



**UNIVERSIDADE FEDERAL DO CEARÁ  
CENTRO DE CIÊNCIAS  
DEPARTAMENTO DE COMPUTAÇÃO  
PROGRAMA DE MESTRADO E DOUTORADO EM CIÊNCIA DA  
COMPUTAÇÃO**

**Tatiane Fernandes Figueiredo**

**Problema de Formação de Equipes Sociotécnicas: Complexidade, Formulações  
Matemáticas e Resultados Computacionais**

**FORTALEZA – CE  
Agosto de 2014**

Tatiane Fernandes Figueiredo

Problema de Formação de Equipes Sociotécnicas: Complexidade, Formulações  
Matemáticas e Resultados Computacionais

Dissertação de Mestrado apresentada ao Programa de Mestrado e Doutorado em Ciência da Computação, do Departamento de Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação. Área de concentração: Otimização (Teoria da Computação).

Orientador: Prof. Dr. Manoel Bezerra  
Campêlo Neto  
Coorientador: Prof. Dra. Ana Shirley Fer-  
reira da Silva

FORTALEZA – CE

Agosto de 2014

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Ciências e Tecnologia

- 
- F493p Figueiredo, Tatiane Fernandes.  
Problema de formação de equipes sociotécnicas: complexidade, formulações matemáticas e resultados computacionais / Tatiane Fernandes Figueiredo. – 2014.  
86 f. : il. color., ; 30 cm.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2014.  
Área de Concentração: Otimização (Teoria da Computação).  
Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.  
Coorientação: Prof. Dra. Ana Shirley Ferreira da Silva.
1. Administração de pessoal. 2. Programação inteira. 3. Heurística. I. Título.

TATIANE FERNANDES FIGUEIREDO

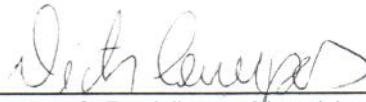
**Problema de Formação de Equipes Sociotécnicas: Complexidade,  
Formulações Matemáticas e Resultados Computacionais**

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como requisito para a obtenção do grau de Mestre em Ciência da Computação.

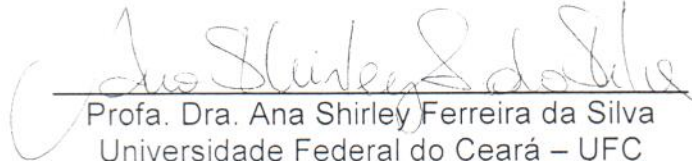
BANCA EXAMINADORA



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



Prof. Dr. Victor Almeida Campos  
Universidade Federal do Ceará – UFC



Profa. Dra. Ana Shirley Ferreira da Silva  
Universidade Federal do Ceará – UFC



Prof. Dr. Carlos Diêgo Rodrigues  
Universidade Federal do Ceará – UFC



Prof. Dr. Humberto César Brandão de Oliveira  
Universidade Federal De Alfenas – UNIFAL

Fortaleza, 14 de agosto de 2014

À minha irmã, Taciana (*in memoriam*).

# Agradecimentos

Ao meus pais e à minha irmã, Taciana (*in memoriam*), por todo incentivo e apoio, sendo sempre minha inspiração e motivação para seguir em frente.

Ao meus queridos mineiros: Paulo José, Suelen Resende, Tiago Silveira, Bruno Bruck e Ana Luiza por me ajudarem a superar desafios que a vida me impôs no decorrer destes anos de estudo e que, às vezes, pareciam impossíveis. Em especial, agradeço à minha melhor amiga, Mariana Dehon, por estar presente em todas as etapas deste mestrado e da minha vida, sempre me apoiando incondicionalmente.

Aos meus queridos cearenses: Rennan, Marcio, Vinícius, Fábio, Luiz Alberto, Cláudio, Eliezer, Arthur, Pablo, Paulo Henrique, Nicolas, Eurinaldo, Jefferson e André por terem me dado oportunidade de fazer parte da suas vidas e me presentear com a sua amizade se tornando minha segunda família no decorrer dos últimos anos. Em especial, ao meu melhor amigo Rafael Teixeira pela lealdade, amizade e por estar sempre ao meu lado.

À Samuel Nascimento pela companhia nos inúmeros finais de semana de trabalho, por estar sempre disposto a ouvir sobre "problemas de agrupamento", pelo apoio e carinho.

Ao professor Manoel Campêlo, orientador deste trabalho, que por muitas vezes foi além, sendo conselheiro e motivador, tornando-se uma referência para mim. Muito obrigada pela enorme dedicação e empenho, pela excelência didática e conhecimentos transmitidos, pela paciência e pela confiança em mim depositada para a realização deste trabalho.

À professora Ana Shirley Silva, coorientadora deste trabalho, pela dedicação e empenho constante em melhorar o meu aprendizado, estando sempre disponível para minhas inúmeras solicitações de ajuda.

Ao professor Carlos Diego Rodrigues, pelas fundamentais sugestões, revisões e incentivos, estando presente durante todo o decorrer deste trabalho.

Ao professor Humberto César Brandão por apresentar o problema de formação de equipes sociais que deu origem a este trabalho e pelos vários anos de amizade, apoio e conhecimentos transmitidos.

Ao professor Victor Campos, pela contruibuição para meu desenvolvimento intelectual, e por aceitar o convite para participar da banca avaliadora deste trabalho.

Aos professores do grupo de pesquisa ParGO, por serem acolhedores, pelo conhecimento transmitido e por todos os momentos de descontração promovidos. Em especial agradeço ao professor Rudini Sampaio e à professora Cláudia Linhares.

Aos pesquisadores do STIC-AmdSud: Javier Marengo, Diego Delle Donne, Lorena Pradenãs, Bertrand Le Cun e Philippe Michelin pela grande contribuição neste trabalho.

Aos amigos e funcionários do MDCC/UFC. Em especial, aos funcionários: Marcos Souza e Orley Carneiro.

À FUNCAP pelo apoio financeiro.

*“Se eu vi mais longe, foi por estar de pé sobre ombros de gigantes”.*

*(Isaac Newton)*

# Resumo

Utilizando conceitos da Teoria dos Sistemas Sociotécnicos, este trabalho define matematicamente os problemas de formação de equipes cooperativas considerando separadamente restrições sociais e técnicas e apresenta a complexidade computacional dos mesmos. Sobretudo, é definido e estudado o problema central deste trabalho, que considera conjuntamente requisitos sociais e técnicos para criação de equipes de trabalho cooperativo, denominado FEST (Problema de Formação de Equipes Sociotécnicas). Duas formulações matemáticas e uma meta-heurística para o FEST são propostas. Uma formulação utiliza um número cúbico de variáveis e restrições, enquanto a segunda formulação possui um número quadrático de variáveis, mas um número exponencial de restrições. A meta-heurística proposta é baseada no *Simulated Annealing* Não-Monotônico com busca local que usa operadores tipo *swap*. A corretude de ambas as formulações é provada. Um algoritmo polinomial para separar as restrições da segunda formulação é apresentado. Mostra-se que as duas formulações fornecem o mesmo limite de programação linear, e desigualdades válidas para fortalecê-lo são propostas. Para a formulação compacta, algumas classes de desigualdades válidas são demonstradas indutoras de facetes sob hipóteses apropriadas. Por fim, foi analisado estatisticamente o desempenho das formulações e da meta-heurística apresentadas. Instâncias reais e geradas aleatoriamente são usadas nos experimentos computacionais.

**Palavras-chave:** trabalho cooperativo, gerência de pessoas, complexidade computacional, programação inteira, combinatória poliédrica, heurística.



# Abstract

Using concepts of the socio-technical systems theory, this dissertation defines mathematically the problems of cooperative teams formation considering social and technical constraints separately, and then presents their computational complexity. Mainly, it is defined and studied the central problem in this work, which jointly considers social and technical requirements for creating teams of cooperative work, to be called FEST (Socio-Technical Teams Formation Problem). Two mathematical formulations and a meta-heuristic are proposed for FEST. One formulation uses a cubic number of variables and constraints, whereas the second one has a quadratic number of variables but an exponential number of constraints. The proposed heuristic is based on the Non-monotonic Simulated Annealing meta-heuristic with local search using swap-like operators. The correctness of both formulations is proved. A polynomial algorithm to separate the constraints of the second formulation is presented. It is proved that the two formulations provide the same linear programming bound, and valid inequalities to strengthen it are proposed. For the compact formulation, some classes of valid inequalities are shown to be facet-inducing under suitable hypotheses. Finally, it is statistically analyzed the performance of the presented formulations and meta-heuristic. Real and random generated instances are used in the computational experiments.

**Key-words:** cooperative work, people management, computational complexity, integer programming, polyhedral combinatorics, heuristic.

# Sumário

	<b>Sumário</b>	<b>9</b>
	<b>Lista de ilustrações</b>	<b>11</b>
<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>CONCEITOS BÁSICOS</b>	<b>14</b>
2.1	Teoria dos grafos	14
2.2	Teoria poliédrica	17
2.3	Programação Inteira	19
<b>3</b>	<b>FORMAÇÃO DE EQUIPES DE TRABALHO</b>	<b>21</b>
3.1	O problema de formação de equipas técnicas	21
3.2	O problema de formação de equipas sociais	25
<b>4</b>	<b>O PROBLEMA DE FORMAÇÃO DE EQUIPES SOCIOTÉCNICAS</b>	<b>28</b>
4.1	Definição do problema	28
4.2	Complexidade do problema	29
4.3	Limite superior	30
4.3.1	Heurística baseada em Simulated Annealing Não-Monotônico	30
4.3.1.1	Controle da temperatura	30
4.3.1.2	Construção da solução inicial	31
4.3.1.3	Busca por solução vizinha	31
4.3.2	Corretude do algoritmo	33
<b>5</b>	<b>FORMULAÇÕES MATEMÁTICAS</b>	<b>35</b>
<b>5.1</b>	<b>Formulação 1</b>	<b>35</b>
<b>5.2</b>	<b>Estudo poliédrico da Formulação 1</b>	<b>39</b>
5.2.1	Dimensão do politopo	39
5.2.2	Estudo facial	42
5.2.3	Desigualdades válidas	48
<b>5.3</b>	<b>Formulação 2</b>	<b>50</b>
5.3.1	Algoritmo de separação	52
5.3.2	Desigualdades válidas	55
<b>5.4</b>	<b>Comparação entre as formulações</b>	<b>58</b>
<b>6</b>	<b>RESULTADOS COMPUTACIONAIS</b>	<b>60</b>

---

<b>6.1</b>	<b>Configuração do ambiente computacional</b> . . . . .	<b>60</b>
<b>6.2</b>	<b>Versões implementadas</b> . . . . .	<b>60</b>
<b>6.3</b>	<b>Instâncias e suas principais características</b> . . . . .	<b>62</b>
6.3.1	Instâncias reais . . . . .	62
6.3.2	Geração de instâncias aleatórias . . . . .	64
<b>6.4</b>	<b>Estudo comparativo entre as formulações</b> . . . . .	<b>65</b>
6.4.1	Resolução do problema inteiro . . . . .	65
6.4.2	Limite inferior . . . . .	74
6.4.3	Limite superior . . . . .	78
6.4.4	Resolução do problema inteiro com limites . . . . .	79
6.4.5	Instâncias reais . . . . .	80
<b>7</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	<b>82</b>
	<b>Referências</b> . . . . .	<b>85</b>

# Lista de ilustrações

Figura 1 – Rede $F(I)$ gerada a partir de uma instância $I$ do $FET$ . Solução para o $PMF$ na rede $F(I)$ destacada pelas arestas mais escuras. . . . .	23
Figura 2 – Exemplo de um sociograma (grafo de afinidades) . . . . .	25
Figura 3 – Grafo bipartido $H = (U \cup W, E(H))$ para $U = \{v_1, v_2, v_3, v_4\}$ , $T = \{1, 2\}$ e $S = \{1, 2\}$ , sendo $s(v_1) = \{1, 2\}$ , $s(v_2) = \{1\}$ , $s(v_3) = s(v_4) = \{2\}$ , $t(1, 1) = t(1, 2) = 1$ , $t(2, 1) = 0$ e $t(2, 2) = 2$ . . . . .	40
Figura 4 – Grafo bipartido $G_j = (U_j \cup W_j, E_j)$ , onde $p =  U_j $ e $q =  W_j $ . . . . .	52
Figura 5 – Rede $G_j(Y, Z)$ , onde $p =  U(j) $ e $q =  s(j) $ . . . . .	53
Figura 6 – Ilustração do corte (— — — arco inexistente; —×××— arco sem fluxo) . . . . .	55
Figura 7 – Rede $F_j(Y, Z)$ , onde $p =  U(j) $ e $q =  s(j) $ . . . . .	57
Figura 8 – tempo de execução (segundos) × instância do grupo A . . . . .	70
Figura 9 – tempo de execução (segundos) × instância do grupo B . . . . .	70
Figura 10 – tempo de execução (segundos) × instância do grupo C . . . . .	71
Figura 11 – Gráfico com as diferenças das médias para os grupos A, B, e C . . . . .	72
Figura 12 – GAP × instância-teste . . . . .	77
Figura 13 – GAP × instância-teste . . . . .	77

# 1 Introdução

Um dos principais mecanismos organizacionais para a melhoria da competitividade é a adequação da estrutura da organização ao seu foco de atenção, isto é, à missão que se pretende atingir, buscando a obtenção de vantagens advindas da diferenciação estrutural.

Adaptações nas estruturas organizacionais impactam sensivelmente a forma como o trabalho é organizado. Uma das alternativas de adaptação estrutural é a formação de equipes de trabalho cooperativo. Como colocado por (WELLINS; BYHAM; DIXON, 1995), a implantação das equipes de trabalho cooperativo torna-se uma das peças centrais para a flexibilização do processo produtivo. Porém, diversos fatores podem alterar o desempenho destas equipes: distribuição adequada de cargos e tarefas, fatores psicológicos, físicos, de convívio, dentre outras (MARTELETO, 2001). Enquanto fatores psicológicos e físicos são dificilmente previsíveis, fatores como o convívio e distribuição de tarefas podem ser avaliados através de metodologias estudadas por pesquisadores da área de gestão de pessoas.

Equipes harmoniosas e competentes trabalham de forma mais dinâmica, sobressaindo-se em requisitos como facilidade de comunicação, integração de esforços e troca de conhecimentos entre os integrantes. (TRIST, 1981) acredita que os problemas de desempenho de equipes de trabalho cooperativo dependem do grau de correspondência entre os indivíduos participantes da organização e as propriedades das atividades desempenhadas. Este modelo é conhecido como Teoria dos Sistemas Sociotécnicos e será a base do trabalho aqui apresentado.

A partir da análise do sistema sociotécnico da organização, este trabalho propõe formulações matemáticas para o problema de formação de equipes, com a finalidade de auxiliar gestores na formação de equipes de trabalho cooperativo. Utilizando dados da rede social da organização e dados profissionais dos indivíduos envolvidos, as formulações buscam por soluções (equipes) que obedecem às restrições profissionais impostas, isto é, restrições de demanda de habilidades profissionais para cada equipe, procurando também maximizar os relacionamentos considerados afetivos entre membros de uma mesma equipe.

O estudo realizado nesta dissertação é apresentado nos próximos 5 capítulos, a seguir descritos. No Capítulo 3, definimos matematicamente os problemas de formação de equipes de trabalho considerando, separadamente, restrições técnicas e sociais. Mostramos a complexidade computacional dos mesmos. O problema principal, foco deste trabalho, é definido no Capítulo 4. Ele considera conjuntamente requisitos técnicos e de relacionamentos para a definição de equipes de trabalho e será chamado FEST - Problema de Formação de Equipes Sociotécnicas. Formulamos matematicamente o problema e estudamos sua complexidade computacional. Ainda no Capítulo 4, caracterizamos uma solução viável de FEST e propomos uma meta-heurística.

No Capítulo 5, apresentamos duas formulações matemáticas para o problema: uma é compacta, com número cúbico de variáveis e restrições, e outra possui um número quadrático de variáveis, mas exponencial de restrições. Mostramos a corretude de ambas as formulações e apresentamos um algoritmo polinomial para separar as restrições da segunda. Propomos também desigualdades válidas para fortalecer ambas as formulações e mostramos que algumas delas induzem facetes.

Os experimentos computacionais com as formulações são o foco do Capítulo 6. Obtemos instâncias reais, a partir da aplicação de questionários sociotécnicos, e geramos outras com um gerador de instâncias pseudo-aleatório que propomos, procurando obter uma boa diversidade de problemas-teste. Implementamos as formulações no solver CPLEX e comparamos seus desempenhos computacionais.

Apresentamos nossas conclusões e perspectivas de trabalhos futuros no Capítulo 7. Antes de tudo, no próximo capítulo, apresentamos alguns conceitos, propriedades e notações que serão usadas ao longo do texto.

## 2 Conceitos Básicos

Neste capítulo, são apresentados alguns conceitos relacionados a teoria dos grafos, teoria poliédrica e programação inteira, que serão utilizados ao longo do trabalho. Descrevemos também uma notação geral, consistente com a que é frequentemente usada nessas áreas. Notações específicas, relacionadas ao nosso desenvolvimento, serão definidas em momentos mais oportunos do texto.

Na Seção 2.1, é apresentada a terminologia utilizada para grafos, seguindo (BONDY; MURTY, 1976). Nas seções 2.2 e 2.3, são apresentadas as notações e conceitos relativos a poliedros e programação inteira, com base em (NEMHAUSER; WOLSEY, 1988). Mais detalhes podem ser encontrados nessas referências.

### 2.1 Teoria dos grafos

Um *grafo*  $G = (V(G), E(G))$  consiste de um conjunto finito não vazio  $V(G)$  de elementos, chamados *vértices*, e um conjunto de pares não-ordenados de elementos distintos de  $V(G)$ , chamado *arestas*, isto é,  $E(G) \subseteq \{\{u, v\} | u, v \in V(G), u \neq v\}$ . Algumas vezes, quando o grafo estiver claro pelo contexto, utilizamos  $V$  e  $E$  para  $V(G)$  e  $E(G)$  respectivamente, para simplificar a notação. O *grafo complementar de  $G$*  é o grafo  $\bar{G} = (V, \bar{E})$ , onde  $\bar{E} = \{\{u, v\} | u, v \in V(G), u \neq v\} \setminus E(G)$ .

Se  $e = \{u, v\}$  é uma aresta, dizemos que  $e$  *incide em  $u$  e em  $v$* ; que  $u$  e  $v$  *são seus extremos*; e que  $u$  e  $v$  *são adjacentes*. Para simplificar a notação, denotamos uma aresta  $e = \{u, v\}$  por  $uv$ .

A *vizinhança* ou *adjacência* de um vértice  $v$  em um grafo  $G$  é o conjunto contendo todos os vértices adjacentes a  $v$ . A vizinhança é frequentemente denotada  $N_G(v)$  ou simplesmente  $N(v)$  quando o grafo está claro pelo contexto. A vizinhança de um subconjunto de vértices  $W \subseteq V$  é  $N(W) = \bigcup_{v \in W} N(v)$ .

Dizemos que um grafo é *completo* se quaisquer dois de seus vértices são adjacentes. Se  $V'$  é um subconjunto de vértices em  $V(G)$ , denotamos por  $\delta(V')$  o conjunto de arestas em  $E(G)$  com um extremo em  $V'$  e o outro em  $V \setminus V'$ . Um conjunto da forma  $\delta(V')$ , onde  $\emptyset \neq V' \subsetneq V$ , é dito um *corte* em  $G$ . Se  $v'$  é um vértice em  $V'$  e  $v$  um vértice em  $V \setminus V'$ , então dizemos que  $\delta(V')$  é um  $(v', v)$ -*corte*.

Se  $G$  e  $G'$  são dois grafos tais que  $V(G') \subseteq V(G)$  e  $E(G') \subseteq E(G)$ , então  $G'$  é dito um *subgrafo* de  $G$  e escrevemos  $G' \subseteq G$ . Se  $V(G) = V(G')$  então dizemos que  $G'$  é um *subgrafo gerador de  $G$* . Se  $W \subseteq E(G)$ , então o subgrafo de  $G$  com o conjunto de arestas  $W$  e vértices consistindo dos extremos das arestas de  $W$  é o *subgrafo gerado ou induzido por  $W$* . Similarmente, se  $W \subseteq V(G)$ , então o subgrafo de  $G$  formado pelos vértices em

$W$  e as arestas de  $G$  com ambas as extremidades em  $W$  é o *subgrafo gerado ou induzido por  $W$* . Em ambos os casos, tal subgrafo é denotado  $G[W]$ .

Se  $v$  é um vértice em  $V(G)$ , o subgrafo de  $G$  obtido com a remoção deste vértice (e das arestas a ele incidentes) é denotado por  $G[V \setminus \{v\}]$ . A mesma notação é utilizada para um subconjunto qualquer de vértices. Se  $W \subseteq V(G)$ , então  $G[V \setminus W]$  representa a operação de remoção de cada vértice  $v \in W$  (e das arestas incidentes a  $v$ ) do grafo  $G$ . Uma *junção*  $G = G_1 + G_2$  dos grafos  $G_1$  e  $G_2$ , com conjuntos disjuntos de vértices  $V_1$  e  $V_2$  e conjuntos de arestas  $E_1$  e  $E_2$ , é o grafo obtido pela união de  $G_1$  e  $G_2$  com todas as arestas entre  $V_1$  e  $V_2$ , ou seja, o grafo  $G = (V_1 \cup V_2, E_1 \cup E_2 \cup (V_1 \times V_2))$ .

Um *emparelhamento* em um grafo  $G$  é um conjunto de arestas não adjacentes entre si, isto é, arestas sem extremos comuns. Se um vértice  $v$  é extremo de alguma aresta de um emparelhamento  $M$ , então dizemos que  $M$  *cobre*  $v$ , ou que  $v$  é *coberto por  $M$* . Dizemos que um emparelhamento  $M$  é *perfeito* se cobre todos os vértices do grafo.

Um emparelhamento  $M$  é dito *maximal* se não existe um emparelhamento  $M'$  que contenha propriamente  $M$ , e é *máximo* se não existe um emparelhamento com maior número de arestas. Se  $M$  é um emparelhamento em  $G$ , então um  $(s, t)$ -caminho em  $G$  cujas arestas pertencem alternadamente a  $M$  e a  $E(G) \setminus M$  (ao percorrê-lo de  $s$  a  $t$ ) é chamado *alternante* (em relação a  $M$ ), ou  *$M$ -alternante*. Um caminho alternante em um emparelhamento  $M$  é chamado *augmentante* (em relação a  $M$ ) ou  *$M$ -augmentante*, se começa e termina num vértice não-coberto e tem pelo menos uma aresta.

Dizemos que um grafo  $G(V, E)$  é *bipartido* se  $V$  pode ser particionado em dois conjuntos disjuntos não vazios  $V_1$  e  $V_2$  e toda aresta em  $G$  tem um extremo em  $V_1$  e outro em  $V_2$ . Designamos por  $G[X, Y]$  um grafo bipartido com partição  $V(G) = X \cup Y$  do conjunto dos vértices. Problemas relacionados com emparelhamentos em grafos bipartidos têm interesse especial e também resultados específicos que convém mencionar.

**Teorema 2.1.** (Hall): Dado  $G = (V_1 \cup V_2, E)$ , existe um emparelhamento que cobre  $V_1$  se e somente se  $|N(S)| \geq |S|$  para todo  $S \subseteq V_1$

**Corolário 2.2.** Um grafo bipartido  $G[X, Y]$  tem emparelhamento perfeito se e somente se  $|X| = |Y|$  e  $|N(S)| \geq |S|$ , para todo  $S \subseteq X$ .

Um *conjunto independente* em  $G$  é um conjunto  $S \subseteq V(G)$  tal que nenhuma aresta de  $G$  possui ambas as extremidades em  $S$ . Uma *clique* em  $G$  é um subconjunto  $C \subseteq V(G)$  tal que quaisquer dois vértices em  $C$  são adjacentes em  $G$ . Assim,  $S$  é um conjunto independente de  $G$  se e somente se  $S$  é uma clique de  $\bar{G}$ . Uma  *$k$ -coloração* de vértices de  $G$  é uma partição  $\Pi$  de  $V(G)$  em conjuntos independentes  $\{S_1, \dots, S_k\}$ . Os conjuntos  $\{S_1, \dots, S_k\}$  são chamados *classes de cores* da coloração.

Um *grafo orientado (digrafo)*  $D = (V, E)$  consiste de um conjunto finito não-vazio  $V(D)$  de elementos, chamados *vértices* ou *nós*, e um conjunto  $E(D)$  de elementos, chamados *arcos* ou *arestas orientadas*, que são pares ordenados de vértices distintos, isto é,



$E(D) \subseteq \{(u, v) | u, v \in V(D), u \neq v\}$ . Se  $a = (u, v)$  é um arco, dizemos que  $a$  incide em  $u$  e em  $v$ ; que  $u$  e  $v$  são seus extremos. Também nos referimos a  $u$  como *início* ou *origem* e a  $v$  como *término* ou *destino* do arco  $a$ .

Dado um vértice  $v \in V(D)$  denotamos por  $\delta^+(v)$  o conjunto de arcos que possuem o vértice  $v$  como início e  $\delta^-(v)$  o conjunto de arcos que possuem o vértice  $v$  como término. Se  $\delta^-(v) = \emptyset$ ,  $v$  é dito uma *fonte*; Se  $\delta^+(v) = \emptyset$ ,  $v$  é dito um *sumidouro*. Se  $V'$  é um subconjunto de vértices em  $V(D)$ , denotamos por  $\delta^+(V')$  o conjunto dos arcos de  $D$  com início em  $V'$  e término em  $V \setminus V'$ . Um conjunto da forma  $\delta^+(V')$ , onde  $\emptyset \neq V' \subsetneq V$ , é dito um *corte* em  $D$ . Se  $v'$  é um vértice em  $V'$  e  $v$  um vértice em  $V \setminus V'$ , então dizemos que  $\delta^+(V')$  é um  $(v', v)$ -*corte*. Usamos também a notação  $[V', V \setminus V']$  para representar um corte  $\delta^+(V')$ .

Todos os conceitos definidos para grafos que não envolvem a noção de orientação se aplicam analogamente para os grafos orientados. Embora usemos preferencialmente a notação  $D = (V, A)$  para um grafo orientado, vamos usar também  $G = (V, E)$ , quando não houver prejuízo para o entendimento, e neste caso  $E$  será um conjunto de arcos.

Uma *rede* é representada por um grafo orientado, onde, para cada arco ou nó pertencente à rede, é associado um valor numérico. Assim, uma rede  $R = (D = (V, A), c)$  consiste de um digrafo  $D$  e uma função  $c : A(D) \rightarrow \mathbb{R}$  associada aos arcos  $A(D)$  e/ou uma função  $c : V(D) \rightarrow \mathbb{R}$  associada aos nós  $V(D)$ . Por abuso de linguagem, vamos dizer que o grafo orientado  $D$  é uma rede, ficando subentendida a existência da função.

Dados uma rede  $D = (V, A)$  com função de capacidade  $c : A(D) \rightarrow \mathbb{R}^+$ , uma fonte  $s \in V(D)$  e um sumidouro  $t \in V(D)$ , um *fluxo* em  $D$  entre  $s$  e  $t$  é uma função  $f : A(D) \rightarrow \mathbb{R}^+$  que associa a cada arco  $e \in A(D)$  um número real não-negativo  $f(e)$  tal que:

$$f(e) \leq c(e), \text{ para cada arco } e \in A(D) \text{ (preservação de capacidade)}$$

$$\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e), \text{ para cada nó } v \in V(D) \setminus \{s, t\} \text{ (preservação de fluxo)}$$

O *valor do fluxo*  $f$  é  $|f| = \sum_{e \in \delta^+(s)} f(e)$  ou, equivalentemente,  $|f| = \sum_{e \in \delta^-(t)} f(e)$ . A capacidade de um  $(s, t)$ -corte  $\delta^+(S)$  é  $c(\delta^+(S)) = \sum_{e \in \delta^+(S)} c(e)$ .

**Teorema 2.3.** *Dados uma rede  $D = (V, A)$  com função de capacidade  $c : A(D) \rightarrow \mathbb{Q}^+$ , uma fonte  $s \in V(D)$  e um sumidouro  $t \in V(D)$ , o máximo valor de um fluxo entre  $s$  e  $t$  é igual a mínima capacidade de um  $(s, t)$ -corte.*

Dados uma rede  $D = (V, A)$  com função de capacidade  $c : A(D) \rightarrow \mathbb{R}^+$  e função de oferta/demanda  $b : V(D) \rightarrow \mathbb{R}^+$ , um *fluxo viável* em  $D$  é uma função  $f : A(D) \rightarrow \mathbb{R}^+$  que associa a cada arco  $e \in A(D)$  um número real não-negativo  $f(e)$  tal que:

$$f(e) \leq c(e), \text{ para cada arco } e \in A(D) \text{ (preservação de capacidade)}$$

$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v), \text{ para cada nó } v \in V(D) \text{ (preservação de fluxo)}$$

Nesta definição, consideramos que  $\sum_{v \in V} b(v) = 0$ . Dependendo se  $b(v) > 0$ ,  $b(v) < 0$  ou  $b(v) = 0$ , dizemos que  $v$  é um *nó de oferta*, *nó de demanda* ou *nó de repasse*, respectivamente.

A seguinte generalização do Teorema de Hall é apresentada em (AHUJA; MAGNANTI; ORLIN, 1993, Teorema 6.12).

**Teorema 2.4.** *Dados uma rede  $D = (V, A)$  com função de capacidade  $c : A(D) \rightarrow \mathbb{Q}^+$  e função de oferta/demanda  $b : V(D) \rightarrow \mathbb{Q}^+$ , existe um fluxo viável em  $D$  se, e somente se,  $b(S) \leq c(\delta^+(S))$ , para todo  $\emptyset \neq S \subsetneq V(D)$ , onde  $b(S) = \sum_{v \in S} b(v)$ .*

## 2.2 Teoria poliédrica

A menos de menção em contrário, os vetores aqui considerados são vetores-coluna. Assim, se  $x$  é um vetor, então  $x^T$  denota seu transposto (um vetor-linha). O *produto interno de dois vetores*  $x, y \in \mathbb{R}^n$  é o escalar  $x^T y := \sum_{i=1}^n x_i y_i$ . Um vetor que possui todas as componentes iguais a 1 é denotado por  $\mathbf{1}$ . Um vetor de zeros é denotado por  $\mathbf{0}$ . Um vetor  $x \in \mathbb{R}^n$  é referido como *unitário* se existe  $1 \leq i \leq n$ , tal que  $x_i = 1$  e  $x_j = 0 \forall j \neq i$ . Utilizamos a notação  $e^i$  para designar o vetor unitário cuja componente 1 está na posição  $i$ .

Dado um subconjunto  $S \subseteq \mathbb{R}^n$ , um vetor  $x \in \mathbb{R}^n$  é uma *combinação linear* dos vetores  $x^1, x^2, \dots, x^t$  em  $S$  se  $x = \sum_{i=1}^t \lambda_i x^i$  para algum  $(\lambda_1, \lambda_2, \dots, \lambda_t)^T \in \mathbb{R}^t$ . Uma tal combinação linear é chamada *afim*, se  $\lambda_1 + \dots + \lambda_t = 1$ ; *cônica*, se  $\lambda_1, \dots, \lambda_t \geq 0$ ; *convexa*, se for afim e cônica. Definimos o *fecho linear* de  $S$ , denotado por  $\text{span}(S)$ , como o conjunto de todos os vetores que são combinação linear de um número finito de vetores em  $S$ . Analogamente, definimos os *fechos afim, cônico e convexo*, denotados por  $\text{afim}(S)$ ,  $\text{cone}(S)$  e  $\text{conv}(S)$ , respectivamente. O  $\text{conv}(S)$  é também conhecido como a *envoltória convexa* de  $S$ .

Um conjunto de vetores  $x^1, \dots, x^k \in \mathbb{R}^n$  é *linearmente independente* se a única solução de  $\sum_{i=1}^k \lambda_i x^i = 0$  é  $\lambda = \mathbf{0}$ . Um conjunto de vetores  $x^1, \dots, x^k \in \mathbb{R}^n$  é *afim independente* se a única solução de  $\sum_{i=1}^k \lambda_i x^i = 0$ ,  $\sum_{i=1}^k \lambda_i = 0$  é  $\lambda = \mathbf{0}$ .

Dado  $S \subseteq \mathbb{R}^n$ , o *posto* de  $S$ , denotado por  $\text{posto}(S)$ , é a cardinalidade de um maior subconjunto de  $S$  que é linearmente independente. Analogamente, o *posto-afim* de  $S$ , denotado por  $\text{posto-afim}(S)$ , é a cardinalidade de um maior subconjunto de  $S$  afim independente. É conhecido que, para todo  $S \subseteq \mathbb{R}^n$ , se  $\mathbf{0} \in \text{afim}(S)$ , então  $\text{posto-afim}(S) = \text{posto}(S) + 1$ ; caso contrário  $\text{posto-afim}(S) = \text{posto}(S)$ . A dimensão de  $S$ , denotada por  $\text{dim}(S)$ , é definida como  $\text{dim}(S) := \text{posto-afim}(S) - 1$ . Dizemos que  $S \subseteq \mathbb{R}^n$  tem *dimensão plena*, se  $\text{dim}(S) = n$ .

