos*, e G é dito *grafo direcionado* (ou *digrafo*). Neste caso, as relações entre os vértices não são simétricas, ou seja, $(u, v) \neq (v, u)$. Caso o grafo não seja claro pelo contexto, utilizamos $V(G)$ e $E(G)$, respectivamente, para o conjunto de vértices e arestas do grafo G . Além disso, dizemos que um grafo G é *ponderado* se existe uma atribuição para cada aresta/arco do grafo, em outros termos existe uma função de pesos $w : E(G) \rightarrow \mathbb{R}$, que relaciona cada $e \in E(G)$ a um valor real.

A Figura 1 apresenta uma representação gráfica de dois grafos, um grafo $G_s = (V, E)$ e um digrafo $G_d = (V, A)$, onde $V = \{1, 2, 3, 4, 5\}$, $E = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 4), (4, 5)\}$ e $A = \{(2, 1), (1, 3), (4, 2), (2, 5), (4, 3), (5, 4)\}$.

Figura 1 – Exemplo de um grafo e um digrafo



Fonte: elaborado pelo autor.

Um u, v -caminho em um grafo $G = (V, E)$ é uma sequência de vértices $P_{u,v} = \langle u = v_1, v_2, \dots, v_w, v_{w+1} = v \rangle$ de forma que nenhum vértice é repetido e $(v_i, v_{i+1}) \in E$, para todo $i \in \{1, \dots, w\}$. Dizemos que $u = v_1$ e $v = v_{w+1}$ são os extremos do caminho, que (v_i, v_{i+1}) são suas arestas, e que w , ou seja, a quantidade de arestas, é seu *comprimento* (ou tamanho). Se G for ponderado, o *peso do caminho* é definido pela soma dos pesos de suas arestas. Para o caso em que permitimos a repetição de vértices, porém não podemos repetir arestas/arcos, chamaremos a sequência de u, v -trilha. Em um u, v -passeio podemos repetir vértices e arestas/arcos, e seu *comprimento* (ou tamanho) será a quantidade de arestas/arcos contando repetições.

Com isso, um grafo não direcionado G é dito *conexo* se existe caminho entre todo par de vértices distintos em G ; caso contrário G é *desconexo*. Um *ciclo* em um grafo G é a união de um u, v -caminho de tamanho maior que 1 e uma aresta (v, u) . Dizemos que G é *acíclico* se não possuir ciclos.

Considere um vértice v de G . Definimos a *vizinhança* de v como os vértices que estão diretamente ligados a v por uma aresta, denotada por $N_G(v) = \{u \in V : (u, v) \in E\}$. O *grau* do vértice v é definido pela cardinalidade da sua vizinhança, $d_G(v) = |N(v)|$.

Um problema clássico em grafos é o problema do *Caminho Mínimo*, cujo objetivo é, dados um grafo G e dois vértices distintos $u, v \in V$, encontrar um u, v -caminho com a menor quantidade de arestas (ou peso, no caso em que G é ponderado). Existem diversos algoritmos na literatura para resolver esse problema, quando não existe ciclo com peso negativo no grafo. Em particular, no caso em que a função de pesos do grafo ponderado é definida por $w : E(G) \rightarrow \mathbb{R}_+$, isto é, todos os pesos das arestas são positivos, o problema é resolvido em tempo polinomial com o *Algoritmo de Dijkstra* (DIJKSTRA, 1959). Como exemplo, seja a Figura 1(a) e os vértices 1 e 5. Dois 1,5-caminhos possíveis são $P_{1,5}^1 = \langle 1, 3, 4, 5 \rangle$ e $P_{1,5}^2 = \langle 1, 2, 5 \rangle$, que possuem respectivamente comprimentos 3 e 2, em particular $P_{1,5}^2$ é o caminho mínimo. No caso com grafo direcionado, considere os vértices 2 e 3 no grafo G_d na Figura 1(b). Dois 2,3-caminhos são $P_{2,3}^1 = \langle 2, 5, 4, 3 \rangle$, de tamanho 3, e $P_{2,3}^2 = \langle 2, 1, 3 \rangle$, de tamanho 2, sendo $P_{2,3}^2$ um caminho mínimo.

Um *emparelhamento* em um grafo $G = (V, E)$ é um subconjunto de arestas $M \subseteq E$ que não possuem uma extremidade em comum. Dizemos que um vértice $v \in V$ é *saturado* pelo emparelhamento M (ou que M cobre v) se v é extremo de alguma aresta de M . Quando todos os vértices do grafo G são cobertos por M , chamamos M de *emparelhamento perfeito*. O emparelhamento M é *máximo* se possui o maior número de arestas entre todos os possíveis emparelhamentos. Um *caminho alternante* em relação a M é tal que as arestas do caminho, na ordem, alternam entre arestas de $E(G) \setminus M$ e de M ou vice-versa. No caso em que M não cobre os vértices extremos do caminho, este é denominado *caminho aumentante* com relação a M , ou caminho M -aumentante.

Durante o presente trabalho usaremos bastante o conceito de grafo de sinal, do inglês *signed graph*. Em um grafo de sinal $G = (V, E)$, temos que $V(G)$ é o conjunto de vértices e cada aresta/arco em $E(G)$ possui um rótulo positivo ou negativo, mais especificamente, o conjunto $E(G) = E^+(G) \cup E^-(G)$ pode ser dividido em dois conjuntos disjuntos, onde $E^+(G)$ representam as arestas/arcs com sinal positivo e $E^-(G)$ com sinal negativo.

Maiores detalhes sobre a teoria dos grafos podem ser encontrados em (CORMEN *et al.*, 2001; WEST, 2000).

2.3 Algoritmos e Classes de Problemas

Um algoritmo é qualquer procedimento bem definido que recebe um valor(ou conjunto de valores) como entrada e produz um valor(ou conjunto de valores) como saída, em um período de tempo finito. O algoritmo pode ser visto como uma ferramenta para resolver *problemas computacionais*, onde o problema especifica a relação entrada/saída desejada para as instâncias do problema e o algoritmo descreve o procedimento computacional específico para obter essa relação entrada/saída desejada para todas as instâncias do problema. Dizemos que um algoritmo é polinomial se seu consumo de tempo no pior caso é limitado por uma função polinomial no tamanho das instâncias do problema.

Um *problema de decisão* é uma pergunta sobre os parâmetros de entrada do problema e tem como resposta um booleano(sim ou não). O conjunto de todos os problemas de decisão que possuem um algoritmo que o resolva em tempo polinomial define a classe de complexidade P . A classe NP consiste nos problemas de decisão que possuem um verificador polinomial, isto é, para cada instância em que a resposta é sim, existe um *certificado* de tamanho polinomial que pode ser usado por um verificador, em tempo polinomial, que a resposta para a instância é realmente sim. Um certificado é uma prova de que a instância tem resposta sim.

Sejam dois problemas de decisão A e B . Dizemos que A é polinomialmente redutível à B se existe um algoritmo polinomial r que transforma uma instância I de A em uma instância $r(I)$ de B , de tal forma que a instância I tem resposta sim para o problema A se, e somente se, a instância $r(I)$ tem resposta sim para o problema B .

Na teoria de complexidade computacional, um problema de decisão é chamado de NP -Difícil se todo problema da classe NP pode ser reduzido polinomialmente a esse problema. Um problema é dito NP -completo se for NP -Difícil e estiver na classe NP . Perceba que um problema de decisão A ser NP -completo significa que pertence a classe NP e qualquer problema diferente que também seja NP -completo pode ser reduzido em tempo polinomial ao problema A .

Nesse trabalho estudaremos problemas de otimização, conceito definido na Subseção 2.1. Entretanto, podemos transformar problemas de otimização em sua versão de decisão quando perguntamos se o valor ótimo do problema de otimização é limitado superiormente, para problemas de minimização, ou limitado inferiormente, para problemas de maximização, por um valor fixo K . Dessa forma, temos um problema de decisão correspondente à sua versão de otimização. Além disso, se o problema de decisão for NP -Completo, então o problema de otimização relativo será NP -Difícil.

3 CAMINHO POSITIVO MÍNIMO

Neste capítulo iremos explorar o problema do Caminho Positivo Mínimo (do inglês, Shortest Positive Path (SPP)). Definiremos o problema e, primeiro, veremos algoritmos para o caso em que o grafo (ou digrafo) de entrada é balanceado, ou seja, não contém ciclo com número ímpar de arestas negativas. Em seguida, analisaremos o caso para grafos gerais não direcionados e, no fim, o caso para digrafos gerais.

3.1 Definição do Problema

Seja $G = (V, E^+ \cup E^-)$ um grafo de sinais, onde V é o conjunto de vértices, E^+ e E^- são os conjuntos de arestas (ou arcos) positivas e negativas, respectivamente. Dado um subgrafo H em G , seja $E^+(H)$ (resp. $E^-(H)$) o conjunto de arestas de H em E^+ (resp. E^-). O sinal de H é o produto dos sinais de suas arestas, quer dizer, H é positivo (resp. negativo) se $|E^-(H)|$ é par (resp. ímpar). Em particular, um caminho ou ciclo é positivo (resp. negativo) se possui uma quantidade par (resp. ímpar) de arestas negativas.

Considere uma função de pesos $w : E^+ \cup E^- \rightarrow \mathbb{R}_+$ definida sobre as arestas (ou arcos). Seja $w(P)$ o peso de um caminho P em G , isto é, a soma dos pesos de suas arestas. Dados $u, v \in V$, seja $\delta^+(u, v)$ (resp. $\delta^-(u, v)$) o menor peso de um u, v -caminho positivo (resp. negativo). Esse valor é $+\infty$, caso tal caminho não exista. Quando os pesos são unitários, $\delta^+(u, v)$ (resp. $\delta^-(u, v)$) é tamanho de um menor u, v -caminho positivo (resp. negativo).

O problema do Caminho Positivo Mínimo (do inglês, Shortest Positive Path (SPP)) consiste em, dados um grafo de sinais e dois vértices u e v , determinar um u, v -caminho positivo de peso mínimo. Em grafos não direcionados, Hansen (1984) menciona que SPP pode ser resolvido em tempo polinomial, usando uma redução para um problema de emparelhamento, sugerida por Edmonds para resolver um problema similar. Na Seção 3.3, detalhamos tal redução.

Para o caso direcionado, Hansen (1984) apresenta um algoritmo polinomial para encontrar uma u, v -trilha positiva (permitindo a repetição de vértices) de peso mínimo. Ele aponta que o algoritmo na verdade encontra um u, v -caminho se não existe ciclo negativo no grafo. Na Seção 3.2, mostramos uma descrição equivalente deste algoritmo e apresentamos uma prova de corretude para SPP, quando não existe ciclo negativo. Vale mencionar que o problema sob esta condição também foi estudado em (GULER *et al.*, 2014; GULER *et al.*, 2015) para analisar a propagação de uma mensagem em uma rede de sinal. Nesses trabalhos, o referido

algoritmo é apresentado sem prova de corretude.

Já para digrafos de sinais arbitrários, Hansen (1984) prova que verificar a existência de um u, v -caminho positivo é NP-Completo, a partir de uma redução do problema de verificar a existência de um u, v -caminho com um número par arestas em um grafo (LAPAUGH; PAPADIMITRIOU, 1984). No trabalho também é apresentado um algoritmo branch-and-bound para este caso. Na Seção 3.4, apresentamos outra redução, bem simples, a partir do problema de determinar um u, v -caminho mínimo passando por um nó intermediário específico. Adicionalmente, apresentamos duas formulações de programação linear-inteira para o problema.

Ainda sobre a literatura do SPP, caminhos positivos de tamanho mínimo foram usados para definir relações de compatibilidade entre indivíduos para pertencerem a uma mesma equipe de trabalho (KOUVATIS *et al.*, 2020; ALBUQUERQUE *et al.*, 2022). Tais noções de compatibilidade se baseiam na teoria do balanço social (HEIDER, 1946; CARTWRIGHT; HARARY, 1956) e serão explorados nos capítulos seguintes.

3.2 Grafos de Sinais Balanceados

Iremos analisar primeiro o problema SPP em grafos (ou digrafos) de sinais balanceados. Um grafo de sinais não direcionado G é dito *balanceado* se não contém ciclo com número ímpar de arestas negativas (CARTWRIGHT; HARARY, 1956). Existem algoritmos polinomiais para verificar se um grafo de sinais é balanceado (HARARY; KABELL, 1980). Vamos estender essa definição para o caso em que o grafo é direcionado, considerando que um digrafo é balanceado se não possui ciclo (direcionado) com número ímpar de arcos negativos. Nesta seção, admita que o grafo (ou digrafo) G é balanceado.

Mostramos que a adaptação do algoritmo de Dijkstra mostrada no Algoritmo 1 resolve SPP em um digrafo balanceado. Na verdade, o algoritmo determina os pesos dos caminhos mínimos entre um vértice s e todos os demais. Vale destacar que o algoritmo também funciona em um grafo não direcionado G , bastando aplicá-lo à orientação simétrica de G (onde cada aresta é substituída por dois arcos com sentidos opostos). Esse algoritmo é uma reescrita do procedimento apresentado em (HANSEN, 1984). Abaixo, provamos sua corretude quando o grafo de sinais de entrada é balanceado.

No que segue, dado $\sigma \in \{-, +\}$, denotamos por $\bar{\sigma}$ o sinal complementar, ou seja, $\{\sigma, \bar{\sigma}\} = \{-, +\}$. Denote por $\mathcal{P}^+$ (resp. $\mathcal{P}^-$) o conjunto de todos os caminhos positivos (resp. negativos) em G .

Algoritmo 1: Algoritmo para o caminho positivo mínimo.

Entrada: $G = (V, E^+ \cup E^-)$, $s \in V$, $w : E^+ \cup E^- \rightarrow \mathbb{R}_+$
Saída: Menores pesos $d_+(v)$ de caminho positivo entre s e cada $v \in V$

```

1  $d_+(v) = d_-(v) = +\infty \ \forall v \in V$ ;
2  $d_+(s) = 0, d_-(s) = +\infty$ ;
3  $d_\sigma(v) = w_{sv} \ \forall \sigma \in \{-, +\} \ \forall v \in V : (s, v) \in E^\sigma$ ;
4  $R_+ = R_- = \{s\}$  /* Vertices com  $d_+$  e  $d_-$  conhecidos */
5  $k = 0$  /* Marcador de iteração */
6 Enquanto  $R_+ \neq V$  faça
7    $k = k + 1$ ;
8    $\epsilon_\sigma = \inf\{d_\sigma(u) : u \in V \setminus R_\sigma\} \ \forall \sigma \in \{-, +\}$ ;
9   se  $(\epsilon_+ \neq +\infty) \parallel (\epsilon_- \neq +\infty)$  então
10    Seja  $\sigma \in \{-, +\}$  tal que  $\epsilon_\sigma = \min\{\epsilon_-, \epsilon_+\}$ ;
11    Seja  $u \in V \setminus R_\sigma$  tal que  $d_\sigma(u) = \epsilon_\sigma$ ;
12     $R_\sigma = R_\sigma \cup \{u\}$ ;
13     $d_\sigma(v) = \min\{d_\sigma(v), d_\sigma(u) + w_{uv}\} \ \forall v \in V \setminus R_\sigma : (u, v) \in E^+$ ;
14     $d_{\bar{\sigma}}(v) = \min\{d_{\bar{\sigma}}(v), d_\sigma(u) + w_{uv}\} \ \forall v \in V \setminus R_{\bar{\sigma}} : (u, v) \in E^-$ ;
15  senão
16     $R_+ = V, R_- = V$ ;
17  fim
18 fim
```

Lema 3.2.1. Em cada iteração k do algoritmo, para todo $\sigma \in \{-, +\}$ e todo $v \in V$, se $d_\sigma(v) < \infty$ então existe caminho em $\mathcal{P}^\sigma$ entre s e v .

Demonstração. Iremos fazer uma prova por indução sobre k , número de iterações. Para $k = 0$, o resultado é trivial. Suponha-o verdadeiro até a iteração $k - 1$. Na iteração k , é suficiente considerar $\varsigma \in \{-, +\}$ e $v \in V$ tais que $d_\varsigma(v)$ tornou-se finito. Isso ocorre no passo 13 com $\varsigma = \sigma$ ou no passo 14 com $\varsigma = \bar{\sigma}$. Devemos mostrar que existe caminho em $\mathcal{P}^\varsigma$ entre s e v . Em ambos os casos, temos que $d_\varsigma(v)$ era infinito e torna-se $d_\sigma(u) + w_{uv} < +\infty$ nesta iteração. Assim, pela hipótese indutiva aplicada a u , existe caminho P_{su} em $\mathcal{P}^\sigma$ entre s e u .

Primeiro, suponha que $v \in V(P_{su})$. Sejam P_{sv} e P_{vu} os dois subcaminhos de P_{su} . Então, $P_{vu} = (v, u)$ ou P_{vu} junto com (u, v) formam um ciclo (positivo). Logo, P_{vu} tem o mesmo sinal da aresta (u, v) . Se $\varsigma = \sigma$, então $(u, v) \in E^+$ e, assim, $P_{vu} \in \mathcal{P}^+$. Logo, P_{su} e P_{sv} têm a mesma paridade de arestas negativas, ou seja, $P_{sv} \in \mathcal{P}^\sigma = \mathcal{P}^\varsigma$. Caso $\varsigma = \bar{\sigma}$, então $(u, v) \in E^-$ e, portanto, $P_{vu} \in \mathcal{P}^-$. Logo, P_{su} e P_{sv} têm paridades diferentes de arestas negativas, ou melhor, $P_{sv} \in \mathcal{P}^{\bar{\sigma}} = \mathcal{P}^\varsigma$. Admita agora que $v \notin V(P_{su})$, ou seja, que $E(P_{su}) \cup \{(u, v)\}$ induz um caminho P_{sv} entre s e v . Se $(u, v) \in E^+$, ou melhor, $\varsigma = \sigma$, então $P_{sv} \in \mathcal{P}^\sigma = \mathcal{P}^\varsigma$. Se $(u, v) \in E^-$, isto é, $\varsigma = \bar{\sigma}$, então $P_{sv} \in \mathcal{P}^{\bar{\sigma}} = \mathcal{P}^\varsigma$. $\square$

Proposição 3.2.1. *Em cada iteração do algoritmo, temos $d_\sigma(v) = \delta^\sigma(s, v)$, para todo $\sigma \in \{-, +\}$ e todo $v \in R_\sigma$. Como $R_+ = V$ ao final do algoritmo, sua corretude segue.*

Demonstração. Primeiro observe que $d_\sigma(v)$, para cada $\sigma \in \{-, +\}$ e $v \in V$, é não crescente ao longo das iterações e, tão logo v entre em R_σ , esse valor permanece inalterado. Vamos chamar essa propriedade de PNC – Propriedade Não Crescente.

