

BRUNO TAKANE

**UM ALGORITMO EXATO PARA O PROBLEMA DA  
DIVERSIDADE MÁXIMA**

Belo Horizonte  
07 de agosto de 2011

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**UM ALGORITMO EXATO PARA O PROBLEMA DA  
DIVERSIDADE MÁXIMA**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção.

BRUNO TAKANE

Belo Horizonte  
07 de agosto de 2011



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Um Algoritmo Exato para o Problema da Diversidade  
Máxima

BRUNO TAKANE

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. GILBERTO DE MIRANDA JÚNIOR – Orientador  
Universidade Federal de Minas Gerais

Ph. D. RICARDO SARAIVA DE CAMARGO – Co-orientador  
Universidade Federal de Minas Gerais

Belo Horizonte, 07 de agosto de 2011

# Resumo

O termo diversidade está relacionado à variedade de características, idéias ou elementos diferentes entre si dentro de um determinado contexto, sendo importante para o pluralismo, heterogeneidade, tolerância mútua e sobrevivência de idéias. Existem diversos tipos de diversidade em diferentes áreas do conhecimento humano. Entre eles, podemos citar a diversidade religiosa, social, linguística, sexual, cultural e biológica. Na área de otimização combinatória, o Problema da Diversidade Máxima (PDM) consiste em selecionar um subconjunto de  $m$  elementos de um conjunto de  $n$  elementos, de tal forma que a diversidade entre os seus elementos selecionados seja máxima.

Neste trabalho é apresentado uma nova formulação para este problema baseado na Técnica de Reformulação de Linearização. Devido às características da formulação proposta e da dificuldade de resolução, o método de Decomposição de Benders Revisado é aplicado ao problema, assim como uma técnica de pré-processamento de modo a acelerar a sua convergência.

Testes são realizados para avaliar o desempenho do método aplicado ao problema e em seguida, uma análise é feita comparando-o com outro algoritmo descrito na literatura. Os resultados computacionais mostram que o método proposto demonstra ser competitivo frente aos métodos exatos descritos na literatura.

**Palavras-chaves:** Problema da Diversidade Máxima; Método de Decomposição de Benders; Técnica de Reformulação de Linearização.

# Abstract

The term diversity is related to the variety of features, ideas or different elements among them within a given context, being important for the pluralism, heterogeneity, mutual tolerance and survival of ideas. There are several types of diversity in different areas of human knowledge. Among them, we can mention religious, social, linguistic, sexual, cultural and biological diversity.

In the context of combinatorial optimization, the Maximum Diversity Problem (MDP) consists of selecting a subset of  $m$  elements from a set of  $n$  elements in such a way that the diversity among the selected elements is maximized.

A new model for this problem is presented in this work based on the Reformulation-Linearization-Technique. Due to the characteristics of the proposed formulation and the difficulty of this resolution, the Revised Benders Decomposition Method is applied to the problem and a pre-processing technique is used in order to accelerate its convergence.

Tests are performed to evaluate the performance of the method applied to the problem and then an analysis is done comparing it with another algorithm described in the literature. The computational results show that the presented method shows to be competitive with the exact methods described in the literature.

Keywords: Maximum Diversity Problem; Benders Decomposition; Reformulation-Linearization-Technique;

*Aos meus pais Akihide e Isaura, verdadeiros exemplos de vida, e aos meus irmãos, Fábio e Érica .*

# Agradecimentos

Eu gostaria de agradecer ao meu orientador Prof. Gilberto de Miranda Júnior pelo compartilhamento de sua sabedoria, motivação nos momentos mais difíceis e conselhos que contribuíram tanto para o meu desenvolvimento.

Ao meu co-orientador Prof. Ricardo Saraiva de Camargo por sua paciência, por seu incentivo e por ter acreditado e ajudado a desenvolver o meu trabalho.

Aos meus pais por terem sempre me incentivado aos estudos e por tudo que fizeram e fazem por mim.

Aos professores, funcionários, colegas e amigos do Departamento de Engenharia de Produção.

À CAPES pelo imprescindível apoio financeiro que possibilitou a dedicação integral ao mestrado.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Revisão da literatura</b>	<b>4</b>
2.1	Definição do Problema da Diversidade Máxima . . . . .	4
2.2	Exemplo . . . . .	5
2.3	Trabalhos relacionados . . . . .	7
2.3.1	Heurísticas e Metaheurísticas . . . . .	8
2.4	Formulação Matemática . . . . .	12
2.4.1	Outros problemas relacionados ao <i>PDM</i> na literatura . . . . .	14
2.5	Técnica de Reformulação de Linearização (TRL) . . . . .	17
2.5.1	Exemplo de aplicação da TRL . . . . .	17
2.6	Decomposição de Benders Revisado . . . . .	19
<b>3</b>	<b>Algoritmo proposto</b>	<b>22</b>
3.1	Aplicação da técnica TRL . . . . .	22
3.2	Aplicação do método de Decomposição de Benders Revisado . . . . .	25
3.3	Proposição de uma solução viável para o modelo . . . . .	28
3.4	Pré-processamento do método de Decomposição de Benders Revisado . . . . .	29
3.5	Fixação de variáveis . . . . .	29
3.6	Método exato para solução do PDM . . . . .	30
<b>4</b>	<b>Resultados computacionais</b>	<b>32</b>
<b>5</b>	<b>Conclusão</b>	<b>42</b>
	<b>Referências Bibliográficas</b>	<b>43</b>

# Lista de Figuras

2.1	Região viável de $X$ . . . . .	18
2.2	Envoltória convexa de $X$ . . . . .	18

# Lista de Tabelas

2.1	Mapeamento de atributos pessoais para valores numéricos . . . . .	6
2.2	Tabela de atributos mapeados dos candidatos à representação do Brasil no evento cultural . . . . .	6
2.3	Matriz de distâncias entre cada par de indivíduos . . . . .	7
2.4	Resultados computacionais de <a href="#">Martí et al. (2010)</a> . . . . .	15
4.1	Resultados de pré-processamento para as instâncias <b>GKD</b> propostas por <a href="#">Martí et al. (2010)</a> sem tempo limite como critério de parada . . . . .	33
4.2	Resultados de pré-processamento para as instâncias <b>SOM</b> propostas por <a href="#">Silva et al. (2004)</a> , considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada . . . . .	34
4.3	Comparativo entre o algoritmo <i>Branch-and-Bound</i> <b>BB</b> ( <a href="#">Martí et al. (2010)</a> ) e o algoritmo <b>DBR</b> proposto para as instâncias <b>GKD</b> por <a href="#">Martí et al. (2010)</a> , considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada . . . . .	36
4.4	Comparativo entre o algoritmo <i>Branch-and-Bound</i> <b>BB</b> ( <a href="#">Martí et al. (2010)</a> ) e o algoritmo <b>DBR</b> proposto para as instâncias <b>SOM</b> por <a href="#">Silva et al. (2004)</a> , considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada . . . . .	37
4.5	Tempo computacional gasto para o algoritmo <b>DBR</b> proposto atingir a otimalidade das instâncias <b>GKD</b> propostas por <a href="#">Martí et al. (2010)</a> , sem tempo limite como critério de parada . . . . .	38
4.6	Tempo computacional gasto para a Heurística <i>GRASP</i> proposta por <a href="#">Silva et al. (2004)</a> calcular uma solução viável, assim como o <b>GAP</b> de otimalidade atingido por este para as instâncias de <b>GKD</b> propostas por <a href="#">Martí et al. (2010)</a> . . . . .	39
4.7	Tempo computacional gasto para a Heurística <i>GRASP</i> proposta por <a href="#">Silva et al. (2004)</a> calcular uma solução viável, assim como o <b>GAP</b> de otimalidade atingido por este para as instâncias de <b>SOM</b> propostas por <a href="#">Silva et al. (2004)</a> . . . . .	40

4.8	Média do percentual de variáveis $x$ e $y$ fixadas através do algoritmo de fixação proposto na seção (seção 3.5) para as instâncias <b>GKD</b> propostas por <a href="#">Martí et al. (2010)</a> . . . . .	41
4.9	Média do percentual de variáveis $x$ e $y$ fixadas através do algoritmo de fixação proposto na seção (seção 3.5) para as instâncias <b>SOM</b> propostas por <a href="#">Silva et al. (2004)</a> . . . . .	41

# Lista de Algoritmos

1	Método de Decomposição de Benders Revisado . . . . .	21
2	Cálculo do valor de cada variável dual . . . . .	27
3	Método de Decomposição de Benders Revisado aplicado ao PDM . . .	31

# Capítulo 1

## Introdução

Diversidade está relacionada à variedade de características (idade, sexo e religião), idéias (pontos de vista, ângulos de abordagem) ou elementos (grupo de estudantes de uma universidade) diferentes entre si dentro de um determinado contexto.

Dependendo do contexto, ela é importante para o pluralismo, heterogeneidade, tolerância mútua e sobrevivência de idéias. Sua carência pode ocasionar diminuição de bens intangíveis, prejuízo financeiro ou até extinção. Por exemplo, na engenharia genética, ela é indispensável para a sobrevivência ou extinção de uma espécie. Pois quando a variação genética dentro da população é pequena, alguma alteração em seu ambiente (como o ataque de uma praga) pode levá-los à extinção por serem todos igualmente suscetíveis a tal mudança. Isso já foi verificado na Bahia, no início dos anos 90, onde uma praga, chamada vassoura-de-bruxa, destruiu plantações inteiras de cacau, provocando uma redução da produção anual de 320,5 mil toneladas para 191,1 mil toneladas, fazendo com que a participação do Brasil no mercado internacional de 14,8% para 4%. Por outro lado, se os indivíduos possuírem genes diferentes, alguns deles provavelmente serão capazes de suportar a mudança e assim a população não se extinguirá. Já no contexto cultural, a diversidade permite o enriquecimento cultural, a troca de idéias, o respeito pela diferença e convívio em um mundo pluralista.

Existem, ainda, outros tipos de diversidade em diferentes áreas do conhecimento humano. Entre eles, podemos citar a diversidade religiosa, social, linguística, sexual e biológica.

Na área de otimização combinatória, o Problema da Diversidade Máxima (PDM) consiste em selecionar um subconjunto de  $m$  elementos de um conjunto de  $n$  elementos, de tal forma que a diversidade entre os seus elementos selecionados seja máxima. Neste problema, o termo diversidade é usualmente substituído por distância entre elementos, em que esta distância é adequada para cada aplicação específica.

O potencial de aplicações deste problema é enorme. Na imigração, poderia-se selecionar imigrantes para se ter uma maior diversidade étnica. Na biologia, a diversidade biológica é fundamental para cruzamentos seletivos com o objetivo de selecionar características desejáveis em animais, plantas e outros seres vivos. Um outro exemplo, seria a aplicação no tratamento médico e combate à doenças, de forma a combater o maior espectro de potenciais agentes causadores de doenças. Pode-se aplicá-lo para a seleção de um grupo de estudantes a representarem o Brasil em um evento cultural no exterior. É comum desejar-se que o grupo tenha as mais diversas características possíveis, como idade, sexo, cultura e região. Pois, desta forma, estaríamos disseminando uma maior variedade cultural presente no Brasil e portanto, teríamos uma melhor representatividade do nosso país.

Em função do potencial de aplicações do PDM e do seu potencial de benefícios econômicos, seria interessante ter-se a melhor seleção (solução ótima) de elementos para que possa justificar o investimento realizado.

Desse modo, o principal objetivo deste trabalho é apresentar uma formulação adequada e um método exato eficiente com geração de planos de cortes para a solução do PDM. De modo a comprovar a eficiência do algoritmo, testes serão realizados a partir de instâncias conhecidas na literatura.

No Capítulo 1 é apresentado uma introdução do trabalho, mostrando o conceito de diversidade, os seus tipos, descrição do PDM, suas aplicações e os objetivos do trabalho.

No Capítulo 2, uma revisão bibliográfica é feita, apresentando formalmente o PDM, abordando suas formulações matemáticas, os trabalhos relacionados e outros nomes dados ao problema na literatura. Além disso, é feita uma introdução sobre os métodos utilizados no trabalho: a Técnica de Reformulação de Linearização (*TRL*) e o método de Decomposição de Benders Revisado.

No Capítulo 3, é proposto uma nova formulação para solução do PDM, mostra-se como aplicar o método de Decomposição de Benders Revisado ao problema, assim como alternativas com o objetivo de acelerar o método ao obter boas soluções iniciais e redução da formulação e ao final é mostrado o algoritmo proposto.

No Capítulo 4, o algoritmo proposto é testado e avaliado. Apresentam-se os resultados e discutem-se os mesmos, mostrando a eficiência do algoritmo proposto no Capítulo 3.

O Capítulo 5 conclui o trabalho e apresenta algumas sugestões para trabalhos futuros.

# Capítulo 2

## Revisão da literatura

Neste Capítulo, uma revisão bibliográfica é feita, apresentando formalmente o PDM, abordando suas formulações matemáticas e outros problemas relacionados na literatura. Além disso, é feita uma introdução sobre os métodos utilizados no trabalho: a Decomposição de Benders, a TRL e uma forma de selecionar os cortes de Benders gerados.

### 2.1 Definição do Problema da Diversidade Máxima

Conforme discutido no Capítulo 1, o PDM consiste em selecionar um subconjunto  $M$  com  $m$  elementos de um conjunto  $N$ , onde  $N = \{1, \dots, n\}$ , de tal forma que a diversidade entre os seus elementos selecionados seja máxima. Para que o PDM tenha significado, deve-se assumir que  $2 \leq m \leq n$ .

No PDM, é comum representar a diversidade como uma distância  $d_{ij}$  entre dois elementos  $i, j \in N$ . Esta distância é calculada como uma métrica normalizada entre os atributos dos elementos de  $N$ . Então, sejam  $R$  o conjunto de atributos, onde  $R = \{1, \dots, r\}$  e  $a_{ik}$  o estado ou valor do atributo  $k \in R$  para o elemento  $i \in N$ , sendo que  $a_{ik}$  pode ser real ou inteiro. Uma distância muito usada (Ghosh (1996)) é dada pela norma-p:

$$d_{ij} = \sqrt[p]{\sum_{k \in R} (|a_{ik} - a_{jk}|)^p} \quad (2.1)$$

Em que a quantidade  $d_{ij}$  representa a distância, ou dessemelhança, entre os elementos  $i, j \in N$  usando uma determinada métrica, onde  $d_{ij} > 0$  para  $i \neq j$  e  $d_{ij} = 0$ , caso contrário. Deste modo, o valor da diversidade total  $\zeta(M)$  para um dado subconjunto  $M$ , pode ser medido como a soma das distâncias entre cada par distinto de



elementos e é dado por:

$$\zeta(M) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left[ \sqrt[p]{\sum_{k \in R} (|a_{ik} - a_{jk}|)^p} \right] \quad (2.2)$$

Como discutido em Kuo et al. (1993), uma métrica pode ser mais adequada que a outra, dependendo da aplicação. Se por exemplo, for considerada a distância euclidiana para o cálculo da diversidade  $d_{ij}$  dada pela equação 2.1, temos, então:

$$d_{ij} = \sqrt{\sum_{k \in R} (a_{ik} - a_{jk})^2} \quad (2.3)$$

E se for considerada distância euclidiana para o cálculo da diversidade total  $\zeta(M)$  dada pela equação 2.2 para um dado conjunto  $M$ , temos:

$$\zeta(M) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left[ \sqrt{\sum_{k \in R} (a_{ik} - a_{jk})^2} \right] \quad (2.4)$$

## 2.2 Exemplo

Para uma melhor compreensão de como aplicar o problema, um exemplo de aplicação é mostrado nesta seção.

