

**UNICAMP**

UNIVERSIDADE ESTADUAL DE  
CAMPINAS

Instituto de Matemática, Estatística e  
Computação Científica

CRISTIANA INÊS JUDAR NUNES

**Modelos Matemáticos para Alocação de  
Facilidades e Centros Consumidores**

Campinas

2018

Cristiana Inês Judar Nunes

## **Modelos Matemáticos para Alocação de Facilidades e Centros Consumidores**

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Matemática Aplicada e Computacional.

Orientador: Angelo Aliano Filho

Coorientador: Cristiano Torezzan

Este exemplar corresponde à versão final da Dissertação defendida pela aluna Cristiana Inês Judar Nunes e orientada pelo Prof. Dr. Angelo Aliano Filho.

Campinas

2018

**Agência(s) de fomento e nº(s) de processo(s):** Não se aplica.

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Ana Regina Machado - CRB 8/5467

N922m Nunes, Cristiana Inês Judar, 1988-  
Modelos matemáticos para alocação de facilidades e centros  
consumidores / Cristiana Inês Judar Nunes. – Campinas, SP : [s.n.], 2018.

Orientador: Angelo Aliano Filho.  
Coorientador: Cristiano Torezzan.  
Dissertação (mestrado profissional) – Universidade Estadual de Campinas,  
Instituto de Matemática, Estatística e Computação Científica.

1. Pesquisa operacional. 2. Otimização multiobjetivo. 3. Modelagem  
matemática. I. Aliano Filho, Angelo, 1988-. II. Torezzan, Cristiano, 1976-. III.  
Universidade Estadual de Campinas. Instituto de Matemática, Estatística e  
Computação Científica. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Mathematical models for allocation of facilities and consumer centers

**Palavras-chave em inglês:**

Operations research

Multiobjective optimization

Mathematical modeling

**Área de concentração:** Matemática Aplicada e Computacional

**Titulação:** Mestra em Matemática Aplicada e Computacional

**Banca examinadora:**

Angelo Aliano Filho [Orientador]

Alireza Mohebi Ashtiani

Ana Paula da Silveira Vargas

**Data de defesa:** 04-05-2018

**Programa de Pós-Graduação:** Matemática Aplicada e Computacional

**Dissertação de Mestrado Profissional defendida em 04 de maio de 2018 e  
aprovada pela banca examinadora composta pelos Profs. Drs.**

**Prof(a). Dr(a). ANGELO ALIANO FILHO**

**Prof(a). Dr(a). ALIREZA MOHEBI ASHTIANI**

**Prof(a). Dr(a). ANA PAULA DA SILVEIRA VARGAS**

As respectivas assinaturas dos membros encontram-se na Ata de defesa

*À minha avó Maria Egydea Peluso Judar (in memoriam).*

# Agradecimentos

A Deus, pela Sua infinita misericórdia, por nunca me abandonar, por sempre me iluminar e por ter me auxiliado na trilhagem de meus caminhos.

Aos meus pais André e Silvia, meus pilares que sempre me apoiaram e se dedicaram para que eu pudesse chegar até aqui. Às minhas avós Maria Egydia (in memoriam), que financiou meus estudos desde o Ensino Fundamental ao Ensino Médio, inclusive cursos de línguas, sempre me incentivando a crescer e acreditando em mim; e Benedicta que me acolheu em São Paulo, cuidou de mim e sempre me aconselhou. À minha irmã Roberta, sobrinhos Pedro e Bruno, tia Inês, tia Dirce, tio junior, tia Sirlei, tia Flávia e tio Ramon pelo incentivo e confiança. Ao meu marido, Lucas, pela compreensão, apoio, incentivo e confiança e por ter se tornado meu pilar.

Ao meu orientador Prof. Dr. Angelo Aliano Filho, por ter acreditado em mim, pela dedicação, pelos encontros à distância, pela paciência e por ter me orientado a aprender e crescer cada dia mais. Também ao meu co-orientador, Prof. Dr. Cristiano Torezzan pela dedicação a esse curso.

Aos meus colegas de mestrado pelos grupos de estudo, e horas de apoio inconstantes, especialmente à Cris parceira das horas mais difíceis. Aos professores que me trouxeram até aqui, em especial a Profa. Dra. Sueli Costa.

À minha ex-chefe Paulina, que permitiu maior flexibilidade para que essa dissertação fosse possível, e que agregou a esse trabalho trazendo novos *insights* até então ainda não explorados. E a todos os meus colegas, de trabalho, de graduação e pessoais pelo apoio e compreensão.

Sem vocês, nada disso seria possível.

*“Não vos amoldeis às estruturas deste mundo,  
mas transformai-vos pela renovação da mente,  
a fim de distinguir qual é a vontade de Deus:  
o que é bom, o que Lhe é agradável, o que é perfeito.  
(Bíblia Sagrada, Romanos 12, 2)*

# Resumo

O presente estudo propõe modelos matemáticos de programação inteira mono e bi-objetivo e técnicas de resolução para alocação de facilidades e centros consumidores em uma rede composta de nós e arestas. Em ambas as formulações, aloca-se um conjunto de facilidades (indústrias, fábricas, centros de distribuição etc) e clientes (comércios, filiais, pontos consumidores etc) escolhendo, dentre as possibilidades, a que minimiza a distância entre as facilidades e os centros consumidores e entre as facilidades. O modelo ainda prevê o atendimento da demanda de cada centro consumidor e respeita a capacidade máxima de distribuição de cada facilidade. Para resolver estes modelos não-lineares, foi necessário empregar técnicas de linearizações de programação inteira e posteriormente utilizar métodos de escalarizações multi-objetivos para determinar um conjunto de soluções eficientes para este problema. Para validar o modelo e os métodos de resolução, apresentamos testes computacionais em instâncias geradas aleatoriamente e as curvas de Pareto associadas, para que o decisor tenha mais segurança em sua tomada de decisão.

**Palavras-chave:** Pesquisa Operacional. Otimização Multi-objetivo. Modelagem Matemática.

# Abstract

The present study proposes mono and bi-objective integer mathematical models and resolution techniques for facilities and costumers centers in a network composed of nodes and edges. In both formulations, a set of facilities (industries, factories, distribution, etc.) and customers (stories, branches, consumer points, etc) choosing among a the possibilities, to minimize the distance between facilities and the consumer centers and facilities. The models still provides for the fulfillment of demand of each consumer center and a maximum distribution capacity of each facility. To solve these nonlinear models, it was necessary to employ linear programming techniques and to use multi-objective scalarization methods to determine a set of efficient solutions for this problem. To validate the model and resolution methods, we present computational tests in randomly generated instances and associated Pareto curves, so that the decision maker has more security in their decision making.

**Keywords:** Operational Research. Multiobjective Optimization. Mathematical Modeling.

# Lista de ilustrações

Figura 1 – Branch-and-Bound binário aplicado no problema (1.8) . . . . .	29
Figura 2 – Mapa rodoviário com as distâncias de um exemplar do Problema (2.1)- (2.15) com $r = 10$ nós . . . . .	49
Figura 3 – Solução ótima de Pareto do Modelo (2.1)-(2.15) ao minimizar $z_1$ . . . . .	49
Figura 4 – Solução ótima do Modelo (2.1)-(2.15) ao minimizar $z_2$ . . . . .	50
Figura 5 – Solução ótima do Modelo (2.1)-(2.15) ao minimizar $z_3 = z_1 + z_2$ . . . . .	51
Figura 6 – Solução ótima do Modelo (2.26)-(2.35) com $\mathcal{L} = \{2, 6, 10\}$ . . . . .	51
Figura 7 – Solução ótima do Modelo (2.26)-(2.35) com $\mathcal{L} = \{4, 8, 5\}$ . . . . .	52
Figura 8 – Interface Gusek . . . . .	54
Figura 9 – Mapa rodoviário com as distâncias das 15 cidades para os testes com- putacionais . . . . .	54
Figura 10 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 1	57
Figura 11 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 2	57
Figura 12 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 3	58
Figura 13 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 4	58
Figura 14 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 5	58
Figura 15 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 6	59
Figura 16 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 7	60
Figura 17 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 8	60
Figura 18 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 9	60
Figura 19 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 1.0$ - Instância 1 . . . . .	63
Figura 20 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 0.5$ - Instância 1 . . . . .	64
Figura 21 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 0.0$ - Instância 1 . . . . .	64
Figura 22 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 1.0$ - Instância 9 . . . . .	65
Figura 23 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 0.5$ - Instância 9 . . . . .	65

Figura 24 – Esquema da solução ótima para o Problema (2.1)-(2.15) com $\lambda = 0.0$ - Instância 9 . . . . .	66
Figura 25 – Pontos não-dominados fornecidos pelo $\varepsilon$ -Restrito e Soma Ponderada do Problema (2.1)-(2.15) da Instância 1 . . . . .	68
Figura 26 – Pontos não-dominados fornecidos pelo $\varepsilon$ -Restrito e Soma Ponderada do Problema (2.1)-(2.15) da Instância 3 . . . . .	69

# Lista de tabelas

Tabela 1 – Pesos e valores para problema da Mochila . . . . .	22
Tabela 2 – Características das classes testadas . . . . .	55
Tabela 3 – Resultados computacionais para as 9 instâncias do Problema (2.26)-(2.35)	56
Tabela 4 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e $\lambda = 1.0$ . . . . .	61
Tabela 5 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e $\lambda = 0.5$ . . . . .	62
Tabela 6 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e $\lambda = 0.0$ . . . . .	62
Tabela 7 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e $\lambda = 0.0, 0.5$ e $1.0$ . . . . .	67
Tabela 8 – Resultados computacionais para o método $\varepsilon$ -Restrito aplicado na Instância 1 para o Modelo (2.36)-(2.40) . . . . .	68
Tabela 9 – Resultados computacionais para o método $\varepsilon$ -Restrito aplicado na Instância 3 para o Modelo (2.36)-(2.40) . . . . .	69

# Lista de abreviaturas e siglas

POM	Problema de Otimização Multi-objetivo
PLI	Problema de Programação Linear Inteira
PNLI	Problema de Programação Não-Linear Inteira
B & B	<i>Branch-and-Bound</i>
GLPK	<i>GNU Linear Programming Kit</i>
GUSEK	<i>GLPK Under Scite Extended Kit</i>
PO	Pesquisa Operacional

# Lista de símbolos

$i$	índice associado às facilidades
$j$	índice associado aos clientes
$k$	índice associado aos nós
$z_1$	soma das distâncias entre as facilidades e centros consumidores
$z_2$	soma das distâncias entre as facilidades
$\lambda$	peso do objetivo $z_1$
$1 - \lambda$	peso do objetivo $z_2$
$p$	número de de facilidades
$q$	número de centros consumidores
$r$	número de nós disponíveis
$C_i$	capacidade máxima de atendimento da facilidade $i$
$dem_j$	demanda do cliente $j$
$d_{k\bar{k}}$	distância entre os nós $k$ e $\bar{k}$
$M$	número grande
$\mathcal{L}$	conjunto dos clientes
$\mathcal{F}$	conjunto das facilidades

# Sumário

Introdução	17
<b>1 REVISÃO DE LITERATURA DE ASPECTOS GERAIS DE OTIMIZAÇÃO INTEIRA</b>	<b>19</b>
Revisão de literatura de aspectos gerais de otimização inteira	19
<b>1.1 Modelagem matemática em Pesquisa Operacional</b>	<b>19</b>
1.1.1 Problema do transporte	20
1.1.2 Problema da Mochila Binário	22
1.1.3 Problema das $p$ -medianas	23
1.1.4 O Problema dos $p$ -centros	24
<b>1.2 Programação inteira</b>	<b>25</b>
1.2.1 O Método <i>Branch-and-bound</i>	26
1.2.2 <i>Branch-and-Bound</i> 0-1	27
1.2.3 Método de Plano de Corte em Programação Inteira	30
1.2.3.1 Método de Plano de Corte Fracionário	30
<b>1.3 “Truques” de Programação inteira</b>	<b>33</b>
1.3.1 Linearizando polinômios 0–1	33
1.3.2 Transformando produto de variáveis binárias por contínuas	35
1.3.3 Representação de valores discretos	36
1.3.4 Custo fixo	36
1.3.5 Restrições Disjuntivas e do tipo “se-então”	36
<b>1.4 Otimização Multi-objetivo</b>	<b>37</b>
1.4.1 Definições e nomenclatura	37
1.4.2 Método da Soma Ponderada	39
1.4.3 Método do $\epsilon$ -Restrito	40
<b>2 MODELOS MATEMÁTICOS DESENVOLVIDOS</b>	<b>42</b>
<b>2.1 Modelo matemático de alocação dos clientes e facilidades</b>	<b>42</b>
<b>2.2 Modelo matemático de alocação apenas dos clientes</b>	<b>47</b>
<b>3 RESULTADOS COMPUTACIONAIS</b>	<b>53</b>
<b>3.1 Fase 1 - Resultados para a Formulação (2.26)-(2.35) que instala apenas os clientes</b>	<b>55</b>
<b>3.2 Fase 2 - Resultados para a Formulação (2.1)-(2.15) que instala os clientes e as facilidades</b>	<b>61</b>

3.3	Simulações com instâncias aleatórias . . . . .	66
3.4	Aplicação do $\varepsilon$ -Restrito . . . . .	67
4	CONCLUSÕES E PERSPECTIVAS FUTURAS . . . . .	70
	REFERÊNCIAS . . . . .	72

# Introdução

Pesquisa Operacional (PO) é um campo da Matemática que busca desenvolver métodos para análise e auxílio na tomada de decisão. Essa disciplina tem se tornado cada vez mais relevante dado o crescente aumento da complexidade e incerteza do século XXI em meio a globalização, internet e a rapidez de chegada de informações que requer decisões táticas e operacionais cada vez mais ágeis e mais eficientes.([ARENALES, 2007](#)).

Entre os cenários analisados pela PO, está a localização de facilidades e de seus respectivos centros de consumos. Exemplos típicos no setor público são alocação de escolas, hospitais, estações de bombeiros e no setor privado podemos citar a localização de fábricas e armazéns, lojas e centros de distribuição etc. Segundo [Filho et al. \(2016\)](#), a maioria dos problemas na vida prática são simultâneos e multi-objetivos, principalmente quando existe um certo grau de conflito entre os objetivos.

Nesse estudo buscamos propor um modelo matemático inovador para alocar facilidades (isto é, indústrias, fábricas, centros de distribuição etc) e centros consumidores (comércios, filiais, pontos consumidores etc) por meio de um modelo não-linear de programação inteira. Além disto, o modelo prevê o fluxo entre as facilidades e os centros de consumo, isto é, qual facilidade irá atender determinado centro. O modelo proposto é bi-objetivo, pois busca minimizar a distância entre as facilidades e também a distância entre os centros consumidores em concomitância. Utilizamos métodos de escalarições multi-objetivo clássicos para determinar o conjunto de soluções para este problema.

Problemas de localização têm grande importância prática, como por exemplo, na alocação de escolas ([BARCELOS; PIZZOLATO; LORENA, 2002](#)), antenas de telecomunicações ([CALDAS et al., 2015](#)), otimização de rolamentos em trajetórias ([OSHMAN; DAVIDSON, 1999](#)), localização de sensores numa rede de *wireless* ([GOPAKUMAR; JACOB, 2008](#)), localização de indústrias acopladas a modelos de transporte ([CUNHA, 2000](#)), localização de organizações de saúde ([GALVÃO; NOBRE; VASCONCELLOS, 1999](#)) e muitas outras aplicações. Ressaltamos que este é um dos primeiros trabalhos a discutir um modelo multi-objetivo de programação inteira baseado no problema das  $p$ -medianas e dos  $p$ -centros para alocar centros e determinar a direção do fluxo entre eles.

Este trabalho está estruturado da seguinte forma:

- O Capítulo 1 traz uma breve revisão bibliográfica sobre tópicos de PO relevantes para o estudo, como os problemas que serviram de base para o desenvolvimento dos modelos propostos, a saber, problema das  $p$ -medianas e dos  $p$ -centros. Apresentamos, algumas técnicas e truques de programação inteira, o método de solução Branch-and-Bound (B & B) e breves conceitos e terminologias da otimização multi-objetivo;
- O Capítulo 2 apresenta os modelos desenvolvidos e suas premissas. Primeiramente, apresentamos o problema de alocação de clientes e com facilidades fixas. Em seguida, é apresentado o modelo para o problema de alocação de clientes e as facilidades.
- O Capítulo 3 delinea os resultados computacionais dos modelos propostos, solucionados pelo solver GLPK (*GNU Linear Programming Kit*) usando a plataforma GUSEK (*GLPK Under Scite Extended Kit*);
- Finalmente, o Capítulo 4 apresenta as conclusões do estudo.

# Capítulo 1

## Revisão de literatura de aspectos gerais de otimização inteira

### 1.1 Modelagem matemática em Pesquisa Operacional

A Pesquisa Operacional (PO) é uma área da Matemática que auxilia na tomada de decisão principalmente em gestão de recursos sejam eles humanos, materiais ou financeiros. Por meio do uso de técnicas de modelagem matemática e eficientes algoritmos computacionais, a PO permite ao gestor tomar decisão em problemas complexos, que muitas vezes seriam impossíveis de serem solucionados sem o uso de tais técnicas.

Nesse contexto, inserem-se diversos problemas que podem ser formulados como um problema de otimização linear e não-linear. Esta seção apresenta alguns exemplos que, embora simplificados, fornecerão uma ideia dos problemas que podem ser modelados com estes conceitos.

Há diversos casos de sucesso envolvendo técnicas de PO. Podemos citar alguns exemplos, segundo [Colin \(2007\)](#):

- O ministério holandês da infra-estrutura e Meio Ambiente (*The Netherlands Rijkswaterstaat*) poupou \$15 Milhões, em 1985, por meio de uma nova política nacional de desenvolvimento de gestão da água, incluindo combinação de novas instalações, procedimentos de operações e custos;
- A Electrobras/CEPAL Brasil, em 1986, poupou \$43 milhões com a distribuição ótima de recursos hídricos e térmicos no sistema nacional de geração de energia;
- A IBM por meio da integração de uma rede nacional de distribuição de peças de reposição para melhorar o serviço de suporte poupou, em 1990, \$20 milhões mais \$250 milhões em inventário reduzido;

- A *American Airlines* teve um incremento de mais de \$500 milhões em seu faturamento, em 1992, com a nova concepção de um sistema de estrutura de preços, *overbooking* (excesso de reservas) e coordenação de vôo para aumentar os benefícios;
- A *New Haven Health Dept.*, em 1993, reduziu o número de infecções de AIDS em 33% por meio de um projeto eficaz para a troca de seringas no combate à propagação da AIDS;
- A *Procter & Gamble* redesenhou o sistema de produção e distribuição dos EUA para reduzir custos e melhorar a velocidade para chegar ao mercado, com isso, poupou 200 milhões em 1997.
- [Zachariah et al. \(2009\)](#) aplica técnicas de PO em países de baixa renda, e discute a sua relevância para os programas de controle de doenças infecciosas. Segundo este trabalho, de 2009, mais de 34 bilhões de dólares foram economizados em vacinas pelo uso otimizado de agentes biológicos de controle de patógenos.
- [Friend e Jessop \(2013\)](#) desenvolveu um algoritmo para auxiliar o governo brasileiro em organizações de localizações de saúde.
- O trabalho [Bastiaanssen, Molden e Makin \(2000\)](#) aplicou técnicas de PO em Agricultura;
- Em 2013, [Galindo e Batta \(2013\)](#) trouxeram uma revisão de estudos em Pesquisa Operacional na área de DOM -*disaster operations management*-, isto é, estudos que buscam reduzir o impacto potencial de catástrofes;
- [SteadieSeifi et al. \(2014\)](#) apresenta uma visão geral estruturada da literatura de transporte multimodal a partir de 2005,
- [Zikmund et al. \(2013\)](#) traz atualizações sobre métodos de Pesquisa Operacional em negócios. Enfatiza a interação entre pesquisa e negócios e como a importância e o alcance da pesquisa variam em diferentes situações de negócios.

Estes foram alguns casos de sucesso envolvendo técnicas de PO que nos motivam e ilustram o uso dessas técnicas. Para melhor compreender estes aspectos, apresentamos na próxima subseção, alguns modelos básicos de PO mais conhecidos na literatura, segundo [Chen, Batson e Dang \(2011\)](#).

