

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Luis Fernando Nazari

**UM MODELO REVISADO PARA ESCALONAMENTO DE  
COMPRESSORES EM CAMPOS DE PETRÓLEO:  
DESIGUALDADES VÁLIDAS E ALGORITMO DE PLANOS DE  
CORTE**

Florianópolis

2011



Luis Fernando Nazari

**UM MODELO REVISADO PARA ESCALONAMENTO DE  
COMPRESSORES EM CAMPOS DE PETRÓLEO:  
DESIGUALDADES VÁLIDAS E ALGORITMO DE PLANOS DE  
CORTE**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Dr. Eduardo Camponogara

Florianópolis

2011



Luis Fernando Nazari

**UM MODELO REVISADO PARA ESCALONAMENTO DE  
COMPRESSORES EM CAMPOS DE PETRÓLEO:  
DESIGUALDADES VÁLIDAS E ALGORITMO DE PLANOS DE  
CORTE**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 07 de dezembro 2011.

---

Prof. Dr. José Eduardo Ribeiro Cury  
Coordenador do Curso

---

Prof. Dr. Eduardo Camponogara  
Orientador

**Banca Examinadora:**

---

Prof. Dr. Eduardo Camponogara  
Presidente

---

Prof. Dr. Daniel Juan Pagano



---

Prof. Dr. Hector Bessa Silveira

---

Prof. Dr. Juliano de Bem Francisco



Dedico este trabalho em memória de Cristina  
Elis Moratelli.



## AGRADECIMENTOS

Agradeço aos meus pais, Luiz Carlos Nazari e Roseli Terezinha Nazari, por quem não tenho palavras para expressar meu amor e que mais uma vez proporcionaram-me a chance de obter um futuro melhor. A minha namorada, Raíssa Spindola, Familiares e Amigos, pelo incentivo, carinho, amor e por sempre acreditarem em minha capacidade. Agradeço em especial a meu orientador, Eduardo Camponogara, professor e amigo, pela paciência, empenho, dedicação e por compartilhar um pouco de seu vasto conhecimento. Agradeço a instituição Petrobrás pelo incentivo a pesquisa feito em diversas universidades do país, este incentivo torna o país ainda melhor. Agradeço a banca examinadora, por disponibilizarem seu tempo e aceitarem o meu convite, e a todos os professores do Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina que contribuíram em minha formação acadêmica.







## RESUMO

O projeto e escalonamento em tempo real de compressores para operar poços de petróleo com o método de elevação por injeção contínua de gás, consiste em resolver um problema não-linear inteiro misto que pode ser visto como um problema de alocação de instalações. Esta dissertação apresenta uma formulação revisada que representa as restrições de pressão de descarga do compressor como uma família de desigualdades lineares, mostrando-se mais efetiva que a formulação antecedente (CAMPONOGARA; CASTRO; PLUCENIO, 2007). Para a resolução do problema, aproximações lineares inteiras mistas da formulação revisada serão obtidas. Isto ocorrerá através da linearização por partes das funções não lineares presentes na formulação, usando variáveis binárias e um conjunto especialmente ordenado de variáveis de tipo II, SOS2. Serão propostas durante o decorrer do trabalho desigualdades válidas para a capacidade do compressor, buscando ganhos computacionais durante a resolução do problema. Este trabalho apresenta também experimentos computacionais comparando as formulações antiga e revisada, avaliações dos impactos ocasionados pela geração de planos de cortes no tempo de resolução e algoritmos de linearização por partes de funções não lineares, baseados em programação dinâmica.

**Palavras-chave:** Escalonamento de compressores. Taxa de injeção de gás. Programação linear inteira mista. Linearização por partes. Planos de corte. Desigualdades de cobertura.



## ABSTRACT

The design and real-time scheduling of lift-gas compressors in oil fields entails solving a mixed-integer nonlinear problem that generalizes the facility location problem. This work presents a revised formulation that represents the constraints on compressor discharge pressure as a family of linear inequalities, which is shown to be tighter than a previous formulation (CAMPO-NOGARA; CASTRO; PLUCENIO, 2007). The problem is approximates solved by piecewise-linearizing the nonlinear functions, using binary variables and specially ordered set of variables. Valid inequalities will be proposed during the course of the work for the compressor capacity, seeking computational gains on resolution of the problem. The work also presents computational experiments comparing the formulations, evaluating the impact of cutting-plane generation on solution speed and piecewise-linearization algorithms for nonlinear functions based on dynamic programming.

**Keywords:** Compressor scheduling. Artificial lifting by continuous gas injection. Piecewise linearization. Cutting-planes. Cover inequalities. Mixed-integer linear programming.



## LISTA DE FIGURAS

Figura 1	Poliedros $P_1$ e $P_2$ . . . . .	32
Figura 2	Poliedros $P_2$ e $Conv(X)$ . . . . .	32
Figura 3	Branch-and-Bound 1. . . . .	35
Figura 4	Branch-and-Bound 2. . . . .	36
Figura 5	Branch-and-Bound 3. . . . .	36
Figura 6	Ilustração da Desigualdade Válida Para o Problema da Mochila, com $x_1 = x_3 = x_5 = 0$ . . . . .	38
Figura 7	Problema de Escalonamento de Compressores. . . . .	41
Figura 8	Processo de Elevação por Injeção Contínua de Gás. . . . .	42
Figura 9	Visão Geométrica da Restrição de Pressão. . . . .	47
Figura 10	Curva de Pressão de Cinco Compressores Típicos. . . . .	48
Figura 11	Aproximação Linear por Partes para a Instância Exemplo com 3 Segmentos. . . . .	56
Figura 12	Aproximação Linear por Partes para a Instância Exemplo com 4 Segmentos. . . . .	59
Figura 13	Linearização por Partes da Instância Exemplo com 4 Segmentos e Garantia de Continuidade nos Pontos de $P$ . . . . .	60
Figura 14	Linearização por Partes da Instância Exemplo com 3 Segmentos de Reta e Continuidade Assegurada por Restrições. . . . .	62
Figura 15	Linearização por Partes da Instância Exemplo com 3 Segmentos de Reta que Definem o Envelope Superior dos Pontos. . . . .	63
Figura 16	Linearização por Partes da Instância Exemplo com 3 Segmentos de Reta que Definem o Envelope Superior dos Pontos e com Restrição de Continuidade. . . . .	64
Figura 17	Modelo Linear por Partes de Sherali. . . . .	67
Figura 18	Modelo Linear por Partes SOS2. . . . .	68



## LISTA DE TABELAS

Tabela 1	Exemplo de Instância para um Problema de Otimização.....	29
Tabela 2	Dados do Compressor.....	49
Tabela 3	Parâmetros da Curva de Pressão.....	49
Tabela 4	Dados dos Poços.....	49
Tabela 5	Dados das Linhas.....	50
Tabela 6	Taxa Máxima de Saída dos Compressores para o Atendimento dos Poços.....	50
Tabela 7	Conjunto de Pontos para Linearização por Partes.....	55
Tabela 8	Valores de $F(j, t)$ e $I(j, t)$ para a Instância Exemplo.....	55
Tabela 9	Valores de $F(j, t)$ e $I(j, t)$ para a Instância Exemplo.....	58
Tabela 10	Comparação Entre as Formulações I.....	76
Tabela 11	Comparação Entre as Formulações II.....	77
Tabela 12	Análise da Desigualdade de Cobertura I.....	78
Tabela 13	Análise da Desigualdade de Cobertura II.....	78
Tabela 14	Comparação Entre as Formulações Com/Sem Variáveis Binárias.....	79
Tabela 15	Comparação Entre as Formulações Com/Sem Variáveis Binárias.....	79
Tabela 16	Dados do Compressor I.....	89
Tabela 17	Parâmetros da Curva de Pressão I.....	89
Tabela 18	Dados dos Poços I.....	90
Tabela 19	Dados da Perda de Pressão nas Linhas I.....	90
Tabela 20	Dados dos Custos das Linhas I.....	91
Tabela 21	Dados do Compressor II.....	95
Tabela 22	Parâmetros da Curva de Pressão II.....	95
Tabela 23	Dados dos Poços II.....	96
Tabela 24	Dados da Perda de Pressão nas Linhas II.....	97
Tabela 25	Dados dos Custos das Linhas II.....	98
Tabela 26	Dados do Compressor III.....	101
Tabela 27	Parâmetros da Curva de Pressão III.....	101
Tabela 28	Dados dos Poços III.....	102
Tabela 29	Dados da Perda de Pressão nas Linhas III.....	102