A prova será por indução sobre o número k da iteração. Para $k = 0$, temos $R_+ = R_- = \{s\}$ e, trivialmente, $d_+(s) = 0$, $d_-(s) = \infty$. Ver linhas 2 e 4. Suponha que o resultado vale até a iteração $k - 1$. Considere agora a iteração $k \geq 1$. Por PNC, é suficiente considerar o vértice u adicionado a R_σ , $\sigma \in \{-, +\}$, nesta iteração. Note que $u \neq s$. Por conveniência, defina R'_- e R'_+ os conjuntos R_- e R_+ na iteração $k - 1$.

Se não existe caminho em $\mathcal{P}^\sigma$ entre s e u , então $d_\sigma(u) = +\infty$ pelo Lema 3.2.1. Portanto, $d_\sigma(u) = \delta^\sigma(s, u)$. Admita, então, que existe caminho de $\mathcal{P}^\sigma$ entre s e u . Seja P um de peso mínimo, ou seja, $w(P) = \delta^\sigma(s, u)$. Para todo vértice v de P , denote por P_v o subcaminho de P entre s e v . Seja x o vértice de P mais longe de s nesse caminho tal que $P_x \in \mathcal{P}^\varsigma$ e $x \in R'_\varsigma$, para algum $\varsigma \in \{-, +\}$. Tal vértice existe pois $P_s \in \mathcal{P}^+$ e $s \in R'_+$. Além disso, $x \neq u$, pois $P_u = P \in \mathcal{P}^\sigma$ e $u \notin R'_\sigma$, implicando $P_u \notin \mathcal{P}^\varsigma$ ou $u \notin R'_\varsigma$, para todo $\varsigma \in \{-, +\}$. Por conseguinte, $w(P) \geq w(P_x)$. Como $P_x \in \mathcal{P}^\varsigma$ e $x \in R'_\varsigma$, podemos aplicar a hipótese de indução para obter $w(P_x) \geq \delta^\varsigma(s, x) = d_\varsigma(x)$ e, portanto, $w(P) \geq d_\varsigma(x)$.

Seja y o vértice seguinte a x em P . Assuma que $xy \in E^+$ (o caso $xy \in E^-$ é similar). Então, $P_y \in \mathcal{P}^\varsigma$. Como y não satisfaz $P_y \in \mathcal{P}^\varsigma$ e $y \in R'_\varsigma$ (pois x é o vértice mais longe de s), concluímos que $y \in V \setminus R'_\varsigma$. Assim, quando x entra em R_ς (linha 12), então $d_\varsigma(y)$ é atualizado na linha 13, de modo que $d_\varsigma(x) \geq d_\varsigma(y)$. E essa desigualdade se mantém pela propriedade PNC. Finalmente, como u é escolhido para entrar em R_σ quando $y \in V \setminus R'_\varsigma$ também é candidato, temos pelas linhas 8, 10 e 11 que $d_\varsigma(y) \geq d_\sigma(u)$. Logo, $d_\varsigma(x) \geq d_\sigma(u)$. Chegamos portanto a $\delta^\sigma(s, u) = w(P) \geq d_\sigma(u)$, ou melhor, $d_\sigma(u) = \delta^\sigma(s, u)$. $\square$

Corolário 3.2.1. *O problema SPP em grafos de sinais balanceados pode ser resolvido em tempo polinomial.*

3.3 Grafos de Sinais Não Direcionados

Nesta seção admita que G é não direcionado, podendo ter ciclos negativos. Defina o *grafo par* de G como o grafo G_p obtido dividindo cada aresta positiva de G em duas arestas.

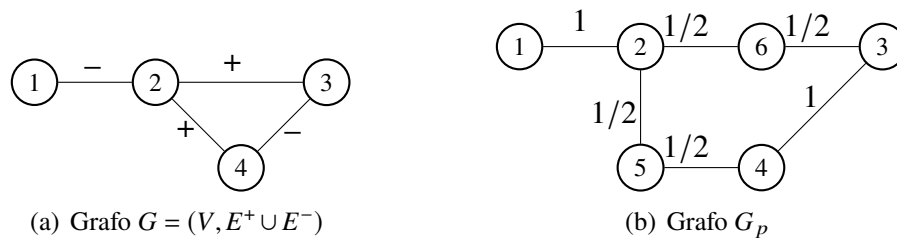
Mais precisamente,

$$V(G_p) = V(G) \cup \{v_e : e \in E^+(G)\}$$

$$E(G_p) = E^-(G) \cup \{(u, v_e), (v_e, v) : e \in E^+(G)\}$$

Além disso, G_p é ponderado em arestas, sendo w_e o peso de cada aresta $e \in E^-(G)$ e $w_e/2$ o peso das arestas (u, v_e) e (v_e, v) para todo $e \in E^+(G)$. Dados $u, v \in V(G)$, existe uma correspondência biunívoca entre os uv -caminhos em G e G_p . Um exemplo pode ser visto na Figura 2, a seguir.

Figura 2 – Exemplo do grafo par G_p de um grafo de sinais G



Fonte: elaborado pelo autor.

Note que a paridade de um uv -caminho em G_p é a paridade do número de arestas negativas no uv -caminho correspondente em G . Além disso, o peso do primeiro caminho é igual ao peso do segundo. Esses fatos levam à seguinte equivalência:

Proposição 3.3.1. *Determinar um s, t -caminho positivo de peso mínimo em G equivale a determinar um s, t -caminho par de peso mínimo em G_p .*

A partir da Prop. 3.3.1, mostramos a seguir que o problema SPP em grafos não direcionados equivale a determinar um emparelhamento perfeito de peso mínimo, sendo portanto polinomial (EDMONDS, 1965a; EDMONDS, 1965b). Para isso, considere um grafo H não direcionado, ponderado em arestas por $\omega : E(H) \rightarrow \mathbb{R}_+$, e dois vértices s e t de H . Para definir um s, t -caminho par de peso mínimo em H , vamos usar um grafo auxiliar H' construído a partir de uma cópia de $H - s$, uma cópia de $H - t$ e arestas ligando os vértices correspondentes nas cópias. Precisamente, $V(H') = \mathcal{V}' \cup \mathcal{V}''$ e $E(H') = \mathcal{E}' \cup \mathcal{E}'' \cup \bar{\mathcal{E}}$, onde

$$\mathcal{V}' = \{v' : v \in V(H - t)\},$$

$$\mathcal{V}'' = \{v'' : v \in V(H - s)\},$$

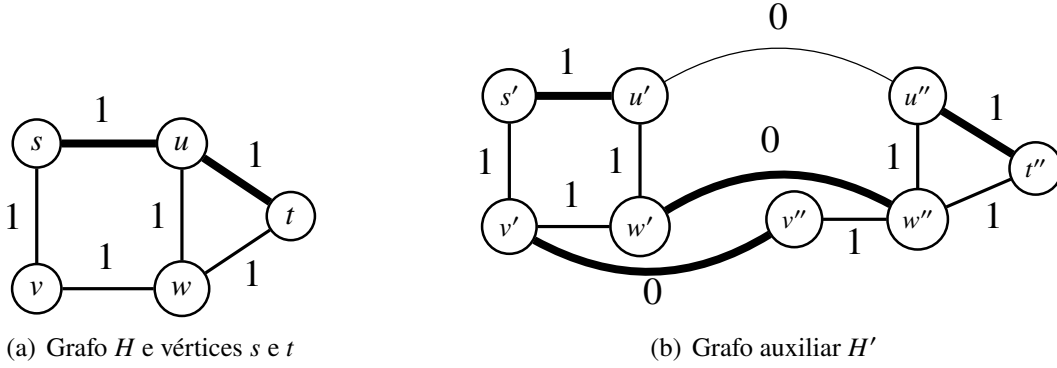
$$\mathcal{E}' = \{(u', v') : (u, v) \in E(H), u, v \neq t\},$$

$$\mathcal{E}'' = \{(u'', v'') : (u, v) \in E(H), u, v \neq s\},$$

$$\bar{\mathcal{E}} = \{(u', u'') : u \in V(H - \{s, t\})\}.$$

O peso das arestas $(u', v') \in \mathcal{E}'$ ou $(u'', v'') \in \mathcal{E}''$ é ω_{uv} , o mesmo da aresta equivalente (u, v) de H . As arestas $(u', u'') \in \bar{\mathcal{E}}$ tem peso zero. A Figura 3 mostra uma ilustração.

Figura 3 – Ilustração do grafo auxiliar H' . Arestas em negrito ilustram a Prop. 3.3.2.



Fonte: elaborado pelo autor.

Proposição 3.3.2 ((LAPAUGH; PAPADIMITRIOU, 1984)). *Um s, t -caminho par de peso mínimo em H é dado por um emparelhamento perfeito de peso mínimo em H' , e vice-versa.*

Demonstração. É suficiente mostrar que todo s, t -caminho par em H corresponde a um emparelhamento perfeito em H' de mesmo peso, e vice-versa.

Primeiro, seja $P = \langle s = v_0, v_1, \dots, v_{2p} = t \rangle$, $p \geq 1$, um s, t -caminho par em H com peso $\omega(P) = \sum_{i=0}^{2p-1} \omega_{v_i v_{i+1}}$. Defina $M = \{(v'_{2i}, v'_{2i+1}) : i = 0, \dots, p-1\} \cup \{(v''_{2i-1}, v''_{2i}) : i = 1, \dots, p\} \cup \{(v', v'') : v \in V(H) \setminus V(P)\}$. Observe que M é um conjunto de arestas disjuntas que cobrem todos os vértices de H' . Logo, M é um emparelhamento perfeito em H' . Além disso, tem peso $\omega(M) = \sum_{i=0}^{p-1} \omega_{v'_{2i} v'_{2i+1}} + \sum_{i=1}^p \omega_{v''_{2i-1} v''_{2i}} = \sum_{i=0}^{2p-1} \omega_{v_i v_{i+1}}$, igual ao peso de P .

Agora, seja M um emparelhamento perfeito em H' com peso $\omega(M)$. Sejam $\mathcal{E}'(M)$, $\mathcal{E}''(M)$ e $\bar{\mathcal{E}}(M)$ as arestas de M que pertencem a $\mathcal{E}'$, $\mathcal{E}''$ e $\bar{\mathcal{E}}$, respectivamente. Note que s é coberto por $\mathcal{E}'(M)$, t é coberto por $\mathcal{E}''(M)$, e $(u', v') \in \mathcal{E}'(M)$ se, e somente se, $(v'', z'') \in \mathcal{E}''(M)$. Logo, $\{(u, v) \in E(H) : (u', v') \in \mathcal{E}'(M) \text{ ou } (u'', v'') \in \mathcal{E}''(M)\}$ são as arestas de um s, t -caminho P em H . Além disso, $w(P) = w(M)$, pois as arestas em $\bar{\mathcal{E}}(M)$ têm peso nulo. $\square$

A partir das Proposições 3.3.1 e 3.3.2 e sabendo que determinar um emparelhamento perfeito de peso mínimo é polinomial (EDMONDS, 1965a; EDMONDS, 1965b), chegamos à seguinte conclusão sobre a complexidade de SPP.

Corolário 3.3.1. *O problema SPP em grafos de sinais não direcionados pode ser resolvido em tempo polinomial.*

Usando a Proposição 3.3.1 e a Proposição 3.3.2, iremos mostrar uma forma alternativa de resolver SPP, transformando-o em um problema de encontrar o emparelhamento induzido por um caminho aumentante. Um caminho aumentante com relação a um emparelhamento

M é um u, v -caminho com um número ímpar de arestas $e_1, e_2, \dots, e_{2p+1}$ tais que (a) u e v não são saturados por M , (b) $e_{2i} \in M$, $i = 1, \dots, p$, e (c) $e_{2i+1} \notin M$, $i = 0, \dots, p$. Note que $(M \setminus \{e_{2i} : i = 1, \dots, p\}) \cup \{e_{2i+1} : i = 0, \dots, p\}$ formam um emparelhamento de tamanho $|M| + 1$. Dizemos que tal emparelhamento é induzido pelo caminho aumentante.

Considere os grafos H e H' apresentados anteriormente. Note que $\bar{E}$ é um emparelhamento em H' de tamanho $|V(H)| - 2$. Mostramos a seguir uma equivalência entre emparelhamentos perfeitos em H' e emparelhamentos induzidos por caminhos aumentantes com relação a $\bar{E}$. Isto nos leva também à equivalência deste último problema com SPP.

Proposição 3.3.3. *M é um emparelhamento perfeito em H' se, e somente se, M é o emparelhamento induzido por um caminho aumentante com relação a $\bar{E}$.*

Demonstração. Como $|\bar{E}| = |V(H)| - 2$, então o emparelhamento induzido por um caminho aumentante com relação a $\bar{E}$ tem tamanho $|V(H)| - 1 = |V(H')|/2$, sendo pois perfeito em H' .

Para a recíproca, considere agora um emparelhamento perfeito M em H' . Pela prova da Proposição 3.3.2, M define um st -caminho par $P = \langle s = v_0, v_1, \dots, v_{2p} = t \rangle$ em H , associado ao caminho $P' = \langle s' = v'_0, v'_1, v''_1, v''_2, v'_2, v'_3, v''_3, \dots, v''_{2p-1}, v''_{2p} = t'' \rangle$ em H' . Além disso, $M \setminus \bar{E} = \{(v'_0, v'_1), (v'_2, v'_3), \dots, (v'_{2p-2}, v'_{2p-1})\} \cup \{(v''_1, v''_2), (v''_3, v''_4), \dots, (v''_{2p-1}, v''_{2p})\}$. Note ainda que P' é aumentante com relação a $\bar{E}$, pois alterna arestas fora de $\bar{E}$ com arestas em $\bar{E}$ e nem s' nem t'' são cobertos por $\bar{E}$. Logo, M é o emparelhamento induzido por P' . $\square$

Corolário 3.3.2. *Determinar um s, t -caminho positivo de peso mínimo em G equivale a determinar um emparelhamento de peso mínimo induzido por um caminho aumentante com relação a $\bar{E}$.*

Pelo corolário acima, SPP também pode ser resolvido em tempo polinomial pelo algoritmo apresentado em (MICALI; VAZIRANI, 1980).

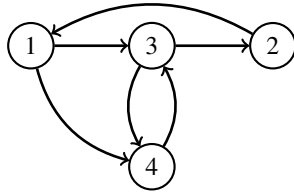
3.4 Digrafos de Sinais

Nesta seção, focaremos no problema SPP em grafos direcionados de sinais (digrafos de sinais) $D = (V, A = A^+ \cup A^-)$, onde V é o conjunto de vértices e A^+ (resp. A^-) são os conjuntos de arcos positivos (resp. negativos).

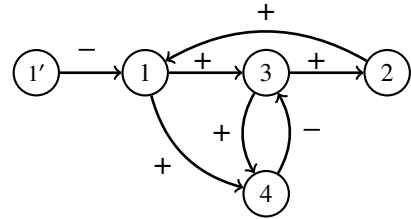
Dados um digrafo D e vértices distintos s, t, m , decidir se existe um s, t -caminho em D tendo m com vértice intermediário é NP-Completo (FORTUNE *et al.*, 1980). Vamos reduzir tal problema para aquele de decidir se existe um s, t -caminho positivo em um digrafo de sinais.

Dados D e $s, t, m \in V(D)$, construímos um digrafo de sinais $G = (V(G), A(G))$ tal que $V(G) = V(D) \cup \{s'\}$ e $A(G) = A(D) \cup \{(s', s)\}$. Os arcos (s', s) e (m, v) , para todo $v \in V(G)$, têm sinal negativo; todos os outros arcos $(u, v) \in A(G)$, com $u \neq m, u \neq s'$, possuem sinal positivo. Veja Figura 4 para uma ilustração.

Figura 4 – Ilustração da redução de NP-Completo



(a) Digrafo D , $(s, t, m) = (1, 2, 4)$



(b) Digrafo de sinais G , $(s', t) = (1', 2)$

Fonte: elaborado pelo autor.

Proposição 3.4.1. *Existe s, t -caminho passando por m em D se, e somente se, existe s', t -caminho positivo em G .*

Demonstração. Perceba que, no digrafo de sinais G , o primeiro arco de um s', t -caminho é (s', s) , com sinal negativo. Com isso, todo s', t -caminho positivo deve passar por m para ter uma quantidade par de arestas negativas. Por outro lado, removendo o vértice s' de um s', t -caminho positivo temos um s, t -caminho passando por m no digrafo D . $\square$

Dado que os problemas acima estão em NP, mostramos que:

Corolário 3.4.1. *Decidir se existe um s, t -caminho positivo em um digrafo de sinais é um problema NP-Completo.*

Note que o problema continua NP-Completo mesmo em grafos orientados, quer dizer, digrafos sem ciclo de tamanho 2 (ou 2-ciclo). De fato, em um digrafo de sinais arbitrário, cada 2-ciclo pode ser transformado em um ciclo de tamanho 4, subdividindo cada arco do 2-ciclo com a adição de um vértice extra e adequando os sinais dos arcos criados para não afetar o sinal dos s, t -caminhos. Precisamente, o arco $a = (u, v)$ de um 2-ciclo, com sinal σ , é transformado em um caminho $\langle u, v_a, v \rangle$, onde os sinais de (u, v_a) e de (v_a, v) são respectivamente σ e $+$.

3.5 Formulações de Programação Inteira

Como o SPP em digrafos de sinais pertence, pois, à classe NP-Difícil, apresentaremos duas formulações de programação linear inteira para o problema. Ao longo desta seção, seja

$G = (V, A = A^+ \cup A^-)$ um digrafo de sinais e $w : A \rightarrow \mathbb{R}_+$ uma função de pesos nos arcos.

Ambas as formulações usam restrições de fluxo para definir uma s, t -trilha junto com restrições que impedem a formação de ciclos. São estas últimas que tornam as formulações diferentes. Para a primeira formulação, utilizaremos a restrição de quebra de ciclo de Miller-Tucker-Zemli (MTZ), apresentada em (MILLER *et al.*, 1960), e usaremos as seguintes variáveis:

$$f_{pq}^{st} = \begin{cases} 1, & \text{se o arco } (p, q) \text{ pertence ao } s, t\text{-caminho,} \\ 0, & \text{caso contrário,} \end{cases} \quad \forall (p, q) \in A$$

$$\pi_p^{st} \geq 0 \quad \forall p \in V : \text{posição do vértice } p \text{ no } s, t\text{-caminho}$$

$$\lambda_{st} \in \mathbb{N} : \text{metade do número de arestas negativas no } s, t\text{-caminho}$$

Propomos assim a formulação $\mathcal{SP}_{MTZ}$:

$$(\mathcal{SP}_{MTZ}) \quad \min \quad \sum_{pq \in A} w_{pq} f_{pq}^{st} \tag{3.1}$$

$$\text{s.a:} \quad \sum_{p \in V: (p, q) \in A} f_{pq}^{st} - \sum_{p \in V: (q, p) \in A} f_{qp}^{st} = \begin{cases} -1, & \text{se } q = s, \\ 1, & \text{se } q = t, \\ 0, & \text{c.c.} \end{cases} \quad \forall q \in V \tag{3.2}$$

$$\pi_p^{st} - \pi_q^{st} + |V| \cdot f_{pq}^{st} \leq (|V| - 1) \quad \forall (p, q) \in A \tag{3.3}$$

$$\sum_{(p, q) \in A^-} f_{pq}^{st} - 2\lambda_{st} = 0 \tag{3.4}$$

$$f_{pq}^{st} \in \{0, 1\}, \quad \forall (p, q) \in A \tag{3.5}$$

$$\pi_s^{st} = 0, \tag{3.6}$$

$$\pi_p^{st} \geq 0 \quad \forall p \in V \setminus \{s\} \tag{3.7}$$

$$\lambda_{st} \in \mathbb{N}^+ \tag{3.8}$$

As restrições (3.2) definem um fluxo saindo de s a t . Para garantir que esse fluxo seja um caminho elementar são adicionadas as clássicas restrições de quebra de ciclo MTZ (3.3). A restrição (3.4) certifica que o caminho criado seja positivo (com número par de arestas negativas). Em (3.5), (3.7) e (3.8) estão as restrições de domínio das variáveis. A função objetivo apresentada em (3.1) deseja minimizar o custo do caminho. Perceba que o modelo ser inviável significa que não existe s, t -caminho positivo.