Suponha que uma instituição governamental fictícia esteja selecionando três alunos de graduação para representarem o Brasil em um evento cultural no exterior. E que após uma pré-seleção, seguindo determinados critérios, tenha restado dez alunos candidatos à viagem. Foi decidido que as três pessoas escolhidas entre as dez restantes deveriam ter os mais diversos atributos possíveis, pois, assim, disseminaria-se uma maior variedade cultural presente no Brasil e portanto, teríamos uma melhor representatividade do nosso país.

Entre os dados contidos no cadastro de cada candidato, foram consideradas relevantes os seguintes atributos para avaliar a diversidade: sexo, idade, região, remuneração familiar e áreas de conhecimento. De forma a resolver este problema através do modelo de diversidade máxima, precisa-se calcular a matriz de distâncias  $d_{ij}$ . Para isso, é feito um mapeamento dos atributos para valores numéricos, conforme mostra a tabela 2.1.

De acordo com o mapeamento da tabela 2.1, o atributo sexo pode assumir dois estados: masculino e feminino, que por sua vez, ao serem mapeados, assumem os valores inteiros 1 e 2, respectivamente. O atributo idade, por sua vez, é dividido em 5 faixas etárias, assumindo valores de 1 a 5 ao serem mapeadas. As regiões também assumem valores de 1 a 5 após o mapeamento, assim como as áreas de

Tabela 2.1: Mapeamento de atributos pessoais para valores numéricos

Classificação	Atributos e Estados				
	Sexo	Idade	Região	Áreas de Conhecimento	Remuneração Familiar (salários mínimos)
1	Masculino	17 - 18	Sudeste	Ciências Exatas	Até 2
2	Feminino	19 - 20	Sul	Ciências Humanas	Entre 3 e 5
3		21 - 22	Nordeste	Ciências Sociais Aplicadas	Entre 5 e 10
4		23 - 24	Norte	Ciências da Saúde	Acima de 10
5		25 - 26	Centro-Oeste	Ciências Agrárias	

conhecimento. E por final, o atributo remuneração familiar foi dividido em 4 faixas, que ao serem mapeadas assumem valores de 1 até 4.

O mapeamento dos atributos dos candidatos finais à representação do Brasil no evento cultural é mostrado na tabela 2.2, em que cada linha representa cada indivíduo com seus respectivos atributos.

Tabela 2.2: Tabela de atributos mapeados dos candidatos à representação do Brasil no evento cultural

Indivíduo	Nome	Sexo	Idade	Região	Áreas de Conhecimento	Remuneração Familiar (salários mínimos)
1	Felipe	1	3	1	4	3
2	Amanda	2	2	3	5	2
3	Sabrina	2	1	2	1	4
4	Armando	1	5	3	3	2
5	Mariana	2	3	5	3	3
6	Leticia	2	4	4	1	2
7	Fernando	1	5	2	4	2
8	Victor	1	1	1	2	1
9	Afonso	1	2	4	5	1
10	Patricia	2	4	1	2	3

Para resolvermos o problema da instituição governamental através do PDM, define-se que  $N$  seja o conjunto de candidatos à representação do Brasil no evento cultural, ou seja  $|N| = n = 10$ . Como deseja-se escolher 3 candidatos do conjunto total  $N$ , então,  $m = 3$ . A matriz de distâncias  $d_{ij}$ , neste caso, é calculada utilizando a distância euclidiana e a partir da equação 2.3 e dos dados da tabela 2.2. Por exemplo, a distância euclidiana entre os indivíduos Armando e Leticia é  $d_{46} = \sqrt{(1-2)^2 + (5-4)^2 + (3-4)^2 + (3-1)^2 + (2-2)^2} = \sqrt{7} \approx 2.6458$ . Seguindo o mesmo raciocínio, a matriz de distância entre cada par de indivíduo é mostrado na tabela 2.3.

Desse modo, para a solução do PDM, deve-se escolher um subconjunto de três elementos cuja soma de suas distâncias fosse maximizada. Para este exemplo, o subconjunto ótimo  $S$  é composto pelos indivíduos 3, 7 e 9 com a diversidade  $\zeta(S)$  dada pela equação 2.4 igual a 14.9180. Portanto, os alunos com os atributos mais

Tabela 2.3: Matriz de distâncias entre cada par de indivíduos

	1	2	3	4	5	6	7	8	9	10
1	0	2.8284	4.0000	3.1623	4.2426	4.5826	2.4495	3.4641	3.8730	2.4495
2	2.8284	0	4.6904	3.7417	3.1623	4.5826	3.4641	4.0000	1.7321	4.2426
3	4.0000	4.6904	0	5.0990	4.2426	4.1231	5.4772	3.4641	5.5678	3.4641
4	3.1623	3.7417	5.0990	0	3.1623	2.6458	1.4142	4.6904	3.8730	2.8284
5	4.2426	3.1623	4.2426	3.1623	0	2.6458	4.0000	5.0990	3.3166	4.2426
6	4.5826	4.5826	4.1231	2.6458	2.6458	0	3.8730	4.5826	4.6904	3.3166
7	2.4495	3.4641	5.4772	1.4142	4.0000	3.8730	0	4.6904	3.8730	2.8284
8	3.4641	4.0000	3.4641	4.6904	5.0990	4.5826	4.6904	0	4.3589	3.7417
9	3.8730	1.7321	5.5678	3.8730	3.3166	4.6904	3.8730	4.3589	0	5.1962
10	2.4495	4.2426	3.4641	2.8284	4.2426	3.3166	2.8284	3.7417	5.1962	0

diversos possíveis, entre os 10 inicialmente selecionados, seriam os alunos Sabrina, Fernando e Afonso.

## 2.3 Trabalhos relacionados

[Glover et al. \(1977\)](#) são os primeiros a caracterizar o PDM como um problema de otimização em um contexto de recursos genéticos.

[Kuo et al. \(1993\)](#) discutem vários contextos em que o problema pode ser levantado, como planejamento de mercado, seleção de portfólio e formação de comitê. Eles mostraram que o problema é NP-difícil e apresentaram duas definições diferentes para o problema utilizando formulações de programação linear inteira mista, os modelos *max-sum* e *max-min*. A primeira tem como objetivo maximizar a soma das diversidades de um subconjunto de elementos selecionados a partir de um conjunto maior. Enquanto que a última procura selecionar um subconjunto de elementos de um conjunto maior, de modo que a distância mínima entre os elementos selecionados seja maximizada. Nesta dissertação será abordada a definição de *max-sum* e por isso, será feita uma revisão de literatura seguindo apenas essa definição.

[Aringhieri et al. \(2008\)](#) citam, ainda, a aplicação durante a formação de equipes de trabalho, júris e grupos de estudantes para trabalho em projetos, pois é comum desejar um número fixo de indivíduos cujas características são tanto diversificadas quanto possível: equipes de trabalho devem incluir a maior abrangência de habilidades, júris devem representar a maior variedade de pontos de vista existentes em uma comunidade, grupos de trabalho devem permitir compartilhar e trocar diferentes formações. Além disso, a diversidade de força de trabalho nas organizações pode aumentar a sua eficiência, através de melhoras em tomada de decisão, soluções

de problemas, maior criatividade e melhor qualidade de gerenciamento (Fernandez (1991),Cox (1993)).

Entre outras aplicações, incluem-se, ainda: em genética animal e vegetal para obter novas variedades por reprodução controlada e com qualidades desejáveis (Porter et al., 1975); estímulo de diversidade étnica entre imigrantes (McConnell, 1988); preservação de diversidade biológica (Glover et al., 1995); equilíbrio ambiental, desenho de produto, gerenciamento de força de trabalho e engenharia genética (Glover et al., 1998);

### 2.3.1 Heurísticas e Metaheurísticas

Desde Kuo et al. (1993), diferentes métodos heurísticos e exatos têm sido propostos para solução do PDM. Glover et al. (1998) propõem duas heurísticas construtivas e outras duas destrutivas para o PDM. Eles fazem uma comparação entre os resultados aproximados de suas heurísticas com as soluções ótimas do problema para instâncias de até  $n = 30$  e mostram que os seus resultados distanciam de até 2% do ótimo.

Estudos recentes mostram que as metaheurísticas têm sido aplicados com sucesso para o PDM. Ghosh (1996) é o primeiro a propor um procedimento GRASP para o PDM. As soluções encontradas desse procedimento são comparadas com as soluções ótimas e ele mostra que consegue obter soluções ótimas para instâncias de tamanho  $n = 30$  e  $m = 6$  e de  $n = 25$  e  $m = 5$  em uma máquina SPARCstation 2 e sistema operacional SunOS 4.1.1, com uma média de 413,85 e 32,17 segundos, respectivamente.

Um outro GRASP foi desenvolvido por Andrade et al. (2003), conseguindo obter soluções melhores que Ghosh (1996) para instâncias de tamanho de até  $n = 250$  e  $m = 100$  em um tempo de até 3500 segundos.

Silva et al. (2004) desenvolvem três heurísticas construtivas e um algoritmo de Busca de Vizinhança Variável. O algoritmo GRASP proposto por eles combina uma entre as três heurísticas construtivas e o algoritmo de busca local proposto por Ghosh (1996) ou o algoritmo de Busca de Vizinhança Variável proposto. Eles fizeram uma comparação do algoritmo GRASP proposto com os apresentados por Ghosh (1996) e Andrade et al. (2003). Os resultados mostram que o algoritmo que combina uma heurística construtiva proposta e o algoritmo de busca local proposto por Ghosh (1996) demonstrou ser o mais eficiente entre as heurísticas propostas, conseguindo resolver instâncias de tamanho de  $n = 500$  e  $m = 200$  com uma média de aproximadamente 11 horas de tempo computacional em uma máquina PC AMD Athlon 1.4 GHz com 256 Mb de memória RAM. Enquanto que o algoritmo que com-

bina uma heurística construtiva e o algoritmo de busca local proposto por Ghosh (1996) foi o que achou melhores soluções para um maior número de instâncias, demorando mais de 23 horas para a instância de tamanho  $n = 500$  e  $m = 200$ .

Um algoritmo de Busca de Vizinhança Variável é proposto por Silva et al. (2004), conseguindo resolver instâncias de até 500 elementos e  $m = 200$  em um tempo aproximado de 27 horas.

Duarte e Martí (2007) propõem dois novos métodos construtivos para o PDM e dois procedimentos GRASP. Uma comparação com os métodos propostos por Glover et al. (1998) e Silva et al. (2004) é feita, considerando que os experimentos de todos os métodos são rodados em 10 segundos. Os seus resultados mostram que o algoritmo proposto consegue obter resultados melhores e um melhor desvio médio percentual da melhor solução encontrada durante cada experimento para instâncias de tamanho de até  $n = 500$  e  $m = 50$  em uma máquina *Pentium IV 3 GHz* com 1 GB de memória RAM.

Outros dois métodos GRASP são propostos por Silva et al. (2007) utilizando diferentes métodos construtivos e incluindo a técnica de *path-relinking*. Em um total de 80 testes de instâncias propostas por Silva et al. (2004), o segundo método proposto obtém 31 melhores resultados médios, enquanto que o método KLD obtém 25 e o algoritmo proposto por Ghosh (1996) obtém somente 7. O primeiro método executou mais rapidamente em 71 dos 80 testes, enquanto que o segundo método alcançou soluções de melhor qualidade, mas em compensação exigiu maior tempo computacional que os outros dois métodos. Como o primeiro método apresenta maior probabilidade de achar boas soluções para um pequeno intervalo de tempo em relação ao segundo método, é introduzido a técnica de *path-relinking* para o primeiro método de forma a melhorar a *trade-off* entre qualidade da solução e tempo computacional. Este método com a técnica de *path-relinking* mostrou obter melhores soluções do que o primeiro método proposto e os métodos GRASP apresentados por Silva et al. (2004), mas também demandaram maior tempo computacional. Estes métodos são testados para instâncias de tamanho de até  $n = 500$  e  $m = 200$  com média de 11 horas de tempo computacional para o tamanho máximo testado em uma máquina *PC AMD Athlon 1.4 GHz* com 256 Mb de memória RAM.

Kochenberger e Glover (2006) propõem um algoritmo de *Busca Tabu* para solução do PDM. Testes computacionais são realizados para instâncias de tamanho de até  $n = 1000$  e  $m = 300$ , mas não são comparados com qualquer outro método da literatura.

Duarte e Martí (2007) propõem dois métodos de *Busca Tabu* e dois métodos GRASP com memória de curto-prazo combinado com um método construtivo proposto por eles mesmos. Considerando que os experimentos de todos os métodos

são rodados em 10 segundos, os testes computacionais mostram que os algoritmos *TabuC2* e *TabuD2* encontram soluções melhores que os os métodos propostos por [Glover et al. \(1998\)](#) e [Silva et al. \(2004\)](#) para instâncias de tamanho  $n = 500$  e  $m = 200$  e  $n = 2000$  e  $m = 200$ . Além disso, o método *TabuD2* teve melhor desvio médio percentual da melhor solução encontrada em relação aos outros métodos. Os testes são realizados em uma máquina *Pentium IV 3 GHz* com 1 GB de memória RAM.

Uma outra heurística construtiva é proposta por [Palubeckis \(2007\)](#). Eles aproveitam a solução retornada por esta heurística em um algoritmo de *Busca Tabu* iterativo proposto por eles que alterna entre *Busca Tabu* e procedimentos de perturbação. O algoritmo proposto consegue encontrar soluções melhores que os métodos propostos por [Duarte e Martí \(2007\)](#) e [Silva et al. \(2007\)](#) para instâncias de tamanho de até 2000 e  $m = 200$  para um intervalo de 20 segundos, utilizando um *notebook Pentium M 1733 MHz*.

[Aringhieri et al. \(2008\)](#) propõem um algoritmo de *Busca Tabu* com funções de memória de curto e longo prazo. Eles fizeram uma comparação entre o algoritmos proposto e o algoritmo e instâncias propostos por [Silva et al. \(2004\)](#). Testes computacionais mostraram melhores soluções para instâncias de tamanho de até  $n = 500$  e  $m = 200$  em até aproximadamente 10 minutos, utilizando um *Pentium IV Mobile 2.8 Ghz* com 512 MB de memória RAM.