Tabela 30 Dados dos Custos das Linhas III ..... 103

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	23
1.1 PROBLEMÁTICA .....	23
1.2 OBJETIVOS .....	24
1.3 REVISÃO DA LITERATURA .....	24
1.4 VISÃO GERAL DA DISSERTAÇÃO .....	26
<b>2 FUNDAMENTOS DE PROGRAMAÇÃO INTEIRA</b> .....	27
2.1 INTRODUÇÃO À OTIMIZAÇÃO .....	27
2.2 PROGRAMAÇÃO LINEAR INTEIRA .....	30
2.2.1 Formulações Alternativas .....	30
2.2.2 Otimalidade, Relaxação e Limites .....	33
2.2.3 Branch-and-Bound .....	34
2.2.4 Planos de Corte .....	37
2.2.5 Desigualdades de Coberturas e <i>Lifting</i> de Desigualdades Válidas .....	38
2.3 SUMÁRIO .....	40
<b>3 DEFINIÇÃO DO PROBLEMA</b> .....	41
3.1 FORMULAÇÃO NÃO LINEAR REVISADA .....	44
3.2 INSTÂNCIA EXEMPLO .....	48
3.3 SUMÁRIO .....	50
<b>4 MODELOS E ALGORITMOS PARA LINEARIZAÇÃO POR PARTES</b> .....	51
4.1 INTRODUÇÃO AO PROBLEMA DE LINEARIZAÇÃO POR PARTES .....	51
4.2 MODELO COM NÚMERO FIXO DE SEGMENTOS .....	52
4.3 MODELO COM NÚMERO MÁXIMO DE SEGMENTOS .....	56
4.4 APROXIMAÇÃO LINEAR POR PARTES CONTÍNUA .....	58
4.4.1 Assegurando Continuidade com os Pontos de Quebra .....	59
4.4.2 Garantindo Continuidade Através de Restrições .....	60
4.5 INDUZINDO ENVELOPE SUPERIOR E INFERIOR .....	62
4.6 SUMÁRIO .....	63
<b>5 FORMULAÇÕES LINEARES POR PARTES</b> .....	65
5.1 FORMULAÇÃO LINEAR POR PARTES UTILIZANDO VARIÁVEIS BINÁRIAS .....	65
5.2 FORMULAÇÃO LINEAR POR PARTES SEM A UTILIZAÇÃO DE VARIÁVEIS BINÁRIAS .....	67
5.3 PLANOS DE CORTE .....	68
5.3.1 Desigualdades de Cobertura de Instalações .....	69

<b>5.3.2 Lifting para a Desigualdade de Cobertura de Instalações . . .</b>	<b>70</b>
<b>5.3.3 Problema de Separação das Desigualdades de Cobertura . . .</b>	<b>72</b>
5.4 SUMÁRIO.....	74
<b>6 ANÁLISE NUMÉRICA .....</b>	<b>75</b>
6.1 COMPARAÇÃO ENTRE AS FORMULAÇÕES ANTIGA E RE- VISADA .....	75
6.2 ANÁLISE COMPUTACIONAL DAS DESIGUALDADES DE CO- BERTURA DE INSTALAÇÕES .....	77
6.3 COMPARAÇÃO ENTRE AS FORMULAÇÕES LINEARES POR PARTES COM E SEM A UTILIZAÇÃO DE VARIÁVEIS BINÁ- RIAS .....	79
6.4 SUMÁRIO.....	80
<b>7 CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>81</b>
<b>REFERÊNCIAS .....</b>	<b>83</b>
<b>ANEXO A - Instância com 7 Compressores e 16 poços.....</b>	<b>89</b>
<b>ANEXO B - Instância com 8 Compressores e 18 poços.....</b>	<b>95</b>
<b>ANEXO C - Instância com 9 Compressores e 14 poços.....</b>	<b>101</b>

# 1 INTRODUÇÃO

## 1.1 PROBLEMÁTICA

A cada dia que passa a demanda de energia no mundo cresce mais e mais, aumentando o consumo e dependência do petróleo. Apesar de ainda existirem em grandes quantidades em nosso planeta, esta fonte é não renovável e no futuro se esgotará.

Recentemente foram descobertos grandes reservatórios de óleo abaixo de uma camada de sal, situados em uma camada chamada de pré-sal. Estima-se que esta camada estende-se por 800 quilômetros do litoral brasileiro desde Santa Catarina até Espírito Santo, podendo chegar a 200 quilômetros de largura, contendo 1,6 trilhão de metros cúbicos de gás e óleo. Porém existem diversas dificuldades para sua extração, o que torna o custo muito alto e ainda não viável à produção de petróleo.

Motivados a buscar formas mais eficientes para extração de óleo, as indústrias petrolíferas buscam usufruir ao máximo de um campo de petróleo e diminuir os custos de operação deste campo. A chave para estes ganhos é a tecnologia de *campos inteligentes*, também conhecidos como *campos digitais* e *campos do futuro*, que defendem a integração dos ativos e operações para fornecer uma operação do sistema completo (DEVRIES, 2005; MURRAY et al., 2006). O uso deste modelo de sistema completo implica no desenvolvimento de modelos matemáticos e algoritmos eficientes para resolver problemas de operação em campos inteligentes (CAMPONOGARA et al., 2010). Um dos problemas envolvidos é o projeto e escalonamento em tempo real de compressores para operar campos de petróleo com o método de elevação por injeção contínua de gás, objetivo do trabalho.

Técnicas artificiais de elevação de petróleo são aplicadas em todo o mundo. Estas técnicas servem para trazer o petróleo à superfície quando a pressão interna do reservatório não é suficientemente alta para elevar o petróleo às instalações de superfície. Uma destas técnicas de elevação é a injeção contínua de gás de alta pressão no fundo do tubo de produção, conhecida como *gas-lift*. Esta técnica é muito interessante, pois proporciona uma ampla gama de condições operacionais e custo de instalação e manutenção relativamente baixo.

## 1.2 OBJETIVOS

O plano de extração de petróleo de um campo, que utiliza o método de elevação por injeção contínua de gás, estabelece taxas de injeção de gás e pressões de injeção para os poços. Estas taxas e pressões podem variar ao longo do tempo em resposta a eventos imprevisíveis, tais como falhas de equipamentos e manutenções forçadas, mudanças dinâmicas nas condições do reservatório e oscilações dinâmicas na disponibilidade de gás, entre outros. Consequentemente, o problema de projeto e escalonamento em tempo real de compressores para plataformas de produção de petróleo é fundamental para a otimização da produção.

Esta dissertação tem como um dos seus objetivos propor um modelo matemático revisado do projeto e escalonamento de compressores que suprem a demanda de gás de elevação, definir a instalação ou ativação de compressores, escaloná-los para suprir a demanda dos poços com objetivo de alcançar um ótimo econômico do processo. Este modelo é baseado no modelo proposto em (CAMPONOVARA et al., 2011; CAMPONOVARA; CASTRO, 2008). O modelo antecedente de (CAMPONOVARA et al., 2011; CAMPONOVARA; CASTRO, 2008) possui dentre suas restrições funções não-lineares o que dificulta sua resolução, portanto o modelo revisado busca contornar este problema e melhorar o modelo anterior.

Já para a resolução do problema será utilizada uma modelagem de otimização baseada em programação inteira mista, pois tal classe possui algoritmos que podem encontrar uma solução ótima do sistema. Para que seja possível esta formulação será necessária a linearização por partes das funções não-lineares, aproximando a curva por vários segmentos de reta. Com este objetivo foram desenvolvidos algoritmos para se obter uma linearização ótima levando em conta o número desejado, ou máximo, de segmentos de retas a serem usados e tendo como função objetivo a minimização de erros quadráticos.

Ao final, outro objetivo do trabalho é propor algoritmos de planos de corte, baseados em desigualdades de cobertura (WOLSEY, 1998), para o problema de escalonamento de compressores com a finalidade de se obter ganhos computacionais durante sua resolução.