Na segunda formulação, usaremos as restrições para evitar ciclos apresentadas em (COSTA *et al.*, 2023), substituindo as variáveis π_p^{uv} pelas seguintes:

$$\mu_p^{st} = \begin{cases} 1, & \text{se o sinal do subcaminho entre } s \text{ e } p \text{ no } s, t\text{-caminho é positivo,} \\ 0, & \text{caso contrário,} \end{cases} \quad \forall p \in V,$$

e as restrições (3.3), (3.4), (3.7) e (3.8), pelas seguintes restrições,

$$\mu_q^{st} \geq f_{pq}^{st} - \mu_p^{st} \quad (p, q) \in A^- \quad (3.9)$$

$$\mu_q^{st} \leq 2 - f_{pq}^{st} - \mu_p^{st} \quad (p, q) \in A^- \quad (3.10)$$

$$\mu_q^{st} \geq f_{pq}^{st} + \mu_p^{st} - 1 \quad (p, q) \in A^+ \quad (3.11)$$

$$\mu_q^{st} \leq 1 - f_{pq}^{st} + \mu_p^{st} \quad (p, q) \in A^+ \quad (3.12)$$

$$\mu_s^{st} = \mu_t^{st} = 1 \quad (3.13)$$

$$\mu_p^{st} \in \{0, 1\}, \quad \forall p \in V \setminus \{s, t\} \quad (3.14)$$

Para entender as restrições acima, primeiro admita que o arco (p, q) faz parte do s, t -caminho, ou seja, $f_{pq}^{st} = 1$. Temos então que $\mu_q^{st} = 1 - \mu_p^{st}$, se $(p, q) \in A^-$, ou $\mu_q^{st} = \mu_p^{st}$, se $(p, q) \in A^+$, mostrando a corretude das restrições. No caso em que $f_{pq}^{st} = 0$, as restrições são redundantes, pois tornam-se $-\mu_p^{st} \leq \mu_q^{st} \leq 2 - \mu_p^{st}$ ou $\mu_p^{st} - 1 \leq \mu_q^{st} \leq 1 + \mu_p^{st}$, dependendo se $(p, q) \in A^-$ ou $(p, q) \in A^+$.

Dessa forma, obtemos a formulação $\mathcal{SP}_{SIGN}$ para caminho positivo de custo mínimo em grafos direcionados (COSTA *et al.*, 2023):

$$(\mathcal{SP}_{SIGN}) \quad \min \sum_{pq \in A} w_{pq} f_{pq}^{st} \quad (3.15)$$

$$\text{s.t.} \quad (3.2), (3.5),$$

$$(3.9) - (3.14)$$

Note que as restrições (3.9) - (3.14) além de estabelecerem sinal positivo para o s, t -caminho, também garantem que não haverá repetição de vértice, dado que os pesos dos arcos são não negativos. De fato, para haver repetição de vértice nesse caso, haveria ciclo positivo de peso nulo (que pode ser descartado) ou ciclo negativo envolvendo um vértice v do s, t -caminho, o que atribuiria sinais conflitantes para μ_v^{st} a partir dos dois vértices antecessores na trilha (o antecessor que não pertence ao ciclo e o que pertence ao ciclo).

Observe também que a integralidade das variáveis μ_p^{st} , para todo $p \in V$, poderia ser descartada, conforme explicação dada acima para as restrições (3.9) - (3.13). Mais precisamente, a integralidade das variáveis f_{pq}^{st} garante que serão binárias as variáveis μ_p^{st} , para todo vértice p do s, t -caminho formado. Por

outro lado, mantendo as variáveis μ^{st} binárias, podemos relaxar a integralidade das variáveis f^{st} , ou seja, deixar $f_{pq}^{st} \geq 0$ para todo $(p, q) \in A$. Seu valor permanecerá binário devido à total unimodularidade da matriz das restrições (3.2) e posto que (3.9) - (3.12) apenas fornecem limites inteiros para as variáveis f^{st} quando as variáveis μ^{st} têm valor binário.

4 FORMAÇÃO DE EQUIPES COM CAMINHOS POSITIVOS MÍNIMOS

Nesta seção, vamos apresentar conceitos iniciais para conseguir definir formalmente o Problema de Formação de Equipes com Caminhos Positivos Mínimos (do inglês, Team Formation Problem with Shortest Positive Paths (TFP_SPP)). Definiremos suas versões em grafos não direcionados e em grafos direcionados. Depois de definir o problema, mostraremos estratégias para solucioná-lo que utilizam formulações de programação inteira e decomposições.

4.1 Definição do Problema

Consideraremos que o conjunto de trabalhadores e suas relações interpessoais podem ser representados por um grafo de sinais $G = (V, E, \text{sign})$, onde $V(G)$ representa o conjunto de trabalhadores, $E(G)$ descreve suas relações diretas, e $\text{sign} : E(G) \rightarrow \{+, -\}$ é uma função que indica o sinal da aresta (resp. arco) que conecta dois trabalhadores, se a aresta (resp. arco) existir. Como sempre, $E^+(G) = \{e \in E(G) : \text{sign}(e) = +\}$ e $E^-(G) = \{e \in E(G) : \text{sign}(e) = -\}$ denotam as arestas positivas e negativas de G , respectivamente.

Considere $P_{uv} = \langle u = v_1, \dots, v_{w+1} = v \rangle$ um u, v -caminho de tamanho w em G , com $(v_i, v_{i+1}) \in E$, para todo $i \in \{1, \dots, w\}$. Como definido na Seção 3.1, o sinal do caminho P_{uv} é dado pela paridade do número de arestas negativas, ou melhor, $\text{sign}(P_{uv}) = \prod_{i \in \{1, \dots, w\}} \text{sign}(v_i, v_{i+1})$. O caminho P_{uv} é positivo se $\text{sign}(P_{uv}) = +$ e negativo se $\text{sign}(P_{uv}) = -$. Vamos denotar por $\mathcal{P}_{uv}^+$ e $\mathcal{P}_{uv}^-$ os conjuntos de u, v -caminhos positivos e negativos, respectivamente.

Similarmente a Kouvatlis *et al.* (2020), a partir do grafo de sinais G , definiremos uma relação de *compatibilidade* $\text{Comp} \subseteq V \times V$, tal que $(u, v) \in \text{Comp}$ se, e somente se, u e v puderem trabalhar juntos numa mesma equipe. Dois requisitos naturais para Comp são: (I) reflexividade – $(u, u) \in \text{Comp}$, e (II) simetria – $(u, v) \in \text{Comp}$ se e somente se $(v, u) \in \text{Comp}$.

Primeiro, definimos Comp quando G é não direcionado. Além da reflexibilidade e simetria, é exigido que a relação Comp satisfaça as duas propriedades intuitivas a seguir:

1. *Compatibilidade por aresta positiva*: $(u, v) \in \text{Comp}$, para todo $(u, v) \in E^+(G)$.
2. *Incompatibilidade por aresta negativa*: $(u, v) \notin \text{Comp}$, para todo $(u, v) \in E^-(G)$.

Para um par de vértices não adjacentes, a compatibilidade/incompatibilidade será definida a partir da existência ou não de caminho positivo entre eles, ou precisamente:

3. *Compatibilidade por algum caminho positivo*: $(u, v) \in \text{Comp}$, para todo $(u, v) \notin E$ tal que $\mathcal{P}_{uv}^+ \neq \emptyset$.
4. *Incompatibilidade por nenhum caminho positivo*: $(u, v) \notin \text{Comp}$, para todo $(u, v) \notin E$ tal que $\mathcal{P}_{uv}^+ = \emptyset$.

Resumindo, quando G é não direcionado, temos

$$Comp = \{(u, v) \in V \times V : (u, v) \notin E^-(G) \text{ e } \mathcal{P}_{uv}^+ \neq \emptyset\}.$$

A *distância de compatibilidade* de um par $(u, v) \in Comp$ é o tamanho (quantidade de arestas) do menor caminho positivo entre u e v , denotada por $\delta^+(u, v)$. Em particular, essa distância é 1 se $(u, v) \in E^+$. Note que $\delta^+(u, v) < +\infty$ para todo $(u, v) \in Comp$.

Utilizando uma ideia semelhante, iremos definir a relação de *compatibilidade direcionada* $\overrightarrow{Comp} \subseteq V \times V$, quando o grafo de sinais G é direcionado, onde $(u, v) \in \overrightarrow{Comp}$ se, e somente se, u e v puderem trabalhar juntos em uma equipe. Para maior diferenciação, vamos usar $G = (V, A)$ no caso direcionado, ou seja, A representa o conjunto de arcos de G . Similarmente, $A^+(G)$ e $A^-(G)$ são os subconjuntos de arcos positivos e negativos, respectivamente.

Além da reflexividade e simetria, as seguintes propriedades devem ser respeitadas em $\overrightarrow{Comp}$. No caso em que existe algum arco entre o par de vértices, temos:

1. *Compatibilidade por apenas arco positivo*: $(u, v) \in \overrightarrow{Comp}$, para todo $(u, v) \in A^+(G)$ e $(v, u) \notin A^-(G)$, ou $(v, u) \in A^+(G)$ e $(u, v) \notin A^-(G)$.
2. *Incompatibilidade por arco negativo*: $(u, v) \notin \overrightarrow{Comp}$, para todo $(u, v) \in V \times V$ tal que $(u, v) \in A^-(G)$ ou $(v, u) \in A^-(G)$.

Quando não existe qualquer arco entre o par de vértices, vamos definir a compatibilidade por:

3. *Compatibilidade por algum caminho direcionado positivo*: $(u, v) \in \overrightarrow{Comp}$, para todo $(u, v) \in V \times V$ tal que $(u, v) \notin A$, $(v, u) \notin A$, e $\mathcal{P}_{uv}^+ \cup \mathcal{P}_{vu}^+ \neq \emptyset$.
4. *Incompatibilidade por nenhum caminho direcionado positivo*: $(u, v) \notin \overrightarrow{Comp}$, para todo $(u, v) \in V \times V$ tal que $(u, v) \notin A$, $(v, u) \notin A$, e $\mathcal{P}_{uv}^+ \cup \mathcal{P}_{vu}^+ = \emptyset$.

Em resumo, quando G é um digrafo, definimos

$$\overrightarrow{Comp} = \{(u, v) \in V \times V : (u, v) \notin A^-(G), (v, u) \notin A^-(G) \text{ e } \mathcal{P}_{uv}^+ \cup \mathcal{P}_{vu}^+ \neq \emptyset\}.$$

A *distância de compatibilidade direcionada* de um par $(u, v) \in \overrightarrow{Comp}$ é o tamanho (quantidade de arcos) do menor caminho positivo de u até v ou de v até u , ou seja, $\min\{\delta^+(u, v), \delta^+(v, u)\}$. Para qualquer $(u, v) \in \overrightarrow{Comp}$, essa distância será menor que $+\infty$ e, em particular, terá valor 1 quando $(u, v) \in A^+(G)$ ou $(v, u) \in A^+(G)$.

Dessa forma, utilizando o grafo de sinais G , que representa os trabalhadores e suas relações diretas, junto com as definições acima, podemos agora definir o Problema de Formação de Equipes com Caminhos Positivos Mínimos - TFP_SPP. Sejam S um conjunto de habilidades, $\bar{s} : V(G) \rightarrow 2^S$ uma função que retorna o conjunto de habilidades em S que cada trabalhador possui, T um conjunto de equipes requeridas e $t : T \times S \rightarrow \mathbb{N}$ uma função que retorna a demanda de trabalhadores para cada tipo de habilidade em cada equipe.

O TFP_SPP consiste em encontrar $|T|$ equipes, respeitando as seguintes condições:

1. cada trabalhador é alocado em no máximo uma equipe utilizando apenas uma de suas habilidades;
2. em cada equipe $j \in T$ há pelo menos $t(j, s)$ trabalhadores alocados com a habilidade $s \in S$;
3. existe *compatibilidade por um caminho (direcionado) positivo* entre cada par de trabalhadores pertencentes a uma mesma equipe, ou seja, se u e v estão na mesma equipe então $(u, v) \in \text{Comp}$ ou $(u, v) \in \overrightarrow{\text{Comp}}$, se G é não direcionado ou direcionado, respectivamente.

O objetivo é minimizar a soma das distâncias de compatibilidade entre cada par de trabalhadores de uma mesma equipe.

No restante do texto, será conveniente uma notação adicional. Para todos os indivíduos $u \in V$ e equipes $j \in T$, vamos usar

$$\bar{s}(j) = \{s \in S : t(j, s) > 0\} \quad \text{e} \quad \bar{s}(u, j) = \bar{s}(u) \cap \bar{s}(j)$$

para expressar, respectivamente, o conjunto de habilidades requeridas em uma equipe j e, dentre essas, aquelas que podem ser desempenhadas pelo indivíduo u . Observe que $\bar{s}(j)$ com $j \in T$ e $\bar{s}(u)$ com $u \in V$ possuem significados diferentes, que ficarão sempre claros pelo contexto. Os conjuntos

$$V_j = \{u \in V : \bar{s}(u, j) \neq \emptyset\} \quad \text{e} \quad V_T = \bigcup_{j \in T} V_j$$

denotarão, respectivamente, os indivíduos que possuem pelo menos uma habilidade requerida na equipe $j \in T$ e os indivíduos com habilidade demandada em alguma equipe. Note que um indivíduo em $V \setminus V_T$ pode definir a compatibilidade de dois outros, mesmo que não seja alocado em qualquer equipe.

4.2 Métodos para o TFP_SPP Não Direcionado

Na descrição dos modelos de PLI para o TFP_SPP em grafos não direcionados, usaremos o conjunto

$$\bar{E} = \{(u, v) \in V \times V : (u, v) \notin E, u \neq v\}$$

para indexar os pares de vértices distintos não adjacentes em $G = (V, E)$. Os pares em E e $\bar{E}$ induzidos por V_j são

$$E_j = \{(u, v) \in E : u, v \in V_j\} \quad \text{e} \quad \bar{E}_j = \{(u, v) \in \bar{E} : u, v \in V_j\},$$

representando respectivamente os pares de indivíduos ligados ou não por uma aresta, dentre aqueles que compartilham alguma habilidade requerida na equipe $j \in T$. Similarmente, temos

$$E_T = \bigcup_{j \in T} E_j \quad \text{e} \quad \bar{E}_T = \bigcup_{j \in T} \bar{E}_j.$$

Além disso, no caso não direcionado, $A = \{(p, q), (q, p) : (p, q) \in E\}$ representará os arcos da orientação simétrica de G , enquanto $A^- \subseteq A$ e $A^+ \subseteq A$ serão aqueles obtidos a partir das arestas negativas e positivas.

4.2.1 Formulação Matemática

Para a descrição do modelo matemático, definimos as seguintes variáveis de decisão:

$$x_{us}^j = \begin{cases} 1, & \text{se o indivíduo } u \text{ está na equipe } j \text{ com habilidade } s, \\ 0, & \text{caso contrário,} \end{cases} \quad \forall j \in T, \forall u \in V_j, \forall s \in \bar{s}(u, j).$$

$$y_{uv}^j = \begin{cases} 1, & \text{se } u \text{ e } v \text{ estão na mesma equipe } j, \\ 0, & \text{caso contrário,} \end{cases} \quad \forall j \in T, \forall (u, v) \in E_j \cup \bar{E}_j$$

$$f_{pq}^{uv} = \begin{cases} 1, & \text{se o arco } (p, q) \text{ pertence ao } u, v\text{-caminho,} \\ 0, & \text{caso contrário,} \end{cases} \quad \forall u, v \in \bar{E}_T, \forall (p, q) \in A.$$

$$\pi_p^{uv} \geq 0 : \text{distância entre } u \text{ e } p \text{ no } u, v\text{-caminho, caso } p \text{ pertença ao caminho} \quad \forall u, v \in \bar{E}_T, \forall p \in V$$

$$\lambda_{uv} \in \mathbb{N} : \text{metade no número de arestas negativas no } u, v\text{-caminho} \quad \forall u, v \in \bar{E}_T$$

Podemos observar que as variáveis y_{uv}^j e y_{vu}^j são, na verdade, a mesma variável. Por isso, sem perda de generalidade, podemos admitir sempre $u < v$ em y_{uv}^j . Entretanto, por facilidade de escrita das expressões, vamos usar indistintamente as duas variáveis, ficando subjacente que são iguais. Comentário similar vale para os índices (u, v) das variáveis f_{pq}^{uv} , π_p^{uv} e λ_{uv} .