### 1.1.1 Problema do transporte

Um problema clássico de PO é o problema de transporte dos centros de produção aos seus respectivos mercados consumidores de modo que o custo total com transporte seja

o menor possível, respeitando os limites de demanda do centro consumidor e capacidade do centro produtor. (ARENALES, 2007).

Suponha que existam  $m$  origens (centros de distribuição) e  $n$  destinos (centros consumidores), com custo de distribuição  $c_{ij}$  da origem  $i$  para o destino  $j$ . Suponha, também, que a origem  $i$  tenha capacidade  $a_i$  e o destino  $j$  demanda  $b_j$ . Como resposta, queremos a variável  $x_{ij}$  que é a quantidade transportada da origem  $i$  ao destino  $j$ .

Elementos da modelagem:

- Essas quantidades não podem ser negativas, logo  $x_{ij} \geq 0$ ,  $i = 1, \dots, m$  e  $j = 1, \dots, n$ ;
- Como  $x_{ij}$  é a quantidade transportada do produto então  $c_{ij}x_{ij}$  é o custo de transporte da origem  $i$  para o destino  $j$ . Assim, o custo total de transporte ( $z$ ) é a soma de todas as quantidades transportadas:

$$z := \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij},$$

que deve ser o menor possível;

- Além disso, as quantidades transportadas não podem ultrapassar a capacidade da origem, ou seja,

$$\sum_{i=1}^n x_{ij} \leq a_i;$$

- As quantidades transportadas não podem ultrapassar a demanda do destino, ou seja,

$$\sum_{i=1}^m x_{ij} = b_j.$$

Assim, o modelo completo do problema de transporte é definido por:

$$\begin{aligned} \text{minimize} \quad & z = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \\ \text{sujeito a} \quad & \sum_{j=1}^n x_{ij} \leq a_i \quad i = 1, \dots, m, \\ & \sum_{i=1}^m x_{ij} = b_j \quad j = 1, \dots, n \\ & x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n \end{aligned} \tag{1.1}$$

Munkres (1957) traz algumas formulações, e suas respectivas provas, para o algoritmo do modelo do transporte. Aneja e Nair (1979) considera este problema em sua versão bi-objetiva, ineditamente. Mais tarde, Vignaux e Michalewicz (1991) desenvolve um Algoritmo Genético para resolver este problema. Em seguida, Musa, Arnaout e Jung (2010) desenvolve um algoritmo de Colônia de Formigas para resolver este problema mas para instâncias complexas.

### 1.1.2 Problema da Mochila Binário

Outro problema clássico de PO, dessa vez utilizando variáveis binárias, é o Problema da Mochila.

Suponha, por exemplo, que um alpinista necessite levar consigo uma mochila com 5 itens, porém esses itens, juntos, excedem a capacidade de 60 quilos que ele seria capaz de carregar. Assim sendo, ele atribui valores de importância para cada um dos 5 itens:

Tabela 1 – Pesos e valores para problema da Mochila

Item	1	2	3	4	5
Peso(kg)	52	23	35	15	7
Valor	100	60	70	15	8

Na sua versão mais geral, seja  $x_i$  uma variável binária, tal que:

$$x_i = \begin{cases} 1, & \text{se o item } i \text{ for colocado na mochila,} \\ 0, & \text{caso contrário.} \end{cases}$$

Buscando maximizar o valor total sujeito às restrições de peso, supondo existir apenas uma unidade de cada item, o modelo de programação inteira é:

$$\begin{aligned} \text{maximizar } z &:= 100x_1 + 60x_2 + 70x_3 + 15x_4 + 8x_5 \\ \text{sujeito a } &52x_1 + 23x_2 + 35x_3 + 15x_4 + 7x_5 \leq 60 \\ &x_i \in \{0, 1\}, \quad i = 1, \dots, 5. \end{aligned}$$

Na sua formulação geral, suponhamos que cada objeto a ser inserido na mochila tenha um benefício  $b_i$  e um peso  $p_i$ . A capacidade máxima da mochila é  $P$ . A formulação matemática deste problema é então dada por:

$$\begin{aligned} \text{maximizar } z &:= \sum_{i=1}^n b_i x_i \\ \text{sujeito a } &\sum_{i=1}^n p_i x_i \leq P \\ &x_i \in \{0, 1\}, \quad i = 1, \dots, n. \end{aligned} \tag{1.2}$$

Uma outra formulação, ligeiramente mais geral que esta é o que considera  $m$  mochilas. Nesse caso, definimos como a variável decisória  $x_{ij}$  como sendo:

$$x_{ij} = \begin{cases} 1, & \text{se o item } i \text{ for colocado na mochila } j, \\ 0, & \text{caso contrário.} \end{cases}$$

Suponhamos que cada mochila tenha capacidade  $P_j$ . Então a formulação matemática que aloca os  $n$  itens dentro destas  $m$  mochilas, com o maior benefício possível e lembrando que cada item pode ser alocado no máximo uma vez, é dada por:

$$\begin{aligned}
 \text{maximizar } z &:= \sum_{i=1}^n \sum_{j=1}^m b_i x_{ij} \\
 \text{sujeito a } & \sum_{i=1}^n p_i x_{ij} \leq P_j, \quad j = 1, \dots, m \\
 & \sum_{j=1}^m x_{ij} \leq 1, \quad i = 1, \dots, n \\
 & x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 1, \dots, m
 \end{aligned} \tag{1.3}$$

Existem outras versões para este problema, como restrições entre diferentes itens, mochilas compartimentadas, etc. A referência (ARENALES, 2007) trata estes caso com maiores detalhes. Embora as formulações (1.2 ou 1.3) são simples de serem escritas e entendidas, este problema é  $NP$ -difícil, isto é, ainda não existem algoritmos que o resolva em tempo polinomial.

Sahni (1975) desenvolveu métodos exatos para resolver este problema. Mais tarde, Sahni (1975) apresenta uma heurística baseada em algoritmos genéticos para o problema da mochila multidimensional, devido à dificuldade em resolver este problema na otimalidade para instâncias mais elevadas.

### 1.1.3 Problema das $p$ -medianas

As primeiras formulações do problema das  $p$ -medianas foram apresentadas em Hakimi (1964) onde demonstrou dois teoremas fundamentais para a obtenção de soluções para o problema das  $p$ -medianas.

Dentre os problemas clássicos de PO, os de localização se destacam pela semelhança com o modelo desenvolvido neste estudo, são eles: Problema das  $p$ -medianas e problema dos  $p$ -centros.

Problema das  $p$ -medianas envolve a localização de  $p$  facilidades e a designação de clientes a estas facilidades de modo a minimizar a soma das distâncias dos clientes às facilidades e tal que cada cliente seja atendido por uma única facilidade. (CHEN; BATSON; DANG, 2011).

Para modelar este problema, definimos as seguintes variáveis decisórias:

$$x_{ij} = \begin{cases} 1, & \text{se o cliente } j \text{ é atendido pela facilidade localizada em } i, \\ 0, & \text{caso contrário.} \end{cases}$$

e

$$y_i = \begin{cases} 1, & \text{se a facilidade é aberta no local } i, \\ 0, & \text{caso contrário.} \end{cases}$$

Se existem  $n$  facilidades e  $m$  clientes, consideramos os conjuntos  $I = \{1, \dots, n\}$  e  $J = \{1, \dots, m\}$  sendo das facilidades e dos clientes, respectivamente. O modelo de programação linear inteira das  $p$ -medianas pode ser escrito da seguinte forma:

$$\begin{aligned} \text{minimize} \quad & z := \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \\ \text{sujeito a} \quad & \sum_{i \in I} x_{ij} = 1 \quad j \in J \\ & x_{ij} \leq y_i \quad i \in I \quad j \in J \\ & \sum_{i \in I} y_i = p \\ & x_{ij} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \quad i \in I \quad j \in J. \end{aligned} \tag{1.4}$$

A função objetivo consiste em minimizar a soma total das distâncias entre clientes e facilidades ( $d_{ij}$ ). O primeiro conjunto de restrições, assegura que cada cliente deve ser atendido por uma única facilidade. O segundo conjunto de restrições garante que cada cliente  $j$  será atendido pela facilidade aberta em  $i$ . O terceiro conjunto de restrições garante que exatamente  $p$  facilidades devem ser abertas.

Note que, se a facilidade  $i$  é instalada, então o cliente  $x_{ij}$  pode ser 0 ou 1, ou seja, o cliente  $j$  pode ou não ser atendido pela facilidade  $i$ . Porém, se a facilidade  $i$  não é instalada, isto é  $y_i = 0$  então  $x_{ij} = 0$  necessariamente, pois nenhum cliente  $j$  poderá ser atendido por nenhuma facilidade  $i$ .

Há diversas aplicações práticas deste modelo. Por exemplo, [Rosário, Carnieri e Steiner \(2002\)](#) utiliza esta formulação para localizar unidades de pronto atendimento de 24 horas, [Lorena et al. \(1999\)](#) aplica em sistemas de localizações geográficas, e em cobertura de arcos ([SMIDERLE; STEINER; WILHELM, 2004](#)).

Dentre os métodos de resolução desenvolvidos, existem autores que optam pela solução por diferentes técnicas heurísticas, como em [Hörner et al. \(2009\)](#), uso da heurística simulated annealing ([CHIYOSHI; GALVAO, 2000](#)) e outros em desenvolver métodos exatos ([SENNE; LORENA, 2003](#)) como relaxação lagrangeana, métodos que exploram uma busca em árvore ([MARANZANA, 1964](#)) e uso do branch-and-bound ([JÄRVINEN; RAJALA; SINERVO, 1972](#)).

#### 1.1.4 O Problema dos $p$ -centros

Este problema envolve a localização de  $p$  facilidades e a designação de clientes a essas  $p$  facilidades de modo a minimizar a distância máxima de clientes às facilidades. Para

formular este problema, considere as variáveis do problema das  $p$ -medianas e a seguinte variável adicional:  $r$  (distância máxima que um cliente percorre). (CHEN; BATSON; DANG, 2011).

O modelo de programação linear inteira dos  $p$ -centros pode ser escrito da seguinte forma:

$$\begin{aligned}
 & \text{minimize} && z := r \\
 & \text{sujeito a} && \sum_{i \in I} x_{ij} d_{ij} \leq r, \quad j \in J \\
 & && \sum_{i \in I} x_{ij} = 1, \quad j \in J \\
 & && x_{ij} \leq y_i, \quad i \in I, \quad j \in J \\
 & && \sum_{i \in I} y_i = p \\
 & && x_{ij} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \quad r \geq 0, \quad i \in I, \quad j \in J.
 \end{aligned} \tag{1.5}$$

Como dito anteriormente, o problema consiste em decidir onde localizar  $p$  centros em uma rede composta por vértices e arestas, de forma a minimizar a maior distância entre a facilidade e o cliente.

Em alguns casos, quando uma demanda estiver associada a cada vértice, pode haver restrições na capacidade de atendimento dos centros (problema de  $p$ -centros com restrições de capacidade). Modelos de localização de facilidades têm sido propostos como ferramentas de auxílio à decisão, principalmente quando é possível usar Sistemas de Informações Geográficas (SIGs) na coleta e na análise dos dados dos problemas (LORENA et al., 2001).

A referência (PLESNÍK, 1987) propõe uma heurística para resolver este problema via métodos exatos, ao passo que Mladenović, Labbé e Hansen (2003) desenvolvem um algoritmo de Busca-Tabu e Busca por Vizinhança Variável especializado para este problema. O artigo recente (CALIK; LABBÉ; YAMAN, 2015) faz uma detalhada revisão destes problemas, como apresenta diferentes aplicações e métodos de resoluções.

## 1.2 Programação inteira

Como visto até o momento, um problema de Programação<sup>1</sup> Linear (PL) é um problema de otimização, em que procura-se encontrar um conjunto de valores contínuos (variáveis respostas) que maximizam ou minimizam uma função linear (função objetivo), ao mesmo tempo que deve satisfazer um conjunto de restrições lineares. Já um problema

<sup>1</sup> O termo “programação”, neste contexto, significa atividades de planejamento que consomem recursos limitados, por exemplo matérias-primas, máquinas, equipamentos, instalações, mão de obra, dinheiro, gestão, tecnologia da informação, etc.

de Programação Linear Inteira (PLI) é um problema de programação linear com a restrição de que pelo menos uma das variáveis seja inteira. Se um PLI apresenta tanto variáveis contínuas como discretas ele é um problema de Programação Inteira Mista e se apresenta apenas variáveis discretas, é um problema de Programação Inteira Puro. Um problema inteiro com variáveis binárias do tipo 0–1 ou decisões do tipo “sim” ou “não” é chamado Problema de Programação Linear Binário. (CHEN; BATSON; DANG, 2011).

Matematicamente, um Problema de Programação Inteira Mista é definido como:

$$\begin{aligned}
 \text{maximizar} \quad & z := \sum_{j=1}^n c_j x_j + \sum_{k=1}^p d_k y_k \\
 \text{sujeito a} \quad & \sum_{j=1}^n a_{ij} x_j + \sum_{k=1}^p g_{ik} y_k \leq b_i \quad i = 1, 2, \dots, m \\
 & x_{ij} \geq 0 \quad j = 1, 2, \dots, n \\
 & y_k \in \mathbb{Z}_+, \quad k = 1, 2, \dots, p.
 \end{aligned} \tag{1.6}$$

em que  $m$  é o número de restrições,  $n$  o número de variáveis contínuas e  $p$  o número de variáveis discretas.

Já o problema de Programação Inteira Pura pode ser escrito como:

$$\begin{aligned}
 \text{maximizar} \quad & z := \sum_{k=1}^p d_k y_k \\
 \text{sujeito a} \quad & \sum_{k=1}^p g_{ik} y_k \leq b_i, \quad i = 1, 2, \dots, m \\
 & y_k \in \mathbb{Z}_+, \quad k = 1, 2, \dots, p.
 \end{aligned} \tag{1.7}$$

Para maiores detalhes, consultar [Chen, Batson e Dang \(2011\)](#).

### 1.2.1 O Método *Branch-and-bound*

O método Branch-and-Bound (B & B) é um método de busca em árvore que se fundamenta na programação linear para explorar o espaço de busca com a intenção de fornecer uma solução inteira para um PLI. Assumimos, nesta explanação, que o problema é inteiro puro. (CHEN; BATSON; DANG, 2011).

Em cada passo do método as variáveis inteiras são relaxadas e o subproblema resultante resolvido por um método da programação linear. Se a solução desse subproblema tiver todas as variáveis inteiras, então os descendentes do ramo da árvore analisado estão, naturalmente, implicitamente enumerados.

Havendo variáveis não inteiras e falhando esse outro critério, é feita a escolha de uma variável a ser ramificada. Escolhida essa variável, deve-se, agora, escolher qual ramo da árvore explorar primeiro, aquele associado ao valor menor ou igual ao piso da

variável, isto é,  $x_i \leq \lfloor x_i \rfloor$  ou o ramo associado ao valor maior ou igual ao teto da variável (isto é, o ramo  $x_i \geq \lceil x_i \rceil$ ).

O algoritmo de B & B funciona no método “divisão e conquista”, assim é necessário que ocorra poda dos nós, ou seja, que em algum momento os problemas parem de gerar subproblemas. A poda dos nós pode ocorrer em três situações:

- (i) **Poda por infactibilidade:** O problema relaxado é infactível;
- (ii) **Poda por otimalidade:** A solução ótima do problema relaxado é inteira;
- (iii) **Poda por qualidade:** O valor de qualquer solução factível do problema relaxado é pior que o valor da melhor solução factível atual.

Há algumas técnicas para escolha dos nós a serem priorizados, a mais aceita é a *backtracking*. Essa técnica busca em profundidade, ou seja, o último nó incluído é o primeiro a ser examinado (*last-in first-out*). Se este nó não for eliminado, o próximo a ser examinado é seu filho; se este nó for eliminado, percorre-se a árvore em direção à raiz até encontrar o próximo nó a ser analisado. Outra técnica é buscar sempre o nó com maior limitante (problema de maximização), a vantagem desta técnica é obter uma árvore com menos ramos, porém não analisa todos os nós, possibilitando inviabilizar a solução. Analisaremos, agora, quando o problema é binário.

### 1.2.2 *Branch-and-Bound* 0-1

Quando o problema envolve variáveis binárias, o método *Branch-and-Bound* também pode ser aplicado. O exemplo abaixo mostra uma aplicação do método em variáveis binárias.

Suponha que temos quatro geradores  $(x_1, x_2, x_3, x_4)$ , cada um com potência 300, 600, 500 e 1600 Watts, respectivamente. Desejamos saber quais deles devem ser ligados de forma a termos pelo menos 700w de potência com custo mínimo. Cada um tem um custo, respectivamente, de 7, 12, 5 e 14 reais, respectivamente. Seja  $x_j$  a variável resposta binária, de tal forma que:

$$x_j = \begin{cases} 1, & \text{se o gerador } j \text{ é ligado,} \\ 0, & \text{caso contrário.} \end{cases}$$

Então, o modelo matemático associado é dado a seguir:

$$\begin{aligned}
 &\text{minimize} && z := 7x_1 + 12x_2 + 5x_3 + 14x_4 \\
 &\text{sujeito a} && 300x_1 + 600x_2 + 500x_3 + 1600x_4 \geq 700 \\
 &&& x_1, x_2, x_3, x_4 \in \{0, 1\}.
 \end{aligned} \tag{1.8}$$

Neste tipo de problema, uma forma de resolver o problema relaxado é por meio da solução parcial, ou seja, uma versão restrita do modelo original, com algumas variáveis fixas. Os complementos factíveis são soluções factíveis do problema candidato correspondente e o valor da função objetivo do melhor complemento factível é o valor ótimo da função objetivo do problema candidato.

Por exemplo, se fixarmos, arbitrariamente,  $x_2 = 1$  e  $x_4 = 0$  temos um novo problema:

$$\begin{aligned}
 &\text{minimize} && z = 7x_1 + 5x_3 \\
 &\text{sujeito a} && 300x_1 + 600x_2 + 500x_3 + 1600x_4 \geq 700 \\
 &&& x_1, x_3 \in \{0, 1\} \\
 &&& x_2 = 1, x_4 = 0.
 \end{aligned}$$

A ideia de resolução parcial segue conforme o algoritmo de B & B, ou seja, se a solução parcial relaxada do problema candidato é infactível, então a solução parcial pode ser podada por infactibilidade. Se for factível com função objetivo pior do que o valor da solução incubente, então a solução parcial pode ser podada por qualidade. Se uma solução ótima de uma relaxação do problema candidato for factível para o modelo original, então ela é o melhor completamento associado à solução parcial e pode ser podado por otimalidade.

Aplicando o método do B & B, no modelo (1.8), temos o esquema de podas apresentado na Figura (1).

Os passos deste método neste exemplo são detalhados a seguir.