## 1.3 REVISÃO DA LITERATURA

O problema de escalonamento de compressores tem sido abordado de forma breve em alguns trabalhos (HUPPLER, 1974; DUTTA-ROY; BARUA; HEIBA, 1997). O trabalho de (DUTTA-ROY; BARUA; HEIBA, 1997) já relaciona os poços,

linhas de fluxo, instalações de superfície e reservatório para a construção de seu modelo. Neste trabalho assume-se como função objetivo a diferença entre os rendimentos gerados pela venda do petróleo, ou gás, e o custo de instalação e operação dos compressores, envolvendo o custo da energia necessária para a operação dos compressores. A técnica utilizada por (DUTTA-ROY; BARUA; HEIBA, 1997) para a resolução do problema com a finalidade de obter ganhos econômicos é conhecida por programação quadrática sucessiva. Esta técnica é um ótimo mecanismo para se encontrar ótimos locais, porém não assegura a determinação de um ótimo global. Já em (HUPPLER, 1974), o problema de escalonamento de compressores é relacionado de uma forma um pouco diferente, onde dada uma demanda de gás exigida pelos poços e uma capacidade de fornecimento de gás, deve-se otimizar o escalonamento de compressores para que se possa perfurar mais poços, considerando a instalação de novos compressores ou não. Porém, (HUPPLER, 1974) considera que a pressão do reservatório é uniforme, o que determina que a pressão necessária fornecida pelo compressor para cada poço do reservatório é a mesma. O problema então, torna-se um problema de decisão de se perfurar novos poços ou não, e o qual pode ser resolvido utilizando-se programação dinâmica.

Após abordagens mais superficiais, o problema de escalonamento de compressores foi definido formalmente como uma generalização do problema de alocação de instalações com duas restrições de capacidades e uma função objetivo não-convexa (CAMPONOGARA; CASTRO; PLUCENIO, 2007; CAMPONOGARA; CASTRO, 2008; CAMPONOGARA et al., 2011). O problema relacionado de selecionar compressores para minimizar o custo de energia, enquanto mantém-se a pressão suficientemente alta nas linhas de conexão das refinarias é abordado em (NGUYEN et al., 2008). Este trabalho compara métodos de resolução para o problema, como algoritmos de programação inteira-mista, algoritmos genéticos e sistemas inteligentes.

O problema de alocação de gás de elevação para um grupo de poços sujeito a restrições de gás e de instalações, também tem sido abordado na literatura (SAEPUDIN et al., 2010; CAMPONOGARA; NAKASHIMA, 2006; CAMPONOGARA; CONTO, 2009).

O trabalho de (CAMPONOGARA; CASTRO; PLUCENIO, 2007) reformulou o problema de escalonamento de compressores como um problema de programação inteira mista (PLIM) pela linearização por partes da função objetivo e da restrição de pressão de descarga dos compressores. A dificuldade de se resolver instâncias representativas do problema utilizando solvers de PLIM, motivou o estudo de planos de cortes baseados em desigualdades de cobertura (WOLSEY, 1998) para a pressão de descarga do compressor e a taxa de injeção de gás. Estes algoritmos de planos de cortes foram propostos em (CAMPONOGARA et al., 2011).

Experimentos numéricos indicaram que a resolução do problema com a utilização dos algoritmos de planos de corte pode reduzir o tempo de solução em torno de 35% à 50%. Porém, apesar do uso dos planos de corte, um solver de PLIM levou mais de três horas para resolver algumas instâncias e ainda não pode lidar com instâncias de grande porte. Recentemente, surgiu a ideia de que esta formulação PLIM do problema de escalonamento de compressores é simplificado por uma representação equivalente das restrições sobre a pressão do compressor com uma família de desigualdades lineares. Esta formulação alternativa motivou o trabalho aqui relatado.

#### 1.4 VISÃO GERAL DA DISSERTAÇÃO

Esta dissertação está dividida em sete capítulos. No próximo capítulo serão apresentados, de forma breve, alguns conceitos de otimização necessários para um melhor entendimento do trabalho. No Capítulo 3 apresentaremos detalhadamente o problema de escalonamento de compressores e sua formulação revisada, comparando-a com a formulação antiga neste mesmo momento. No Capítulo 4 serão apresentados algoritmos baseados em programação dinâmica para linearização por partes de funções não lineares. O Capítulo 5 destina-se às formulações lineares por partes do problema de escalonamento de compressores, transformando o problema em um problema de programação inteira mista. Neste mesmo capítulo também serão apresentados os planos de corte desenvolvidos. O Capítulo 6 é composto por experimentos numéricos para demonstrar a efetividade da formulação revisada, comparado-a à formulação antiga e o impacto dos planos de corte no tempo de resolução do problema. Por fim, no Capítulo 7 apresentaremos as conclusões deste trabalho e perspectivas de trabalhos futuros.

## 2 FUNDAMENTOS DE PROGRAMAÇÃO INTEIRA

### 2.1 INTRODUÇÃO À OTIMIZAÇÃO

A otimização é uma área da matemática aplicada que busca obter o comportamento ótimo de um processo. Este processo deve conter variáveis de decisão, onde a escolha de seus valores influenciará diretamente no comportamento do processo. Estes valores são determinados através de um conjunto de informações, compostas por uma ou mais funções objetivo e restrições impostas às variáveis. Este conjunto é chamado de *modelo do processo*.

O modelo de um processo serve para caracterizar algo real e nos permite o estudo do processo com custos reduzidos. Um modelo pode ser *concreto*, como a miniatura de um avião utilizada em testes em um túnel de vento, ou *abstrato*, que usualmente são representações matemáticas da realidade em estudo. É válido ressaltar que a modelagem de um processo não é trivial e a diferença entre a solução do modelo e a solução real depende diretamente da precisão do modelo em descrever a realidade.

A modelagem utilizada em problemas de otimização é a modelagem matemática. Um modelo matemático contém os seguintes elementos, (ANDRADE, 1998):

1. **variáveis de decisão e parâmetros:** as variáveis de decisão são os valores a serem determinados pela solução do problema e os parâmetros são valores que possuem aspectos fixos, ou melhor, são valores que podem variar mas não durante a resolução do problema;
2. **restrições:** são funções que procuram expressar as limitações físicas do sistema. O conjunto de restrições definem o espaço das soluções que podem ser aplicadas ao problema;
3. **função objetivo:** é a função que define a qualidade da solução em relação às variáveis de decisão.

Um problema geral de otimização pode ser escrito em programação matemática como:

$$\begin{aligned}
 P : & \text{Maximize } f(x) \\
 & \text{Sujeito a: } g(x) \leq 0 \\
 & x \in \mathbb{R}^n
 \end{aligned}$$

onde  $x$  representa o vetor de variáveis de decisão,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função

objetivo e  $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$  são as restrições do problema que limitam o espaço de soluções factíveis.

Como ilustração, pode-se apresentar um problema onde uma pessoa deseja escolher entre diversos produtos de um supermercado quais levar para sua casa. Os produtos devem ser escolhidos de forma que atendam ao máximo suas necessidades no dia-a-dia, sendo que eles não devem exceder a quantia de dinheiro disponível.

Os dados do problema são:

- $N$  é o número de produtos disponíveis;
- $c_n$  é um valor que representa a utilidade associada ao item  $n$  para o dia-a-dia;
- $b$  é a quantia disponível de dinheiro;
- $a_n$  é o preço do produto  $n$ ;

Na especificação do problema, devemos definir as variáveis de decisão, a função objetivo e as restrições:

**Variáveis de decisão:**  $x_n$  assume valor 1 se o produto  $n$  for comprado, e 0 caso contrário;

**Função objetivo:** O valor da utilidade total dos produtos selecionados  $f(x) = \sum_{n=1}^N c_n x_n$ , que deve ser maximizado.

**Restrições:** O preço total dos produtos selecionados não deve exceder a quantia disponível e as variáveis de decisão devem assumir valores binários do conjunto  $\mathbb{B} = \{0, 1\}$ .

Em programação matemática, o problema fica:

$$\begin{aligned}
 P_M : \text{Maximize } & \sum_{n=1}^N c_n x_n \\
 \text{Sujeito a: } & \sum_{n=1}^N a_n x_n \leq b \\
 & x_n \in \mathbb{B} \qquad n = 1, \dots, N
 \end{aligned}$$

Dado o modelo, chamamos de *instância* o conjunto de dados que caracterizam um problema. Para o exemplo acima uma instância é dada na Tabela 1. Considerando que a quantia disponível seja  $b = \text{R\$ } 6,00$ , obtemos a seguinte formulação em programação matemática:

Tabela 1 – Exemplo de Instância para um Problema de Otimização.

$n$	Item	Utilidade	Preço
1	Pasta Dental	2	R\$ 5,00
2	Carne	7	R\$ 5,00
3	Sabonete	2	R\$ 4,00
4	Salada	4	R\$ 3,00
5	Pão	3	R\$ 2,00
6	Arroz	7	R\$ 2,00
7	Água	6	R\$ 1,00

$$\text{Maximize } 2x_1 + 7x_2 + 2x_3 + 4x_4 + 3x_5 + 7x_6 + 6x_7 \quad (2.1)$$

$$\text{Sujeito a: } 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 \leq 6$$

$$x \in \mathbb{B}^7.$$

onde sua solução é  $x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0, x_6 = 1$  e  $x_7 = 1$ .