O **Modelo- $\mathcal{F}_{MTZ}$** de PLI para TFP_SPP é descrito como:

$$(\mathcal{F}_{MTZ}) \quad \min \sum_{(u,v) \in \bar{E}_T} \sum_{(p,q) \in A} f_{pq}^{uv} + \sum_{(u,v) \in E^+} \sum_{j \in T} y_{uv}^j \quad (4.1)$$

$$\text{s.a.} \quad \sum_{j \in T} \sum_{s \in \bar{s}(u,j)} x_{us}^j \leq 1, \quad \forall u \in V_T \quad (4.2)$$

$$\sum_{u \in V_j : s \in \bar{s}(u)} x_{us}^j = t(j, s), \quad \forall j \in T, \forall s \in \bar{s}(j) \quad (4.3)$$

$$\sum_{j \in T} y_{uv}^j = 0 \quad \forall (u, v) \in E_T \cap E^- \quad (4.4)$$

$$\sum_{s \in \bar{s}(u,j)} x_{us}^j + \sum_{s \in \bar{s}(v,j)} x_{vs}^j - y_{uv}^j \leq 1, \quad \forall j \in T, \forall (u, v) \in E_j \cup \bar{E}_j \quad (4.5)$$

$$y_{uv}^j \leq \sum_{s \in \bar{s}(u,j)} x_{us}^j, \quad \forall j \in T, \forall (u, v) \in E_j \cup \bar{E}_j \quad (4.6)$$

$$y_{uv}^j \leq \sum_{s \in \bar{s}(v,j)} x_{vs}^j, \quad \forall j \in T, \forall (u, v) \in E_j \cup \bar{E}_j \quad (4.7)$$

$$\sum_{p \in V: (p,q) \in A} f_{pq}^{uv} - \sum_{p \in V: (q,p) \in A} f_{qp}^{uv} = \begin{cases} -\sum_{j \in T} y_{uv}^j, & \text{se } q = u, \\ \sum_{j \in T} y_{uv}^j, & \text{se } q = v, \\ 0, & \text{c.c.} \end{cases} \quad \forall (u,v) \in \bar{E}_T, \forall q \in V \quad (4.8)$$

$$f_{pq}^{uv} \leq \sum_{j \in T} y_{uv}^j, \quad \forall (u,v) \in \bar{E}_T, \forall (p,q) \in A \quad (4.9)$$

$$\sum_{(p,q) \in A^-} f_{pq}^{uv} - 2\lambda_{uv} = 0, \quad \forall (u,v) \in \bar{E}_T \quad (4.10)$$

$$\pi_p^{uv} - \pi_q^{uv} + |V| \cdot f_{pq}^{uv} \leq (|V| - 1), \quad \forall (u,v) \in \bar{E}_T, \forall (p,q) \in A \quad (4.11)$$

$$x_{us}^j \in \{0, 1\}, \quad \forall j \in T, \forall u \in V_j, \forall s \in \bar{s}(u, j) \quad (4.12)$$

$$y_{uv}^j \in \{0, 1\}, \quad \forall j \in T, \forall (u,v) \in E_j \cup \bar{E}_j \quad (4.13)$$

$$f_{pq}^{uv} \in \{0, 1\}, \quad \forall (u,v) \in \bar{E}_T, \forall (p,q) \in A \quad (4.14)$$

$$\pi_p^{uv} \geq 0, \quad \forall (u,v) \in \bar{E}_T, \forall p \in V \quad (4.15)$$

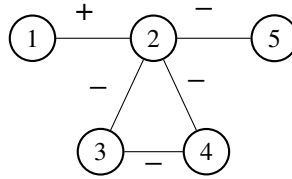
$$\lambda_{uv} \in \mathbb{N}^+, \quad \forall (u,v) \in \bar{E}_T \quad (4.16)$$

As restrições (4.2) garantem que cada indivíduo pertença a no máximo uma equipe, exercendo uma única habilidade, enquanto as restrições (4.3) asseguram que as equipes serão criadas com suas respectivas demandas de habilidades sendo atendidas. As restrições (4.5), (4.6) e (4.7) certificam que a variável y_{uv}^j tenha o valor igual a um se os indivíduos u e v estiverem na mesma equipe j e valor igual a zero caso contrário. Para manter a propriedade da *incompatibilidade por aresta negativa*, a restrição (4.4) não permite que dois indivíduos ligados por aresta negativa estejam na mesma equipe. As restrições (4.8) e (4.9) estabelecem que um fluxo de valor 1 deve ser enviado de u a v (com $u < v$ e $(u,v) \in \bar{E}$), se u e v estiverem numa mesma equipe j , ou seja, $\sum_{j \in T} y_{uv}^j = 1$. Para garantir que os arcos por onde passam esse fluxo definam precisamente um caminho, são usadas as restrições de quebra de ciclo (4.11), a serem discutidas a abaixo. As restrições (4.10) asseguram que tal caminho seja positivo, isto é, possua um número par de arestas negativas. As restrições (4.12)-(4.16) definem o domínio correto das variáveis.

As restrições (4.11) são as clássicas desigualdades MTZ para quebra de ciclos (MILLER *et al.*, 1960). Note que, se passa fluxo em um arco (p,q) na formação do uv -caminho, o rótulo do vértice q (π_q^{uv}) é pelo menos uma unidade maior que o rótulo do vértice p (π_p^{uv}), proibindo a formação de ciclos. Observamos que tais restrições são de fato necessárias quando o grafo de entrada é arbitrário. Como exemplo, considere o grafo da Figura 5. Note que o passeio $\langle 1, 2, 3, 4, 2, 5 \rangle$ define um $(1, 5)$ -fluxo que satisfaz as restrições (4.8) e (4.10). Entretanto, o caminho elementar $\langle 1, 2 \rangle$ que se extrai desse passeio não

é positivo. Na verdade, 1 e 5 não poderiam figurar na mesma equipe pois $(1,5) \notin E$ e não existe caminho positivo entre 1 e 5. Por sua vez, a inclusão das restrições (4.11) proíbe esse fluxo, pois exclui o ciclo $\langle 2,3,4,2 \rangle$. Vale observar que essas restrições são, porém, desnecessárias quando o grafo de entrada é balanceado, isto é, não possui ciclo com quantidade ímpar de arestas negativas (HARARY, 1953). Isto porque (4.8)–(4.10) para $(u,v) \notin E$ definem a união de um uv -caminho com ciclos (envolvendo ou não os vértices desse caminho). E como tais ciclos não afetam a compatibilidade do caminho, a função objetivo garante que eles não serão formados.

Figura 5 – Exemplo da necessidade da restrição de quebra de ciclo



Fonte: elaborado pelo autor.

Por fim, a função objetivo (4.1) minimiza o tamanho dos caminhos positivos para cada par de vértice $u, v \in V$ pertencentes a uma mesma equipe. A primeira parcela da expressão soma as distâncias de compatibilidade para cada par $(u, v) \in \bar{E}$ de uma mesma equipe, por meio do u, v -caminho definido pelas variáveis f^{uv} . Já a segunda parcela, computa diretamente esse valor quando $(u, v) \in E^+$, usando as variáveis y_{uv}^j .

Perceba que, semelhante ao que foi apresentado na seção 3.5, podemos substituir as variáveis π_p^{uv} pelas variáveis

$$\mu_p^{uv} = \begin{cases} 1, & \text{se o subcaminho de } u \text{ a } p \text{ no } u, v\text{-caminho é positivo,} \\ 0, & \text{caso contrário,} \end{cases} \quad \forall (u, v) \in \bar{E}_T, \forall p \in V,$$

e as restrições (4.10), (4.11), (4.15) e (4.16) pelas restrições

$$\mu_q^{uv} \geq f_{pq}^{uv} - \mu_p^{uv} \quad \forall (u, v) \in \bar{E}_T, (p, q) \in A^- \quad (4.17)$$

$$\mu_q^{uv} \leq 2 - f_{pq}^{uv} - \mu_p^{uv} \quad \forall (u, v) \in \bar{E}_T, (p, q) \in A^- \quad (4.18)$$

$$\mu_q^{uv} \geq f_{pq}^{uv} + \mu_p^{uv} - 1 \quad \forall (u, v) \in \bar{E}_T, (p, q) \in A^+ \quad (4.19)$$

$$\mu_q^{uv} \leq 1 - f_{pq}^{uv} + \mu_p^{uv} \quad \forall (u, v) \in \bar{E}_T, (p, q) \in A^+ \quad (4.20)$$

$$\mu_u^{uv} = \mu_v^{uv} = 1 \quad \forall (u, v) \in \bar{E}_T \quad (4.21)$$

$$\mu_p^{uv} \in \{0, 1\}, \quad \forall (u, v) \in \bar{E}_T, \forall p \in V \quad (4.22)$$

Chamaremos de **Modelo- $\mathcal{F}_{SIGN}$** aquele obtido após a substituição das restrições de quebra de ciclo ((4.11) e (4.15)) e caminho positivo ((4.10) e (4.16)) pelas restrições (4.17) - (4.22). A justificativa de corretude é a mesma apresentada na Seção 3.5 para a formulação $\mathcal{SP}_{SIGN}$.

De forma análoga à argumentação feita no final da Seção 3.5, podemos concluir que a integralidade das variáveis f poderia ser descartada no **Modelo- $\mathcal{F}_{SIGN}$** .

4.2.2 Decompondo a Formulação

Podemos notar que o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** possuem um grande número de variáveis e restrições. Com intuito de melhorar seu desempenho, nas próximas subseções é apresentada uma decomposição para o **Modelo- $\mathcal{F}_{MTZ}$** , particionando-o em dois problemas independentes.

Para entender que o **Modelo- $\mathcal{F}_{MTZ}$** é decomponível, observe primeiramente que as restrições (4.8)-(4.11) são separáveis para cada $(u, v) \in \bar{E}_T$. Além disso, considere uma variável binária $w_{uv} = \sum_{j \in T} y_{uv}^j$, de modo que $w_{uv} = 1$, se u e v estiverem na mesma equipe, e $w_{uv} = 0$, caso contrário. Como a função objetivo (4.1) contabiliza os caminhos positivos entre pares de trabalhadores apenas se estes estão em uma mesma equipe, ela pode ser reescrita como a seguinte função objetivo quadrática equivalente:

$$\min \sum_{(u,v) \in \bar{E}_T} w_{uv} \sum_{(p,q) \in A} f_{pq}^{uv} + \sum_{(u,v) \in E^+} w_{uv} \quad (4.23)$$

Agora, com a substituição da expressão $\sum_{j \in T} y_{uv}^j$, que aparece nas restrições (4.8) e (4.9), pela variável w_{uv} , pode-se perceber que as restrições (4.8)-(4.11), além de separáveis para cada $(u, v) \in \bar{E}_T$, também podem ser desacopladas do **Modelo- $\mathcal{F}_{MTZ}$** , considerando os dois possíveis valores para w_{uv} . De fato, considere um par de vértices $(u, v) \in \bar{E}_T$. Se $w_{uv} = 0$, então $f_{pq}^{uv} = 0$, para todo $(p, q) \in A$, por (4.9). Se $w_{uv} = 1$, pela função objetivo (4.23) e restrições (4.8)-(4.11) e (4.14)-(4.16), temos que os valores de f_{pq}^{uv} definem um u, v -caminho positivo mínimo. Em qualquer caso, o somatório $\sum_{(p,q) \in A} f_{pq}^{uv}$ em (4.23) pode ser substituído pelo tamanho de um uv -caminho positivo mínimo.

Em outros termos, a função objetivo (4.23) pode ser reescrita nas variáveis y como:

$$\min \sum_{(u,v) \in \bar{E}_T \cup E^+} \delta^+(u, v) \sum_{j \in T} y_{uv}^j. \quad (4.24)$$

Com isso, as restrições (4.8)-(4.11) e (4.14)-(4.16), que envolvem as variáveis f , π e λ , tornam-se desnecessárias na formulação. Dessa forma, o problema pode ser decomposto em dois: o primeiro, para determinar caminhos positivos mínimos entre pares de vértices, e o principal, para alocar os indivíduos nas equipes.

Perceba que argumento semelhante pode ser considerado no **Modelo- $\mathcal{F}_{SIGN}$** .

4.2.3 Cálculo dos Caminhos Positivos Mínimos

Como visto, a formulação matemática proposta para o TFP_SPP pode ser simplificada, a partir do cálculo preliminar de caminhos mínimos positivos entre os pares de vértices $(u, v) \in \bar{E}_T$. Estudamos esse problema no Capítulo 3. Em particular, para grafos de sinais não direcionados foi apresentada uma transformação para o problema do Emparelhamento Perfeito de Custo Mínimo (em inglês, Minimum Weight Perfect Matching Problem (MWPMP)) na Subseção 3.3. Sendo assim, para determinar os valores $\delta^+(u, v)$, para cada $(u, v) \in \bar{E}_T$, usados na reescrita (4.24) da função objetivo, aplicaremos tal transformação no grafo de entrada G e resolveremos o problema MWPMP no grafo resultante, usando o algoritmo polinomial apresentado em (GERARDS, 1995), que busca caminhos aumentantes a partir de floresta húngara. Denotaremos por $MinMatch(u, v)$ este algoritmo aplicado à transformação referente à (u, v) . Então temos:

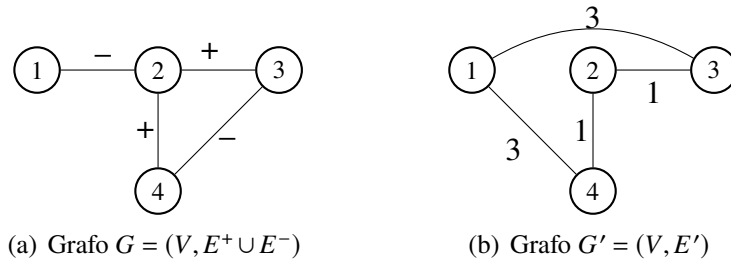
$$\delta^+(u, v) = \begin{cases} 1 & \text{se } (u, v) \in E^+, \\ +\infty & \text{se } (u, v) \in E^-, \\ \text{solução de } MinMatch(u, v) & \text{se } (u, v) \in \bar{E}_T, \end{cases}$$

onde a solução de $MinMatch(u, v)$ é $+\infty$ caso não exista um emparelhamento perfeito.

4.2.4 Modelo Simplificado

Para a descrição do modelo simplificado, mencionado na Seção 4.2.2, vamos gerar o *grafo de compatibilidade* $G' = (V_T, E')$, não direcionado e ponderado nas arestas, onde o conjunto de arestas é $E' = E^+(G) \cup \{(u, v) \in \bar{E}_T : \delta^+(u, v) < +\infty\}$ e peso da aresta (u, v) é a $\delta^+(u, v)$. Note que os vértices de G' compreendem os indivíduos com alguma habilidade demandada nas equipes, e que as arestas de G' representam pares de indivíduos que poderiam figurar numa mesma equipe por estarem ligados por algum caminho positivo em G . A Figura 6 ilustra a obtenção de G' a partir de um grafo de sinais G , onde admitimos $V(G) = V_T$.

Figura 6 – Exemplo de obtenção do grafo de compatibilidade G' a partir do grafo de sinais G .



Fonte: elaborado pelo autor.

Dados a tupla $(S, T, \bar{s}, t)$, como definida na Seção 4.1, e o grafo de compatibilidade $G' =$

(V_T, E') com função de pesos $\delta^+ : E' \rightarrow \mathbb{N}$, conforme apresentado acima, seja

$$E'_j = \{(u, v) \in E' : u, v \in V_j\}$$

o conjunto de pares de vértices em que ambos podem figurar na equipe j . Uma solução ótima para TFP_SPP pode ser obtida com o **Modelo- $\mathcal{TF}$** , expresso a seguir, essencialmente obtido do **Modelo- $\mathcal{F}_{MTZ}$** com a exclusão das restrições (4.8)-(4.11) e (4.14)-(4.16).

$$(TF) \quad \min \sum_{(u,v) \in E'} \sum_{j \in T} \delta^+(u, v) \cdot y_{uv}^j \quad (4.25)$$

$$\text{s.a.} \quad \sum_{j \in T} \sum_{s \in \bar{s}(u, j)} x_{us}^j \leq 1, \quad \forall u \in V_T \quad (4.26)$$

$$\sum_{u \in V_j : s \in \bar{s}(u)} x_{us}^j = t(j, s), \quad \forall j \in T, \forall s \in \bar{s}(j) \quad (4.27)$$

$$\sum_{s \in \bar{s}(u, j)} x_{us}^j + \sum_{s \in \bar{s}(v, j)} x_{vs}^j - y_{uv}^j \leq 1, \quad \forall j \in T, \forall (u, v) \in E'_j \quad (4.28)$$

$$\sum_{s \in \bar{s}(u, j)} x_{us}^j + \sum_{s \in \bar{s}(v, j)} x_{vs}^j \leq 1, \quad \forall j \in T, \forall u, v \in V_j : (u, v) \notin E'_j \quad (4.29)$$

$$y_{uv}^j \leq \sum_{s \in \bar{s}(u, j)} x_{us}^j, \quad \forall j \in T, \forall (u, v) \in E'_j \quad (4.30)$$

$$y_{uv}^j \leq \sum_{s \in \bar{s}(v, j)} x_{vs}^j, \quad \forall j \in T, \forall (u, v) \in E'_j \quad (4.31)$$

$$x_{us}^j \in \{0, 1\}, \quad \forall j \in T, \forall u \in V_j, \forall s \in \bar{s}(u, j) \quad (4.32)$$

$$y_{uv}^j \geq 0, \quad \forall j \in T, \forall (u, v) \in E'_j \quad (4.33)$$

Pela simplificação, neste modelo, a indexação das variáveis y^j passa a ser por meio das arestas em E'_j , e não mais pelos pares em $E_j \cup \bar{E}_j$. Entretanto, ainda usamos indistintamente as variáveis y_{uv}^j e y_{vu}^j , que são iguais. Além disso, descartamos a integralidade dessas variáveis.

4.3 Métodos para o TFP_SPP Direcionado

Agora, focaremos no problema TFP_SPP em digrafos de sinais. Primeiro, observamos que o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** podem ser facilmente adaptados para este caso. Discutiremos, porém, apenas, as versões simplificadas destes modelos. Semelhante ao que foi feito para o caso não direcionado, podemos calcular previamente os caminhos positivos mínimos no digrafo para compor a função objetivo dos modelos simplificados que alocam os indivíduos nas equipes. Dado que o problema de decidir se existe caminho positivo entre um par de vértices em um grafo direcionado é NP-Completo, Corolário 3.4.1.

A diferença é que agora tais caminhos serão determinados por métodos de PLI, baseados nos modelos $\mathcal{SP}_{MTZ}$ e $\mathcal{SP}_{SIGN}$, apresentados na Subseção 3.5.

Ao longo desta seção, consideraremos um digrafo de sinais $G = (V, A = A^+ \cup A^-)$ e denotaremos os pares de vértices não relacionados em G por

$$\bar{A} = \{(u, v) \in V \times V : (u, v) \notin A, (v, u) \notin A, u \neq v\}.$$

4.3.1 Cálculo dos Caminhos Direcionados Positivos Mínimos

No Capítulo 3, mais especificamente, nas Seções 3.4 e 3.5, estudamos o problema do u, v -caminho positivo mínimo em grafos direcionados. Como esse problema pertence à classe NP-Difícil, iremos usar os modelos de PLI $\mathcal{SP}_{MTZ}$ e $\mathcal{SP}_{SIGN}$ para resolver o problema para todo par de vértices $(u, v) \in \bar{A}$. Usaremos indistintamente $\mathcal{SP}(u, v)$ para nos referirmos ao valor da solução de qualquer destes modelos para o caminho positivo mínimo entre os vértices u e v . Este valor é $+\infty$ caso os modelos sejam inviáveis, ou seja, não exista u, v -caminho positivo.

Desse modo, a distância de compatibilidade associada à relação $\overrightarrow{Comp}$ é dada por:

$$\vec{\delta}^+(u, v) = \begin{cases} 1 & \text{se } (u, v) \in A^+(G) \text{ e } (v, u) \notin A^-(G), \\ 1 & \text{se } (v, u) \in A^+(G) \text{ e } (u, v) \notin A^-(G), \\ +\infty & \text{se } (u, v) \in A^-(G) \text{ ou } (v, u) \in A^-(G), \\ \inf\{\mathcal{SP}(u, v), \mathcal{SP}(v, u)\} & \text{se } (u, v) \in \bar{A} \end{cases}$$

Da mesma forma que na Seção 4.2.3, criamos o grafo de compatibilidade $G' = (V_T, E')$, ponderado em arestas, com o conjunto de vértices sendo $V(G') = V_T$, conjunto de arestas $E' = \{(u, v) : \vec{\delta}^+(u, v) < +\infty\}$ e peso da aresta (u, v) igual ao valor de $\vec{\delta}^+(u, v)$.