- Passo 1: Neste exemplo de *Branch-and-Bound* Binário, a etapa 0 obteve  $x_{*4} = 0,44$  com valor  $z^0 = 6,12$  e como  $x_{*4}$  é uma variável binária foi ramificada para 1 no nó 1 e para 0 no nó 2;
- Passo 2: O nó 1 gerou valores binários com  $z^1 = 14$  e foi podado por otimalidade;
- Passo 3: No nó 2  $x_{*2} = 0,33$  com  $z^2 = 9$  assim  $x_{*2}$  foi ramificado para 1 no nó 3 e 0 no nó 6;

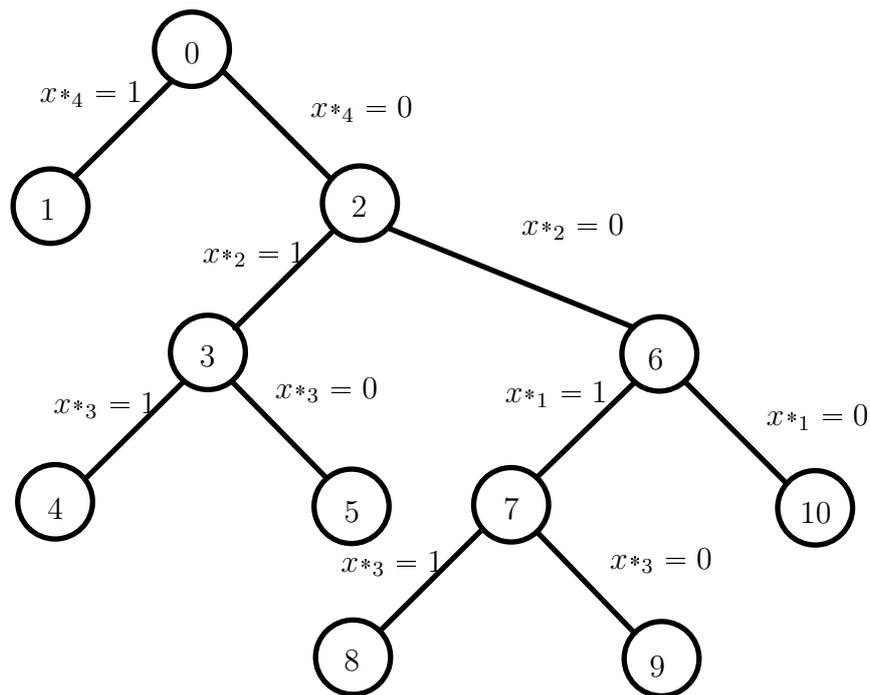


Figura 1 – Branch-and-Bound binário aplicado no problema (1.8)

- Passo 4: No nó 3  $x^*_3 = 0$ , 2 e  $z^3 = 13$  e  $x^*_3$  é ramificado para 1 no nó 4 e 0 no nó 5;
- Passo 5: O nó 4 gerou valores binários em todas as variáveis com  $z^4 = 17$ ;
- Passo 6: O nó 5 gerou  $x^*_1 = 0$ , 33 e  $z^5 = 14,33$  como o nó 4 já foi podado por otimalidade com valor maior então o nó 5 é podado por qualidade;
- Passo 7: o nó 6 gerou  $x^*_1 = 0$ , 67 e  $Z^6 = 9,67$ , e  $x^*_1$  é ramificado em 1 no nó 7 e 0 no nó 10;
- Passo 8: o nó 7 gerou  $x^*_3 = 0$ , 8 e  $z^7 = 11$  e  $x^*_3$  é ramificado em 1 no nó 8 e 0 no nó 9;
- Passo 9: o nó 10, por sua vez, é um problema infactível, gerando poda por infactibilidade;
- Passo 10: O nó 9 gerou problema infactível com poda por infactibilidade e;
- Passo 11: O nó 8 gerou valores binários com  $z^8 = 12$  e foi podado por otimalidade.

Assim, encerra-se o método B & B que tem por objetivo maximizar a função  $z$  e a solução final é o nó 4 com  $z^* = 17$  e  $x^* = (0, 1, 1, 0)^T$ .

Vejamos, agora, alguns truques para obtermos um problema de otimização inteiro linear quando sua formulação original é não-linear. Tais técnicas serão empregadas nesta dissertação.

### 1.2.3 Método de Plano de Corte em Programação Inteira

O método plano de corte nada mais é do que uma restrição de igualdade (ou desigualdade) que corta uma parte fracionária da região viável de um PI sem excluir qualquer solução. Entre os anos de 1950 e 1960 foram desenvolvidos diversos modelos de plano de corte, os mais notórios são os de dupla abordagem: métodos de plano de corte fracionados, duplos e mistos. A principal diferença entre os diversos planos de corte é a forma como a restrição é gerada. (CHEN; BATSON; DANG, 2011).

Os métodos do plano de corte duplo começam com uma ótima solução PI e requer a aplicação de uma série de etapas para re-otimizar e criar novos PLs, cada um adicionando uma ou mais restrições ao quadro simplex atual. Os passos deste procedimento estão resumidos a seguir:

- Passo 1: Resolva o PL inicial. Se a solução for inteira, então pare, o PI também estará resolvido. Caso contrário, siga para o passo 2;
- Passo 2: No tableau do Simplex, selecione uma linha para ser a linha geradora do corte;
- Passo 3: Derive uma restrição de corte da linha geradora, sem “cortar” nenhum ponto inteiro factível e cortando a solução ótima do PL relaxado original;
- Passo 4: Aplicar o método Dual Simplex no problema linear aumentado para restaurar a factibilidade do mesmo;
- Passo 5: Se o novo PL as restrições de integralidade, o PI está resolvido, caso contrário, volte ao Passo 2.

A seguir, ilustraremos como gerar uma restrição que corta sempre a solução do PL relaxado sem excluir nenhum ponto inteiro factível. Tal método se chama *corte de Gomory*.

#### 1.2.3.1 Método de Plano de Corte Fracionário

O método de plano de corte fracionário é aplicável para PIs puros. O processo inicia-se em um tableau simplex otimizado do Problema Linear relaxado:

$$\begin{aligned}
 & \text{maximize} && z + \sum_k d_k y_k = d_0 \\
 & \text{sujeito a} && y_{Bi} + \sum_k g_{ij} y_k = b_i \quad i = 1, \dots, m \\
 & && y_{Bi} \geq 0 \quad i = 1, \dots, m \\
 & && y_k = 0 \quad k = 1, \dots, p,
 \end{aligned} \tag{1.9}$$

em que  $y_{B_i}$  são as variáveis básicas e  $y_k$  as variáveis não-básicas. Seleccionamos a linha geradora nas equações básicas dada por:

$$y_{B_r} + \sum_k g_{rk} y_k = b_r \quad \text{para algum } k = 1, \dots, p. \quad (1.10)$$

Separando a parte fracionária da parte inteira desta linha, temos que:

$$y_{B_r} + \sum_k \{(g_{rk} - [g_{rk}]) + [g_{rk}]\} y_k = (b_r - [b_r]) + [b_r].$$

em que  $[a]$  denota o maior inteiro  $\leq a$ .

Rearrajando (1.10) temos que:

$$\sum_k [f_{rk}] y_k - f_{r0} = [b_r] - \sum_k f_{rk} y_k.$$

Assim, a condição de integralidade é dada por  $f_{r0} - \sum_k k f_{rk} y_k \geq 0$  ou

$$\sum_k k f_{rk} y_k \geq f_{r0}, \quad (1.11)$$

é o *corte fracionário de Gomory*, onde  $f_{rk}$  e  $f_{r0}$  são as partes fracionárias de  $y_{rk}$  e  $y_{B_r}$ , respectivamente.

Para mais informações sobre plano de corte de Gomory veja [Chen, Batson e Dang \(2011\)](#).

**Exemplo 1** (Exemplo de aplicação de cortes de Gomory). Tome o PI a seguir:

$$\begin{aligned} \text{minimize } & z = -3x_1 - 8x_2 \\ \text{sujeito a } & 2x_1 + 5x_2 \leq 11 \\ & 2x_1 - 2x_2 \leq 5 \\ & x_1 \in \mathbb{Z}_+, x_2 \in \mathbb{Z}_+. \end{aligned}$$

Incluindo as variáveis de folga e relaxando as condições de integralidade deste problema, temos que:

$$\begin{aligned} \text{minimize } & z = -3x_1 - 8x_2 \\ \text{sujeito a } & 2x_1 + 5x_2 + x_3 = 11 \\ & 2x_1 - 2x_2 + x_4 = 5 \\ & x_i \geq 0, \quad i = 1, \dots, 4. \end{aligned}$$

A solução do problema relaxado é:  $(x_1^*, x_2^*, x_3^*, x_4^*) = (0, \frac{11}{5}, 0, \frac{47}{5})$ , de modo que os índices da base ótima são:  $\mathcal{B} = \{2, 4\}$ , isto é, a matriz básica ótima é:

$$B = \begin{bmatrix} 5 & 0 \\ -2 & 1 \end{bmatrix}.$$

Desta maneira, as equações básicas no ponto ótimo são dadas por:

$$\begin{aligned} x_2 + \frac{2}{5}x_1 + \frac{1}{5}x_3 &= \frac{11}{5} \\ x_4 + \frac{14}{5}x_1 + \frac{2}{5}x_3 &= \frac{47}{5}. \end{aligned}$$

Agora, geramos um corte na linha  $i = 2$ , usando a expressão (1.11). Calculando os elementos fracionários, temos que:

$$f_0 = y_{20} - \lfloor y_{20} \rfloor = \frac{11}{5} - \lfloor \frac{11}{5} \rfloor = \frac{1}{5},$$

$$f_1 = \frac{2}{5} - \lfloor \frac{2}{5} \rfloor = \frac{2}{5},$$

$$f_3 = \frac{1}{5} - \lfloor \frac{1}{5} \rfloor = \frac{1}{5},$$

Desta forma o corte de Gomory será dado por:

$$\frac{2}{5}x_1 + \frac{1}{5}x_3 \geq 2$$

ou, em termos das variáveis básicas e após algumas manipulações, resulta na restrição  $x_2 \leq 2$  ou  $x_2 + x_5 = 2$ , com  $x_5 \geq 0$

Inserindo esta restrição no PL,

$$\begin{aligned} \text{minimize } z &= -3x_1 - 8x_2 \\ \text{sujeito a } 2x_1 + 5x_2 + x_3 &= 11 \\ 2x_1 - 2x_2 + x_4 &= 5 \\ x_2 + x_5 &= 2 \\ x_i &\geq 0, \quad i = 1, \dots, 5. \end{aligned}$$

Temos que a sua solução é o vetor  $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (1/2, 2, 0, 8, 0)$ , de modo que a base ótima é o conjunto de índices  $\mathcal{B} = \{1, 2, 4\}$ . As equações básicas no ótimo são dadas por:

$$\begin{aligned} x_1 + \frac{1}{2}x_3 - \frac{5}{2}x_5 &= \frac{1}{2} \\ x_2 + x_5 &= 2 \\ x_4 - x_3 + 7x_5 &= 8 \end{aligned}$$

Gerando o corte na linha  $i = 1$ , vamos calcular os elementos fracionários:

$$f_0 = y_1 0 - [y_1 0] = \frac{1}{2} - 0 = \frac{1}{2},$$

$$f_3 = \frac{1}{2} - \left\lfloor \frac{1}{2} \right\rfloor = \frac{1}{2},$$

$$f_5 = \frac{5}{2} - \left\lfloor \frac{5}{2} \right\rfloor = \frac{1}{2}.$$

Assim, o segundo corte de Gomory será dado por:

$$\frac{1}{2}x_3 + \frac{1}{2}x_5 \geq \frac{1}{2},$$

ou

$$x_3 + x_5 - x_6 = 1,$$

com  $x_6 \geq 0$ .

Agora resolvemos o PL

$$\begin{aligned} \text{minimize} \quad & z = -3x_1 - 8x_2 \\ \text{sujeito a} \quad & 2x_1 + 5x_2 + x_3 = 11 \\ & 2x_1 - 2x_2 + x_4 = 5 \\ & x_2 + x_5 = 2 \\ & x_3 + x_5 - x_6 = 1 \\ & x_i \geq 0, \quad i = 1, \dots, 6. \end{aligned}$$

encontramos a solução ótima  $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (3, 1, 0, 1, 1, 0)$ . Como é uma solução inteira, o problema inteiro está resolvido.

## 1.3 “Truques” de Programação inteira

### 1.3.1 Linearizando polinômios 0–1

Para que as técnicas de programação linear possam ser aplicadas, é necessário um problema linear e isso nem sempre ocorre. Porém, existem técnicas de linearização e um problema não-linear muitas vezes pode ser linearizado (CHEN; BATSON; DANG, 2011).

Se as variáveis do problema são estritamente binárias, i.e. valem 0 ou 1, então qualquer multiplicação entre elas vale 0 ou 1, ou seja  $y_i^n = y_i$  para qualquer  $i$  e para qualquer  $n$ .

**Proposição 1** (Propriedade). O produto  $z = y_1 y_2 \cdots y_n$  com  $y_j \in \{0, 1\}, \forall j = 1, 2, \dots, n$  é equivalente a:

$$\sum_{i=1}^n y_j \leq y_Q + (n - 1),$$

$$\sum_{i=1}^n y_j \geq n y_Q$$

$$y_j, y_Q \in \{0, 1\}.$$

onde  $y_Q$  é uma nova variável. Note que se qualquer  $y_j = 0$  então a primeira condição é redundante, a segunda condição se torna  $y_Q \leq 0$ , e então  $y_Q = 0$ . Se todos  $y_j = 1$ , então a primeira condição implicará  $y_Q \geq 1$  e a igualdade ocorrerá graças à segunda restrição, como desejado.

Considere o seguinte problema não-linear inteiro:

$$\begin{aligned} \text{maximize} \quad & z := 2y_1 y_2 y_3^2 + y_1^2 y_2 \\ \text{Sujeito a} \quad & -y_1 + 2y_2 \leq 5 \\ & 12y_1 + 7y_2^2 y_3 - 3y_1 y_3 \leq 16 \\ & y_i \in \{0, 1\}, i = 1, 2, 3. \end{aligned}$$

A linearização ocorre da seguinte forma:

1. Substitui-se todos os expoentes maiores que 1 por 1;
2. Substitui-se todos termos multiplicativos por uma única variável binária.

Assim, sejam as variáveis:  $y_{123} = y_1 y_2 y_3, y_{12} = y_1 y_2, y_{23} = y_2 y_3$  e  $y_{13} = y_1 y_3$ . Por exemplo, o produto  $2y_1 y_2 y_3^2$  é equivalente a  $2y_1 y_2 y_3$  e que linearizado se transforma em

$$3y_{123} \leq y_1 + y_2 + y_3 \leq y_{123} + 2,$$

Note que, se  $y_1$  ou  $y_2$  ou  $y_3$  for 0, então  $y_{123}$  não pode ser 1, pois  $y_1 + y_2 + y_3 \geq 3y_{123}$ . Por outro lado, se  $y_1 = y_2 = y_3 = 1$ , então  $y_{123}$  deverá ser obrigatoriamente 1.

Fazendo o mesmo para todas as expressões não lineares, temos um problema equivalente mas linear dado por:

$$\begin{aligned}
\text{maximize} \quad & z := 2y_{123} + y_{12} \\
\text{Sujeito a} \quad & 12y_1 + 7y_{23} - 3y_{13} \leq 16 \\
& y_1 + y_2 + y_3 \geq 3y_{123} \\
& y_1 + y_2 + y_3 \leq y_{123} + 2 \\
& y_1 + y_2 \geq 2y_{12} \\
& y_1 + y_2 \leq y_{12} + 1 \\
& y_2 + y_3 \geq 2y_{23} \\
& y_2 + y_3 \leq y_{23} + 1 \\
& y_1 + y_3 \geq 2y_{13} \\
& y_1 + y_3 \leq y_{13} + 1,
\end{aligned}$$

que pode ser resolvido usando um solver de PLI.

### 1.3.2 Transformando produto de variáveis binárias por contínuas

Em muitas situações, teremos que escrever o produto  $y \cdot x$  onde  $y$  é binária e  $x$  é real. Supomos ainda que  $x \leq M$ . Para tanto, procedemos da seguinte forma: seja  $z$  uma nova variável, de modo que  $z = x \cdot y$ . Inserimos novas restrições do tipo:

$$\begin{aligned}
z &\leq My, \\
z &\leq x, \\
x - M(1 - y) &\leq z,
\end{aligned} \tag{1.12}$$

Se  $y = 0$ , então as restrições anteriores forçam  $z = 0$ . Caso contrário, se  $y = 1$ , então as mesmas restrições anteriores forçam  $z = x \leq M$ .

Se o objetivo é linearizar variáveis inteiras, por exemplo,  $y_1 \cdot y_2$ , suponhamos que  $y_1 \leq M$ . Em seguida, podemos escrever a decomposição binária de  $y_1$ , isto é:

$$y_1 = \sum_{i=1}^{\theta} 2^{i-1} \alpha_i,$$

onde  $\alpha_i \in \{0, 1\}$  e  $\theta = \lceil \log_2(M) \rceil + 1$ . Daí temos:

$$y_2 \cdot y_1 = y_2 \cdot \sum_{i=1}^{\theta} 2^{i-1} \alpha_i = \sum_{i=1}^{\theta} 2^{i-1} (y_2 \cdot \alpha_i),$$

e agora, para linearizar  $y_2 \cdot \alpha_i$  empregamos as desigualdades (1.12) (CHEN; BATSON; DANG, 2011).

### 1.3.3 Representação de valores discretos

Suponha que em um certo problema a variável  $x$  assumira apenas os valores do conjunto  $X = \{1, 4, 6, 8, 10, 24\}$ . Para representar esta condição, definimos as variáveis binárias  $y_i$ ,  $i = 1, \dots, 6$  e as restrições:

$$x = 2y_1 + 4y_2 + 6y_3 + 8y_4 + 10y_5 + 24y_6$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 = 1.$$

Dessa forma,  $x$  assumirá apenas um dos valores do conjunto  $X$ .

### 1.3.4 Custo fixo

Suponha que uma determinada variável  $x_j$  se nula, não possui custo de confecção (setup, por exemplo) mas, a partir do momento que é estritamente positiva e independentemente do valor, tem um custo fixo  $c_j$  associado. Podemos formular matematicamente esta situação da seguinte maneira, via variáveis binárias:

$$\begin{aligned} \text{Minimize} \quad & z = \sum_{j=1}^n c_j y_j \\ \text{sujeito a} \quad & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m \\ & x_j \leq M y_j, \quad j = 1, 2, \dots, n \\ & x_j \geq 0, y_j \in \{0, 1\}, \end{aligned}$$

isto é, se  $y_j = 0$  implica  $x_j = 0$  e caso contrário, o segundo conjunto de restrições será redundante.

### 1.3.5 Restrições Disjuntivas e do tipo “se-então”

Considere as desigualdades

$$f_1(x_1, \dots, x_n) \leq b_1$$

e

$$f_2(x_1, \dots, x_n) \leq b_2.$$

Se desejarmos que apenas uma das desigualdades esteja ativada, definimos a variável binária  $y$ , que será 1 se  $f_1 \leq b_1$  e 0 se  $f_2 \leq b_2$ . Isso pode ser representado por:

$$f_1(x_1, \dots, x_n) - b_1 \leq M(1 - y)$$

$$f_2(x_1, \dots, x_n) - b_2 \leq My.$$

Considere agora uma situação em que, se  $f_1(x_1, \dots, x_n) \leq b_1$  então  $f_2(x_1, \dots, x_n) \leq b_2$ . Esta condição, pelo princípio da lógica, é equivalente a

$$f_1(x_1, \dots, x_n) > b_1 \quad \text{ou} \quad f_2(x_1, \dots, x_n) \leq b_2$$

ou ainda

$$-f_1(x_1, \dots, x_n) \leq -b_1 \quad \text{ou} \quad f_2(x_1, \dots, x_n) \leq b_2.$$

Aplicando a mesma transformação vista anteriormente, podemos modelar estas duas restrições como:

$$\begin{aligned} -f_1(x_1, \dots, x_n) &\leq -b_1 + My \\ f_2(x_1, \dots, x_n) &\leq b_2 + M(1 - y), \end{aligned}$$

onde  $y \in \{0, 1\}$ .

(CHEN; BATSON; DANG, 2011). Na próxima seção iremos dar algumas noções da área de otimização multi-objetivo.

## 1.4 Otimização Multi-objetivo

Como já dito anteriormente, a maioria dos problemas na vida prática são simultâneos e multi-objetivos, principalmente quando existe um certo grau de conflito entre os objetivos, isto é, quando um objetivo é individualmente minimizado, o outro é piorado. Dito de outra forma, não existe uma solução que minimiza ambos os objetivos ao mesmo tempo. Nesse caso, a solução deste problema multi-objetivo será dada por um conjunto não unitário de soluções, ditas eficientes no sentido de Pareto, de maneira que um objetivo não pode ser minimizado sem aumentar o outro e vice-versa (FILHO et al., 2016). Nesta seção apresentamos definições, nomenclaturas e estratégias de otimização para um problema multi-objetivo.

### 1.4.1 Definições e nomenclatura

O Problema de Otimização multi-objetivo (POM), como o próprio nome sugere, é uma problema com mais de um objetivo. Além disso, assume-se que esses objetivos são conflitantes entre si, isto é, minimizar um implica maximizar o outro e assim por diante.(FILHO et al., 2016).