O *posto de uma matriz*  $A \in \mathbb{R}^{m \times n}$ , denotado por  $\text{posto}(A)$ , é o posto do conjunto de vetores-coluna de  $A$ , que prova-se ser igual ao posto do conjunto de vetores-linha de  $A$ .

Dizemos que  $A$  tem *posto-linha completo*, se  $\text{posto}(A) = m$ ; e tem *posto-coluna completo* se  $\text{posto}(A) = n$ . O posto da matriz nula é zero.

Um *poliedro* é um conjunto  $P \subseteq \mathbb{R}^n$  de vetores definidos por um número finito de desigualdades lineares, i.e.  $P = \{x \in \mathbb{R}^n | Ax \leq b\}$ , onde  $(A, b)$  é uma matriz  $m \times (n + 1)$ , que é chamada uma *representação* de  $P$ . A submatriz  $(A^-, b^-)$  com maior número de linhas de  $(A, b)$  tal que  $A^-x = b^-, \forall x \in P$ , é a *matriz de igualdades* de  $P$  para a representação  $(A, b)$ . Caso essa submatriz não exista, consideramos que  $(A^-, b^-)$  é a matriz nula. Um poliedro limitado é chamado *politopo*. Ao longo deste texto, consideramos somente poliedros (e politopos) racionais, ou seja, que tem uma representação dada por uma matriz de valores racionais.

**Lema 2.5.** *Seja  $P = \{x \in \mathbb{R}^n | Ax \leq b\} \neq \emptyset$  um poliedro. Então  $\dim(P) = n - \text{posto}(A^-)$ .*

**Corolário 2.6.** *Seja  $P \subseteq \mathbb{R}^n$  um poliedro e  $(\lambda, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}$  tal que  $P \subseteq \{x \in \mathbb{R}^n : \lambda^T x = \lambda_0\}$ . Então,  $P$  tem dimensão plena se, e somente se,  $(\lambda, \lambda_0) = (\mathbf{0}, \mathbf{0})$ .*

Uma desigualdade  $\lambda^T x \leq \lambda_0$  [ou simplesmente  $(\lambda, \lambda_0)$ ] é chamada *desigualdade válida* para  $P$  se ela é satisfeita por todos os vetores em  $P$ , i.e.  $\lambda^T x \leq \lambda_0 \forall x \in P$ . Uma desigualdade válida  $(\lambda, \lambda_0)$  é dita uma *igualdade de  $P$*  se  $\lambda^T x = \lambda_0 \forall x \in P$ . Se  $(\lambda, \lambda_0)$  é uma desigualdade válida para  $P$  e  $F = \{x \in P | \lambda^T x = \lambda_0\}$  então  $F$  é chamado de *face* de  $P$ . Uma face  $F$  de  $P$  é chamada de *faceta* de  $P$ , se  $\dim(F) = \dim(P) - 1$ . Neste caso, dizemos que a desigualdade  $\lambda^T x \leq \lambda_0$  é *indutora de faceta*. Pode-se demonstrar que todo poliedro pode ser descrito por suas igualdades e suas desigualdades indutoras de facetas apenas.

**Teorema 2.7.** *Sejam  $P$  um poliedro, com  $\dim(P) = n - k$ , e  $F_i \subset P$  suas facetas,  $i = 1, 2, \dots, t$ . Então,*

$$P = \{x \in \mathbb{R}^n : (a^i)^T x = b_i, i = 1, \dots, k \text{ e } (a^i)^T x \leq b_i, i = k + 1, \dots, k + t\},$$

onde:

1.  $((a^i)^T, b_i), i = 1, \dots, k$ , são as linhas de uma matriz de posto  $k$ , chamada matriz de igualdades de  $P$ ;
2. Para  $i = k + 1, \dots, k + t$ ,  $((a^i)^T, b_i)$  induz a faceta  $F_i$ .

Em particular, se  $P$  tem dimensão plena, então

$$P = \{x \in \mathbb{R}^n : (a^i)^T x \leq b_i, i = 1, \dots, t\},$$

onde  $((a^i)^T, b_i)$  induz a faceta  $F_i$ , para  $i = 1, 2, \dots, t$ .

As seguintes propriedades são úteis para caracterizar facetas.

**Teorema 2.8.** *Seja  $P$  um poliedro. Sejam  $(A^{\bar{}}, b^{\bar{}})$  a matriz de igualdades de  $P$  em qualquer representação de  $P$  e  $F = \{x \in P : \pi^T x = \pi_0\}$  uma face de  $P$ , com  $\emptyset \neq F \neq P$ . São equivalentes as assertivas:*

- (i)  $F$  é faceta.
- (ii)  $F \subseteq \{x \in P : \lambda^T x = \lambda_0\} \Leftrightarrow (\lambda^T, \lambda_0) = \alpha(\pi^T, \pi_0) + u^T(A^{\bar{}}, b^{\bar{}})$  para  $\alpha \in \mathbb{R}$ ,  $u \in \mathbb{R}^{m^{\bar{}}}$ , onde  $m^{\bar{}}$  é o número de linhas de  $A^{\bar{}}$ .

Em particular, se  $P$  tem dimensão plena,  $F$  ser faceta equivale a

$$F \subseteq \{x \in P : \lambda^T x = \lambda_0\} \Leftrightarrow (\lambda^T, \lambda_0) = \alpha(\pi^T, \pi_0) \text{ para } \alpha \in \mathbb{R}.$$

## 2.3 Programação Inteira

Um problema de programação inteira é da forma

$$(PPI) \quad \min c^T x \text{ sujeito a } x \in X,$$

onde  $c \in \mathbb{Q}^n$ ,  $X = \{x \in P \cap \mathbb{Z}^n : x_j \in \mathbb{Z}, j \in J \subseteq \{1, 2, \dots, n\}\}$  e  $P \subseteq \mathbb{R}_+^n$  é um poliedro racional. A função linear  $f(x) = c^T x$  é chamada *função objetivo*, e o conjunto  $X$  é o conjunto viável. As desigualdades que definem o poliedro  $P$  são chamadas *restrições*, enquanto  $x_j \in \mathbb{Z}$ ,  $j \in J$ , são as *restrições de integralidade*. Cada ponto  $x \in X$  é uma *solução viável*, e uma *solução ótima*  $x^*$  é uma solução viável que satisfaz  $c^T x^* \leq c^T x$ ,  $\forall x \in X$ .

Associado a (PPI), temos o seguinte problema de programação linear

$$(PPL) \quad \min c^T x \text{ sujeito a } x \in P,$$

chamado relaxação linear, que fornece um limite inferior para o problema inteiro. Já o problema

$$(PPC) \quad \min c^T x \text{ sujeito a } x \in \text{conv}(X),$$

é equivalente a (PPI). Na verdade, pode-se mostrar que este último também é um problema de programação linear, pois:

**Teorema 2.9.**  *$\text{conv}(X)$  é um poliedro.*

Como problemas de programação linear podem ser resolvidos eficientemente, parece razoável abordar (PPI) via (PPC). A questão, porém, é caracterizar o poliedro  $\text{conv}(X)$ , que pode ser definido por um número exponencial de desigualdades. Pelos resultados da subseção anterior, bastaria considerarmos as desigualdades que definem facetas de  $\text{conv}(X)$ .

Mesmo que as conseguíssemos caracterizar, ainda restaria a questão de como usá-las no modelo, se são potencialmente em número exponencial. Uma estratégia é usar o método

de planos de corte, que funciona assim. Primeiro, resolva o problema com um número pequeno de restrições (uma alternativa é começar com as restrições da relaxação linear ou um subconjunto delas) e obtenha um ponto ótimo  $x$ . Em seguida, resolva o *problema de separação*, que consiste em verificar se  $x$  satisfaz todas as desigualdades válidas que não estão presentes no modelo, ou determinar uma delas que esteja violada. No primeiro caso, o processo pára; no segundo, inclui-se a restrição violada e repete-se o processo. O problema de separação costuma ser um outro problema de otimização, que pode ser fácil para algumas classes de desigualdades. Idealmente, gostaríamos de saber separar eficientemente desigualdades indutoras de facetas.

Na verdade, o algoritmo de planos de corte descrito acima funciona não apenas para resolver o (PPI) completamente, como para encontrar limites inferiores para o valor da solução, o que é obtido ao final de cada iteração do processo.

## 3 Formação de Equipes de Trabalho

O Problema de Formação de Equipes Sociotécnicas, objeto principal deste trabalho, é a composição de dois outros problemas básicos que tratamos neste capítulo: O Problema de Formação de Equipes Técnicas (*FET*) e O Problema de Formação de Equipes Sociais (*FES*). Na Seção 3.1, são feitas algumas considerações sobre o problema de formação de equipes utilizando dados técnicos dos envolvidos, apresentada uma definição formal para este problema e realizado um estudo de sua complexidade. Na Seção 3.2, são apresentadas metodologias para estudo matemático de redes sociais de organizações, assim como considerações sobre o relacionamento deste estudo com o Problema de Formação de Equipes Sociais. São apresentados também uma definição formal para o problema e um estudo de sua complexidade.

### 3.1 O problema de formação de equipes técnicas

O sistema técnico de equipes de trabalho, de acordo com (PASMORE et al., 1982), são ferramentas, artefatos, métodos, procedimentos e conhecimentos utilizados para atingir as metas propostas por uma organização. A maneira como o sistema técnico de uma organização é projetado irá impor restrições ao ambiente profissional, pois é de extrema importância que os indivíduos participantes de uma organização possuam as habilidades profissionais requeridas para realização de determinadas tarefas.

O Problema de Formação de Equipes Técnicas pode ser visto como sendo o problema de encontrar um conjunto de equipes em uma organização, cada uma delas possuindo uma demanda específica de indivíduos com determinadas habilidades profissionais. Formalmente, temos:

**Problema 3.1.** *O Problema de Formação de Equipes Técnicas (FET)*

**Entrada:** Uma tupla  $(U, S, T, s, t)$ , sendo  $U$  o conjunto de indivíduos participantes de uma organização;  $S$  o conjunto de habilidades requeridas pela organização;  $T$  o conjunto de equipes que devem ser formadas;  $s$  uma função  $s : U \rightarrow 2^S$  que retorna o conjunto de habilidades em  $S$  que um dado indivíduo  $u \in U$  possui; e  $t$  uma função  $t : T \times S \rightarrow \mathbb{N}$  que retorna a demanda de indivíduos para cada tipo de habilidade em cada equipe, isto é, em uma equipe  $j \in T$  são necessários  $t(j, s)$  indivíduos com habilidade  $s \in S$ ;

**Questão:** Determinar um conjunto de  $|T|$  equipes, onde:

1. cada indivíduo  $u \in U$  é alocado a no máximo uma equipe  $j \in T$ , desempenhando exatamente uma habilidade  $s \in s(u)$  na equipe onde está alocado; e
2. em cada equipe  $j \in T$  existe pelo menos  $t(j, s)$  indivíduos desempenhando a habilidade  $s$ , para todo  $s \in S$ .

Ou a verificação de que tais equipes não podem ser formadas.

**Obsevação 3.1.** A saída do problema *FET* pode ser descrita por uma família  $\{U_{js}\}_{j \in T, s \in S}$  de subconjuntos disjuntos de  $U$ , onde  $U_{js} \subseteq \{u \in U : s \in s(u)\}$  são os indivíduos desempenhando a habilidade  $s \in S$  na equipe  $j \in T$  e, portanto,  $|U_{js}| \geq t(j, s)$ .

Por simplicidade e sem prejuízo no entendimento da notação, vamos usar também a letra  $s$  para definir uma função  $s : T \rightarrow 2^S$  que informa as habilidades  $s \in S$  demandadas em uma equipe  $j \in T$ , ou seja,  $s(j) = \{s \in S : t(j, s) > 0\}$ . Denotamos por  $s(u, j) = s(u) \cap s(j)$  o conjunto de habilidades de um indivíduo  $u \in U$  necessárias em uma equipe  $j \in T$ .

Mostramos que o Problema de Formação de Equipes Técnicas pode ser reduzido a um Problema de Fluxo Máximo (*PMF*). Neste último problema, dada uma tupla  $F = (G, c, o, d)$ , onde

- $(G, c)$  é uma rede, definida por um grafo orientado  $G = (V, E)$  e uma função de capacidade  $c : E(G) \rightarrow \mathbb{R}^+$ , e
- $o$  e  $d$  são os nós de origem e de destino, respectivamente,

queremos encontrar um fluxo máximo em  $G$ , entre  $o$  e  $d$ , respeitando a capacidade dos arcos, ou seja, queremos determinar uma função  $f : E(G) \rightarrow \mathbb{R}^+$  satisfazendo:

1.  $f(uv) \leq c(uv), \forall uv \in E(G)$  (restrição de capacidade)
2.  $\sum_{uv \in \delta^-(v)} f(uv) = \sum_{uv \in \delta^+(v)} f(vu), \forall v \in V \setminus \{o, d\}$  (conservação de fluxo)

tal que o valor de  $f$ , dado por  $|f| = \sum_{ov \in E(G)} f(ov)$ , seja máximo.

Note que, pelas restrições de conservação de fluxo, o fluxo que sai do nó origem  $o$  é repassado por alguns nós intermediários até chegar ao nó destino  $d$ . Sendo assim, o valor do fluxo também é dado por  $|f| = \sum_{ud \in E(G)} f(ud)$ .

Em outros termos, o *PMF* consiste em encontrar um fluxo  $f^*$  de valor máximo na rede que pode escoar entre dois vértices (origem e destino), respeitando a conservação de fluxo (o fluxo que entra deve ser igual ao que sai, exceto na origem e no destino) e as restrições de capacidade de cada arco. Muitos problemas podem ser vistos como um Problema de Fluxo Máximo ou então como uma sequência de Problemas de Fluxo Máximo (GOLDBERG; TARJAN, 1990).

A redução de *FET* em *PMF* é apresentada a seguir. À medida que descrevemos a redução, utilizamos a seguinte instância como ilustração:

- $T = \{1, 2\}$  e  $S = \{1, 2\}$ ;
- $U = \{v_1, v_2, v_3, v_4\}$ , sendo:  $s(v_1) = \{1, 2\}$ ,  $s(v_2) = \{1\}$ ,  $s(v_3) = \{2\}$ ,  $s(v_4) = \{2\}$ ;
- $t(1, 1) = 1$ ,  $t(1, 2) = 1$ ,  $t(2, 1) = 0$  e  $t(2, 2) = 2$ .

A Figura 1 apresenta a rede  $F(I_{FET})$  gerada a partir da instância  $I_{FET}$  acima. Uma solução é descrita pelos arcos mais escuros.

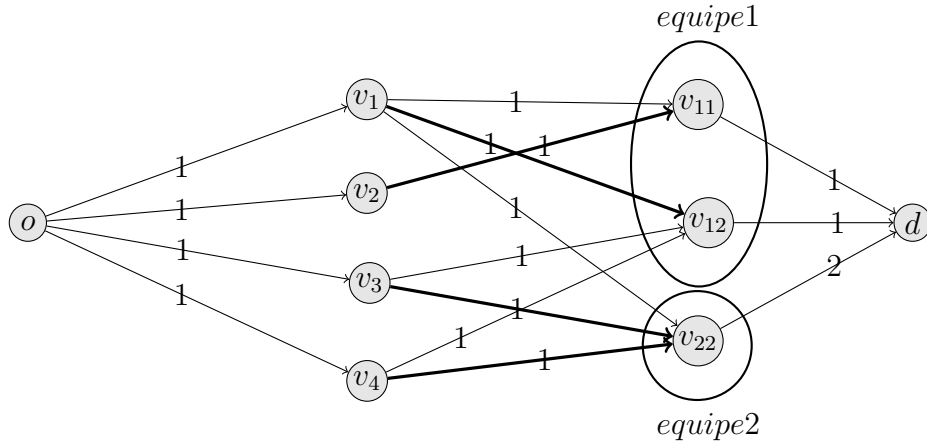


Figura 1 – Rede  $F(I)$  gerada a partir de uma instância  $I$  do  $FET$ . Solução para o  $PMF$  na rede  $F(I)$  destacada pelas arestas mais escuras.

Em geral, dada uma instância  $I_{FET}$ , construímos a rede  $F(I_{FET})$  como segue. Criamos um vértice origem, denotado por  $o$ . Para cada indivíduo  $u \in U$  criamos um vértice  $v_u$  e incluímos um arco  $ov_u$  com  $c(ov_u) = 1$ ; a instância utilizada no exemplo possui 4 indivíduos e, portanto, criamos os vértices  $v_1, v_2, v_3$  e  $v_4$ . Para cada equipe  $j \in T$  e cada habilidade  $s \in S$  criamos um vértice denotado por  $v_{js}$ , se e somente se  $t(j, s) \neq 0$ ; na instância utilizada como exemplo são necessárias duas equipes, sendo que a equipe 1 necessita de 1 indivíduo com habilidade 1 e 1 indivíduo com habilidade 2 ( $t(1, 1) = 1$  e  $t(1, 2) = 1$ ), e a equipe 2 necessita de 2 indivíduos com habilidade 2 ( $t(2, 1) = 0$  e  $t(2, 2) = 2$ ). Por isso criamos os vértices  $v_{11}, v_{12}$  e  $v_{21}$ . Incluímos arcos  $v_u v_{js}$  com  $c(v_u v_{js}) = 1$  se, e somente se,  $s \in s(u)$ ; na instância utilizada como exemplo o indivíduo  $v_1$  possui habilidade 1 e 2, o indivíduo  $v_2$  possui apenas a habilidade 1, e os indivíduos  $v_3$  e  $v_4$  apenas a habilidade 2. Por fim, criamos um vértice de destino  $d$  e, entre cada vértice  $v_{js}$  e o vértice destino  $d$ , incluímos um arco  $v_{js}d$  com  $c(v_{js}d) = t(j, s)$ .

Formalmente, para  $I_{FET} = (U, S, T, s, t)$ , temos que  $F(I_{FET}) = (G, c, o, d)$  onde

$$\begin{aligned} V(G) &= \{v_u : u \in U\} \cup \{v_{js} : j \in T, s \in s(j)\} \cup \{o, d\}, \\ E(G) &= \{(o, v_u) : u \in U\} \cup \{v_u v_{js} : u \in U, j \in T, s \in S, s \in s(u, j)\} \\ &\quad \cup \{(v_{js}, d) : j \in T, s \in s(j)\}, \end{aligned}$$



$$\begin{aligned} c(ov_u) &= c(v_u v_{v_{j_s}}) = 1, \quad \forall u \in U, j \in T, s \in s(u, j), \\ c(v_{j_s} d) &= t(j, s), \quad \forall j \in T, s \in s(j). \end{aligned}$$

**Teorema 3.2.** *Sejam  $I_{FET}$  uma instância de FET e  $f^*$  uma solução ótima inteira de  $F(I_{FET})$ . Então  $I_{FET}$  possui solução se, e somente se,  $|f^*| = \sum_{j \in T} \sum_{s \in S} t(j, s)$ . Mais ainda, se  $f_{v_u v_{j_s}}^* = 1$  então  $u$  deve ser alocado na equipe  $j$  com habilidade  $s$ .*

*Demonstração.* Seja  $F(I_{FET}) = (G, c, o, d)$  com  $G = (V, E)$ .

$\Leftarrow$  Seja  $f^*$  inteiro com  $|f^*| = \sum_{j \in T} \sum_{s \in s(j)} t(j, s)$ . Como  $f$  é inteiro, então  $f_{v_u v_{j_s}}^* \in \{0, 1\}$ ,  $\forall (v_u, v_{j_s}) \in E$ . Aloque na equipe  $j \in T$  com habilidade  $s \in s(j)$  os indivíduos do conjunto  $U_{j_s} = \{u \in U : f_{v_u v_{j_s}}^* = 1\}$ . Pela Observação 3.1, devemos mostrar que  $U_{j_s} \subseteq \{u : s \in s(u)\}$  e  $|U_{j_s}| \geq t(j, s)$ ,  $\forall j \in T, \forall s \in s(j)$ , e que esses conjuntos são disjuntos. Claramente, se  $u \in U_{j_s}$  então  $c(v_u v_{j_s}) \geq 1$ , implicando que  $s \in s(u)$ . Como  $\sum_{j \in T, s \in s(u, j)} f_{v_u v_{j_s}}^* = f_{ov_u}^* \leq 1 \quad \forall u \in U$ , podemos deduzir que os conjuntos são disjuntos. Resta, pois, mostrar que  $|U_{j_s}| = t(j, s)$ ,  $\forall j \in T, \forall s \in s(j)$ .

Pela hipótese e pela conservação de fluxo temos que  $\sum_{j \in T} \sum_{s \in s(j)} f_{v_{j_s} d}^* = |f^*| = \sum_{j \in T} \sum_{s \in s(j)} t(j, s)$ . E como  $f_{v_{j_s} d}^* \leq t(j, s)$ , concluímos que  $f_{v_{j_s} d}^* = t(j, s)$ ,  $\forall j \in T$  e  $\forall s \in s(j)$ . Além disso, pela conservação de fluxo,  $\sum_{u \in U: v_u v_{j_s} \in E} f_{v_u v_{j_s}}^* = f_{v_{j_s} d}^* = t(j, s)$ . Como  $f_{v_u v_{j_s}}^* \in \{0, 1\}$ , o termo mais à esquerda na expressão acima é exatamente  $|U_{j_s}|$ , mostrando o resultado desejado.  $\Rightarrow$  Considere uma solução de  $I_{FET}$  representada pelos conjuntos  $U_{j_s}$ , para todo  $j \in T$  e  $s \in s(j)$ , formados pelos indivíduos alocados na equipe  $j$  com habilidade  $s$ . Sem perda de generalidade, podemos considerar que  $|U_{j_s}| = t(j, s)$ . Defina,  $f$  tal que:

$$f_{v_u v_{j_s}} = \begin{cases} 1, & \text{se } u \in U_{j_s} \\ 0, & \text{c.c.} \end{cases} \quad (3.1)$$

$$f_{ov_u} = \sum_{(j, s) \in T \times S: (v_u, v_{j_s}) \in E} f_{v_u v_{j_s}} \quad (3.2)$$

$$f_{v_{j_s} d} = \sum_{u \in U: (v_u, v_{j_s}) \in E} f_{v_u v_{j_s}} \quad (3.3)$$

Queremos mostrar que  $f$  é fluxo máximo. Por (3.2), (3.3) temos a conservação de fluxo. Mais ainda, como cada indivíduo é alocado a no máximo uma equipe com no máximo uma habilidade, segue-se de (3.1) e (3.2) que  $f_{ov_u} \leq 1$ . E por (3.1) e (3.3) temos que  $f_{v_{j_s} d} = |U_{j_s}| = t(j, s)$ . Assim  $f$  respeita as restrições de capacidade. Além disso, estas últimas igualdades mostram que  $f$  satura todos os arcos chegando em  $d$ , sendo pois máximo.  $\square$

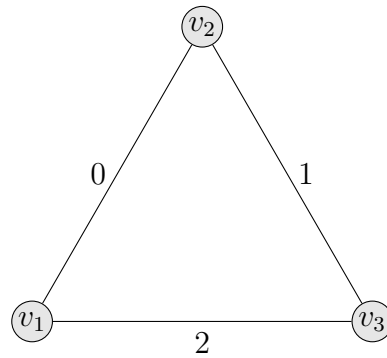


Figura 2 – Exemplo de um sociograma (grafo de afinidades)

Sabemos que o problema de fluxo máximo pode ser resolvido em tempo polinomial, por exemplo através do algoritmo proposto por (EDMONDS; KARP, 1972). Daí derivamos a complexidade de FET.

**Corolário 3.3.** *O problema FET é polinomial.*

## 3.2 O problema de formação de equipes sociais

O sistema social de uma organização são todos os seres humanos que trabalham nela, com suas inter-relações sociais. Procurando estudar e compreender o sistema social, (MORENO, 1975) criou a sociometria, técnica que permite mensurar as relações interpessoais em um grupo para um determinado contexto. Como tal, ela se ocupa do estudo matemático das características psicossociais de indivíduos pertencentes a um grupo, dos métodos experimentais e dos resultados da aplicação de princípios quantitativos, direcionando a sua investigação para a pesquisa do desenvolvimento e organização dos grupos e da situação dos indivíduos que os compõem. Para atingir seus objetivos, a sociometria utiliza instrumentos sociométricos, fazendo uso principalmente do teste sociométrico, desenvolvido por (MORENO, 1975) em suas pesquisas.

A partir da aplicação desse teste, pode-se estudar, medir e compreender uma rede de relacionamentos (rede social). O teste sociométrico pode ser aplicado de diversas formas. A mais comumente utilizada consiste na aplicação de um questionário para cada elemento de uma rede social, onde se procura encontrar vínculos entre os indivíduos que compõem esta rede.

Com o resultado do teste é possível construir um sociograma da rede social em estudo. O sociograma criado pode ser visto como um grafo completo  $G = (V, E)$ , no qual o conjunto de vértices  $V$  representa os indivíduos da rede social, e as relações interpessoais são representadas pelas arestas do conjunto  $E$ , que são ponderadas para descrever o nível de afinidade entre cada par de indivíduos.

Neste trabalho representaremos os relacionamentos através de uma função  $d : E(G) \rightarrow \mathbb{N}$ , onde quanto menor o valor de  $d(uv)$ , maior é a *afinidade* entre dois indivíduos  $u$  e  $v$ . A Figura 2 apresenta um exemplo de sociograma. Vemos que os indivíduos 1 e 2 possuem o melhor relacionamento, enquanto 1 e 3 apresentam a menor afinidade.

Dadas as descrições acima, o Problema de Formação de Equipes Sociais pode ser visto como sendo o problema de encontrar um conjunto de equipes em uma organização, onde cada equipe possua uma quantidade desejada de indivíduos com máxima afinidade possível. Formalmente, temos:

**Problema 3.2.** *O Problema de Formação de Equipes Sociais (FES)*

**Entrada:** Uma tupla  $(G, d, w, q)$ , sendo  $w$  um número natural e  $q : \{1, \dots, w\} \rightarrow \mathbb{N}$  uma função, representando respectivamente o número de equipes a serem criadas e o tamanho mínimo destas equipes;  $G = (V, E)$  um grafo completo; e  $d : E \rightarrow \mathbb{N}$  uma função, onde  $d(uv)$  representa o grau de afinidade entre  $u$  e  $v$ .

**Questão:** Determinar  $w$  subgrafos induzidos disjuntos, onde o subgrafo  $i$  tem tamanho no mínimo  $q_i$ ,  $\forall i = \{1, \dots, w\}$ , que minimizem o somatório dos pesos  $d(uv)$  das arestas pertencentes aos subgrafos escolhidos.

Para estudarmos a complexidade computacional de *FES*, vamos apresentar uma redução a partir do problema de conjunto independente máximo em um grafo (*CIM*). Começamos apresentando versões de decisão dos dois problemas:

- Problema de Formação de Equipes Sociais - versão de decisão (*FES<sub>D</sub>*): Dados números naturais  $w$ ,  $q_i \forall i = \{1, \dots, w\}$  e  $k$ , um grafo completo  $G = (V, E)$  com uma função de pesos atribuídos às arestas  $d : E \rightarrow \mathbb{N}$ , existem  $w$  subgrafos induzidos disjuntos cujo somatório dos pesos de suas arestas totalize no máximo  $k$ ?
- Problema Conjunto Independente - versão de decisão (*CIM<sub>D</sub>*): Dado um grafo  $G = (V, E)$  e um inteiro  $t$ , existe um conjunto independente de tamanho  $t$ ?

Note que uma entrada para *CIM<sub>D</sub>* é uma dupla  $(G, t)$  e uma entrada para *FES<sub>D</sub>* é uma tupla  $(G, d, w, q, k)$ .

REDUÇÃO *CIM<sub>D</sub>*  $\rightarrow$  *FES<sub>D</sub>*: Dada uma instância  $I_{CIM_D} = (G, t)$ , definimos a instância  $f(I_{CIM_D})$  de *FES<sub>D</sub>* tal que  $f(I_{CIM_D}) = (G', d, w = 1, q_1 = t, k = 0)$ , onde  $G'$  é o grafo completo com  $|V(G)|$  vértices e  $d : E(G') \rightarrow \mathbb{N}$  é dada por:

$$d(uv) = \begin{cases} 1, & \text{se } uv \in E(G) \\ 0, & \text{c.c.} \end{cases} \quad (3.4)$$

**Proposição 3.4.** *FES<sub>D</sub> é NP-Completo.*

*Demonstração.* Claramente,  $FES_D$  está em NP. Como  $CIM_D$  é NP-Completo (ROBSON, 1986), resta mostrar que a resposta (negativa ou positiva) para uma instância  $I_{CIM_D}$  coincide com a resposta para a instância  $f(I_{CIM_D})$ . Primeiramente, suponha que  $G$  possui um conjunto independente de tamanho  $t$ . Tal conjunto independente define  $w = 1$  equipe com  $q_i = t$  indivíduos de valor  $k = 0$ , conforme (3.4). Por outro lado, se existe uma equipe com pelo menos  $t$  indivíduos cujo grau total de relacionamento é (no máximo) zero, por (3.4) qualquer subconjunto de  $t$  elementos corresponde a um  $CIM$  no grafo  $G$ .  $\square$

**Corolário 3.5.**  *$FES$  é NP-Difícil.*

Consideramos agora uma versão restrita de  $FES$ , onde o número de equipes é fixo, ou seja,  $w$  é constante e não faz parte da entrada do problema. Vamos denotar essa versão por  $w-FES$  e sua versão de decisão correspondente por  $w-FES_D$ .

REDUÇÃO  $CIM_D \rightarrow w-FES_D$ : Dada uma instância  $I_{CIM_D} = (G, t)$ , definimos a instância  $f_w(I_{CIM_D})$  tal que  $f_w(I_{CIM_D}) = (G'_w, d, q_1 = q_2 = \dots = q_w = t, k = 0)$ , onde  $G'_w$  e  $d$  são obtidos com base em um grafo auxiliar  $G_w$ , que é a junção de  $w$  cópias de  $G$ , da seguinte forma:

1.  $G'_w$  é o grafo completo com  $|V(G_w)|$  vértices;
2.  $d$  marca as arestas de  $G'_w$  existentes em  $G_w$  (originárias de  $G$  ou introduzidas pela junção), ou seja,

$$d(uv) = \begin{cases} 1, & \text{se } uv \in E(G_w) \\ 0, & E(G'_w) \setminus E(G_w). \end{cases}$$

**Proposição 3.6.** *Para todo  $w \geq 1$ ,  $w-FES_D$  é NP-Completo e  $w-FES$  é NP-Difícil.*

*Demonstração.* Seja  $w \geq 1$ . Claramente  $w-FES_D$  está em NP. Primeiramente, suponha que existe um conjunto independente de tamanho  $t$  em  $G$ . Como  $G_w$  é a junção de  $w$  cópias de  $G$ , existem  $w$  cópias desse conjunto independente em  $G_w$ . Logo, podemos formar  $w$  equipes de tamanho  $t$  cada e valor zero no total em  $G'_w$ . Para a inversa, observe que, sendo o valor da solução zero, cada equipe (subgrafo) de tamanho pelo menos  $t$  deve estar contida em uma mesma cópia de  $G$  em  $G_w$ . Logo, temos pelo menos um conjunto independente em  $G$  de tamanho  $t$ . O resultado segue do fato que  $CIM_D$  é NP-Completo.  $\square$

Vale mencionar que um algoritmo heurístico para resolução do problema  $FES$  foi proposto por (OLIVEIRA et al., 2010). Como dissemos, porém, o foco do nosso trabalho é no problema que combina  $FES$  e  $FET$ , como definimos no próximo capítulo.

## 4 O Problema de Formação de Equipes Sociotécnicas

Neste capítulo, apresentamos o Problema de Formação de Equipes Sociotécnicas (*FEST*), foco de estudo deste trabalho. A definição formal do problema aparece na Seção 4.1. Na Seção 4.2, analisamos a complexidade de encontrar uma solução viável e uma solução ótima para este problema e, na Seção 4.3, apresentamos um algoritmo para obtenção de bons limites superiores.

### 4.1 Definição do problema

O Problema de Formação de Equipes Sociotécnicas (*FEST*) é uma generalização dos problemas *FET* e *FES*, apresentados nas Seções 3.1 e 3.2. Para descrevê-lo, acrescentamos ao sociograma da organização informações sobre as habilidades profissionais que cada indivíduo integrante da organização estudada possui, ou seja, dado um grafo completo  $G = (V, E)$  ponderado nas arestas, que descreve o sociograma, associamos a cada vértice  $v \in V(G)$  um conjunto de habilidades  $s(v) \subseteq S$ . Desejamos então formar um conjunto de equipes as mais harmoniosas possíveis, considerando os graus de relacionamentos, e que respeitem requisitos técnicos, definidos em função do número de indivíduos alocados em cada equipe e exercendo cada habilidade. Abusando da representação do problema através do grafo, ao longo do texto, vamos usar indistintamente os termos indivíduo e vértice ou subgrafo e equipe.