4.3.2 Modelo Simplificado

Dados a tupla $(S, T, \bar{s}, t)$, como definida na Seção 4.1, e o grafo de compatibilidade $G' = (V_T, E')$, conforme apresentado na Subseção 4.3.1, podemos usar o mesmo **Modelo- $\mathcal{TF}$** , da Seção 4.2.4, para resolver o problema TFP_SPP *direcionado*. A única diferença são os coeficientes da função objetivo, agora dados pelos valores $\vec{\delta}^+(u, v)$, calculados pelos modelos $\mathcal{SP}_{MTZ}$ ou $\mathcal{SP}_{SIGN}$.

Perceba que os coeficientes da função objetivo do modelo **Modelo- $\mathcal{TF}$** estão diretamente relacionados à forma com definimos a relação de compatibilidade $\overrightarrow{Comp}$. Para algumas definições alternativas de *compatibilidade direcionada* podemos apenas mudar a definição dos valores de $\vec{\delta}^+(u, v)$. Por exemplo, poderíamos considerar uma relação mais restritiva, exigindo a existência simultânea de u, v -caminho positivo e v, u -caminho positivo para que o par (u, v) fosse compatível.

4.4 Desigualdades Válidas para o Modelo- $T\mathcal{F}$

Nesta seção, apresentaremos algumas desigualdades válidas buscando fortalecer o **Modelo- $T\mathcal{F}$** , que se aplicam, portanto, para os casos não direcionado e direcionado. Elas são definidas a partir do grafo de compatibilidade $G' = (V_T, E')$.

Sejam $S' \subseteq S$ e $j \in T$. Dizemos que um par $(u, v) \in V \times V$ é (j, S') -associável se as seguintes condições são satisfeitas:

- i) $(u, v) \in E'_j$; e
- ii) $\bar{s}(u, j) \cap S' \neq \emptyset$ e $\bar{s}(v, j) \cap S' \neq \emptyset$; e
- iii) se $\bar{s}(u, j) \cap S' = \bar{s}(v, j) \cap S' = \{s\}$, então $t(j, s) \geq 2$.

Denotamos por $\mathcal{A}(j, S')$ o conjunto de todos os pares (j, S') -associáveis e por $\mathcal{A}_u(j, S')$ o conjunto dos vértices v tais que (u, v) é (j, S') -associável. Seja $t(j, S') = \sum_{s \in \bar{s}(j) \cap S'} t(j, s)$.

A partir da definição acima podemos perceber que um par de indivíduos (u, v) é (j, S') -associável se, e somente se, ambos podem ser alocados na equipe j com alguma habilidade de S' . Em particular, $\mathcal{A}(j, S)$ compreende os pares que podem figurar simultaneamente na equipe $j \in T$. Isto nos leva à seguinte condição para fixação de variáveis y :

Proposição 4.4.1. *Para todos $j \in T$ e $(u, v) \in E'_j \setminus \mathcal{A}(j, S)$, podemos fixar $y_{uv}^j = 0$ em toda solução viável.*

Demonstração. Sejam $j \in T$ e $(u, v) \in E'_j \setminus \mathcal{A}(j, S)$. Então devemos ter $\bar{s}(u, j) = \bar{s}(v, j) = \{s\}$ e $t(j, s) \leq 1$. A restrição (4.27) implica que $x_{us}^j + x_{vs}^j \leq 1$, ou melhor, $x_{us}^j = 0$ ou $x_{vs}^j = 0$. Logo, (4.30)-(4.31) garantem $y_{uv}^j = 0$. $\square$

Além disso, concluímos que a equipe j é uma clique no seguinte subgrafo do grafo de compatibilidade G' :

$$G'_j = (V_j, \mathcal{A}(j, S)).$$

Isso implica que no máximo um elemento de qualquer conjunto independente de G'_j pode fazer parte da equipe j , ou melhor, que

$$\sum_{u \in I} \sum_{s \in \bar{s}(u, j)} x_{us}^j \leq 1, \quad \forall j \in T, \forall I \in \mathcal{I}_j, \quad (4.34)$$

onde $\mathcal{I}_j$ é o conjunto dos conjuntos independentes em G'_j . Essas desigualdades generalizam as restrições (4.29).

Proposição 4.4.2. *As desigualdades (4.34) são válidas.*

Demonstração. Sejam $j \in T$ e $I \in \mathcal{I}_j$. Suponha que a desigualdade correspondente em (4.34) não é válida. Por (4.26), existem $u, v \in I$, $u \neq v$, tais que $\sum_{s \in \bar{s}(u, j)} x_{us}^j = 1$ e $\sum_{s \in \bar{s}(v, j)} x_{vs}^j = 1$. Como $(u, v) \notin \mathcal{A}(j, S)$,

então $(u, v) \notin E'_j$ ou $y_{uv}^j = 0$ (pela Proposição 4.4.1). Assim, temos que (4.29) ou (4.28) estaria violada: um absurdo. $\square$

Além disso, perceba que, apesar da variável y_{uv}^j do **Modelo- $\mathcal{TF}$** ser contínua, sempre lhe será atribuído no ótimo um valor binário, pelas restrições (4.28), (4.30) e (4.31). Dessa maneira, usando as restrições (4.26) e (4.27) podemos substituir as restrições (4.30) e (4.31) por:

$$\sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j = (t(j, S) - 1) \sum_{s \in \bar{s}(u, j)} x_{us}^j, \quad \forall j \in T, \forall u \in V_j \quad (4.35)$$

E ainda acrescentar:

$$\sum_{(u, v) \in \mathcal{A}(j, S)} y_{uv}^j = t(j, S)(t(j, S) - 1)/2 \quad \forall j \in T. \quad (4.36)$$

Proposição 4.4.3. *As igualdades (4.35) e (4.36) são válidas.*

Demonstração. Sejam $j \in T$ e $u \in V_j$. Primeiro, suponha que $\sum_{s \in \bar{s}(u, j)} x_{us}^j = 0$. Por (4.30), concluímos que $y_{uv}^j = 0$, para todo $v \in \mathcal{A}_u(j, S)$. Logo, (4.35) é válida neste caso. Admita agora $\sum_{s \in \bar{s}(u, j)} x_{us}^j = 1$, que corresponde ao caso complementar por (4.26). Devido à soma das equações (4.27) em $s \in \bar{s}(j)$, chegamos a

$$\sum_{v \in V_j} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = \sum_{s \in \bar{s}(j)} \sum_{v \in V_j: s \in \bar{s}(v)} x_{vs}^j = \sum_{s \in \bar{s}(j)} t(j, s) = t(j, S),$$

ou melhor,

$$\sum_{v \in V_j \setminus \{u\}} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = t(j, S) - 1 \quad (4.37)$$

Somando (4.31) para $v \in \mathcal{A}_u(j, S)$ e usando a expressão acima, temos que

$$\sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j \leq \sum_{v \in \mathcal{A}_u(j, S)} \sum_{s \in \bar{s}(v, j)} x_{vs}^j \leq \sum_{v \in V_j \setminus \{u\}} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = t(j, S) - 1.$$

Por outro lado, (4.28) se reduz às desigualdades

$$y_{uv}^j \geq \sum_{s \in \bar{s}(v, j)} x_{vs}^j \quad \forall (u, v) \in E'_j,$$

que somadas levam a

$$\sum_{v: (u, v) \in E'_j} y_{uv}^j \geq \sum_{v: (u, v) \in E'_j} \sum_{s \in \bar{s}(v, j)} x_{vs}^j$$

Adicionalmente, pela Proposição 4.4.1 e por (4.29), chegamos respectivamente a

$$\sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j = \sum_{v: (u, v) \in E'_j} y_{uv}^j \quad \text{e} \quad \sum_{v: (u, v) \notin E'_j} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = 0.$$

Juntando as três últimas expressões e (4.37), concluímos que

$$\sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j = \sum_{v: (u, v) \in E'_j} y_{uv}^j \geq \sum_{v: (u, v) \in E'_j} \sum_{s \in \bar{s}(v, j)} x_{vs}^j + \sum_{v: (u, v) \notin E'_j} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = \sum_{v \in V_j \setminus \{u\}} \sum_{s \in \bar{s}(v, j)} x_{vs}^j = t(j, S) - 1.$$

Portanto, (4.35) também é válida quando $\sum_{s \in \bar{s}(u, j)} x_{us}^j = 1$.

Finalmente, somando (4.35) em $u \in V_j$, derivamos

$$\sum_{(u, v) \in \mathcal{A}(j, S)} y_{uv}^j = \frac{1}{2} \sum_{u \in V_j} \sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j = \frac{t(j, S) - 1}{2} \sum_{u \in V_j} \sum_{s \in \bar{s}(u, j)} x_{us}^j = \frac{t(j, S)(t(j, S) - 1)}{2}$$

□

Corolário 4.4.1. *Sejam $j \in T$ e $u \in V_j$. Se $\mathcal{A}_u(j, S) = \emptyset$ e $t(j, s) > 1$. Então $\sum_{s \in \bar{s}(u, j)} x_{us}^j = 0$. Se $t(j, S) < 2$, então $\sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j = 0$.*

Ainda utilizando o conjunto de vértices $\mathcal{A}_u(j, S')$, defina

$$\Phi(u, j, S') := \sum_{v \in \mathcal{A}_u(j, S): \bar{s}(v, j) \cap S' \neq \emptyset} y_{uv}^j - (t(j, S') - 1) \sum_{s \in \bar{s}(u, j) \cap S'} x_{us}^j - t(j, S') \sum_{s \in \bar{s}(u, j) \setminus S'} x_{us}^j. \quad (4.38)$$

Em particular quando $S' = S$ ficamos com a seguinte expressão:

$$\Phi(u, j, S) := \sum_{v \in \mathcal{A}_u(j, S)} y_{uv}^j - (t(j, S) - 1) \sum_{s \in \bar{s}(u, j)} x_{us}^j. \quad (4.39)$$

Em outro caso especial, $S' = \{s'\}$ com $s' \in \bar{s}(u, j)$, a expressão fica simplificada como:

$$\Phi(u, j, s') := \sum_{v \in \mathcal{B}_u(j, s')} y_{uv}^j - (t(j, s') - 1) x_{us'}^j - t(j, s') \sum_{s \in \bar{s}(u, j), s \neq s'} x_{us}^j, \quad (4.40)$$

onde

$$\mathcal{B}_u(j, s') = \{v \in \mathcal{A}_u(j, S) : s' \in \bar{s}(v, j)\}$$

$$= \{v : (u, v) \in E'_j, s' \in \bar{s}(u, j) \cap \bar{s}(v, j), \text{ e } \bar{s}(u, j) \cup \bar{s}(v, j) \neq \{s'\} \text{ ou } t(j, s') \geq 2\}$$

Proposição 4.4.4. *Para todos $j \in T$, $u \in V_j$ e $S' \subseteq S$, a desigualdade $\Phi(u, j, S') \geq 0$ é válida.*

Demonstração. Sejam $j \in T$ uma equipe, $u \in V_j$ um indivíduo e $S' \subseteq S$ um subconjunto de habilidades.

Considere uma solução viável x . Primeiramente, admita que u não pertence à equipe j . Então teremos que

$\sum_{s \in \bar{s}(u, j) \cap S'} x_{us}^j = \sum_{s \in \bar{s}(u, j) \setminus S'} x_{us}^j = 0$. Logo $\Phi(u, j, S') \geq 0$ é satisfeita pois $y_{uv}^j \geq 0$, para todo $v \in V_j$ tal que $\bar{s}(v, j) \neq \emptyset$.

Agora, olharemos o caso em que o indivíduo u está na equipe j usando a habilidade $s \in \bar{s}(u, j)$. Então, teremos $\sum_{s \in \bar{s}(u, j) \cap S'} x_{us}^j = 1$ e $\sum_{s \in \bar{s}(u, j) \setminus S'} x_{us}^j = 0$ (se $s \in S'$), ou $\sum_{s \in \bar{s}(u, j) \cap S'} x_{us}^j = 0$ e $\sum_{s \in \bar{s}(u, j) \setminus S'} x_{us}^j = 1$ (se $s \notin S'$). Logo, $\Phi(u, j, S') \geq 0$ torna-se:

$$\sum_{v \in \mathcal{A}_u(j, S): \bar{s}(v, j) \cap S' \neq \emptyset} y_{uv}^j \geq \begin{cases} t(j, S') - 1, & \text{se } s \in S' \\ t(j, S'), & \text{se } s \notin S' \end{cases}$$

Por conveniência, defina $S^* = S' \setminus \{s\}$, se $s \in S'$, $\bar{s}(u, j) = \{s\}$ e $t(j, s) = 1$, e $S^* = S'$, caso contrário. Dessa forma, a desigualdade $\Phi(u, j, S') \geq 0$ reescreve-se como:

$$\sum_{v \in \mathcal{A}_u(j, S) : \bar{s}(v, j) \cap S' \neq \emptyset} y_{uv}^j \geq \begin{cases} t(j, S^*) - 1, & \text{se } s \in S^* \\ t(j, S^*), & \text{se } s \notin S^* \end{cases}$$

A equivalência entre as duas expressões destacadas acima vale trivialmente se $S' = S^*$. No outro caso, ela segue do fato que $s \in S'$, $s \notin S^*$ e $t(j, S^*) = t(j, S') - 1$.

Seja $V^* = \{v \in V_j \setminus \{u\} : \sum_{s' \in \bar{s}(v, j) \cap S^*} x_{vs'}^j = 1\}$ o conjunto de indivíduos que estão usando uma habilidade em S^* na equipe j que não seja o indivíduo u . Lembre que $x_{us}^j = 1$. Logo por (4.28) temos $\sum_{v \in V^*} y_{uv}^j = |V^*|$. Além disso, $|V^*| = t(j, S^*)$ se $s \notin S^*$ e $|V^*| = t(j, S^*) - 1$ se $s \in S^*$. Portanto, precisamos mostrar que $\sum_{v \in \mathcal{A}_u(j, S) : \bar{s}(v, j) \cap S' \neq \emptyset} y_{uv}^j \geq |V^*| = \sum_{v \in V^*} y_{uv}^j$ ou, de uma outra forma, que $V^* \subseteq \{v \in \mathcal{A}_u(j, S) : \bar{s}(v, j) \cap S' \neq \emptyset\}$.

Seja $v \in V^*$. Pela definição de V^* , deduzimos que $\bar{s}(v, j) \neq \emptyset$. Além disso, $s \in \bar{s}(u, j) \neq \emptyset$. Logo, $(u, v) \in E'_j$. Suponha agora que $\bar{s}(u, j) = \bar{s}(v, j) = \{s'\}$. Então, $s' = s$. Como $v \in V^*$, então $s \in S^* \subseteq S'$. Portanto, pela definição de S^* teremos $S^* = S'$ e $t(j, s) \geq 2$. Logo, $v \in \mathcal{A}_u(j, S)$ e, consequentemente, chegamos a $V^* \subseteq \{v \in \mathcal{A}_u(j, S) : \bar{s}(v, j) \cap S' \neq \emptyset\}$. $\square$

5 EXPERIMENTOS E RESULTADOS COMPUTACIONAIS

Neste capítulo, mostraremos primeiro o ambiente computacional e as instâncias que serão utilizadas nos experimentos computacionais. Depois, usaremos os métodos apresentados nos Capítulos 3 e 4 aplicados às instâncias apresentadas e faremos uma análise comparativa entre os métodos.

5.1 Ambiente Computacional

Os experimentos foram realizados utilizando uma máquina com o processador Intel® Core™ i7-12700U CPU com 4.90 GHz, 32 GB RAM e sistema operacional Ubuntu 22.04 LTS. Os modelos foram implementados usando a linguagem de programação C++ e o compilador g++, e resolvidos com o solver *IBM ILOG CPLEX 22.1.1* em sua versão padrão de *branch-and-cut*. Para as formulações de PLI, **Modelo- $\mathcal{F}_{MTZ}$** , **Modelo- $\mathcal{F}_{SIGN}$** , **Modelo- $\mathcal{TF}$** e **Modelo- $\mathcal{TF}$** combinado com desigualdades válidas, os testes foram realizados considerando um limite de tempo 7200 segundos (2 horas) para a resolução de cada uma das instâncias.

5.2 Instâncias

Uma instância do problema é definida por um grafo de sinais G , o número de equipes desejadas $|T|$, o número de habilidades consideradas $|S|$, a função $\bar{s} : V \rightarrow 2^S$, que estabelece as habilidades dos indivíduos, e a função $t : T \times S \rightarrow \mathbb{N}$, que registra o número de indivíduos de cada habilidade por equipe. Usamos $S = \{1, \dots, |S|\}$ e $T = \{1, \dots, |T|\}$.

As instâncias utilizadas nos experimentos são as mesmas usadas em (FIGUEIREDO, 2021) como instâncias de teste para outros problemas de formação de equipes em grafos de sinais. Cada grafo é originalmente direcionado; uma versão não direcionada é também obtida, colocando uma aresta entre u e v se existe um arco (u, v) ou (v, u) . O sinal da aresta será positivo e, somente se, pelo menos um dos arcos correspondentes for positivo.

As instâncias podem ser particionadas em 3 tipos, de acordo com a origem do grafo de sinais. O primeiro tipo compreende grafos de sinais gerados de forma pseudoaleatória, aqui denominados de grafos *Randômicos*, ou apenas *Random*. Os outros dois tipos de grafos de sinais foram gerados a partir de dados obtidos das redes sociais online *BitCoin* e *Epinions*, disponibilizados no link <https://snap.stanford.edu/data/index.html>. A quantidade de vértices $|V|$ varia em $\{25, 50, 100\}$, e as médias das densidades de arestas estão próximas a 0,3, 0,5 ou 0,7. Os grafos foram então divididos em três classes, 1, 2 e 3, de acordo com sua densidade de arestas, 0,3, 0,5 e 0,7, respectivamente. Há um grafo de cada tipo em cada classe, que será identificado pelo nome do tipo junto com o número da classe. Por exemplo,

BitCoin 3 representa um grafo de sinais gerado a partir da rede *BitCoin* possuindo uma densidade de arestas próxima de 70%. Nas instâncias em que o grafo possui $|V| = 25$, temos $|S| = 10$ e $|T| = 2$; no caso em que $|V| = 50$, foi fixado $|T| = |S| = 5$; para o caso em que $|V| = 100$, teremos $|S| = 5$ e $|T| = 10$.