[Aringhieri e Cordone \(2006\)](#) propõem um outro algoritmo de *Busca Tabu*, um algoritmo de Busca de Vizinhança Variável e um algoritmo *Scatter Search*. Eles fizeram uma comparação entre os procedimentos *GRASP* propostos por [Andrade et al. \(2003\)](#), [Andrade et al. \(2005\)](#), [Santos et al. \(2005\)](#) e [Silva et al. \(2004\)](#) para as instâncias propostas por esta última. Entre esses algoritmos, todos os melhores resultados foram obtidos pelo algoritmo *GRASP* proposto por [Santos et al. \(2005\)](#), exceto para uma instância. Para 17 instâncias em um total de 20, o algoritmo de *Busca Tabu* proposto é igual aos melhores resultados reportados na literatura e é melhor para outras duas instâncias. Este algoritmo demonstra ter o melhor compromisso entre qualidade de solução e tempo computacional entre os algoritmos propostos. Ele resolve instâncias de tamanho de até  $n = 500$  e  $m = 200$  com até 627 segundos, enquanto que o algoritmo *GRASP* proposto por [Santos et al. \(2005\)](#) gasta entre 308 e 557432 segundos. O algoritmo *Scatter Search* proposto alcança os mesmos resultados que o algoritmo de *Busca Tabu* proposto, mas demanda maior esforço computacional, gastando entre 3 e 26000 segundos. Já o algoritmo de Busca de Vizinhança Variável proposto consegue melhores resultados da literatura para 9 instâncias. Os testes foram executados em um *Pentium IV Mobile 2.8 Ghz* com 512 MB de memória RAM.

[Gallego et al. \(2006\)](#) propõem uma variante do algoritmo *Scatter Search*. Eles conseguem obter soluções melhores e com menor desvio médio percentual da mel-



hor solução encontrada em relação aos algoritmos propostos por [Silva et al. \(2004\)](#) e [Duarte e Martí \(2007\)](#). Os testes são realizados para as instâncias propostas por [Glover et al. \(1998\)](#), [Silva et al. \(2004\)](#) e outras instâncias criadas pelos autores de tamanho até  $n = 2000$  e  $m = 200$ , considerando 30 segundos de execução de cada algoritmo e depois de 3 minutos de busca local, usando um *Pentium IV 3 GHz* com 3 GB de memória RAM.

Em comparação com os algoritmos aproximados, poucos métodos exatos foram propostos para o PDM. [Erkut e Neuman \(1991\)](#) propõem um algoritmo *branch and bound* capaz de resolver instâncias de até  $n = 50$  e  $m = 10$  com uma média de 5095 segundos em um computador AT 10 MHz.

[Cutler e Klastorin \(1997\)](#) apresentam duas heurísticas e um algoritmo *branch and bound* que são testados para instâncias de até  $n = 40$ .

De forma a tratar o problema de diversidade de grupos de trabalho, [Bhadury et al. \(2000\)](#) apresentam um modelo capaz de reduzir o problema a um de fluxo em redes e resolveu-o através de um algoritmo exato para uma base de dados real em um custo de pós-graduação.

[Pisinger \(2006\)](#) propõe limites superiores para o PDM (chamado de Problema de Dispersão Soma-p em [Pisinger \(2006\)](#)) baseados em relaxação Lagrangeana, programação semi-definida e técnicas de reformulação. Além disso, ele propõe um algoritmo *branch and bound* capaz de resolver instâncias de tamanho  $n = 90$  e  $m$  aleatório no intervalo  $[2, \dots, n - 2]$  em 3957 segundos em média. Os resultados são valores médios de 10 instâncias e são rodados em uma máquina AMD 64-bit 2.4 GHz. Ele também propõe um algoritmo de redução da instância, através de fixação de variáveis. Caso a variável  $x_i, i \in N$  for fixada em 0, a linha  $i$  e coluna  $i$  correspondente são retiradas da matriz de distância e  $n$  é, então, reduzido. É importante observar que ele considera  $d_{ii} \neq 0, i \in N$ .

O trabalho mais recente de [Martí et al. \(2010\)](#) propõe um algoritmo *branch and bound*. Considerando um limite de execução do algoritmo de 1 hora e instâncias de [Glover et al. \(1998\)](#), eles conseguem obter soluções ótimas para instâncias de tamanho  $n = 100$  e  $m = 10$  em 4,4 segundos, um GAP de otimalidade de 5.4% para instâncias de tamanho  $n = 150$  e  $m = 15$  em 1834,4 segundos e um GAP de otimalidade de 10.9% para instâncias de tamanho  $n = 150$  e  $m = 45$  em 3600 segundos. Já para as instâncias desenvolvidas por [Silva et al. \(2004\)](#), eles também conseguem obter soluções ótimas para instâncias de tamanho  $n = 100$  e  $m = 10$  em 38,3 segundos, um GAP de otimalidade de 26,7% para instâncias de tamanho  $n = 150$  e  $m = 15$  e um GAP de otimalidade de 35,6% para instâncias de tamanho  $n = 150$  e  $m = 45$ , ambas em 3600 segundos. Os testes são realizados em um *Pentium IV 3 GHz* com 3 GB de memória RAM.

## 2.4 Formulação Matemática

Kuo et al. (1993) apresentam três formulações inteiras para *PDM*, considerando a definição *max-sum*, citada na seção 2.3, que tem como objetivo maximizar a soma das diversidades de um subconjunto de elementos selecionados a partir de um conjunto maior. A primeira formulação **F1** modela o *PDM* como um programa inteiro quadrático 0-1. Uma vez que **F1** é não-linear, a segunda formulação **F2** o converte em um programa linear inteiro-misto. Para a terceira formulação **F3**, é feita uma transformação de variáveis em **F1** para transformá-lo em um problema com menos variáveis e menos restrições.

Considerando a definição do *PDM* discutida na seção 2.1 e seja  $x_i, i \in N$ , uma variável que toma o valor 1 se o elemento  $i$  é selecionado e 0 caso contrário. O problema de diversidade máxima pode ser modelado como um programa inteiro quadrático 0-1:

### Formulação F1

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j \quad (2.5)$$

$$\text{s.a: } \sum_{i \in N} x_i = m \quad (2.6)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N \quad (2.7)$$

No modelo, a função objetivo (2.5) maximiza a soma das diversidades entre os  $m$  elementos a serem selecionados do conjunto  $N$ . A restrição (2.6) garante que exatamente  $m$  elementos sejam selecionados do conjunto  $N$ . E as restrições (2.7) são de integralidade para as variáveis  $x_i, i \in N$ .

Uma vez que a não-linearidade de **F1** possa ser inconveniente, Kuo et al. (1993) utiliza uma abordagem mostrada em Glover e Wolsey (1974) para transformá-lo em um modelo linear 0 – 1, através da substituição do produto  $x_i x_j$  em uma nova variável  $y_{ij}$ , propondo a formulação **F2**.



### Formulação F2

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} y_{ij} \quad (2.8)$$

$$\text{s.a: } \sum_{i \in N} x_i = m \quad (2.9)$$

$$x_i + x_j - y_{ij} \leq 1, \quad \forall i, j \in N : i < j \quad (2.10)$$

$$-x_i + y_{ij} \leq 0, \quad \forall i, j \in N : i < j \quad (2.11)$$

$$-x_j + y_{ij} \leq 0, \quad \forall i, j \in N : i < j \quad (2.12)$$

$$y_{ij} \geq 0, \quad \forall i, j \in N : i < j \quad (2.13)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N \quad (2.14)$$

Neste modelo, a função objetivo (2.8) maximiza a soma das diversidades entre os  $m$  elementos a serem selecionados do conjunto  $N$ . A restrição (2.9) garante que exatamente  $m$  elementos sejam selecionados do conjunto  $N$ . As restrições (2.10) fazem com que as variáveis  $y_{ij}$  sejam igual a 1, sempre que  $x_i$  e  $x_j$  forem iguais a 1. As restrições (2.11) exigem que quando  $x_i$  pertencer à solução, então existirá algum  $j$ , tal que  $y_{ij} = 1, i, j \in N$ . Do mesmo modo, As restrições (2.12) exigem que quando  $x_j$  pertencer à solução, então existirá algum  $i$ , tal que  $y_{ij} = 1, i, j \in N$ . As restrições (2.13) restringem o domínio das variáveis  $y_{ij}$  a valores não-nulos e as restrições (2.14) são de integralidade para as variáveis  $x_i, i \in N$ .

Segundo Gallego et al. (2006), F2 produz uma relaxação linear fraca. Quando os valores  $d_{ij}$  são uniformemente distribuídos, a relaxação linear resulta em  $x_i = \frac{m}{n}$ , para todo  $i$ .

Para o PDM, Kuo et al. (1993) utilizam as desigualdades propostas por Glover (1975) para manipular programas inteiros quadráticos com variáveis reais e binárias. Para o PDM, sejam  $U_i = \sum_{j=i+1}^n \max(0, d_{ij})$  e  $L_i = \sum_{j=i+1}^n \min(0, d_{ij})$  e seja  $w_i, i \in N$ , uma variável tal que  $w_i = \sum_{j=i+1}^n d_{ij} x_j$ , se  $x_i = 1$  e  $w_i \leq 0$ , se  $x_i = 0$ , a formulação F1 pode ser reescrita da seguinte forma:

### Formulação F3

$$\max \sum_{i=1}^{n-1} w_i \quad (2.15)$$

$$\text{s.a: } \sum_{i \in N} x_i = m \quad (2.16)$$

$$- U_i x_i + w_i \leq 0, \quad \forall i = 1, \dots, n-1 \quad (2.17)$$

$$- \sum_{j=i+1}^n d_{ij} x_j + L_i * (1 - x_i) + w_i \leq 0, \quad \forall i = 1, \dots, n-1 \quad (2.18)$$

$$w_i \in \mathfrak{R}, \quad \forall i \in N \quad (2.19)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N \quad (2.20)$$

A restrição 2.15 garante que a função objetivo tenha o mesmo valor que a restrição 2.5. As restrições 2.17 asseguram que  $w_i \leq 0$ , quando  $x_i = 0$ . Enquanto que as restrições 2.18 garantem que  $w_i = \sum_{j=i+1}^n d_{ij} x_j$ , quando  $x_i = 1$ . As restrições 2.19 definem o domínio da variável  $w_i, i \in N$ . E as restrições (2.14) são de integralidade para as variáveis  $x_i, i \in N$ .

Ao fazer uma comparação com F1, a formulação F2 tem  $n(n-1)/2$  novas variáveis e  $3n(n-1)/2$  novas desigualdades. Enquanto que a formulação F3 possui apenas  $n-1$  variáveis novas e  $2(n-1)$  novas desigualdades.

Utilizando o *CPLEX 8.0* com o limite máximo de tempo igual a uma hora, Martí et al. (2010) fazem uma comparação entre as três formulações, F1, F2 e F3 para um conjunto de 75 instâncias propostas por Glover et al. (1998) de tamanho de até  $n = 30$  e  $m = 24$ . A tabela 2.4 mostra o *GAP* médio de otimalidade, tempo médio em segundos e o número de soluções ótimas alcançadas por cada formulação. Eles mostram que a formulação F3 tem melhores resultados em comparação com F1 e F2 e em um menor intervalo de tempo. Em que o *GAP* de otimalidade médio é calculado como o limite superior menos a melhor solução achada, ambos retornados pelo *CPLEX*, dividido pelo limite inferior e multiplicado por 100. O *GAP* de otimalidade médio, então, é calculado somando o *GAP* de otimalidade das 5 instâncias diferentes de cada tamanho e dividindo-se por 5.

#### 2.4.1 Outros problemas relacionados ao PDM na literatura

O PDM pode ser encontrado com outros nomes na literatura, considerando a definição de *max-sum* para o PDM. No trabalho de Kuby (1987), o PDM é chamado de Problema de Dispersão Maxisum que tem como objetivo maximizar a distância de sepa-

Tabela 2.4: Resultados computacionais de [Martí et al. \(2010\)](#)

Formulação	F1	F2	F3
<b>GAP médio (%)</b>	0.7	0.8	0.0
<b>Tempo médio (s)</b>	592.6	711.7	96.9
<b>Soluções ótimas</b>	65	66	75

ração média entre as facilidades já abertas. Ele propõe uma formulação inteira 0 – 1 capaz de resolver instâncias de tamanho  $n = 25$  e  $m = 9$  minutos e de tamanho  $n = 25$  e  $m = 10$  em 15 minutos, usando um computador *Boston University IBM 3090*.

Nos trabalhos de [Kincaid e Yellin \(1993\)](#) e [Pisinger \(2006\)](#), o PDM é tratado como Problema de Dispersão Soma- $p$  que é um problema de localização de  $p$  facilidades dado  $n$  locais pré-definidos, onde  $p \leq n$ , tal que a soma de distâncias entre as  $p$  facilidades seja maximizada.

Para alguns casos, trata-se de uma particularidade do problema proposto. Por exemplo, para o *Problema  $p$ -Maxiano* ([Erkut \(1990\)](#)), tem-se uma configuração com  $m$  facilidades já instaladas e o seu objetivo é maximizar a soma de distâncias entre os pares já existentes e as  $p$  novas facilidades mais a soma de distâncias entre os pares das  $p$  novas facilidades. Para o caso em que  $m$  é igual a zero, o problema se reduz ao PDM. Para este problema, considerando que nenhuma facilidade já estava instalada, [Erkut \(1990\)](#) foi capaz de resolver instâncias de tamanho  $n = 25$  e  $m = 10$  em menos de 45 segundos através de um algoritmo *branch-and-bound* em um computador *AT 10 MHz*.

O PDM pode ser tratado também como Problema de Soma- $p$ -Defesa ([Kincaid \(1992\)](#) e [Cappanera \(1999\)](#)), em que o objetivo é maximizar a soma de todas as distâncias entre todos os pares de facilidades a serem instaladas. [Kincaid \(1992\)](#) desenvolve um método de *Simulated Annealing* capaz de resolver instâncias de tamanho  $n = 33$  e  $m = 15$ , propostos por [Karg e Thompson \(1964\)](#), com média de 438 segundos e um algoritmo de *Busca Tabu* que resolve instâncias de mesmo tamanho com média de 80 segundos.