**Definição 2** (Problema de Otimização Multi-objetivo). Um POM é um problema com  $r > 1$  funções-objetivos conflitantes, sujeito a um conjunto viável  $\mathcal{X} \subseteq \mathbb{R}^n$  é definido por:

$$\begin{aligned} \text{minimize} \quad & z := (f_1(x), \dots, f_n(x)) \\ \text{sujeito a} \quad & x \in \mathcal{X} = \{x \in \mathbb{R}^n \mid g_j(x) \leq 0, j = 1, \dots, l\}, \end{aligned}$$

em que,

- $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  é o vetor das variáveis decisórias;
- $f_i(x)$  é a  $i$ -ésima função objetivo a ser minimizada,  $i = 1, \dots, r$ ;
- $\mathcal{X} \subseteq \mathbb{R}^n$  é o espaço factível de decisões gerado pelas  $l$  restrições  $g_j(x) \leq 0$ .

Como são  $r$  funções-objetivos conflitantes entre si, não existe um ponto ótimo  $x^*$  que minimize todas as funções ao mesmo tempo. A estratégia, então, é minimizar individualmente cada uma das funções-objetivos e construir o vetor ideal:

**Definição 3** (Vetor ótimo). O vetor  $z^I = (z_1^I, \dots, z_r^I)^T \in \mathbb{Z}$  é dito Ideal quando sua  $k$ -ésima componente é o valor mínimo do problema mono-objetivo:

$$\begin{aligned} & \text{minimize} && z_k := f_k(x) \\ & \text{sujeito a} && x \in \mathcal{X} = \{x \in \mathbb{R}^n \mid g_j(x) \leq 0, j = 1, \dots, l\}. \end{aligned}$$

para todo  $k = 1, \dots, r$ .

Ou seja, o vetor ideal é a solução individual de cada uma das funções-objetivo do POM. A comparação entre essas soluções é feita por relações de dominância <sup>2</sup>.

**Definição 4** (Dominância). Uma solução  $x^1$  domina uma outra solução  $x^2$ , ambas pertencentes a  $\mathcal{X}$ , se satisfazem as condições abaixo:

- A solução  $x^1$  não é pior que a solução  $x^2$  em todos os objetivos, ou seja,  $f_k(x^1) \leq f_k(x^2)$ ,  $k = 1, \dots, r$ .
- A solução  $x^1$  é estritamente melhor que a solução  $x^2$  em pelo menos uma componente  $k^* \in \{1, \dots, r\}$ , isto é,  $f_{k^*}(x^1) < f_{k^*}(x^2)$ .

É conhecido como “fronteira de Pareto”, em um problema multi-objetivo, o conjunto de pontos em uma fronteira <sup>3</sup> mais inferior e à esquerda do espaço  $\mathbb{Z}$ . O conjunto não dominado é o mais inferior, ou seja, uma solução eficiente. Isto é, se  $x^*$  é uma solução eficiente, então:

- Existe pelo menos um  $x \in \nu$  tal que  $x^* \leq x$ ;
- Para todo  $x^* \neq y^* \in \nu$ ,  $x^*$  não é dominado por  $y^*$  e vice-versa.

A literatura específica na área de Otimização Multi-objetivo relata os métodos de Soma Ponderada e  $\varepsilon$ -Restrito como os métodos mais utilizadas.

<sup>2</sup> Notação:  $x^1 \leq x^2$  então  $x^1$  domina  $x^2$  ou  $x^2$  é dominado por  $x^1$ .

<sup>3</sup> A fronteira só é possível ser vista quando  $r \leq 3$

### 1.4.2 Método da Soma Ponderada

O Método da Soma Ponderada transforma o problema multi-objetivo em mono-objetivo escalonando as funções-objetivo de acordo com pesos atribuídos a cada uma delas. É óbvio que quanto mais (ou menos) importante for aquela função-objetivo no problema, maior (ou menor) será o peso atribuído a ela. Por ser muito simples e fácil de aplicar, esse é o método mais difundido na literatura. Este será o método que será aplicado para o problema relatado nesta dissertação. Para mais detalhes e profundidade sobre métodos de escalarizações em otimização multi-objetivo, consultar as referências [Miettinen \(1999\)](#), [Ehrgott \(2000\)](#) e [Filho et al. \(2016\)](#).

Assim, um POM solucionado por meio do Método da Soma Ponderada é escrito como:

$$\begin{aligned} \text{minimize} \quad & z^w := \sum_{k=1}^r w_k \cdot f_k(x) \\ \text{sujeito a} \quad & x \in \mathcal{X}, \end{aligned} \tag{1.13}$$

em que, como a solução do problema (1.13) não se altera quando a função objetivo é dividida por um contraste positivo pode-se considerar sem perda de generalidade,  $\sum_{k=1}^r w_k = 1$  e  $w_k \in [0, 1]$ . Assumimos que as funções  $z_k = f_k(x)$  já estejam normalizadas.

Algumas propriedades e teoremas desta técnica são apresentados a seguir.

**Teorema 5.** Se  $w_k > 0$  para todo  $k = 1, \dots, r$ , então a solução do Problema (1.13) é eficiente para o POM original.

*Demonstração.* Seja  $x^*$  a solução ótima única do problema (1.13), por definição  $\langle w, f(x^*) \rangle < \langle w, f(x) \rangle, \forall x$ . Suponha que  $x^*$  não é eficiente, isto é, existe um  $x$  tal que  $f(x) \leq f(x^*)$  e  $f(x) \neq f(x^*)$ . Como  $w \geq 0$ , isto implicaria contradizer a unicidade de solução  $x^*$ , e portanto  $x^*$  é eficiente.  $\square$

**Teorema 6.** Se o Problema (1.13) tem solução uma única  $x^*$ , então  $x^*$  é eficiente.

*Demonstração.* Este Teorema pode ser demonstrado utilizando-se argumentos similares ao anterior.  $\square$

A recíproca do Teorema (5), em geral, não é verdadeira. Dito de outra forma, dada qualquer solução eficiente  $x^*$ , não existe um vetor de pesos  $w$  de modo que  $x^*$  possa ser obtida como solução ótima do Problema (1.13). Contudo, para problemas multi-objetivos convexos a recíproca do teorema anterior é verdadeira.

**Teorema 7.** Se  $x^*$  é uma solução eficiente para o problema multi-objetivo (1.13) convexo, então existe um vetor  $w = (w_1, \dots, w_r)^T \geq 0$  com  $\sum_{k=1}^r w_k = 1$ , tal que  $x^*$  é solução do Problema (1.13).

*Demonstração.* Apresentada em Ehrgott (2000). □

### 1.4.3 Método do $\varepsilon$ -Restrito

A fim de relevar as dificuldades do método da Soma Ponderada, principalmente para POM com fronteira de Pareto não convexa, a técnica  $\varepsilon$ -Restrito é um método alternativo e que supera algumas dificuldades dos métodos precedentes. Proposta em Haimes, Lasdon e Wismer (1971), este método escalariza um POM tomando uma função como objetivo e restringindo as demais com limitantes especificados. Quando esses limitantes variam de modo adequado, soluções eficientes podem ser obtidas. O Problema Restrito é definido como:

$$\begin{aligned} & \text{minimize} && z^\varepsilon := f_i(x) \\ & \text{sujeito a} && x \in \mathcal{X}, \\ & && f_k(x) \leq \varepsilon_k, \quad k = 1, \dots, r, \quad k \neq i, \end{aligned} \tag{1.14}$$

onde  $i \in \{1, \dots, r\}$  é alguma função-objetivo previamente escolhida,  $\varepsilon_k$  é um limitante superior para a função-objetivo  $f_k(x)$ ,  $k \neq i$ . Conforme estes limitantes variam, diferentes soluções eficientes podem ser obtidas com esta escalarização. Uma diferença crucial entre os dois primeiros métodos aqui abordados é que a região admissível do problema escalarizado varia, enquanto na outra abordagem apenas a função-objetivo sofre alterações.

Conforme o lado direito da restrição  $f_k(x) \leq \varepsilon$  é modificado, a minimização de  $f_i(x)$  leva à uma solução eficiente, desde que a solução do Problema Restrito seja única. Se  $\varepsilon$  assumir valores inteiros no intervalo  $\mathcal{I} = [z_k^-, z_k^+]$  com  $z_k^-$  e  $z_k^+$  os valores de máximo e mínimo da função objetivo  $k$ . Note que se  $\varepsilon$  não estiver neste intervalo, o Problema Restrito pode ser infactível.

Os teoremas a seguir fornecem os suportes teóricos deste procedimento.

**Teorema 8.** A solução do Problema (1.14) é eficiente.

*Demonstração.* Seja  $x^* \in \mathcal{X}$  uma solução para o Problema Restrito. Suponha que  $x^*$  não seja eficiente, logo existe uma outra solução  $\bar{x} \neq x^*$ , de modo que  $f_k(\bar{x}) < f_k(x^*) \leq \varepsilon_k$ , para todo  $k = 1, \dots, r$ . Logo  $\bar{x}$  é admissível para o Problema Restrito. Da última desigualdade, podemos afirmar em particular que  $f_i(\bar{x}) < f_i(x^*)$ , o que é uma contradição, pois  $x^*$  era solução ótima de (1.14). Assim,  $x^*$  é fracamente eficiente. □

**Teorema 9.** Um vetor  $x^*$  é eficiente se, e somente se, é solução do Problema (1.14) para todo  $i = 1, \dots, r$  e  $\varepsilon_k = f_k(x^*)$ ,  $k \neq i$ .

*Demonstração.* ( $\Rightarrow$ ) Seja  $x^*$  eficiente, devemos mostrar que este vetor é solução do Problema Restrito para qualquer escolha para  $i$ . Para tanto, suponhamos que  $x^*$  não seja solução para este problema para algum  $i \in \{1, \dots, r\}$  e  $\varepsilon_k = f_k(x^*)$  para  $k = 1, \dots, r$  e  $k \neq i$ . Então, deve existir uma outra solução  $\mathcal{X} \ni \bar{x} \neq x^*$  tal que  $f_i(\bar{x}) < f_i(x^*)$  e  $f_k(\bar{x}) \leq f_k(x^*)$  para  $k \neq i$ . Isto contradiz a eficiência de  $x^*$ . Noutras palavras,  $x^*$  é solução para o Problema Restrito independente da função-objetivo tomada.

( $\Leftarrow$ ) Se  $x^* \in \mathcal{X}$  é solução do Problema (1.14) para todo  $i = 1, \dots, r$ , então não existe um  $\bar{x} \in \mathcal{X}$  tal que  $f_i(\bar{x}) < f_i(x^*)$  e  $f_k(\bar{x}) \leq f_k(x^*)$  para  $k \neq i$ . Isto é exatamente a definição de eficiência para  $x^*$ .  $\square$

O Teorema (9) é útil porque não é necessário se preocupar qual função-objetivo considerar no Problema Restrito. O resultado a seguir alerta sobre o caso de quando temos múltiplas soluções neste procedimento.

**Teorema 10.** Se o Problema (1.14) para algum  $i = 1, \dots, r$  com  $\varepsilon_k = f_k(x^*)$  para uma dada solução  $x^* \in \mathcal{X}$  tiver única solução, então  $x^*$  é eficiente.

*Demonstração.* Considere  $x^*$  única solução do Problema Restrito. Assumimos que ela não seja eficiente. Isto implica que existe uma outra solução  $\mathcal{X} \ni \bar{x} \neq x^*$  tal que  $f_k(\bar{x}) \leq f_k(x^*)$  para todo  $k = 1, \dots, r$  e para ao menos um índice  $i$ , vale  $f_i(\bar{x}) < f_i(x^*)$ . No entanto,  $x^*$  é única solução, logo ela deve ser factível, de sorte que  $f_k(\bar{x}) \leq \varepsilon_k = f_k(x^*)$ ,  $k \neq i$  e também  $f_i(x^*) < f_i(\bar{x})$ . Tem-se aqui duas desigualdades conflitantes, logo  $x^*$  deve ser eficiente.  $\square$

Assim, para se ter certeza que uma solução produzida por este procedimento seja fortemente eficiente, é necessário garantir a unicidade da solução fornecida pelo Problema Restrito ou resolver  $r$  Problemas Restritos (cada um com uma função-objetivo diferente). Em geral, é difícil checar se a solução do subproblema é única. Do ponto de vista computacional, múltiplas soluções podem ser evitadas se o Problema Restrito for ligeiramente modificado da seguinte maneira:

$$\begin{aligned} \text{minimize} \quad & z^\varepsilon := f_i(x) + \rho \cdot \sum_{\substack{k=1 \\ k \neq i}}^r f_k(x) \\ \text{sujeito a} \quad & x \in \mathcal{X}, \\ & f_k(x) \leq \varepsilon_k, \quad k = 1, \dots, r, \quad k \neq i, \end{aligned} \tag{1.15}$$

onde  $\rho > 0$ . Esta soma adicional na função-objetivo visa eliminar múltiplas soluções e “forçar” a eficiência da solução encontrada.

## Capítulo 2

# Modelos matemáticos desenvolvidos

Nas próximas seções, iremos apresentar modelos não-lineares de programação inteira mono e bi-objetivo, que aloca um conjunto de facilidades (indústrias, fábricas, centro de distribuições, matrizes, etc...) e lojas (comércio, filiais, pontos consumidores, etc...). No primeiro modelo, de dois objetivos, minimizamos duas distâncias: entre a facilidade e o centro consumidor e entre as facilidades. No segundo modelo, de um único objetivo, minimizamos apenas a distância entre as facilidades e os clientes. Os modelos ainda preveem o atendimento da demanda de cada ponto e respeita a capacidade máxima de distribuição da facilidade. Falando da primeira formulação, temos funções não-lineares (na verdade, bi e tri lineares) e que são conflitantes entre si, isto é, a minimização de uma distância acarreta o aumento da outra e vice-versa, de modo que para resolver este problema, necessitamos empregar algum método de escalarização multi-objetivo para gerar um conjunto de *soluções eficientes*. A seguir, a apresentação destes dois modelos.

### 2.1 Modelo matemático de alocação dos clientes e facilidades

Suponhamos que em um centro urbano, há  $r$  lugares disponíveis (nós, terrenos, lotes, quadras, nós, etc) para comportar uma facilidade, centro de distribuição ou ficar vago. Suponha que queiramos escolher no máximo  $p$  facilidades para serem alocadas nestes lugares e que existam exatamente  $q$  clientes (onde  $p + q \leq r$ ) com demandas  $dem_i$  conhecidas em um certo período. Entre cada lugar  $k$ , conhecemos as distâncias entre eles, digamos,  $d_{k\bar{k}}$  é a distância entre os lugares  $k$  e  $\bar{k}$ . Ainda: cada filial só deve receber suas demandas de apenas uma facilidade. Porém, uma mesma facilidade pode atender a mais de um cliente. Este problema tem uma conexão com o famoso Problema das  $p$ -medianas visto no capítulo anterior.

O primeiro objetivo deste problema é minimizar a soma de todas as possíveis distâncias entre as facilidades e os clientes (rotas). Vamos denotar este objetivo de  $z_1$ .

O segundo objetivo a ser considerado é minimizar a soma de todas as possíveis

distâncias entre todas as facilidades. Denotemos este segundo objetivo por  $z_2$ .

Nota-se um certo grau de conflito entre  $z_1$  e  $z_2$ . Vamos entender melhor isto. Se  $z_1$  for minimizado, então o modelo vai tentar aproximar as  $q$  filiais de suas facilidades. Ao fazer isto, os nós próximos a cada facilidade serão ocupados por clientes, e isto faz com que a distância entre as facilidades fique grande. A modelagem leva a uma formulação não-linear para  $z_1$ . Às vezes isso pode representar uma situação ruim, pois pode haver um problema de logística associado aos caminhões, funcionários, mão-de-obra terceirizada que dificulta a transição entre facilidades longínquas.

Por outro lado, se formos minimizar  $z_2$  (e sem se preocupar com  $z_1$ ), concentramos em nós próximos todas as facilidades, mantendo-as em regiões vizinhas. Dessa forma, a soma total das distância entre cada facilidade e seu ponto de atendimento ficará grande, prejudicando também a frota de caminhões, atendimento pontual, emergencial, entre outros fatores. Ademais, o objetivo  $z_2$  também é uma função não-linear a variáveis binárias, como veremos mais adiante. Do ponto de vista matemático, torna-se um problema de grande interesse, além de ser um problema de otimização bi-objetivo, isto é, não existe uma solução que minimiza  $z_1$  e  $z_2$  concomitantemente. Nosso objetivo neste trabalho, será em encontrar um conjunto de soluções para este problema, todas igualmente importantes, que estabelecem um *trade-off* entre estes objetivos (*soluções eficientes*). As funções  $z_1$  e  $z_2$  aplicadas em cada solução eficiente, produzem uma fronteira no espaço  $\mathbb{R}^2$ , denominada *fronteira de Pareto*, como discutido no capítulo anterior. O decisor pode olhar a fronteira, escolher um ponto da mesma, e tomar uma decisão entre escolher uma ou outra alternativa em ocupar estes nós com facilidades, filiais ou deixá-los vago.

Para definir e apresentar o modelo matemático que propomos, sejam:

- $i$ : índice das facilidades;
- $j$ : índice dos clientes;
- $k$ : índice do local (nós);
- $p$ : número de facilidade máximo a serem criadas;
- $q$ : número de clientes a serem atendidos;
- $r$ : número disponível de lugares para alocar as facilidades/clientes;
- $f_i$ : custo para montar a facilidade  $i$ ;
- $C_i$ : capacidade máxima de atendimento da facilidade  $i$ ;
- $dem_j$ : demanda do cliente  $j$ ;
- $d_{k\bar{k}}$ : distância entre os locais  $k$  e  $\bar{k}$ .

Variáveis decisórias são:

$$x_{ij} = \begin{cases} 1, & \text{se a facilidade } i \text{ atenderá o cliente } j \\ 0, & \text{caso contrário,} \end{cases}$$

para  $i = 1, \dots, p$  e  $j = 1, \dots, q$ .

$$y_{ik} = \begin{cases} 1, & \text{se a facilidade } i \text{ é alocada no local } k \\ 0, & \text{caso contrário,} \end{cases}$$

para  $i = 1, \dots, p$  e  $k = 1, \dots, r$ .

$$z_{jk} = \begin{cases} 1, & \text{se o cliente } j \text{ é alocado no local } k \\ 0, & \text{caso contrário,} \end{cases}$$

para  $j = 1, \dots, q$  e  $k = 1, \dots, r$ .

$$w_i = \begin{cases} 1, & \text{se a facilidade } i \text{ é instalada} \\ 0, & \text{caso contrário,} \end{cases}$$

para  $i = 1, \dots, p$ .

A formulação matemática não-linear bi-objetiva é dada por (2.1)-(2.15).

A primeira função objetivo (2.1) visa minimizar a soma das distâncias entre as facilidades e os clientes instalados. Note que é uma função tri-linear, uma vez que esta distância apenas será somada, efetivamente, se existir um arco entre os nós  $k$  e  $\bar{k}$ ,  $y_{ik} = 1$  e  $z_{j\bar{k}} = 1$ . Veremos mais adiante uma maneira de escrevê-la de maneira linear.

A segunda função objetivo (2.2) é também não-linear e minimiza a soma de todas as distâncias entre as facilidades instaladas. É claro que a soma só irá efetivamente acrescentar termos não nulos se  $y_{ik} = 1$  e  $y_{i\bar{k}} = 1$ , em qualquer outra situação o termo desta soma será nulo. A soma mínima  $z_2$  será quando as facilidades estão localizadas da maneira mais “amontoadas” possível.

As restrições (2.3) forçam que cada cliente receba demanda de uma única facilidade. As restrições (2.4) e (2.5) impõem que, a cada nó, uma única facilidade e cliente sejam instalados, respectivamente. As restrições (2.6) e (2.7) obrigam que todas as facilidades e clientes sejam instalados no máximo uma vez em todos os nós, respectivamente. As restrições (2.8) e (2.9) ligam as variáveis  $w_i$  com  $y_{ik}$ , no seguinte sentido: se  $w_i = 0$  então  $y_{ik} = 0$ . Porém, se  $w_i = 1$ , então o atendimento da facilidade  $i$  não pode ultrapassar sua capacidade.