O problema *FEST* é formalmente definido a seguir.

**Problema 4.1.** *O Problema de Formação de Equipes Sociotécnicas (FEST)*

**Entrada:** Uma tupla  $(G, d, S, T, s, t)$ , dada por:

- Grafo completo  $G = (V, E)$ , onde  $V(G)$  representa os indivíduos de uma organização e  $E(G)$  representa os relacionamentos entre pares de indivíduos;
- Função peso  $d : E \rightarrow \mathbb{N}$  associada às arestas em  $E(G)$ , onde quanto menor o valor de  $d(uv)$  maior é o grau de afinidade entre um par de indivíduos  $u$  e  $v$ ;
- Conjunto  $S$  de habilidades que podem ser associadas a cada vértice  $u \in V(G)$ ;
- Conjunto  $T$  de tipos de subgrafos (equipes) que deverão ser criados;
- Função  $s : V \rightarrow 2^S$  que informa o conjunto das habilidades  $s \in S$  que vértice  $u \in V$  possui;

- Função de demanda  $t : T \times S \rightarrow \mathbb{N}$  que informa quantos vértices com habilidade  $s$  são necessários em cada equipe  $j$ ,  $\forall j \in T$  e  $\forall s \in S$ ;

**Questão:** Determinar  $|T|$  subgrafos induzidos disjuntos, onde:

1. cada vértice  $u \in V(G)$  pertence a no máximo um subgrafo  $j$ , assumindo exatamente uma habilidade  $s \in s(u)$  no subgrafo onde está alocado;
2. em cada subgrafo  $j$  existe pelo menos  $t(j, s)$  vértices assumindo a habilidade  $s$ , para todo  $j \in T$  e para todo  $s \in S$ ; e
3. o somatório dos pesos  $d(uv)$  das arestas pertencentes aos subgrafos seja minimizado;

Ou a verificação de que tais equipes não podem ser formadas.

Assim como na Seção 3.2, adotamos a notação  $s(j) = \{s \in S : t(j, s) > 0\}$  e  $s(u, j) = s(u) \cap s(j)$ , para todo  $u \in V$  e  $j \in T$ , para expressar, respectivamente, o conjunto de habilidades requeridas em uma equipe  $j$  e, dentre essas, aquelas que podem ser desempenhadas pelo indivíduo  $u$ .

## 4.2 Complexidade do problema

Observe que o problema *FES*, definido na Seção 3.2, corresponde a um caso particular de *FEST* quando  $|S| = 1$ . De fato, se  $I_{FET} = (G, d, w, q)$  é uma instância de *FES*, então  $I_{FEST} = (G, d, S = \{1\}, T = \{1, 2, \dots, w\}, s(u) = 1 \forall u \in V, t(j, 1) = q_j \forall j \in T)$  é uma instância de *FEST*. Sendo assim, concluímos que

**Corolário 4.1.** *O problema FEST é NP-Difícil.*

Por outro lado, *FEST* é uma versão de otimização do problema *FET*. Note que, se  $I_{FEST} = (G, d, S, T, s, t)$  é uma instância de *FEST*, então  $I_{FET} = (V(G), S, T, s, t)$  é uma instância de *FET*. Isto nos leva ao seguinte resultado.

**Proposição 4.2.** *Uma solução é viável para uma instância  $I_{FEST} = (G, d, S, T, s, t)$  de FEST se, e somente, se ela for solução da instância  $I_{FET} = (V(G), S, T, s, t)$  de FET.*

*Demonstração.* É suficiente observar que as condições 1 e 2 que definem *FES* são as mesmas condições 1 e 2 que definem *FEST*.  $\square$

**Corolário 4.3.** *Encontrar uma solução viável para FEST e, portanto, um limite superior, é um problema polinomial.*

### 4.3 Limite superior

O desenvolvimento de um bom limite superior pode ajudar na eficiência e qualidade de algoritmos exatos. Descreveremos nesta seção um algoritmo meta-heurístico que fornece um bom limite superior para o *FEST*. O algoritmo aqui implementado é uma meta-heurística iterativa de base evolucionária, onde partimos de uma solução inicial viável e a cada iteração podemos melhorar ou não a solução inicial, respeitando sempre as características de viabilidade das soluções futuras.

#### 4.3.1 Heurística baseada em Simulated Annealing Não-Monotônico

O algoritmo aqui apresentado é baseado em uma variação da meta-heurística *Simulated Annealing (SA) Não-Monotônico*. O *SA* é uma meta-heurística probabilística, proposta no trabalho de (KIRKPATRICK, 1975), cuja característica principal é a possibilidade de aceitar movimentos que pioram a solução atual do sistema com o intuito de escapar de mínimos locais ao longo do processo de otimização. Movimentos que melhoram a solução corrente são sempre aceitos, enquanto aqueles que pioram são efetivamente realizados com probabilidade  $e^{-\Delta/temp}$ , onde  $\Delta > 0$  é o incremento no custo da solução corrente e  $temp$  é um parâmetro, chamado temperatura, que controla o taxa de aceitação de soluções piores. Note que maiores probabilidades para aceitar uma solução pior ocorre quando  $\Delta$  é menor e  $temp$  é maior. Por isso, a temperatura  $temp$  vai sendo reduzida ao longo do processo. No *SA* tradicional, essa redução é realizada progressivamente. Na variação não-monotônica, em certos momentos, pode haver também um acréscimo na temperatura.

Uma descrição geral do algoritmo que propomos está apresentada no pseudo-código mostrado no Algoritmo 1.

Detalhamos a seguir os passos do algoritmo.

##### 4.3.1.1 Controle da temperatura

A temperatura do  $SA_{NM}$  é inicializada com valor de 100 unidades. A cada iteração esta temperatura sofre um decaimento para 99,9% do seu valor atual:

$$temperatura_{(i+1)} = 0,99 * temperatura_{(i)} \quad (4.1)$$

Quando a temperatura assume um valor inferior 0,01, então o sistema é reaquecido para 100 unidades. Isto caracteriza o caráter Não-Monotônico do modelo e permite que o sistema efetue diversas buscas locais ao longo do processo de otimização do *Simulated Annealing*. Com o reaquecimento, o sistema volta a ter novas chances de pesquisar outras regiões do espaço de soluções ao longo do processo de otimização.

**Algoritmo 1:** Descrição geral do *Simulated Annealing* Não-Monotônico para *FEST*


---

**Entrada:**  $G = (V, E), d, S, T, s, t$   
**Dados:**  $Q, Q^*, temperatura$

```

1  $Q \leftarrow \text{criarSolucaoInicial}();$ 
2  $Q^* \leftarrow Q;$ 
3  $temperatura \leftarrow 100;$ 
4 repita
5    $temperatura \leftarrow temperatura * 0,99;$ 
6   se ( $temperatura < 0,01$ ) então
7      $temperatura \leftarrow 100;$ 
8   fim
9    $Q' \leftarrow \text{solucaoVizinha}(Q);$ 
10   $\Delta \leftarrow \text{funcaoObjetivo}(Q') - \text{funcaoObjetivo}(Q);$ 
11  se ( $\Delta < 0$  ou  $\text{numeroAleatorio}() < e^{-\Delta/temperatura}$ ) então
12     $Q \leftarrow Q';$ 
13    se ( $\Delta < 0$ ) então
14       $Q^* \leftarrow Q$ 
15    fim
16  fim
17 até  $\text{tempoDeExecucao} > 1000$  segundos;
18 retorna  $Q^*$ 

```

---

## 4.3.1.2 Construção da solução inicial

Como mostra a Proposição 4.2, dada uma instância  $I_{FEST} = (G, d, S, T, s, t)$ , podemos construir uma solução inicial viável para o *FEST*, resolvendo apenas o Problema de Formação de Equipes Técnicas (*FET*). Assim, uma solução inicial viável para o *FEST* pode ser obtida em tempo polinomial, através de um algoritmo de fluxo máximo. Os detalhes da construção da rede e da aplicação do algoritmo de fluxo máximo podem ser vistos na Seção 3.1. O algoritmo de fluxo máximo utilizado em nossa implementação foi desenvolvido por (GOLDBERG; TARJAN, 1988).

A solução  $Q$  para o *FEST*, obtida pela aplicação do algoritmo de fluxo máximo, é representada por conjuntos  $U(j)$ , para cada  $j \in T$ , que contêm os indivíduos da equipe  $j$ . Cada conjunto  $U(j)$  é codificado como um vetor  $|V(G)|$ -dimensional, onde a entrada correspondente a  $u \in V(G)$  é igual a  $s \in S$ , se  $u$  participa da equipe  $j$  desempenhando a habilidade  $s$ , ou recebe 0 para registrar que  $u$  não participa da equipe  $j$ .

## 4.3.1.3 Busca por solução vizinha

Dada uma solução viável  $Q$  para um problema particular  $P$ , podemos definir diversos algoritmos para encontrar soluções na vizinhança  $N(Q)$  de  $Q$ . Uma possibilidade simples é o operador *swap*, definido no trabalho de (EIBEN; RUTTKAY, 1996). Aplicado a um Algoritmo Genético, este operador escolhe dois genes aleatoriamente em um cromossomo



e permuta suas posições.

Baseados no operador *swap*, criamos dois operadores para busca de soluções vizinhas, com funcionalidades diferentes. O primeiro operador, denominado *swap1*, efetua trocas entre dois indivíduos alocados em equipes distintas, desde que ambos os indivíduos desempenhem a mesma habilidade em suas respectivas equipes. Primeiramente, são sorteadas duas equipes  $j_1$  e  $j_2$  e dois indivíduos  $u_1 \in U(j_1)$  e  $u_2 \in U(j_2)$ , onde ambos desempenhem a mesma habilidade em suas respectivas equipes. Após o *swap1*, o indivíduo  $u_2 \in U(j_1)$  e o indivíduo  $u_1 \in U(j_2)$ .

O segundo operador, denominado *swap2*, efetua trocas entre dois indivíduos de equipes distintas, onde as habilidades desempenhadas por cada indivíduo são diferentes. Neste caso, são sorteadas duas equipes  $j_1$  e  $j_2$  e dois indivíduos  $u_1 \in U(j_1)$ , desempenhando habilidade  $s_1$  e que possua a habilidade  $s_2$ , e  $u_2 \in U(j_2)$ , desempenhando habilidade  $s_2$  e possuindo a habilidade  $s_1$ , com  $s_1 \neq s_2$ . Após o *swap2*, o indivíduo  $u_2 \in U(j_1)$  desempenhando a habilidade  $s_1$  e o indivíduo  $u_1 \in U(j_2)$  desempenhando a habilidade  $s_2$ . Para que o operador *swap2* seja escolhido pelo algoritmo de busca de solução vizinha é necessário que sempre sejam escolhidos dois indivíduos que possuam as habilidades  $s_1$  e  $s_2$  em comum. Além disso, os indivíduos devem sempre desempenhar habilidades diferentes em suas respectivas equipes.

Na verdade, para que os elementos sorteados satisfaçam as condições especificadas acima, usamos a seguinte estratégia. Primeiro, sorteamos uma habilidade  $s_1$ , depois uma equipe  $j_1$  que demanda tal habilidade e, por último, um indivíduo  $u_1$  alocado em  $j_1$  e desempenhando habilidade  $s_1$ . Para o *swap1*, sorteamos uma outra equipe  $j_2$  que também demande  $s_1$  e um indivíduo  $u_2$  de  $j_2$  que desempenhe essa habilidade. Para o *swap2*, sorteamos uma outra habilidade  $s_2 \in s(u_1, j_1)$ , para logo depois escolher aleatoriamente uma equipe que demande por  $s_1$  e  $s_2$ . Sorteamos em seguida um indivíduo  $u_2$  da equipe  $j_2$  desempenhando  $s_2$  e que também possua  $s_1$ . Caso algum dos sorteios não seja possível por falta de opções, voltamos a realizar o sorteio imediatamente anterior e repetimos o processo.

Note que sempre deverão ser escolhidas duas equipes distintas para efetuar quaisquer dos operadores de vizinhança definidos, pois caso contrário não haveria impactos com relação à função objetivo da solução vizinha de  $Q$ . O algoritmo para busca de solução vizinha é representado pelo pseudo-código do Algoritmo 2.

O método *sorteioHabilidade* retorna qual a habilidade  $s_1 \in S$  que irá ser utilizada para definir as escolhas de indivíduos e equipes que participarão da operação de busca por vizinhança. Note que esse sorteio é realizado em função de um vetor de probabilidades denominado *prob1*, dada como entrada, onde as habilidades que possuem maior demanda tem uma probabilidade maior de serem sorteadas. Precisamente,

$$prob1(s) = \frac{\sum_{j \in T} t(j, s)}{\sum_{s \in S} \sum_{j \in T} t(j, s)}, \quad \forall s \in S.$$

---

**Algoritmo 2:** Descrição geral do algoritmo de busca de solução vizinha

---

**Entrada:**  $G(V, E), d, S, T, s, t, \text{prob1}, \text{prob2}$   
**Dados:**  $s_1, s_2, j_1, j_2, u_1, u_2$

```

1  $s_1 \leftarrow \text{sorteioHabilidade}(\text{prob1});$ 
2  $j_1 \leftarrow \text{sorteioEquipe}(s_1);$ 
3  $u_1 \leftarrow \text{sorteioPessoa}(j_1, s_1);$ 
4 se  $(|s(u_1)| = 1)$  ou  $(\text{numeroAleatorio}() \leq \text{prob2})$  então
5    $j_2 \leftarrow \text{sorteioEquipe}(s_1);$ 
6    $u_2 \leftarrow \text{sorteioPessoa}(j_2, s_1);$ 
7    $\text{swap1}(u_1, u_2);$ 
8 fim
9 senão
10   $s_2 \leftarrow \text{sorteioHabilidade}(u_1, j_1);$ 
11   $j_2 \leftarrow \text{sorteioEquipe}(s_1, s_2);$ 
12   $u_2 \leftarrow \text{sorteioPessoa}(j_2, s_1, s_2);$ 
13   $\text{swap2}(u_1, u_2);$ 
14 fim
```

---

Em seguida, o método *sorteioEquipe* indica a primeira equipe que participará da operação. Apenas equipes que possuam demandas de habilidade  $s_1$  poderão ser sorteadas. Posteriormente, o método *sorteioPessoa* indica um indivíduo  $u_1 \in U(j_1)$  que desempenhe a habilidade  $s_1$  para efetuar um *swap*.

Caso o indivíduo  $u_1$  sorteado possua apenas a habilidade que ele desempenha, não é possível executar o *swap2*. Caso contrário, consideramos as duas possibilidades de *swap*. A escolha do *swap* a ser realizada também é feita em função de uma probabilidade denominada *prob2*, dada como entrada, onde quanto maior for o  $\sum_{u \in V} |s(u)|$ , maior a probabilidade de escolha do *swap2*. Precisamente, a probabilidade de realizar o *swap2* é

$$\text{prob2} = \frac{\sum_{u \in V} |s(u)|}{|V| \times |S|}$$

### 4.3.2 Corretude do algoritmo

Note que, como a solução inicial dada ao Algoritmo 1 é sempre viável (Proposição 4.2) e as trocas realizadas pelos métodos *swap1* e *swap2* sempre retornam solução vizinhas viáveis, pois ambos os métodos conservam as equipes disjuntas e mantêm atendidas as demandas de cada habilidade  $s \in s(j)$  em cada equipe  $j \in T$ , temos:

**Proposição 4.4.** *Uma solução  $Q$  obtida pelo Algoritmo 1 é sempre viável para o FEST, fornecendo um limite superior para o problema.*

No Capítulo 6, apresentamos os resultados de nossos experimentos computacionais com esse algoritmo heurístico e os comparamos com os resultados fornecidos pelas formulações

de programação inteira que apresentamos no próximo capítulo. Também fornecemos a solução da heurística para o solver, de modo a agilizar o processo de poda.

## 5 Formulações Matemáticas

Neste capítulo apresentamos duas formulações matemáticas para o problema *FEST*. Na Seção 5.1, descrevemos uma formulação matemática, por nós proposta, que denominamos Formulação 1. Ela é definida usando  $\mathcal{O}(|U| \times |T| \times |S| + |U|^2)$  variáveis e  $\mathcal{O}(|U|^2 \times |T| + |T| \times |S|)$  restrições. Mostramos a corretude dessa formulação. Na Seção 5.2, realizamos um estudo poliédrico, identificando facetas, e propomos classes de desigualdades para fortalecer a formulação. Na Seção 5.3, é apresentada uma segunda formulação matemática, denominada Formulação 2, que foi sugerida por (MARENCO; DELLE, 2013). Essa segunda formulação possui  $\mathcal{O}(|U| \times |T| + |U|^2)$  variáveis, mas um número exponencial de restrições. Mostramos a corretude dessa formulação e que suas restrições podem ser separadas em tempo polinomial. Adaptamos para a Formulação 2 algumas das desigualdades propostas para Formulação 1. Fechamos o capítulo com uma breve comparação teórica entre as duas formulações, incluindo a relação entre suas relaxações lineares, motivando as experiências computacionais a serem apresentadas no capítulo seguinte.

### 5.1 Formulação 1

Dada uma instância  $I_{FEST} = (G, d, S, T, t, s)$  para o problema *FEST*, como definida na Seção 4.1, considere as seguintes variáveis.

Para cada vértice  $u \in V(G)$ , para cada subgrafo  $j \in T$  e para cada habilidade  $s \in S$ , introduzimos uma variável binária  $X_{ujs}$ , relacionando os vértices a subgrafos e a habilidades, onde:

$$X_{ujs} = \begin{cases} 1, & \text{se } u \text{ está no subgrafo } j \text{ assumindo a habilidade } s, \\ 0, & \text{caso contrário.} \end{cases} \quad \begin{array}{l} \forall u \in V(G), \\ \forall j \in T, \\ \forall s \in s(u, j). \end{array} \quad (5.1)$$

Para cada aresta  $uv \in E(G)$ , definimos uma variável binária  $Z_{uv}$  para indicar as arestas que aparecem dentro de um mesmo subgrafo, onde:

$$Z_{uv} = \begin{cases} 1, & \text{se } u \text{ e } v \text{ pertencem ao mesmo subgrafo,} \\ 0, & \text{caso contrário.} \end{cases} \quad \forall uv \in E(G). \quad (5.2)$$

Uma formulação para *FEST* é então:

$$\min \quad d(X, Z) = \sum_{uv \in E(G)} d(u, v) Z_{uv} \quad (5.3)$$

$$\text{s.a:} \quad \sum_{s \in s(u, j)} X_{ujs} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} \leq 1, \quad \forall j \in T, \forall u, v \in V(G): s(u, j) \neq \emptyset, s(v, j) \neq \emptyset, \quad (5.4)$$

$$\sum_{j \in T} \sum_{s \in s(u, j)} X_{ujs} \leq 1, \quad \forall u \in V(G), \quad (5.5)$$

$$\sum_{u: s \in s(u)} X_{ujs} \geq t(j, s), \quad \forall j \in T, s \in s(j), \quad (5.6)$$

$$X_{ujs} \in \{0, 1\}, \quad \forall u \in V(G), \forall j \in T, \forall s \in s(u, j) \quad (5.7)$$

$$Z_{uv} \in \{0, 1\}, \quad \forall uv \in E(G). \quad (5.8)$$

**Obsevação 5.1.** Nas restrições (5.6), podemos usar desigualdades ou igualdades. Optamos pelas desigualdades por razões que ficarão claras no estudo poliédrico da formulação.

A validade dessa observação é decorrência da seguinte propriedade.

**Lema 5.2.** Seja  $(X, Z)$  uma solução viável para a Formulação 1. Para cada  $j \in T$  e  $s \in s(j)$ , escolha  $\Delta_{js} \subseteq \{u \in V : X_{ujs} = 1\}$  tal que  $|\Delta_{js}| = \sum_{u \in V: s \in s(u)} X_{ujs} - t(j, s)$ . Então,  $(X^0, Z)$  também é viável, satisfazendo as desigualdades (5.6) na igualdade e  $d(X^0, Z) = d(X, Z)$ , onde

$$X_{ujs}^0 = \begin{cases} 0, & u \in \Delta_{js}, \\ X_{ujs}, & \text{caso contrário,} \end{cases} \quad \forall u \in V, \forall j \in T, \forall s \in s(u, j).$$

*Demonstração.* Primeiro, perceba que os conjuntos  $\Delta_{js}$ , para todo  $j \in T$  e  $s \in s(j)$ , estão bem definidos porque  $(X, Z)$  verifica (5.6). E a definição de  $X^0$  retira apenas o excesso  $(\sum_{u \in V: s \in s(u)} X_{ujs} - t(j, s))$  de indivíduos que estão desempenhando a habilidade  $s$  na equipe  $j$ . Logo,  $X^0$  passa a satisfazer (5.6) na igualdade. Adicionalmente, como  $X^0 \leq X$  e  $(X, Z)$  satisfaz (5.4)–(5.5) e (5.7)–(5.8), essas mesmas restrições são verificadas por  $(X^0, Z)$ . Portanto,  $(X^0, Z)$  é uma solução viável. E mantém o valor de  $(X, Z)$ , pois as variáveis  $Z$  não foram modificadas.  $\square$

Outras formas de construir soluções viáveis de mesmo valor ou menor que aquele de uma solução viável dada são apresentadas a seguir.

**Lema 5.3.** Seja  $(X, Z)$  uma solução viável para a Formulação 1. Então  $(X, Z^0)$  também é viável, com  $d(X, Z^0) \leq d(X, Z)$ , onde

$$Z_{uv}^0 = \begin{cases} 1, & \sum_{s \in s(u, j)} X_{ujs} = \sum_{s \in s(v, j)} X_{vjs} = 1, \text{ para algum } j \in T \\ 0, & \text{caso contrário.} \end{cases}$$

Por outro lado,  $(X, Z^1)$  também é viável, com  $d(X, Z^1) = d(X, Z)$ , onde

$$Z_{uv}^1 = \begin{cases} 1, & Z_{uv} = 1 \text{ ou } d(uv) = 0 \\ 0, & \text{caso contrário.} \end{cases}$$

*Demonstração.* Para a viabilidade de  $(X, Z^0)$ , é suficiente mostrar que esta solução satisfaz (5.4), dada a viabilidade de  $(X, Z)$ . Note que a definição das componentes 1 em  $Z^0$  tem exatamente este efeito. Além disso, se  $Z_{uv} = 0$  então  $\sum_{s \in s(u,j)} X_{ujs} + \sum_{s \in s(u,j)} X_{vjs} \leq 1$  para todo  $j \in T$  e, por conseguinte,  $Z_{uv}^0 = 0$ . Ou seja, as componentes 1 em  $Z^0$  são um subconjunto daquelas em  $Z$ . Assim, como  $d(uv) \geq 0$ , deduzimos que  $d(X, Z^0) \leq d(X, Z)$ .

Para a viabilidade de  $(X, Z^1)$ , de novo é suficiente mostrar que esta solução satisfaz (5.4). Note que  $Z^1$  tem as mesmas componentes 1 que  $Z$  e, possivelmente, mais algumas. E converter uma componente de 0 para 1 em  $Z$  mantém a viabilidade da solução. Além disso, como  $Z_{uv} \neq Z_{uv}^1$  somente se  $d(uv) = 0$ , temos que  $d(X, Z) = d(X, Z^1)$ .  $\square$

Reservamos a notação  $(X^0, Z)$ ,  $(X, Z^0)$ ,  $(X, Z^1)$  para identificar soluções viáveis de (5.4)–(5.8) construídas, respectivamente, conforme os lemas 5.2 e 5.3, a partir de uma solução viável  $(X, Z)$ . Dizemos que  $(X^0, Z)$  é uma *solução reduzida* (uma *redução* de  $(X, Z)$ ), enquanto  $(X, Z^0)$  é uma *solução pura* (a *purificação* de  $(X, Z)$ ). A purificação de uma redução de  $(X, Z)$  será denotada por  $(X^0, Z^0)$ . Como consequência imediata dos lemas 5.2 e 5.3, temos:

**Corolário 5.4.** *Se  $(X, Z)$  é uma solução ótima para a Formulação 1, então  $(X^0, Z)$  e  $(X, Z^0)$ ,  $(X, Z^1)$  e  $(X^0, Z^0)$  também o são. Mais ainda, em  $(X^0, Z^0)$  temos  $\sum_{uv \in E} Z_{uv}^0 = \sum_{j \in T} n_j(n_j - 1)/2$ , onde  $n_j = \sum_{s \in s(j)} t(j, s) \forall j \in T$ .*

*Demonstração.* A otimalidade das soluções é garantida pelos dois lemas acima. Além disso, em uma solução reduzida, cada subgrafo  $j \in T$  possui exatamente  $n_j$  vértices e, portanto,  $n_j(n_j - 1)/2$  arestas. Por outro lado, em uma solução pura,  $Z_{uv} = 1$  se, e somente se,  $u$  e  $v$  estão em um mesmo subgrafo  $j \in T$ . Logo, o somatório das variáveis  $Z_{uv}$  em uma solução pura é exatamente a soma das arestas dos subgrafos. Daí, temos a igualdade estabelecida no enunciado.  $\square$

Os resultados auxiliares acima são úteis para demonstrarmos a corretude da Formulação 1.

**Proposição 5.5.** *Seja  $I_{FEST}$  uma instância de  $FEST$ . Toda solução viável para a Formulação 1 de valor  $w$  define uma solução viável para  $I_{FEST}$  de valor menor ou igual a  $w$ . Toda solução viável para  $I_{FEST}$  pode ser mapeada em uma solução viável de igual valor para a Formulação 1.*

*Demonstração.* Seja  $(X, Z)$  uma solução viável para (5.4)–(5.8) e  $(X, Z^0)$  sua purificação. Temos que  $d(X, Z^0) \leq d(X, Z)$ . A solução para  $FEST$  é obtida alocando um indivíduo

$u$  na equipe  $j$  desempenhando a habilidade  $s$  se, e somente se,  $X_{uj_s} = 1$ . Vamos mostrar que essa alocação satisfaz as condições 1 e 2 do Problema 3.2.

Por (5.5) é garantido que, para cada vértice  $u \in V(G)$ , no máximo uma variável  $X_{uj_s}$ , para todo  $j \in T$  e  $s \in s(u, j)$ , terá valor igual a 1. Logo, podemos garantir que toda solução viável da Formulação 1 restringe que cada vértice  $u$  seja alocado em no máximo um subgrafo  $j$ , assumindo apenas uma habilidade  $s$  (item 1 da definição do *FEST*). Por (5.6) é garantido que, para cada  $j \in T$  e para cada  $s \in s(j)$ , existirão no mínimo  $t(j, s)$  variáveis  $X_{uj_s}$  com valor igual a 1, onde  $u \in V(G)$ , o que corresponde no *FEST* a satisfazer todas as demandas  $t(j, s)$  de vértices assumindo habilidade  $s$  em cada subgrafo  $j$ , para todo  $j \in T$  e para todo  $s \in S$  (item 2 da definição do *FEST*). Logo, obtemos uma solução viável para *FEST*.

Adicionalmente, pelo Lema 5.3, temos que  $Z_{uv}^0 = 1$  se, e somente se,  $\sum_{s \in s(u, j)} X_{uj_s} = \sum_{s \in s(v, j)} X_{vj_s} = 1$ , para algum  $j \in T$ , ou seja, quando  $u$  e  $v$  estiverem em uma mesma equipe  $j$ . Logo,  $d(X, Z^0) = \sum_{uv \in E: Z_{uv}^0 = 1} d(uv)$  é o valor da solução construída para *FEST*, que portanto é menor ou igual a  $d(X, Z)$ .

Considere agora uma solução viável para *FEST* e defina as variáveis  $X_{uj_s}$  e  $Z_{uv}$  segundo (5.1) e (5.2). Pelo item 1 da definição do *FEST*, temos que toda solução para o *FEST* deve sempre alocar um vértice  $u \in V(G)$  em no máximo um subgrafo  $j$ , desempenhando exatamente uma habilidade  $s \in S$  no subgrafo onde está alocado. Logo, a restrição (5.5) da Formulação 1 é satisfeita. O item 2 da definição de *FEST* garante que toda solução deve satisfazer as demandas para cada habilidade  $s \in s(j)$  para cada subgrafo  $j$ , ou seja, em cada subgrafo  $j$  deve existir pelo menos  $t(j, s)$  vértices desempenhando a habilidade  $s$ , para todo  $j \in T$  e para todo  $s \in (s)$ . Este requisito leva a que a restrição (5.6) seja satisfeita. Além disso, por (5.1) e (5.2), temos que se  $Z_{uv} = 0$  então  $\sum_{s \in s(u, j)} X_{uj_s} = 0$  ou  $\sum_{s \in s(v, j)} X_{vj_s} = 0$ , para todo  $j \in T$ . Deste fato e de (5.5), deduzimos que a restrição (5.4) é válida para  $(X, Z)$ . Também as definições (5.1) e (5.2) asseguram que  $d(X, Z)$  é igual ao valor da solução de *FEST* tomada como base.  $\square$

**Corolário 5.6.** *Seja  $I_{FEST}$  uma instância de *FEST*. Toda solução ótima obtida pela Formulação 1 define uma solução ótima para  $I_{FEST}$ , e toda solução ótima para  $I_{FEST}$  pode ser mapeada em uma solução ótima para a Formulação 1.*

*Demonstração.* Seja  $(X^*, Z^*)$  uma solução ótima (5.3)–(5.8) com valor  $d(X^*, Z^*)$ . Pela Proposição 5.5,  $(X^*, Z^*)$  define uma solução  $Q$  viável para  $I_{FEST}$  de valor  $d(Q) \leq d(X^*, Z^*)$ . Por outro lado, a mesma proposição assegura que  $Q$  é mapeada em uma solução  $(\bar{X}, \bar{Z})$  viável para a formulação tal que  $d(\bar{X}, \bar{Z}) = d(Q) \leq d(X^*, Z^*)$ . Logo,  $(\bar{X}, \bar{Z})$  também é ótima. Consequentemente,  $Q$  é ótima; do contrário, pela proposição, haveria uma solução viável para a formulação de valor menor que  $d(\bar{X}, \bar{Z})$ .  $\square$

Mostramos agora que podemos admitir que  $\min\{d(uv) : uv \in E\} = 0$ .

**Proposição 5.7.** *Dada uma instância  $I_{FEST} = (G, d, T, S, t, s)$ , onde  $d : E(G) \rightarrow \mathbb{N}$ , considere a instância  $I'_{FEST} = (G, d', T, S, t, s)$ , com  $d' : E(G) \rightarrow \mathbb{N}$ , onde  $d'(uv) = d(uv) - \min\{d(wz) : wz \in E\}$  para todo  $uv \in E(G)$ . Se  $(X, Z)$  é uma solução ótima para  $I'_{FEST}$ , então  $(X^0, Z^0)$  é ótima para  $I_{FEST}$ .*

*Demonstração.* Seja  $k := \min\{d(wz) : wz \in E\}$ . É suficiente considerar o caso onde  $k > 0$ . Note que  $d'(uv) \geq 0$ ,  $\forall uv \in E$  e, portanto,  $I'_{FEST}$  é realmente uma instância de  $FEST$ , para a qual se aplicam todos os resultados acima, nesta seção. Observe também que as soluções viáveis de  $I'_{FEST}$  e  $I_{FEST}$  são as mesmas.

Seja  $(X, Z)$  uma solução ótima de  $I'_{FEST}$ . Pelo Corolário 5.6,  $(X^0, Z^0)$  é ótima para  $I'_{FEST}$  e,

$$d'(X, Z) = d'(X^0, Z^0) = \sum_{uv \in E} (d(uv) - k)Z_{uv}^0 = \sum_{uv \in E} d(uv)Z_{uv}^0 - km = d(X^0, Z^0) - km,$$

onde  $m := \sum_{j \in T} n_j(n_j - 1)/2$  e  $n_j = \sum_{s \in s(j)} t(j, s) \forall j \in T$ .

Considere agora uma solução ótima  $(\bar{X}, \bar{Z})$  para  $I_{FEST}$ . Como  $k > 0$ , deve ocorrer  $(\bar{X}, \bar{Z}) = (X^0, Z^0)$ . De fato, qualquer conversão de 1 para 0 de uma componente em  $X^0$  possibilitaria o mesmo em  $Z^0$ , e uma tal possibilidade em  $Z^0$  reduziria estritamente a função objetivo, pois  $k > 0$ . Como  $(\bar{X}, \bar{Z})$  é uma solução reduzida, pura e viável para  $I'_{FEST}$ , temos que

$$d'(X, Z) \leq d'(\bar{X}, \bar{Z}) = \sum_{uv \in E} (d(uv) - k)\bar{Z}_{uv} = \sum_{uv \in E} d(uv)\bar{Z}_{uv} - km = d(\bar{X}, \bar{Z}) - km.$$

As duas últimas expressões destacadas levam a  $d(X^0, Z^0) \leq d(\bar{X}, \bar{Z})$ . Então, como  $(X^0, Z^0)$  é viável para  $I_{FEST}$ , concluímos que será, na verdade, ótima.  $\square$

## 5.2 Estudo poliédrico da Formulação 1

Nesta seção, estudamos o politopo associado a Formulação 1, ou seja,

$$P_1(FEST) = \text{conv}\{(X, Z) : (X, Z) \text{ satisfaz (5.4)–(5.8)}\}.$$

### 5.2.1 Dimensão do politopo

Pela Proposição 4.2 e Teorema 3.2, uma condição necessária e suficiente para a existência de um ponto em  $P_1(FEST)$  é dada pela solução de um problema de fluxo máximo. Aqui apresentamos uma condição equivalente, porém mais adequada para o desenvolvimento dessa seção.