Outros nomes dados ao PDM são:

- Problema de Dispersão Max-AVG ([Ravi et al. \(1994\)](#)): consiste em selecionar facilidades, tal que se maximize a distância média entre as facilidades selecionadas. Eles propõem um algoritmo de aproximação para o problema, mas nenhum teste é realizado;
- Problema de Máxima Dispersão ([Hassin et al. \(1997\)](#) e [Fekete e Meijer \(2004\)](#)): prob-

lema de achar  $k$  vértices em um grafo completo com pesos não-negativos nos arcos. Seu objetivo é selecionar um dado número de locais de um conjunto discreto de candidatos, tal que a distância média entre os locais selecionados seja maximizada. [Hassin et al. \(1997\)](#) e [Fekete e Meijer \(2004\)](#) propõem algoritmos de aproximação para o problema com garantia de performance limitada.;

- Problema de  $k$ -subgrafo denso ([Feige et al. \(2001\)](#)): problema de maximização, em que deseja-se achar um subgrafo denso com  $k$  vértices, dado um grafo. Tem como entrada um grafo  $G$  e um parâmetro  $k$ , em que deseja-se achar um conjunto de  $k$  vértices com grau máximo médio no subgrafo induzido por este conjunto. [Feige et al. \(2001\)](#) propõem um algoritmo de aproximação para este problema;
- Problema de Clique-Remoto ([Chandra e Halldorsson \(2001\)](#) e [Birnbaur e Goldman \(2009\)](#)): consiste em achar um subconjunto de  $k$  vértices, formando um subgrafo induzido de peso máximo, dado um número inteiro positivo  $k$  e um grafo completo com pesos não-negativos nos arcos satisfazendo a desigualdade triangular. [Chandra e Halldorsson \(2001\)](#) introduzem diversos tipos de problemas, inclusive o Problema de Clique-Remoto, mas não chegam a apresentar um algoritmo para resolvê-lo. Enquanto que [Birnbaur e Goldman \(2009\)](#) propõem um algoritmo guloso simples para sua solução.;
- Problema do Subgrafo com máximo arco-ponderado ([Macambira \(2002\)](#)): tem como objetivo achar um subgrafo tal que a soma dos pesos associados aos arcos do subgrafo seja maximizada, sujeito a uma restrição de cardinalidade. Eles propõem um algoritmo de *Busca Tabu* capaz de resolver instâncias de tamanho  $n = 100$  elementos e  $m = 50$  com um tempo médio de aproximadamente 12 minutos, em um *Pentium III 600 MHz* com 256 *Mbytes* de memória RAM.

De modo a apresentar uma nova formulação e um método exato eficiente com geração de planos de cortes ao longo dos nós das árvores *Branch-and-Bound* para a solução do PDM, a Técnica de Reformulação de Linearização e o método de Decomposição de Benders Revisado serão descritos nas seções seguintes.

## 2.5 Técnica de Reformulação de Linearização (TRL)

A TRL gera relaxações de programação linear justas e desigualdades válidas fortes para uma larga classe de problemas de programação discreta combinatória e contínua não-convexa. Ela é capaz de gerar vários níveis hierárquicos de representações do problema original, aonde cada nível aumenta o número de variáveis e restrições, mas fortalece a relaxação de programação linear. O seu desenvolvimento originou em [Adams e Sherali \(1986\)](#), [Adams e Sherali \(1990\)](#) e [Adams e Sherali \(1993\)](#), focando inicialmente em problemas 0-1 e lineares mistos 0-1.

Dada uma região viável  $X$  de um problema linear inteiro misto 0-1 ou inteiro quadrático 0-1 descrito por desigualdades e igualdades usando variáveis binárias  $x = [x_1, \dots, x_s]$  e contínuas limitadas  $y = [y_1, \dots, y_w]$ , a TRL permite gerar uma região viável  $X_d$  de nível hierárquico  $d \in \{0, 1, \dots, s\}$ , tal que a relaxação linear  $\psi(X_d)$  é maior ou igual a do nível anterior ou  $\psi(X_d) \geq \psi(X_{d-1})$  e  $X_0 = X$ .

Para se gerar a região viável  $X_d$  do nível  $d$ , usa-se a região viável  $X_{d-1}$  onde cada igualdade é multiplicada pelas variáveis binárias  $x_j, \forall j \in (1, \dots, s)$ , e cada desigualdade, incluindo as restrições de domínio de variáveis são multiplicadas tanto pelas variáveis binárias  $x_j$  quanto por seus complementos  $(1 - x_j), \forall j \in (1, \dots, s)$ . Após usar a relação  $x_j^2 = x_j$  e  $x_j(1 - x_j) = 0$  para cada variável binária  $x_j, \forall j \in (1, \dots, s)$ , e substituindo cada termo não-linear  $y_i \prod_{j \in (1, \dots, s)} x_j, \forall i \in (1, \dots, w)$  por uma variável auxiliar  $v_{ij}$ , obtém-se então um conjunto poliedral de dimensão superior definido em termos das variáveis originais  $(x, y)$  e das novas variáveis  $v_{ij}$ . Dessa forma, denotando-se a projeção  $X_d$  no espaço de variáveis originais como  $X_{P_d}$ , [Adams e Sherali \(1986\)](#) demonstram que à medida que se aumenta  $d$ , têm-se que:

$$X_{P_0} \supseteq X_{P_1} \supseteq X_{P_2} \supseteq \dots \supseteq X_{P_n} = \text{conv}(X) \quad (2.21)$$

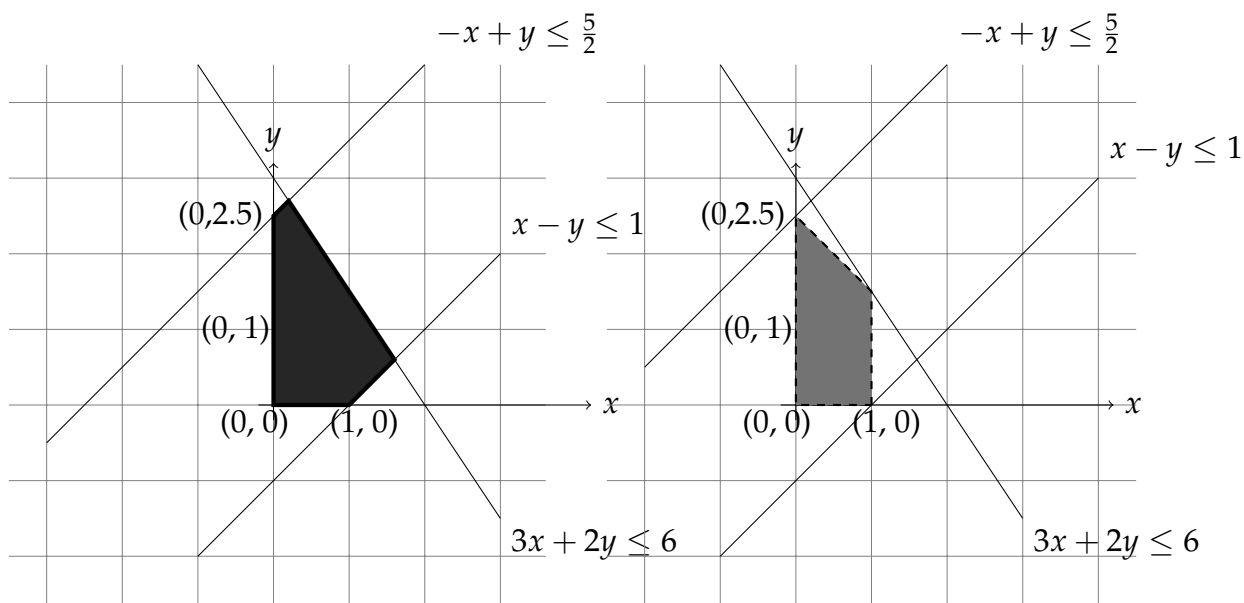
onde  $\text{conv}(X)$  representa a envoltória convexa do espaço de soluções viáveis  $X$ .

### 2.5.1 Exemplo de aplicação da TRL

Para uma melhor compreensão de como aplicar o TRL, um exemplo é mostrado nesta seção.

Considere um formulação inteira-mista 0-1 com a seguinte região viável  $X$ , ilustrada na figura 2.1.

$$X = \{(x, y) : 3x + 2y \leq 6, -x + y \leq \frac{5}{2}, x - y \leq 1, x \in \{0, 1\}, y \geq 0\} \quad (2.22)$$

Figura 2.1: Região viável de  $X$ Figura 2.2: Envoltória convexa de  $X$ 

Para este caso, têm-se que  $s = 1$  e, portanto, a relaxação  $X_1$  no nível  $d = 1$  gera a representação da envoltória convexa de  $X$ . Para construir  $X_1$ , multiplica-se as restrições de  $X$  por  $x$  e  $(1 - x)$  e usa-se a relação  $x^2 = x$ , além de fazer a substituição  $v = xy$ . Desta forma, encontra-se as seguintes restrições que representam o conjunto  $X_1$ :

$$3x + 2y \leq 6 = \begin{cases} *x & \Rightarrow v \leq \frac{3}{2}x & (2.23) \\ *(1-x) & \Rightarrow v \geq 3x + y - 3 & (2.24) \end{cases}$$

$$-x + y \leq \frac{5}{2} = \begin{cases} *x & \Rightarrow v \leq \frac{7}{2}x & (2.25) \\ *(1-x) & \Rightarrow v \geq \frac{5}{2}x + y - \frac{5}{2} & (2.26) \end{cases}$$

$$x - y \leq 1 = \begin{cases} *x & \Rightarrow v \geq 0 & (2.27) \\ *(1-x) & \Rightarrow v \leq -x + y + 1 & (2.28) \end{cases}$$

$$y \geq 0 = \begin{cases} *x & \Rightarrow v \geq 0 & (2.29) \\ *(1-x) & \Rightarrow v \leq y & (2.30) \end{cases}$$

Para verificar a projeção  $X_{P1}$  de  $X_1$  no espaço de variáveis  $(x, y)$ , reescreve-se  $X_1$  como:

$$X_1 = \{(x, y, v) : v \geq 3x + y - 3, v \geq \frac{5}{2}x + y - \frac{5}{2}, v \geq 0, \\ v \leq \frac{3}{2}x, v \leq \frac{7}{2}x, v \leq -x + y + 1, v \leq y\} \quad (2.31)$$

Ao usar a eliminação de *Fourier-Motzkin*, a equação 2.31 implica na projeção:

$$X_1 = \{(x, y) : \max \{3x + y - 3, \frac{5}{2}x + y - \frac{5}{2}, 0\} \leq \\ \min \{\frac{3}{2}x, \frac{7}{2}x, -x + y + 1, y\}\} \quad (2.32)$$

Ao reescrever as desigualdades equivalentes de 2.32 e retirando-se as restrições redundantes, obtém-se o conjunto 2.33 que descreve  $conv(X)$ , conforme mostra a figura 2.2, lembrando que a variável  $x$  é binária e a variável  $y$  é linear.

$$X_{P1} = X_{Ps} = \{(x, y) : 2x + 2y \leq 5, x - y \leq 1, 0 \leq x \leq 1, y \geq 0\} \quad (2.33)$$

## 2.6 Decomposição de Benders Revisado

Nesta seção é descrito o método de Decomposição de Benders Revisado proposto em Martin (1999) que difere em relação ao método original ao gerar cortes ao longo da árvore *Branch-and-Bound*. Ele é especializado para uma classe de problemas lineares que possuem uma formulação com uma relaxação linear (RL) fraca e uma formulação estendida com uma RL mais forte. O problema linear genérico sob consideração é dado por:

$$\min \mathbf{c}^T \mathbf{x} \quad (2.34)$$

$$\text{s.a: } \mathbf{Ax} \geq \mathbf{b} \quad (2.35)$$

$$\mathbf{x} \in \{0, 1\} \quad (2.36)$$

Considerando que a formulação 2.34 – 2.36 é uma formulação com um pequeno número de variáveis e com uma RL fraca, é possível obter uma formulação com uma RL melhor, mas com mais variáveis e restrições usando a técnica TRL, descrita na

seção 2.5, obtendo-se então uma nova família de desigualdades  $\mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{y} \geq \mathbf{f}$ :

$$\min \mathbf{c}^T \mathbf{x} \quad (2.37)$$

s.a: restrições 2.35 – 2.36 e:

$$\mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{y} \geq \mathbf{f} \quad (2.38)$$

$$\mathbf{y} \geq \mathbf{0} \quad (2.39)$$

Para qualquer  $\bar{\mathbf{x}} \in X$ , onde  $X = \{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \in 0,1\}$ , uma das duas afirmações seguintes é verdadeira:

1.  $\mathbf{D}\bar{\mathbf{x}} + \mathbf{E}\mathbf{y} \geq \mathbf{f}$

2.  $\mathbf{D}\bar{\mathbf{x}} + \mathbf{E}\mathbf{y} < \mathbf{f}$

Para o caso 1 implica que existe um  $\mathbf{y} \in Y$ , onde  $Y = \{\mathbf{y} : \mathbf{E}\mathbf{y} \geq \mathbf{f} - \mathbf{D}\bar{\mathbf{x}}\}$  é a região de viabilidade de  $\mathbf{y}$ . Já o caso 2 implica que não existe  $\mathbf{y} \in Y$  e pelo *Lema de Farkas*, pode-se achar um hiperplano que separa o ponto  $\bar{\mathbf{x}}$  da região  $Y$ . Uma forma de achá-lo é através da introdução de variáveis e que viabilizariam o subsistema  $\mathbf{E}\mathbf{y} < \mathbf{f} - \mathbf{D}\bar{\mathbf{x}}$  (variáveis de erro). Obtendo assim um subsistema definido por:

$$\min \mathbf{e} \quad (2.40)$$

$$\text{s.a: } \mathbf{E}\mathbf{y} + \mathbf{e} = \mathbf{f} - \mathbf{D}\bar{\mathbf{x}} \quad (2.41)$$

$$\mathbf{e} \geq \mathbf{0} \quad (2.42)$$

Associando variáveis duais  $\mathbf{u}$  às restrições 2.41, pode-se encontrar o hiperplano através do subproblema dual escrito como:

$$\max (\mathbf{f} - \mathbf{D}\bar{\mathbf{x}})^T \mathbf{u} \quad (2.43)$$

$$\text{s.a: } \mathbf{E}^T \mathbf{u} \leq \mathbf{0} \quad (2.44)$$

$$\mathbf{1}^T \mathbf{u} \leq 1 \quad (2.45)$$

$$\mathbf{u} \in \mathbb{R} \quad (2.46)$$

Para todo  $\bar{\mathbf{x}} \in X$ , tal que não existe  $\mathbf{y} \in Y$ , pode-se achar um raio extremo, tal que  $(\mathbf{f} - \mathbf{D}\bar{\mathbf{x}})^T \bar{\mathbf{u}} > 0$ . Este raio extremo provê um novo corte  $\bar{\mathbf{u}}^T \mathbf{D}\mathbf{x} \geq \bar{\mathbf{u}}^T \mathbf{f}$ , resultando no seguinte problema mestre **PM**:



**PM**

$$\min \mathbf{c}^T \mathbf{x} \quad (2.47)$$

$$\text{s.a: } \mathbf{Ax} \geq b \quad (2.48)$$

$$\bar{\mathbf{u}}^T \mathbf{Dx} \geq \bar{\mathbf{u}}^T \mathbf{f}, \quad \forall \bar{\mathbf{u}} \in \mathbf{U} \quad (2.49)$$

$$\mathbf{x} \in \{0, 1\} \quad (2.50)$$

onde  $\mathbf{U} = \{\mathbf{u} : \max(\mathbf{f} - \mathbf{D}\bar{\mathbf{x}})^T \mathbf{u}, \text{ s.a: } \mathbf{E}^T \leq \mathbf{0}, \text{ é ilimitado}\}$ .