As restrições (2.10) ligam as variáveis  $y_{ik}$  com  $x_{ij}$  no seguinte senso: se a facilidade  $i$  não for instalada ( $\sum_{k=1}^r y_{ik} = 0$ ), então ela não pode atender nenhum cliente (ou

seja  $x_{ij} = 0$ ). A mesma situação é para as restrições (2.11): se o cliente  $j$  não for instalado, então este não pode atendido.

As restrições (2.12) impõem um limite máximo de facilidades enquanto (2.13) impõem que devem haver exatamente  $p$  clientes em toda rede. As restrições (2.14) dizem que, em cada nó, não pode haver um cliente e uma facilidade instalada ao mesmo tempo. Finalmente, (2.15) é o domínio de todas as variáveis decisórias deste problema.

$$\text{minimize } z_1 = \sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r \sum_{\bar{k}=1}^r d_{k\bar{k}} \cdot x_{ij} \cdot y_{ik} \cdot z_{j\bar{k}} \quad (2.1)$$

$$\text{minimize } z_2 = \sum_{i=1}^p \sum_{\bar{i}=1}^p \sum_{k=1}^{r-1} \sum_{\bar{k}=k+1}^r d_{k\bar{k}} \cdot y_{ik} \cdot y_{\bar{i}\bar{k}} \quad (2.2)$$

$$\text{sujeito a } \sum_{i=1}^p x_{ij} = 1, \quad j = 1, \dots, q, \quad (2.3)$$

$$\sum_{i=1}^p y_{ik} \leq 1, \quad k = 1, \dots, r, \quad (2.4)$$

$$\sum_{j=1}^q z_{jk} \leq 1, \quad k = 1, \dots, r, \quad (2.5)$$

$$\sum_{k=1}^r y_{ik} \leq 1, \quad i = 1, \dots, p, \quad (2.6)$$

$$\sum_{k=1}^r z_{jk} \leq 1, \quad j = 1, \dots, q, \quad (2.7)$$

$$\sum_{k=1}^r y_{ik} \leq w_i, \quad i = 1, \dots, p, \quad (2.8)$$

$$\sum_{j=1}^q dem_j \cdot x_{ij} \leq C_i \cdot w_i, \quad i = 1, \dots, p, \quad (2.9)$$

$$x_{ij} \leq \sum_{k=1}^r y_{ik}, \quad i = 1, \dots, p, \quad j = 1, \dots, q, \quad (2.10)$$

$$x_{ij} \leq \sum_{k=1}^r z_{jk}, \quad i = 1, \dots, p, \quad j = 1, \dots, q, \quad (2.11)$$

$$\sum_{i=1}^p \sum_{k=1}^r y_{ik} \leq p, \quad (2.12)$$

$$\sum_{j=1}^q \sum_{k=1}^r z_{jk} = q, \quad (2.13)$$

$$\sum_{i=1}^p y_{ik} + \sum_{j=1}^q z_{jk} \leq 1, \quad k = 1, \dots, r, \quad (2.14)$$

$$x_{ij} \in \{0, 1\}, \quad y_{ik} \in \{0, 1\}, \quad z_{jk} \in \{0, 1\}, \quad w_i \in \{0, 1\}, \quad (2.15)$$

$$i = 1, \dots, p, \quad j = 1, \dots, q, \quad k = 1, \dots, r.$$

Um agravante são as não linearidades  $x_{ij} \cdot y_{ik} \cdot z_{j\bar{k}}$  em  $z_1$  e  $y_{ik} \cdot y_{i\bar{k}}$  em  $z_2$ . Estas, por sua vez, podem ser linearizadas da seguinte forma, como apresentado no capítulo anterior.

Para  $z_1$ , seja  $v_{ijk\bar{k}} := x_{ij} \cdot y_{ik} \cdot z_{j\bar{k}}$  que será 1 se todas as variáveis forem 1 e 0 em qualquer outro caso contrário. Então impondo as restrições lineares:

$$\begin{aligned} x_{ij} + y_{ik} + z_{j\bar{k}} - v_{ijk\bar{k}} &\leq 2 \\ -x_{ij} - y_{ik} - z_{j\bar{k}} + 3v_{ijk\bar{k}} &\leq 0 \end{aligned} \quad (2.16)$$

para todo  $i = 1, \dots, p; j = 1, \dots, q; k = 1, \dots, r; \bar{k} = 1, \dots, r$ , teremos uma linearização do modelo original para a função  $z_1$  respeitando tal condição.

Para  $z_2$ , seja  $u_{i\bar{i}k\bar{k}} := y_{ik} \cdot y_{i\bar{k}}$ , que será 1 se ambas forem 1 e 0 em qualquer outro caso. Impondo as restrições lineares:

$$\begin{aligned} y_{ik} + y_{i\bar{k}} - u_{i\bar{i}k\bar{k}} &\leq 1 \\ -y_{ik} - y_{i\bar{k}} + 2u_{i\bar{i}k\bar{k}} &\leq 0 \end{aligned} \quad (2.17)$$

para todo  $i = 1, \dots, p, \bar{i} = 1, \dots, p, j = 1, \dots, q, k = 1, \dots, r, \bar{k} = k+1, \dots, r$  temos a condição descrita no parágrafo anterior satisfeita. Inserindo as desigualdades (2.16) e (2.17) no modelo (2.1)-(2.15), teremos uma versão linear deste problema, apresentada a seguir.

$$\text{minimize } z_1 = \sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r \sum_{\bar{k}=1}^r d_{k\bar{k}} \cdot v_{ijk\bar{k}} \quad (2.18)$$

$$\text{minimize } z_2 = \sum_{i=1}^p \sum_{\bar{i}=1}^p \sum_{k=1}^{r-1} \sum_{\bar{k}=k+1}^r d_{k\bar{k}} \cdot u_{i\bar{i}k\bar{k}} \quad (2.19)$$

$$\text{sujeito a } (2.3) - (2.14) \quad (2.20)$$

$$\begin{aligned} y_{ik} + y_{i\bar{k}} - u_{i\bar{i}k\bar{k}} &\leq 1, \\ i &= 1, \dots, p, \bar{i} = 1, \dots, p, k = 1, \dots, r-1, \bar{k} = k+1, \dots, r \end{aligned} \quad (2.21)$$

$$\begin{aligned} -y_{ik} - y_{i\bar{k}} + 2u_{i\bar{i}k\bar{k}} &\leq 0, \\ i &= 1, \dots, p, \bar{i} = 1, \dots, p, k = 1, \dots, r-1, \bar{k} = k+1, \dots, r \end{aligned} \quad (2.22)$$

$$\begin{aligned} x_{ij} + y_{ik} + z_{j\bar{k}} - v_{ijk\bar{k}} &\leq 2, \\ i &= 1, \dots, p, j = 1, \dots, q, k = 1, \dots, r, \bar{k} = 1, \dots, r \end{aligned} \quad (2.23)$$

$$\begin{aligned} -x_{ij} - y_{ik} - z_{j\bar{k}} + 3v_{ijk\bar{k}} &\leq 0, \\ i &= 1, \dots, p, j = 1, \dots, q, k = 1, \dots, r, \bar{k} = 1, \dots, r \end{aligned} \quad (2.24)$$

$$\begin{aligned} x_{ij} \in \{0, 1\}, \quad y_{ik} \in \{0, 1\}, \quad z_{jk} \in \{0, 1\}, \quad w_i \in \{0, 1\}, \quad u_{i\bar{i}k\bar{k}} \geq 0 \\ i = 1, \dots, p, \quad \bar{i} = 1, \dots, p, \quad j = 1, \dots, q, \quad k = 1, \dots, r, \quad \bar{k} = k+1, \dots, r. \end{aligned} \quad (2.25)$$

Nas formulações (2.18)-(2.25) serão aplicados os solver de programação linear para obter a fronteira de Pareto para este problema. Na próxima seção, iremos apresentar um modelo similar, mas que aloca apenas os clientes considerando-se as facilidades já alocadas.

## 2.2 Modelo matemático de alocação apenas dos clientes

Nesta formulação, alocamos apenas os clientes nos nós dados. As facilidades já tem sua instalação fixada a priori.

Suponhamos que conheçamos o conjunto  $\mathcal{L} \subset \{1, \dots, r\}$  nas quais as  $|\mathcal{L}| \leq r$  facilidades estão alocadas e  $\mathcal{N} = \{1, \dots, r\} \setminus \mathcal{L}$  o conjunto dos nós livres (desocupados). Suponhamos também que sabemos quais das facilidades foram utilizadas e aonde estão alocadas dentre os nós do conjunto  $\mathcal{L}$ . Denotamos este conjunto por  $\mathcal{F}$ .

O modelo matemático mono-objetivo não-linear que aloca os clientes nos nós desocupados, e dimensiona o escoamento é apresentado a seguir, minimizando a soma das distâncias entre os clientes e as facilidades ( $z_1$ ). Lembrando que as variáveis  $x_{ij}$  e  $z_{jk}$  possuem a mesma definição do modelo apresentado na seção anterior.

$$\text{minimize} \quad z_1 := \sum_{i \in \mathcal{F}} \sum_{j=1}^q \sum_{k \in \mathcal{N}} \sum_{\bar{k} \in \mathcal{L}} d_{k\bar{k}} \cdot x_{ij} \cdot y_{i\bar{k}} \cdot z_{jk} \quad (2.26)$$

$$\text{sujeito a} \quad \sum_{i \in \mathcal{F}} x_{ij} = 1, \quad j = 1, \dots, q, \quad (2.27)$$

$$\sum_{j=1}^q y_{ik} \leq 1, \quad k \in \mathcal{N}, \quad (2.28)$$

$$\sum_{k \in \mathcal{N}} z_{jk} \leq 1, \quad j = 1, \dots, q, \quad (2.29)$$

$$\sum_{j=1}^q dem_j \cdot x_{ij} \leq C_i, \quad i \in \mathcal{F}, \quad (2.30)$$

$$x_{ij} \leq \sum_{k \in \mathcal{N}} z_{jk}, \quad i \in \mathcal{F}, \quad j = 1, \dots, q, \quad (2.31)$$

$$\sum_{j=1}^q \sum_{k \in \mathcal{N}} z_{jk} = q, \quad (2.32)$$

$$\sum_{i \in \mathcal{F}} y_{i\bar{k}} = 1, \quad \bar{k} \in \mathcal{L}, \quad (2.33)$$

$$\sum_{\bar{k} \in \mathcal{L}} y_{i\bar{k}} = 1, \quad i \in \mathcal{F}, \quad (2.34)$$

$$x_{ij} \in \{0, 1\}, \quad y_{i\bar{k}} \in \{0, 1\}, \quad z_{jk} \in \{0, 1\}, \quad (2.35)$$

$$i \in \mathcal{F}, \quad j = 1, \dots, q, \quad k \in \mathcal{N}, \quad \bar{k} \in \mathcal{L}.$$

Linearizando-se o modelo (2.26)-(2.35), usando as mesma técnicas da seção precedente, temos a versão linearizada apresentada a seguir:

$$\text{minimize} \quad z_1 := \sum_{i \in \mathcal{F}} \sum_{j=1}^q \sum_{k \in \mathcal{N}} \sum_{\bar{k} \in \mathcal{L}} d_{k\bar{k}} \cdot v_{ijk\bar{k}} \quad (2.36)$$

$$\text{sujeito a} \quad (2.27) - (2.35) \quad (2.37)$$

$$x_{ij} + z_{jk} + y_{i\bar{k}} - v_{ijk\bar{k}} \leq 2, \quad i \in \mathcal{F}, j = 1, \dots, q, k \in \mathcal{N}, \bar{k} \in \mathcal{L}, \quad (2.38)$$

$$-x_{ij} - z_{jk} - y_{i\bar{k}} + 3v_{ijk\bar{k}} \leq 0, \quad i \in \mathcal{F}, j = 1, \dots, q, k \in \mathcal{N}, \bar{k} \in \mathcal{L}, \quad (2.39)$$

$$x_{ij} \in \{0, 1\}, \quad y_{i\bar{k}} \in \{0, 1\}, \quad z_{jk} \in \{0, 1\}, \quad v_{ijk\bar{k}} \geq 0 \quad (2.40)$$

$$i \in \mathcal{F}, \quad j = 1, \dots, q, \quad k \in \mathcal{N}, \quad \bar{k} \in \mathcal{L}.$$

Nas formulações lineares (2.36)-(2.40) serão empregados os métodos de escalizações para resolvermos os subproblemas para assim, gerar a fronteira de Pareto deste problema.

A seguir, para facilitar o entendimento dos modelos matemáticos que apresentamos, fornecemos uma interpretação geométrica com um exemplo de  $r = 10$  nós.

**Exemplo 2** (Aplicação do Modelo (2.1)-(2.15) de alocação de facilidades e clientes). Vamos ilustrar uma região com  $r = 10$  nós, cada nós representando um ponto que pode acolher uma das três possibilidades mencionadas no parágrafo anterior. A matriz simétrica que dá a distância entre cada nó  $(i, j)$  é exibida a seguir.

$$\mathbf{D} = \begin{bmatrix} 0 & 20 & 50 & 50 & 50 & 60 & 80 & 90 & 130 & 80 \\ 20 & 0 & 65 & 40 & 30 & 40 & 60 & 125 & 85 & 100 \\ 50 & 65 & 0 & 95 & 95 & 133 & 110 & 45 & 85 & 100 \\ 50 & 40 & 95 & 0 & 100 & 110 & 130 & 130 & 45 & 30 \\ 50 & 30 & 95 & 100 & 0 & 10 & 30 & 95 & 55 & 70 \\ 60 & 40 & 133 & 110 & 10 & 0 & 23 & 88 & 48 & 63 \\ 80 & 60 & 110 & 130 & 30 & 23 & 0 & 65 & 25 & 40 \\ 90 & 125 & 45 & 130 & 95 & 88 & 65 & 0 & 40 & 55 \\ 130 & 85 & 85 & 45 & 55 & 48 & 25 & 40 & 0 & 15 \\ 80 & 100 & 100 & 30 & 70 & 63 & 40 & 55 & 15 & 0 \end{bmatrix}$$

Suponha que o mapa geográfico dos nós é apresentado pela Figura (2) onde as flechas denotam que existem as rodovias interligando estes nós. Por exemplo, não há

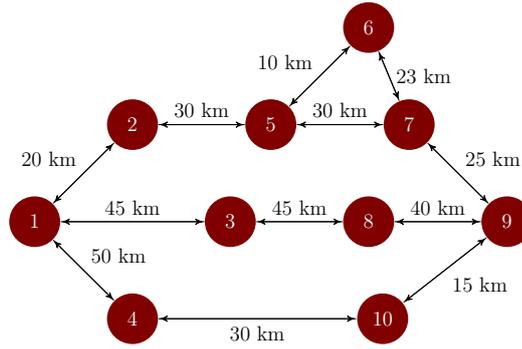


Figura 2 – Mapa rodoviário com as distâncias de um exemplar do Problema (2.1)-(2.15) com  $r = 10$  nós

rodovia ou caminho ligando os nós 2 e 3. Porém, para sair de 2 e ir até 3, precisamos passar por 1. Assim, podemos dizer que a distância de 2 a 3 é  $20 + 45 = 65$  km.

Assuma ainda que no máximo  $p = 4$  facilidades devam ser instaladas, exatamente  $q = 5$  clientes devem ser alocados. As capacidades de cada facilidade são dadas pelo vetor

$$C = [1000, 1000, 1000, 1070]^T,$$

e as demandas de cada cliente dado pelo vetor

$$dem = [500, 500, 300, 600, 800]^T.$$

A solução ótima do problema minimizando-se  $z_1$  é dada pelo seguinte esquema apresentado na Figura (3).

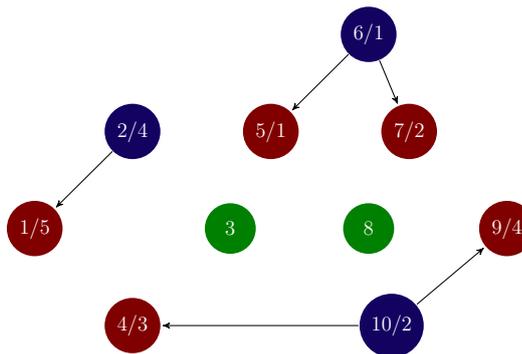


Figura 3 – Solução ótima de Pareto do Modelo (2.1)-(2.15) ao minimizar  $z_1$

Nesses diagramas (e em todo restante deste texto), as bolas azuis significam as facilidades, as vermelhas os clientes e as verdes significam nós desocupados. As os números à esquerda, dentro dos nós, indicam a localização do nós. Os números após a barra invertida indicam qual facilidade/cliente foram alocados. Por exemplo, na solução da Figura (3) o nó 6 alocou a facilidade 1, enquanto o nó 4 alocou o cliente 3. A existência de uma aresta com sentido, indica que há escoamento da respectiva facilidade para seu cliente.

Sendo assim, a facilidade 2, localizada no nó 10, atende aos clientes 3 e 4 localizados nos nós 4 e 9, respectivamente.

Esta solução tem um  $z_1$  mínimo de 98 km. Note como a facilidade fica próxima de seu ponto de atendimento. Por outro lado, as facilidades ficaram uma longe das outras, resultando em um valor de  $z_2$  é de 203 km. A quantidade que cada facilidade fornece é de  $[1000, 900, 0, 800]$ . Repare que a facilidade 3 não foi aberta.

Por outro lado, determinando a alocação ótima minimizando-se o objetivo  $z_2$ , teremos o seguinte roteiro e alocação das facilidades e consumidores, apresentado na Figura (4):

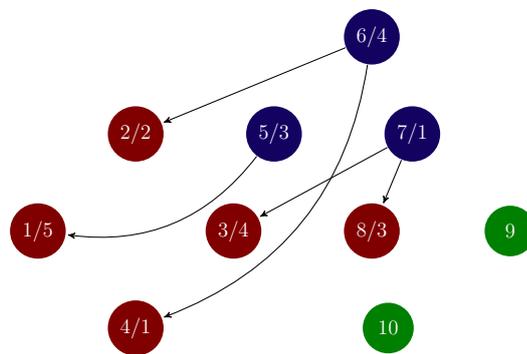


Figura 4 – Solução ótima do Modelo (2.1)-(2.15) ao minimizar  $z_2$

Nessa solução, temos um valor mínimo de  $z_2$  igual à 63 km (soma de todas as distâncias entre os nós 5, 6 e 7) ao passo que as distâncias entre todos os clientes é igual à 375 km. A quantidade que cada facilidade fornece é de  $[900, 0, 800, 1000]$ .

Nesta solução, note como o modelo alocou 3 das 4 facilidades e encontramos um valor de  $z_2$  mínimo de 63 km, ao somarmos  $30 + 10 + 23$ . Porém, o valor das distâncias entre as facilidades e os clientes saltou de 98 km para 245 km.

Poderemos pensar em gerar uma solução intermediária, digamos, uma que é a combinação de  $z_1$  e  $z_2$ . Resolvendo o mesmo problema considerando-se como objetivo a minimização de  $z_3 := z_1 + z_2$  (as duas funções objetivo com o mesmo peso), geramos a seguinte solução, representada pela Figura (5).

Esta solução eficiente é intermediária e busca minimizar  $z_1$  e  $z_2$  ao mesmo tempo, estabelecendo um compromisso entre elas. Sua representação é dada na Figura (5). outros pesos poderiam ser dados e assim, muitas outras soluções-compromisso poderia ser obtidas. Os valores para as funções  $z_1$  e  $z_2$  são iguais à 135 km e 123 km, respectivamente. A quantidade que cada capacidade enviou é igual à  $[900, 1000, 0, 800]$ .

**Exemplo 3** (Aplicação do Modelo (2.26)-(2.35) de alocação apenas das facilidades). Uma simulação usando a formulação (2.26)-(2.35) com os seguintes dados foi realizada. Suponhamos que o conjunto  $\mathcal{L} = \{2, 6, 10\}$  é dos nós ocupados pelas facilidades  $\mathcal{F} =$

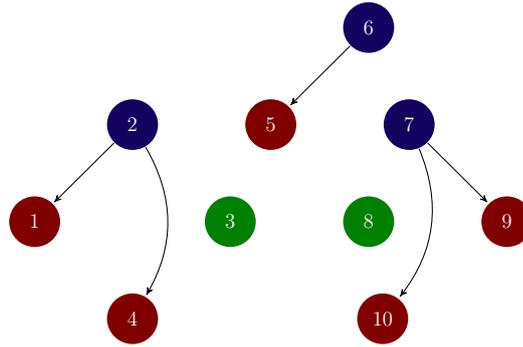


Figura 5 – Solução ótima do Modelo (2.1)-(2.15) ao minimizar  $z_3 = z_1 + z_2$

$\{1, 2, 3\}$ , isto é, facilidade 1 alocada no nó 2, facilidade 2 no nó 6 e facilidade 3 no nó 10. A questão é saber onde os  $q = 5$  clientes serão instalados nos 7 nós livres e qual o fluxo de cada facilidade para cada cliente.