Conforme as características da função objetivo e das restrições, os problemas de otimização podem ser classificados de diferentes formas:

- **Programação Linear:** é uma classe de problemas de otimização onde tanto a função objetivo quanto as restrições são funções lineares. Os problemas de programação linear são interessantes pelo fato de já obterem algoritmos eficientes para sua resolução, como exemplo, o algoritmo Simplex e os métodos de ponto interior (VANDERBEI, 1996);
- **Programação Linear Inteira:** também conhecida como programação inteira, é um sub-conjunto dos problemas de programação linear. Porém, neste caso, além das restrições e o objetivo serem lineares, é preciso que as variáveis de decisão sejam estritamente inteiras;
- **Programação Linear Inteira-Mista:** também conhecida como programação inteira mista, é uma classe de problemas que contém tanto variáveis de decisão inteiras quanto contínuas, além das características de um problema de programação linear;
- **Programação Não-Linear:** é uma classe de problemas mais geral. Ela contém funções não-lineares tanto nas funções objetivo quanto nas restrições, podendo conter variáveis de decisão contínuas ou não.

## 2.2 PROGRAMAÇÃO LINEAR INTEIRA

Durante o decorrer do trabalho serão necessários alguns conceitos de programação inteira que podem ser estendidos a problemas de programação inteira-mista. Estes conceitos serão brevemente abordados nessa seção para obter o maior entendimento do trabalho. Para abordagens mais profundas das definições expostas a seguir, recomenda-se a leitura de (WOLSEY, 1998).

A programação inteira, como comentado anteriormente, é um problema de programação matemática que além de possuir sua função objetivo e restrições lineares, possui suas variáveis de decisões estritamente inteiras. Em programação matemática, podemos expressar um problema de programação inteira como:

$$\begin{aligned}
 P_I : \text{Maximize } & f(x) \\
 \text{Sujeito a: } & g(x) \leq 0 \\
 & x \in \mathbb{Z}^n
 \end{aligned}$$

onde  $x$ ,  $f$  e  $g$  são respectivamente o vetor de variáveis de decisão, função objetivo (linear) e restrições (funções lineares). É importante ressaltar aqui, que a solução do problema acima considerando o vetor de variáveis de decisão  $x \in \mathbb{R}^n$  pode não ser a mesma, e ainda, pode estar muito distante da solução do problema  $P_I$ .

### 2.2.1 Formulações Alternativas

Considere o problema de minimizar uma função linear  $f(x_1, x_2)$ , tal que  $x \in X$ , com  $X = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$ . Observe agora a seguintes formulações:

$$P_1 : \min f(x_1, x_2) \quad (2.2a)$$

sujeito a:

$$3,25x_1 - x_2 \geq 0,875 \quad (2.2b)$$

$$0,7273x_1 + x_2 \leq 5.472 \quad (2.2c)$$

$$x_1 \leq 3,8 \quad (2.2d)$$

$$0,75 \leq x_2 \leq 3 \quad (2.2e)$$

$$x_1 \in \mathbb{Z} \quad (2.2f)$$

$$x_2 \in \mathbb{Z} \quad (2.2g)$$

$$P_2 : \min f(x_1, x_2) \quad (2.3a)$$

sujeito a:

$$1,4x_1 + x_2 \geq 2,334 \quad (2.3b)$$

$$1.0417x_1 - x_2 \geq -1.3333 \quad (2.3c)$$

$$2,5x_1 + x_2 \leq 11 \quad (2.3d)$$

$$0,7362x_1 - x_2 \leq 1.2975 \quad (2.3e)$$

$$x_1 \in \mathbb{Z} \quad (2.3f)$$

$$x_2 \in \mathbb{Z} \quad (2.3g)$$

**Definição 1** Um subconjunto do  $\mathbb{R}^n$  descrito por um conjunto finito de restrições lineares  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$  é um poliedro.

**Definição 2** Um poliedro  $P \subseteq \mathbb{R}^{n+p}$  é uma formulação para um conjunto  $X \subseteq \mathbb{Z}^n \times \mathbb{R}^p$  se e somente se  $X = P \cap (\mathbb{Z}^n \times \mathbb{R}^p)$ .

Note através da Figura 1 que, tanto o poliedro  $P_1$ , quanto o poliedro de  $P_2$ , representado por retas tracejadas, juntamente com a restrição de que  $(x_1, x_2) \in \mathbb{Z}^2$ , são formulações para o conjunto  $X$ . Ou seja, representam o conjunto de *pontos factíveis* do problema. Portanto, ambas formulações são corretas. Observe ainda que este mesmo problema possui infinitas formulações. Cabe agora determinar qual a melhor formulação.

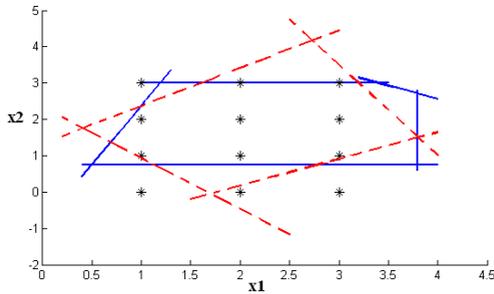


Figura 1 – Poliedros  $P_1$  e  $P_2$ .

**Definição 3** Dado um conjunto  $X \subseteq \mathbb{R}^n$  e duas formulações  $P_1$  e  $P_2$  para  $X$ ,  $P_1$  é uma formulação melhor que  $P_2$  se  $P_1 \subset P_2$ .

**Definição 4** O fecho convexo ou envelope convexo de um conjunto  $X \in \mathbb{R}^n$  é o menor conjunto convexo que contém  $X$ .

**Proposição 1** Todos os pontos extremos de  $\text{conv}(X)$ , o fecho convexo, pertencem a  $X$ .

Para o exemplo ilustrado na Figura 1, o fecho convexo de  $X$  é apresentado na Figura 2, por retas tracejadas, juntamente com o poliedro  $P_2$ . Observe que  $\text{conv}(X) \subset P_2$ . Logo a formulação  $\text{conv}(X)$  é melhor que a formulação  $P_2$ .

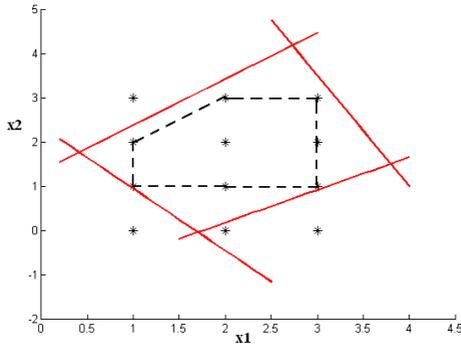


Figura 2 – Poliedros  $P_2$  e  $\text{Conv}(X)$ .

### 2.2.2 Otimalidade, Relaxação e Limites

Dado um problema de programação inteira (PI)

$$z = \max \{c^T x : x \in X \subseteq \mathbb{Z}^n\},$$

surge a questão de como provar que uma solução  $x^*$  do problema é ótima. Uma forma de responder esta questão também nos proporciona um critério de parada para algoritmos de solução desta classe. Deve-se encontrar um limite inferior  $\underline{z} \leq z$  e um limite superior  $\bar{z} \geq z$ , e caso  $\underline{z} = z = \bar{z}$ , obtemos a otimalidade. Basicamente, qualquer algoritmo deverá encontrar uma sequência decrescente de limites superiores

$$\bar{z}_1 \geq \bar{z}_2 \geq \bar{z}_3 \geq \dots \geq \bar{z}_n \geq z$$

e uma sequência crescente de limites inferiores

$$\underline{z}_1 \leq \underline{z}_2 \leq \underline{z}_3 \leq \dots \leq \underline{z}_m \leq z.$$

Dado  $\epsilon \geq 0$ , podemos definir um critério de parada como  $|\underline{z}_m - \bar{z}_n| \leq \epsilon$ , onde podemos ter certeza que a solução candidata encontrada neste momento está a no máximo  $\epsilon$  unidades da solução ótima.

**Limite Primal:** Qualquer solução factível  $x \in X$  resulta em um limite primal, que no caso de uma função objetivo de maximização representa um limite inferior. Encontrar uma solução factível pode ser uma tarefa fácil ou algumas vezes tão trabalhosa quanto resolver o problema.

**Limite Dual:** No caso de um problema de maximização este limite representará um limite superior. Caso contrário, representará um limite inferior para o problema. O método mais utilizado para obtê-lo é chamado de relaxação. Esta abordagem busca facilitar a resolução do problema obtendo uma solução no mínimo tão grande quanto a solução do problema original  $z$  (caso de maximização). Existem vários métodos de relaxação. A que será utilizada mais adiante no trabalho é a relaxação linear.