Para cada  $T' \in T$  e  $S' \subseteq S$ , denotamos por

$$n(T', S') = \sum_{j \in T'} \sum_{s \in S' \cap s(j)} t(j, s)$$



a demanda de indivíduos nas equipes indexadas por  $T'$  para desempenhar as habilidades indexadas por  $S'$ . Quando  $T' = \{j\}$  ou  $S' = \{s\}$ , a notação é simplificada para  $n(j, S')$  ou  $n(T', s)$ . Usamos

$$V(S') = \{u \in V(G) : s(u) \cap S' \neq \emptyset\}$$

para denotar os indivíduos que possuem alguma das habilidades em  $S'$ . Adotamos

$$s(T') = \bigcup_{j \in T'} s(j).$$

Supomos, sem perda de generalidade, que  $s(T) = S$  e  $V(S) = V(G)$ , ou seja, toda habilidade é requerida por algum time e todo indivíduo tem alguma habilidade demanda por algum time.

**Proposição 5.8.**  $P_1(FEST) \neq \emptyset$  se, e somente se,  $|V(S')| \geq n(T, S') \forall S' \subseteq S$ .

*Demonstração.*  $P_1(FEST) \neq \emptyset$  se, e se somente, existe solução viável para (5.4)–(5.8). Pela Proposição 5.5, esta segunda condição é equivalente a pedir que exista solução viável para  $FEST$ . Argumentamos aqui que a viabilidade de  $FEST$  é, por sua vez, equivalente à existência de um emparelhamento que satura  $W$  no grafo bipartido  $H = (U \cup W, E(H))$ , onde

$$U = \{v_u : u \in V(G)\}, \quad W = \{v_{js}^k : j \in T, s \in s(j), k = 1, 2, \dots, t(j, s)\},$$

$$E(H) = \{v_u v_{js}^k \in U \times W : s \in s(u, j), k = 1, 2, \dots, t(j, s)\}.$$

A Figura 3 apresenta o grafo bipartido  $H = (U \cup W, E(H))$  referente à instância usada como exemplo na Figura 1.

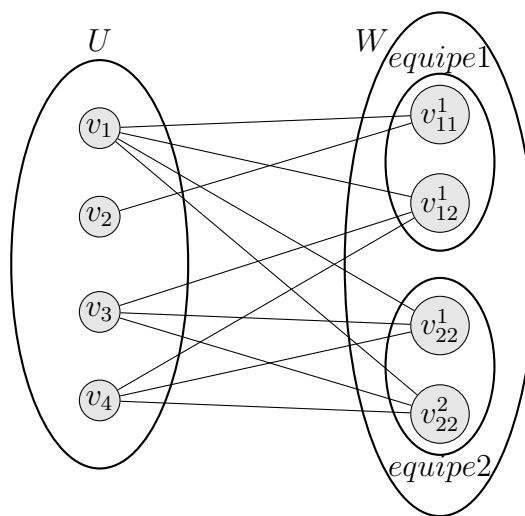


Figura 3 – Grafo bipartido  $H = (U \cup W, E(H))$  para  $U = \{v_1, v_2, v_3, v_4\}$ ,  $T = \{1, 2\}$  e  $S = \{1, 2\}$ , sendo  $s(v_1) = \{1, 2\}$ ,  $s(v_2) = \{1\}$ ,  $s(v_3) = s(v_4) = \{2\}$ ,  $t(1, 1) = t(1, 2) = 1$ ,  $t(2, 1) = 0$  e  $t(2, 2) = 2$ .

Note que esse grafo  $H$  tem, de um lado, os indivíduos  $e$ , do outro, um par equipe-habilidade, repetido tantas vezes quantas sejam a demanda de indivíduos para esse par. Por sua vez, as arestas de  $H$  relacionam as possíveis alocações de um indivíduo a um par equipe-habilidade. Dessa forma, uma solução viável para  $FEST$  é um emparelhamento em  $H$  que sature o conjunto  $W$ . Pelo Teorema de Hall, tal emparelhamento existe se, e somente se,  $|W'| \leq |N_H(W')| \forall W' \subseteq W$ .

Resta mostrar a equivalência entre essa condição e aquela estabelecida no enunciado da proposição. Primeiro, suponha que a condição de Hall aconteça e tome  $S' \subseteq S$ . Seja  $W' = \{v_{j_s}^k \in W : j \in T, s \in s(j) \cap S', k = 1, 2, \dots, t(j, s)\}$ . Podemos mostrar que  $|W'| = n(T, S')$  e  $|N_H(W')| = |V(S')|$ , de modo que obtemos a desigualdade da proposição para  $S'$ . De fato, temos claramente que  $|W'| = \sum_{j \in T} \sum_{s \in S' \cap s(j)} t(j, s) = n(T, S')$ . Por outro lado, como  $N_H(v_{j_s}^k) = \{u \in U : s \in s(u) \cap s(j)\}$  e  $s(T) = S$ , derivamos que  $N_H(W') = \bigcup_{s \in S'} \bigcup_{j \in T: s \in s(j)} N_H(v_{j_s}^1) = \{u \in U : s(u) \cap S' \neq \emptyset\} = V(S')$ .

Assuma agora que as desigualdades da proposição sejam válidas. Seja  $W' \subseteq W$ . Defina  $S' = \{s : v_{j_s}^k \in W', \text{ para algum } j \in T, s \in s(j) \text{ e } k \in \{0, 1, \dots, t(j, s)\}\}$ . É suficiente mostrar que  $|W'| \leq n(T, S')$  e  $N_H(W') \supseteq V(S')$ . Primeiro, note que  $S'$  contém toda habilidade  $s$  relacionada a algum vértice em  $W'$ , e o número de vértices em  $W'$  relacionados a  $s$  é no máximo  $n(T, s)$ . Portanto,  $|W'| \leq \sum_{s \in S'} n(T, s) = n(T, S')$ . Por outro lado, seja  $u \in V(S')$ . Então, existe  $s \in s(u) \cap S'$ . Pela definição de  $S'$ , existem  $j \in T$  e  $k \in \{0, 1, \dots, t(j, s)\}$  tais que  $s \in s(j)$  e  $v_{j_s}^k \in W'$ . Então  $s \in s(u, j)$  implicando que  $u \in N_H(v_{j_s}^k)$ . Logo,  $V(S') \subseteq N_H(W')$ .  $\square$

Em vários dos próximos resultados, vamos considerar a seguinte hipótese, que é mais forte que a estabelecida acima para termos  $P_1(FEST)$  não-vazio.

**Hipótese 5.1** ( $H_1$ ). *Para todo  $S' \subseteq S$ ,  $|V(S')| \geq n(T, S') + 1$ .*

Observamos que essa hipótese não é restritiva quando cada equipe demanda pelo menos dois indivíduos, ou seja,  $n(j, S) \geq 2 \forall j \in T$ . De fato, acrescentamos no grafo um vértice fantasma  $f$ , com  $s(f) = S$ , e consideramos  $d(uf) = +\infty \forall u \in V(G)$ . Supondo a instância original viável, a instância modificada satisfaz  $H_1$ . Além disso,  $f$  nunca irá ser alocado em alguma equipe em uma solução ótima. Isto mostra que, resolvendo a instância modificada, resolvemos também a instância original.

**Lema 5.9.** *Suponha  $H_1$ . Para todo  $u \in V(G)$ , existe solução viável em que  $u$  não está alocado em qualquer equipe. Para todo  $u \in V(G)$ ,  $j \in T$  e  $s \in s(u, j)$ , existe solução viável em que  $u$  está alocado na equipe  $j$  desempenhando a habilidade  $s$ .*

*Demonstração.* Considere o problema definido com a retirada de  $u$  do grafo. Devido a  $H_1$ , obtemos a condição requerida pela Proposição 5.8 e, portanto, há uma solução viável. Tal solução fornece uma solução viável  $(X, Z)$  para o problema original onde  $u$  não está

alocado. A partir de  $(X, Z)$ , podemos construir outra solução viável  $(\bar{X}, \bar{Z})$ , incluindo  $u$  na equipe  $j$  com habilidade  $s$ . Formalmente,  $\bar{X}$  difere de  $X$  apenas na componente  $\bar{X}_{ujs} = 1 \neq 0 = X_{ujs}$ , e  $\bar{Z}$  tem  $\bar{Z}_{uv} = 1$  para todo  $v$  alocado na equipe  $j$  em  $(X, Z)$ . Esta solução é viável, pois podemos ter mais indivíduos em  $j$  com habilidade  $s$  que a sua demanda mínima  $t(j, s)$ .  $\square$

Uma solução pura onde  $u$  não está alocado em qualquer equipe será denotada genericamente por  $(X^{\bar{u}}, Z^{\bar{u}})$ , enquanto uma solução pura onde  $u$  está alocado em alguma equipe será representada por  $(X^u, Z^u)$ . Neste último caso, se queremos identificar a equipe  $j$  a que  $u$  pertence e a habilidade  $s$  que desempenha, estedemos a notação para  $(X^{ujs}, Z^{ujs})$ . Usamos  $e^{uv} \in \mathbb{B}^{|E(G)|}$  e  $e^{ujs} \in \mathbb{B}^{|V| \times |T| \times |S|}$ , respectivamente, para os vetores unitários com 1 na posição  $(u, v)$  e na posição  $(u, j, s)$ . Note que  $X^{ujs} = X^{\bar{u}} + e^{ujs}$ .

**Proposição 5.10.**  $P_1(FEST)$  tem dimensão plena se, e somente se,  $H_1$  é válida.

*Demonstração.* Suponha, primeiro, que  $H_1$  não é válida. Então existe  $S' \subseteq S$  tal que  $|V(S')| \leq n(T, S')$ . Se ocorre desigualdade estrita, então não há indivíduos suficientes para atender a demanda das habilidades em  $S'$ , e o problema é inviável. No caso de igualdade, todos os indivíduos que possuem alguma habilidade em  $S'$  estarão alocados em qualquer solução viável. Em outras palavras,  $\sum_{j \in T} \sum_{s \in S' \cap s(j)} X_{ujs} = 1$ , para todo  $u \in V(G)$  tal que  $s(u) \cap S' \neq \emptyset$ , são igualdades de  $P_1(FEST)$ . Em ambos os casos, o politopo não tem dimensão plena.

Suponha agora que  $H_1$  é válida. Considere que

$$P_1(FEST) \subseteq \{(X, Z) : \lambda^T X + \pi^T Z = \theta\}.$$

Sejam  $u, v \in V$ . Pelo Lema 5.9,  $(X^{\bar{u}}, Z^{\bar{u}}) \in P_1(FEST)$ . Note que  $Z_{uv}^{\bar{u}} = 0$ , pois  $(X^{\bar{u}}, Z^{\bar{u}})$  é pura, e que  $(X^{\bar{u}}, Z^{\bar{u}} + e^{uv}) \in P_1(FEST)$ , porque sempre podemos converter para 1 uma entrada em  $Z$  em qualquer solução viável. Por conseguinte,  $\lambda^T X^{\bar{u}} + \pi^T Z^{\bar{u}} = \lambda^T X^{\bar{u}} + \pi^T (Z^{\bar{u}} + e^{uv})$  ou ainda  $\pi^T e^{uv} = \pi_{uv} = 0$ . Então,  $\pi = \mathbf{0}$ .

Sejam  $j \in T$ ,  $s \in s(u, j)$ . Pelo Lema 5.9,  $(X^{ujs}, Z^{ujs}) \in P_1(FEST)$ . Lembre  $X^{ujs} = X^{\bar{u}} + e^{ujs}$ . Então,  $0 = \lambda^T (X^{ujs} - X^{\bar{u}}) + \pi^T (Z^{ujs} - Z^{\bar{u}}) = \lambda^T e^{ujs} = \lambda_{ujs}$ . Portanto,  $\lambda = \mathbf{0}$ .

Como  $P_1(FEST) \neq \emptyset$ , então  $\theta = 0$ . Pelo Corolário 2.6, temos que  $P_1(FEST)$  tem dimensão plena.  $\square$

## 5.2.2 Estudo facial

Analisamos agora quando as desigualdades que definem a Formulação 1 ou generalizações delas induzem facetas de  $P_1(FEST)$ . Em cada caso, consideramos uma desigualdade válida que define uma face não-vazia  $F$  de  $P_1(FEST)$  e supomos que  $F \subseteq F'$ , onde

$$F' := \{(X, Z) : \lambda^T X + \pi^T Z = \theta\}.$$

Admitindo a hipótese  $H_1$ , que garante ser o politopo de dimensão plena, mostramos que os coeficientes em  $(\lambda, \pi, \theta)$  são proporcionais àqueles da desigualdade válida em consideração. De acordo com o Teorema 2.8, isto mostra que  $F$  é uma faceta. Na verdade, basta considerar as entradas em  $(\lambda, \pi)$ , pois, se satisfazem a proporcionalidade, o mesmo ocorrerá com  $\theta$ .

**Proposição 5.11.** *Suponha  $H_1$ . Para todo  $u \in V(G)$ , a desigualdade  $\sum_{j \in T} \sum_{s \in s(u,j)} X_{uj s} \leq 1$  induz faceta de  $P_1(FEST)$ .*

*Demonstração.* Dado  $u \in V$ , seja  $F := \left\{ (X, Z) \in P_1(FEST) : \sum_{j \in T} \sum_{s \in s(u,j)} X_{uj s} = 1 \right\}$ . Suponha  $F \subseteq F'$ . Devemos mostrar que  $\pi = 0$  e que existe  $a \in \mathbb{R}$  tal que, para todos  $w \in V(G)$ ,  $j \in T$ ,  $s \in s(w, j)$ ,

$$\lambda_{wjs} = \begin{cases} a, & \text{se } w = u, \\ 0, & \text{caso contrário.} \end{cases}$$

Primeiro observe que toda solução viável  $(X, Z)$  na qual  $u$  participa de alguma equipe pertence a  $F$  e, conseqüentemente, satisfaz  $\lambda^T X + \pi^T Z = \theta$ . Vamos mostrar primeiro que  $\pi = 0$ .

Sejam  $w \in V \setminus \{u\}$  e  $v \in V \setminus \{w\}$ . Sem perda de generalidade, podemos considerar que  $u$  está alocado em alguma equipe na solução pura  $(X^{\bar{w}}, Z^{\bar{w}})$ . Sendo assim, os pontos  $(X^{\bar{w}}, Z^{\bar{w}})$  e  $(X^{\bar{w}}, Z^{\bar{w}} + e^{wv})$  pertencem a  $P_1(FEST)$  e, mais ainda, estão em  $F \subseteq F'$ . Portanto, ambos satisfazem a equação  $\lambda^T X + \pi^T Z = \theta$ . Daí, concluímos que  $\pi_{wv} = 0$ . Isto mostra todos os zeros em  $\pi$ , ou seja, que  $\pi_{wv} = 0 \forall w, v \in V, w \neq v$ .

Agora, sejam  $w \in V$ ,  $j \in T$ ,  $s \in s(w, j)$ . Primeiro, suponha que  $w \neq u$ . Na solução  $(X^{\bar{w}}, Z^{\bar{w}}) \in F'$  dada acima, inclua  $w$  na equipe  $j$  com habilidade  $s$  para gerar  $(X^{wjs}, Z^{wjs})$ . Esta nova solução também está em  $F \subseteq F'$ . Sendo assim,  $0 = \lambda^T (X^{wjs} - X^{\bar{w}}) + \pi^T (Z^{wjs} - Z^{\bar{w}}) = \lambda^T e^{wjs} = \lambda_{wjs}$ . Finalmente, suponha  $w = u$ . Note que  $(X^{uj s}, Z^{uj s}) \in F \subseteq F'$ . Considerando as entradas nulas em  $\pi$  e  $\lambda$  já demonstradas, temos que  $\theta = \lambda^T X^{uj s} + \pi^T Z^{uj s} = \lambda_{uj s}$ .  $\square$

Para as próximas facetas, consideramos a seguir uma generalização da hipótese  $H_1$ . Note que  $H_\epsilon$  é mais forte que  $H_{\epsilon'}$  se  $\epsilon > \epsilon'$ .

**Hipótese 5.2 ( $H_\epsilon$ ).** *Para todo  $S' \subseteq S$ ,  $|V(S')| \geq n(T, S') + \epsilon$ .*

Para as duas outras desigualdades da Formulação 1, usamos a hipótese  $H_2$  como condição suficiente para indução de facetas. Com isso, garantimos a existência de soluções viáveis com as propriedades estabelecidas no lema abaixo, cuja demonstração é bem semelhante àquela do Lema 5.9.

**Lema 5.12.** *Suponha  $H_2$ . Para todos  $u, v \in V(G)$ , é válido afirmar que:*

1. existe solução viável pura em que  $u$  e  $v$  não estão alocados em qualquer equipe, a ser denotada  $(X^{\bar{u},\bar{v}}, Z^{\bar{u},\bar{v}})$ .
2. para todo  $j \in T$  e  $s \in s(v, j)$ , existe solução viável, a ser denotada  $(X^{\bar{u},vjs}, Z^{\bar{u},vjs})$ , em que  $u$  não está alocado e  $v$  está alocado na equipe  $j$  desempenhando a habilidade  $s$ .
3. para todos  $j, \ell \in T$ ,  $s \in s(u, j)$  e  $r \in s(v, \ell)$ , existe solução viável, a ser denotada  $(X^{uj s, v\ell r}, Z^{uj s, v\ell r})$ , em que  $u$  e  $v$  estão alocados, respectivamente, na equipe  $j$  com habilidade  $s$  e na equipe  $\ell$  com habilidade  $r$ .

*Demonstração.* Considere o problema definido para o grafo  $G' = G[V \setminus \{u, v\}]$ . Por  $H_2$ , obtemos para  $G'$  a condição requerida pela Proposição 5.8 e, portanto, há uma solução viável para o problema sobre  $G'$ . Tal solução se traduz em  $(X^{\bar{u},\bar{v}}, Z^{\bar{u},\bar{v}})$ . A partir desta, podemos construir  $(X^{\bar{u},vjs}, Z^{\bar{u},vjs})$ , incluindo  $v$  na equipe  $j$  com habilidade  $s$ . Similarmente, obtemos  $(X^{uj s, v\ell r}, Z^{uj s, v\ell r})$ , adicionando  $u$  à equipe  $j$  com habilidade  $s$  e  $v$  à equipe  $\ell$  com habilidade  $r$ .  $\square$

Quando não for preciso especificar a equipe ou habilidade associada à uma alocação, a notação acima definida será simplificada com a retirada dos super-índices correspondentes. Novamente, destacamos que  $X^{\bar{u},vjs} = X^{\bar{u},\bar{v}} + e^{vjs}$  e  $X^{uj s, v\ell r} = X^{\bar{u},\bar{v}} + e^{uj s}$ .

**Proposição 5.13.** *Suponha  $H_2$ . Para todo  $j \in T$  e  $s \in s(j)$ , a desigualdade  $\sum_{u \in V(s)} X_{uj s} \geq t(j, s)$  induz faceta de  $P_1(FEST)$ .*

*Demonstração.* Seja  $F := \left\{ (X, Z) \in P_1(FEST) : \sum_{u \in V(s)} X_{uj s} = t(j, s) \right\}$ , onde  $j \in T$  e  $s \in s(j)$  são dados. Suponha  $F \subseteq F'$ . Devemos mostrar que  $\pi = 0$  e que existe  $a \in \mathbb{R}$  tal que, para todo  $w \in V(G)$ ,  $\ell \in T$ ,  $r \in s(w, \ell)$ ,

$$\lambda_{wjs} = \begin{cases} a, & \text{se } \ell = j \text{ e } r = s, \\ 0, & \text{caso contrário.} \end{cases}$$

Primeiro observe que toda solução viável reduzida pertence a  $F$ . Vamos mostrar primeiro que  $\pi = 0$ .

Sejam  $w \in V \setminus \{u\}$  e  $v \in V \setminus \{w\}$ . Sem perda de generalidade, podemos considerar que a solução  $(X^{\bar{w}}, Z^{\bar{w}})$  é reduzida e pura. Sendo assim, os pontos  $(X^{\bar{w}}, Z^{\bar{w}})$  e  $(X^{\bar{w}}, Z^{\bar{w}} + e^{wv})$  pertencem  $F \subseteq F'$ , satisfazendo a equação  $\lambda^T X + \pi^T Z = \theta$ . Daí, concluímos que  $\pi_{wv} = 0$ . Isto mostra todos os zeros em  $\pi$ , ou seja, que  $\pi_{wv} = 0 \forall w, v \in V, w \neq v$ .

Agora, sejam  $w \in V$ ,  $\ell \in T$ ,  $r \in s(w, \ell)$ . Primeiro, suponha que  $\ell \neq j$  ou  $r \neq s$ . Incluindo  $w$  na equipe  $\ell$  com habilidade  $r$  na solução  $(X^{\bar{w}}, Z^{\bar{w}}) \in F'$  dada acima, geramos uma nova solução  $(X^{w\ell r}, Z^{w\ell r})$  também em  $F \subseteq F'$ . Embora não seja reduzida, esta nova solução mantém a igualdade que define  $F$ . Sendo assim,  $0 = \lambda^T (X^{w\ell r} - X^{\bar{w}}) + \pi^T (Z^{w\ell r} - Z^{\bar{w}}) = \lambda^T e^{w\ell r} = \lambda_{w\ell r}$ .

Finalmente, suponha que  $\ell = j$  e  $r = s$ . Então,  $w \in V(s)$ . Tome  $v \in V(s)$ ,  $v \neq w$ . Considere a solução  $(X^{\bar{w},vjs}, Z^{\bar{w},vjs})$ , que existe devido a  $H_2$ . Podemos considerar que ela é reduzida (se necessário a redução pode ser feita mantendo  $v$  na equipe  $j$  com habilidade  $s \in s(v, j)$ ). Então, ela pertence a  $F \subseteq F'$ . Considere também a solução  $(X, Z)$ , reduzida e pura, obtida de  $(X^{\bar{w},vjs}, Z^{\bar{w},vjs})$  com a substituição de  $v$  por  $w$  na equipe  $j$ . Ou seja,  $X = X^{\bar{w},vjs} + e^{wjs} - e^{vjs}$  e  $Z = Z^{\bar{w},vjs} + \sum_{u: Z_{uv}^{\bar{w},vjs}=1} (e^{uw} - e^{uv})$ . Note que  $(X, Y) \in F'$ . Então, usando  $\pi = \mathbf{0}$ , chegamos a  $0 = \lambda^T(X - X^{\bar{w},vjs}) + \pi^T(Z - Z^{\bar{w},vjs}) = \lambda_{wjs} - \lambda_{vjs}$ , ou seja,  $\lambda_{wjs} = \lambda_{vjs}$ .  $\square$

**Proposição 5.14.** *Suponha  $H_2$ . Para todos  $j \in T$ ,  $u, v \in V(G)$  tais que  $s(u, j) \neq \emptyset$  e  $s(v, j) \neq \emptyset$ , a desigualdade  $\sum_{s \in s(u, j)} X_{ujs} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} \leq 1$  induz faceta de  $P_1(FEST)$ .*

*Demonstração.* Seja  $F := \left\{ (X, Z) \in P_1(FEST) : \sum_{s \in s(u, j)} X_{ujs} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} = 1 \right\}$ , onde  $j \in T$  e  $u, v \in V(G)$ , tais que  $s(u, j) \neq \emptyset$  e  $s(v, j) \neq \emptyset$ , são dados. Suponha  $F \subseteq F'$ . Devemos mostrar que existe  $a \in \mathbb{R}$  tal que

$$\lambda_{w\ell r} = \begin{cases} a, & \text{se } w \in \{u, v\} \text{ e } \ell = j, \\ 0, & \text{caso contrário,} \end{cases} \quad \pi_{wy} = \begin{cases} -a, & \text{se } (w, y) = (u, v), \\ 0, & \text{caso contrário,} \end{cases}$$

para todos  $w, y \in V$ ,  $\ell \in T$ ,  $r \in s(w, \ell)$ .

Primeiro observe que toda solução pura  $(X, Z)$  onde  $u$  ou  $v$  participem da equipe  $j$  pertence a  $F$ . Vamos mostrar primeiro as entradas nulas em  $\pi$  e  $\lambda$ .

Sejam  $w \in V \setminus \{u, v\}$  e  $y \in V \setminus \{w\}$ . Os pontos  $(X^{\bar{w},u}, Z^{\bar{w},u})$  e  $(X^{\bar{w},u}, Z^{\bar{w},u} + e^{wy})$  pertencem  $P_1(FEST)$  e, mais ainda, estão em  $F \subseteq F'$ . Portanto, ambos satisfazem a equação  $\lambda^T X + \pi^T Z = \theta$ . Daí, concluímos que  $\pi_{wy} = 0$ . Isto mostra todos os zeros em  $\pi$ , ou seja, que  $\pi_{wy} = 0 \forall (w, y) \neq (u, v)$ .

Sejam  $w \in V$ ,  $\ell \in T$ ,  $r \in s(w, \ell)$ . Primeiro, suponha que  $w \notin \{u, v\}$ . Considere  $(X^{\bar{w},u}, Z^{\bar{w},u}) \in F'$  como acima. Obtenha  $(X^{w\ell r, u}, Z^{w\ell r, u})$  com a inclusão de  $w$  na solução  $(X^{\bar{w},u}, Z^{\bar{w},u})$ , de modo que  $u$  e  $v$  estejam alocados de forma idêntica nas duas soluções. Sendo assim,  $Z_{uv}^{\bar{w},u} = Z_{uv}^{w\ell r, u}$  levando a  $\pi^T(Z^{w\ell r, u} - Z^{\bar{w},u}) = 0$ . Como as duas soluções estão em  $F'$ , deduzimos que  $0 = \lambda^T(X^{w\ell r, u} - X^{\bar{w},u}) + \pi^T(Z^{w\ell r, u} - Z^{\bar{w},u}) = \lambda^T e^{w\ell r} = \lambda_{w\ell r}$ .

Agora suponha que  $w \in \{u, v\}$ ,  $\ell \neq j$  e  $r \in s(w, \ell)$ . Sejam  $w' \in \{u, v\} \setminus \{w\}$  e  $s \in s(w', j)$ . Temos que  $(X^{\bar{w},w'js}, Z^{\bar{w},w'js}) \in F \subseteq F'$ . Além disso, incluindo nessa solução o indivíduo  $w$  na equipe  $\ell$  com habilidade  $r$ , obtemos a solução  $(X^{w\ell r, w'js}, Z^{w\ell r, w'js})$  também pertencente a  $F \subseteq F'$ . Note que  $Z_{uv}^{w\ell r, w'js} = Z_{uv}^{\bar{w},w'js} = 0$ , pois  $u$  e  $v$  não estão na mesma equipe em ambas as soluções. Este fato e as entradas nulas em  $\pi$  levam a  $0 = \lambda^T(X^{w\ell r, w'js} - X^{\bar{w},w'js}) + \pi^T(Z^{w\ell r, w'js} - Z^{\bar{w},w'js}) = \lambda^T e^{w\ell r} = \lambda_{w\ell r}$ . Com isso, mostramos todas as entradas nulas em  $\lambda$ .

Mostramos finalmente as entradas não-nulas. Sejam  $\{w, w'\} = \{u, v\}$ ,  $r \in s(w, j)$  e  $s \in s(w', j)$ . Considere uma solução  $(X^{\bar{w},w'js}, Z^{\bar{w},w'js})$ , onde podemos incluir  $w$  na mesma

equipe  $j$  com habilidade  $r$  para gerar  $(X^{wjr,w'js}, Z^{wjr,w'js})$ . Ambas as soluções pertencem a  $F'$ . Note que  $Z_{uv}^{\bar{w},w'} = 0$  e  $Z_{uv}^{w,w'} = 1$ . Então temos que  $0 = \lambda^T(X^{wjr,w'js} - X^{\bar{w},w'js}) + \pi^T(Z^{wjr,w'js} - Z^{\bar{w},w'js}) = \lambda_{wjr} + \pi_{uv}$ , ou seja,  $\lambda_{wjr} = -\pi_{uv}$ . Isto mostra a correta relação entre os coeficientes não-nulos em  $\lambda$  e  $\pi$ .  $\square$

Apresentamos uma recíproca para a Proposição 5.14.

**Proposição 5.15.** *Sejam  $j \in T$ ,  $u, v \in V(G)$  tais que  $s(u, j) \neq \emptyset$  e  $s(v, j) \neq \emptyset$ . A desigualdade  $\sum_{s \in s(u, j)} X_{uj s} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} \leq 1$  não induz faceta de  $P_1(FEST)$  nos seguintes casos:*

1. *Se  $u$  for alocado em  $j$ , então  $v$  também será alocado em  $j$ ;*
2. *Se  $u$  for alocado em  $j$ , então  $v$  não poderá ser alocado em  $j$ ;*
3. *Se  $w \neq u, v$  for alocado em  $\ell = j$  com habilidade  $r$ , então  $u$  e  $v$  não poderão ser alocados em  $j$ ;*
4. *Se  $w = u$  for alocado em  $\ell \neq j$  com habilidade  $r$ , então  $v$  também será alocado em  $\ell$ ;*
5. *Se  $w \neq u, v$  for alocado em  $\ell \neq j$  com habilidade  $r$ , então  $u$  e  $v$  também serão alocados em  $\ell$ ;*

*Demonstração.* No caso 1, a desigualdade é dominada por  $\sum_{s \in s(u, j)} X_{uj s} - Z_{uv} \leq 0$  e  $\sum_{s \in s(v, j)} X_{vjs} \leq 1$ . No caso 2, a desigualdade  $\sum_{s \in s(u, j)} X_{uj s} + \sum_{s \in s(v, j)} X_{vjs} \leq 1$  é mais forte. Nos casos 3, 4 e 5, seja  $S' \subseteq s(w, \ell)$  o conjunto de habilidades  $r$  com a propriedade mencionada. A desigualdade é dominada por  $\sum_{r \in S'} X_{wjr} + \sum_{s \in s(u, j)} X_{uj s} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} \leq 1$  no caso 3 e por  $2 \sum_{r \in S'} X_{wlr} + \sum_{s \in s(u, j)} X_{uj s} + \sum_{s \in s(v, j)} X_{vjs} - Z_{uv} \leq 1$  nos outros dois.  $\square$

As desigualdades (5.4) podem ser generalizadas como segue.

**Proposição 5.16.** *Para todos  $j \in T$ ,  $V' \subseteq \{v \in V : s(v, j) \neq \emptyset\}$  e  $\alpha \in \mathbb{Z}$ , a desigualdade*

$$\sum_{\substack{u, v \in V' \\ u \neq v}} Z_{uv} \geq \sum_{u \in V'} \sum_{s \in s(u, j)} \alpha X_{uj s} - \frac{\alpha(\alpha + 1)}{2} \quad (5.9)$$

*é válida para  $P_1(FEST)$ . Uma solução pura satisfaz a inequação na igualdade se, dentre os indivíduos de  $V'$ ,  $\alpha$  ou  $\alpha + 1$  estão alocados na equipe  $j$  e no máximo 1 está alocado em outra equipe.*

*Demonstração.* Sejam  $j \in T$ ,  $V' \subseteq \{v \in V : s(v, j) \neq \emptyset\}$  e  $\alpha \in \mathbb{Z}$ . Seja  $(X, Z)$  viável para a Formulação 1. Denote por  $x = \sum_{u \in V'} \sum_{s \in s(u, j)} X_{ujs}$  a quantidade de indivíduos de  $V'$  alocados na equipe  $j$  na solução  $(X, Z)$ . Claramente,

$$\sum_{\substack{u, v \in V' \\ u \neq v}} Z_{uv} \geq \frac{x(x-1)}{2}. \quad (5.10)$$

Então devemos mostrar que  $\frac{x(x-1)}{2} \geq ax - \frac{\alpha(\alpha+1)}{2}$  ou, equivalentemente,  $x^2 - (2\alpha+1)x + \alpha(\alpha+1) \geq 0$ . Note que esta desigualdade ocorre sempre que  $x \leq \alpha$  ou  $x \geq \alpha+1$ . Como  $x \in \mathbb{Z}$  e  $\alpha \in \mathbb{Z}$ , a validade da desigualdade é obtida. A segunda parte do enunciado, segue pelo fato de  $\alpha$  e  $\alpha+1$  serem as raízes dessa equação do segundo grau e de que ocorre igualdade em (5.10) em uma solução pura como a descrita no enunciado.  $\square$

Observe que a restrição (5.4) é um caso particular de (5.9) quando  $V' = \{u, v\}$  e  $\alpha = 1$ . Já sabemos que tal desigualdade induz faceta sob a hipótese  $H_2$ . Estendemos esse resultado para  $|V'| \geq 3$ .