Outros características dos grafos de sinais das instâncias podem ser vistos nas Tabelas 1 e 2. Para cada tipo de grafo ('Tipo') e número de vértices (' $|V|$ '), são apresentadas a porcentagem de arestas (resp. arcos) positivas e negativas, nas colunas ' $|E^+|(\%)$ ' e ' $|E^-|(\%)$ ', respectivamente. Note que todos os grafos em ambas as tabelas possuem uma porcentagem bem maior de arestas (resp. arcos) positivas que negativas. Os maiores percentuais de arestas (resp. arcos) negativas aparecem em *Random 1* e *Bitcoin 3*.

Tabela 1 – Porcentagem de arestas positivas e negativas nos grafos não direcionados

Tipo	$ V $	$ E^+ (\%)$	$ E^- (\%)$
Random 1	25	77,13%	22,87%
	50	80,84%	19,16%
	100	79,84%	20,16%
Random 2	25	90,72%	9,28%
	50	89,35%	10,65%
	100	89,49%	10,51%
Random 3	25	97,09%	2,91%
	50	95,22%	4,78%
	100	97,39%	2,61%
Bitcoin 1	25	95,17%	4,83%
	50	92,12%	7,88%
	100	92,31%	7,69%
Bitcoin 2	25	93,46%	6,54%
	50	92,95%	7,05%
	100	93,35%	6,65%
Bitcoin 3	25	71,19%	28,81%
	50	58,89%	41,11%
	100	60,11%	39,89%
Epinions 1	25	72,25%	27,75%
	50	91,78%	8,22%
	100	82,59%	17,41%
Epinions 2	25	87,96%	12,04%
	50	82,26%	17,74%
	100	88,15%	11,85%
Epinions 3	25	97,89%	2,11%
	50	87,71%	12,29%
	100	81,49%	18,51%

Tabela 2 – Porcentagem de arcos positivos e negativos nos grafos direcionados

Tipo	$ V $	$ A^+ (\%)$	$ A^- (\%)$
Random 1	25	72,50%	27,50%
	50	74,45%	25,55%
	100	76,36%	23,64%
Random 2	25	85,55%	14,45%
	50	85,13%	14,87%
	100	85,39%	14,61%
Random 3	25	93,33%	6,67%
	50	89,51%	10,49%
	100	93,42%	6,58%
Bitcoin 1	25	92,31%	7,69%
	50	93,77%	6,23%
	100	93,59%	6,41%
Bitcoin 2	25	95,95%	4,05%
	50	93,47%	6,53%
	100	93,71%	6,29%
Bitcoin 3	25	75,84%	24,16%
	50	65,64%	34,36%
	100	67,36%	32,64%
Epinions 1	25	78,12%	21,88%
	50	94,29%	5,71%
	100	87,06%	12,94%
Epinions 2	25	92,18%	7,82%
	50	88,46%	11,54%
	100	91,26%	8,74%
Epinions 3	25	98,68%	1,32%
	50	92,50%	7,50%
	100	85,55%	14,45%

Ambas as funções $\bar{s}(u)$ e $t(j, s)$, conforme descrito em (FIGUEIREDO, 2021), foram criadas utilizando parâmetros pré-estabelecidos para determinar a distribuição das habilidades pelos indivíduos e a requisição destes pelas equipes. Para cada valor $|V|$, foram geradas de forma pseudoaleatória 6 funções do tipo t . Associada à cada função t , foi gerada uma função $\bar{s}$, definindo aleatoriamente as habilidades de cada indivíduo, de tal modo que seja sempre possível satisfazer as requisições de habilidades de cada equipe (sem considerar compatibilidade). Maiores detalhes sobre as instâncias podem ser vistos em (FIGUEIREDO, 2021).

Com o acréscimo das restrições de *compatibilidade* ou *compatibilidade direcionada*, que não existiam em (FIGUEIREDO, 2021), pode ocorrer a situação em que, para alguns desses 6 grupos de funções $\bar{s}$ e t , não exista uma ou mais equipes que satisfaçam os requisitos de habilidade, levando portanto a uma instância inviável para o TFP_SCP. Neste trabalho, as instâncias que resultaram inviáveis foram eliminadas de todas tabelas apresentadas a partir da Seção 5.4. Além disso, também foram removidas as

instâncias com grafos de sinais desconexos.

As Tabelas 3 e 4 identificam tais características para cada instância não direcionada e direcionada, respectivamente. As informações apresentadas são: ‘Classe’, ‘ $|V|$ ’, ‘ $[\bar{s}, t]$ ’, representando a classe a que o grafo pertence (1, 2 ou 3), a quantidade de vértices (25, 50 ou 100) e a combinação das funções $\bar{s}$ e $t(j, s)$ (1, 2, 3, 4, 5 ou 6), respectivamente. Temos 3 possíveis casos para cada instância: “✓”, “I” e “D”, representado nessa ordem o caso em que a instância possui solução viável, não possui solução viável e quando o grafo é desconexo. No total teremos 127 instâncias viáveis com grafos não direcionados e 54 para os direcionados.

Tabela 3 – Teste de Viabilidade de Instâncias com Grafos Não Direcionados

Instância Não direcionada	Classe $ V $ $[\bar{s}, t]$	1						2						3					
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
<i>Random</i>	25	I	I	✓	I	I	I	✓	✓	✓	✓	I	✓	✓	I	✓	✓	✓	✓
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	100	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>BitCoin</i>	25	✓	✓	✓	✓	✓	✓	✓	I	✓	✓	✓	✓	I	I	✓	✓	I	I
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	100	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Epinions</i>	25	✓	I	✓	I	I	I	D						D					
	50	✓	✓	✓	✓	✓	✓	D						✓	✓	✓	✓	I	✓
	100	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Total ✓	127																		

Fonte: elaborado pelo autor.

Tabela 4 – Teste de Viabilidade de Instâncias com Grafos Direcionados

Instância Direcionada	Classe	1						2						3					
	$\begin{matrix} [\bar{s}, t] \\ V \end{matrix}$	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Random	25	I	I	I	I	I	I	I	I	✓	✓	I	I	I	I	✓	✓	I	I
	50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	100	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
BitCoin	25	D																	
	50	D																	
	100	D																	
Epinions	25	✓	I	✓	I	I	I	D						D					
	50	✓	✓	✓	✓	✓	✓	D						D					
	100	✓	✓	✓	✓	✓	✓	D						D					
Total ✓	54																		

Fonte: elaborado pelo autor.

5.3 Resultados Computacionais para o SPP

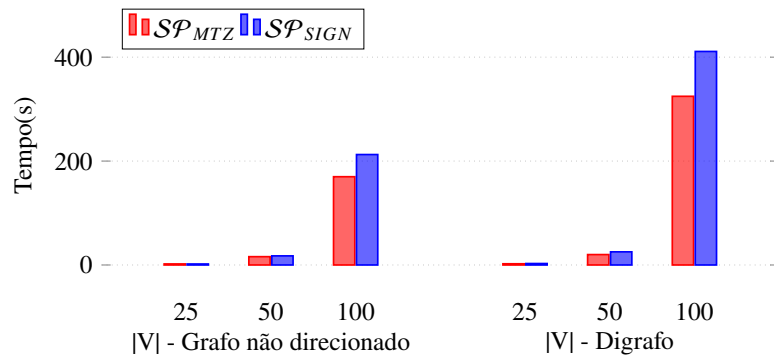
Apresentaremos resultados computacionais para o SPP com instâncias em que os grafos e digrafos são conexos, que são 25 instâncias com grafos não direcionados e 12 com grafos direcionados. A exigência para os digrafos é que sejam fracamente conexos, isto é, que exista pelo menos um caminho entre cada dois vértices distintos.

Compararemos os métodos apresentados nas Subseções 4.2.3 e 4.3.1 para determinação dos caminhos positivos mínimos e consequente construção do grafo de compatibilidade G' . Para o caso não direcionado, usaremos o algoritmo $MinMatch(u, v)$ e os modelos de PLI $\mathcal{SP}_{MTZ}$ e $\mathcal{SP}_{SIGN}$. Para o caso direcionado, usaremos apenas os dois modelos. Além disso, testamos o balanceamento de todos os grafos. Como a grande maioria não é balanceado, não utilizaremos na comparação ao Algoritmo 1, apresentado na Seção 3.2.

Nas Tabelas 5 e 6 temos as colunas ' $MinMatch(s)$ ', ' $\mathcal{SP}_{MTZ}(s)$ ', ' $\mathcal{SP}_{SIGN}(s)$ ', representando os tempos de execução (em segundos) de cada método para criar o grafo G' , no caso não direcionado e direcionado, respectivamente. Conseguimos perceber que o algoritmo $MinMatch$ consome tempo de execução menor na maioria dos casos com 25 vértices presentes na Tabela 5, quando comparado com os métodos $\mathcal{SP}_{MTZ}$ e $\mathcal{SP}_{SIGN}$. Entretanto, considerando a quantidade de vértices 50 e 100 na Tabela 5 e todas as quantidades de vértices na Tabela 6, na maioria dos grafos o método com menor gasto de tempo é o $\mathcal{SP}_{MTZ}$. Esta evidência é confirmada pelos valores na linha 'Média(s)', exibidos na última linha dessas tabelas, que representam o valor do tempo médio em segundos para todos os grafos testados.

No gráfico da Figura 7 apresentamos as médias dos tempos de execução dos modelos $\mathcal{SP}_{MTZ}$ e $\mathcal{SP}_{SIGN}$ para cada quantidade de vértices, em grafos não direcionados e em digrafos, respectivamente. Perceba um aumento significativo na média do tempo quando o número de vértices aumenta para 100, ainda mais no cenário com digrafos. Além disso, corroborando com a análise das tabelas anteriores, podemos ver que o modelo $\mathcal{SP}_{MTZ}$ tende a ter menores médias para ambos os tipos de grafo e todas quantidades de vértices. Não exibimos na figura as médias de $MinMatch$ (para o caso não direcionado) porque os tempos de execução para 100 vértices são bastante superiores aos dos outros métodos, o que distorceria o gráfico.

Figura 7 – Médias dos tempos por quantidade de vértices para cada método do SPP



Fonte: elaborado pelo autor.

Tabela 5 – Tempos de computação dos métodos para o SPP em grafos não direcionados

Instância	$ V $	$MinMatch(s)$	$\mathcal{SP}_{MTZ}(s)$	$\mathcal{SP}_{SIGN}(s)$
Random 1	25	0,49	0,90	0,94
	50	26,69	10,75	11,98
	100	1189,50	161,06	193,05
Random 2	25	0,91	0,62	0,65
	50	46,96	7,80	8,62
	100	2871,86	180,85	172,50
Random 3	25	1,30	0,35	0,35
	50	75,09	2,79	2,78
	100	4853,24	33,58	37,25
Bitcoin 1	25	0,79	0,82	0,81
	50	27,72	13,30	13,42
	100	847,44	198,49	210,16
Bitcoin 2	25	0,47	1,43	1,25
	50	16,03	15,42	16,60
	100	238,76	116,85	144,51
Bitcoin 3	25	0,22	8,85	7,94
	50	6,17	41,70	56,40
	100	169,46	264,89	544,57
Epinions 1	25	0,52	1,17	1,42
	50	44,92	8,74	9,40
	100	1567,39	151,20	177,23
Epinions 2	100	1043,69	200,16	233,73
Epinions 3	50	10,11	27,45	20,96
	100	280,04	221,84	200,00
Média(s)		554,99	69,63	86,11

Fonte: elaborado pelo autor.

Tabela 6 – Tempos de computação dos métodos para o SPP em grafos direcionados

Instância	$ V $	$\mathcal{SP}_{MTZ}(s)$	$\mathcal{SP}_{SIGN}(s)$
Random 1	25	1,93	2,37
	50	22,11	29,37
	100	287,54	390,83
Random 2	25	1,72	2,08
	50	22,05	27,17
	100	475,37	549,03
Random 3	25	2,15	2,31
	50	14,49	19,22
	100	211,02	292,98
Epinions 1	25	3,46	4,51
	50	21,43	25,40
	100	302,54	429,32
Média(s)		113,82	147,88

Fonte: elaborado pelo autor.

5.4 Resultados Computacionais para o TFP_SPP Não Direcionado

Nesta seção, analisaremos 5 métodos de resolução para instâncias do TFP_SPP não direcionado. Os dois primeiros são o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** apresentados na Subseção 4.2.1 e estarão relacionados às colunas ' $\mathcal{F}_{MTZ}$ ' e ' $\mathcal{F}_{SIGN}$ ' nas próximas tabelas. Os outros métodos serão originados das formulações decompostas mostradas nas Subseções 4.2.2 a 4.2.4. Para a criação dos grafos G' na primeira etapa de resolução, usaremos os métodos que apresentaram melhores resultados para cada quantidade de vértices na maioria das instâncias, resultados exibidos nas Tabelas 5 e 6, a saber: o algoritmo *MinMatch*, que por simplicidade chamaremos apenas de *MM*, para grafos com 25 vértices, e o modelo $\mathcal{SP}_{MTZ}$, para as instâncias com 50 e 100 vértices. Além disso, serão consideradas 3 versões do **Modelo- $\mathcal{TF}$** : o **Modelo- $\mathcal{TF}$** como apresentado na Subseção 4.2.4; o **Modelo- $\mathcal{TF}_{d1}$** , obtido do anterior com a adição das desigualdades válidas (4.39); o **Modelo- $\mathcal{TF}_{d2}$** , **dado pelo Modelo- $\mathcal{TF}$** substituindo (4.30) e (4.31) por (4.35) e (4.36). Usaremos nas tabelas, por simplicidade, as nomenclaturas $\mathcal{TF}$, $\mathcal{TF}_{d1}$ e $\mathcal{TF}_{d2}$. Dessa forma, a coluna ' $MM + \mathcal{TF}_{d1}$ ' representa o método de solução do TFP_SPP obtido com a construção do grafo G' pelo algoritmo *MM* (*MinMatch*) junto com a resolução do **Modelo- $\mathcal{TF}_{d1}$** . Denotamos as outras combinações de forma similar.

5.4.1 Dimensões dos Modelos

Antes de avaliarmos o desempenho computacional dos modelos mencionados acima, iremos comparar as suas dimensões. Primeiro, analisaremos o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** . O segundo não precisa das variáveis λ_{uv} , para todo $(u, v) \in \bar{E}_T$, usada no primeiro. Além disso, emprega as variáveis μ_p^{uv} , em lugar das variáveis π_p^{uv} , ambas em mesma quantidade. Assim, é fácil ver que o **Modelo- $\mathcal{F}_{MTZ}$** possui $|\bar{E}_T|$ variáveis a mais que o **Modelo- $\mathcal{F}_{SIGN}$** . Com respeito às restrições, sendo

$$k = \sum_{j \in T} |\bar{s}(j)| \quad \text{e} \quad m = \sum_{j \in T} |\bar{E}_j| = \sum_{j \in T} |V_j|(|V_j| - 1)/2,$$

chegamos às quantidades apresentadas na Tabela 7.

Tabela 7 – Número de restrições no **Modelo- $\mathcal{F}_{MTZ}$** e no **Modelo- $\mathcal{F}_{SIGN}$**

Formulação	#Restrições
$\mathcal{F}_{MTZ}$	$ V_T + k + E_T \cap E^- + 3m + \bar{E}_T \cdot V + \bar{E}_T \cdot A + (\bar{E}_T \cdot A + \bar{E}_T)$
$\mathcal{F}_{SIGN}$	$ V_T + k + E_T \cap E^- + 3m + \bar{E}_T \cdot V + \bar{E}_T \cdot A + (2 \cdot \bar{E}_T \cdot A^- + 2 \cdot \bar{E}_T \cdot A^+ + 2 \cdot \bar{E}_T)$

Fonte: elaborado pelo autor.

Agora, analisando apenas as restrições que são diferentes entre os modelos, ou seja, (4.10)-(4.11) em $\mathcal{F}_{MTZ}$ e (4.17)-(4.21) em $\mathcal{F}_{SIGN}$, chegamos à seguinte relação que nos mostra que a formulação $\mathcal{F}_{SIGN}$ tem $|\bar{E}_T| \cdot (|A| + 1)$ restrições a mais que a formulação $\mathcal{F}_{MTZ}$.

- número de restrições (4.10)-(4.11):

$$|\bar{E}_T| \cdot |A| + |\bar{E}_T| = |\bar{E}_T| \cdot (|A| + 1)$$

- número de restrições (4.17)-(4.21):

$$2 \cdot |\bar{E}_T| \cdot |A^-| + 2 \cdot |\bar{E}_T| \cdot |A^+| + 2 \cdot |E_T| = |\bar{E}_T| \cdot (2 \cdot |A| + 2) = 2 \cdot |\bar{E}_T| \cdot (|A| + 1)$$

Analisando agora os modelos $T\mathcal{F}$, $T\mathcal{F}_{d1}$ e $T\mathcal{F}_{d2}$, claramente todos possuem a mesma quantidade de variáveis. Entretanto, quando comparamos o número de restrições, usando

$$m' = \sum_{j \in T} |E'_j| \quad \text{e} \quad n = \sum_{j \in T} |V_j|$$

obtemos a tabela a seguir.

Tabela 8 – Número de restrições nas três versões do **Modelo- $T\mathcal{F}$**

Modelo	#Restrições
$T\mathcal{F}$	$ V_T + k + m + 2m'$
$T\mathcal{F}_{d1}$	$ V_T + k + m + 2m' + n$
$T\mathcal{F}_{d2}$	$ V_T + k + m + n + T $

Fonte: elaborado pelo autor.

5.4.2 Comparação do Desempenho dos Modelos

Nesta subseção, analisaremos o desempenho computacional dos modelos mencionados no início da seção. Em cada tabela, a coluna ‘Tempo(s)’ representa o tempo de execução em segundos de um modelo em cada instância. As siglas ‘TL’ ou ‘EM’ nessa coluna indicam que o modelo atingiu o tempo limite (TL) ou que a execução finalizou por consumo excessivo de memória (EM), respectivamente. O tempo limite de cada execução foi fixado em 7200 segundos (2 horas). Para instâncias maiores, esse tempo limite pode ser excedido sem obtenção de solução ótima. Quando relevante, vamos registrar o gap relativo retornado pelo solver *CPLEX*, que chamaremos simplesmente de *gap*, definido pela fórmula $gap = 100 \cdot \frac{|BB-BI|}{1E^{-10}+|BI|}$, em que *BB* (*best bound*) representa o melhor limitante inferior encontrado para a solução ótima e *BI* (*best integer*) é o valor da melhor solução inteira encontrada. Nas configurações padrões do solver *CPLEX*, quando ocorre ‘EM’, não é salva informação sobre solução viável encontrada, por isso nesse caso não registraremos o gap e consideraremos que formulação não retornou nenhuma solução viável.

Para auxiliar na análise das tabelas, incluímos as seguintes linhas: ‘Média(s)’, ‘Média(s)*’ e ‘#Viáveis / #Ótimas’, caracterizando respectivamente a média dos tempos de execução para cada modelo em todas as instâncias, a média dos tempos de execução para instâncias em que todos os modelos retornaram a solução ótima, a quantidade de soluções viáveis (#Viáveis) e a quantidade de soluções ótimas (#Ótimas).