**Definição 5** Para um problema de programação linear  $P_I = \max \{c^T x : x \in P \cap \mathbb{Z}^n\}$  com o poliedro  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ , sua relaxação linear é  $P_R = \max \{c^T x : x \in P\}$ .

Observe que a relaxação linear de um problema de programação inteira resulta em um problema de programação linear, pois desconsideramos a restrição de integridade e, como comentado, este problema pode ser facilmente resolvido.

Uma vez obtido o limite dual devemos reduzi-lo. Isto pode ser feito

através de algoritmos de planos de corte e algoritmos *branch-and-bound*. Este último, além de reduzir o limite dual, pode ao mesmo tempo encontrar e elevar o limite primal.

### 2.2.3 Branch-and-Bound

O algoritmo *branch-and-bound* é baseado na estratégia de dividir e conquistar, ou seja, procura dividir o problema principal em sub-problemas menores e de resolução mais simples. Esta abordagem busca, através da união das soluções dos sub-problemas, solucionar o problema original. Como o algoritmo não será usado diretamente no trabalho, ilustraremos o funcionamento do mesmo através do seguinte exemplo:

$$S : z = \max 4x_1 - x_2 \quad (2.4a)$$

sujeito a:

$$7x_1 - 2x_2 \leq 14 \quad (2.4b)$$

$$x_2 \leq 3 \quad (2.4c)$$

$$2x_1 - 2x_2 \leq 3 \quad (2.4d)$$

$$x \in \mathbb{Z}_+^2 \quad (2.4e)$$

**Bounding:** O primeiro passo na resolução do problema utilizando o algoritmo *branch-and-bound* é encontrar um limite superior. Para este fim aplicaremos a abordagem da relaxação linear. Aplicado a relaxação, deve-se resolver o problema de programação linear resultante. O limite superior encontrado para o problema acima usando o algoritmo simplex é  $\bar{z} = \frac{59}{7}$ , com a solução  $(x_1, x_2) = (\frac{20}{7}, 3)$ . Note que a solução encontrada não é factível. Por convenção, como nenhuma solução factível é conhecida, será assumido o limite primal como  $\underline{z} = -\infty$ .

**Branching:** Como  $\underline{z} \leq \bar{z}$  vamos dividir, ou melhor, ramificar. A ideia para aplicar a ramificação é escolher uma das variáveis não-inteiras e dividir o problema original em dois sub-problemas do tipo:

$$S_1 = S \cap \{x : x_j \leq \lfloor \bar{x}_j \rfloor\}$$

$$S_2 = S \cap \{x : x_j \geq \lceil \bar{x}_j \rceil\}$$

onde  $x_j = \bar{x}_j \notin \mathbb{Z}$ . Claramente  $S = S_1 \cup S_2$  e  $S_1 \cap S_2 = \emptyset$ . Como  $x_1 = \frac{20}{7}$ ,

temos

$$S_1 = S \cap \{x : x_1 \leq 2\} \text{ e}$$

$$S_2 = S \cap \{x : x_1 \geq 3\}.$$

Observe na Figura 3 a primeira ramificação.

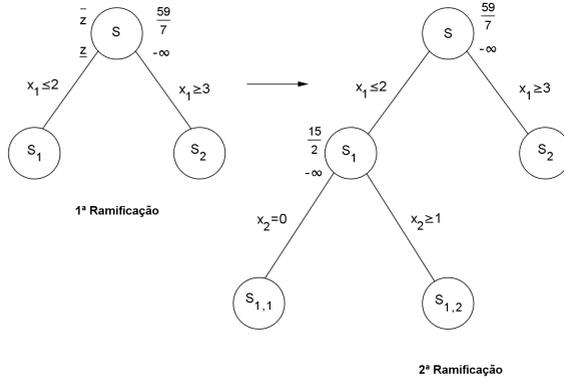


Figura 3 – Branch-and-Bound 1.

Após a subdivisão do problema em dois ramos, deve-se escolher um dos nós,  $S_1$  ou  $S_2$ , para realizar o mesmo procedimento feito ao ramo  $S$ . Arbitrariamente, escolheu-se o ramo  $S_1$ . O limite dual encontrado para o problema  $S_1$ , usando novamente o algoritmo simplex é  $\bar{z} = \frac{15}{2}$ , com a solução  $(x_1, x_2) = (2, \frac{1}{2})$ . A solução encontrada não é factível, sendo assim, ramificaremos o nó  $S_1$  conforme a segunda ramificação da Figura 3, um ramo com  $x_2 = 0$  e outro com  $x_2 \geq 1$ . Ramificado  $S_1$ , novamente deve ser feita a escolha de um dos nós para sua resolução.

Arbitrariamente será escolhido  $S_2$ . Durante sua resolução descobre-se que este problema é infactível, ou seja,  $\bar{z} = -\infty$ . Isso faz com que possamos eliminar o ramo  $S_2$ , nos restando  $S_{1,1}$  e  $S_{1,2}$ .

O nó  $S_{1,2}$  resulta em sua solução  $\bar{z}_{1,2} = 7$ , com  $x_{1,2} = (2, 1)$ . Note que  $x_{1,2} = (2, 1)$  é factível, então podemos considerar  $\underline{z}_{1,2} = 7$  e assim obter  $\underline{z}_{1,2} = \bar{z}_{1,2}$ . Logo  $x = (2, 1)$  é a solução ótima para  $S_{1,2}$ . Pode-se também considerar  $\underline{z}_{1,2}$  como  $\underline{z}$  de  $S$ , pois  $x = (2, 1)$  é factível, e ainda cortar o nó  $S_{1,2}$  por otimalidade, ou seja, não é mais necessário examiná-lo pois já está resolvido. Observe a Figura 4.

Nos resta agora apenas o nó  $S_{1,1}$ . A sua resolução resulta em  $\bar{z}_{1,1} = 6$ .

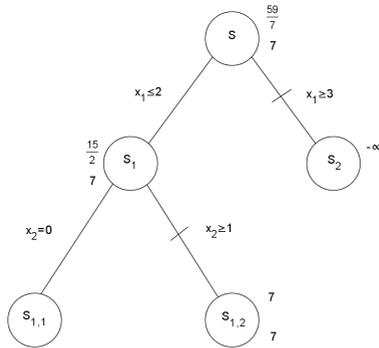


Figura 4 – Branch-and-Bound 2.

Como  $\underline{z} = \max \{ \underline{z}_k \} = 7$  e  $\bar{z}_{1,1} < \underline{z}$  o nó pode ser eliminado e desconsiderado, pois seu limite superior é menor que o maior limite inferior do problema. Veja a Figura 5.

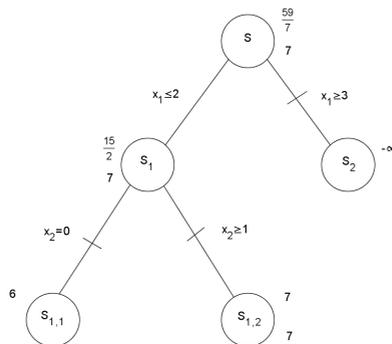


Figura 5 – Branch-and-Bound 3.

Como não nos resta mais nós a serem analisados, a solução  $x^* = (2, 1)$  com  $z = 7$  é ótima.

### 2.2.4 Planos de Corte

Considere o conjunto de soluções  $X$  de um problema clássico de otimização, o problema da mochila 0-1 (WOLSEY, 1998), que é dado por:

$$X = \{x \in \mathbb{B}^5 : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2\}. \quad (2.5)$$

onde  $\mathbb{B} = \{0, 1\}$ .

Para  $x_2 = x_4 = 0$ , pode-se observar que a desigualdade jamais será satisfeita, o que implica que uma solução factível para o problema deve satisfazer:

$$x_2 + x_4 \geq 1. \quad (2.6)$$

A desigualdade (2.6) é válida para a instância apresentada em (2.5), pois ela é verdadeira para todos os pontos factíveis contidos em  $X$ . Podemos então, propor uma formulação revisada para o problema (2.5):

$$X = \left\{ x \in \mathbb{B}^5 : \begin{array}{l} 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2, \\ x_2 + x_4 \geq 1 \end{array} \right\} \quad (2.7)$$

Esta formulação reduz o campo de busca pela solução ótima do problema da mochila, conforme ilustra a Figura 6. A inclusão da desigualdade (2.6) ao problema pode ocasionar ganhos computacionais durante sua resolução.