O número de desigualdades 2.49 pode ser muito elevado, entretanto, apenas algumas delas estarão ativas na solução ótima, sugerindo, então, uma estratégia de relaxação, ou seja, adicionar as desigualdades sob demanda. Temos então o Problema Mestre Relaxado (**PMR**):

**PMR**

$$\min \mathbf{c}^T \mathbf{x} \quad (2.51)$$

$$\text{s.a: } \mathbf{Ax} \geq b \quad (2.52)$$

$$(\bar{\mathbf{u}}^h)^T \mathbf{Dx} \geq (\bar{\mathbf{u}}^h)^T \mathbf{f}, \quad \forall \bar{\mathbf{u}} \in \mathbf{U}, h = 1, \dots, H \quad (2.53)$$

$$\mathbf{x} \in \{0, 1\} \quad (2.54)$$

onde  $H$  é o número máximo de desigualdades necessárias para se obter a solução ótima, podendo ser geradas ao longo da árvore de *Branch-and-Bound*. Um algoritmo para o método de Decomposição de Benders Revisado aplicado ao PDM é descrito a seguir, onde  $\Phi(\mathbf{u})$  é o valor da função objetivo 2.43,  $\Omega$  é um conjunto de nós viáveis da árvore de *Branch-and-Bound* e  $\delta$  é um nó da árvore com as restrições de *branching* e as desigualdades 2.49:

---

**Algoritmo 1:** Método de Decomposição de Benders Revisado
 

---

Faça  $\Omega = \{\delta\}$ , onde  $\delta$  é o nó raiz do sistema (2.47) - (2.50).

**Enquanto**  $\Omega \neq \emptyset$  **faça**

  Selecione  $\delta^* \in \Omega$

$\Omega \leftarrow \Omega \setminus \{\delta^*\}$

**Faça**

    Resolva  $\delta^*$  obtendo  $\bar{\mathbf{x}}$  e  $\Phi(\bar{\mathbf{x}})$

    Resolva (2.43) - (2.46), obtendo  $\Phi(\mathbf{u})$ .

    Se  $\Phi(\mathbf{u}) > 0$ , então adicione o corte 2.49 a **PMR**.

**enquanto**  $\Phi(\bar{\mathbf{u}} > 0)$

    Faça a ramificação resultando nos nós  $\delta'$  e  $\delta''$  e  $\Omega \leftarrow \Omega \cup \{\delta', \delta''\}$ .

**fim enquanto**

---

# Capítulo 3

## Algoritmo proposto

Neste Capítulo, uma nova formulação baseada na TRL (2.5) e um algoritmo baseado no método de Decomposição de Benders Revisado (2.6) são propostos para solução do PDM.

### 3.1 Aplicação da técnica TRL

Para apresentar uma nova formulação para o PDM, considera-se a formulação original F1 do PDM proposta por Kuo et al. (1993) com região viável  $X = \{x_i, i \in N : \sum_{i \in N} x_i = m, x_i \in \{0, 1\}, \forall i \in N\}$ :

#### Formulação F1

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j \quad (3.1)$$

$$\text{s.a: } \sum_{i \in N} x_i = m \quad (3.2)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N \quad (3.3)$$

A formulação (3.1) – (3.3) é não-linear e sua linearização F2, equações (2.8) - 2.14), tem um pequeno número de variáveis e uma RL fraca. É possível obter uma formulação com uma RL melhor, mas com mais variáveis e restrições, usando a técnica TRL, descrita na seção 2.5. Para isso, aplica-se o método TRL - nível  $d = 1$  sobre a região viável  $X_0 = X$ , onde cada igualdade é multiplicada pelas variáveis binárias  $x_j, \forall j \in N$ , e cada desigualdade, incluindo as restrições de domínio das variáveis, são multiplicadas tanto pelas variáveis binárias  $x_j$  quanto por seus complementos  $(1 - x_j), \forall j \in N$ . Dessa forma, variáveis auxiliares lineares  $y_{ij}$  são cri-

adas ao substituí-las por cada termo não-linear  $x_i x_j, \forall i, j \in N$ . Assim, encontra-se as seguintes restrições:

$$(3.2) \times x_j \Rightarrow \sum_{j:i<j} y_{ij} + \sum_{j:i>j} y_{ji} = (m-1)x_i \quad (3.4)$$

$$x_i \geq 0 \begin{cases} \times x_j & \Rightarrow y_{ij} \geq 0 \\ \times (1-x_j) & \Rightarrow y_{ij} \leq x_i \end{cases} \quad (3.5)$$

$$x_i \leq 1 \begin{cases} \times x_j & \Rightarrow y_{ij} \leq x_j \\ \times (1-x_j) & \Rightarrow y_{ij} \geq x_i + x_j - 1 \end{cases} \quad (3.7)$$

$$(3.8)$$

Após a aplicação da TRL sobre a formulação **F1** equações (3.1) – (3.3), obtém-se a formulação **FB1** com o mesmo número de variáveis binárias que **F1**, mas com uma quantidade maior de variáveis lineares.

### Formulação FB1

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} y_{ij} \quad (3.9)$$

s.a: restrições (3.2) – (3.3)

$$\sum_{j:i<j} y_{ij} + \sum_{j:i>j} y_{ji} = (m-1)x_i \quad \forall i \in N \quad (3.10)$$

$$y_{ij} \geq x_i + x_j - 1 \quad \forall i, j \in N : i < j \quad (3.11)$$

$$y_{ij} \leq x_i \quad \forall i, j \in N : i < j \quad (3.12)$$

$$y_{ij} \leq x_j \quad \forall i, j \in N : i < j \quad (3.13)$$

$$y_{ij} \geq 0, \quad \forall i, j \in N : i < j \quad (3.14)$$

A formulação acima é claramente um problema linear inteiro misto. Ela tem uma RL melhor que **F1** e **F2**, mas também uma quantidade maior de restrições e variáveis lineares. Ela pode ser fortalecida ainda mais através da aplicação do método TRL - nível  $d = 2$  sobre a região viável  $X_1 = \{x_i, y_{ij}, i, j \in N, i < j : \text{restrições}(3.2) - (3.2), (3.10) - (3.14)\}$ , onde cada igualdade é multiplicada pelas variáveis binárias  $x_k, \forall k \in N$ , e cada desigualdade, incluindo as restrições de domínio das variáveis, são multiplicadas tanto pelas variáveis binárias  $x_k$  quanto por seus complementos  $(1-x_k), \forall k \in N$ . Dessa forma, variáveis auxiliares lineares  $z_{ijk}$  são criadas ao

substituí-las por cada termo não-linear  $y_{ij}x_k, \forall i, j, k \in N$ . Assim, nove famílias de desigualdades válidas são adicionadas à formulação **FB1**:

$$(3.10) \times x_k \Rightarrow \sum_{i:i < j < k} z_{ijk} + \sum_{i:j < i < k} z_{jik} + \sum_{i:j < k < i} z_{jki} = (m-2)y_{jk} \quad (3.15)$$

$$(3.14) \begin{cases} \times x_k & \Rightarrow z_{ijk} \geq 0 \\ \times (1-x_k) & \Rightarrow z_{ijk} \leq y_{ij} \end{cases} \quad (3.16)$$

$$(3.17)$$

$$(3.12) \begin{cases} \times x_k & \Rightarrow z_{ijk} \leq y_{ij} \\ \times (1-x_k) & \Rightarrow -z_{ijk} + y_{ik} + y_{ij} - x_i \leq 0 \end{cases} \quad (3.18)$$

$$(3.19)$$

$$(3.13) \begin{cases} \times x_k & \Rightarrow z_{ijk} \leq y_{jk} \\ \times (1-x_k) & \Rightarrow -z_{ijk} + y_{jk} + y_{ij} - x_j \leq 0 \end{cases} \quad (3.20)$$

$$(3.21)$$

$$(3.11) \begin{cases} \times x_k & \Rightarrow -z_{ijk} + y_{ik} + y_{jk} - x_k \leq 0 \\ \times (1-x_k) & \Rightarrow x_i + x_j + x_k - y_{ij} - y_{jk} - y_{ik} + z_{ijk} \leq 1 \end{cases} \quad (3.22)$$

$$(3.23)$$

Após a aplicação da TRL nível  $d = 2$  sobre a região viável  $X_1 = \{x_i, y_{ij}, i, j \in N, i < j : \text{restrições}(3.2) - (3.3), (3.10) - (3.14)\}$ , obtêm-se a formulação **FB2**:

### Formulação FB2

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} y_{ij} \quad (3.24)$$

s.a: restrições (3.2) – (3.3), (3.10) – (3.14)

$$\sum_{i:i < j < k} z_{ijk} + \sum_{i:j < i < k} z_{jik} + \sum_{i:j < k < i} z_{jki} = (m-2)y_{jk} \quad \forall j, k \in N : j < k \quad (3.25)$$

$$z_{ijk} \leq y_{ij} \quad \forall i, j, k \in N : i < j, j < k \quad (3.26)$$

$$z_{ijk} \leq y_{ik} \quad \forall i, j, k \in N : i < j, j < k \quad (3.27)$$

$$z_{ijk} \leq y_{jk} \quad \forall i, j, k \in N : i < j, j < k \quad (3.28)$$

$$-z_{ijk} + y_{ik} + y_{jk} - x_k \leq 0 \quad \forall i, j, k \in N : i < j, j < k \quad (3.29)$$

$$-z_{ijk} + y_{jk} + y_{ij} - x_j \leq 0 \quad \forall i, j, k \in N : i < j, j < k \quad (3.30)$$

$$-z_{ijk} + y_{ik} + y_{ij} - x_i \leq 0 \quad \forall i, j, k \in N : i < j, j < k \quad (3.31)$$

$$x_i + x_j + x_k - y_{ij} - y_{jk} - y_{ik} + z_{ijk} \leq 1 \quad \forall i, j, k \in N : i < j, j < k \quad (3.32)$$

$$z_{ijk} \geq 0 \quad \forall i, j, k \in N : i < j, j < k \quad (3.33)$$

Pode-se observar que **FB2** tem o mesmo número de variáveis binárias que **F1** e **F2** e uma quantidade maior ainda de variáveis lineares que a formulação **FB1**, ou seja, uma quantidade muito maior de variáveis lineares que binárias, o que torna interessante usar o método de Decomposição de Benders Revisado descrita na seção 2.6.

## 3.2 Aplicação do método de Decomposição de Benders Revisado

Para a aplicação do método de Decomposição de Benders Revisado ao PDM, observa-se que todas as famílias de desigualdades geradas, com exceção de (3.25), têm uma estrutura especial, ou seja, estão escritas na forma  $\forall i, j, k \in N, i < j, j < k$ . Apesar da restrição (3.25) contribuir muito para a formulação, ela a torna mais difícil de resolver e portanto não vamos considerá-la para solução do problema da região de viabilidade  $Z = \{z_{ijk}, \forall i, j, k \in N, i < j, j < k : \text{s.a (3.26) – (3.32)}\}$ , optando-se por usar uma representação parcial do modelo gerado através do TRL - nível  $d = 2$ , conforme já sugerido por Adams e Sherali (1986).

Para qualquer  $\bar{x}_i, \bar{y}_{ij} \in X_1, \forall i, j \in N, i < j$ , onde  $X_1 = \{x_i, y_{ij}, \forall i, j \in N, i < j : \text{s.a: (3.2) - (3.3), (3.10) - (3.14)}\}$ , uma das duas afirmações seguintes é verdadeira:

1.  $\exists z_{ijk} \in Z, \forall i, j, k \in N, i < j < k$ , onde  $Z = \{(3.26) - (3.33)\}$  ou
2.  $\nexists z_{ijk} \in Z, \forall i, j, k \in N, i < j < k$ .

Para o caso 1 implica que existe um  $z_{ijk} \in Z, \forall i, j, k \in N, i < j, j < k$ , onde  $Z$  é a região de viabilidade de  $z_{ijk}$ . Já o caso 2 implica que não existe um  $z_{ijk} \in Z, \forall i, j, k \in N, i < j, j < k$  e pelo *Lema de Farkas* pode-se achar um hiperplano que separa o ponto  $\bar{x}_i, \bar{y}_{ij} \in X_1, \forall i, j \in N, i < J$  da região  $Z$ . Uma forma de achá-lo é através da introdução de variáveis de erro e que viabilizariam o subsistema:

$$\min e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 \quad (3.34)$$

$$\text{s.a: } -z_{ijk} + e_1 \geq -\bar{y}_{ij} \quad \forall i, j, k \in N : i < j, j < k \quad (3.35)$$

$$-z_{ijk} + e_2 \geq -\bar{y}_{ik} \quad \forall i, j, k \in N : i < j, j < k \quad (3.36)$$

$$-z_{ijk} + e_3 \geq -\bar{y}_{jk} \quad \forall i, j, k \in N : i < j, j < k \quad (3.37)$$

$$z_{ijk} + e_4 \geq \bar{y}_{ik} + \bar{y}_{jk} - \bar{x}_k \quad \forall i, j, k \in N : i < j, j < k \quad (3.38)$$

$$z_{ijk} + e_5 \geq \bar{y}_{jk} + \bar{y}_{ij} - \bar{x}_j \quad \forall i, j, k \in N : i < j, j < k \quad (3.39)$$

$$z_{ijk} + e_6 \geq \bar{y}_{ik} + \bar{y}_{ij} - \bar{x}_i \quad \forall i, j, k \in N : i < j, j < k \quad (3.40)$$

$$-z_{ijk} + e_6 \geq \bar{x}_i + \bar{x}_j + \bar{x}_k - \bar{y}_{ij} - \bar{y}_{jk} - \bar{y}_{ik} - 1 \quad \forall i, j, k \in N : i < j, j < k \quad (3.41)$$

$$e_1, e_2, e_3, e_4, e_5, e_6, e_7 \geq 0 \quad (3.42)$$

$$z_{ijk} \geq 0 \quad \forall i, j, k \in N : i < j, j < k \quad (3.43)$$

Associando variáveis duais  $u_1 - u_7$  às restrições (3.35) - (3.41), pode-se encontrar o hiperplano através do subproblema dual escrito como:

$$\begin{aligned} \max & -\bar{y}_{ij}u_1 - \bar{y}_{ik}u_2 - \bar{y}_{jk}u_3 \\ & + (\bar{y}_{ik} + \bar{y}_{jk} - \bar{x}_k)u_4 + (\bar{y}_{jk} + \bar{y}_{ij} - \bar{x}_j)u_5 + (\bar{y}_{ik} + \bar{y}_{ij} - \bar{x}_i)u_6 \\ & + (\bar{x}_i + \bar{x}_j + \bar{x}_k - \bar{y}_{ij} - \bar{y}_{jk} - \bar{y}_{ik})u_7 \end{aligned} \quad (3.44)$$

$$\text{s.a: } -u_1 - u_2 - u_3 + u_4 + u_5 + u_6 - u_7 \leq 0 \quad (3.45)$$

$$u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7 \leq 1 \quad (3.46)$$

$$u_1, u_2, u_3, u_4, u_5, u_6, u_7 \geq 0 \quad (3.47)$$

Para qualquer  $\bar{x}_i, \bar{y}_{ij} \in X_1, \forall i, j \in N, i < j$ , tal que não existe  $z_{ijk} \in Z, \forall i, j, k \in N, i < j, j < k$ , pode-se achar um raio extremo que provê um novo corte para o problema mestre **PM**, tal que (3.44)  $> 0$ .