**Proposição 5.17.** *Sejam  $j \in T$ ,  $V' \subseteq \{v \in V : s(v, j) \neq \emptyset\}$ , com  $|V'| \geq 3$ , e  $\alpha \in \{1, 2, \dots, |V'| - 2\}$ . Suponha  $H_\epsilon$ , para  $\epsilon = |V'|$ . A desigualdade (5.9) induz faceta de  $P_1(FEST)$ .*

*Demonstração.* Para  $j \in T$ ,  $V' \subseteq \{v \in V : s(v, j) \neq \emptyset\}$ , com  $|V'| \geq 3$ , e  $\alpha \in \{1, \dots, |V'| - 2\}$ , seja  $F := \left\{ (X, Z) \in P_1(FEST) : \sum_{u \in V'} \sum_{s \in s(u, j)} \alpha X_{ujs} - \sum_{\substack{u, v \in V' \\ u \neq v}} Z_{uv} = \frac{\alpha(\alpha+1)}{2} \right\}$ . Suponha que  $F \subseteq F'$ . Devemos mostrar que existe  $a \in \mathbb{R}$  tal que

$$\lambda_{w\ell r} = \begin{cases} a\alpha, & \text{se } w \in V' \text{ e } \ell = j, \\ 0, & \text{caso contrário,} \end{cases} \quad \pi_{wv} = \begin{cases} -a, & \text{se } w, v \in V' \\ 0, & \text{caso contrário,} \end{cases}$$

para todos  $w, v \in V$ ,  $w \neq v$ ,  $\ell \in T$ ,  $r \in s(w, \ell)$ .

Duas propriedades serão usadas nessa prova. Pela Proposição 5.16, uma solução pura pertence a  $F$  se nela estão alocados  $\alpha$  ou  $\alpha+1$  indivíduos de  $V'$  na equipe  $j$  e no máximo 1 indivíduo de  $V'$  em outra equipe. Pela hipótese  $H_\epsilon$ , para qualquer subconjunto  $W \subset V$ , com  $|W| \leq |V'|$ , existe uma solução pura onde nenhum indivíduo de  $W$  está alocado.

Sejam  $w \in V$ ,  $\ell \in T$  e  $r \in s(w, \ell)$ . Considere um conjunto  $W \subseteq V' \setminus \{w\}$  com  $|W| = |V'| - 1$ . Logo,  $|W| \geq \alpha$  e, dependendo se  $w \in V'$  ou  $w \notin V'$ , temos que  $|V' \setminus (W \cup \{w\})|$  igual a 0 ou 1, respectivamente. Tome uma solução em que nenhum indivíduo de  $W \cup \{w\}$  está alocado e inclua, na equipe  $j$ ,  $\alpha$  indivíduos de  $W$ , obtendo uma solução pura  $(\bar{X}, \bar{Z})$ . Seja também  $(\hat{X}, \hat{Z})$  a solução pura obtida de  $(\bar{X}, \bar{Z})$  com a inclusão de  $w$  na equipe  $\ell$  com habilidade  $r$ .

Primeiro suponha que  $w \notin V'$ . Então  $(\bar{X}, \bar{Z})$  tem  $\alpha$  ou  $\alpha+1$  indivíduos de  $V'$  na equipe em  $j$  e no máximo um elemento de  $V'$  em outra equipe. O mesmo acontece com



$(\hat{X}, \hat{Z})$ . Sendo assim, os pontos  $(\bar{X}, \bar{Z})$ ,  $(\bar{X}, \bar{Z} + e^{wv})$  e  $(\hat{X}, \hat{Z})$  estão em  $F \subseteq F'$ , para qualquer  $v \in V \setminus \{w\}$ . As duas primeiras soluções mostram que  $\pi_{wv} = 0 \forall v \in V \setminus \{w\}$ . Os zeros em  $\pi$ , a primeira e a última soluções levam a  $\lambda_{w\ell r} = 0$ .

Para os zeros em  $\lambda$  que ainda faltam ser mostrados, suponha agora que  $w \in V'$  e  $\ell \in T \setminus \{j\}$ . Neste caso, ambas  $(\bar{X}, \bar{Z})$  e  $(\hat{X}, \hat{Z})$  têm exatamente  $\alpha$  indivíduos de  $V'$  na equipe  $j$ , sendo que apenas a segunda tem mais um outro indivíduo de  $V'$  alocado (na equipe  $\ell \neq j$ ). De novo,  $(\bar{X}, \bar{Z})$  e  $(\hat{X}, \hat{Z})$  pertencem a  $F \subseteq F'$ . Note que  $\hat{X}$  difere de  $\bar{X}$  apenas na componente associada a  $w\ell r$ . Note também que  $\hat{Z}$  difere de  $\bar{Z}$  somente nas componentes indexadas por  $wu$ , para  $u$  alocado em  $\ell$  na solução  $(\bar{X}, \bar{Z})$ , ou seja, com  $u \notin V'$  e, por conseguinte, com  $\pi_{wu} = 0$ . Assim, deduzimos que  $\lambda_{w\ell r} = 0$ .

Finalmente, suponha que  $w \in V'$  e  $\ell = j$  (lembre que  $s(w, j) \neq \emptyset$ ). Como  $w \in V'$ , vimos que há exatamente  $\alpha$  indivíduos de  $V'$  alocados na solução  $(\bar{X}, \bar{Z})$  e que todos estão na equipe  $j$ . Situação similar ocorre com  $(\hat{X}, \hat{Z})$ , com a diferença de que agora são  $\alpha + 1$  indivíduos em  $j$ . Então, ambas as soluções pertencem a  $F'$ . Mais ainda,  $\lambda^T(\hat{X} - \bar{X}) = \lambda^T e^{wj r} = \lambda_{wj r}$  e  $\pi^T(\hat{Z} - \bar{Z}) = \sum_{v \in W_\alpha} \pi_{wv}$ , onde  $W_\alpha \subseteq W$  é o conjunto dos  $\alpha$  indivíduos de  $V'$  alocados em  $j$  na solução  $(\bar{X}, \bar{Z})$ . Para obter a segunda igualdade estamos usando o fato de que  $\pi_{wv} = 0 \forall v \notin V'$ . Consequentemente,  $\lambda_{wj r} = -\sum_{v \in W_\alpha} \pi_{wv}$ . E esta igualdade vale para todo  $w \in V'$ , todo  $r \in s(w, j)$  e todo  $W_\alpha \subseteq V' \setminus \{w\}$ , com  $|W_\alpha| = \alpha$ .

Sejam  $u, v \in V' \setminus \{w\}$ ,  $u \neq v$ , que existem porque  $|V'| \geq 3$ . Como  $\alpha \leq |V'| - 2$ , podemos tomar  $W_\alpha \subseteq V' \setminus \{w\}$  tal que  $u \in W_\alpha$ ,  $v \notin W_\alpha$  e  $|W_\alpha| = \alpha$ . Seja  $W'_\alpha = W_\alpha \setminus \{u\} \cup \{v\}$ . Como  $\lambda_{wj r} = -\sum_{v \in W_\alpha} \pi_{wv}$  e  $\lambda_{wj r} = -\sum_{v \in W'_\alpha} \pi_{wv}$ , concluímos que  $\pi_{wu} = \pi_{wv}$ . Ou seja,  $\pi_{wu} = \pi_{wv} \forall w \in V', \forall u, v \in V' \setminus \{w\}$ .

A partir disso, deduzimos todas as relações que ainda faltam entre as componentes não nulas de  $\pi$  e  $\lambda$ . De fato, sejam  $(w, u)$  e  $(w', v)$  dois pares de elementos de  $V'$ , onde  $u, v, w, w'$  são todos distintos. Mostramos que  $\pi_{wu} = \pi_{w'u} = \pi_{w'v}$ . Logo, existe uma constante  $a$  tal que  $\pi_{wv} = -a \forall w, v \in V', v \neq w$  e, consequentemente,  $\lambda_{wj r} = -\sum_{v \in W_\alpha} \pi_{wv} = \alpha a \forall w \in V', \forall r \in s(w, j)$ .  $\square$

### 5.2.3 Desigualdades válidas

Apresentamos aqui algumas classes de desigualdades válidas que podem fortalecer a Formulação 1. Sejam  $S' \subseteq S$  e  $T' \subseteq T$ . Para  $j \in T'$ , dizemos que um par  $u, v \in V$  é  $(j, S')$ -compatível se as seguintes condições são satisfeitas:

- $s(u, j) \cap S' \neq \emptyset$  e  $s(v, j) \cap S' \neq \emptyset$ ;
- se  $s(u, j) \cap S' = s(v, j) \cap S' = \{s\}$ , então  $t(j, s) \geq 2$ .

Dizemos que  $u, v$  é  $(T', S')$ -compatível se  $u, v$  é  $(j, S')$ -compatível para algum  $j \in T'$ . Denotamos por  $C(u, T', S')$  o conjunto dos vértices  $v$  tal que  $u, v$  é  $(T', S')$ -compatível.

Defina

$$\Theta(u, T', S') := \sum_{v \in C(u, T', S')} Z_{uv} - \sum_{j \in T'} \sum_{s \in s(u, j) \cap S'} (n(j, S') - 1) X_{uj_s}$$

**Proposição 5.18.** *Para todos  $u \in V(G)$ ,  $T' \subseteq T$  e  $S' \subseteq S$ ,  $\Theta(u, T', S') \geq 0$  é válida para  $P_1(FEST)$ .*

*Demonstração.* Seja  $(X, Z)$  viável para  $FEST$ . Se  $\sum_{j \in T'} \sum_{s \in s(u, j) \cap S'} X_{uj_s} = 0$ ,  $\Theta(u, T', S') \geq 0$  é claramente satisfeita pela não-negatividade de  $Z$ . Do contrário, nessa solução,  $u$  está alocado em uma equipe  $j \in T'$  com habilidade  $s \in s(u, j) \cap S'$ . Neste caso,  $\Theta(u, T', S') \geq 0$  reduz-se a  $\sum_{v \in C(u, T', S')} Z_{uv} \geq n(j, S') - 1$ . Se  $s(u, j) \cap S' = \{s\}$  e  $t(j, s) = 1$ , defina  $\hat{S} = S' \setminus \{s\}$ , de modo que  $n(j, \hat{S}) = n(j, S') - 1$ . Caso contrário, defina  $\hat{S} = S'$ . Então,  $\Theta(u, T', S') \geq 0$  torna-se

$$\sum_{v \in C(u, T', S')} Z_{uv} \geq \begin{cases} n(j, \hat{S}) - 1, & \text{se } s \in \hat{S}, \\ n(j, \hat{S}), & \text{se } s \notin \hat{S}. \end{cases}$$

Seja  $\hat{V} = \{v \in V \setminus \{u\} : \sum_{r \in s(v, j) \cap \hat{S}} X_{vjr} = 1\}$  o conjunto de indivíduos em  $V \setminus \{u\}$  que, na solução  $(X, Z)$ , estão na equipe  $j$  desempenhando alguma das habilidades em  $\hat{S}$ . Logo,  $|\hat{V}| \geq n(j, \hat{S})$ , se  $s \notin \hat{S}$ , e  $|\hat{V}| \geq n(j, \hat{S}) - 1$ , se  $s \in \hat{S}$ . Como  $X_{uj_s} = 1$ , por (5.4), temos que  $Z_{uv} = 1 \forall v \in \hat{V}$ . Logo é suficiente mostrar que  $C(u, T', S') \supseteq \hat{V}$ .

Tome  $v \in \hat{V}$ . Claramente,  $s(v, j) \cap S' \supseteq s(v, j) \cap \hat{S} \neq \emptyset$  e  $s \in s(u, j) \cap S'$ . Além disso, se  $s(u, j) \cap S' = s(v, j) \cap S'$  é um conjunto unitário, então este conjunto é  $\{s\}$  e  $s \in \hat{S}$ . Pela definição de  $\hat{S}$ , deve ocorrer  $t(j, s) \geq 2$ . Concluímos que  $v \in C(u, T', S')$  e, portanto,  $\hat{V} \subseteq C(u, T', S') \cap V(S')$ .  $\square$

Outra classe de desigualdades, semelhante à anterior, mas que considera compatibilidades em todo o conjunto  $S$  e não apenas restritas às habilidades em  $S'$  ou, mais precisamente,  $C(u, T', S)$  em lugar de  $C(u, T', S')$ , é dada a seguir. Para  $u \in V$ ,  $T' \subseteq T$  e  $S' \subseteq S$ , seja

$$\Phi(u, T', S') := \sum_{\substack{v \in C(u, T', S) \\ v \in V(S')}} Z_{uv} - \sum_{j \in T'} \sum_{s \in s(u, j) \cap S'} (n(j, S') - 1) X_{uj_s} - \sum_{j \in T'} \sum_{s \in s(u, j) \setminus S'} n(j, S') X_{uj_s}$$

**Proposição 5.19.** *Para todos  $u \in V(G)$ ,  $T' \subseteq T$  e  $S' \subseteq S$ ,  $\Phi(u, T', S') \geq 0$  é válida para  $P_1(FEST)$ .*

*Demonstração.* Seja  $(X, Z)$  viável para  $FEST$ . Se nessa solução  $u$  não está alocado ou pertence a uma equipe  $j \notin T'$ ,  $\Phi(u, T', S') \geq 0$  é claramente satisfeita porque  $Z \geq \mathbf{0}$ . Suponha agora que  $u$  está alocado em uma equipe  $j \in T'$  com uma habilidade  $s \in s(u, j)$ . Então,  $\Phi(u, T', S') \geq 0$  reduz-se a

$$\sum_{v \in C(u, T', S) \cap V(S')} Z_{uv} \geq \begin{cases} n(j, S') - 1, & \text{se } s \in S', \\ n(j, S'), & \text{se } s \notin S'. \end{cases}$$

Se  $s \in S'$ ,  $s(u, j) = \{s\}$  e  $t(j, s) = 1$ , defina  $\hat{S} = S' \setminus \{s\}$ , de modo que  $n(j, \hat{S}) = n(j, S') - 1$ . Caso contrário, defina  $\hat{S} = S'$ . Então,  $\Phi(u, T', S') \geq 0$  torna-se

$$\sum_{v \in C(u, T', S) \cap V(S')} Z_{uv} \geq \begin{cases} n(j, \hat{S}) - 1, & \text{se } s \in \hat{S}, \\ n(j, \hat{S}), & \text{se } s \notin \hat{S}. \end{cases}$$

Seja  $\hat{V} = \{v \in V \setminus \{u\} : \sum_{r \in s(v, j) \cap \hat{S}} X_{vjr} = 1\}$  o conjunto de indivíduos em  $V \setminus \{u\}$  que, na solução  $(X, Z)$ , estão na equipe  $j$  desempenhando alguma das habilidades em  $\hat{S}$ . Logo,  $|\hat{V}| \geq n(j, \hat{S})$ , se  $s \notin \hat{S}$ , e  $|\hat{V}| \geq n(j, \hat{S}) - 1$ , se  $s \in \hat{S}$ . Lembre que  $X_{ujs} = 1$ . Então, por (5.4), temos que  $Z_{uv} = 1 \forall v \in \hat{V}$ . Logo é suficiente mostrar que  $C(u, T', S) \cap V(S') \supseteq \hat{V}$ .

Tome  $v \in \hat{V}$ . Claramente,  $v \in V(\hat{S}) \subseteq V(S')$ . Além disso,  $s(v, j) \cap \hat{S} \neq \emptyset$  e  $s \in s(u, j)$ . Então, se  $s(u, j) = s(v, j)$  é um conjunto unitário, devemos ter  $s(u, j) = s(v, j) = \{s\}$  e, por conseguinte,  $s \in \hat{S} \subseteq S'$ . Pela definição de  $\hat{S}$ , deve ocorrer  $t(j, s) \geq 2$ . Concluimos que  $v \in C(u, T', S)$ , implicando  $\hat{V} \subseteq C(u, T', S) \cap V(S')$ .  $\square$

Alguns casos particulares da desigualdade  $\Phi(u, T', S') \geq 0$  que usaremos para fortalecer a Formulação 1 ocorrem quando  $(T', S') = (T, S)$  ou  $(T', S') = (T, \{s\})$  ou  $(T', S') = (\{j\}, \{s\})$ . As expressões ficam respectivamente:

$$\Phi(u, T, S) = \sum_{v \in C(u, T, S)} Z_{uv} - \sum_{j \in T} \sum_{s \in s(u, j)} (n(j, S) - 1) X_{ujs}$$

$$\Phi(u, T, s) = \sum_{\substack{v \in C(u, T, S) \\ v \in V(s)}} Z_{uv} - \sum_{j \in T: s \in s(u, j)} (t(j, s) - 1) X_{ujs} - \sum_{j \in T} \sum_{s' \in s(u, j) \setminus \{s\}} t(j, s) X_{ujs'}$$

$$\Phi(u, j, s) = \sum_{\substack{v \in C(u, j, S) \\ v \in V(s)}} Z_{uv} - (t(j, s) - 1) X_{ujs} - \sum_{s' \in s(u, j) \setminus \{s\}} t(j, s) X_{ujs'}$$

Note que  $\Phi(u, T, S) = \Theta(u, T, S)$ .

### 5.3 Formulação 2

Dada um instância  $I_{FEST} = (G, d, S, T, s, t)$ , como definida na Seção 4.1, considere as seguintes variáveis. Para cada vértice  $u \in V(G)$ , para cada subgrafo  $j \in T$ , introduzimos uma variável binária  $Y_{uj}$ , relacionando um vértice  $u$  a um subgrafo  $j$ , onde:

$$Y_{uj} = \begin{cases} 1, & \text{se } u \text{ está no subgrafo } j, \\ 0, & \text{caso contrário,} \end{cases} \quad \forall u \in V(G), \forall j \in T : s(u, j) \neq \emptyset.$$

Para cada  $uv \in E(G)$ , definimos uma variável binária  $Z_{uv}$  relacionada às arestas, onde:

$$Z_{uv} = \begin{cases} 1, & \text{se } u \text{ e } v \text{ estão no mesmo subgrafo,} \\ 0, & \text{caso contrário,} \end{cases} \quad \forall uv \in E(G).$$

Uma outra formulação para o *FEST* é então:

$$\min \sum_{uv \in E(G)} d(uv) Z_{uv} = d(Y, Z) \quad (5.11)$$

$$s.a. \quad Y_{uj} + Y_{vj} - Z_{uv} \leq 1, \quad \forall j \in T, \forall u, v \in V(G): s(u, j) \neq \emptyset, s(v, j) \neq \emptyset, \quad (5.12)$$

$$\sum_{j \in T} Y_{uj} \leq 1, \quad \forall u \in V(G). \quad (5.13)$$

$$\sum_{u \in V(S')} Y_{uj} \geq \sum_{s \in S'} t(j, s), \quad \forall j \in T, S' \subseteq s(j). \quad (5.14)$$

$$Y_{uj} \in \{0, 1\}, \quad \forall u \in V(G), \forall j \in T : s(u, j) \neq \emptyset, \quad (5.15)$$

$$Z_{uv} \in \{0, 1\}, \quad \forall uv \in E(G). \quad (5.16)$$

As formulações 1 e 2 estão relacionadas através da seguinte transformação de variáveis:

$$Y_{uj} = \sum_{s \in s(u, j)} X_{uj s} \quad \forall u \in V, \forall j \in T. \quad (5.17)$$

**Proposição 5.20.** *Se  $(X, Z)$  é uma solução viável para a relaxação linear de (5.4)–(5.8), então  $(Y, Z)$  é uma solução viável para a relaxação linear de (5.12)–(5.16), onde  $Y$  é dado por (5.17). Ambas as soluções têm o mesmo valor.*

*Demonstração.* Seja  $(X, Z)$  satisfazendo (5.4)–(5.6),  $0 \leq X \leq \mathbf{1}$  e  $0 \leq Z \leq \mathbf{1}$ . Então, por (5.17),  $(Y, Z)$  claramente satisfaz (5.12)–(5.13),  $0 \leq Y \leq \mathbf{1}$  e  $0 \leq Z \leq \mathbf{1}$ . Para mostrar que  $(Y, Z)$  satisfaz (5.14), tome  $j \in T$  e  $S' \subseteq s(j)$  e veja que

$$\sum_{u \in V(S')} Y_{uj} = \sum_{\substack{u \in V: \\ s(u, j) \cap S' \neq \emptyset}} \sum_{s \in s(u, j)} X_{uj s} \geq \sum_{\substack{u \in V: \\ s(u, j) \cap S' \neq \emptyset}} \sum_{s \in s(u, j) \cap S'} X_{uj s} = \sum_{u \in V} \sum_{s \in s(u, j) \cap S'} X_{uj s}.$$

Por outro lado, somando as inequações (5.6) para  $s \in S'$  obtemos

$$\sum_{s \in S'} \sum_{u: s \in s(u, j)} X_{uj s} = \sum_{u \in V} \sum_{s \in s(u, j) \cap S'} X_{uj s} \geq \sum_{s \in S'} t(j, s)$$

As duas expressões destacadas resultam em (5.14). □

Similarmente à Formulação 1, podemos definir a solução pura  $(Y, Z^0)$  e uma solução reduzida  $(Y^0, Z)$  relativas a uma solução  $(Y, Z)$  viável para a Formulação 2.

Usando a relação (5.17), podemos derivar resultados para a Formulação 2 similares àqueles obtidos na Seção 5.1 para a Formulação 1. Apresentamos estes resultados aqui por completude, mas omitimos algumas provas que são adaptações diretas das anteriores.

**Proposição 5.21.** *Seja  $I_{FEST}$  uma instância de *FEST*. Toda solução viável para Formulação 2 de valor  $w$  define uma solução viável para  $I_{FEST}$  de valor menor ou igual a  $w$ . Toda solução viável para  $I_{FEST}$  pode ser mapeada em uma solução viável de igual valor para a Formulação 2.*

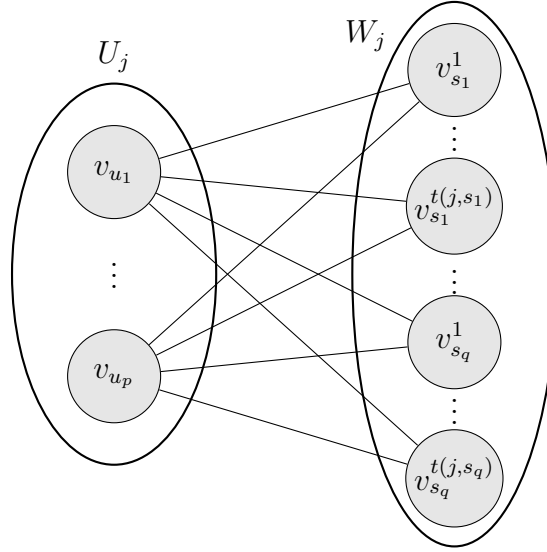


Figura 4 – Grafo bipartido  $G_j = (U_j \cup W_j, E_j)$ , onde  $p = |U_j|$  e  $q = |W_j|$ .

*Demonstração.* Seja  $(Y, Z)$  uma solução viável para (5.11)–(5.16) e  $(Y, Z^0)$  sua purificação. Temos que  $d(Y, Z^0) \leq d(Y, Z)$ . A solução para  $I_{FEST}$ , com valor  $d(Y, Z^0)$ , é obtida alocando um indivíduo  $u$  na equipe  $j$  se, e somente se,  $Y_{uj} = 1$ . Seguindo a demonstração da Proposição 5.5, resta-nos mostrar que é possível distribuir os indivíduos de cada equipe  $j$  para atender as demandas  $t(j, s) \forall s \in s(j)$ . Para tal, seja  $U_j = \{v_u : Y_{uj} = 1\}$  e construa o grafo bipartido  $G_j = (U_j \cup W_j, E_j)$ , onde  $W_j = \{v_s^k : s \in s(j), k = 1, 2, \dots, t(j, s)\}$  e  $E_j = \{v_u v_s^k : s \in s(u, j), k = 1, 2, \dots, t(s, j)\}$ , ou seja,  $W_j$  tem  $t(s, j)$  cópias de um vértice associado a  $s \in s(j)$ , e cada aresta em  $E_j$  marca uma possível alocação entre um indivíduo  $u$  e uma habilidade  $s$  que ele pode desempenhar. A Figura 4 ilustra a construção do grafo bipartido  $G_j = (U_j \cup W_j, E_j)$  quando  $s(u) = s(j) \forall u \in U_j$ ,  $p = |U_j|$  e  $q = |W_j|$ .

Note que um emparelhamento que cobre  $W_j$  é exatamente uma distribuição possível dos indivíduos da equipe  $j$ , respeitando as demandas de habilidades desta equipe. E a restrição (5.14) é equivalente à condição de Hall, que garante a existência de tal emparelhamento.

Considere agora uma solução viável para  $I_{FEST}$ . Pela Proposição 5.5, ela define uma solução  $(X, Z)$  viável para a Formulação, de mesmo valor 1. Pela Proposição 5.20, podemos obter uma solução  $(Y, Z)$  para a Formulação 2 de igual valor.  $\square$

**Corolário 5.22.** *Seja  $I_{FEST}$  uma instância de  $FEST$ . Toda solução ótima obtida pela Formulação 2 define uma solução ótima para  $I_{FEST}$ , e toda solução ótima para  $I_{FEST}$  pode ser mapeada em uma solução ótima para a Formulação 2.*

### 5.3.1 Algoritmo de separação

Observe que existe um número exponencial, no tamanho de  $S$ , de restrições do tipo (5.14) na Formulação 2. Em casos como este, separar tais restrições da formulação pode

ser um alternativa interessante. Removemos as restrições (5.14) da formulação original, resolvemos o problema relaxado assim obtido e verificamos se sua solução viola alguma das restrições excluídas ou satisfaz todas elas. No primeiro caso, incluímos na formulação as restrições violadas identificadas e repetimos o processo. No segundo, temos a solução ótima. A etapa para identificar restrições violadas chama-se *separação* e o processo como um todo define o *método de planos de corte*.

Em nosso caso, o problema de separação para uma solução  $(Y, Z)$ ,  $Y \geq \mathbf{0}$ , corresponde a  $|T|$  problemas de fluxo máximo, como detalhamos a seguir. Para cada  $j \in T$ , construímos uma rede denotada por  $G_j(Y, Z)$  da seguinte forma:

- criamos um vértice de origem denotado por  $o$ .
- para cada  $u \in U(j) := \{u \in V(G) : Y_{uj} > 0\}$ , criamos um vértice  $v_u$  e incluímos um arco com capacidade  $Y_{uj}$  de  $o$  para  $v_u$ ;
- para cada habilidade  $s \in s(j)$ , criamos um vértice denotado por  $v_s$  e incluímos arcos com capacidade infinita de  $v_u$  para  $v_s$ , se somente se,  $s \in s(u, j)$ ;
- por fim, criamos um vértice de destino, denotado por  $d$ , e entre cada vértice  $v_s$  e  $d$ , incluímos um arco com capacidade  $t(j, s)$ .

Formalmente,

$$V(G_j(Y, Z)) = \{o, d\} \cup \{v_u : u \in U(j)\} \cup \{v_s : s \in s(j)\}.$$

$$E(G_j(Y, Z)) = \{(o, v_u) : u \in U(j)\} \cup \{(v_s, d) : s \in s(j)\} \cup \{(v_u, v_s) : s \in s(u, j)\}.$$

A Figura 5 apresenta a construção da rede  $G_j(Y, Z)$  para  $s(u) = S(j) \forall u \in U(j)$ .

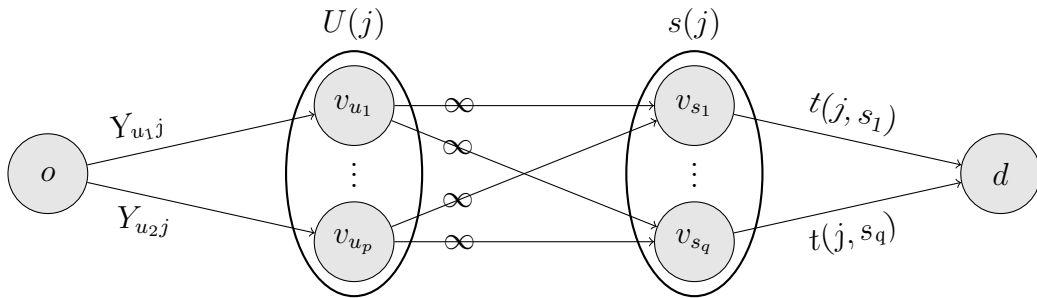


Figura 5 – Rede  $G_j(Y, Z)$ , onde  $p = |U(j)|$  e  $q = |s(j)|$ .

**Proposição 5.23.** *Seja  $(Y, Z) \in \mathbb{R}^{|V| \times |T|} \times \mathbb{R}^{|E|}$ . Sejam  $j \in T$  e  $f$  um fluxo máximo em  $G_j(Y, Z)$ . Se  $|f| = \sum_{s \in s(j)} t(j, s)$ , então todas as restrições (5.14) relativas a  $j \in T$  são satisfeitas por  $(Y, Z)$ . Caso contrário, podemos encontrar um conjunto  $S' \subseteq s(j)$  que viola uma restrição do tipo (5.14) em  $(Y, Z)$ .*

*Demonstração.* Pelas restrições de capacidade e conservação de fluxo, temos que:

$$Y_{uj} \geq f_{ov_u} = \sum_{s \in s(u,j)} f_{v_u v_s} \quad \forall u \in U(j) \quad (5.18)$$

$$t(j, s) \geq f_{v_s d} = \sum_{u \in U(j): s \in s(u,j)} f_{v_u v_s}, \quad \forall s \in s(j) \quad (5.19)$$

Primeiro suponha que  $|f| = \sum_{s \in s(j)} t(j, s)$ . Como  $|f| = \sum_{s \in s(j)} f_{v_s d}$ , então deve ocorrer igualdade em (5.19), ou seja,

$$\sum_{u \in U(j): s \in s(u,j)} f_{v_u v_s} = t(j, s), \quad \forall s \in s(j) \quad (5.20)$$

Tome  $S' \subseteq s(j)$ . Somando as desigualdades em (5.18) para  $u \in U(j) \cap V(S')$ , obtemos

$$\sum_{u \in U(j) \cap V(S')} Y_{uj} \geq \sum_{u \in U(j) \cap V(S')} \sum_{s \in s(u,j)} f_{v_u v_s} \quad (5.21)$$

Note que o lado direito da expressão acima vale

$$\sum_{u \in U(j): s(u) \cap S' \neq \emptyset} \sum_{s \in s(u,j)} f_{v_u v_s} \geq \sum_{s \in S'} \sum_{u \in U(j): s \in s(u,j)} f_{v_u v_s} \quad (5.22)$$

Por (5.21), (5.22) e (5.20) chegamos a:

$$\sum_{u \in U(j) \cap V(S')} Y_{uj} = \sum_{u \in V(S')} Y_{uj} \geq \sum_{s \in S'} t(j, s)$$

Portanto todas as restrições (5.14) relativas a  $j$  são satisfeitas por  $(Y, Z)$ .