**Definição 6** *Uma desigualdade  $\pi^T x \leq \pi_0$  é uma desigualdade válida para um conjunto  $X \subseteq \mathbb{R}^n$  se, e somente se,  $\pi^T x \leq \pi_0$  para todo  $x \in X$ .*

Um poliedro possui infinitas desigualdades válidas. Cabe então saber quais delas serão úteis se adicionadas à formulação, e quais delas devem ser descartadas. O que é facilmente observável, é que se deseja desigualdades que diminuam o volume do poliedro que contenha as soluções factíveis, ou seja, as desigualdades que levam a uma formulação mais apertada. Tais desigualdades são chamadas de desigualdades fortes.

O algoritmo de planos de corte consiste em um processo iterativo que pode ser resumido em duas etapas. A primeira é a de resolver a relaxação linear do problema. Em seguida, deve-se inserir na formulação uma desigualdade válida que elimine a solução encontrada, caso esta não seja inteira. De forma intuitiva, esta desigualdade válida é chamada de corte. O algoritmo itera até que a solução da relaxação seja inteira, o que implica na obtenção do

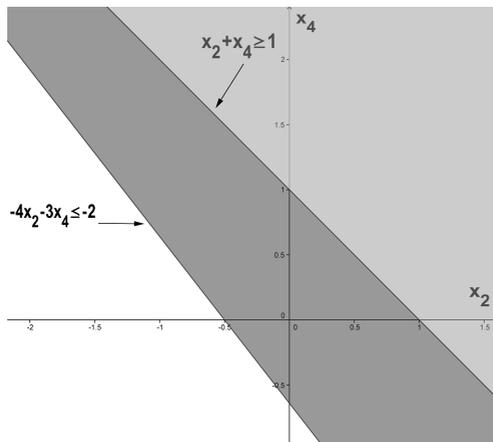


Figura 6 – Ilustração da Desigualdade Válida Para o Problema da Mochila, com  $x_1 = x_3 = x_5 = 0$ .

ótimo global. A obtenção dos cortes é fundamental para o algoritmo. Como esta é uma tarefa difícil, nem sempre será possível encontrar cortes até que seja resolvido o problema.

Existem vários procedimentos para se obter desigualdades válidas para problemas de programação inteira. O procedimento que será utilizado mais adiante no trabalho é baseado em um método conhecido como desigualdades de cobertura (*cover inequalities*) (WOLSEY, 1998).

### 2.2.5 Desigualdades de Coberturas e *Lifting* de Desigualdades Válidas

Considere os conjuntos  $X = \{x \in \mathbb{B}^n : \sum_{j=1}^n a_j x_j \leq b\}$  e  $N = \{1, \dots, n\}$ .

**Definição 7** Um conjunto  $C \subseteq N$  é uma cobertura se  $\sum_{j \in C} a_j > b$ . Uma cobertura é mínima se para todo  $j \in C$ ,  $C \setminus \{j\}$  não é uma cobertura.

Considerando o conjunto  $X = \{x \in \mathbb{B}^7 : 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 \leq 6\}$  e o conjunto  $N = \{1, \dots, 7\}$  do problema (2.1), apresentado anteriormente,  $C = \{4, 5, 6\}$  é uma cobertura, pois  $a_4 + a_5 + a_6 = 3 + 2 + 2 = 7 > 6$ .

**Proposição 2** Dado uma cobertura  $C \subseteq N$ , a desigualdade de cobertura

$$\sum_{j \in C} x_j \leq |C| - 1 \quad (2.8)$$

é uma desigualdade válida para  $X$ .

Temos então, para a cobertura  $C = \{4, 5, 6\}$  a desigualdade válida  $x_4 + x_5 + x_6 \leq 2$ .

Após determinado uma cobertura, podemos estender a desigualdade de cobertura para outras variáveis de decisão, tornando-a mais forte, ou seja, dado uma cobertura  $C \subseteq N = \{1, \dots, n\}$ , deseja-se encontrar valores para  $\alpha_j, j \in N \setminus C$ , de maneira que a desigualdade

$$\sum_{j \in C} x_j + \sum_{j \in N \setminus C} \alpha_j x_j \leq |C| - 1$$

ainda seja válida para  $X = \{x \in \mathbb{B}^n : \sum_{j=1}^n a_j x_j \leq b\}$ .

A seguir será apresentado um procedimento para calcular os fatores  $\alpha_j$ . Este procedimento é conhecido por *lifting*.

Então, seja  $\langle j_1, \dots, j_r \rangle$  uma ordenação de  $N \setminus C$ . Para calcular o maior valor possível de  $\alpha_{j_t}$  no qual  $\alpha_{j_i} x_{j_i} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$  seja válida, devemos resolver o problema

$$\begin{aligned} \zeta_t = \text{Maximize } & \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \\ \text{Sujeito a: } & \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \leq b - a_{j_t} \\ & x \in \mathbb{B}^{|C|+t-1} \end{aligned} \quad (2.9)$$

obtendo  $\alpha_{j_t} = |C| - 1 - \zeta_t$ .

Se a cobertura  $C$  for mínima, a desigualdade gerada utilizando este procedimento define uma faceta de  $\text{conv}(X)$ .

Sabemos que  $C = \{4, 5, 6\}$  é uma cobertura para a restrição da mochila (2.1). Após construir uma ordenação de  $N \setminus C = \langle j_1, j_2, j_3, j_4 \rangle = \langle 1, 2, 3, 7 \rangle$ ,  $r = 4$ , resolve-se o problema (2.9) para cada elemento de  $N \setminus C$  em sua ordenação. Obtem-se então, a seguinte desigualdade válida:

$$\begin{aligned} \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + x_4 + x_5 + x_6 + \alpha_7 x_7 &\leq 2 \\ 2x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 &\leq 2. \end{aligned}$$

## 2.3 SUMÁRIO

Neste capítulo vimos que a otimização é uma área que busca obter o comportamento ótimo de um processo. Vimos que ela é dividida conforme características de suas funções objetivo e restrições. A programação inteira é uma de suas divisões e alguns de seus conceitos serão utilizados durante o decorrer do trabalho. Vimos que um mesmo problema inteiro pode ter infinitas formulações, podendo algumas vezes compará-las definindo a melhor formulação. Algoritmos de resolução desta classe de problemas foram aqui ilustrados, o algoritmo *branch-and-bound* e o algoritmo de planos de corte. O algoritmo de planos de corte é fundamental para o trabalho e mais a frente será abordado novamente. Vale ressaltar mais uma vez que os conceitos aqui apresentados foram abordados da forma mais sucinta e breve possível. Para maiores detalhes, recomenda-se a leitura de (WOLSEY, 1998).

### 3 DEFINIÇÃO DO PROBLEMA

O problema de escalonamento de compressores (PEC) para operar poços de petróleo com o método de elevação por injeção contínua de gás, consiste na instalação ou ativação de compressores, e escalonamento dos mesmos para suprir a demanda de gás dos poços.

Os compressores que serão utilizados para suprir a demanda de um conjunto de poços  $M = \{1, \dots, m\}$  podem ser escolhidos dentre um conjunto  $N = \{1, \dots, n\}$  de compressores, conforme a Figura 7.

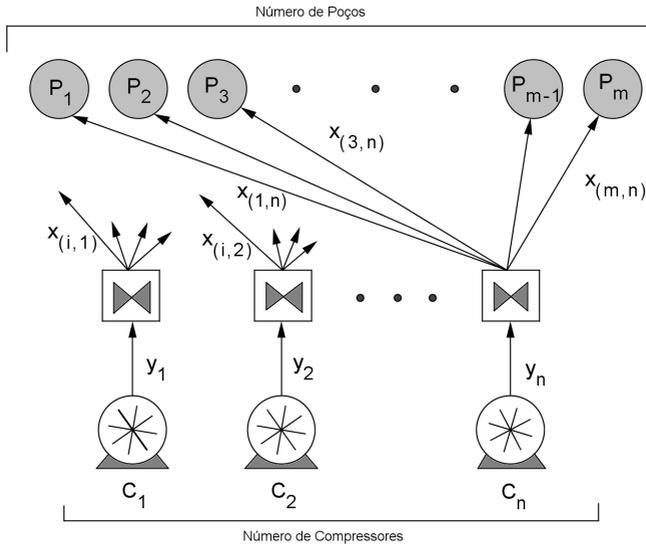


Figura 7 – Problema de Escalonamento de Compressores.

As variáveis  $y_j$ , com  $j \in N$ , e  $x_{i,j}$ , com  $(i,j) \in M \times N$ , são variáveis binárias. A variável  $y_j$  representará a instalação ou ativação de um compressor  $j$ . Sendo assim,  $y_j$  assumirá o valor 1 caso o compressor  $j$  esteja operando, e zero caso contrário. Já a variável  $x_{i,j}$  representará a linha que conecta o poço  $i$  com o compressor  $j$ , com isso,  $x_{i,j}$  assumirá o valor 1 caso o poço  $i$  seja operado pelo compressor  $j$ , e zero caso contrário.