A Tabela 10 especificamente possui a linha ‘Média(s)**’, que mostra a média dos tempos das instâncias em que os modelos $T\mathcal{F}$, $T\mathcal{F}_{d1}$ e $T\mathcal{F}_{d2}$ chegaram à solução ótima.

A Tabela 9 apresenta os resultados dos testes com grafos de sinais possuindo 25 vértices. Pode-se notar que tanto os modelos $\mathcal{F}_{MTZ}$ e $\mathcal{F}_{SIGN}$ quanto a metodologia de decomposição, modelos $MM + T\mathcal{F}$, $MM + T\mathcal{F}_{d1}$ e $MM + T\mathcal{F}_{d2}$, obtiveram a solução ótima antes do limite de tempo proposto. É importante destacar que há uma notável diferença entre os desempenhos computacionais das duas abordagens. O tempo de execução das estratégias de decomposição é inferior aos dos modelos $\mathcal{F}_{MTZ}$ e $\mathcal{F}_{SIGN}$ em todas, exceto duas, instâncias testadas. Essa diferença pode ser evidenciada também a partir da média dos tempos de execução de todas as instâncias, apresentada na última linha da Tabela 9, sendo acima de 5 segundos para os modelos $\mathcal{F}_{MTZ}$ e $\mathcal{F}_{SIGN}$, enquanto 1 segundo é a maior média para os métodos que combinam o algoritmo MM (Algoritmo *MinMatch*) e as formulações $T\mathcal{F}$, $T\mathcal{F}_{d1}$ ou $T\mathcal{F}_{d2}$. Em geral, independentemente do tipo de grafo e sua densidade, quando $|V| = 25$, as abordagens com decomposição demonstram ter maior eficiência que os modelos $\mathcal{F}_{MTZ}$ e $\mathcal{F}_{SIGN}$ com relação ao tempo de execução.

Tabela 9 – Desempenho dos modelos para o TFP_SPP em grafos não direcionados de 25 vértices

Instância			$\mathcal{F}_{MTZ}$	$\mathcal{F}_{SIGN}$	$MM + T\mathcal{F}$	$MM + T\mathcal{F}_{d1}$	$MM + T\mathcal{F}_{d2}$
Tipo	$ V $	$[\bar{s}, t]$	Tempo(s)	Tempo(s)	Tempo(s)	Tempo(s)	Tempo(s)
Random 1	25	3	2,35	2,41	0,53	0,98	0,67
Random 2	25	1	1,68	1,76	0,99	1,03	0,98
		2	3,39	3,79	1,12	1,01	0,96
		3	3,68	4,41	1,02	1,12	0,93
		4	2,10	2,03	0,95	1,00	0,92
		6	1,25	1,27	0,98	0,97	0,94
Random 3	25	1	1,08	1,19	1,44	1,38	1,31
		3	3,72	3,81	1,78	2,09	1,33
		4	1,46	1,66	1,40	1,37	1,32
		5	1,31	1,44	1,47	1,46	1,44
		6	3,37	3,03	1,68	2,08	1,39
Bitcoin 1	25	1	2,37	2,61	0,85	0,94	0,86
		2	3,41	3,41	0,99	0,91	0,92
		3	12,03	11,03	0,99	0,93	0,91
		4	3,00	3,14	1,07	0,94	0,84
		5	3,92	4,11	0,95	1,02	1,12
		6	8,73	9,41	0,96	1,05	0,95
Bitcoin 2	25	1	3,27	3,42	0,68	0,50	0,62
		3	19,04	23,38	0,69	0,77	0,55
		4	5,06	5,95	0,62	0,61	0,64
		5	5,03	6,02	0,75	0,71	0,56
		6	20,64	17,67	0,87	0,79	0,58
Bitcoin 3	25	3	28,94	19,49	0,37	0,71	0,31
		4	1,20	1,20	0,25	0,23	0,23
Epinions 1	25	1	2,15	2,21	0,62	0,62	0,53
		3	3,87	4,67	0,62	0,70	0,53
Média(s)			5,69	5,56	0,95	1,00	0,86

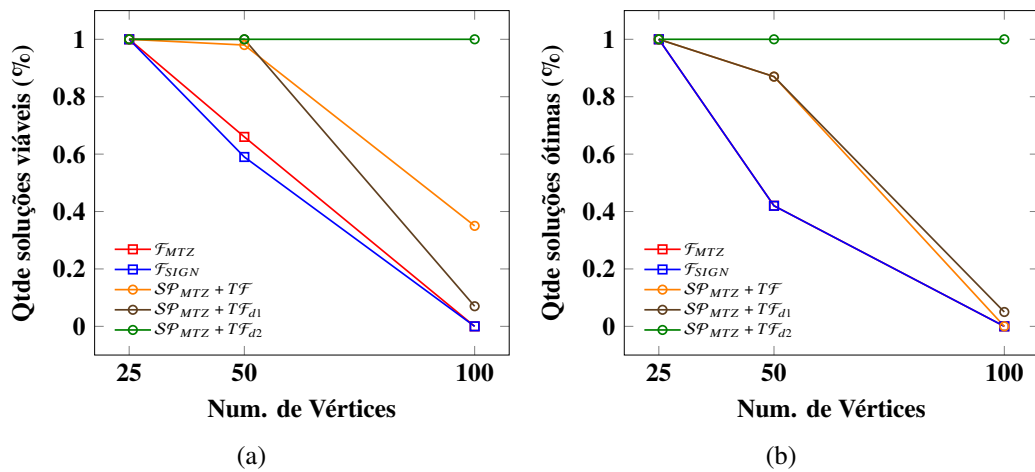
Fonte: elaborado pelo autor.

A partir da Tabela 10, analisaremos agora os resultados computacionais para instâncias com grafos não direcionados com $|V| = 50$. Comparando os tempos de execução, percebemos o aumento da diferença, já existente na Tabela 9 apresentada anteriormente, entre as formulações $\mathcal{F}_{MTZ}$ e $\mathcal{F}_{SIGN}$ e as estratégias de decomposição da formulação. Essa melhor performance no tempo é confirmada pela média dos tempos considerando as instâncias em que todos os modelos chegaram à solução ótima, apresentado na linha ‘Média(s)*’. Mais ainda, comparando os três modelos simplificados por meio das médias dos tempos das instâncias resolvidas pelos três (linha ‘Média(s)**’), fica claro o desempenho significativamente superior do modelo $T\mathcal{F}_{d2}$.

Como vimos, o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** apresentaram desempenho bem inferior ao dos modelos decompostos e já não conseguiram resolver boa parte das instâncias com 50 vértices no tempo limite. Sendo assim, para as instâncias com $|V| = 100$ (ver Tabela 11), vamos apresentar os resultados apenas para os modelos $T\mathcal{F}$, $T\mathcal{F}_{d1}$ e $T\mathcal{F}_{d2}$.

Na Tabela 11, percebemos claramente o aumento da dificuldade dos modelos em obterem soluções ótimas para as instâncias maiores, fato evidenciado pelo número de casos com tempo limite de execução ou excesso de memória. O modelo $T\mathcal{F}_{d1}$, principalmente, sofreu um maior impacto, devido ao crescimento do número de restrições, que levou a uma demanda excessiva de memória em todas as instâncias, à exceção de 4 delas. Mesmo o número de soluções viáveis encontradas por $T\mathcal{F}_{d1}$ foi baixo, inclusive menor que $T\mathcal{F}$, como confirma o gráfico da Figura 8(a). Embora com um aumento do tempo, em relação à Tabela 10, a formulação $T\mathcal{F}_{d2}$ conseguiu se manter estável e chegar à solução ótima de todas as instâncias com 100 vértices testadas.

Figura 8 – Porcentagem de instâncias, por número de vértices, com soluções viáveis e ótimas encontradas por cada modelo para grafos não direcionados



Fonte: elaborado pelo autor.

Tabela 10 – Desempenho dos modelos para o TFP_SPP em grafos não direcionados de 50 vértices

Instância			F_{MTZ}		F_{SIGN}		$SP_{MTZ} + T\mathcal{F}$		$SP_{MTZ} + T\mathcal{F}_{d1}$		$SP_{MTZ} + T\mathcal{F}_{d2}$		
Tipo	V	$[\bar{s}, t]$	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)	
Random 1	50	1	18,89	TL	-	EM	0	55,22	0	194,18	0	10,82	
		2	-	EM	-	EM	0	4673,06	0	3002,14	0	10,85	
		3	0	238,94	0	319,31	0	11,10	0	10,91	0	10,80	
		4	0	71,05	0	80,42	0	10,92	0	10,78	0	10,83	
		5	0	1481,35	0	2740,46	0	16,39	0	31,01	0	10,89	
		6	0	2141,04	0	2383,90	0	15,86	0	13,87	0	11,04	
Random 2	50	1	26,67	TL	26,67	TL	0	2620,54	11,43	TL	0	7,88	
		2	33,33	TL	33,33	TL	16,67	TL	14,29	TL	0	7,86	
		3	0	153,06	0	281,13	0	8,32	0	7,82	0	7,82	
		4	0	131,56	0	155,28	0	8,49	0	7,82	0	7,82	
		5	-	EM	-	EM	0	77,21	0	20,51	0	7,88	
		6	11,66	TL	12,99	TL	0	30,93	0	34,43	0	7,96	
Random 3	50	1	26,67	TL	26,67	TL	19,33	TL	15,89	TL	0	2,64	
		2	33,33	TL	33,33	TL	23,33	TL	26,67	TL	0	2,64	
		3	0	16,22	0	22,11	0	2,83	0	2,61	0	2,60	
		4	0	46,87	0	58,82	0	2,82	0	2,61	0	2,60	
		5	-	EM	-	EM	-	EM	0	213,17	0	2,63	
		6	-	EM	-	EM	0	2686,72	0	69,97	0	2,68	
Bitcoin 1	50	1	20,00	TL	20,00	TL	0	27,34	0	55,52	0	13,38	
		2	33,33	TL		EM	0	2526,08	0	3136,42	0	13,36	
		3	0	369,88	0	268,67	0	13,53	0	13,32	0	13,33	
		4	0	122,88	0	187,17	0	14,48	0	13,37	0	13,41	
		5	11,90	TL	12,50	TL	0	22,28	0	21,93	0	13,57	
		6	0	3513,33	0	5042,55	0	23,71	0	165,65	0	14,32	
Bitcoin 2	50	1	-	EM		EM	0	545,80	0	295,65	0	15,49	
		2	-	EM		EM	30,00	TL	20,00	TL	0	15,49	
		3	0	303,46	0	458,79	0	15,72	0	15,44	0	15,44	
		4	0	189,68	0	271,80	0	15,90	0	15,44	0	15,44	
		5	17,24	TL	18,28	TL	0	44,99	0	29,36	0	15,63	
		6	16,86	TL		EM	0	69,05	0	41,65	0	15,46	
Bitcoin 3	50	1	-	EM		EM	0	46,37	0	51,16	0	41,77	
		2	-	EM		EM	0	68,22	0	257,29	0	41,79	
		3	0	172,979	0	562,46	0	41,97	0	41,72	0	41,77	
		4	0	138,96	0	113,94	0	41,88	0	41,72	0	41,76	
		5	-	EM	-	EM	0	45,15	0	43,49	0	41,91	
		6	-	EM	-	EM	0	46,30	0	43,74	0	42,01	
Epinions 1	50	1	-	EM	-	EM	0	1079,00	0	50,96	0	8,82	
		2	-	EM	-	EM	20,00	TL	16,15	TL	0	8,81	
		3	0	182,155	0	268,41	0	9,29	0	8,76	0	8,94	
		4	0	161,333	0	262,91	0	8,94	0	8,76	0	8,76	
		5	-	EM	-	EM	0	591,30	0	25,06	0	8,78	
		6	-	EM	-	EM	0	82,32	0	46,23	0	8,84	
Epinions 3	50	1	-	EM	-	EM	0	30,86	0	40,23	0	27,50	
		2	-	EM	-	EM	0	48,40	0	385,30	0	27,54	
		3	0	101,51	0	114,72	0	27,90	0	27,45	0	27,46	
		4	0	73,41	0	92,29	0	27,64	0	27,45	0	27,46	
		5	0	2098,1	0	2529,1	0	34,26	0	35,91	0	27,85	
		6											
#Viáveis / #Ótimas				31 / 20		28 / 20		46 / 41		47 / 41		47 / 47	
Média(s)*				585,39		810,71		17,60		25,12		16,02	
Média(s)**				-		-		328,71		208,69		17,41	

Fonte: elaborado pelo autor.

Tabela 11 – Desempenho dos modelos para o TFP_SPP com grafos não direcionados de 100 vértices

Instância			$\mathcal{SP}_{MTZ} + T\mathcal{F}$		$\mathcal{SP}_{MTZ} + T\mathcal{F}_{d1}$		$\mathcal{SP}_{MTZ} + T\mathcal{F} + d2$	
Tipo	V	$[\bar{s}, t]$	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)
Random 1	100	1	72,73	TL	-	EM	0	322,38
		2	74,50	TL	-	EM	0	322,28
		3	-	EM	-	EM	0	325,57
		4	48,26	TL	-	EM	0	325,04
		5	-	EM	-	EM	0	322,89
		6	-	EM	-	EM	0	324,72
Random 2	100	1	72,73	TL	-	EM	0	359,78
		2	81,82	TL	-	EM	0	361,85
		3	-	EM	-	EM	0	360,98
		4	53,66	TL	-	EM	0	360,93
		5	-	EM	-	EM	0	360,67
		6	-	EM	-	EM	0	360,13
Random 3	100	1	78,18	TL	-	EM	0	67,44
		2	-	EM	-	EM	0	67,33
		3	92,68	TL	-	EM	0	68,47
		4	-	EM	-	EM	0	68,17
		5	-	EM	-	EM	0	67,75
		6	-	EM	-	EM	0	68,32
Bitcoin 1	100	1	68,48	TL	-	EM	0	456,90
		2	74,55	TL	-	EM	0	398,20
		3	-	EM	-	EM	0	399,12
		4	-	EM	-	EM	0	404,39
		5	-	EM	-	EM	0	402,07
		6	-	EM	-	EM	0	403,90
Bitcoin 2	100	1	-	EM	-	EM	0	322,90
		2	81,82	TL	-	EM	0	235,74
		3	-	EM	-	EM	0	236,62
		4	-	EM	-	EM	0	236,89
		5	-	EM	-	EM	0	237,31
		6	-	EM	-	EM	0	236,79
Bitcoin 3	100	1	-	EM	-	EM	0	382,47
		2	-	EM	-	EM	0	534,87
		3	-	EM	-	EM	0	536,72
		4	-	EM	0	2809,95	0	629,48
		5	-	EM	1,78	TL	0	543,65
		6	-	EM	-	EM	0	540,87
Epinions 1	100	1	-	EM	-	EM	0	449,74
		2	-	EM	-	EM	0	304,52
		3	73,90	TL	-	EM	0	303,25
		4	46,34	TL	-	EM	0	303,69
		5	-	EM	-	EM	0	303,56
		6	-	EM	-	EM	0	304,36
Epinions 2	100	1	62,73	TL	-	EM	0	351,76
		2	78,18	TL	-	EM	0	398,18
		3	-	EM	-	EM	0	397,99
		4	35,89	TL	0	2377,89	0	402,39
		5	-	EM	-	EM	0	401,94
		6	-	EM	-	EM	0	400,45
Epinions 3	100	1	59,38	TL	-	EM	0	420,39
		2	71,65	TL	-	EM	0	443,56
		3	61,67	TL	-	EM	0	446,53
		4	-	EM	0	806,067	0	453,94
		5	-	EM	-	EM	0	450,51
		6	-	EM	-	EM	0	449,21
#Viáveis / #Ótimas				19 / 0		4 / 3		54 / 54

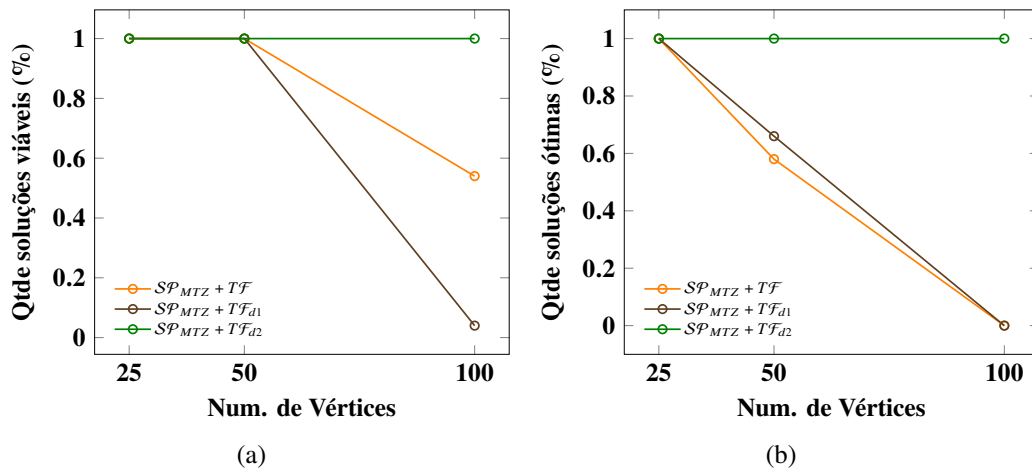
Fonte: elaborado pelo autor.

5.5 Resultados Computacionais para o TFP_SPP Direcionado

Focaremos nesta subseção em analisar os resultados computacionais dos modelos em digrafos de sinais. Nas Tabelas 12 e 13, as colunas possuem o mesmo significado apresentado no início da Subseção 5.4.2.

A análise dos resultados para digrafos é bem similar ao caso de grafos não direcionados. Para digrafos com 25 vértices, os três modelos tiveram desempenho similar. Com 50 vértices, já começa a prevalecer a versão TF_{d2} . Essa superioridade é ainda mais expressiva nas instâncias com 100 vértices. Novamente a abordagem via TF_{d2} consegue encontrar o ótimo de todas as instâncias testadas dentro do tempo limite. Colaborando com essa análise, a Figura 9 mostra que o percentual de instâncias com soluções viáveis e ótimas obtidas por TF_{d2} é igual ou superior aos outros modelos e que a diferença vai se tornando mais expressiva quando aumenta o número de vértices. Analisando apenas as formulações TF e TF_{d1} , no gráfico Figura 9(a) podemos ver que TF é superior a TF_{d1} em número de soluções viáveis, quando temos $|V| = 100$.

Figura 9 – Porcentagem de instâncias, por número de vértices, com soluções viáveis e ótimas encontradas por cada modelo para grafos direcionados



Fonte: elaborado pelo autor.