Desse modo, fixada uma solução  $(x, y) = (\bar{x}, \bar{y})$ , bastaria resolver o dual em  $u_1 - u_7$  para obter um corte violado. No entanto, observando a estrutura das restrições (3.26) – (3.32), pode-se notar que a variável  $z_{ijk}$  está canalizada, ou seja, pode-se reescrever os pares de desigualdades (3.26) e (3.29), (3.27) e (3.30), (3.28) e (3.31) e (3.32) e (3.33), na forma das desigualdades (3.48), (3.49), (3.50) e (3.51), respectivamente:

$$\bar{y}_{ik} + \bar{y}_{jk} - \bar{x}_k \leq z_{ijk} \leq \bar{y}_{ij} \quad (3.48)$$

$$\bar{y}_{jk} + \bar{y}_{ij} - \bar{x}_j \leq z_{ijk} \leq \bar{y}_{ik} \quad (3.49)$$

$$\bar{y}_{ik} + \bar{y}_{ij} - \bar{x}_i \leq z_{ijk} \leq \bar{y}_{jk} \quad (3.50)$$

$$0 \leq z_{ijk} \leq \bar{y}_{ij} + \bar{y}_{jk} + \bar{y}_{ik} - \bar{x}_i - \bar{x}_j - \bar{x}_k + 1 \quad (3.51)$$

---

**Algoritmo 2:** Cálculo do valor de cada variável dual

---

$\forall i, j, k \in N : i < j, j < k$

Se  $(\bar{y}_{ik} + \bar{y}_{jk} - \bar{x}_k > \bar{y}_{ij})$ , então  $u_1 = u_4 = 1/7$ . Senão  $u_1 = u_4 = 0$ .

Se  $(\bar{y}_{jk} + \bar{y}_{ij} - \bar{x}_j > \bar{y}_{ik})$ , então  $u_2 = u_5 = 1/7$ . Senão  $u_2 = u_5 = 0$ .

Se  $(\bar{y}_{ik} + \bar{y}_{ij} - \bar{x}_i > \bar{y}_{jk})$ , então  $u_3 = u_6 = 1/7$ . Senão  $u_3 = u_6 = 0$ .

Se  $(\bar{y}_{ij} + \bar{y}_{jk} + \bar{y}_{ik} - \bar{x}_i - \bar{x}_j - \bar{x}_k + 1 < 0)$ , então  $u_7 = 1/7$ . Senão  $u_7 = 0$ .

---

Ou seja, se por exemplo,  $(\bar{y}_{ik} + \bar{y}_{jk} - \bar{x}_k > \bar{y}_{ij})$ , então o par de desigualdades (3.26) e (3.29) seriam desrespeitados, o que implicaria em penalidade nas variáveis  $u_1$  e  $u_4$  no subproblema dual (3.44) - (3.47). Se caso todas os pares de desigualdades (3.26) e (3.29), (3.27) e (3.30), (3.28) e (3.31) e (3.32) e (3.33) fossem desrespeitados, então, conforme a restrição (3.46), a soma das variáveis duais deve ser menor ou igual a 1 e, portanto, o valor máximo de cada variável dual  $u_1 - u_7$  seria igual a 1/7, pois desta forma o subsistema (3.45) - (3.47) continuaria sendo viável e um novo corte  $-\bar{u}_{1ijk}y_{ij} - \bar{u}_{2ijk}y_{ik} - \bar{u}_{3ijk}y_{jk} + \bar{u}_{4ijk}(y_{ik} + y_{jk} - x_k) + \bar{u}_{5ijk}(y_{jk} + y_{ij} - x_j) + \bar{u}_{6ijk}(y_{ik} + y_{ij} - x_i) + \bar{u}_{7ijk}(x_i + x_j + x_k - y_{ij} - y_{jk} - y_{ik}) \leq 0$  poderia ser adicionado ao Problema Mestre Relaxado (**PMR**), onde as desigualdades são adicionadas sob demanda:

### Formulação PMR

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} y_{ij} \quad (3.52)$$

$$\text{s.a: } \sum_{i \in N} x_i = m \quad (3.53)$$

$$\sum_{j:i < j} y_{ij} + \sum_{j:i > j} y_{ji} = (m - 1)x_i \quad \forall i \in N \quad (3.54)$$

$$y_{ij} \geq x_i + x_j - 1 \quad \forall i, j \in N : i < j \quad (3.55)$$

$$y_{ij} \leq x_i \quad \forall i, j \in N : i < j \quad (3.56)$$

$$y_{ij} \leq x_j \quad \forall i, j \in N : i < j \quad (3.57)$$

$$y_{ij} \geq 0 \quad \forall i \in N \quad (3.58)$$

$$0 \leq x_i \leq 1 \quad \forall i \in N \quad (3.59)$$

$$\begin{aligned} & -\bar{u}_{1ijk} y_{ij} - \bar{u}_{2ijk} y_{ik} - \bar{u}_{3ijk} y_{jk} \\ & + \bar{u}_{4ijk} (y_{ik} + y_{jk} - x_k) + \bar{u}_{5ijk} (y_{jk} + y_{ij} - x_j) \\ & + \bar{u}_{6ijk} (y_{ik} + y_{ij} - x_i) \\ & + \bar{u}_{7ijk} (x_i + x_j + x_k - y_{ij} - y_{jk} - y_{ik}) \leq 0 \end{aligned} \quad \forall i, j, k \in N : i < j, j < k, h = 1, \dots, H \quad (3.60)$$

onde  $H$  é o número máximo de desigualdades necessárias para se obter a solução ótima, podendo ser geradas ao longo da árvore de *Branch-and-Bound*.

### 3.3 Proposição de uma solução viável para o modelo

Uma vez que gasta-se um grande quantidade de tempo para que o *CPLEX* consiga achar uma solução viável, foi usada uma heurística *GRASP* proposta por [Silva et al. \(2004\)](#) de modo a obter uma boa solução viável inicial.

Ela consiste em construir uma solução inicial selecionando-se um elemento aleatoriamente da lista restrita de candidatos *LRC* de tamanho  $k$  a cada iteração de construção, onde  $k$  varia ao longo do conjunto de iterações, até que  $m$  elementos sejam selecionados.

Para o primeiro elemento da solução construída, a *LRC* é criada calculando a contribuição de cada elemento na solução através da soma das  $k$  maiores distâncias entre cada elemento. Os  $m - 1$  elementos restantes são selecionados da *LRC* que é reconstruída para cada iteração, usando um procedimento adaptativo.



Como as soluções geradas pela fase de construção podem não ser localmente ótimos, então a fase de busca local tem como objetivo tentar melhorar cada solução construída ao substituir sucessivamente a solução atual por uma melhor de sua vizinhança, até que nenhuma solução melhor seja encontrada. Para esta fase de busca local, foi usado o algoritmo proposto por Ghosh (1996).

### 3.4 Pré-processamento do método de Decomposição de Benders Revisado

De modo a acelerar a convergência do método proposto, é utilizado uma técnica de pré-processamento no método de Decomposição de Benders Revisado McDaniel e Devine (1977). Ela consiste em gerar cortes a partir de soluções fracionárias, ao desconsiderar as restrições de integralidade e resolvendo o Problema Mestre durante algumas iterações. A idéia geral é de gerar cortes a partir de um programa de programação linear ao invés de programas inteiros.

Existem algumas regras para determinar o número de vezes que o problema mestre deve ser resolvido com as restrições de integralidade relaxadas. Algumas possibilidades são:

- Até que não se consiga mais realizar iterações, ou seja, até quando não se perceba que o limite inferior possa ser melhorado;
- resolver o problema relaxado para as primeiras  $k$  iterações,  $k$  inteiro positivo, e então resolver o problema inteiro;
- ou então até que seja atingido um critério de convergência  $\delta$ . Por exemplo um  $GAP$  de otimalidade  $< \delta$ .

### 3.5 Fixação de variáveis

De forma a reduzir o tamanho da formulação, um algoritmo de fixação de variáveis proposto por Mitchell (1997) é aplicado imediatamente após o algoritmo de pré-processamento (seção 3.4). Seja uma solução primal viável atual  $(\hat{x}, \hat{y})$  com valor  $\bar{v}$ , os custos reduzidos  $r_i$  da variável  $x_i$ ,  $\forall i \in N$ , e um limite inferior  $\bar{v}_{LB}$  para o problema de programação inteira. Mitchell (1997) mostra que se  $\hat{x} = 0$  e  $(r_i > (\bar{v} - \bar{v}_{LB})), \forall i \in N$ , então  $x_i$  pode ser fixado em zero em qualquer solução ótima do problema e isso implica que  $y_{ij}$  também pode ser fixado em zero, uma vez

que  $y_{ij} = x_i x_j, \forall i, j \in N$ . Ao fixar a variável  $x_i$  para 0, a linha  $i$  e coluna  $j$  correspondentes são removidas da matriz de distâncias  $d_{ij}$  e o conjunto  $|N| = n$  é decrescido. É importante observar que o custo reduzido da variável  $x_i, \forall i \in N$ , é dado por  $r_i = \sum_{j:i < j} w_{ij} + \sum_{j:i > j} w_{ji}$ , onde  $w_{ij}, \forall i, j \in N$ , é o custo reduzido da variável  $y_{ij}$ .

### 3.6 Método exato para solução do PDM

Nesta seção é descrito o método de Decomposição de Benders aplicado ao PDM com geração de planos de cortes ao longo dos nós da árvore *Branch-and-Bound*. Ele é descrito a seguir, onde:

- *NHS*: é o número de iterações de pré-processamento em que o problema mestre relaxado é resolvido de forma a gerar cortes através do subproblema;
- *cHS*: é o contador do número de iterações de pré-processamento executados;
- *NBB*: é o número planos de cortes gerados nos nós da árvore *Branch-and-Bound*;
- *cBB*: é o número de iterações de planos de cortes gerados nos nós da árvore *Branch-and-Bound*;
- $\zeta(\mathbf{x}, \mathbf{y})$ : é o valor da função objetivo (3.52) para uma solução  $(\mathbf{x}, \mathbf{y})$ ;
- $r_i$ : são os custos reduzidos das variáveis  $x_i, \forall i \in N$ ;
- $\phi(\mathbf{u})$ : é o valor da função objetivo (3.44);
- $\Omega$ : é um conjunto de nós viáveis da árvore de *Branch-and-Bound* e
- $\delta$ : é um nó da árvore com as restrições de *branching* e as desigualdades (3.60):

É importante observar que o **Passo 6** trata-se do algoritmo **1** com um novo critério de parada *NBB* responsável por limitar o número de cortes adicionados ao longo da árvore *Branch-and-Bound* sob demanda.

---

**Algoritmo 3:** Método de Decomposição de Benders Revisado aplicado ao PDM
 

---

**Inicialização:** Faça  $cHS \leftarrow 0$  e  $cBB \leftarrow 0$ .

**Passo 1 (Pré-processamento):**

**Enquanto**  $cHS < NHS$  **faça**

Resolva o **PMR** (3.52) – (3.58), obtendo  $\bar{x}_i, \bar{y}_{ij}, \forall i, j, i < j$ .

Resolva o subproblema (3.44) – (3.47), através do algoritmo 2, obtendo  $u_i, 1 < i < 7$ . Se existe uma variável dual não-nula e se o seu coeficiente correspondente na função objetivo (3.44) for também não-nulo,

$\forall i, j, k \in N, i < j < k$ , então adicione o corte correspondente (3.60) no **PM**.

Incremente  $cHS$ .

**fim enquanto**

**Passo 2:** Obtenha  $\bar{v}_{LB} \leftarrow GRASP$ (seção 3.3).

**Passo 3(Fixação de variáveis:** Resolva **PMR** (3.52) – (3.58), obtendo  $r_i, \forall i \in N$  e faça  $\bar{v} \leftarrow \zeta(\mathbf{x}, \mathbf{y})$ .

Se  $(r_i > (\bar{v} - \bar{v}_{LB}))$ ,  $\forall i \in N$ , então  $x_i \leftarrow 0$  e  $y_{ij} \leftarrow 0$ , remova da matriz de distâncias  $d_{ij}$  a linha  $i$  e coluna  $i$  correspondentes e decrémente o conjunto  $|N| = n$ .

Reescreva o **PMR** (3.52) – (3.58) sem as linhas e colunas removidas. Reinsira os cortes gerados no **Passo 1** adaptados para o modelo reduzido.

**Passo 4:** Caso alguma variável da solução gerada no **Passo 3** tenha valor igual a 1 e ela tenha sido fixada em 0, então gere uma nova solução através da heurística para o modelo reduzido.

**Passo 5:** Adicione as restrições de integralidade (3.3) no modelo reduzido, insira a solução no *CPLEX* e entre no *Branch-and-Bound*.

**Passo 6:** Faça  $\Omega = \{\delta\}$ , onde  $\delta$  é o nó raiz do sistema (3.52) – (3.60).

**Enquanto**  $cHS < NHS$  **faça**

Selecione  $\delta^* \in \Omega$

$\Omega \leftarrow \Omega \setminus \{\delta^*\}$

**Faça**

Resolva  $\delta^*$  obtendo  $\bar{x}$  e  $\phi(\bar{x})$ .

Resolva o algoritmo 2, obtendo  $\phi(\mathbf{u})$ .

Se existe uma variável dual não-nula e o seu coeficiente correspondente na função objetivo (3.44) for também não-nulo,  $\forall i, j, k \in N, i < j < k$ , então adicione o corte correspondente (3.60) no **PM**. Incremente  $cBB$ .

**enquanto**  $(\phi(\mathbf{u}) > 0$  ou  $cBB < NBB)$

Faça a ramificação resultando nos nós  $\delta'$  e  $\delta''$  e  $\Omega \leftarrow \Omega \cup \{\delta', \delta''\}$ .

**fim enquanto**

---

# Capítulo 4

## Resultados computacionais

Neste capítulo são apresentados os resultados computacionais alcançados de forma a avaliar o desempenho do método de Decomposição de Benders Revisado (**DBR**) para solução do PDM proposto na seção 3.6. Ele foi implementado em linguagem C/C++ através usando o *ILOG CPLEX 12.1 Concert Technology*.

Todos os testes foram feitos usando um computador *Intel(R) Core(TM) 2 Quad CPU Q8400 2.66Ghz* com 4GB de memória RAM e sistema operacional *Linux*. Foram utilizados dois conjuntos de instâncias para os testes:

- **GKD**: propostos por [Martí et al. \(2010\)](#), consistem em 50 matrizes cujos valores são distâncias euclidianas entre cada par de pontos com coordenadas geradas aleatoriamente no intervalo  $[0, 10]$ , onde cada ponto tem  $r$  coordenadas variando entre 2 e 21. Os tamanhos das matrizes são tais que  $n = 25$  para  $m = 2$  e  $m = 7$ ,  $n = 50$  para  $m = 5$  e  $m = 15$ ,  $n = 100$  para  $m = 10$  e  $m = 30$ ,  $n = 125$  para  $m = 12$  e  $m = 37$  e  $n = 150$  para  $m = 15$  e  $m = 45$ .
- **SOM**: propostos por [Silva et al. \(2004\)](#), consistem em 50 matrizes cujos valores inteiros são gerados seguindo uma distribuição uniforme no intervalo  $[0, 9]$ . Os tamanhos das matrizes são os mesmos para o conjunto de instâncias **GKD**.