Cada compressor  $j$  é caracterizado por uma curva de performance  $p_j^c(q_j^c)$ , que descreve sua pressão de descarga  $p_j^c$  em relação à taxa de saída de gás comprimido  $q_j^c$ .

Caso o compressor  $j$  seja ativado, este deve operar a uma taxa de saída

de gás  $q_j^c$  no intervalo  $R_j = [q_j^{c,\min}, q_j^{c,\max}]$ , que denotam respectivamente a taxa de saída de gás mínima e máxima de operação do compressor.

Cada poço  $i$  demanda uma taxa de injeção de gás  $q_i^w$ , que não poderá resultar em uma pressão de descarga menor que  $p_i^w$ . Esta pressão é a pressão mínima necessária para que o gás consiga entrar de forma contínua no espaço anular do poço e, conseqüentemente, na coluna de produção através das válvulas. A entrada do gás na coluna de produção diminui a densidade do fluido que ali está, tornando a pressão do reservatório maior que a pressão hidrostática da coluna de produção, fazendo com que o fluido seja levado à superfície. De forma ilustrativa a Figura 8 pode facilitar a compreensão dos conceitos dados acima.

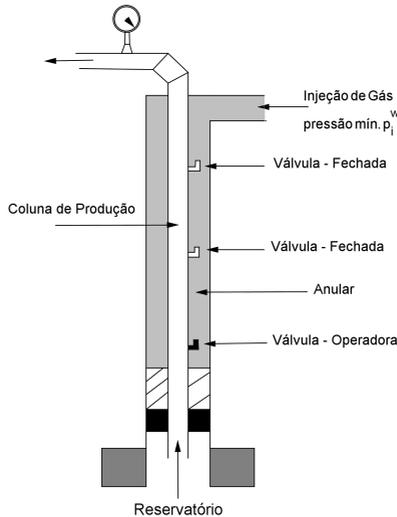


Figura 8 – Processo de Elevação por Injeção Contínua de Gás.

Considerando que há uma perda de pressão na linha que liga o compressor  $j$  e o poço  $i$ , dada por  $l_{i,j}$ , a taxa de saída de gás  $q_i^c$  deverá ser escolhida de forma que

$$p_j^c(q_j^c) \geq p_i^w + l_{i,j}. \quad (3.1)$$

Devido a esta restrição, considera-se que não serão todos os compressores pertencentes a  $N$  que poderão servir o poço  $i$ . Sendo assim, será denotado por  $N_i \subseteq N$  o conjunto de compressores que podem atender a este poço, assim como  $M_j \subseteq M$  denota o conjunto de poços que o compressor  $j$

pode servir.

**Suposição 1** Assume-se sobre a curva de performance  $p_j^c$  do compressor  $j$  que:

1.  $p_j^c(q_j^c)$  não aumenta conforme  $q_j^c$  aumenta de  $q_j^{c,\min}$  até  $q_j^{c,\max}$ .
2.  $q_i^w \leq q_j^{c,\max}$  e  $p_j^c(\max\{q_i^w, q_j^{c,\min}\}) \geq p_i^w + l_{i,j}, \forall i \in M_j$ .

**Definição 8**  $q_{i,j}^{c,\max} = \max\{q_j^c \in R_j : p_j^c(q_j^c) \geq p_i^w + l_{i,j}\}$  é a taxa de saída máxima do compressor  $j \in N_i$  em que a pressão de descarga é suficientemente alta para servir o poço  $i$ .

Note que a condição 2 da Suposição 1 implica que  $q_{i,j}^{c,\max} \geq q_i^w$  quando  $j \in N_i$ . E que para manter a região de operação  $R_j$  de um compressor  $j$  a mais apertada possível, pode-se considerar  $q_j^{c,\min} \geq \min\{q_i^w : i \in M_j\}$  e  $q_j^{c,\max} = \max\{q_{i,j}^{c,\max} : i \in M_j\}$ .

### 3.1 FORMULAÇÃO NÃO LINEAR REVISADA

Em programação matemática, o problema de escalonamento de compressores (PEC) pode ser formulado como:

$$P : \min f = \sum_{j \in N} c_j y_j + \sum_{i \in M} \sum_{j \in N_i} c_{i,j} x_{i,j} + \sum_{j \in N} h_j^c(q_j^c) \quad (3.2a)$$

sujeito a:

$$x_{i,j} \leq y_j, \quad i \in M, \quad j \in N_i \quad (3.2b)$$

$$\sum_{j \in N_i} x_{i,j} = 1, \quad i \in M \quad (3.2c)$$

$$q_j^c \leq q_{i,j}^{c,\max} x_{i,j} + q_j^{c,\max} (y_j - x_{i,j}), \quad j \in N, i \in M_j \quad (3.2d)$$

$$q_j^c \geq q_j^{c,\min} y_j, \quad j \in N \quad (3.2e)$$

$$\sum_{i \in M_j} q_i^w x_{i,j} \leq q_j^c, \quad j \in N \quad (3.2f)$$

$$y_j \leq 1, \quad j \in N \quad (3.2g)$$

$$x_{i,j} \geq 0, \quad i \in M, \quad j \in N_i \quad (3.2h)$$

$$y_j \in \mathbb{Z}, \quad j \in N \quad (3.2i)$$

$$x_{i,j} \in \mathbb{Z}, \quad i \in M, \quad j \in N_i \quad (3.2j)$$

onde  $c_j$  é o custo de instalação ou de ativação do compressor  $j$ ,  $c_{i,j}$  é o custo de instalação ou manutenção das linhas que servem o poço  $i$  do compressor  $j$ , e  $h_j$  é uma função não-linear do custo de operação do compressor  $j$ , com a seguintes variáveis a serem determinadas:

- $y_j$  é uma variável binária que assume valor 1 se o compressor  $j$  é instalado ou está operando.
- $x_{i,j}$  é uma variável binária que assume valor 1 se o poço  $i$  é operado pelo compressor  $j$ .
- $q_i$  é taxa de saída de gás do compressor  $j$ .

O problema de escalonamento de compressores  $P$  é um problema não-convexo, não-linear inteiro-misto de dificuldade considerável.

Apesar do PEC ser um problema de natureza estática, podemos lidar com falhas de equipamentos. As falhas de compressores e as manutenções nas linhas podem ser solucionadas através de uma nova otimização do processo, eliminando o equipamento com falha das variáveis de decisão.

Outra especificação que ainda pode ser inserida ao problema  $P$  é a decisão da utilização do gás excedente inserido nos poços. Uma vez que um poço  $i$  possui um demanda  $q_i^w$ , o excedente a esta demanda pode ser reutilizado, vendido ou até queimado. Caso o gás excedente seja reutilizado ou exportado, pode-se acrescentar um termo negativo na função objetivo (3.2a), referente ao lucro obtido. Este termo é negativo, pois tal solução diminuiria os custos. Porém o que é mais visto em plataformas de petróleo é a queima deste gás. Caso a plataforma opere com desperdício de gás, pode-se acrescentar um termo positivo na função objetivo (3.2a), referente ao custo do gás queimado.

**Proposição 3**  $P$  é NP-Difícil.

**Prova:** Para provar que um problema é NP-Difícil, basta mostrar que este problema é tão difícil de se resolver quanto qualquer outro problema NP.

O problema de alocação de instalações com capacidade ilimitada, denominado pela sigla em inglês UFLP, é conhecido por ser um problema NP-Difícil (CORNUEJOLS, 1983). Então, seja  $I_{\text{uff}}$  uma instância do UFLP dado por um conjunto  $N = \{1, \dots, n\}$  de instalações, um conjunto  $M = \{1, \dots, m\}$  de clientes, um custo  $c_j$  para construção de uma instalação  $j$  e o custo  $c_{i,j}$  para suprir a demanda do cliente  $i$  através da instalação  $j$ . Esta instância do UFLP será reduzida a uma instância  $I_{\text{cs}}$  do problema de escalonamento de compressores, onde a mesma é caracterizada por  $N_i = N$  para todo  $i \in M$ ,  $M_j = M$  para todo  $j \in N$ ,  $q_j^{c,\max} = m$  e  $q_j^{c,\min} = 0$  para todo  $j \in N$ ,  $q_{i,j}^{c,\max} = m$  para todo  $(i, j) \in M \times N$ ,  $q_i^w = 1$  para todo  $i \in M$  e  $h_j^c = 0$  para todo  $j \in N$ .