Suponha agora que  $|f| < \sum_{s \in s(j)} t(j, s)$ . Dados  $V', V''$  subconjuntos disjuntos de vértices de  $G_j(Y, Z)$ , seja  $f(V', V'') = \sum_{(i,j) \in E(G_j): i \in V', j \in V''} f_{ij}$  o fluxo que passa de  $V'$  para  $V''$ . Seja  $[C_o, C_d]$  o corte de arestas mínimo associado a  $f$ , tal que  $o \in C_o$  e  $d \in C_d$ . Defina

$$\begin{aligned} U_o &= \{v_u \in C_o : u \in U(j)\}, & U_d &= \{v_u \in C_d : u \in U(j)\} \\ S_o &= \{v_s \in C_o : s \in s(j)\}, & S_d &= \{v_s \in C_d : s \in s(j)\}. \end{aligned}$$

Ver Figura 6. Como os arcos  $v_u v_s$  têm capacidade infinita, não pode haver arco na rede de  $U_o$  para  $S_d$ ; do contrário haveria um arco residual entre  $U_o$  e  $S_d$ . Sendo assim

$$[C_o, C_d] = \{(o, v_u) : u \in U_d\} \cup \{(v_s, d) : s \in S_o\}.$$

Também não pode haver fluxo passando de um nó  $v_u \in U_d$  para um nó  $v_s \in S_o$ ; do contrário haveria capacidade residual no arco reverso, que ligaria  $S_o$  a  $U_d$ . Dessas duas observações, deduzimos que

$$f(U_o, S_d) = f(U_d, S_o) = 0.$$

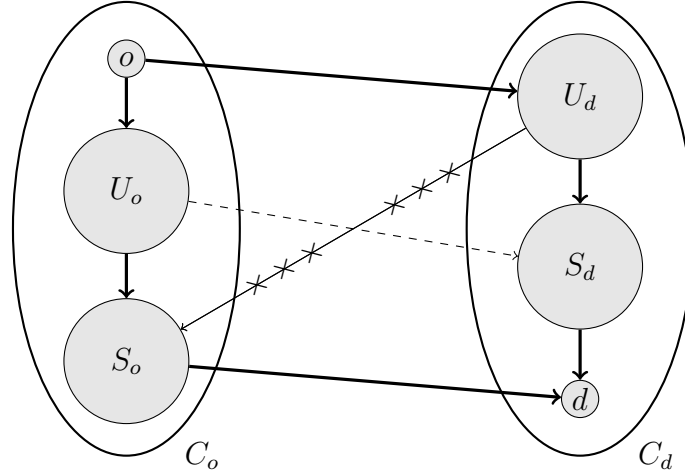


Figura 6 – Ilustração do corte (— — — arco inexistente; —×××— arco sem fluxo)

Além disso, pela conservação de fluxo, ocorre que

$$f(o, U_d) = f(U_d, S_o) + f(U_d, S_d) = f(U_d, S_d) \quad (5.23)$$

$$f(S_d, d) = f(U_o, S_d) + f(U_d, S_d) = f(U_d, S_d) \quad (5.24)$$

$$|f| = f(S_d, d) + f(S_o, d) \quad (5.25)$$

o que nos leva a

$$f(o, U_d) = |f| - f(S_o, d).$$

Novamente pela capacidade infinita dos arcos  $v_u v_s$ ,  $f$  deve saturar a capacidade dos arcos do corte, ou seja,

$$f_{ov_u} = Y_{uj} \quad \forall v_u \in U_d \quad e \quad f_{v_s d} = t(j, s) \quad \forall v_s \in S_o.$$

Substituindo essas igualdades na expressão acima, obtemos

$$\sum_{v_u \in U_d} Y_{uj} = |f| - \sum_{v_s \in S_o} t(j, s).$$

Pela hipótese, temos que  $|f| - \sum_{v_s \in S_o} t(j, s) < \sum_{v_s \in S_d} t(j, s)$ . Note ainda que  $S_d \neq \emptyset$ , pois do contrário teríamos  $C_d = \{d\}$ , contrariando a hipótese. E como não há arcos de  $U_o$  a  $S_d$ , temos que  $U_d$  engloba todos os indivíduos com alguma habilidade induzida por  $S_d$ , ou seja,  $U_d = \{v_u : u \in U(j), s(u) \cap S' \neq \emptyset\}$ , onde  $S' = \{s : v_s \in S_d\}$ . Logo, temos a violação

$$\sum_{u: s(u) \cap S' \neq \emptyset} Y_{uj} < \sum_{s \in S'} t(j, s).$$

□

### 5.3.2 Desigualdades válidas

Na Proposição 5.20, mostramos como converter uma solução viável para Formulação 1 em uma solução viável para a Formulação 2. Começamos essa seção mostrando como tal



transformação pode ser feita no sentido inverso. Tal resultado é útil para obter desigualdades válidas para a Formulação 2 a partir de desigualdades válidas para a Formulação 1.

A conversão de uma solução viável  $(Y, Z)$  para a Formulação 2 em uma solução viável  $(X, Z)$  para a Formulação 1 passa essencialmente pela distribuição, para cada  $j \in T$ , dos valores  $Y_{uj}$  que respeitam as desigualdades (5.14) em valores  $X_{ujs}$  que satisfaçam as restrições (5.6). Para mostrar que isso é possível, definimos, para cada  $j \in T$ , uma rede  $F_j(Y, Z)$  na qual procuramos um fluxo viável. Tal rede é assim construída:

- para cada  $u \in U(j) := \{u \in V(G) : Y_{uj} > 0\}$ , criamos um vértice de oferta  $v_u$  com oferta de fluxo  $b(v_u) = Y_{uj}$ ;
- para cada  $u \in U(j)$  e cada habilidade  $s \in s(u, j)$ , criamos um vértice de repasse, denotado por  $v_{us}$ , e incluímos arcos com capacidade infinita de  $v_u$  para  $v_{us}$ ;
- para cada habilidade  $s \in s(j)$ , criamos um vértice de demanda, denotado por  $v_s$ , com demanda  $b(v_s) = -t(j, s)$  e incluímos arcos com capacidade infinita de  $v_{us}$  para  $v_s$  se, e somente se,  $s \in s(u, j)$ ;
- por fim, criamos mais um vértice sumidouro, denotado por  $d$ , com capacidade  $b(d) = \sum_{u \in V} Y_{uj} - \sum_{s \in S} t(j, s)$  e, entre cada vértice  $v_{us}$  e  $d$ , incluímos um arco com capacidade infinita.

Formalmente,

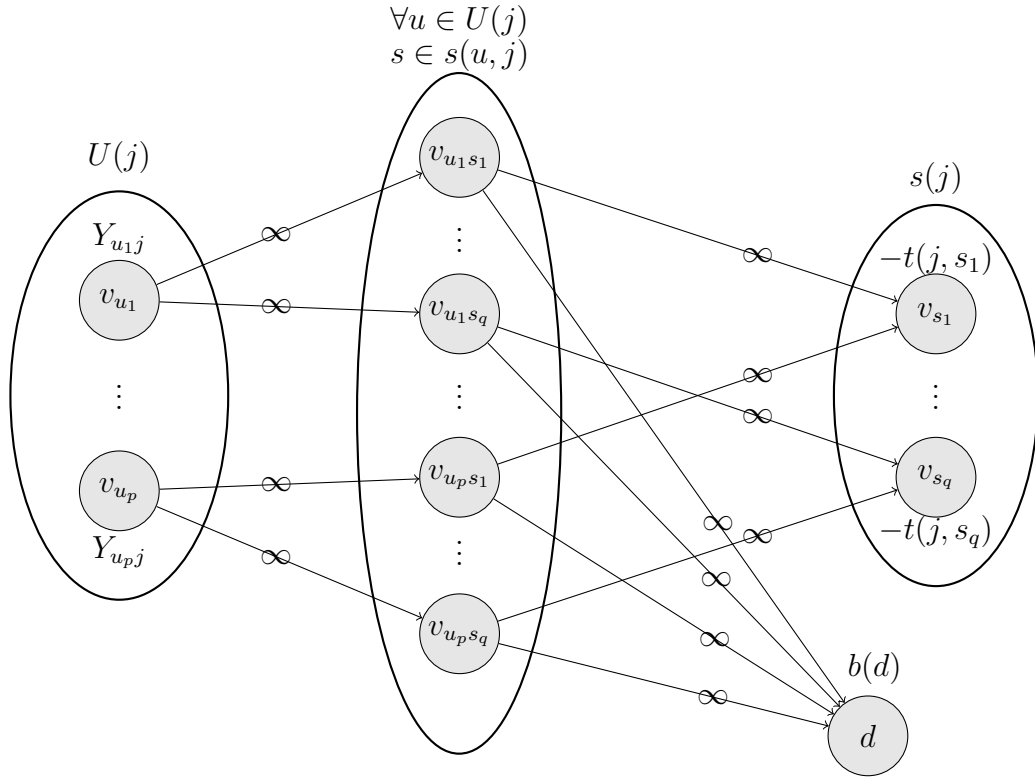
$$V(F_j(Y, Z)) = \{d\} \cup \{v_u : u \in U(j)\} \cup \{v_s : s \in s(j)\} \cup \{v_{us} : u \in U(j), s \in s(u, j)\}.$$

$$E(F_j(Y, Z)) = \{(v_u, v_{us}), (v_{us}, v_s), (v_{us}, d) : u \in U(j), s \in s(u, j)\}.$$

A Figura 5 apresenta a construção da rede  $F_j(Y, Z)$  para  $s(u) = s(j) \forall u \in U(j)$ .

**Proposição 5.24.** *Se  $(Y, Z)$  é uma solução viável para a relaxação linear de (5.12)–(5.16), então existe solução  $(X, Z)$  viável para a relaxação linear (5.4)–(5.8), onde  $X$  satisfaz as relações (5.17). Ambas as soluções têm o mesmo valor.*

*Demonstração.* Seja  $(Y, Z)$  satisfazendo (5.12)–(5.14),  $0 \leq Y \leq \mathbf{1}$  e  $0 \leq Z \leq \mathbf{1}$ . Primeiro mostramos que, para todo  $j \in T$ , existe fluxo viável em  $F_j(Y, Z)$ . Pelo Teorema 2.4, devemos mostrar que  $b(W)$  é inferior ou igual à capacidade do corte  $[W, \overline{W}]$  em  $F_j(Y, Z)$ , para todo  $\emptyset \neq W \subsetneq V(F_j(Y, Z))$ . Como as capacidades de todos os arcos são infinitas, é suficiente considerar  $W$  tal que  $[W, \overline{W}] = \emptyset$ . Seja então um subconjunto  $W$  com tal propriedade. Note que  $b(W) = -b(\overline{W})$ , pois  $b(V(F_j(Y, Z))) = 0$ . Então devemos mostrar que  $b(\overline{W}) \geq 0$ . Seja  $S' = \{s \in s(j) : v_s \in \overline{W}\}$ . Se  $S' = \emptyset$ , então claramente  $b(\overline{W}) \geq 0$ . Suponha agora que  $S' \neq \emptyset$ . Como  $[W, \overline{W}] = \emptyset$ , todo  $v_{us}$  tal que  $s \in S'$  deve pertencer a  $\overline{W}$ , e conseqüentemente, o mesmo ocorre com  $u \in V(S')$ . Então,  $b(\overline{W}) \geq \sum_{u \in V(S')} b(v_u) + \sum_{s \in S'} b(v_s) = \sum_{u \in V(S')} Y_{uj} - \sum_{s \in S'} t(j, s)$ . Logo, por (5.14),  $b(\overline{W}) \geq 0$ .


 Figura 7 – Rede  $F_j(Y, Z)$ , onde  $p = |U(j)|$  e  $q = |s(j)|$ .

Seja  $f^j$  um fluxo viável em  $F_j(Y, Z)$ . Defina

$$X_{ujs} = f_{v_u v_s}^j \quad \forall j \in T, \forall u \in V, \forall s \in s(u, j).$$

Então, a equação de conservação de fluxo em  $v_u$ , dada por  $\sum_{s \in s(u, j)} f_{v_u v_s}^j = b(v_u)$ , torna-se  $\sum_{s \in s(u, j)} X_{ujs} = Y_{uj}$ . Portanto, temos as relações (5.17) satisfeitas. Com isso, o cumprimento às restrições (5.12)-(5.13) leva diretamente ao mesmo resultado para as restrições (5.4)-(5.5). Além disso, temos claramente  $0 \leq X \leq \mathbf{1}$  e  $0 \leq Z \leq \mathbf{1}$ . Por outro lado, a conservação de fluxo em  $v_{us}$  e em  $v_s$  fornece  $X_{ujs} = f_{v_{us} v_s}^j + f_{v_{us} d}^j$  e  $\sum_{u \in V(s)} f_{v_{us} v_s}^j = t(j, s)$ , respectivamente. Logo,  $\sum_{u \in V(s)} X_{ujs} \geq t(j, s)$ , mostrando que  $X$  satisfaz (5.6).  $\square$

A transformação dada pela Proposição 5.24 proporciona uma conversão de desigualdades válidas da Formulação 1 em desigualdades válidas para a Formulação 2.

**Proposição 5.25.** *Seja  $\lambda^T X + \pi^T Z \leq \theta$  uma desigualdade válida para a Formulação 1. Se, para todos  $u \in V$  e  $j \in T$ ,  $\lambda_{ujs} = \mu_{uj}$ , para todo  $s \in s(u, j)$ , então  $\sum_{u \in V} \sum_{j \in T: s(u, j) \neq \emptyset} \mu_{uj} Y_{uj} + \pi^T Z \leq \theta$  é válida para a Formulação 2.*

*Demonstração.* Seja  $(Y, Z)$  uma solução viável de (5.12)–(5.16). Seja  $(X, Z)$  a solução viável correspondente de (5.4)–(5.8) tal que  $\sum_{s \in s(u, j)} X_{ujs} = Y_{uj}$ . Então,

$$\begin{aligned} \sum_{u \in V} \sum_{j \in T: s(u, j) \neq \emptyset} \mu_{uj} Y_{uj} + \pi^T Z &= \sum_{u \in V} \sum_{j \in T: s(u, j) \neq \emptyset} \mu_{uj} \sum_{s \in s(u, j)} X_{ujs} + \pi^T Z \\ &= \sum_{u \in V} \sum_{j \in T} \sum_{s \in s(u, j)} \lambda_{ujs} X_{ujs} + \pi^T Z \leq \theta. \end{aligned}$$

□

A partir das proposições 5.18 e 5.25, obtemos:

**Corolário 5.26.** *Para todos  $u \in V(G)$ ,  $T' \subseteq T$  e  $S' \subseteq S$ , a seguinte desigualdade é válida para a Formulação 2:*

$$\sum_{v \in C(u, T', S')} Z_{uv} - \sum_{\substack{j \in T' \\ s(u, j) \cap S' \neq \emptyset}} (n(j, S') - 1) Y_{uj} \geq 0$$

## 5.4 Comparação entre as formulações

As duas formulações apresentadas modelam corretamente o problema e qualquer delas poderia ser usada, em princípio, para resolvê-lo. Alguns aspectos práticos porém merecem ser observados.

Embora definidas sob a mesma estratégia, de determinar a alocação dos indivíduos nas equipes, a primeira estabelece também a habilidade que cada indivíduo vai desempenhar. Esse detalhamento é feito às custas de um maior número de variáveis. Note que temos  $|V| \times |T| \times |S| + |E|$  variáveis na Formulação 1, enquanto esse número cai para  $|V| \times |T| + |E|$  variáveis na Formulação 2. Por outro lado, o número de restrições dessa segunda formulação é potencialmente exponencial, já que há uma restrição em (5.14) para cada subconjunto de  $S$  e cada  $j \in T$ . Com isso o modelo completo pode tornar-se intratável por um **solver**. Uma alternativa é separar iterativamente essas restrições e usar um algoritmo de planos de corte. Como visto na Subseção 5.3.1, tal separação pode ser feita em tempo polinomial. A prevalência de uma formulação sobre a outra não é, portanto, clara.

Queremos registrar ainda que o acréscimo de restrições para fortalecer ambas as formulações pode se tornar mandatário. Essa é a perspectiva que podemos ter ao verificar que as relaxações lineares podem ser muito fracas em alguns casos.

Na verdade, uma consequência direta das proposições 5.20 e 5.24 é:

**Proposição 5.27.** *As relaxações lineares das formulações 1 e 2 possuem o mesmo valor ótimo.*

O exemplo ilustrativo abaixo mostra que o limite inferior fornecido pelas relaxações lineares pode estar, de fato, muito distante do valor ótimo inteiro.

Considere uma instância de  $I_{FEEST}$  onde há quatro indivíduos com idêntico grau de relacionamento entre si, todos com duas habilidades, a serem distribuídos em duas equipes, cada um demandando um indivíduo de cada habilidade. Formalmente, temos:

- $V(G) = \{a, b, c, d\}$ ,  $d : E(G) \rightarrow \mathbb{N}$  constante, igual a  $M$ .
- $S = \{r, r'\}$ , sendo  $s(a) = s(b) = s(c) = s(d) = S$ .

- $T = \{1, 2\}$ , sendo  $t(1, r) = t(1, r') = 1$  e  $t(2, r) = t(2, r') = 1$

Claramente, uma solução ótima possui valor  $2M$ . Por outro lado, fazendo  $X_{u1r} = 1/2$ , para  $u \in \{a, b\}$ ,  $X_{u2r'} = 1/2$ , para  $u \in \{c, d\}$ , e  $Z = 0$ , obtemos uma solução viável relaxada para a Formulação 1, de valor nulo. Evidentemente, fazendo  $Y_{u1} = 1/2$ , para  $u \in \{a, b\}$ ,  $Y_{u2} = 1/2$ , para  $u \in \{c, d\}$ , e  $Z = 0$ , obtemos uma solução viável relaxada para a Formulação 2, também de valor nulo. Note que há várias soluções ótimas relaxadas (de valor nulo) que são fracionárias.

No próximo capítulo, realizamos um estudo computacional das duas formulações, procurando comparar seus desempenhos na resolução de instâncias de *FEST*.

## 6 Resultados Computacionais

Neste capítulo, apresentamos quatro implementações baseadas nas formulações descritas nas seções 5.1 e 5.3, assim como uma avaliação comparativa realizada através de experimentos computacionais e a análise dos seus resultados. Na Seção 6.1 apresentamos o ambiente computacional utilizado para realização dos testes. Na Seção 6.2 mostramos as principais características das quatro implementações criadas e, na seção seguinte, as principais características das instâncias utilizadas na realização dos testes computacionais. Os resultados computacionais obtidos com as formulações e com a heurística proposta na Subseção 4.3.1, assim como uma análise estatística dos mesmos, são o objeto da última seção deste capítulo.

### 6.1 Configuração do ambiente computacional

A implementação dos algoritmos exatos deste trabalho faz uso do pacote de otimização ILOG CPLEX® versão 12.4. Neste pacote está presente a biblioteca denominada Callable Library, que permite o desenvolvimento de modelos matemáticos utilizando a linguagem de programação C. Os experimentos realizados foram executados utilizando uma máquina com processador de dois núcleos de 2.4GHz, 2.0GB de memória RAM e sistema operacional Linux de 32 bits.

### 6.2 Versões implementadas

Para realização dos testes computacionais com as formulações matemáticas propostas neste trabalho, foram desenvolvidas quatro diferentes implementações, duas baseadas na Formulação 1 e duas baseadas na Formulação 2. Cada uma destas versões é primeiro avaliada sem o conhecimento da solução fornecida pelo algoritmo heurístico proposto na Subseção 4.3.1. Depois analisamos a contribuição de tal informação para o desempenho das formulações.

Em todas elas, fazemos uso da Proposição 5.7 para fixar algumas variáveis e eliminar restrições. Transformando a instância de entrada de modo a obtermos  $\min\{d(uv) : uv \in E(G)\} = 0$ , fixamos  $Z_{uv} = 1$  sempre que  $d(uv) = 0$ . Note que sempre existe uma solução ótima satisfazendo essas fixações. Consequentemente, toda restrição em (5.4) referente a uma variável  $Z_{uv}$  fixada em 1 pode ser eliminada do modelo. Observamos experimentalmente que a adição de uma solução viável inicial ao modelo e o uso da simplificação proposta podem trazer significativos ganhos computacionais.

Denominamos *versão A1* e *versão A1 fortalecida* as duas implementações desenvolvidas como base na Formulação 1, apresentada na Seção 5.1. A primeira é exatamente a simplificação da Formulação 1 com a eliminação das variáveis e restrições descrita acima. Na segunda, adicionalmente a essa simplificação, incluímos algumas das desigualdades válidas descritas na Subseção 5.2.3, procurando fortalecer a versão A1. Especificamente, a versão A1 fortalecida inclui as desigualdades  $\Phi(u, T, S') \geq 0$ , para todo  $u \in V$  e para:

- $S' = S$ ;
- $S' = \{s\} \forall s \in S$ ;
- $S' = s(j) \forall j \in T$ ; e
- $S' = \{s \in S : n(T, \{s\}) \geq 0.2 n(T, S)\}$ .

Neste último caso,  $S'$  compreende as habilidades cuja demanda seja pelo menos 20% da demanda total de indivíduos. Lembre que no primeiro caso  $\Phi()$  e  $\Theta()$  coincidem.

Experiências com outras desigualdades das classes  $\Theta(u, T', S') \geq 0$  e  $\Phi(u, T', S') \geq 0$  também foram realizadas, mas não trouxeram melhorias significativas e não serão aqui consideradas na versão A2 fortalecida.

Denominamos por *versão B2* e *versão B2 fortalecida* as implementações baseadas na Formulação 2 apresentada na Seção 5.3. Em ambas, aplicamos um procedimento de planos de corte, já que a Formulação 2 apresenta um número exponencial de restrições em (5.16). Tal procedimento usa o algoritmo de separação que se origina da Proposição 5.23, ou seja, um algoritmo que resolve o problema do fluxo máximo em uma rede. Por simplicidade, denotaremos por Formulação 2' o modelo obtido da Formulação 2 com a remoção das restrições (5.16).

A versão B2 e a versão B2 fortalecida foram implementadas utilizando a Formulação 2', inicialmente com as restrições de integralidade relaxadas. Obtida uma solução ótima relaxada  $(Y, Z)$ , para cada equipe  $j \in T$  é construída a rede  $G_j(Y, Z)$ . Detalhes da construção das redes podem ser vistos na Subseção 5.3.1. Para cada  $j \in T$ , posteriormente à construção de cada rede  $G_j(X, Y)$ , é aplicado um algoritmo de fluxo máximo e verificado se o fluxo máximo encontrado é menor que a demanda de habilidades necessárias na equipe  $j$ . Em caso afirmativo, a restrição do tipo (5.16) referente à equipe  $j$  e ao conjuntos  $S'$  violado é adicionada à Formulação 2', e todo o processo de resolução do modelo é repetido. Caso a resposta seja negativa para toda equipe  $j$ , a solução ótima  $(Y, Z)$  encontrada para Formulação 2' também é solução ótima para o Formulação 2 com integralidade relaxada. Neste caso, se a solução ótima relaxada  $(Y, Z)$  for inteira, então também é ótima para a Formulação 2; senão, introduzimos as restrições de integralidade, e todo o processo de geração de restrições é repetido até que nenhuma restrição (5.16) esteja violada.

No intuito de melhorar o desempenho do algoritmo de planos de corte, implementamos e testamos quatro formas de inclusão de restrições do tipo (5.16). Dada uma solução  $(Y, Z)$  para a Formulação 2', a primeira forma de inclusão cria e verifica cada rede  $F_j(Y, Z)$

ordenadamente, somente até encontrar e adicionar a primeira restrição violada, caso exista alguma. A segunda forma verifica as  $|T|$  redes e adiciona todas as restrições violadas encontradas, dada a solução  $(Y, Z)$ , caso exista uma ou mais restrições desse tipo. Estas duas estratégias (e especialmente a segunda) tendem a tornar o modelo pesado após um certo número de iterações do processo.

As outras duas formas procuram exatamente manter estável o tamanho do modelo. Elas são semelhantes às duas primeiras, mas introduzem uma modificação: após a inserção de uma ou mais restrições violadas é realizada uma verificação, onde o valor da variável de folga de cada restrição do tipo (5.16) inserida em iterações anteriores é comparado a um dado valor  $\varepsilon$ . Caso uma variável de folga possua valor maior ou igual a  $\varepsilon$ , a restrição correspondente é removida do modelo.

Foram realizados testes computacionais com as quatro formas de inclusão de restrições apresentadas. O processo utilizando a quarta forma de inclusão com valor de  $\varepsilon = 0,5$  obteve melhor desempenho entre as todas as formas avaliadas, sendo então implementada na versão *B2* e versão *B2* fortalecida. Na verdade, estas versões diferem apenas do fato de que a fortalecida inclui a desigualdade apresentada no Corolário 5.26 para  $u \in V$ ,  $T' = T$  e  $S' = S$ . Lembre que essas desigualdades consistem na contraparte de  $\Theta(u, T, S) \geq 0$  para a Formulação 2.

## 6.3 Instâncias e suas principais características

As quatro implementações criadas foram avaliadas através de testes computacionais utilizando instâncias reais e instâncias fictícias do *FEST*, esta últimas obtidas por um gerador de instâncias pseudo-aleatório que desenvolvemos.

### 6.3.1 Instâncias reais

As instâncias reais foram obtidas através de um sistema de informação criado para auxiliar gestores na formação de equipes sociotécnicas. Através deste sistema, funcionários de organizações respondem a questionários sociométricos, que posteriormente são utilizados para geração do grafo de afinidades (sociograma). Analisando as respostas de um funcionário  $u$ , a afinidade de trabalho em equipe deste com um indivíduo  $v$  é categorizada em um de três níveis: S (sim) - boa afinidade; N (não) - sem afinidade; I - indiferente. Note que o nível de afinidade de  $u$  com relação a  $v$  pode não ser o mesmo daquele de  $v$  com respeito a  $u$ . As habilidades profissionais de cada funcionário também são cadastradas no sistema pelos gestores de cada organização. Após todas as informações serem cadastradas no sistema, as instâncias para o *FEST* são geradas. Mais detalhes podem ser obtidos em (FIGUEIREDO, 2011).

O sistema de informação apresentado foi empregado em duas organizações, tendo sido assim obtidas duas instâncias reais para o *FEST*, denominadas alfa e beta. A função  $d : E \rightarrow \mathbb{N}$  foi assim definida:

$u \rightarrow v$	$v \rightarrow u$	$d(uv)$
S	S	0
S	I	1
I	S	1
I	I	2

$u \rightarrow v$	$v \rightarrow u$	$d(uv)$
S	N	3
N	S	3
N	I	4
I	N	4
N	N	5

A organização alfa necessitava separar 34 indivíduos em sete equipes, e a organização beta necessitava separar 41 indivíduos em 6 equipes. As tabelas 1 e 2 contêm as características gerais das instâncias alfa e beta, onde:

- $|V|$ : quantidade de vértices existentes no grafo  $G$  (número de indivíduos participantes da organização);
- $|T|$ : quantidade de equipes a serem formadas;
- $|S|$ : número de habilidades demandadas nas equipes;
- qtd: quantidade de pesos diferentes existentes para  $E(G)$  (tipos de afinidades existentes entre um par de indivíduos);
- porcE(i): porcentagem de arestas com peso  $d(uv) = i \in \mathbb{N}$ , onde  $i \in \{0, 1, \dots, qtd - 1\}$ , existentes no grafo  $G$ .
- porcS(i): porcentagem de indivíduos com habilidade  $s_i \in S$ ,  $\forall i = \{1, 2, \dots, |S|\}$  existentes na organização.

Instância	$ V $	qtd	porcE	$ S $	porcS	$ T $
alfa.fest	34	6	{0.65, 0.12, 0.075, 0.045, 0.07, 0.04}	2	{0.50, 0.50}	7
beta.fest	41	6	{0.54, 0.04, 0.04, 0.14, 0.13, 0.11}	2	{0.50, 0.50}	6

Tabela 1 – Principais características de instâncias reais para *FEST*

$j$	1	2	3	4	5	6	7	$j$	1	2	3	4	5	6
$t(j, 1)$	5	5	5	4	4	4	3	$t(j, 1)$	6	6	6	6	6	7
$t(j, 2)$	0	0	0	1	1	1	1	$t(j, 2)$	0	1	1	1	1	0
$n(j, S)$	5	5	5	5	5	5	4	$n(j, S)$	6	7	7	7	7	7
alfa.fest								beta.fest						

Tabela 2 – Demandas das instâncias alfa e beta



### 6.3.2 Geração de instâncias aleatórias

Devido à quantidade reduzida de instâncias reais disponíveis para realização de testes, foi criado um conjunto de instâncias-teste através de um gerador de instâncias pseudo-aleatório. De maneira similar às instâncias reais existentes, foram geradas 120 instâncias para o *FEST*.

Os parâmetros usados para gerar uma instância  $I_{FEST} = (G, d, S, T, s, t)$  são:

- $|V(G)| \in \{20, 25, 30, 35, 40\}$ ;
- $|S| \in \{2, 5, 8\}$ ;
- $|T| \in \{2, 3, 4, 5, 6, 7, 8\}$ ; e
- $qtd$ ,  $probE$  e  $probS$ , que governam as funções  $d$ ,  $s$  e  $t$ .

O parâmetro  $qtd$  especifica a quantidade de níveis de afinidade entre os indivíduos. O valor  $probE(i)$ ,  $\forall i = 1, 2, \dots, qtd$ , representa a probabilidade de ser atribuído o  $i$ -ésimo nível de habilidade a uma aresta. Em outras palavras, a escolha do valor do peso  $d(uv) \in \{1, 2, \dots, qtd\}$  de cada aresta  $uv \in E(G)$  é definido através de um sorteio aleatório e  $d(uv) = i$  com probabilidade  $probE(i)$ . A Tabela 3 apresenta as probabilidades utilizadas neste trabalho.

Identificador	qtd	probE
arProb1	5	{0.2, 0.1, 0.1, 0.2, 0.4}
arProb2	5	{0.2, 0.3, 0.3, 0.1, 0.1}
arProb3	5	{0.1, 0.1, 0.2, 0.5, 0.1}
arProb4	5	{0.6, 0.25, 0.05, 0.05, 0.05}
arProb5	10	{0.1, 0.05, 0.025, 0.025, 0.05, 0.05, 0.075, 0.075, 0.15, 0.4}
arProb6	10	{0.15, 0.3, 0.2, 0.075, 0.075, 0.05, 0.05, 0.05, 0.025, 0.25}
arProb7	10	{0.1, 0.05, 0.05, 0.075, 0.3, 0.3, 0.05, 0.025, 0.025, 0.025}
arProb8	10	{0.5, 0.2, 0.1, 0.05, 0.025, 0.025, 0.025, 0.025, 0.025, 0.025}

Tabela 3 – Probabilidades para os pesos das arestas  $uv \in E(G)$

A geração da função  $s : V \rightarrow 2^S$  é feita de modo semelhante à definição dos pesos das arestas, porém agora governada pelo parâmetro  $probS(s)$ ,  $\forall s = 1, 2, \dots, |S|$ , que representa a probabilidade de um indivíduo possuir a habilidade  $s$ . Sendo assim, o conjunto  $s(u)$ , para todo  $u \in V(G)$ , também é definido através de um sorteio aleatório, onde  $s \in s(u)$  com probabilidade  $probS(s)$ . A Tabela 4 apresenta as probabilidades de ocorrência de habilidades utilizadas neste trabalho.

Por fim, a Tabela 5 detalha as principais características das demandas usadas para definir o conjunto de instâncias-teste.

Dados os parâmetros descritos acima para criação das instância-teste, a Tabela 6 e a Tabela 7 apresentam as principais características das 120 instâncias geradas.

Identificador	$ S $	probS
habProb1	2	{0.50, 0.50}
habProb2	2	{0.70, 0.50}
habProb3	2	{0.50, 0.65}
habProb4	2	{0.70, 0.85}
habProb5	5	{0.40, 0.40, 0.25, 0.25, 0.25}
habProb6	5	{0.20, 0.20, 0.30, 0.40, 0.40}
habProb7	5	{0.25, 0.25, 0.25, 0.25, 0.25}
habProb8	5	{0.35, 0.50, 0.40, 0.45, 0.35}
habProb9	8	{0.30, 0.30, 0.25, 0.25, 0.20, 0.20, 0.15, 0.15}
habProb10	8	{0.15, 0.15, 0.20, 0.20, 0.25, 0.25, 0.30, 0.30}
habProb11	8	{0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15}
habProb12	8	{0.40, 0.30, 0.25, 0.40, 0.30, 0.35, 0.25, 0.35}

Tabela 4 – Probabilidades de ocorrência de habilidades  $s \in S$ 

## 6.4 Estudo comparativo entre as formulações

Nesta seção apresentamos uma comparação realizada através de experimentos computacionais das implementações: versão  $A1$ , versão  $A1$  fortalecida, versão  $B2$  e versão  $B2$  fortalecida. O objetivo dessa comparação é, após uma análise estatística dos resultados obtidos, apresentar qual das implementações possui um melhor desempenho, considerando o tempo de execução de cada implementação para resolução exata do conjunto de instância-teste apresentadas na seção anterior.

### 6.4.1 Resolução do problema inteiro

A priori foram realizados testes com as implementações versão  $A1$ , versão  $A1$  fortalecida, versão  $B2$  e versão  $B2$  fortalecida para o problema inteiro. A Tabela 8 apresenta os dados dos resultados computacionais obtidos considerando apenas as instâncias em que todas as formulações obtiveram soluções ótimas inteiras em um limite máximo de tempo de execução de 5000 segundos.

Nesta tabela, estão presentes a coluna de identificação da instância solucionada (Inst), o tempo de execução (segundos) da versão  $A1$  ( $A1$ ), o tempo de execução (segundos) da versão  $A1$  fortalecida ( $A1_f$ ), o tempo de execução (segundos) da versão  $B2$  ( $B2$ ), o tempo de execução (segundos) da versão  $B2$  fortalecida ( $B2_f$ ), o valor da função objetivo do nó inicial da versão  $A1$  ( $R_{A1}$ ), o valor da função objetivo do nó inicial da versão  $A1$  fortalecida ( $R_{A1_f}$ ), o valor da função objetivo do nó inicial da versão  $B2$  ( $R_{B2}$ ), o valor da função objetivo do nó inicial da versão  $B2$  fortalecida ( $R_{B2_f}$ ), e o valor ótimo da função objetivo ( $OPT$ ). Os resultados apresentados estão ordenados de forma crescente pela média dos tempos de execução das quatro versões implementadas.