Para todos os testes realizados, o número de iterações de pré-processamento *NHS* (seção 3.4) e o critério de parada *NBB* do **Passo 6** do algoritmo 3 (seção 3.6) foram escolhidos empiricamente para cada instância. O *GAP* de otimalidade é calculado como o limite superior (retornada pelo *CPLEX*) menos o limite inferior, dividido pelo limite superior e multiplicado por 100, ambos retornados pelo *CPLEX*. Para o conjunto **GKD** não foi estabelecido um tempo limite como critério de parada para o algoritmo, enquanto que para o conjunto **SOM** estabeleceu-se um limite de execução igual a uma hora (3600s), devido às suas instâncias serem caracterizadas

por uma baixa variabilidade e portanto com um grande número de soluções de-geradas, o que desfavorece a utilização de um método exato para a sua solução, enquanto beneficia heurísticas a encontrarem bons limites superiores.

As tabelas 4.1 e 4.2 mostram o número de iterações de pré-processamento NHS (seção 3.4), aplicados a cada instância, o tempo de execução em segundos T(s) e o GAP de otimalidade, durante a fase de pré-processamento. As instâncias que não tiveram iteração de pré-processamento estão marcadas com o símbolo “-”.

Para o conjunto GKD, os resultados mostram que as 25 instâncias que tiveram iterações de pré-processamento, em um total de 50, obtiveram GAP de otimalidade inferior a 2% com exceção da instância 17, sendo que 14(28%) instâncias atingiram a otimalidade apenas nesta fase de pré-processamento. Apenas 5 instâncias gastaram mais uma hora de tempo computacional nesta fase.

Tabela 4.1: Resultados de pré-processamento para as instâncias GKD propostas por Martí et al. (2010) sem tempo limite como critério de parada

Instância	n	m	NHS	T(s)	GAP(%)	Instância	n	m	NHS	T(s)	GAP(%)
1	25	2	0	-	-	26	100	30	3	285,13	0,25
2	25	2	0	-	-	27	100	30	3	820,76	0,02
3	25	2	0	-	-	28	100	30	2	54,59	0,00
4	25	2	0	-	-	29	100	30	3	147,31	0,00
5	25	2	0	-	-	30	100	30	3	390,49	0,00
			Média	-	-				Média	339,66	0,05
6	25	7	2	0,05	0,00	31	125	12	0	-	-
7	25	7	2	0,03	0,00	32	125	12	0	-	-
8	25	7	2	0,07	0,00	33	125	12	0	-	-
9	25	7	2	0,03	0,00	34	125	12	0	-	-
10	25	7	2	0,04	0,00	35	125	12	0	-	-
			Média	0,044	0,00				Média	-	-
11	50	5	0	-	-	36	125	37	3	3003,99	0,78
12	50	5	0	-	-	37	125	37	3	1419,36	0,00
13	50	5	0	-	-	38	125	37	3	3219,65	0,17
14	50	5	0	-	-	39	125	37	4	15562,09	0,07
15	50	5	0	-	-	40	125	37	4	1837,3	0,27
			Média	-	-				Média	5008,48	0,26
16	50	15	3	4,13	0,78	41	150	15	0	-	-
17	50	15	2	1,32	10,23	42	150	15	0	-	-
18	50	15	3	3,05	0,00	43	150	15	0	-	-
19	50	15	3	3,52	0,00	44	150	15	0	-	-
20	50	15	5	6,45	0,93	45	150	15	0	-	-
			Média	3,69	2,39				Média	-	-
21	100	10	0	-	-	46	150	45	4	35514,69	0,00
22	100	10	0	-	-	47	150	45	3	15066,3	0,00
23	100	10	0	-	-	48	150	45	2	8259,55	1,37
24	100	10	0	-	-	49	150	45	3	11024,29	0,00
25	100	10	0	-	-	50	150	45	2	3811,85	1,48
			Média	-	-				Média	14735,34	0,57

Enquanto que para o conjunto **SOM**, entre as 30 instâncias que tiveram iterações de pré-processamento, em um total de 50, apenas 4 atingiram a otimalidade nesta fase de pré-processamento. Para as instâncias restantes, obteve-se **GAP** de otimalidade variando entre 4,57% e 13,64% para tamanhos entre  $n = 25$  para  $m = 7$  e  $n = 50$  para  $m = 15$ , enquanto que para os de tamanho entre  $n = 100$  para  $m = 30$  e  $n = 150$  para  $m = 45$ , o **GAP** de otimalidade variou entre 13,43% e 24,43%. Observa-se para este conjunto que 11 instâncias gastaram uma hora de tempo computacional nesta fase.

Tabela 4.2: Resultados de pré-processamento para as instâncias **SOM** propostas por [Silva et al. \(2004\)](#), considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada

Instância	n	m	NHS	T(s)	GAP(%)	Instância	n	m	NHS	T(s)	GAP(%)
1	25	2	0	-	-	26	100	30	4,00	1235,63	14,84
2	25	2	0	-	-	27	100	30	4,00	1338,66	15,16
3	25	2	0	-	-	28	100	30	4,00	1841,97	13,43
4	25	2	0	-	-	29	100	30	4,00	1392,33	14,87
5	25	2	0	-	-	30	100	30	4,00	3600,00	15,81
			Média	-	-				Média		14,82
6	25	7	2	0,09	0,00	31	125	12	0,00	-	-
7	25	7	3	0,15	0,00	32	125	12	0,00	-	-
8	25	7	3	0,12	0,00	33	125	12	0,00	-	-
9	25	7	2	0,09	4,57	34	125	12	0,00	-	-
10	25	7	3	0,11	0,00	35	125	12	0,00	-	-
			Média	0,11	0,91				Média	-	-
11	50	5	0	-	-	36	125	37	3,00	3600,00	17,96
12	50	5	0	-	-	37	125	37	3,00	3600,00	19,04
13	50	5	0	-	-	38	125	37	3,00	3600,00	18,14
14	50	5	0	-	-	39	125	37	3,00	3600,00	18,92
15	50	5	0	-	-	40	125	37	3,00	3600,00	17,98
			Média	-	-				Média		18,41
16	50	15	2	2,04	13,47	41	150	15	4,00	468,90	24,43
17	50	15	2	2,88	9,27	42	150	15	4,00	601,30	23,35
18	50	15	3	10,94	8,25	43	150	15	4,00	460,76	22,25
19	50	15	2	2,49	13,64	44	150	15	4,00	512,51	21,75
20	50	15	2	3,19	9,31	45	150	15	4,00	721,00	23,35
			Média	4,31	10,79				Média		23,03
21	100	10	0	-	-	46	150	45	2,00	3600,00	26,30
22	100	10	0	-	-	47	150	45	2,00	3600,00	25,78
23	100	10	0	-	-	48	150	45	2,00	3600,00	26,84
24	100	10	0	-	-	49	150	45	2,00	3600,00	27,60
25	100	10	0	-	-	50	150	45	2,00	3600,00	26,18
			Média	-	-				Média		26,54

As tabelas 4.3 e 4.4 fazem comparação entre o algoritmo *Branch-and-Bound* (**BB**) proposto por Martí et al. (2010) e **DBR** com um limite de execução igual a uma hora, uma vez que eles reportam os seus resultados considerando esse limite. Além disso, eles mostram apenas a média do tempo computacional em segundos e do **GAP** de otimalidade para cada combinação de valores de  $n$  e  $m$ .

Para **GKD**, observa-se que o algoritmo **DBR** atinge a otimalidade em 38 instâncias em um total de 50 com apenas uma hora de tempo computacional. Além disso, **DBR** obtém um **GAP** médio consideravelmente menor que **BB** para todos os tamanhos, sendo que o maior **GAP** médio de **DBR** é de 1,02% para as instâncias de tamanho  $n = 150$  e  $m = 45$ , contra 13,70% para as instâncias de tamanho  $n = 125$  e  $m = 37$  de **BB**. O algoritmo **DBR** perde apenas em tempo computacional para pequenas instâncias variando de  $n = 25$  e  $m = 2$  a  $n = 100$  e  $m = 10$ , mas atingindo o mesmo **GAP** de otimalidade médio em relação a **BB** para esses tamanhos.

Os resultados mostram que para o conjunto **SOM**, **DBR** gasta mais tempo na média que **BB** para atingir a otimalidade para pequenas instâncias de tamanho variando entre  $n = 25$  para  $m = 7$  e  $n = 100$  para  $m = 10$ . Mas para instâncias maiores, **DBR** consegue atingir um **GAP** médio relativamente menor que **BB** para um intervalo de 3600 segundos, com exceção das instâncias de tamanho  $n = 125$  para  $m = 12$ , uma vez que **BB** consegue atingir a otimalidade com 2678,9 segundos.

Tabela 4.3: Comparativo entre o algoritmo *Branch-and-Bound* **BB**(Martí et al. (2010)) e o algoritmo **DBR** proposto para as instâncias **GKD** por Martí et al. (2010), considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada

Instância	n	m	BB		DBR		Instância	n	m	BB		DBR	
			T(s)	GAP(%)	T(s)	GAP(%)				T(s)	GAP(%)	T(s)	GAP(%)
1	25	2			0,16	0,00	26	100	30			582,42	0,00
2	25	2			0,12	0,00	27	100	30			3600	0,80
3	25	2			0,11	0,00	28	100	30			89,4	0,00
4	25	2			0,12	0,00	29	100	30			182,48	0,00
5	25	2			0,11	0,00	30	100	30			425,43	0,00
		Média	0	0,00	0,12	0,00			Média	3576,2	8,6	975,95	0,16
6	25	7			0,41	0,00	31	125	12			485,38	0,00
7	25	7			0,38	0,00	32	125	12			669,41	0,00
8	25	7			0,41	0,00	33	125	12			69,27	0,00
9	25	7			0,39	0,00	34	125	12			28,11	0,00
10	25	7			0,41	0,00	35	125	12			178,92	0,00
		Média	0	0,00	0,40	0,00			Média	297,9	0	286,22	0,00
11	50	5			1,30	0,00	36	125	37			3600	0,78
12	50	5			1,35	0,00	37	125	37			1518,5	0,00
13	50	5			1,34	0,00	38	125	37			3600	0,17
14	50	5			1,36	0,00	39	125	37			3600	0,23
15	50	5			2,43	0,00	40	125	37			3600	0,00
		Média	0	0,00	1,56	0,00			Média	3600	13,70	3183,70	0,24
16	50	15			28,01	0,00	41	150	15			381,07	0,00
17	50	15			403,55	0,00	42	150	15			1383,99	0,00
18	50	15			6,55	0,00	43	150	15			3600	1,33
19	50	15			7,21	0,00	44	150	15			461,54	0,00
20	50	15			29,12	0,00	45	150	15			1115,76	0,00
		Média	0,4	0,00	94,89	0,00			Média	1834,4	5,40	1388,47	0,27
21	100	10			17,51	0,00	46	150	45			3600	1,23
22	100	10			14,66	0,00	47	150	45			3600	0,47
23	100	10			57,01	0,00	48	150	45			3600	1,49
24	100	10			23,55	0,00	49	150	45			3600	0,43
25	100	10			14,90	0,00	50	150	45			3600	1,48
		Média	4,4	0,00	25,53	0,00			Média	3600	10,90	3600	1,02



Tabela 4.4: Comparativo entre o algoritmo *Branch-and-Bound* **BB**(Martí et al. (2010)) e o algoritmo **DBR** proposto para as instâncias **SOM** por Silva et al. (2004), considerando um tempo limite de execução igual a uma hora (3600s) como critério de parada

Instância	n	m	BB		DBR		Instância	n	m	BB		DBR	
			T(s)	GAP(%)	T(s)	GAP(%)				T(s)	GAP(%)	T(s)	GAP(%)
1	25	2			0,11	0,00	26	100	30			3600	14,84
2	25	2			0,11	0,00	27	100	30			3600	15,16
3	25	2			0,11	0,00	28	100	30			3600	13,43
4	25	2			0,10	0,00	29	100	30			3600	14,87
5	25	2			0,12	0,00	30	100	30			3600	15,81
		Média	0,00	0,00	0,11	0,00			Média	3600	31,70	3600	14,82
6	25	7			0,43	0,00	31	125	12			3600	15,61
7	25	7			0,50	0,00	32	125	12			3600	14,29
8	25	7			0,46	0,00	33	125	12			3600	18,51
9	25	7			1,39	0,00	34	125	12			3600	15,80
10	25	7			0,45	0,00	35	125	12			3600	15,99
		Média	0,00	0,00	0,65	0,00			Média	2678,90	0,00	3600	16,04
11	50	5			1,94	0,00	36	125	37			3600	17,96
12	50	5			1,61	0,00	37	125	37			3600	19,04
13	50	5			1,78	0,00	38	125	37			3600	18,14
14	50	5			1,75	0,00	39	125	37			3600	18,92
15	50	5			1,79	0,00	40	125	37			3600	17,98
		Média	0,00	0,00	1,77	0,00			Média	3600	34,60	3600	18,41
16	50	15			481,58	0,00	41	150	15			3600	24,43
17	50	15			151,38	0,00	42	150	15			3600	23,35
18	50	15			838,01	0,00	43	150	15			3600	22,25
19	50	15			741,52	0,00	44	150	15			3600	21,75
20	50	15			194,46	0,00	45	150	15			3600	23,35
		Média	22,80	0,00	481,39	0,00			Média	3600	26,70	3600	23,03
21	100	10			2449,63	0,00	46	150	45			3600	26,30
22	100	10			2202,45	0,00	47	150	45			3600	25,78
23	100	10			1022,73	0,00	48	150	45			3600	26,84
24	100	10			1227,96	0,00	49	150	45			3600	27,60
25	100	10			1514,70	0,00	50	150	45			3600	26,18
		Média	38,30	0,00	1683,49	0,00			Média	3600	35,60	3600	26,54

A tabela 4.5 mostra o tempo computacional gasto em segundos para **DBR** atingir a otimalidade para as instâncias que não conseguiram atingir o ótimo para um limite máximo de uma hora. Observa-se que **DBR** consegue atingir a otimalidade com menos de 86400 segundos (24 horas de máquina), com exceção da instância 36 que gasta quase 86000 segundos para fechar um **GAP** de 0,78% e da instância 48 que gasta aproximadamente 91000 segundos para fechar um **GAP** de 1,49%.