Minimizando-se a função  $z_1$ , temos uma solução ótima, com  $z_1 = 98$  km. Nesta configuração, visou-se ter um distanciamento mínimo entre os clientes e as facilidades. A representação desta solução é dada na Figura (6).

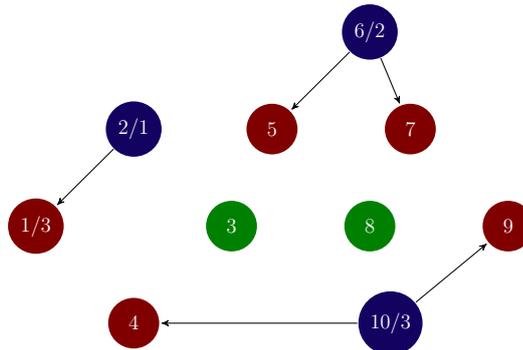


Figura 6 – Solução ótima do Modelo (2.26)-(2.35) com  $\mathcal{L} = \{2, 6, 10\}$

Suponhamos agora que o conjunto  $\mathcal{L} = \{4, 8, 5\}$  é dos nós ocupados pelas facilidades  $\mathcal{F} = \{1, 2, 3\}$ , isto é, facilidade 1 alocada no nó 4, facilidade 2 no nó 8 e facilidade 3 no nó 5. A solução ótima para este problema é dada pela Figura (7) a seguir, com um  $z_1$  mínimo de 150 km.

Este modelo ilustra a possibilidade do gestor, antes de querer saber onde os clientes serão alocados, bem como a maneira que será o fluxo da mercadoria, testar diferentes situações onde as facilidades deverão ser construídas para um melhor escoamento. Vemos que na primeira situação, a instalação das facilidades nos nós 2, 6 e 10 foi melhor.

Os exemplos anteriores ilustram um forte potencial de aplicação destas formulações. No próximo capítulo, ilustramos simulações computacionais para modelos reais gerados aleatoriamente.

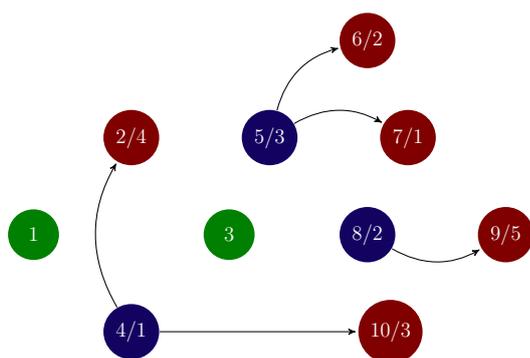


Figura 7 – Solução ótima do Modelo (2.26)-(2.35) com  $\mathcal{L} = \{4, 8, 5\}$

## Capítulo 3

# Resultados computacionais

Neste Capítulo apresentaremos os resultados das simulações do modelo apresentado no Capítulo anterior. Foram simuladas 6 instâncias para cada um dos modelos. Em cada instância são variados os valores dos parâmetros  $r$  (quantidade de nós disponíveis),  $q$  (quantidade de clientes a serem alocados) e  $p$  (quantidade de facilidades que serão alocadas) assim como os valores limites de tempo e gap. Para cada instância simulada, o peso  $\lambda$  entre facilidades e  $1 - \lambda$  entre clientes foi 0.0, 0.5 e 1.0. Cada uma das instâncias simuladas será discutida mais adiante. Os problemas simulados marcados com (\*) foram realizados em um computador Intel(R) Core(TM) i3 M350 de 2.27GHz, com 2 GB de memória em um sistema operacional *Windows 7 Professional de 32 Bits*.

O software utilizado para resolução dos problemas foi o GUSEK © (*GLPK Under Scite Extended Kit*) Versão 0.2. O *GUSEK* é um *software* livre, com interface de desenvolvimento para modelos de Programação Linear e Programação Linear Inteira Mista; os modelos são desenvolvidos de acordo com a linguagem de modelagem própria do GLPK, a *GMPL (GNU MathProg Language)* e para obter as soluções o *software* utiliza o método de *Branch-and-Bound*, já comentado no Capítulo 1. Foi realizada toda a implementação de linguagem para que fosse possível o uso dos modelos apresentados no Capítulo anterior nas instâncias simuladas. O *software* pode ser facilmente obtido em <http://gusek.sourceforge.net>.

A interface do programa utilizado é apresentada na Figura (8):

Foi necessário fixar alguns parâmetros do GLPK para que fosse possível obter solução ao menos aproximada e em tempo aceitável (menor que 1 hora). Além dos parâmetros padrão (*cover, clique, gomory, mir*), também estabelecemos um tempo máximo de resolução de 3600 segundos para cada instância ou um número máximo de nós na árvore B & B de no máximo 70.000 nós. Quando um dos dois critérios foi atendido, o solver parou de resolver o problema.

O mapa geográfico das 15 cidades que usamos nas simulações encontra-se

```

1  #!-cover --clicque --gurobi --mir --trilim 600 --mipgap 0.01 --pcost --cuts
2  param lambda, >=0, <=1;
3
4  param p, integer, > 0;
5  /* número de facilidades */
6
7  param q, integer, > 0;
8  /* Número de clientes */
9
10 param r, integer, > 0;
11 /* Número de locais */
12
13 param f(i in 1..p), integer, >0;
14 /* Custo de instalação da facilidade i */
15
16 param Cap(i in 1..p), integer, >0;
17
18 param dem(j in 1..q), integer, >0;
19
20 param d(i in 1..r, j in 1..r), >= 0;
21 /* distância entre os locais i e j */
22
23 var x(i in 1..p, j in 1..q), >= 0, binary;
24 /* x[i,j] = 1 é um se a facilidade i atende o local j e 0 caso contrário */
25
26 var y(i in 1..p, k in 1..r), >= 0, binary;
27 /* y[i,k] = 1 é um se a facilidade i é instalada no local k 0 caso contrário */
28
29 var z(j in 1..q, k in 1..r), >= 0, binary;
30 /* z[j,k] = 1 é um se o cliente j é instalado no local k 0 caso contrário */
31
32 var w(i in 1..p), >= 0, binary;
33 /* w[i] = 1 se a facilidade i é instalada 0 caso contrário */
34
35 var u(i in 1..p, i_barra in 1..p, k in 1..r-1, k_barra in k+1..r), >=0;
36 #Variáveis da linearização
    
```

Figura 8 – Interface Gurobi

apresentado na Figura (9). Usando-se ele, conseguimos obter todas as distâncias entre as cidades, isto é, a matriz  $D = (d_{ij})$  que contém todas as distâncias  $d_{ij}$  entre os nós  $i$  e  $j$ .

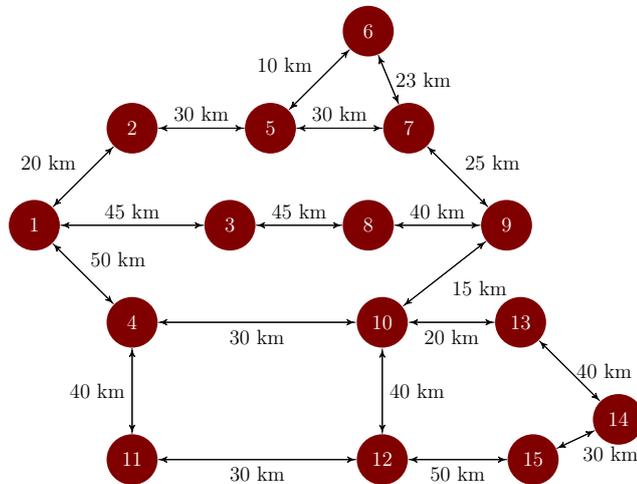


Figura 9 – Mapa rodoviário com as distâncias das 15 cidades para os testes computacionais

O vetor com as capacidades foi o seguinte:

$$C = (3000; 1000; 1500; 1200; 800)^T,$$

e para as demandas dos clientes foi atribuído o seguinte vetor:

$$dem = (500; 500; 300; 600; 800; 500; 500; 200; 300; 400)^T.$$

Se a instância possui menos de  $r = 15$  nós, menos de  $p = 5$  facilidades ou menos de  $q = 10$  clientes, tomamos as primeiras componentes destes vetores, conforme a dimensão da instância utilizada.

Em ambas as fases, as características destas instâncias, estão apresentadas na Tabela (2). Nossa intenção é comparar os modelos matemáticos e as soluções que eles produzem. Isso será feito nas próximas sub-seções.

Tabela 2 – Características das classes testadas

Classe	$r$	$p$	$q$
1	10	2	4
2	10	2	6
3	10	2	8
4	12	4	4
5	12	4	6
6	12	4	8
7	15	5	6
8	15	5	8
9	15	5	10

### 3.1 Fase 1 - Resultados para a Formulação (2.26)-(2.35) que instala apenas os clientes

Nesta fase, foram coletadas as seguintes informações, em cada instância:

- $z_1$  : Soma das distâncias (rota total) entre as facilidades e os clientes (em km);
- $z_c$ : Valor da soma das distâncias apenas entre os clientes (em km). Esta distância é calculada por:

$$z_c := \sum_{i=1}^q \sum_{\bar{i}=1}^q \sum_{k=1}^{r-1} \sum_{\bar{k}=k+1}^r d_{k\bar{k}} \cdot z_{ik} \cdot z_{i\bar{k}}.$$

Este objetivo não foi otimizado.

- Tempo CPU: Tempo utilizado pela CPU para resolver cada problema escalar (em segundos);
- Gap: Erro percentual entre a solução relaxada e a melhor solução inteira determinada (em %);
- Número de variáveis;
- Número de restrições.

Na Fase 1 implementamos o modelo (2.26)-(2.35) que supõe as facilidades fixas e aloca apenas os clientes. Consideramos as nove instâncias distintas neste problema.

Tabela 3 – Resultados computacionais para as 9 instâncias do Problema (2.26)-(2.35)

Instância	$z_1$	$z_c$	# restr.	# var.	Gap	CPU
1	135	85	184	219	0,00%	0,1875
2	295	85	268	321	0,00%	0,3438
3	438	85	352	423	0,00%	0,9586
4	125	460	592	609	0,00%	1,5687
5	228	460	872	903	0,01%	358,21
6	348	460	1152	1197	5,84%	1665,45
7	175	880	1640	1664	9,71%	3600,00
8	273	880	2170	2210	16,98%	3600,00
9	398	880	2700	2756	31,76%	3600,00

Os resultados computacionais para os valores das funções  $z_1$ ,  $z_c$ , número de variáveis e restrições, o gap da solução inteira e o tempo de CPU utilizado em cada classe estão apresentados na Tabela (4).

É importante tecermos alguns comentários em relação aos resultados computacionais presentes na tabela 4. É evidente um aumento considerável no número de variáveis e restrições deste modelo matemático com sensível modificação no número de nós, clientes ou facilidades. Por exemplo, da instância 1 para a instância 2, apenas 2 clientes a mais são considerados. Isso acarretou um aumento em 45% em restrições e variáveis. Outra análise: comparando as Instâncias 6 e 8, onde há 3 nós e uma facilidade a mais, o número de variáveis e restrições quase dobrou.

Em relação ao valor ótimo da função objetivo, nós notamos que, quando há maior possibilidade de remoção de clientes entre os nós (isto é, quando a diferença entre  $r - p$  e  $q$  aumenta), mais reduzidas ficam as rotas entre os atendimentos. Isso pode ser notado nas Instâncias 1, 4 e 7, em detrimento das rotas onde esta diferença é nula (Instâncias 3, 6 e 9). Como as facilidades não mudam de lugar, o valor de  $z_c$  é constante entre as instâncias de mesmo valor de  $q$ . Na próxima fase iremos fazer uma comparação destes valores com o que o modelo fornece.

A resolução deste problema foi considerada complexa. As instâncias de 1 a 4 foram facilmente resolvidas. Nas instâncias 5 e 6, o GUSEK encerrou a busca pela solução ótima com menos de 1 hora porque o critério de 70.000 foi atingido. A partir da instância 7, o tempo de 1 hora não foi suficiente para encontrar a solução ótima, e o gap foi tanto maior quanto o número de variáveis e restrições.

Nas Figuras (10) a (18), exibimos os esquemas das soluções de cada instância, lembrando que os nós azuis representam as facilidades, os nós vermelhos os clientes, os nós verdes estão vagos e a flecha com sentido indicando que há escoamento daquela facilidade para o referido cliente.

Por exemplo, na Instância 1, as facilidades 1 e 2 estão alocadas nos nós,

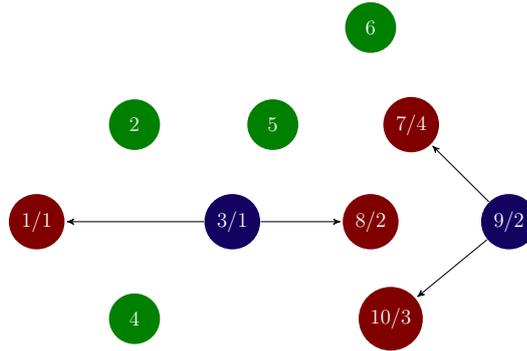


Figura 10 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 1

respectivamente 3 e 9, *a priori*, fornecendo os clientes localizados nos nós 1 e 8 e aos clientes 7 e 10, respectivamente. Nesta mesma instância, o valor mínimo determinado pelo modelo foi a rota indicada no esquema, totalizando-se 135 km. Isso ocasionou com que as facilidades ficassem a uma distância de 85 km entre si.

Já na solução proveniente da Instância 2, como dois novos clientes foram adicionados, o modelo os alocou no nó 2 (atendido pela facilidade 2) e no nó 4 (atendido pela facilidade 1), como ilustra a Figura (11).

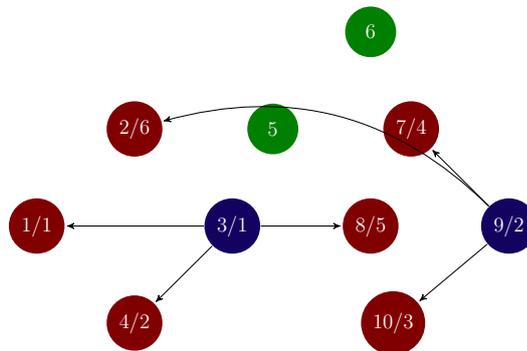


Figura 11 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 2

Para a Instância 3, todos os nós são ocupados, restando ao modelo decidir onde os 8 clientes serão alocados e qual o sentido das rodas. A solução ótima forneceu uma rota de 438 km, re-allocando os clientes de modo que fiquem “mais próximos possíveis” de suas respectivas facilidades, como mostra a Figura (12).

As Figuras (13), (14) e (15) mostram as soluções ótimas de rota mínima para as instâncias com  $q = 4$ ,  $q = 6$  e  $q = 8$  clientes, respectivamente. Quando  $q = 4$ , não houve necessidade de se abrirem as 4 facilidades até então disponíveis, cabendo à facilidade 2 de capacidade de 1000, atender os dois clientes (demanda 500 cada).

Passando à outra instância, a facilidade 2 passa a atender os clientes 3 (de demanda 300) e 4 (de demanda 600) localizados nos nós 6 (que antes estava vago) e 7,

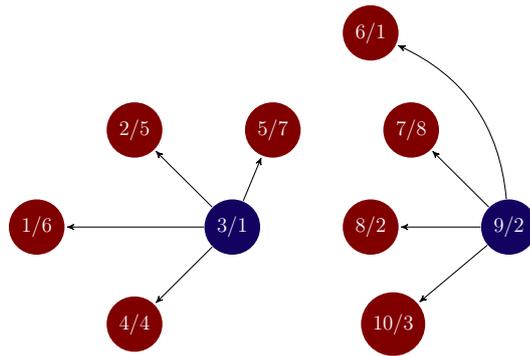


Figura 12 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 3

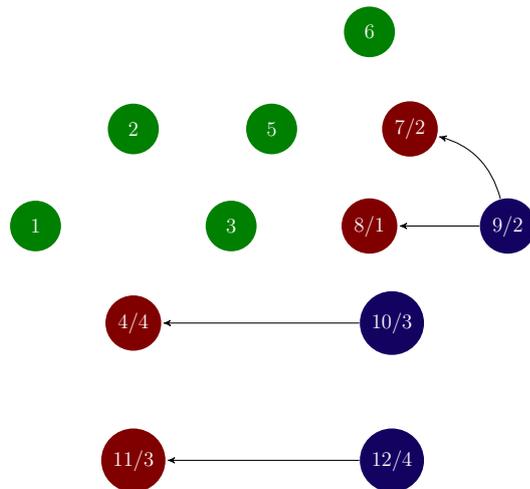


Figura 13 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 4

respectivamente. Uma facilidade no nós 3 foi aberta, que atende os clientes nos nós 1 e 8. Os demais atendimentos permaneceram iguais como na solução anterior.

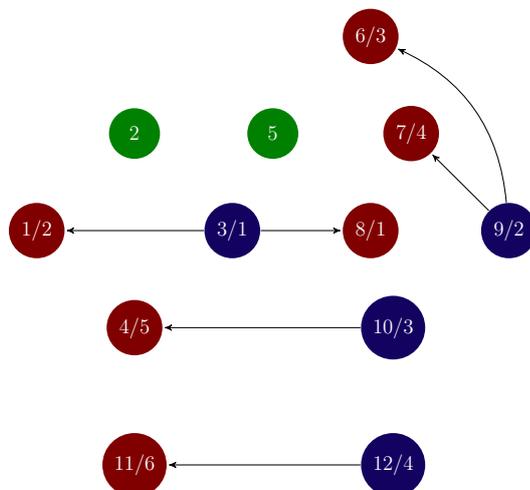


Figura 14 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 5

A instância 6 tem dois clientes a mais em relação à anterior. O modelo aloca

no nó 2 o cliente 6 e no nó 5 o cliente 8, atendidos, respectivamente, pelas facilidades 1 e 2. Olhando para a Figura (15), uma pergunta que poderíamos fazer era a seguinte: “*porque a facilidade 2, em vez de atender aos clientes 2, 3 e 7, não atende os clientes 2, 3 e 8, uma vez que o cliente 7 está mais próximo desta facilidade ao invés do cliente 8?*” Isto pode ser respondido olhando para as capacidades das facilidades e a demanda de cada cliente. Tal como está exibido, a facilidade 2 terá sua capacidade (de 1000) completamente usada ao atender o cliente 2 (de demanda 500), o cliente 3 (de demanda 300) e o cliente 8 (de demanda 200), tornando a solução factível. Se isso não fosse respeitado, e apenas a distância fosse levada em conta, o cliente 7 (de demanda 500) ao ser suprido, infringiria a capacidade máxima de atendimento desta facilidade.

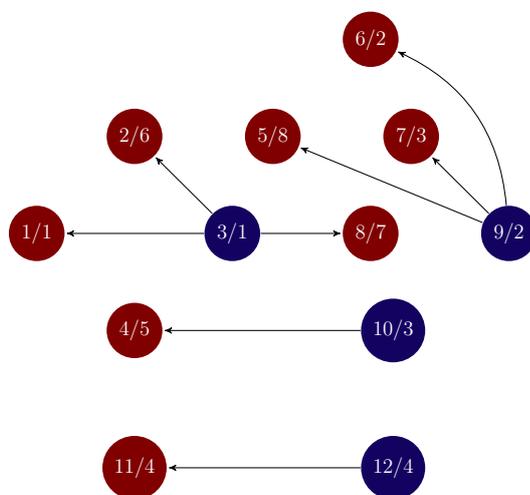


Figura 15 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 6

Para as instâncias 7, 8 e 9, há 6, 7 e 8 clientes a serem atendidos pelas facilidades 1, 2, 3 e 4 nos nós 3, 9, 10, 12 e 14 que podem ou não fornecer. Por exemplo, na instância 7, apenas as facilidades 2, 3 e 4 foram abertas, pois o número de 6 clientes não necessitou que as 4 fossem abertas. Como há nós vagos, então o modelo matemático alocou os clientes de modo que as rotas tivessem mínima distância.