Tabela 12 – Desempenho dos modelos para o TFP_SPP em digrafos com 25 e 50 vértices

Instância			$\mathcal{SP}_{MTZ} + T\mathcal{F}$		$\mathcal{SP}_{MTZ} + T\mathcal{F}_{d1}$		$\mathcal{SP}_{MTZ} + T\mathcal{F}_{d2}$	
Tipo	V	$[\bar{s}, t]$	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)
Random 2	25	3	0	1,92	0	2,04	0	1,82
		4	0	1,85	0	1,74	0	1,73
Random 3	25	3	0	2,27	0	2,25	0	2,22
		4	0	2,20	0	2,20	0	2,17
Epinions 1	25	1	0	3,51	0	3,63	0	3,47
		3	0	3,58	0	3,56	0	3,53
Random 1	50	1	0	2180,33	13,33	TL	0	22,16
		2	19,26	TL	26,67	TL	0	22,21
		3	0	22,78	0	22,13	0	22,12
		4	0	22,33	0	22,13	0	22,12
		5	0	105,74	0	32,69	0	22,30
		6	0	37,33	0	25,25	0	22,35
Random 2	50	1	16,67	TL	17,38	TL	0	22,12
		2	18,89	TL	25,33	TL	0	22,10
		3	0	22,77	0	22,07	0	22,07
		4	0	22,95	0	22,07	0	22,07
		5	2,14	TL	0	236,28	0	22,09
		6	0	286,71	0	32,26	0	22,19
Random 3	50	1	20,00	TL	16,67	TL	0	14,53
		2	20,00	TL	13,33	TL	0	14,56
		3	0	15,03	0	14,51	0	14,51
		4	0	14,79	0	14,51	0	14,51
		5	2,20	TL	0	439,00	0	14,53
		6	0	140,09	0	30,40	0	14,52
Epinions 1	50	1	18,33	TL	20,00	TL	0	21,48
		2	26,25	TL	26,67	TL	0	21,51
		3	0	21,82	0	21,45	0	21,44
		4	0	21,81	0	21,45	0	21,50
		5	2,45	TL	0	27,02	0	21,48
		6	0	89,28	0	31,35	0	21,60
#Viáveis / #Ótimas				30 / 20		30 / 22		30 / 30
Média(s)*				44,15		17,25		14,64

Tabela 13 – Desempenho dos modelos para o TFP_SPP em digrafos com 100 vértices

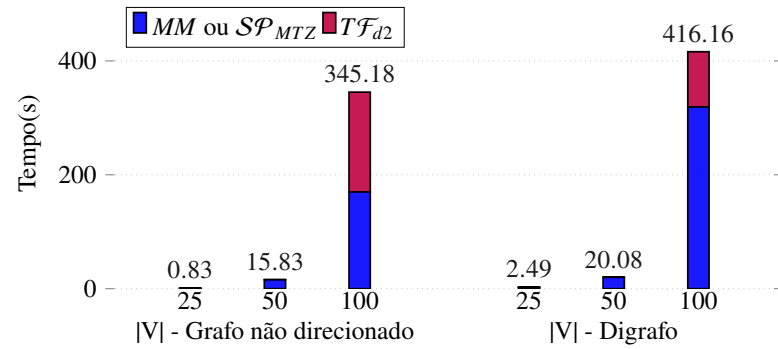
Instância			$\mathcal{SP}_{MTZ} + T\mathcal{F}$		$\mathcal{SP}_{MTZ} + T\mathcal{F}_{d1}$		$\mathcal{SP}_{MTZ} + T\mathcal{F}_{d2}$	
Tipo	$ V $	$[\bar{s}, t]$	gap(%)	Tempo(s)	gap(%)	Tempo(s)	gap(%)	Tempo(s)
Random 1	100	1	74,55	TL	-	EM	0	391,22
		2	81,82	TL	-	EM	0	391,00
		3	87,80	TL	-	EM	0	391,78
		4	-	EM	-	EM	0	391,90
		5	52,35	TL	-	EM	0	391,54
		6	-	EM	-	EM	0	391,47
Random 2	100	1	80,00	TL	-	EM	0	549,32
		2	81,82	TL	-	EM	0	549,38
		3	73,17	TL	-	EM	0	549,95
		4	-	EM	-	EM	0	550,03
		5	-	EM	23,33	TL	0	549,62
		6	-	EM	-	EM	0	549,63
Random 3	100	1	78,18	TL	-	EM	0	293,37
		2	81,82	TL	-	EM	0	293,33
		3	91,46	TL	-	EM	0	293,80
		4	-	EM	-	EM	0	293,73
		5	-	EM	-	EM	0	293,55
		6	-	EM	-	EM	0	293,55
Epinions 1	100	1	81,82	TL	-	EM	0	429,79
		2	83,64	TL	-	EM	0	429,72
		3	-	EM	-	EM	0	430,72
		4	51,22	TL	-	EM	0	430,14
		5	-	EM	-	EM	0	429,58
		6	-	EM	-	EM	0	429,83
#Viáveis / #Ótimas				13 / 0		1 / 0		24 / 24

O único modelo decomposto testado que conseguiu solução ótima no tempo limite em todas as instâncias, tanto em grafos não direcionados quanto em digrafos, foi $T\mathcal{F}_{d2}$ combinado com o algoritmo MM (*MinMatch*), para 25 vértices, ou o modelo $\mathcal{SP}_{MTZ}$, para 50 e 100 vértices. Buscando avaliar a proporção do tempo total de computação entre os dois subproblemas da decomposição, geramos o gráfico da Figura 10. Para todas as instâncias com uma mesma quantidade de vértices, exibimos o tempo médio para determinar os caminhos positivos mínimos entre os pares de vértices e gerar o grafo G' , via MM ou $\mathcal{SP}_{MTZ}$, e o tempo médio para alocar os indivíduos às equipes, via modelo $T\mathcal{F}_{d2}$. Vale mencionar que, na realidade, para as várias instâncias baseadas no mesmo grafo de sinais G , o grafo G' é construído uma única vez. Entretanto, o tempo para obtenção de G' é multiplicado pelo número dessas instâncias na construção do gráfico.

Perceba que, para grafos não direcionados ou direcionados com 25 ou 50 vértices, os subproblemas SPP's tendem a ter uma maior influência no tempo total de execução. Quando o número de vértices sobe para $|V| = 100$, isso ainda acontece para digrafos, mas já começa a haver um equilíbrio

entre as parcelas de tempo em grafos não direcionados. Lembre que no caso de digrafos o número de subproblemas SSP's resolvidas é o dobro daquele em grafos não direcionados.

Figura 10 – Médias dos tempos de computação, para cada quantidade de vértices, de cada subproblema da decomposição



Fonte: elaborado pelo autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, introduzimos o Problema de Formação de Equipes com Caminhos Positivos (TFP_SPP). Foram propostos dois modelos de PLI que se diferenciam pela tipo de restrição usada para a quebra de ciclos. Conseguimos mostrar que cada modelo é decomponível, gerando dois subproblemas distintos: o SPP e um problema muito similar ao TFP. Essa estratégia pode ser utilizada tanto quando o grafo de sinais de entrada é não direcionado como quando é direcionado. Revisitamos a literatura do SPP, apresentando diferentes formas de resolvê-lo, a depender do tipo de grafo de sinais de entrada. Para o segundo subproblema, similar ao TFP, derivamos algumas classes de desigualdades válidas que buscam melhorar a performance computacional do modelo matemático. Testes computacionais comprovaram que a estratégia de decomposição proposta é superior à resolução direta dos modelos de PLI originais, seja considerando o tempo de execução ou a obtenção de solução ótima no tempo limite. Uma melhoria adicional no tempo de execução e no número de soluções ótimas encontradas pelo modelo decomposto foi percebido com a utilização de algumas desigualdades válidas propostas.

Note que o **Modelo- $\mathcal{F}_{MTZ}$** e o **Modelo- $\mathcal{F}_{SIGN}$** exigem maior recurso de memória e tempo de computação, quando comparados a cada um dos modelos da decomposição separadamente, devido ao maior número de variáveis e restrições, conforme visto na Subseção 5.4.1. Acreditamos que esse seja um dos motivos pelos quais a metodologia de decomposição mostrou-se mais eficiente que a resolução direta dos modelos originais. Embora as instâncias de teste possuam diversidade na densidade de arestas (resp. arcos) do grafo de sinais, percebemos que esse fator não influenciou na conclusão da comparação entre as formulações originais e a estratégia de decomposição. Entre os modelos simplificados, observamos que as desigualdades do incluídas no modelo $T\mathcal{F}_{d1}$ podem deixá-lo pesado, à medida que aumenta o número de vértices. Por outro lado, as modificações feitas na versão $T\mathcal{F}_{d2}$ foram bem sucedidas, tonando-a significativamente superior a todos os outros modelos, tanto em relação a tempo de execução quanto à quantidade de instâncias resolvidas à otimalidade no tempo limite.

Entre possíveis trabalhos futuros, podemos citar a busca por métodos mais eficientes para resolver algum dos subproblemas da decomposição proposta. O SPP se configura, por si só, como um problema relevante. Note que o tempo para resolvê-lo, usando os métodos aqui considerados, consegue superar, na maioria dos casos com $|V| = 100$, o tempo para obter uma solução ótima do modelo $T\mathcal{F}_{d2}$. Com respeito a $T\mathcal{F}_{d2}$, embora as desigualdades válidas acrescentadas tenham reduzido significativamente o tempo de resolução, acreditamos que outras desigualdades ainda poderiam ser úteis, incluindo outras apresentadas nesta dissertação, quando usadas num processo bem delineado de planos-de-corte.

Vale ainda destacar como questão relevante para pesquisa futura o estudo de variações do TFP_SPP obtidas a partir de outras relações de *compatibilidade* entre indivíduos. Por exemplo, consideramos que uma versão interessante seria redefinir a distância de compatibilidade entre dois

indivíduos, tornando-a relativa a cada equipe, da seguinte forma: a distância entre u e v na equipe j seria o tamanho do menor u, v -caminho positivo tendo apenas vértices da equipe j como intermediários.

REFERÊNCIAS

- AGARWAL, S.; ADJIRACKOR, T. Impact of teamwork on organizational productivity in some selected basic schools in the accra metropolitan assembly. **European Journal of Business, Economics and Accountancy**, v. 4, n. 6, p. 40–52, 2016.
- ALBUQUERQUE, F.; CAMPÊLO, M.; FIGUEIREDO, T. O problema de formação de equipes com caminhos mínimos compatíveis: uma abordagem exata. In: **Anais do Simpósio Brasileiro de Pesquisa Operacional**. [S. l.: s. n.], 2022. v. 54, n. 152933, p. 2965–1476.
- ALBUQUERQUE, F.; CAMPÊLO, M.; FIGUEIREDO, T. O problema do caminho positivo mínimo. In: SBC. **Anais do VIII Encontro de Teoria da Computação (ETC 2023)**. [S. l.], 2023. p. 89–93.
- ARINIK, N.; FIGUEIREDO, R.; LABATUT, V. Signed graph analysis for the interpretation of voting behavior. **International Workshop on Recommender Systems and Social Network Analysis**, 2017.
- BARRICK, M. R.; STEWART, G. L.; NEUBERT, M. J.; MOUNT, M. K. Relating member ability and personality to work-team processes and team effectiveness. **Journal of applied psychology**, American Psychological Association, v. 83, n. 3, p. 377, 1998.
- BASFORD, T. E.; OFFERMANN, L. R. Beyond leadership: The impact of coworker relationships on employee motivation and intent to stay. **Journal of Management amp; Organization**, Cambridge University Press, v. 18, n. 6, p. 807–817, 2012.
- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear programming and network flows**. [S. l.]: John Wiley & Sons, 2011.
- BREGALDA, P. F.; OLIVEIRA, A. A. F. de; BORNSTEIN, C. T. **Introdução à programação linear**. [S. l.]: Campus, 1983.
- CAMPÊLO, M.; FIGUEIREDO, T.; SILVA, A. The sociotechnical teams formation problem: a mathematical optimization approach. **Annals of Operations Research**, v. 286, p. 201–216, 2020.
- CAMPÊLO, M.; FIGUEIREDO, T. F. Integer programming approaches to the multiple team formation problem. **Computers Operations Research**, v. 133, p. 105354, 2021. ISSN 0305-0548.
- CARTWRIGHT, D.; HARARY, F. Structural balance: a generalization of heider's theory. **Psychological Review**, v. 63, n. 5, p. 277–293, 1956.
- CHVATAL, V.; CHVATAL, V. *et al.* **Linear programming**. [S. l.]: Macmillan, 1983.
- COHEN, S. G.; BAILEY, D. E. What makes teams work: Group effectiveness research from the shop floor to the executive suite. **Journal of Management**, v. 23, n. 3, p. 239–290, 1997. ISSN 0149-2063. A Special Issue Of The Journal Of Management.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. 2nd. ed. [S. l.]: The MIT Press, 2001. ISBN 0262032937.
- CORNELISSEN, T. Do social interactions in the workplace lead to productivity spillover among co-workers? **IZA World of Labor**, 2016.
- COSTA, I. S.; FIGUEIREDO, R.; REQUEJO, C. The shortest path in signed graphs. In: ALMEIDA, J. P.; GERALDES, C. S.; LOPES, I. C.; MONIZ, S.; OLIVEIRA, J. F.; PINTO, A. A. (Ed.). **Operational Research**. Cham: Springer International Publishing, 2023. p. 53–71. ISBN 978-3-031-20788-4.
- DANTZIG, G. **Linear programming and extensions**. [S. l.]: Princeton university press, 1963.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959.

EDMONDS, J. Maximum matching and a polyhedron with 0, 1-vertices. **Journal of research of the National Bureau of Standards B**, v. 69, n. 125-130, p. 55–56, 1965.

EDMONDS, J. Paths, trees, and flowers. **Canadian Journal of mathematics**, Cambridge University Press, v. 17, p. 449–467, 1965.

FIGUEIREDO, T. F. **Team formation problems: an integer linear optimization approach**. Tese (Doutorado) – Federal University of Ceará, 2021.

FORTUNE, S.; HOPCROFT, J.; WYLLIE, J. The directed subgraph homeomorphism problem. **Theoretical Computer Science**, v. 10, n. 2, p. 111–121, 1980. ISSN 0304-3975.

GARCÍA-CAMPAYO, J.; PUEBLA-GUEDEA, M.; HERRERA-MERCADAL, P.; DAUDÉN, E. Burnout syndrome and demotivation among health care personnel. managing stressful situations: The importance of teamwork. **Actas Dermo-Sifiliográficas (English Edition)**, v. 107, n. 5, p. 400–406, 2016. ISSN 1578-2190.

GERARDS, A. Chapter 3 matching. In: **Network Models**. [S. l.]: Elsevier, 1995, (Handbooks in Operations Research and Management Science, v. 7). p. 135–224.

GULER, B.; VARAN, B.; TUTUNCUOGLU, K.; NAFEA, M.; ZEWAİL, A. A.; YENER, A.; OCTEAU, D. Optimal strategies for targeted influence in signed networks. In: **IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)**. [S. l.: s. n.], 2014. p. 906–911.

GULER, B.; VARAN, B.; TUTUNCUOGLU, K.; NAFEA, M.; ZEWAİL, A.; YENER, A.; OCTEAU, D. Using social sensors for influence propagation in networks with positive and negative relationships. **IEEE Journal of Selected Topics in Signal Processing**, v. 9, n. 2, p. 360–373, 2015.

GUMMER, B. Go team go! **Administration in Social Work**, Routledge, v. 19, n. 4, p. 85–100, 1996.

GUTIÉRREZ-BAHAMONDES, J.; ASTUDILLO, C.; BALLESTEROS-PÉREZ, P.; MORA-MELIÁ, D.; CANDIA-VEJAR, A. The multiple team formation problem using sociometry. **Computers and Operations**, v. 75, p. 150–162, 05 2016.

HANSEN, P. Shortest paths in signed graphs. In: BURKARD, R.; CUNINGHAME-GREEN, R.; ZIMMERMANN, U. (Ed.). **Algebraic and Combinatorial Methods in Operations Research**. [S. l.]: North-Holland, 1984, (North-Holland Mathematics Studies, v. 95). p. 201–214.

HARARY, F. On the notion of balance of a signed graph. **The Michigan Mathematical Journal**, The University of Michigan, v. 2, n. 2, p. 143–146, 1953.

HARARY, F.; KABELL, J. A. A simple algorithm to detect balance in signed graphs. **Mathematical Social Sciences**, v. 1, n. 1, p. 131–136, 1980.

HEIDER, F. Attitudes and cognitive organization. **The Journal of psychology**, Taylor & Francis, v. 21, n. 1, p. 107–112, 1946.

INOHARA, T. Clusterability of groups and information exchange in group decision making with approval voting system. **Applied Mathematics and Computation**, Elsevier, v. 136, n. 1, p. 1–15, 2003.

KOUVATIS, I.; SEMERTZIDIS, K.; ZERVA, M.; PITOURA, E.; TSAPARAS, P. Forming compatible teams in signed networks. In: **Proceedings of the 23rd International Conference on Extending Database Technology (EDBT)**. [S. l.]: OpenProceedings.org, 2020. p. 363–366.

- LAPAUGH, A. S.; PAPADIMITRIOU, C. H. The even-path problem for graphs and digraphs. **Networks**, v. 14, n. 4, p. 507–513, 1984.
- LAPPAS, T.; LIU, K.; TERZI, E. Finding a team of experts in social networks. In: . New York, NY, USA: Association for Computing Machinery, 2009. (KDD '09), p. 467–476. ISBN 9781605584959.
- LEONARD, M.; GRAHAM, S.; BONACUM, D. The human factor: the critical importance of effective teamwork and communication in providing safe care. **BMJ Quality & Safety**, BMJ Publishing Group Ltd, v. 13, n. suppl 1, p. i85–i90, 2004.
- LI, Y.; CHEN, W.; WANG, Y.; ZHANG, Z.-L. Voter model on signed social networks. **Internet Mathematics**, Taylor & Francis, v. 11, n. 2, p. 93–133, 2015.
- MICALI, S.; VAZIRANI, V. V. An $O(|V|^{1/2}|E|)$ algoithm for finding maximum matching in general graphs. In: **21st Annual Symposium on Foundations of Computer Science (SFCS 1980)**. [S. l.: s. n.], 1980. p. 17–27.
- MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 7, n. 4, 1960.
- NEUMAN, G. A.; WAGNER, S. H.; CHRISTIANSEN, N. D. The relationship between work-team personality composition and the job performance of teams. **Group & Organization Management**, v. 24, n. 1, p. 28–45, 1999.
- SCHRIJVER, A. **Theory of linear and integer programming**. [S. l.]: John Wiley & Sons, 1998.
- SHI, G.; PROUTIERE, A.; JOHANSSON, M.; BARAS, J. S.; JOHANSSON, K. H. The evolution of beliefs over signed social networks. **Operations Research**, INFORMS, v. 64, n. 3, p. 585–604, 2016.
- UMBERSON, D.; MONTEZ, J. K. Social relationships and health: a flashpoint for health policy. **Journal of health and social behavior**, v. 51 Suppl, p. S54–66, 2010.
- WEST, D. B. **Introduction to Graph Theory**. 2nd. ed. [S. l.]: Pearson, 2000.
- WOLSEY, L. A. **Integer programming**. [S. l.]: John Wiley & Sons, 2020.