Tabela 4.5: Tempo computacional gasto para o algoritmo **DBR** proposto atingir a otimalidade das instâncias **GKD** propostas por [Martí et al. \(2010\)](#), sem tempo limite como critério de parada

<b>Instância</b>	<b>n</b>	<b>m</b>	<b>T(s)</b>
27	100	30	6607,31
36	125	37	89395.48
38	125	37	9360.50
39	125	37	17157.96
40	125	37	5325.80
43	150	15	4618.82
46	150	45	35652.98
47	150	45	15123.47
48	150	45	94727.12
49	150	45	11164.32
50	150	45	69379.51

As tabelas 4.6 e 4.7 mostram o tempo gasto em segundos para a heurística *GRASP* (seção 3.3) calcular soluções viáveis para o PDM, assim como o **GAP** de otimalidade calculado como o ótimo menos a solução viável obtida, dividido pelo ótimo e multiplicado por 100. Os resultados mostram que a heurística usada é capaz de encontrar soluções iniciais muito boas para o **BDR** em até 140 segundos, acelerando o método ainda mais. Os símbolos "-" da tabela 4.7 significam que o **GAP** de otimalidade não pôde ser calculado, uma vez que o ótimo não havia sido encontrado.

Tabela 4.6: Tempo computacional gasto para a Heurística *GRASP* proposta por [Silva et al. \(2004\)](#) calcular uma solução viável, assim como o **GAP** de otimalidade atingido por este para as instâncias de **GKD** propostas por [Martí et al. \(2010\)](#)

Instância	n	m	T(s)	GAP(%)	Instância	n	m	T(s)	GAP(%)
1	25	2	0.11	0.00	26	100	30	34.76	0.00
2	25	2	0.10	0.00	27	100	30	34.65	0.00
3	25	2	0.11	0.00	28	100	30	34.91	0.00
4	25	2	0.11	0.00	29	100	30	35.16	0.00
5	25	2	0.10	0.00	30	100	30	34.93	0.00
6	25	7	0.36	0.00	31	125	12	21.75	0.00
7	25	7	0.35	0.00	32	125	12	21.85	0.00
8	25	7	0.34	0.00	33	125	12	21.84	0.00
9	25	7	0.36	0.00	34	125	12	21.60	0.00
10	25	7	0.37	0.00	35	125	12	21.70	0.00
11	50	5	1.17	0.00	36	125	37	72.40	0.00
12	50	5	1.15	0.00	37	125	37	76.81	0.00
13	50	5	1.18	0.00	38	125	37	72.58	0.00
14	50	5	1.17	0.00	39	125	37	73.16	0.00
15	50	5	1.18	0.00	40	125	37	72.70	0.00
16	50	15	3.54	0.00	41	150	15	41.35	0.00
17	50	15	3.66	0,13	42	150	15	41.38	0.00
18	50	15	3.50	0.00	43	150	15	41.45	0.00
19	50	15	3.69	0.00	44	150	15	41.42	0.00
20	50	15	3.53	0.00	45	150	15	41.63	0.00
21	100	10	10.97	0.00	46	150	45	138.29	0.00
22	100	10	11.04	0,01	47	150	45	138.94	0.00
23	100	10	11.49	0.00	48	150	45	136.70	0.00
24	100	10	11.28	0.00	49	150	45	140.03	0.00
25	100	10	10.87	0.00	50	150	45	138.22	0.00

Tabela 4.7: Tempo computacional gasto para a Heurística *GRASP* proposta por [Silva et al. \(2004\)](#) calcular uma solução viável, assim como o **GAP** de otimalidade atingido por este para as instâncias de **SOM** propostas por [Silva et al. \(2004\)](#)

Instância	n	m	T(s)	GAP(%)	Instância	n	m	T(s)	GAP(%)
1	25	2	0.10	0.00	26	100	30	33.50	-
2	25	2	0.09	0.00	27	100	30	33.86	-
3	25	2	0.09	0.00	28	100	30	33.61	-
4	25	2	0.10	0.00	29	100	30	34.01	-
5	25	2	0.10	0.00	30	100	30	33.54	-
6	25	7	0.34	0.00	31	125	12	20.65	-
7	25	7	0.35	0.00	32	125	12	20.90	-
8	25	7	0.34	0.00	33	125	12	20.50	-
9	25	7	0.34	0.00	34	125	12	20.44	-
10	25	7	0.34	0.00	35	125	12	20.69	-
11	50	5	1.11	0.00	36	125	37	68.99	-
12	50	5	1.15	0.00	37	125	37	70.45	-
13	50	5	1.14	0.00	38	125	37	69.34	-
14	50	5	1.13	0.00	39	125	37	70.78	-
15	50	5	1.14	0.00	40	125	37	69.97	-
16	50	15	3.44	0.00	41	150	15	38.39	-
17	50	15	2.88	0.00	42	150	15	38.42	-
18	50	15	3.37	0.00	43	150	15	38.77	-
19	50	15	2.49	0.00	44	150	15	38.95	-
20	50	15	3.42	0.00	45	150	15	38.46	-
21	100	10	10.36	0.00	46	150	45	132.29	-
22	100	10	10.39	0.00	47	150	45	132.71	-
23	100	10	10.50	0.00	48	150	45	131.31	-
24	100	10	10.41	0.00	49	150	45	128.32	-
25	100	10	10.34	0.00	50	150	45	129.11	-

As tabelas 4.8 e 4.9 mostram a média do percentual de variáveis  $x$  e  $y$  fixadas através do algoritmo de fixação proposto na seção (seção 3.5) para os valores ótimos atingidos pelo algoritmo **DBR**. O símbolo “-” significa que nenhuma variável foi fixada.

Pode-se observar que um grande percentual de variáveis foram fixadas para o conjunto **GKD**, contribuindo para a redução da formulação. Isso se deve à qualidade dos limites superior e inferior retornados pelo pré-processamento e pela heurística *GRASP*, respectivamente. Enquanto que para o conjunto **SOM**, apenas um percentual das variáveis binárias  $x$  puderam ser fixadas (até 33,40%) para instâncias com tamanho com  $n > 50$ , dificultando a sua solução.

É importante observar que mesmo reduzindo consideravelmente o tamanho da

formulação, isso não implica em reduzir a complexidade na mesma proporção, uma vez que  $m$  permanece constante e por isso, torna-se mais difícil de se selecionar elementos.

Tabela 4.8: Média do percentual de variáveis  $x$  e  $y$  fixadas através do algoritmo de fixação proposto na seção (seção 3.5) para as instâncias **GKD** propostas por [Martí et al. \(2010\)](#)

<b>n</b>	<b>m</b>	<b>X</b>	<b>Y</b>
25	2	92.00	99,67
25	7	-	-
50	5	66.00	87,02
50	15	20.00	35,75
100	10	64,40	87,17
100	30	36.00	58,78
125	12	67,52	89,49
125	37	28,60	47,99
150	15	62,53	86,05
150	45	12.00	22,62

Tabela 4.9: Média do percentual de variáveis  $x$  e  $y$  fixadas através do algoritmo de fixação proposto na seção (seção 3.5) para as instâncias **SOM** propostas por [Silva et al. \(2004\)](#)

<b>n</b>	<b>m</b>	<b>X</b>	<b>Y</b>
25	2	92.00	99,67
25	7	-	-
50	5	33,40	54,05
50	15	0,40	0,80
100	10	9,20	17,50
100	30	0,00	0,00
125	12	5,78	11,19
125	37	0,00	0,00
150	15	2,00	3,96
150	45	0,00	0,00

# Capítulo 5

## Conclusão

Neste trabalho, foi proposta uma nova formulação para o Problema de Diversidade Máxima baseada na Técnica de Linearização de Reformulação. Para resolução deste problema foi apresentado um método de Decomposição de Benders Revisado com geração de cortes ao longo da árvore *Branch-and-Bound*.

Este método contou com um pré-processamento do método que foi capaz de acelerar a sua convergência, chegando a resolver instância de tamanho  $n = 125$  e  $m = 37$  com menos de uma hora de processamento, além de obter um *GAP* de otimalidade inferior a 2% em 48% das instâncias de um conjunto, desde que o número máximo de iterações nesta etapa fosse calibrado. Além disso, contou com uma heurística capaz de prover bons limites inferiores para o método proposto em até 140 segundo, acelerando-o ainda mais. Utilizou-se também uma técnica de fixação de variáveis que foi capaz de reduzir o tamanho da formulação em até 92% das variáveis inteiras.

Os resultados apresentados mostram que esse algoritmo é capaz de resolver problemas de até 150% com 15 selecionados com menos de 1 hora de processamento, demonstrando ser competitivo frente aos métodos propostos na literatura para solução do problema.

Como possíveis trabalhos futuros, poderia-se testar a fixação das variáveis para valores igual a um. Além disso, poderia-se testar a inclusão da restrição que foi retirada e resolvê-la através do método de geração de colunas.

# Referências Bibliográficas

- Adams, W. P. e Serali, H. D. (1986). A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290.
- Adams, W. P. e Serali, H. D. (1990). Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2):217–226.
- Adams, W. P. e Serali, H. D. (1993). Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(3):279–306.
- Andrade, P.; Plastino, A. e Martins, S. (2005). Grasp with path-relinking for the maximum diversity problem. In Nikolettseas, S., editor, *Proceedings of the 4th International Workshop on Efficient and Experimental Algorithms (WEA 2005)*, volume 3539 of *Lecture Notes in Computer Science*, pp. 558–569. Springer.
- Andrade, P.; Plastino, A.; Ochi, L. e Martins, S. (2003). Grasp for the maximum diversity problem. In *Proceedings of MIC*, pp. 25–28.
- Aringhieri, R. e Cordone, R. (2006). Better and faster solutions for the maximum diversity problem. Technical report, Universit degli Studi di Milano, Polo Didattico e di Ricerca di Crema.
- Aringhieri, R.; Cordone, R. e Melzani, Y. (2008). Tabu search vs. grasp for the maximum diversity problem. *4OR: A Quarterly Journal of Operations Research*, 6(1):45–60.
- Bhadury, J.; Mighty, E. J. e Damar, H. (2000). Maximizing workforce diversity: a network flow approach. *Omega*, 28:143–153.
- Birnbaum, B. E. e Goldman, K. J. (2009). An improved analysis for a greedy remote-clique algorithm using factor-revealing lps. *Algorithmica*, 55(1):42–59.
- Cappanera, P. (1999). A survey on obnoxious facility location problems. Technical Report 99-11, Dipartimento di Elettronica ed Informazione, Politecnico di Milano, 19.

- Chandra, B. e Halldorsson, M. (2001). Approximation algorithms for dispersion problems. *Journal of Algorithms*, 38(2):438–465.
- Cox, T. (1993). *Cultural diversity in organizations: theory, research and practice*. Berrett-Koehler Publishers, Inc, San Francisco, CA.
- Cutler, M. e Klastorin, T. (1997). The max diversity problem. Technical report, University of Washington, Department of Management Science.
- Duarte, A. e Martí, R. (2007). Tabu search and grasp for the maximum diversity problem. *European Journal of Operational Research*, 178:71–84.
- Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60.
- Erkut, E. e Neuman, S. (1991). Comparison of four models for dispersing facilities. *INFOR Canadian Journal of Operational Research and Information Processing*, 29:68–86.
- Feige, U.; Kortsarz, G. e Peleg, D. (2001). The dense k-subgraph problem. *Algorithmica*, 29:410–421.
- Fekete, S. P. e Meijer, H. (2004). Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, 38:501–511.
- Fernandez, J. (1991). *Managing a diverse work force*. Lexington Books, Lexington, MA.
- Gallego, M.; Duarte, A.; Laguna, M. e Martí, M. (2006). Hybrid heuristics for the maximum diversity problem. Technical report, University of Valencia, Spain.
- Ghosh, J. (1996). Computational aspects of the maximum diversity problem. *Operations Research Letters*, 19:175–191.
- Glover, F. (1975). Improve linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.
- Glover, F.; Hersh, G. e McMillian, C. (1977). Selecting subset of maximum diversity. Technical report, University of Colorado, Boulder.
- Glover, F.; Kuo., C. e Dhir, K. (1995). A discrete optimization model for preserving biological diversity. *Appl. Math. Modelling*, 19(11):696–701.
- Glover, F.; Kuo., C. e Dhir, K. (1998). Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences*, 19(1):109–132.



- Glover, F. e Wolsey, R. (1974). Converting the 0-1 polynomial programming problem to a 0-1 linear programming. *Operations Research*, 22(1):180–182.
- Hassin, R.; Rubinstein, S. e Tamir, A. (1997). Approximation algorithms for maximum dispersion. *Operations Research Letters*, 21:133–137.
- Karg, R. e Thompson, G. (1964). Heuristic approach to solving travelling salesman problems. *Management Science*, pp. 224–248.
- Kincaid, R. (1992). Good solutions to discrete noxious location problems via meta-heuristics. *Annals of Operations Research*, 40(1):265–281.
- Kincaid, R. e Yellin, L. G. (1993). The discrete p-dispersion-sum problem: results on trees and graphs. *Computers & Operations Research*, 1(2):171–186.
- Kochenberger, G. e Glover, F. (2006). A unified framework for modeling and solving combinatorial optimization problems: a tutorial. In Hager, W.; Huang, S.; Pardalos, P. e Prokopyev, O., editores, *Multiscale Optimization Methods and Applications*, pp. 101–124. Springer.
- Kuby, M. (1987). Programming models for facility dispersion: The p-dispersion and maximum dispersion problem. *Geographical Analysis*, 19(4):315–329.
- Kuo, C.; Glover, F. e Dhir, K. S. (1993). Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24:1171–1185.
- Macambira, E. (2002). An application of tabu search heuristic for the maximum edge-weighted subgraph problem. *Annals of Operations Research*, 177:175–190.
- Martí, R.; Gallego, M. e Duarte, A. (2010). A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research*, 200(1):36–44.
- Martin, R. K. (1999). *Large Scale Linear and Integer Optimization; A Unified Approach*, chapter Large integer programs: projection and inverse projection, pp. 613–617. Kluwer.
- McConnell, S. (1988). The new battle over immigration. *Fortune*, 117:89–102.
- McDaniel, D. e Devine, M. (1977). A modified benders partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.
- Mitchell, J. E. (1997). Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems. *European Journal of Operational Research*, 97(1):139–148.

- Palubeckis, G. (2007). Iterated tabu search for the maximum diversity problem. *Applied Mathematics and Computation*, 189:371–383.
- Pisinger, D. (2006). Upper bounds and exact algorithms for p-dispersion problems. *Computers & Operations Research*, 33:1380–1398.
- Porter, W. M.; Rawal, K. M.; Rachie, K. O.; Wien, H. C. e Williams, R. C. (1975). Cowpea germplasm catalog no 1. *International Institute of Tropical Agriculture*.
- Ravi, S.; Rosenkrantz, D. e Tayi, G. (1994). Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310.
- Santos, M.; Ribeiro, A.; Plastino, A. e Martins, S. (2005). A hybrid grasp with data mining for the maximum diversity problem. In Maria, J. B.; ; Blum, C.; Roli, A. e Sampels, M., editores, *Hybrid Metaheuristics*, volume 3636 of *Lecture Notes in Computer Science*, pp. 116–127. Springer.
- Silva, G.; Andrade, M.; Ochi, L.; Martins, S. e Plastino, A. (2007). New heuristics for the maximum diversity problem. *Journal of Heuristics*, 13(4):315–336.
- Silva, G.; Ochi, L. e Martins, S. (2004). Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In *Experimental and Efficient Algorithms*, volume 3059 of *Lecture Notes in Computer Science*, pp. 498–512. Springer.