Na Figura (17), como dois novos clientes foram criados, houve necessidade de abrir uma nova facilidade, no nó 3, que atende um cliente no nó 1. Além disso, um novo cliente no nó 6 foi alocado e atendido pela facilidade 2, permanecendo como na solução anterior os demais atendimentos.

Finalmente, na Instância 9, temos todos os 15 nós sendo ocupados, restando ao modelo localizar os clientes e o fluxo. Note que os nós de índice maior que dez ficam “isolados”, sendo atendidos basicamente pelas facilidades que estão nesta região. Em outras palavras, observou-se que, por exemplo, uma facilidade localizada no nó 15 não atende um cliente localizado nos nós de 1–9.

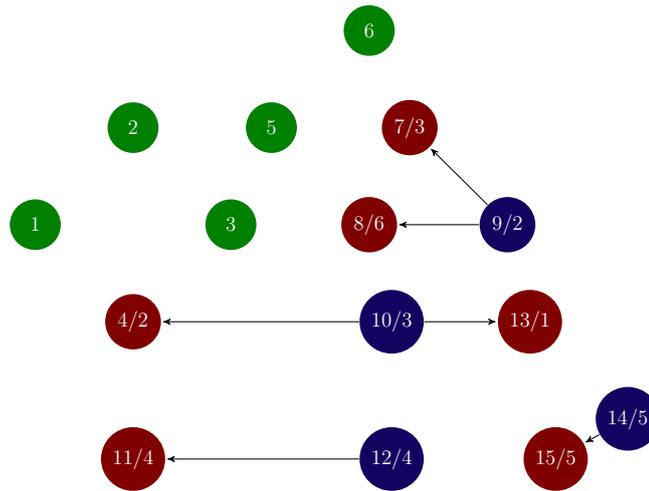


Figura 16 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 7

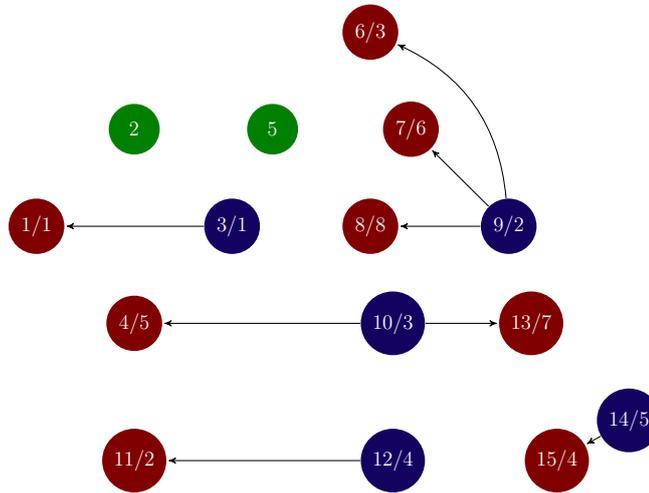


Figura 17 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 8

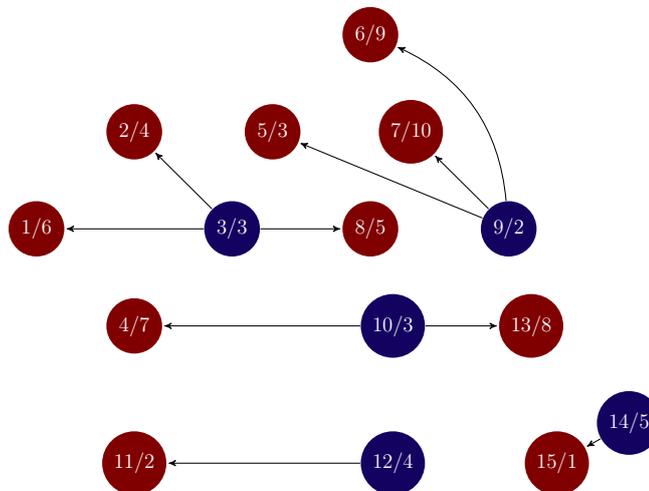


Figura 18 – Esquema da solução ótima para o Problema (2.26)-(2.35) - Instância 9

## 3.2 Fase 2 - Resultados para a Formulação (2.1)-(2.15) que instala os clientes e as facilidades

Para testar e validar a formulação (2.1)-(2.15), em cada instância foram coletadas as seguintes informações:

- $z_1$  : Valor da distância (rota) entre as facilidades e os clientes (em km);
- $z_2$ : Valor da soma das distâncias entre as facilidades (em km);
- $\lambda$  e  $1 - \lambda$ : peso dado às funções  $z_1$  e  $z_2$  do modelo. Foram adotados os seguintes valores para  $\lambda$ : 1.0, 0.5 e 0.0;
- Tempo CPU: Tempo utilizado pela CPU para resolver cada problema escalar (em segundos);
- Gap: Erro percentual entre a solução relaxada e a melhor solução inteira determinada (em %);
- Número de variáveis;
- Número de restrições.

Os resultados computacionais para os valores das funções  $z_1$ ,  $z_2$ , número de variáveis e restrições, o gap da solução inteira e o tempo de cpu utilizado em cada classe estão apresentados nas Tabelas (4), (5) e (6), adotando-se, respectivamente, os valores  $\lambda = 0.0$ ,  $\lambda = 0.5$  e  $\lambda = 1.0$ .

Tabela 4 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e  $\lambda = 1.0$

Instância	$z_1$	$z_2$	# restr.	# var.	Gap	CPU
1	78	63	2022	1050	0,00%	14,95
2	165	55	2834	1474	12,36%	365,20
3	275	85	3646	1898	15,35%	580,00
4	68	368	6810	3476	22,36%	1536,55
5	128	543	9134	4660	27,32%	3600,00
6	240	530	11458	5844	29,54%	3600,00
7	118	713	18884	9575	33,54%	3600,00
8	178	986	23408	11865	55,36%	3600,00
9	340	1235	27932	14155	74,11%	3600,00

Os resultados mostram que o modelo tem a natureza multi-objetiva, de fato. Para a mesma instância e notando a variação de  $\lambda$ , vemos que a medida que um objetivo cresce o outro decresce. Vamos tomar como exemplo a Instância 9. Quando  $\lambda = 1.0$ , apenas

Tabela 5 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e  $\lambda = 0.5$ 

Instância	$z_1$	$z_2$	# restr.	# var.	Gap	CPU
1	100	30	2022	1050	0,00%	14,07
2	165	55	2834	1474	15,37%	380,70
3	300	55	3646	1898	12,35%	408,32
4	125	191	6810	3476	17,65%	789,23
5	205	214	9134	4660	11,65%	1235,11
6	315	235	11458	5844	17,98%	1956,32
7	255	403	18884	9575	24,87%	3600,00
8	290	462	23408	11865	42,98%	3600,00
9	443	460	27932	14155	63,74%	3600,00

Tabela 6 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e  $\lambda = 0.0$ 

Instância	$z_1$	$z_2$	# restr.	# var.	Gap	CPU
1	151	10	2022	1050	0,00%	16,40
2	302	10	2834	1474	0,00%	442,23
3	499	10	3646	1898	0,00%	616,32
4	115	191	6810	3476	4,52%	1789,32
5	210	191	9134	4660	6,36%	1987,22
6	350	191	11458	5844	9,21%	1878,35
7	200	379	18884	9575	18,32%	3600,00
8	350	379	23408	11865	27,88%	3600,00
9	585	379	27932	14155	39,41%	3600,00

o objetivo  $z_1$  é minimizado, resultando em uma solução ótima que fornece  $z_1 = 340$  e um  $z_2 = 1235$  (melhor solução para  $z_1$ ). Colocando um peso de  $\lambda = 0.5$  em cada objetivo, temos uma solução ótima que resulta em um  $z_1 = 443$  e um  $z_2 = 460$ . Finalmente, pondo  $\lambda = 0.0$ , o modelo fornece uma solução ótima com  $z_1 = 585$  e um  $z_2 = 379$  (melhor solução para  $z_2$ ). A abordagem multi-objetiva permitiu-nos obtermos 3 soluções bastante distintas de mesma importância (estas, soluções eficientes no senso de Pareto), em que a amplitude de  $z_1 = 245$  km ao passo que a amplitude de  $z_2 = 775$  km. Estas três soluções estão representadas pelas Figuras (19), (20) e (21), respectivamente.

Outra análise que faz-se necessária é o crescimento do número de variáveis e restrições deste problema com os parâmetros  $p$ ,  $q$  e  $r$ . Comparando, por exemplo, as Instâncias 1 e 2 onde apenas dois clientes a mais são considerados, o aumento no número de variáveis e restrições aumentou em 40%. A formulação matemática, nesse sentido, é muito sensível e isto está intimamente associado às restrições que linearizam as variáveis  $u$  e  $v$  do modelo. Ainda, comparando as formulações instalando e não instalando as facilidades, vemos uma profunda diferença na dimensão. Considere a Instância 3 nos dois modelos propostos. A diferença percentual em número de variáveis é de 10.3 vezes em número de restrições e de 4.5 vezes em número de variáveis. Sem dúvida, isto é um ponto importante

no tempo de resolução da Formulação (2.1)-(2.15).

A única instância que foi resolvida na otimalidade para os 3 valores de  $\lambda$  foi a Instância 1. As demais, o tempo de 1 hora dado ao GUSEK ou os 70.000 nós fizeram o solver obter uma solução sub-ótima. Isso permite-nos concluir que este modelo é mais complexo de ser resolvido em relação ao Modelo (2.26)-(2.35). Outra observação interessante é o fato dos pesos interferirem no tempo de resolução deste modelo bi-objetivo. Note que, até a Instância 3 e usando  $\lambda = 0.0$ , os problemas foram resolvidos otimamente (gap nulo). Por outro lado, a medida que o peso do objetivo  $z_1$  é aumentado, o GUSEK começou a ter dificuldades no sentido de não obter a solução ótima destas instâncias em um tempo inferior à 1 hora ou 70.000 nós da árvore. Isso pode ser explicado porque o objetivo  $z_1$  é tri-linear e não-convexo, diferentemente do objetivo  $z_2$  que é quadrático e convexo. Delineando-se a mesma observação para outro exemplo, note o gap da instância 9 para  $\lambda = 1.0$  (74%),  $\lambda = 0.5$  (63%) e  $\lambda = 1.0$  (39%), reiterando as nossas observações neste parágrafo.

Nesta fase, apenas apresentamos as 3 soluções eficientes para cada uma das Instâncias 1 e 9, a fim de evitar um texto excessivamente longo. Para a Instância 1, encontramos a representação das soluções nas Figuras (22), (23) e (24) para  $\lambda$  igual à 1.0, 0.5 e 0.0, respectivamente.

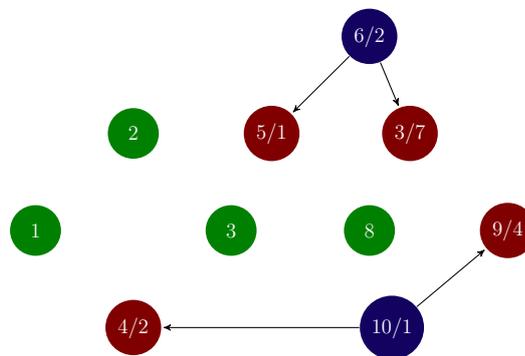


Figura 19 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 1.0$  - Instância 1

Comparando a solução apresentada na Figura (10), concernente à Formulação (2.26)-(2.35) e a da Figura (19), referente ao Modelo (2.1)-(2.15), notamos uma diferença significativa entre elas. A comparação é justa porque em ambas soluções,  $z_1$  é minimizado. Nesta solução, o valor de  $z_1$  mínimo é de 78 km, ao passo que naquela este valor é de 135 km, quase a metade. Isto é explicado porque o modelo matemático aloca as facilidades nos nós onde é mais favorável à redução das rotas. Note que nesta solução, sempre os clientes estão nos nós que são vizinhos às facilidades. Isso mostra que o modelo que determina concomitantemente a localização dos clientes e as facilidades, embora seja mais caro de ser otimizado, escolhe mais adequadamente os nós.

Na Figura (20), apresentamos uma solução cujo  $z_1 = 100$  e  $z_2 = 30$ , que busca

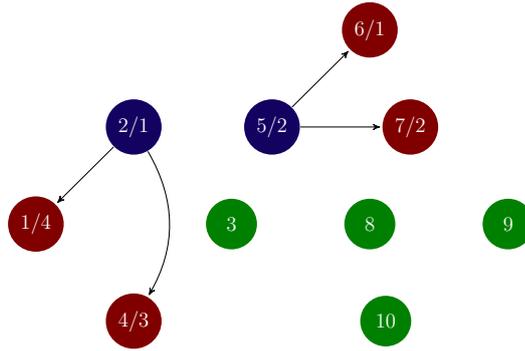


Figura 20 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 0.5$  - Instância 1

minimizar as rotas e o distanciamento entre as facilidades. Note que as facilidades e os clientes, à medida que  $\lambda$  muda de valor, tomam outras posições.

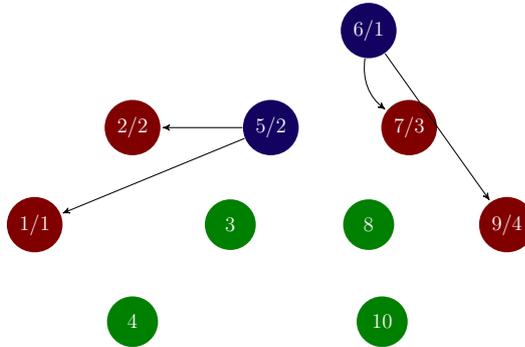


Figura 21 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 0.0$  - Instância 1

Ainda associada a esta instância, apresentamos na Figura (21) a solução com  $\lambda = 0.0$ , isto é, o modo de escoamento de mercadorias que visa apenas aproximas as facilidades. A melhor escolha foi alocá-las nos nós 5 e 6, que são os mais próximos.

Nas Figuras (22), (23) e (24), apresentamos os esquemas das soluções para a Instância 9 para os três valores de  $\lambda$ . Nenhuma destas soluções apresentadas são ótimas globais.

Notamos, conforme observado anteriormente, que a variação de  $\lambda$  promove soluções bastante distintas umas das outras. Na Figura (22), por exemplo, a soma das rotas foi de  $z_1 = 340$  km, ao passo que na solução apresentada na Figura (23) esse valor é de 443 km. Esse aumento é compensado com a maior aproximação entre as facilidades, que cai de 1235 km para 460 km entre estas soluções. Observe também que, como a capacidade da Facilidade 1 é de 3000, ela atendeu a 6 clientes e foi posicionada em ambas soluções no nó 9.

Agora, comparando as Figuras (18) e (22), respectivamente dos modelos que alocam as facilidades “manualmente” e “automaticamente”, vemos muitas diferenças entre

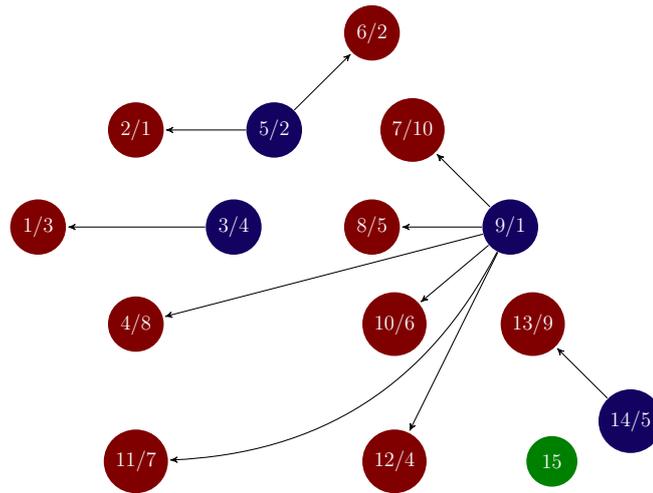


Figura 22 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 1.0$  - Instância 9

estas soluções. Em termos quantitativos, a primeira solução resulta em um  $z_1 = 398km$  ao passo que a segunda é de 340 km. Porém, o distanciamento entre as facilidades, na primeira solução é de 880 km e na segunda, 1235 km, mesmo esta solução usando uma facilidade a menos.

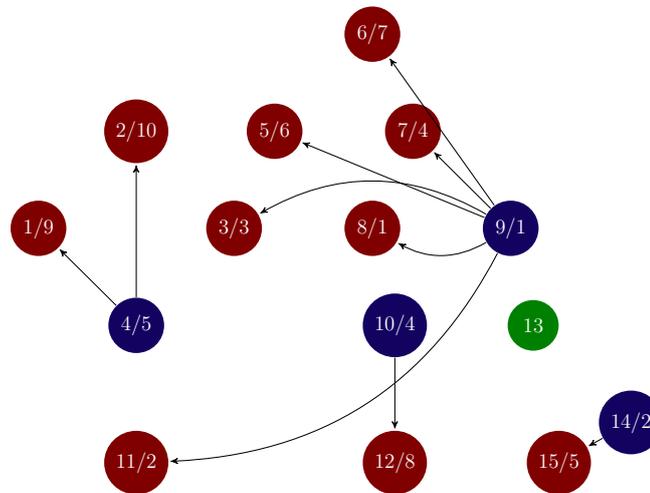


Figura 23 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 0.5$  - Instância 9

Finalmente, na Figura (24) é apresentada a solução visando aproximar as facilidades, que foi obtida com uma hora de resolução com o GUSEK, resultando em um  $z_1 = 585$  km e um  $z_2 = 379$  km. A Facilidade 1 foi mudada para o nó 6 e agora, atende apenas o cliente 2. A maior sobrecarga foi para a Facilidade 3, que atende os Clientes 6, 7 e 9.

Na próxima seção, a fim de obtermos um maior número de soluções eficientes, ilustramos uma simulação para a aplicação do método  $\varepsilon$ -Restrito no Problema (2.1)-(2.15)

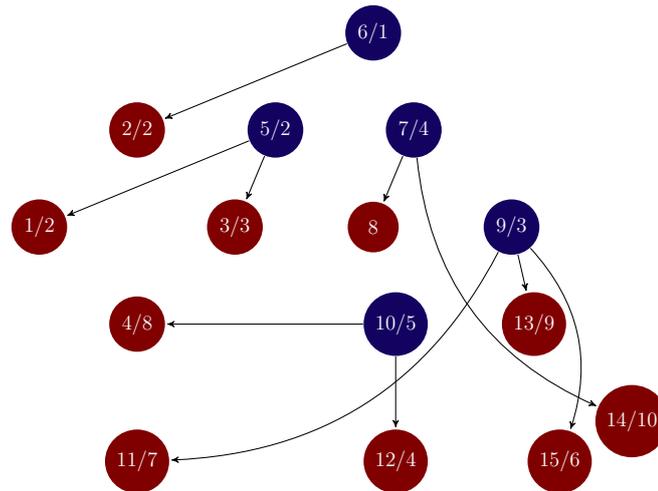


Figura 24 – Esquema da solução ótima para o Problema (2.1)-(2.15) com  $\lambda = 0.0$  - Instância 9

bi-objetivo.

### 3.3 Simulações com instâncias aleatórias

Nesta seção, apresentamos alguns resultados computacionais das nove instâncias utilizadas nas Fases 1 e 2, usando-se a formulação (2.1)-(2.15), isto é, aquela que aloca os clientes e as facilidades. A diferença é que geramos dados aleatórios para as capacidades de cada facilidade, demanda de cada cliente e a matriz das distâncias entre os nós, da seguinte forma:

- O vetor de capacidades  $C$  foi gerado num intervalo de 300 a 700;
- O vetor das demandas  $dem$  foi gerado num intervalo de 1000 a 2000;
- A matriz das distâncias foi gerada com cada entrada num intervalo de 10 a 100.

Dito isto, para cada classe geramos 10 problemas com diferentes dados e os resolvemos empregando-se  $\lambda = 0.0$ ,  $\lambda = 0.5$  e  $\lambda = 1.0$ .