Identificador	$ V(G) $	$ S $	$ T $	$n(j,S)$	$t(j,S)$
demanda1	20	8	5	4	{ 0, 0, 1, 0, 0, 2, 0, 1 }
				4	{ 1, 1, 0, 1, 1, 0, 0, 0 }
				4	{ 0, 0, 0, 2, 1, 0, 1, 0 }
				3	{ 0, 1, 0, 2, 0, 0, 0, 0 }
				5	{ 0, 0, 0, 2, 0, 0, 3, 0 }
demanda2		5	2	10	{ 2, 2, 2, 3, 1 }
				10	
demanda3		2	4	5	{ 2, 3 }
				5	
				4	{ 3, 1 }
				6	{ 4, 2 }
demanda4	25	8	5	5	{ 0, 2, 2, 0, 0, 1, 0, 0 }
				4	{ 0, 1, 0, 0, 0, 0, 1, 2 }
				6	{ 2, 0, 0, 0, 0, 0, 1, 3 }
				4	{ 1, 0, 0, 1, 0, 0, 1, 1 }
				6	{ 2, 2, 0, 0, 1, 1, 0, 0 }
demanda5		5	4	4	{ 2, 1, 0, 1, 0 }
				8	{ 1, 2, 1, 2, 2 }
				5	{ 1, 1, 1, 1, 1 }
				7	{ 1, 1, 2, 1, 2 }
demanda6		2	3	8	{ 5, 3 }
				6	{ 3, 3 }
				10	{ 5, 5 }
demanda7	30	8	6	5	{ 2, 0, 0, 0, 0, 3, 0, 0 }
				5	{ 0, 0, 0, 2, 1, 0, 1, 1 }
				5	{ 2, 0, 0, 1, 1, 0, 0, 1 }
				5	{ 1, 1, 1, 0, 1, 0, 1, 0 }
				5	{ 0, 1, 0, 0, 1, 2, 0, 1 }
				5	{ 0, 0, 2, 1, 0, 0, 0, 2 }
demanda8		5	3	10	{ 2, 2, 2, 3, 1 }
				10	
				10	{ 2, 2, 2, 2, 2 }
demanda9		2	3	10	{ 5, 5 }
				6	{ 3, 3 }
				14	{ 9, 5 }
demanda10	35	8	7	5	{ 0, 1, 1, 0, 0, 1, 1, 1 }
				5	
				5	
				4	{ 2, 1, 1, 0, 0, 0, 0, 0 }
				6	{ 1, 1, 0, 1, 2, 1, 0, 0 }
				4	{ 0, 1, 1, 0, 0, 0, 2, 0 }
				6	{ 2, 0, 0, 0, 1, 2, 0, 1 }
demanda11		5	6	4	{ 1, 0, 3, 0, 0 }
				4	
				6	{ 1, 1, 2, 1, 1 }
				6	
				5	{ 1, 1, 1, 1, 1 }
demanda12		2	3	11	{ 7, 4 }
				7	{ 4, 3 }
				15	{ 9, 6 }
demanda13	40	8	8	5	{ 0, 0, 0, 1, 0, 2, 0, 2 }
				5	
				5	
				5	
				5	
				4	{ 0, 1, 1, 0, 0, 0, 2, 0 }
				6	{ 1, 1, 3, 0, 0, 1, 0, 0 }
demanda14		5	4	5	{ 0, 0, 1, 1, 1, 1, 1, 0 }
				7	{ 2, 1, 1, 1, 2 }
				7	
				13	{ 2, 4, 2, 3, 2 }
demanda15		2	5	13	
				8	{ 4, 4 }
				7	{ 5, 2 }
				9	{ 5, 4 }
				6	{ 3, 3 }
				10	{ 6, 4 }

Tabela 5 – Demandas utilizadas nas instâncias-teste geradas

Instância	$V G $	probE	probS	demanda
a1.fest	20	arProb1	habProb12	demanda1
a2.fest	20	arProb2	habProb11	
a3.fest	20	arProb3	habProb10	
a4.fest	20	arProb4	habProb9	
a5.fest	20	arProb5	habProb9	
a6.fest	20	arProb6	habProb10	
a7.fest	20	arProb7	habProb11	
a8.fest	20	arProb8	habProb12	
a9.fest	20	arProb1	habProb8	
a10.fest	20	arProb2	habProb7	
a11.fest	20	arProb3	habProb6	
a12.fest	20	arProb4	habProb5	
a13.fest	20	arProb5	habProb5	
a14.fest	20	arProb6	habProb6	
a15.fest	20	arProb7	habProb7	
a16.fest	20	arProb8	habProb8	
a17.fest	20	arProb1	habProb4	demanda3
a18.fest	20	arProb2	habProb3	
a19.fest	20	arProb3	habProb2	
a20.fest	20	arProb4	habProb1	
a21.fest	20	arProb5	habProb1	
a22.fest	20	arProb6	habProb2	
a23.fest	20	arProb7	habProb3	
a24.fest	20	arProb8	habProb4	
b1.fest	25	arProb1	habProb12	demanda4
b2.fest	25	arProb2	habProb11	
b3.fest	25	arProb3	habProb10	
b4.fest	25	arProb4	habProb9	
b5.fest	25	arProb5	habProb9	
b6.fest	25	arProb6	habProb10	
b7.fest	25	arProb7	habProb11	
b8.fest	25	arProb8	habProb12	
b9.fest	25	arProb1	habProb8	demanda5
b10.fest	25	arProb2	habProb7	
b11.fest	25	arProb3	habProb6	
b12.fest	25	arProb4	habProb5	
b13.fest	25	arProb5	habProb5	
b14.fest	25	arProb6	habProb6	
b15.fest	25	arProb7	habProb7	
b16.fest	25	arProb8	habProb8	
b17.fest	25	arProb1	habProb4	demanda6
b18.fest	25	arProb2	habProb3	
b19.fest	25	arProb3	habProb2	
b20.fest	25	arProb4	habProb1	
b21.fest	25	arProb5	habProb1	
b22.fest	25	arProb6	habProb2	
b23.fest	25	arProb7	habProb3	
b24.fest	25	arProb8	habProb4	
c1.fest	30	arProb1	habProb12	demanda7
c2.fest	30	arProb2	habProb11	
c3.fest	30	arProb3	habProb10	
c4.fest	30	arProb4	habProb9	
c5.fest	30	arProb5	habProb9	
c6.fest	30	arProb6	habProb10	
c7.fest	30	arProb7	habProb11	
c8.fest	30	arProb8	habProb12	
c9.fest	30	arProb1	habProb8	demanda8
c10.fest	30	arProb2	habProb7	
c11.fest	30	arProb3	habProb6	
c12.fest	30	arProb4	habProb5	
c13.fest	30	arProb5	habProb5	
c14.fest	30	arProb6	habProb6	
c15.fest	30	arProb7	habProb7	
c16.fest	30	arProb8	habProb8	

Tabela 6 – Principais características das instâncias-teste do *FEST* - Parte I

Instância	$V G $	probE	probS	demanda
c17.fest	30	arProb1	habProb4	demanda9
c18.fest	30	arProb2	habProb3	
c19.fest	30	arProb3	habProb2	
c20.fest	30	arProb4	habProb1	
c21.fest	30	arProb5	habProb1	
c22.fest	30	arProb6	habProb2	
c23.fest	30	arProb7	habProb3	
c24.fest	30	arProb8	habProb4	
d1.fest	35	arProb1	habProb12	demanda10
d2.fest	35	arProb2	habProb11	
d3.fest	35	arProb3	habProb10	
d4.fest	35	arProb4	habProb9	
d5.fest	35	arProb5	habProb9	
d6.fest	35	arProb6	habProb10	
d7.fest	35	arProb7	habProb11	
d8.fest	35	arProb8	habProb12	
d9.fest	35	arProb1	habProb8	demanda11
d10.fest	35	arProb2	habProb7	
d11.fest	35	arProb3	habProb6	
d12.fest	35	arProb4	habProb5	
d13.fest	35	arProb5	habProb5	
d14.fest	35	arProb6	habProb6	
d15.fest	35	arProb7	habProb7	
d16.fest	35	arProb8	habProb8	
d17.fest	35	arProb1	habProb4	demanda12
d18.fest	35	arProb2	habProb3	
d19.fest	35	arProb3	habProb2	
d20.fest	35	arProb4	habProb1	
d21.fest	35	arProb5	habProb1	
d22.fest	35	arProb6	habProb2	
d23.fest	35	arProb7	habProb3	
d24.fest	35	arProb8	habProb4	
e1.fest	40	arProb1	habProb12	demanda13
e2.fest	40	arProb2	habProb11	
e3.fest	40	arProb3	habProb10	
e4.fest	40	arProb4	habProb9	
e5.fest	40	arProb5	habProb9	
e6.fest	40	arProb6	habProb10	
e7.fest	40	arProb7	habProb11	
e8.fest	40	arProb8	habProb12	
e9.fest	40	arProb1	habProb8	demanda14
e10.fest	40	arProb2	habProb7	
e11.fest	40	arProb3	habProb6	
e12.fest	40	arProb4	habProb5	
e13.fest	40	arProb5	habProb5	
e14.fest	40	arProb6	habProb6	
e15.fest	40	arProb7	habProb7	
e16.fest	40	arProb8	habProb8	
e17.fest	40	arProb1	habProb4	demanda15
e18.fest	40	arProb2	habProb3	
e19.fest	40	arProb3	habProb2	
e20.fest	40	arProb4	habProb1	
e21.fest	40	arProb5	habProb1	
e22.fest	40	arProb6	habProb2	
e23.fest	40	arProb7	habProb3	
e24.fest	40	arProb8	habProb4	

Tabela 7 – Principais características das instâncias-teste do *FEST* - Parte II

Inst	A1	A1 <sub>f</sub>	B2	B2 <sub>f</sub>	R <sub>A1</sub>	R <sub>A1<sub>f</sub></sub>	R <sub>B2</sub>	R <sub>B2<sub>f</sub></sub>	OPT
a4.fest	0,02	0,01	0,03	0,05	69	9	124	124	9
a20.fest	0,15	0,03	0,46	0,25	131	1	131	131	1
a6.fest	0,11	0,04	0,59	0,4	284	46	420	420	46
b4.fest	0,24	0,06	0,43	0,42	191	11	216	216	11
a2.fest	0,25	0,03	0,47	0,62	171	29	310	310	29
a7.fest	0,5	0,05	0,45	0,49	417	101	764	764	101
a12.fest	0,42	0,43	0,44	1,4	157	44	157	157	44
a8.fest	0,46	0,44	1,57	1,07	637	73	806	806	73
a3.fest	1,05	0,42	1,17	1,01	344	55	443	443	55
a1.fest	0,41	0,39	2,22	1,05	407	52	485	485	52
c4.fest	0,57	0,34	2,23	1,62	276	9	299	299	9
a5.fest	0,51	0,6	2,69	3,1	800	112	1024	1024	112
b12.fest	0,64	0,39	2,41	3,85	192	12	192	192	12
a9.fest	0,88	0,6	6,63	1,53	434	167	437	437	167
a14.fest	0,79	0,57	8,96	1,86	529	192	529	529	192
b6.fest	2,6	0,5	7,53	3,29	552	80	787	787	80
b8.fest	3,2	1,22	11,77	4,7	1049	123	1274	1274	123
b20.fest	3,55	3,5	7,76	5,95	180	20	209	209	20
a10.fest	0,98	1,05	13,71	2,58	323	125	323	323	125
a16.fest	4,37	2,11	14,97	2,67	856	328	861	861	328
b1.fest	3,61	1,6	14,77	6,84	624	94	784	784	94
b5.fest	2,34	1,03	16,16	7,96	1304	200	1635	1635	200
a13.fest	1,37	0,73	23,59	5,28	1096	462	1100	1100	462
a15.fest	0,86	1,08	26,99	2,24	724	317	754	754	317
a11.fest	1,5	0,58	26,13	3,81	463	193	463	463	193
c6.fest	5,6	1,26	13,41	13,16	991	67	1114	1114	67
b2.fest	2,73	1,47	10,19	25,02	423	62	500	500	62
c2.fest	4,73	2,27	17,72	15,29	573	46	635	635	46
d4.fest	1,05	1,73	12,86	24,73	414	7	431	431	7
a18.fest	4,16	3,37	26,52	13,12	276	23	276	276	23
b3.fest	3,47	1,84	41,63	23,84	499	92	716	716	92
c7.fest	6,64	2,78	55,17	39,06	1531	151	1671	1671	151
a17.fest	22,53	13,3	31,73	37,94	467	48	467	467	48
a24.fest	31,35	28,81	69,18	40,66	886	88	886	886	93
b14.fest	19,58	3,71	86,19	59,16	685	84	723	723	84
b7.fest	10,75	4,25	130,76	23,91	941	155	1162	1162	155
a23.fest	12,81	21,89	92,13	79,44	728	88	728	728	88
b16.fest	20,83	7,69	133,64	53,06	1206	165	1329	1329	165
a22.fest	23,02	12,25	84,99	98,33	445	39	445	445	39
c1.fest	17,96	6,17	199,66	123,22	995	101	1082	1082	101
b22.fest	105,05	37,49	157,34	66,24	715	134	762	762	134
a21.fest	93,17	14,25	188,91	88,66	1139	158	1139	1139	158
c8.fest	45,07	14,17	206,37	194,33	1846	138	1943	1943	138
b24.fest	180,45	157,4	362,05	424,8	1194	244	1294	1294	244
c5.fest	56,98	4,98	602,55	79,83	2180	238	2454	2454	238
c3.fest	52,33	7	940,11	199,68	959	95	1042	1042	95
b18.fest	204,48	110,88	1390,15	468,82	433	92	468	468	92
b9.fest	59,33	15,27	1600,75	177,18	696	108	763	763	108
b15.fest	373,14	59,63	2067,9	624,58	1224	194	1224	1224	194
a19.fest	135,85	13,31	1983,86	54,43	474	69	474	474	69
b13.fest	69,05	13,32	2655,49	449,99	1574	263	1727	1727	263
b10.fest	417,84	82,29	2873,8	806,94	500	66	500	500	66
b21.fest	2255,92	156,53	3287,67	368,26	1536	366	1682	1682	366
b17.fest	422,29	306,4	3969,76	1132,85	680	156	732	732	156

Tabela 8 – Resultados obtidos com cada implementação para o problema inteiro

Para uma melhor visualização, estes mesmos dados são também apresentados graficamente nas Figuras 8, 9 e 10. Para cada uma das instâncias-teste, contabilizamos a média do tempo de execução obtido pelas 4 implementações, agrupando-as da seguinte forma:

- grupo A - resultados com média do tempo de execução entre 0 e 5 segundos
- grupo B - resultados com média do tempo de execução entre 5 e 50 segundos
- grupo C - resultados com média do tempo de execução entre 50 e 1500 segundos

Note que o eixo vertical de cada gráfico está representado em escala logarítmica.

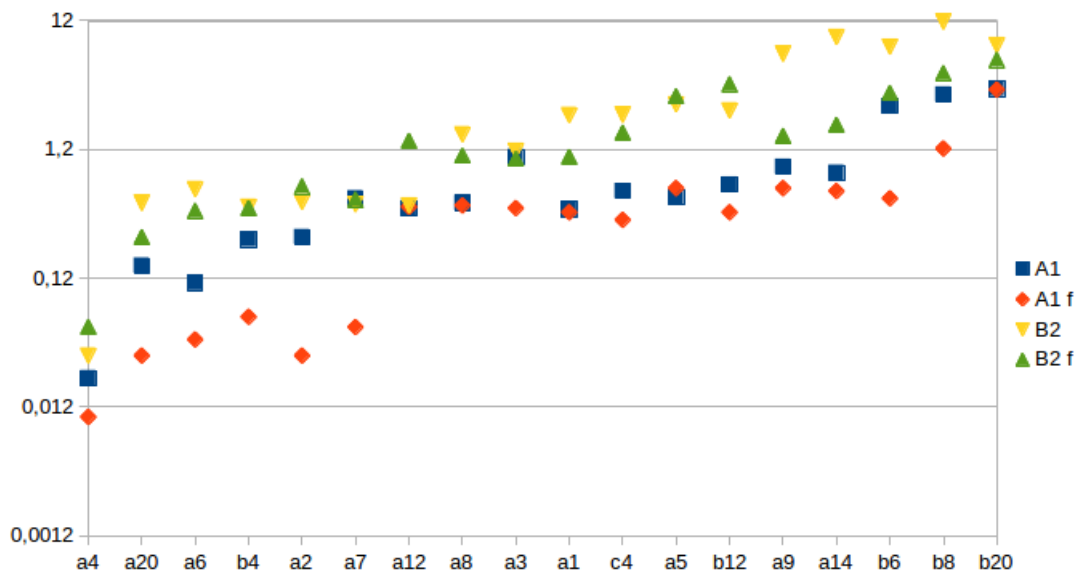


Figura 8 – tempo de execução (segundos) × instância do grupo A

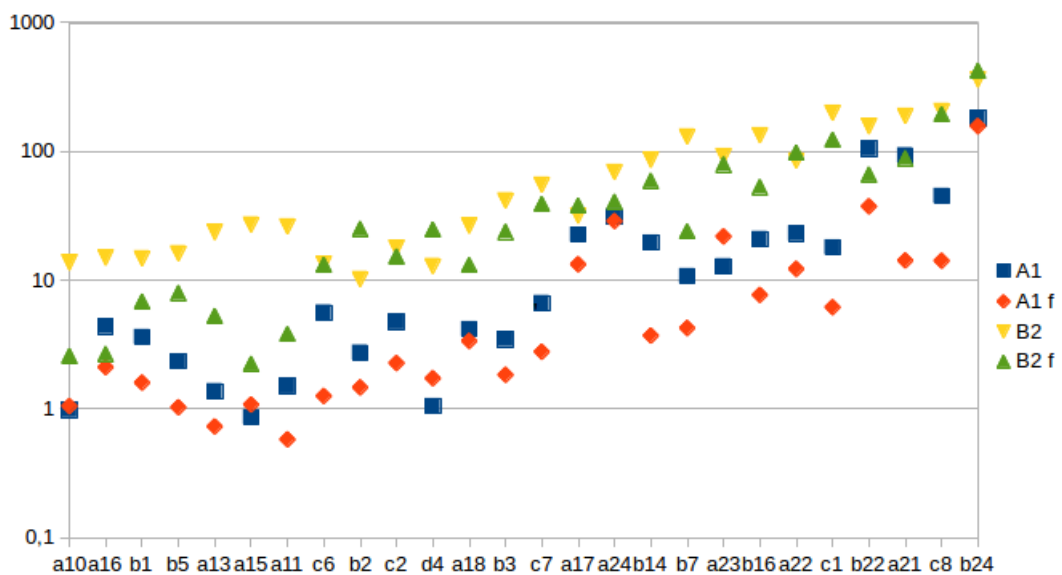


Figura 9 – tempo de execução (segundos) × instância do grupo B

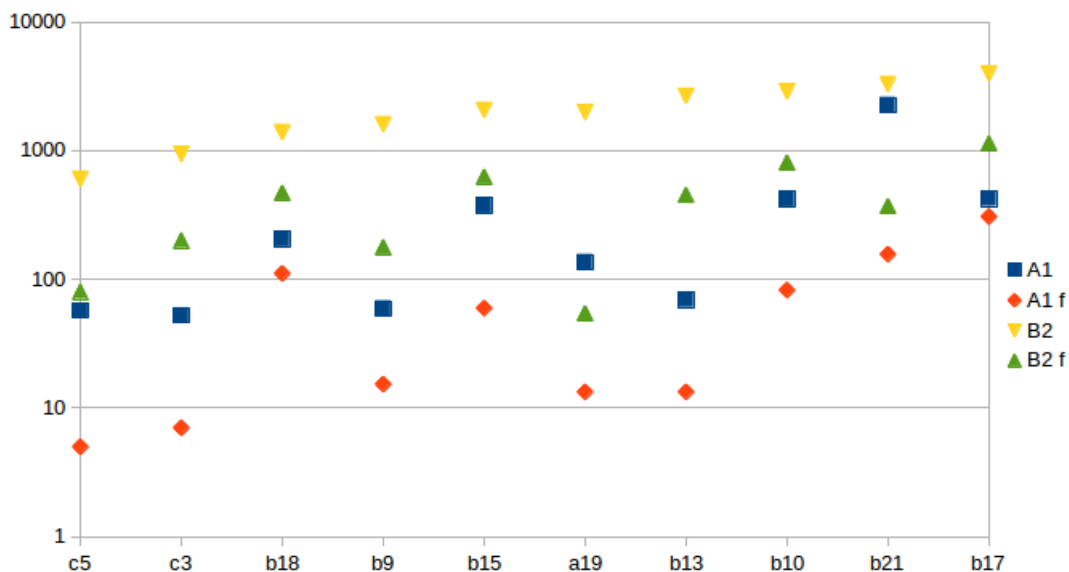


Figura 10 – tempo de execução (segundos) × instância do grupo C

As tabelas 9 e 10 apresentam a média ( $\mu$ ) e o desvio padrão ( $\sigma$ ) do tempo de execução de cada grupo de instâncias.

Instâncias	$\mu_{A1}$	$\mu_{A1_f}$	$\mu_{B2_f}$	$\mu_{B2_f}$
Grupo A	0,9083	0,534	3,21167	1,814
Grupo B	6,080625	4,063125	25,920625	16,5125
Grupo C	228,795	52,444	1150,704	278,6855

Tabela 9 – Médias dos tempos de computação das várias implementações para a solução do problema inteiro

Instâncias	$\sigma_{A1}$	$\sigma_{A1_f}$	$\sigma_{B2_f}$	$\sigma_{B2_f}$
Grupo A	1,0629606268	0,7998619162	3,6150441377	1,6832679079
Grupo B	8,4774024864	7,2495459341	16,6229389575	13,6921580963
Grupo C	495,5158945616	77,459946553	1249,8737515715	297,206766879

Tabela 10 – Desvios padrões dos tempos de computação das várias implementações para a solução do problema inteiro

Para avaliar o desempenho das implementações, apresentamos uma análise de variância dos resultados obtidos, no intuito de verificar se de fato a implementação que possui menor média de tempo de execução ( $\mu_{A1_f}$ ) pode ser considerada estatisticamente melhor que as demais implementações. Para isso faremos uso do teste de comparação múltipla denominado *MCB* (*Multiple Comparisons with the Best*). Proposto por Jason Hsu (HSU, 1996), o *MCB* tem como característica principal comparar todos os tratamentos com o melhor.



A análise de variância foi aplicada a cada grupo ( $A$ ,  $B$  e  $C$ ) com o nível de significância de 5%, com o seguinte teste de hipótese:

$$\begin{cases} H_0 : \mu_{A1_f} < \mu_{A1}, \mu_{B2}, \mu_{A2_f} \\ H_1 : \text{existe ao menos uma média estatisticamente igual a } \mu_{A1_f} \end{cases}$$

Para realização do *MCB* os dados de cada grupo ( $A$ ,  $B$  e  $C$ ) foram normalizados utilizando a transformação Box-Cox (BOX; COX, 1964), garantindo a homogeneidade das variâncias através do Teste de Bartlett (BARTLETT, 1937).

A análise de variância aplicada indicou a não existência de diferença estatisticamente significativa entre as  $\mu_{A1}$  e  $\mu_{A1_f}$  para o Grupo  $A$ , sendo aceita a hipótese  $H_0$  para os demais grupos ( $B$  e  $C$ ). O gráfico 11 apresenta as diferenças das médias estatísticas para cada grupo teste.

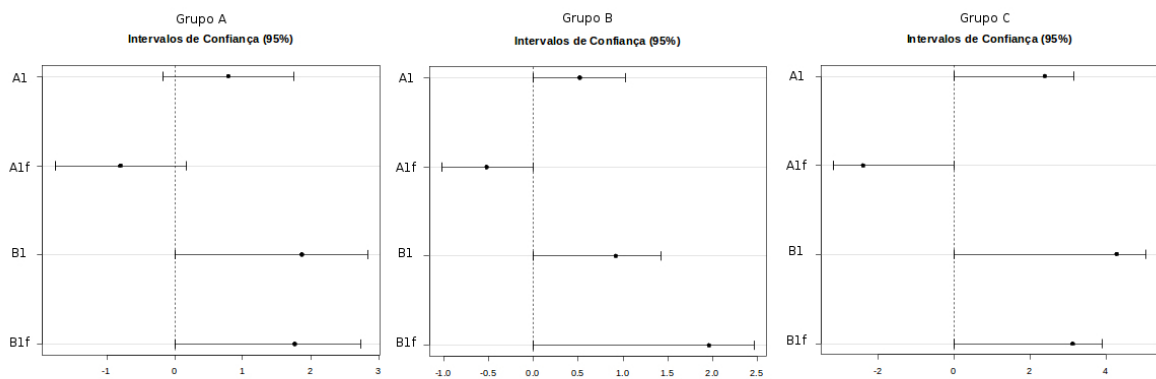


Figura 11 – Gráfico com as diferenças das médias para os grupos  $A$ ,  $B$ , e  $C$

A ausência de diferença estatística entre as médias  $\mu_{A1}$  e  $\mu_{A1_f}$  para o Grupo  $A$  explica-se devido ao fato das instâncias pertencentes a este grupo serem consideradas fáceis (tempo de execução inferior a 15 segundos), não havendo grandes diferenças no tempo de execução das implementações propostas. Por outro lado, como o intervalo de confiança referente à versão  $A1_f$  nos grupos  $B$  e  $C$  está totalmente contido no semi-eixo negativo, podemos dizer que ela é a melhor (menor média) entre as demais implementações. Sendo assim, podemos considerar em termos gerais que estatisticamente a versão  $A1_f$  obtém um melhor desempenho em relação às demais implementações, justificando o fato de somente a versão  $A1_f$  resolver um subconjunto significativo de instâncias-teste no tempo máximo de execução determinado neste trabalho (5000 segundos).

A Tabela 11 apresenta o tempo de execução das instâncias-teste que foram resolvidas no tempo máximo de execução de 5000 segundos por alguma implementação (mas não por todas). Versões que obtiveram tempo de execução maior que o limite estipulado para resolução de uma determinada instância foram representadas com o símbolo  $TL$  e implementações que não conseguiram obter a solução ótima por problemas de escassez de memória são representadas com o símbolo  $EM$ . Claramente, a versão  $A1_f$  é superior

nessa comparação. Ela sempre consegue resolver todas as instâncias que alguma outra versão também é capaz, além de várias outras instâncias para as quais nenhuma outra versão encontra o ótimo no tempo limite.

Inst	A1	A1 <sub>f</sub>	B2	B2 <sub>f</sub>	OPT
b19.fest	1251,62	231,7	TL	535,94	166
b11.fest	143,76	12,53	TL	677,3	110
d1.fest	1103,43	1096,88	TL	TL	97
d2.fest	757,58	14,59	TL	4917,1	66
c12.fest	8	16,47	EM	EM	44
d12.fest	2,81	0,81	EM	EM	2
c20.fest	21,77	18,52	EM	EM	51
c14.fest	667,97	28,4	EM	EM	204
d6.fest	443,08	14,9	EM	EM	84
d7.fest	537,45	35,43	EM	EM	180
d14.fest	379,83	839,8	EM	EM	57
d16.fest	975,24	1038,82	EM	EM	107
c11.fest	4185,48	375,37	EM	EM	270
d3.fest	2384,35	406,21	EM	EM	116
c16.fest	1122,84	940,24	EM	EM	412
c18.fest	4669,54	1339,33	EM	EM	164
d8.fest	1712,01	526,41	EM	EM	167
d9.fest	2160,54	1925,08	EM	EM	73
d10.fest	1922,02	450,46	EM	EM	46
c15.fest	TL	2122,95	EM	EM	423
c13.fest	TL	904,44	EM	EM	607
d15.fest	TL	809,49	EM	EM	150
e1.fest	TL	4988,84	EM	EM	110
d11.fest	TL	2385,24	EM	EM	86
d13.fest	TL	4497,58	EM	EM	191
d5.fest	TL	326,34	EM	EM	246
e1.fest	TL	4988,84	EM	EM	110
c9.fest	TL	4541,84	EM	EM	223
c10.fest	TL	793,29	TL	TL	153
c21.fest	TL	1990,69	EM	EM	655
c22.fest	TL	1162,12	EM	EM	254
c23.fest	EM	567,86	EM	EM	448
c24.fest	TL	2242,08	EM	EM	443
c17.fest	EM	2019,36	EM	EM	263
d20.fest	82,91	47,28	TL	TL	50
e2.fest	TL	3231,46	EM	EM	69
e4.fest	143,3	39,1	TL	TL	8
e5.fest	TL	4666,36	TL	TL	277
e6.fest	389,45	121,88	TL	TL	96
e8.fest	EM	4755,33	EM	EM	142
e12.fest	2947	829,49	EM	EM	49
e20.fest	642,87	436,37	TL	TL	11

Tabela 11 – Tempos de computação das várias implementações para a solução do problema inteiro

### 6.4.2 Limite inferior

Embora a versão  $A1_f$  tenha resolvido um grande número de instâncias-teste dentro do tempo de execução limite estipulado (96 instâncias das 120 instâncias criadas), a implementação ainda encontra dificuldades para a resolução de instâncias consideradas mais complexas. Das 24 instâncias-teste não resolvidas apenas cinco instâncias foram solucionadas utilizando um tempo de execução consideravelmente maior que o tempo limite estipulado. Para as 19 instâncias-teste não resolvidas restantes nenhuma versão proposta conseguiu obter uma solução ótima devido a escassez de memória. Procurando analisar melhor essa questão, efetuamos testes com as relaxações lineares das implementações propostas, buscando avaliar a qualidade do limite inferior gerado. Consideramos também uma segunda relaxação para a versão  $A1_f$ , onde apenas a integralidade das variáveis  $X_{ujs}$  são relaxadas. Essa versão mista é denotada  $A1_{fm}$ . As tabelas 12 e 13 apresentam o limite inferior dado por cada implementação relaxada para as 96 instâncias-teste já solucionadas.

Nestas tabelas, estão presentes a coluna de identificação da instância solucionada (Inst), o valor ótimo da função objetivo dado pela versão  $A1$  relaxada ( $A1_r$ ), o valor ótimo da função objetivo dado pela versão  $A1$  fortalecida relaxada ( $A1_{fr}$ ), o valor ótimo da função objetivo dado pela versão  $A1$  fortalecida mista ( $A1_{fm}$ ), o valor ótimo da função objetivo dado pela versão  $B2$  relaxada ( $B2_r$ ), o valor ótimo da função objetivo dado pela versão  $B2$  fortalecida relaxada ( $B2_{fr}$ ) e o valor ótimo inteiro ( $OPT$ ). Além disso, apresentamos o tempo de execução (segundos) da versão  $A1$  fortalecida relaxada ( $T(A1_{fr})$ ), o tempo de execução (segundos) da versão  $A1$  fortalecida mista ( $T(A1_{fm})$ ), o tempo de execução (segundos) da versão  $B2$  fortalecida relaxada ( $T(B2_{fr})$ ), o tempo de execução (segundos) da versão  $A1$  fortalecida inteira ( $T(OPT)$ ).

Como demonstrando na Proposição 5.27, o limite inferior dado pela versão  $A1$  relaxada é igual ao limite inferior dado pela versão  $B2$  relaxada, sendo os limites inferiores obtidos por estas versões consideravelmente piores em comparação com as versões fortalecidas  $A1_f$  e  $B2_f$ , demonstrando que a inclusão das desigualdades apresentadas pode de fato fortalecer as formulações propostas neste trabalho.

Para uma melhor visualização dos resultados, apresentamos graficamente a comparação entre o melhor limite inferior (LI) (obtido pela versão  $A1_{fm}$ ) e o valor ótimo (OPT) dessas 96 instâncias, através do  $GAP = (OPT-LI)/OPT$ . O gráfico representado na Figura 12 apresenta todas as instâncias-teste que resultam em um GAP de otimalidade de até 35% (49 instâncias). O gráfico representado na Figura 13 apresenta todas as instâncias-teste que resultam em um GAP de otimalidade acima de 35% (47 instâncias).

Como pode-se notar, embora a versão  $A1_{fm}$  apresente um bom limite inferior para um número razoável de instâncias-teste, encontrando a solução ótima para algumas delas, esta relaxação ainda apresenta dificuldades para encontrar bons limites inferiores para instâncias consideradas mais complexas. Além disso, para algumas instâncias-teste o tempo de execução da relaxação  $A1_{fm}$  é equivalente ou maior ao tempo de execução da

Inst	$A1_r$	$A1_{fr}$	$A1_{fm}$	$B1_r$	$B1_{fr}$	$OPT$	$T(A1_{fr})$	$T(A1_{fm})$	$T(B1_{fr})$	$T(OPT)$
a1.fest	26,0	34,0	48,0	26,0	28,5	52,0	0,009	0,841	0,013	0,39
a2.fest	17,0	26,0	29,0	17,0	18,5	29,0	0,008	0,148	0,014	0,03
a3.fest	30,5	30,5	55,0	30,5	36,9	55,0	0,008	0,486	0,008	0,42
a4.fest	6,3	7,1	9,0	6,3	6,3	9,0	0,004	0,033	0,005	0,01
a5.fest	43,0	81,6	112,0	43,0	63,2	112,0	0,013	0,699	0,009	0,6
a6.fest	35,6	42,6	46,0	35,6	36,5	46,0	0,006	0,041	0,007	0,04
a7.fest	53,1	94,5	101,0	53,1	73,2	101,0	0,011	0,397	0,007	0,05
a8.fest	36,7	43,9	54,0	36,7	38,5	73,0	0,007	0,199	0,009	0,44
a9.fest	0,0	109,5	110,0	0,0	83,0	167,0	0,026	0,029	0,006	0,6
a10.fest	0,0	77,0	77,0	0,0	67,0	125,0	0,030	0,045	0,005	1,05
a11.fest	0,0	158,0	158,0	0,0	145,0	193,0	0,032	0,028	0,005	0,58
a12.fest	0,0	15,0	15,0	0,0	4,0	44,0	0,009	0,018	0,004	0,43
a13.fest	0,0	370,0	370,0	0,0	323,0	462,0	0,023	0,029	0,006	0,73
a14.fest	0,0	115,0	115,0	0,0	77,0	192,0	0,019	0,030	0,005	0,57
a15.fest	0,0	258,0	258,0	0,0	230,0	317,0	0,024	0,031	0,005	1,08
a16.fest	0,0	199,0	199,0	0,0	155,0	328,0	0,026	0,044	0,005	2,11
a17.fest	0,0	10,2	12,0	0,0	8,5	48,0	0,020	0,214	0,007	13,3
a18.fest	0,0	10,4	11,0	0,0	8,6	23,0	0,027	0,162	0,008	3,37
a19.fest	0,0	48,0	50,0	0,0	42,4	69,0	0,039	1,211	0,010	13,31
a20.fest	0,0	0,0	0,0	0,0	0,0	1,0	0,003	0,019	0,007	0,03
a21.fest	0,0	103,5	112,0	0,0	101,0	158,0	0,031	0,019	0,009	14,25
a22.fest	0,0	11,6	13,0	0,0	10,5	39,0	0,035	0,421	0,009	12,25
a23.fest	0,0	37,8	42,0	0,0	28,5	88,0	0,028	0,816	0,010	21,89
a24.fest	0,0	15,9	21,0	0,0	12,9	93,0	0,022	0,396	0,009	28,81
b1.fest	32,0	66,5	94,0	32,0	39,0	94,0	0,015	0,452	0,016	1,6
b2.fest	17,0	43,0	61,0	17,0	22,2	62,0	0,017	1,776	0,012	1,47
b3.fest	32,0	67,8	92,0	32,0	46,4	92,0	0,032	0,544	0,007	1,84
b4.fest	3,5	5,6	8,0	3,5	3,5	11,0	0,006	0,135	0,009	0,06
b5.fest	63,0	161,4	200,0	63,0	112,4	200,0	0,028	0,761	0,013	1,03
b6.fest	32,5	61,0	80,0	32,5	41,3	80,0	0,014	0,644	0,009	0,5
b7.fest	37,0	111,0	155,0	37,0	55,5	155,0	0,032	4,552	0,012	4,25
b8.fest	39,5	67,4	106,0	39,5	40,5	123,0	0,012	3,050	0,011	1,22
b9.fest	0,0	53,4	69,0	0,0	18,9	108,0	0,047	1,443	0,012	15,27
b10.fest	0,0	31,8	42,0	0,0	14,3	66,0	0,064	0,028	0,011	82,29
b11.fest	0,0	67,7	90,0	0,0	44,5	110,0	0,087	29,380	0,012	12,53
b12.fest	0,0	0,1	1,0	0,0	0,0	12,0	0,013	0,082	0,009	1,39
b13.fest	0,0	179,8	247,0	0,0	120,6	263,0	0,090	1,803	0,014	13,32
b14.fest	0,0	47,8	66,0	0,0	23,2	84,0	0,051	1,879	0,011	3,71
b15.fest	0,0	117,2	165,0	0,0	85,9	194,0	0,102	20,058	0,014	59,63
b16.fest	0,0	68,7	91,0	0,0	10,8	165,0	0,043	1,012	0,011	7,69
b17.fest	0,0	55,0	55,0	0,0	36,5	156,0	0,030	490,760	0,009	306,4
b18.fest	0,0	39,5	40,0	0,0	29,0	92,0	0,038	178,459	0,009	110,88
b19.fest	0,0	107,1	108,0	0,0	93,2	166,0	0,064	296,278	0,010	231,7
b20.fest	0,0	0,0	0,0	0,0	0,0	20,0	0,003	0,261	0,008	3,8
b21.fest	0,0	228,5	229,0	0,0	223,0	366,0	0,059	198,033	0,010	156,53
b22.fest	0,0	54,5	55,0	0,0	50,5	134,0	0,046	5,733	0,010	37,49
b23.fest	0,0	163,5	166,0	0,0	146,6	270,0	0,051	239,917	0,011	326,47
b24.fest	0,0	63,5	64,0	0,0	39,5	244,0	0,031	123,046	0,010	157,4

Tabela 12 – Resultados obtidos pelas implementações relaxadas - Parte I

Inst	$A1_r$	$A1_{fr}$	$A1_{fm}$	$B1_r$	$B1_{fr}$	$OPT$	$T(A1_{fr})$	$T(A1_{fm})$	$T(B1_{fr})$	$T(OPT)$
c1.fest	7,3	58,6	98,0	7,3	7,3	101,0	0,044	10,981	0,018	6,17
c2.fest	3,3	24,0	43,0	3,3	3,3	46,0	0,027	0,728	0,014	2,27
c3.fest	2,0	61,5	95,0	2,0	2,0	95,0	0,070	1,881	0,015	7
c4.fest	0,5	1,0	2,0	0,5	0,5	9,0	0,005	0,206	0,010	0,34
c5.fest	10,6	169,0	238,0	10,6	10,6	238,0	0,057	6,273	0,015	4,98
c6.fest	7,2	39,5	67,0	7,2	7,2	67,0	0,030	9,570	0,016	1,26
c7.fest	14,5	113,4	151,0	14,5	14,5	151,0	0,057	0,670	0,017	2,78
c8.fest	3,1	47,4	118,0	3,1	3,1	138,0	0,033	8,596	0,018	14,17
c9.fest	0,0	71,8	83,0	0,0	47,5	223	0,115	664,090	0,014	4541,84
c10.fest	0,0	72,0	77,0	0,0	51,5	153	0,121	24,610	0,017	793,29
c11.fest	0,0	194,3	215,0	0,0	156,0	270,0	0,175	187,802	0,017	375,37
c12.fest	0,0	1,0	1,0	0,0	0,0	44,0	0,007	0,039	0,010	16,47
c13.fest	0,0	385,5	407,0	0,0	348,0	607	0,184	209,061	0,016	904,44
c14.fest	0,0	101,0	114,0	0,0	70,0	204	0,128	15,776	0,016	28,40
c15.fest	0,0	279,5	302,0	0,0	239,0	423	0,187	1917,756	0,017	2122,95
c16.fest	0,0	153,1	161,0	0,0	65,0	412	0,058	861,487	0,017	940,24
c17.fest	16,6	93,9	263,0	16,6	80,5	263	0,079	318,245	0,026	2019,36
c18.fest	10,5	68,3	102,0	10,5	66,2	164	0,162	993,942	0,029	1339,33
c20.fest	3,5	3,6	10,0	3,5	3,5	51	0,008	20,136	0,016	18,52
c21.fest	36,8	433,7	433,7	36,8	417,3	655	0,208	471,601	0,033	1990,69
c22.fest	13,4	102,1	102,1	13,4	95,5	254	0,112	307,477	0,029	1162,12
c23.fest	26,9	292,1	292,1	26,9	280,1	448	0,176	667,708	0,035	567,86
c24.fest	31,0	145,1	145,1	31,0	118,4	443	0,087	397,582	0,033	2242,08
d1.fest	6,3	24,5	45,0	6,3	6,3	97	0,076	92,890	0,069	1096,88
d2.fest	8,0	37,7	54,0	8,0	11,8	66	0,072	19,979	0,040	14,59
d3.fest	12,3	71,3	99,0	12,3	30,3	116	0,163	353,923	0,028	406,21
d4.fest	0,3	0,5	2,0	0,3	0,3	7	0,009	0,160	0,032	1,73
d5.fest	23,3	162,0	212,0	23,3	92,1	246	0,211	403,221	0,065	326,34
d6.fest	8,5	45,3	69,0	8,5	15,5	84	0,143	690,060	0,032	14,90
d7.fest	19,0	121,3	152,0	19,0	47,6	180	0,104	32,940	0,033	35,43
d8.fest	23,3	57,2	98,0	23,3	24,7	167	0,055	64,760	0,068	526,41
d9.fest	0,0	18,0	18,0	0,0	2,0	73	0,051	0,190	0,020	1925,08
d10.fest	0,0	16,0	16,0	0,0	0,0	46	0,071	0,120	0,019	450,46
d11.fest	0,0	44,0	45,0	0,0	15,0	86	0,167	0,260	0,026	2385,24
d12.fest	0,0	0,0	0,0	0,0	0,0	2	0,007	0,020	0,017	0,81
d13.fest	0,0	105,9	109,0	0,0	45,0	191	0,207	1,000	0,028	5497,58
d14.fest	0,0	24,0	24,0	0,0	4,0	57	0,094	0,210	0,026	839,80
d15.fest	0,0	86,4	89,0	0,0	27,0	150	0,149	0,610	0,024	809,49
d16.fest	0,0	22,0	22,0	0,0	2,0	107	0,060	0,090	0,023	1038,82
d20.fest	0,0	10	0,0	0,0	0,0	50	0,005	0,010	0,011	47,28
e1.fest	15,8	42,3	58,0	15,8	17,5	110	0,120	11,740	0,140	4988,84
e2.fest	17,5	38,7	46,0	17,5	20,5	69	0,086	1,440	0,070	3231,41
e4.fest	0,0	3,8	6,0	0,0	3,8	8	0,007	0,260	3,800	39,1
e5.fest	59,5	172,3	203,0	59,5	106,8	277	0,106	5,820	0,057	4666,36
e6.fest	30,0	65,0	76,0	30,0	36,5	96	0,056	1,620	0,076	121,88
e8.fest	9,7	41,9	62,0	9,7	9,7	142	0,054	3,500	0,075	4755,33
e12.fest	0,0	0,0	12,0	0,0	0,0	49	0,007	0,040	0,036	4755,33
e20.fest	0,0	0,0	3,0	0,0	0,0	11	0,007	0,030	0,033	436,37

Tabela 13 – Resultados obtidos pelas implementações relaxadas - Parte II

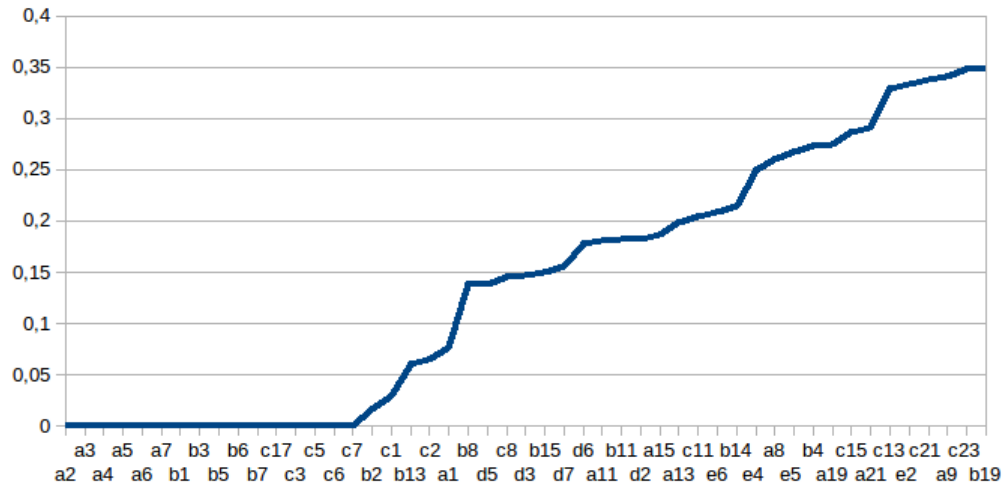


Figura 12 – GAP × instância-teste

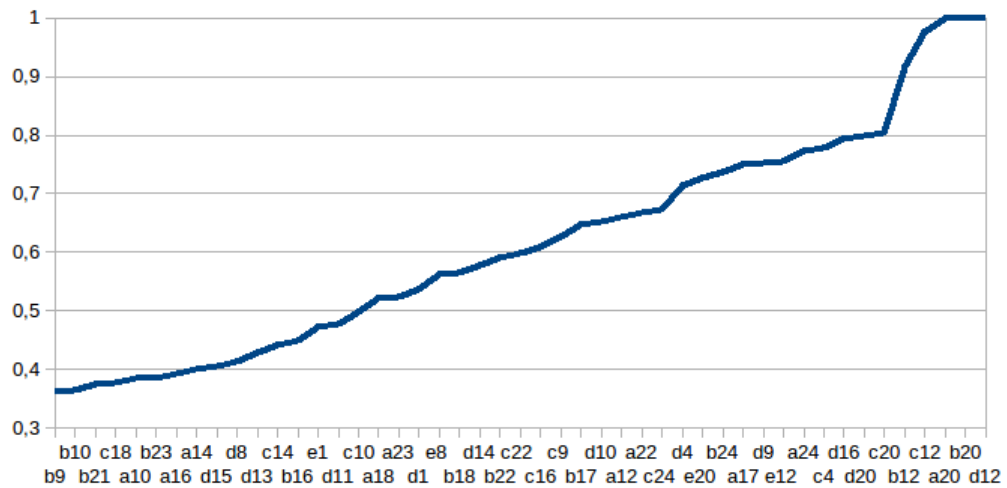


Figura 13 – GAP × instância-teste

versão  $A1_f$  inteira. Acredita-se que a inclusão de novas desigualdades válidas para um maior fortalecimento da Formulação 1 possa melhorar o desempenho desta formulação, a partir da geração de melhores limites inferiores.

A Tabela 14 apresenta os limites inferiores obtidos pelas implementações para as instâncias-teste ainda não resolvidas por qualquer das quatro versões.

Inst	$A1_r$	$A1_{fr}$	$A1_{fm}$	$B1_r$	$B1_{fr}$	$T(A1_{fr})$	$T(A1_{fm})$	$T(B1_{fr})$
c19.fest	18,4	208,0	EM	18,4	197,9	0,145	457,819	0,033
d17.fest	0,0	118,3	122,0	0,0	86,4	0,104	85,730	0,018
d18.fest	0,0	80,4	82,0	0,0	74,0	0,169	19,500	0,018
d19.fest	0,0	233,7	240,0	0,0	200,9	0,161	608,360	0,018
d21.fest	0,0	488,1	493,0	0,0	476,0	0,309	36,960	0,020
d22.fest	0,0	84,6	86,0	0,0	81,0	0,149	2,150	0,024
d23.fest	0,0	338,9	344,0	0,0	323,8	0,154	409,590	0,018
d24.fest	0,0	144,8	152,0	0,0	75,0	0,086	394,320	0,020
e3.fest	23,1	67,1	81,0	23,1	23,1	0,067	4,810	0,051
e7.fest	29,5	119,3	146,0	29,5	51,5	0,091	8,890	0,053
e9.fest	0,0	136,4	143,0	0,0	58,0	0,343	3,300	0,050
e10.fest	0,0	72,2	77,0	0,0	38,0	0,294	11,130	0,039
e11.fest	0,0	221,0	229,0	0,0	166,0	1,122	699,750	0,042
e13.fest	0,0	502,8	514,0	0,0	483,0	0,021	594,790	0,065
e14.fest	0,0	94,8	96,0	0,0	68,5	2,097	1,020	0,059
e15.fest	0,0	352,7	368,0	0,0	239,0	0,434	30,830	0,049
e16.fest	0,0	52,4	55,0	0,0	18,0	0,560	1,120	0,050
e17.fest	0,0	13,0	17,0	0,0	8,0	0,246	14,010	0,053
e18.fest	0,0	7,7	11,0	0,0	1,3	0,208	111,540	0,173
e19.fest	0,0	93,0	96,0	0,0	81,8	0,539	47,340	0,060
e21.fest	0,0	255,8	EM	0,0	244,3	1,227	EM	0,221
e22.fest	0,0	57,3	EM	0,0	53,5	0,893	EM	0,221
e23.fest	0,0	133,3	EM	0,0	115,7	0,705	EM	0,104
e24.fest	0,0	6,5	EM	0,0	3,3	0,193	EM	0,157

Tabela 14 – Resultados obtidos para instâncias-teste não resolvidas pelas quatro versões

### 6.4.3 Limite superior

Buscando encontrar um bom limite superior para as instâncias-teste não resolvidas pelas implementações exatas propostas neste trabalho, efetuamos testes computacionais com o algoritmo heurístico proposto na Subseção 4.3.1. Dentre as 96 instâncias-teste resolvidas pela versão  $A1_f$ , o algoritmo heurístico proposto obteve solução ótima para 94 instâncias. A Tabela 15 apresenta o limite superior encontrado pelo algoritmo heurístico para as duas instâncias-teste restantes, o valor ótimo encontrado pela versão  $A1_f$  e o  $GAP$  de otimalidade, calculado da seguinte forma:

$$GAP = (\text{limite superior} - OPT)/OPT$$

Inst	Limite superior	OPT	GAP
d15.fest	159,0	150,0	0,060
d16.fest	108,0	107,0	0,009

Tabela 15 – Tabela com limites superiores

Os limites superiores obtidos pelo algoritmo heurístico para as 24 instâncias-teste não solucionadas pelas formulações são apresentados na Tabela 16.

Inst	Limite superior	Limite inferior
c19.fest	304,0	208,0
d17.fest	335,0	122,0
d18.fest	207,0	82,0
d19.fest	360,0	240,0
d21.fest	807,0	493,0
d22.fest	275,0	86,0
d23.fest	567,0	344,0
d24.fest	581,0	152,0
e3.fest	115,0	81,0
e7.fest	203,0	146,0
e9.fest	354,0	143,0
e10.fest	202,0	77,0
e11.fest	374,0	229,0
e13.fest	873,0	514,0
e14.fest	287,0	96,0
e15.fest	612,0	368,0
e16.fest	516,0	55,0
e17.fest	197,0	17,0
e18.fest	112,0	11,0
e19.fest	241,0	96,0
e21.fest	550,0	255,8
e22.fest	185,0	57,3
e23.fest	346,0	133,3
e24.fest	319,0	6,5

Tabela 16 – Tabela com os limites superiores e inferiores das instâncias não resolvidas

#### 6.4.4 Resolução do problema inteiro com limites

Na tentativa de resolver otimamente as instâncias-teste ainda não solucionadas pela versão  $A1_f$ , adicionamos os limites inferiores e superiores apresentados na Tabela 16. Inicializamos a versão  $A1_f$  com uma solução inicial (limite superior), dada pelo algoritmo heurístico apresentado na Subseção 4.3.1. Utilizamos o limite superior para podar os nós criados pelo solver CPLEX durante o processo de resolução de uma instância-teste, quando o limite inferior de um nó é maior ou igual que o limite superior dado. A Tabela 17 apresenta as soluções obtidas pela versão  $A1_f$  utilizando a inclusão dos limites como apresentado.

Como pode-se notar embora a versão com inclusão de limites tenha solucionado todas as instâncias-teste ainda não resolvidas até aqui, o tempo de execução para resolução de quase todas as instâncias solucionadas supera o tempo de execução limite proposto.



Inst	OPT	Tempo(OPT)
c19.fest	304,0	6827,45
d17.fest	335,0	8932,08
d18.fest	207,0	7382,05
d19.fest	360,0	1928,23
d21.fest	807,0	2391,16
d22.fest	275,0	10892,33
d23.fest	567,0	9812,70
d24.fest	581,0	5092,55
e3.fest	115,0	7593,25
e7.fest	203,0	9316,84
e9.fest	354,0	6191,03
e10.fest	202,0	2972,49
e11.fest	374,0	3410,32
e13.fest	873,0	4614,07
e14.fest	287,0	6943,19
e15.fest	612,0	4367,28
e16.fest	516,0	3656,95
e17.fest	197,0	8347,38
e18.fest	112,0	6028,10
e19.fest	241,0	7695,47
e21.fest	550,0	8295,85
e22.fest	185,0	7592,33
e23.fest	346,0	15833,63
e24.fest	319,0	12286,53

Tabela 17 – Soluções ótimas encontradas com a adição de limites ao problema inteiro

Apenas 7 instâncias foram solucionadas com tempo de execução inferior à 5000 segundos, porém não houve problemas com escassez de memória como ocorrido com a versão  $A1_f$  sem a inclusão de limites. Vale ressaltar também que o algoritmo heurístico proposto na Subseção 4.3.1 obteve limite superior igual à solução ótima para todas as instâncias-teste apresentadas nesta subseção. Em outras palavras, nosso algoritmo heurístico resolveu na otimalidade 118 instâncias das 120 instâncias-teste criadas, obtendo um desempenho consideravelmente bom em comparação as formulações propostas.

#### 6.4.5 Instâncias reais

Além dos resultados obtidos para as instâncias-teste criadas, apresentamos os resultados para as instâncias reais (alfa e beta). A instância alfa foi resolvida utilizando apenas a versão  $A1_f$  sem adição de limites. Para resolução da instância beta, utilizamos a versão  $A1_{fm}$  para geração de um limite inferior e o algoritmo heurístico apresentado na Subseção 4.3.1 para geração de um limite superior, incluindo os limites encontrados na versão  $A1_f$  inteira. A Tabela 18 apresenta o tempo em segundos  $T(A1_f)$  para resolução

de cada instância, o limite superior encontrado pela algoritmo heurístico, o limite inferior encontrado pela relaxação  $A1_{fm}$  e o valor ótimo da função objetivo (OPT).

Inst	$T(A1_f)$	Limite inferior	Limite superior	$OPT$
alfa.fest	1248.25	0	23	23
beta.fest	5285,03	94	170	170

Tabela 18 – Resultados para as instâncias reais

## 7 Conclusões e Trabalhos Futuros

Neste trabalho definimos e apresentamos a complexidade de problemas de formação de equipes de trabalho. Definimos matematicamente o problema de formação de equipes de trabalho com restrições de habilidades profissionais e mostramos que este pode ser resolvido em tempo polinomial. Definimos também o problema de formação de equipes de trabalho com restrições de relacionamentos entre indivíduos e mostramos que o mesmo é NP-Difícil. Utilizando conceitos da Teoria dos Sistemas Sociotécnicos, definimos matematicamente o problema de formação de equipes de trabalho que considera simultaneamente dados referentes às habilidades profissionais dos indivíduos participantes de uma organização, assim como dados da rede social desta organização (relacionamentos entre os indivíduos participantes da organização). Denominamos este problema por *FEST* (Problema de formação de equipes sociotécnicas) e mostramos que também é um problema NP-Difícil, fato que motivou o estudo de formulações matemáticas, técnicas de programação inteira e algoritmo heurístico para sua resolução.

Para *FEST*, propomos uma formulação matemática com um número cúbico de variáveis e restrições, denominada Formulação 1, e apresentamos uma prova de sua corretude. Apresentamos também uma segunda formulação matemática, denominada Formulação 2, que foi proposta por (MARENCO; DELLE, 2013). Essa segunda formulação possui um número quadrático de variáveis, mas um número exponencial de restrições. Mostramos a corretude dessa formulação e que suas restrições podem ser separadas em tempo polinomial, onde utilizamos um algoritmo de fluxo máximo para verificação das restrições violadas.

Realizamos um estudo do politopo associado à Formulação 1, demonstramos três classes de desigualdades indutoras de facetas e duas outras classes de desigualdades válidas que fortalecem o modelo. As hipóteses utilizadas para demonstração das facetas são suficientes, mas não necessárias (podem ser enfraquecidas). Isto acontece, por exemplo, se cada indivíduo possui todas as habilidades requeridas pela organização. Procuramos uniformizar as hipóteses consideradas, relacionando-as à condição de Hall, que garante que o politopo seja não-vazio. Pudemos constatar que, à medida que a condição de Hall é satisfeita com maior folga (i.e. hipótese  $H_\epsilon$  válida para maiores valores de  $\epsilon$ ), mais desigualdades válidas das classes apresentadas passam a definir facetas. Vale destacar ainda que, mesmo as duas classes que não foram demonstradas ser indutoras de facetas, podem sê-lo sob condições específicas.

Com a Formulação 2, realizamos um estudo mais conciso. Procuramos principalmente relacioná-la com a Formulação 1. As relações estabelecidas permitiram, por exemplo, adaptar para a Formulação 2 algumas das desigualdades válidas para a Formulação 1. Também como consequência dessas relações, mostramos que as relaxações lineares de

ambas as formulações fornecem o mesmo limite inferior, que, em alguns casos, pode ser muito distante do valor ótimo.

Para efetuarmos testes computacionais com as duas formulações apresentadas, fizemos uso de instâncias reais criadas a partir de um sistema de informação e de instâncias aleatórias criadas a partir de um gerador pseudo-aleatório que propomos. Procuramos recriar instâncias-teste condizentes com as instâncias reais obtidas.

Os primeiros testes realizados mostraram que ambas as formulações apresentam dificuldades para solucionar um grupo de instâncias-teste criadas através do gerador pseudo-aleatório. O estudo realizado através deste primeiro teste computacional mostrou que a dificuldade de resolução de uma instâncias-teste do *FEST*, por ambas as formulações, é influenciada diretamente pela quantidade de tipos de arestas existentes no grafo de afinidades  $G$  e pela estrutura da matriz  $|V(G)| \times |S|$ , que marca o conjunto de habilidades  $s(u) \subseteq S$  de cada indivíduo  $u \in V(G)$ .

Verificamos que, quanto maior a quantidade de arestas com pesos maiores (menores níveis de afinidade entre indivíduos), maior é a complexidade para resolução da instância-teste criada. Note, por exemplo, que os pesos menores ficam próximos de zero (senão zero) quando fazemos a redução dos pesos da instância. Observamos ainda que a dificuldade de resolução de uma instância cresce com uma maior concentração de arestas com pesos idênticos, ou seja, quando os relacionamentos indicados pelos indivíduos de uma organização estão concentrados em apenas uma faixa do intervalo de tipos de arestas existentes. Isto deve-se ao fato de que a ocorrência de muitas arestas com mesmo peso pode aumentar a probabilidade de existência de várias soluções diferentes com mesmo valor.

Já a esparsidade da matriz  $|V| \times |S|$  afeta a complexidade da resolução de uma instância-teste através das formulação 1 e 2 de maneiras diferentes. Se a matriz  $|V| \times |S|$  é densa, as duas formulações apresentam mais dificuldade para encontrar a solução ótima. Este fato ocorre porque uma matriz  $|V| \times |S|$  densa pode aumentar a probabilidade de existência de várias soluções diferentes de mesmo valor para uma instância-teste. Porém, caso a esparsidade da matriz  $|V| \times |S|$  seja grande, embora a probabilidade de existência de várias soluções ótimas seja menor, a Formulação 2 apresenta dificuldades também para resolução de instâncias-teste com este tipo de matriz. Isto ocorre porque a Formulação 2' não estabelece a priori, para cada indivíduo  $u$  pertencente a cada equipe  $j$  criada, qual a habilidade  $s \in s(u, j)$  que este indivíduo desempenhará. Esta verificação é feita através do algoritmo de separação apresentado na seção 5.3.1. Sendo assim, quando a esparsidade da matriz  $|V| \times |S|$  é alta, a quantidade de verificações realizadas pelo algoritmo de separação pode aumentar de maneira significativa.

Os experimentos computacionais com as formulações mostraram que a Formulação A1 fortalecida é superior a Formulação A1, A2, e A2 fortalecida para a resolução das instâncias consideradas, sendo que a inclusão das desigualdades válidas foi fundamental.

De um total de 120 instâncias aleatórias testadas, a Formulação A1 fortalecida conseguiu resolver 96 instâncias contra 78 resolvidas pela Formulação A1. Já a Formulação A2, mesmo com o fortalecimento, resolveu apenas 57 delas. Na verdade, o fortalecimento da Formulação A1 propiciou uma significativa redução nos tempos de computação, conforme teste estatístico realizado.

Embora estes experimentos tenham considerado um tempo limite de 5000s para cada execução, verificamos que a extensão desse limite não mudaria o resultado acima descrito, a não ser para a Formulação A1 fortalecida, que conseguiria resolver otimamente mais 5 instâncias. Em todas as outras situações haveria escassez de memória.

Além das formulações, propomos, implementamos e testamos um algoritmo heurístico baseado no *Simulated Annealing* Não-Monotônico com busca em vizinhança que usa operadores tipo *swap*. Tal heurística mostrou-se bastante eficiente para as 120 instâncias-teste, tendo resolvido 118 delas na otimalidade em tempo computacional inferior a 1000 segundos. O limite superior e a solução viável fornecida pela heurística foram depois utilizadas na inicialização das Formulações A1, A1 fortalecida, A2 e A2 fortalecida. Esta informação não foi de grande ajuda para as formulações A1, A2 e A2 fortalecida. Já a Formulação A1 fortalecida pode solucionar otimamente as outras 24 instâncias restantes que não haviam sido solucionadas, porém em tempo superior a 5000s.

Como citado anteriormente as relaxações lineares de ambas formulações, em alguns casos, fornecem um limite inferior muito distante do valor ótimo, indicando que mesmo as versões fortalecidas ainda podem encontrar dificuldades na resolução de instância-teste complexas. No intuito de fortalecer ainda mais essas formulações, espera-se em trabalhos futuros explorar de melhor forma as desigualdades apresentadas neste trabalho, criando algoritmos de separação eficientes para as classes de desigualdades que fornecem um grande número de restrições, assim como estudar e propor novas desigualdades válidas que envolvam principalmente as variáveis  $X$ . Outra possível proposta para melhoria da Formulação A2 seria a adição de restrições (5.14) para alguns conjuntos  $S' \in S$  inicialmente no modelo, possivelmente reduzindo o número de iterações do algoritmo de planos de corte. Note que, sendo ruim o limite da relaxação linear, o número de resoluções de subproblemas com as variáveis inteiras pode ficar proibitivo. Pretende-se também gerar novas instâncias-teste complexas e de maior porte para a realização de experimentos computacionais para testar as possíveis novas desigualdades válidas que serão adicionadas às formulações e para avaliar o desempenho da heurística.

# Referências

- AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. *Network Flows: Theory, Algorithms, and Applications*. [S.l.]: Prentice Hall, 1993.
- BARTLETT, M. S. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, The Royal Society, v. 160, n. 901, p. 268–282, 1937.
- BONDY, J. A.; MURTY, U. S. R. *Graph theory with applications*. [S.l.]: Macmillan London, 1976.
- BOX, G. E.; COX, D. R. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 211–252, 1964.
- EDMONDS, J.; KARP, R. M. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, ACM, v. 19, n. 2, p. 248–264, 1972.
- EIBEN, A.; RUTTKAY, Z. Self-adaptivity for constraint satisfaction: learning penalty functions. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. [S.l.: s.n.], 1996. p. 258–261.
- FIGUEIREDO, T. F. *Adaptação do Algoritmo de Subida de Encosta para o Problema de Formação de Equipes de Trabalho Cooperativo*. 51 f. Monografia (Graduação) — Universidade Federal De Alfenas - UNIFAL, Minas Gerais, 2011. Monografia.
- GOLDBERG, A. V.; TARJAN, R. E. A new approach to the maximum-flow problem. *J. ACM*, v. 35, n. 4, p. 921–940, out. 1988.
- GOLDBERG, A. V.; TARJAN, R. E. Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research*, INFORMS, v. 15, n. 3, p. 430–466, 1990.
- HSU, J. *Multiple comparisons: theory and methods*. [S.l.]: CRC Press, 1996.
- KIRKPATRICK, D. L. *Evaluating training programs*. [S.l.]: Tata McGraw-Hill Education, 1975.
- MARENCO, J.; DELLE, D. D. Team formation with social-technical constraints. 2013.
- MARTELETO, R. M. Análise de redes sociais: aplicações nos estudos de transferência de informação. *Ciência da Informação*, v. 30, n. 1, p. 71–81, 2001.
- MORENO, J. L. *Psicodrama*. [S.l.]: Editora Cultrix, 1975.
- NEMHAUSER, G. L.; WOLSEY, L. A. *Integer and combinatorial optimization*. [S.l.]: Wiley New York, 1988.
- OLIVEIRA, H. C. B. et al. Clusterização em redes sociais através do simulated annealing não monotônico. In: *Anais do Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2010. p. 1996–2005.

PASMORE, W. et al. Sociotechnical systems: A north american reflection on empirical studies of the seventies. *Human relations*, Sage Publications, v. 35, n. 12, p. 1179–1204, 1982.

ROBSON, J. M. Algorithms for maximum independent sets. *Journal of Algorithms*, Elsevier, v. 7, n. 3, p. 425–440, 1986.

TRIST, E. The evolution of socio-technical systems. *Occasional paper*, v. 2, 1981.

WELLINS, R. S.; BYHAM, W. C.; DIXON, G. R. *Equipes Zapp! na prática: como vinte organizações classe mundial estão vencendo através das equipes*. [S.l.]: Ed. Campos, 1995.