Apresentamos na Tabela (7) os resultados concernentes aos valores médios de cada função objetivo,  $z_1$  e  $z_2$ , bem como o tempo de CPU médio utilizado em segundos para resolver cada problema ponderado (e não limitado a 3600 segundos, mas no máximo a 100.000 nós na árvore do Branch-and-Bound).

Percebemos nitidamente o grau de conflito entre os objetivos envolvidos, pois a medida que  $z_1$  diminui,  $z_2$  aumenta e vice-versa, o que era de se esperar. Reparamos também que, em média, as soluções obtidas com  $\lambda = 0.5$  possuem, aproximadamente, a média dos valores objetivos obtidos com  $\lambda = 0.0$  e  $\lambda = 1.0$ . Outra notoriedade é o crescimento

Tabela 7 – Resultados computacionais para as 9 instâncias do Problema (2.1)-(2.15) e  $\lambda = 0.0, 0.5$  e  $1.0$

Classe	$\lambda = 0.0$			$\lambda = 0.5$			$\lambda = 1.0$		
	$z_1$	$z_2$	CPU	$z_1$	$z_2$	CPU	$z_1$	$z_2$	CPU
1	99,8	14,1	1,5	77,0	22,5	38,6	70,8	38,6	21,8
2	164,9	15,3	18,8	136,1	25,7	102,3	129,7	36,8	106,6
3	228,4	14,0	42,8	194,4	27,2	137,4	189,8	33,6	139,9
4	86,2	144,3	239,6	75,9	150,4	217,9	66,1	183,2	274,1
5	139,2	141,3	780,2	132,5	151,7	680,3	114,4	205,9	716,3
6	213,2	146,0	3198,3	197,9	158,9	3116,5	168,3	225,4	3187,3
7	172,5	241,3	3156,3	133,2	258,4	3113,2	105,3	369,7	3456,3
8	179,2	244,0	3884,3	177,2	260,6	3004,3	141,4	366,4	2918,3
9	284,5	300,4	3588,3	274,9	311,8	3653,2	210,5	403,7	3637,2

rápido do tempo computacional com o aumento de  $p, q$  e  $r$ , isto é, a complexidade de resolução deste problema é muito sensível com esses parâmetros. Para finalizar a discussão, os problemas com  $\lambda = 0.0$  são mais rapidamente resolvidos do que os com  $\lambda = 1.0$ , pelo menos nas instâncias de 1 a 4. Nas demais, essa diferença não é notável.

### 3.4 Aplicação do $\varepsilon$ -Restrito

Ilustremos a aplicação dos resultados computacionais para as Instâncias 1 e 3 da formulação matemática (2.36)-(2.40) usando-se um outro método de escalarização: o  $\varepsilon$ -Restrito.

Assim, cada subproblema restrito a seguir:

$$\text{minimize } z_1 := \sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r \sum_{\bar{k}=1}^r d_{k\bar{k}} \cdot v_{ijk\bar{k}} \quad (3.1)$$

$$\text{sujeito a } (2.3) - (2.25) \quad (3.2)$$

$$z_2 := \sum_{i=1}^p \sum_{\bar{i}=1}^p \sum_{k=1}^{r-1} \sum_{\bar{k}=k+1}^r d_{k\bar{k}} \cdot u_{i\bar{i}k\bar{k}} \leq \varepsilon \quad (3.3)$$

determina uma solução eficiente para a formulação (2.36)-(2.40). Empregamos esta ideia para gerar mais pontos dominados para a fronteira de Pareto (e consequentemente mais soluções eficientes) desta instância e mostrar que o problema dado possui muitas outras soluções que não são determinadas pela Soma Ponderada.

Para tanto, colocamos a função  $z_2$  como restrição e minimizamos a função  $z_1$ . O lado direito da função-restrição assume o  $\varepsilon$ . Sejam  $\varepsilon^-$  e  $\varepsilon^+$  os menos e maiores valores que a função  $z_2$  assume no conjunto admissível do problema. Então, fizemos 10 atribuições uniformemente distribuídas para  $\varepsilon$  no intervalo  $[\varepsilon^-, \varepsilon^+]$ . Usando as Tabelas (4), (5) e (6), vemos que  $z_2 \in [10, 63]$  na Instância 1.

Os resultados para os valores da função objetivo e tempo de processamento em cada subproblema (3.1)-(3.3) encontram-se na Tabela (8).

Tabela 8 – Resultados computacionais para o método  $\varepsilon$ -Restrito aplicado na Instância 1 para o Modelo (2.36)-(2.40)

Solução	$z_1$	$z_2$	$\varepsilon$	CPU
1	151	10	10	0,125
2	143	15	16	0,375
3	115	23	24	0,500
4	108	25	27	0,671
5	100	30	30	1,486
6	93	40	41	2,234
7	88	48	50	2,353
8	80	55	55	2,563
9	78	63	63	2,895

A representação da fronteira de Pareto que este método gerou, está apresentada na Figura (25), juntamente com os pontos fornecidos pelo método da Soma Ponderada.

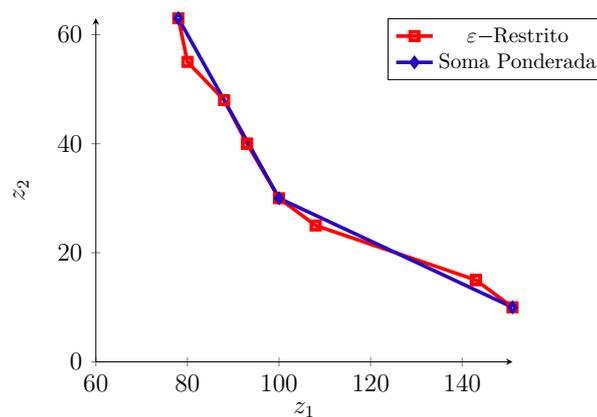


Figura 25 – Pontos não-dominados fornecidos pelo  $\varepsilon$ -Restrito e Soma Ponderada do Problema (2.1)-(2.15) da Instância 1

Note que a Soma Ponderada não iria gerar mais de 5 soluções eficientes para este problema, mesmo se usássemos muitas escolhas para  $\lambda$ . Isso porque a fronteira de Pareto para este problema nesta instância possui as soluções 2, 5 e 6 como eficientes mas não-suportadas, isto é, pertencentes à parte não-convexa da fronteira de Pareto. A vantagem do  $\varepsilon$ -Restrito está em detectar estas soluções, promovendo uma maior diversidade entre as soluções. Cabe ressaltar que, este método, resolveu 10 problemas mas determinou 8 soluções eficientes. Talvez mais de 10 atribuições de  $\varepsilon$  poderiam gerar mais soluções eficientes, por exemplo, entre as soluções 2 e 3, como se pode ver na Figura (25).

Semelhantemente, aplicamos o mesmo algoritmo na Instância 3. Neste caso, observando as Tabelas (4), (5) e (6), vemos que  $z_2 \in [10, 85]$ . Então fizemos 10 atribuições

para  $\varepsilon$  neste intervalo. Os resultados dos valores das funções-objetivo o tempo de resolução de cada subproblema restrito são apresentados na Tabela (9).

Tabela 9 – Resultados computacionais para o método  $\varepsilon$ -Restrito aplicado na Instância 3 para o Modelo (2.36)-(2.40)

Solução	$z_1$	$z_2$	$\varepsilon$	CPU
1	499	10	10	483,56
2	448	15	18	486,44
3	403	25	26	512,36
4	385	30	30	585,98
5	357	40	42	613,25
6	320	48	50	624,78
7	300	55	58	627,18
8	288	60	78	612,87
9	275	85	85	653,11

A fronteira de Pareto desta instância, gerada pelos métodos  $\varepsilon$ -Restrito e soma Ponderada estão apresentadas na Figura (26).

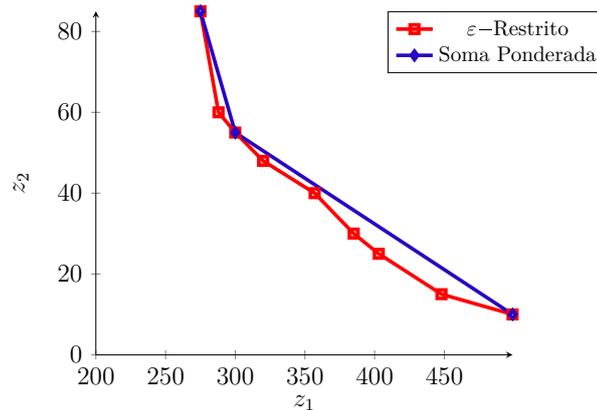


Figura 26 – Pontos não-dominados fornecidos pelo  $\varepsilon$ -Restrito e Soma Ponderada do Problema (2.1)-(2.15) da Instância 3

Novamente, por mais que aumentamos o número de pesos  $\lambda$  na soma Ponderada, não conseguiríamos mais de 6 soluções eficientes (apenas as suportadas, sendo 1, 2, 6, 7, 8 e 9). O método  $\varepsilon$ -Restrito em 10 atribuições para  $\varepsilon$  determinou 9 soluções eficientes, onde as 5 a mais tem imagem no espaço de critério sendo pontos não-suportados. Sem dúvida, é um método muito mais eficaz se o gestor deseja ter um maior rol de soluções distintas para tomar sua decisão.

Um fato interessante é que os problemas restritos ficam cada vez mais caros de serem otimizados a medida que  $\varepsilon$  aumenta. Isso está ligado ao fato do “peso” da função  $z_1$  ir aumentando a medida que  $\varepsilon$  aumenta, e como vimos anteriormente, esta função é mais difícil de ser minimizada pelo GUSEK, confirmando nossas expectativas.

## Capítulo 4

# Conclusões e Perspectivas Futuras

Este trabalho apresentou dois modelos matemáticos inéditos de alocação de centros consumidores e facilidades baseado no modelo clássico das  $p$ -medianas e  $p$ -centros. No primeiro, de natureza multi-objetivo, os clientes e as facilidades são alocados na rede, ao passo que no segundo, mono-objetivo, as facilidades têm suas localizações bem definidas e apenas os clientes devem ser alocados nos nós remanescentes. Em ambos, o fluxo da facilidade para os clientes devem ser também dimensionados, respeitando a capacidade máxima de fornecimento das facilidades e atendendo à demanda mínima de cada centro consumidor. As formulações desenvolvidas envolvem apenas variáveis binárias e as funções objetivo são não-lineares (cúbicas e quadráticas).

Para resolver o primeiro modelo, e a fim de utilizar métodos clássicos de escalarizações de otimização multi-objetivo, como Soma Ponderada e  $\varepsilon$ -Restrito usando pacotes de programação matemática linear inteira, linearizamos estas funções não-lineares, usando “truques” de programação inteira. Sobretudo, o problema cresce consideravelmente, tanto no número de variáveis quanto no número de restrições.

Este trabalho ilustrou o quão difícil foi resolver os sub-problemas gerados pelos métodos de otimização multi-objetivo, mesmo para instâncias de pequena dimensão. Isso pode ser justificado pela relaxação linear fraca e o elevado número de variáveis decisórias envolvidas. Mesmo com essa dificuldade, o trabalho mostrou que é possível gerar fronteiras de Pareto para este problema e mostrar ao decisor, um conjunto de alternativas distintas igualmente importantes, afim de auxiliá-lo em sua tomada de decisão. O trabalho mostrou também uma boa performance da técnica  $\varepsilon$ -restrito, pois determina um número maior de soluções eficientes para este problema, ao passo que a Soma Ponderada apenas determina soluções suportadas (que são em menor número).

A principal contribuição deste trabalho foram os modelos matemáticos inéditos apresentados, que podem ser aplicados em armazéns distribuidores, filiais, transportadoras, mercados varejistas, etc, que podem ter sua frota melhor dimensionada, para o atendimento da demanda de toda clientela.

Como perspectivas futuras de pesquisa, pontuamos:

- Utilizar outros pacotes para resolver os sub-problemas, como o CPLEX;
- Em vez de linearizar as formulações, empregar pacotes de programação não-linear nos sub-problemas;
- Desenvolver um algoritmo heurístico para resolver instâncias com elevada dimensão.

## Referências

- ANEJA, Y. P.; NAIR, K. P. Bicriteria transportation problem. *Management Science*, INFORMS, v. 25, n. 1, p. 73–78, 1979. Citado na página 21.
- ARENALES, M. A. V.; morabito, r. & yanasse, h. *Pesquisa Operacional—para cursos de Engenharia*. Rio de Janeiro: Elsevier, 2007. Citado 3 vezes nas páginas 17, 21 e 23.
- BARCELOS, F. B.; PIZZOLATO, N. D.; LORENA, L. A. N. Avaliação da localização de escolas com modelos capacitado e não-capacitado e uso de uma ferramenta gis: estudo de caso de vitória/es. *ES XXXIV SBPO-Rio de Janeiro-2002*, 2002. Citado na página 17.
- BASTIAANSEN, W. G.; MOLDEN, D. J.; MAKIN, I. W. Remote sensing for irrigated agriculture: examples from research and possible applications. *Agricultural water management*, Elsevier, v. 46, n. 2, p. 137–155, 2000. Citado na página 20.
- CALDAS, F. U.; MATIAS, Í. de O.; VIANNA, D. S.; SHIMOYA, A. Uso de algoritmo genético para otimização do posicionamento de racks de redes de computadores. *Revista Científica Linkania Master*, v. 1, n. 10, 2015. Citado na página 17.
- CALIK, H.; LABBÉ, M.; YAMAN, H. p-center problems. In: *Location Science*. [S.l.]: Springer, 2015. p. 79–92. Citado na página 25.
- CHEN, D.-S.; BATSON, R. G.; DANG, Y. *Applied integer programming: modeling and solution*. [S.l.]: John Wiley & Sons, 2011. Citado 9 vezes nas páginas 20, 23, 25, 26, 30, 31, 33, 35 e 37.
- CHIYOSHI, F.; GALVAO, R. D. A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research*, Springer, v. 96, n. 1-4, p. 61–74, 2000. Citado na página 24.
- COLIN, E. C. *Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas*. [S.l.]: Livros Técnicos e Científicos, 2007. Citado na página 19.
- CUNHA, C. B. da. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. *Transportes*, v. 8, n. 2, 2000. Citado na página 17.
- EHRGOTT, M. *Multicriteria Optimization*. [S.l.]: Springer-Verlag, 2000. (Lecture Notes in Economics and Mathematical Systems). Citado 2 vezes nas páginas 39 e 40.
- FILHO, A. A. et al. Novas extensões de técnicas de escalarizações no problema de corte unidimensional inteiro multiobjetivo. [sn], 2016. Citado 3 vezes nas páginas 17, 37 e 39.

- FRIEND, J.; JESSOP, N. *Local Government and Strategic Choice (Routledge Revivals): An Operational Research Approach to the Processes of Public Planning*. [S.l.]: Routledge, 2013. Citado na página 20.
- GALINDO, G.; BATTA, R. Review of recent developments in or/ms research in disaster operations management. *European Journal of Operational Research*, Elsevier, v. 230, n. 2, p. 201–211, 2013. Citado na página 20.
- GALVÃO, R. D.; NOBRE, F. F.; VASCONCELLOS, M. M. Modelos matemáticos de localização aplicados à organização espacial de unidades de saúde. *Revista de Saúde Pública*, SciELO Public Health, v. 33, n. 4, p. 422–434, 1999. Citado na página 17.
- GOPAKUMAR, A.; JACOB, L. Localization in wireless sensor networks using particle swarm optimization. IET, 2008. Citado na página 17.
- HAIMES, Y. Y.; LASDON, L. S.; WISMER, D. A. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems*, 1971. Citado na página 40.
- HAKIMI, S. L. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, informs, v. 12, n. 3, p. 450–459, 1964. Citado na página 23.
- HÖRNER, D. et al. Resolução do problema das p-medianas não capacitado: comparação de algumas técnicas heurísticas. Florianópolis, SC, 2009. Citado na página 24.
- JÄRVINEN, P.; RAJALA, J.; SINERVO, H. A branch-and-bound algorithm for seeking the p-median. *Operations Research*, INFORMS, v. 20, n. 1, p. 173–178, 1972. Citado na página 24.
- LORENA, L. A. N.; SENNE, E. L. F.; PAIVA, J. A. d. C.; PEREIRA, M. A. Integração de modelos de localização a sistemas de informações geográficas. *Gestão & Produção*, Universidade Federal de São Carlos (UFSCar), p. 180–195, 2001. Citado na página 25.
- LORENA, L. A. N.; SENNE, E. L. F.; PAIVA, J. d. C.; MARCONDES, S. P. B. Integração de um modelo de p-medianas a sistemas de informações geográficas. *In 31º Simpósio Brasileiro de Pesquisa Operacional, Juiz de Fora, MG. Anais. Rio de Janeiro, RJ: SOBRAPO*, p. 635–647, 1999. Citado na página 24.
- MARANZANA, F. On the location of supply points to minimize transport costs. *Journal of the Operational Research Society*, Taylor & Francis, v. 15, n. 3, p. 261–270, 1964. Citado na página 24.
- MIETTINEN, K. *Nonlinear Multiobjective Optimization*. [S.l.]: Kluwer Academic Publishers Dordrecht, 1999. v. 12. (International Series in Operations Research and Management Science, v. 12). Citado na página 39.
- MLADENOVIC, N.; LABBÉ, M.; HANSEN, P. Solving the p-center problem with tabu search and variable neighborhood search. *Networks*, Wiley Online Library, v. 42, n. 1, p. 48–64, 2003. Citado na página 25.
- MUNKRES, J. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, SIAM, v. 5, n. 1, p. 32–38, 1957. Citado na página 21.

- MUSA, R.; ARNAOUT, J.-P.; JUNG, H. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*, Elsevier, v. 59, n. 1, p. 85–92, 2010. Citado na página 21.
- OSHMANN, Y.; DAVIDSON, P. Optimization of observer trajectories for bearings-only target localization. *IEEE Transactions on Aerospace and Electronic Systems*, IEEE, v. 35, n. 3, p. 892–902, 1999. Citado na página 17.
- PLESNÍK, J. A heuristic for the p-center problems in graphs. *Discrete Applied Mathematics*, Elsevier, v. 17, n. 3, p. 263–268, 1987. Citado na página 25.
- ROSÁRIO, R. R. L. do; CARNIERI, C.; STEINER, M. T. A. Proposta de solução para o problema das p-medianas na localização de unidades de saúde 24 horas. *XXII Encontro Nacional de Engenharia de Produção, Curitiba*, 2002. Citado na página 24.
- SAHNI, S. Approximate algorithms for the 0/1 knapsack problem. *Journal of the ACM (JACM)*, ACM, v. 22, n. 1, p. 115–124, 1975. Citado na página 23.
- SENNE, E. L. F.; LORENA, L. A. N. Abordagens complementares para problemas de p-medianas. *Produção*, Associação Brasileira de Engenharia de Produção, p. 78–87, 2003. Citado na página 24.
- SMIDERLE, A.; STEINER, M. A.; WILHELM, V. Técnicas da pesquisa operacional aplicadas a um problema de cobertura de arcos. *Trends in Applied and Computational Mathematics*, v. 5, n. 2, p. 347–356, 2004. Citado na página 24.
- STEADIESEIFI, M.; DELLAERT, N. P.; NUIJTEN, W.; WOENSEL, T. V.; RAOUFI, R. Multimodal freight transportation planning: A literature review. *European journal of operational research*, Elsevier, v. 233, n. 1, p. 1–15, 2014. Citado na página 20.
- VIGNAUX, G. A.; MICHALEWICZ, Z. A genetic algorithm for the linear transportation problem. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 21, n. 2, p. 445–452, 1991. Citado na página 21.
- ZACHARIAH, R.; HARRIES, A. D.; ISHIKAWA, N.; RIEDER, H. L.; BISSELL, K.; LASERSON, K.; MASSAQUOI, M.; HERP, M. V.; REID, T. Operational research in low-income countries: what, why, and how? *The Lancet Infectious Diseases*, v. 9, n. 11, p. 711 – 717, 2009. ISSN 1473-3099. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1473309909702294>>. Citado na página 20.
- ZIKMUND, W. G.; BABIN, B. J.; CARR, J. C.; GRIFFIN, M. *Business research methods*. [S.l.]: Cengage Learning, 2013. Citado na página 20.