Claramente é possível verificar que uma solução ótima de  $I_{\text{uff}}$  é dada por uma solução ótima de  $I_{\text{cs}}$ , logo  $P$  é NP-Difícil.  $\square$

A formulação do problema de escalamento de compressores apresentado acima difere da formulação precedente, que será denominada como formulação antiga, abordada em (CAMPONOGARA et al., 2011; CAMPONOGARA; CASTRO, 2008). Esta formulação pode ser observada a seguir:

$$P_{\text{old}} : \min f = \sum_{j \in N} c_j y_j + \sum_{i \in M} \sum_{j \in N_i} c_{i,j} x_{i,j} + \sum_{j \in N} h_j^c(q_j^c) \quad (3.3a)$$

sujeito a:

$$x_{i,j} \leq y_j, \quad i \in M, \quad j \in N_i \quad (3.3b)$$

$$\sum_{j \in N_i} x_{i,j} = 1, \quad i \in M \quad (3.3c)$$

$$p_j^c(q_j^c) \geq (p_i^w + l_{i,j})x_{i,j}, \quad i \in M, \quad j \in N_i \quad (3.3d)$$

$$q_j^{c,\min} y_j \leq q_j^c \leq q_j^{c,\max} y_j, \quad j \in N \quad (3.3e)$$

$$\sum_{i \in M_j} q_i^w x_{i,j} \leq q_j^c, \quad j \in N \quad (3.3f)$$

$$y_j \leq 1, \quad j \in N \quad (3.3g)$$

$$x_{i,j} \geq 0, \quad i \in M, \quad j \in N_i \quad (3.3h)$$

$$y_j \in \mathbb{Z}, \quad j \in N \quad (3.3i)$$

$$x_{i,j} \in \mathbb{Z}, \quad i \in M, \quad j \in N_i \quad (3.3j)$$

Note que, na formulação antiga, a restrição de pressão é dada pela inequação não linear  $p_j^c(q_j^c) \geq (p_i^w + l_{i,j})x_{i,j}$ . Sendo assim, para tratar este modelo é preciso assumir duas condições: (i)  $p_j^c$  é côncava, assegurando a convexidade do conjunto factível da desigualdade não linear; e (ii)  $p_j^c$  é não-negativa quando o compressor não esteja operando ( $q_j^c = 0$ ). O modelo revisado apresentado nas equações (3.2a) até (3.2j) não exige tais condições. Em seu lugar, assumimos apenas que  $p_j^c$  é não-crescente. A grande diferença está na representação da restrição de pressão através da vazão  $q_j^c$  e não mais através da curva de pressão  $p_j^c(q_j^c)$ , que é função não-linear e que trás complexidade ao modelo. Na formulação revisada, a vazão  $q_j^c$  deve ser determinada no intervalo  $[q_j^{c,\min}, q_j^{c,\max}]$ , pois neste intervalo a restrição de pressão é satisfeita através da definição de  $q_{i,j}^{c,\max}$ . A visão geométrica de tal mudança pode ser observada na Figura 9.

Para entender melhor a força da formulação revisada, seja  $\mathbf{y} = (y_j : j \in N)$ ,  $\mathbf{q} = (q_j^c : j \in N)$ ,  $\mathbf{x} = (x_{i,j} : i \in M, j \in N_i)$  e  $\mathbf{u} = (\mathbf{y}, \mathbf{q}, \mathbf{x})$  o vetor de variáveis de decisão. O poliedro definido pela formulação revisada

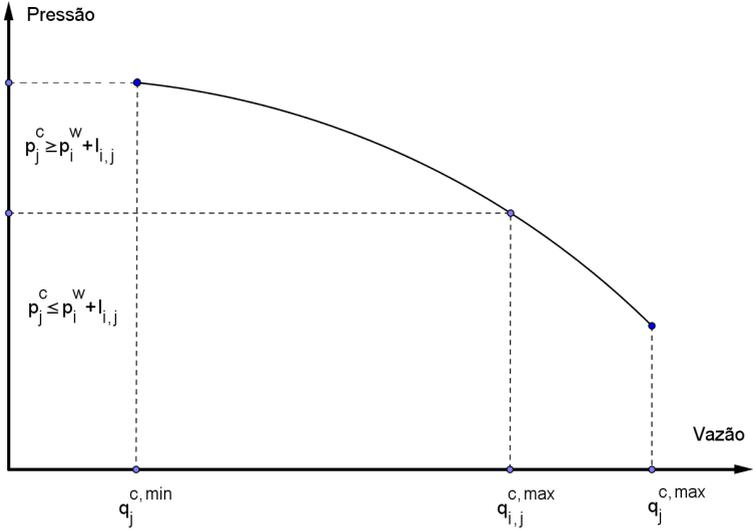


Figura 9 – Visão Geométrica da Restrição de Pressão.

é dado por  $\mathcal{P} = \{\mathbf{u} \in \mathbb{R}^{2n+s(\mathbf{x})} : \mathbf{u} \text{ satisfaz (3.2b)–(3.2h)}\}$ , onde  $s(\mathbf{x})$  representa o número de entradas do vetor  $\mathbf{x}$ . Isto significa que a região factível do problema é dada por  $\mathcal{X} = \mathcal{P} \cap (\mathbb{Z}^n \times \mathbb{R}^n \times \mathbb{Z}^{s(\mathbf{x})})$ . Agora, seja  $\tilde{\mathcal{P}} = \{\mathbf{u} \in \mathbb{R}^{2n+s(\mathbf{x})} : \mathbf{u} \text{ satisfaz (3.3b)–(3.3h)}\}$  o poliedro definido pela formulação antiga.

**Proposição 4** Se  $p_j^c(q_j^c)$  é côncava e não crescente para a região estendida  $q_j^c \in [0, q_j^{c, \max}]$  e todo  $j \in N$ , então  $\mathcal{P} \subseteq \tilde{\mathcal{P}}$ .

**Prova:** Seja qualquer  $\mathbf{u} \in \mathcal{P}$ . Para provar que  $\mathcal{P} \subseteq \tilde{\mathcal{P}}$  devemos mostrar que  $\mathbf{u} \in \tilde{\mathcal{P}}$ . Pode-se observar facilmente que para qualquer  $j \in N$  e  $i \in M_j$ , a inequação (3.2d) implica que  $q_j^c \leq q_{i,j}^{c, \max} x_{i,j} + q_j^{c, \max} (y_j - x_{i,j}) \leq q_j^{c, \max} x_{i,j} + q_j^{c, \max} (y_j - x_{i,j}) = q_j^{c, \max} y_j$ . Da mesma forma, a desigualdade (3.2d) implica que  $q_j^c \leq q_{i,j}^{c, \max} x_{i,j} + q_j^{c, \max} (y_j - x_{i,j}) \leq q_{i,j}^{c, \max} x_{i,j} + q_j^{c, \max} (1 - x_{i,j})$ . Sendo assim,  $p_j^c(q_j^c) \geq p_j^c(q_{i,j}^{c, \max} x_{i,j} + q_j^{c, \max} (1 - x_{i,j})) \geq x_{i,j} p_j^c(q_{i,j}^{c, \max}) + (1 - x_{i,j}) p_j^c(q_j^{c, \max}) \geq x_{i,j} p_j^c(q_{i,j}^{c, \max}) \geq (p_i^w + l_{i,j}) x_{i,j}$ , onde a primeira desigualdade pode ser explicada pelo fato de que  $p_j^c$  é não

decrecente. Já a segunda e quarta desigualdades seguem respectivamente da hipótese de que  $p_j^c$  é côncava e da definição de  $q_{i,j}^{c,\max}$ . Consequentemente,  $\mathbf{u}$  atende as restrições  $q_j^c \leq q_j^{c,\max} y_j$  e  $p_j^c(q_j^c) \geq (p_i^w + l_{i,j})x_{i,j}$ . Logo,  $\mathbf{u} \in \tilde{\mathcal{P}}$ .  $\square$

Em outras palavras, a Proposição 4 implica que o poliedro resultante pela formulação revisada é o poliedro mais próximo do fecho-convexo dentre as duas formulações. Ou seja, a região de busca por um ponto ótimo na formulação revisada é menor ou igual a região de busca da formulação antiga.

### 3.2 INSTÂNCIA EXEMPLO

Para eventuais ilustrações que serão realizadas ao problema PEC, será utilizada uma instância exemplo retirada de (CAMPONOGARA et al., 2011). Esta instância consiste de  $n = 5$  compressores e  $m = 6$  poços, com as respectivas propriedades apresentadas nas Tabelas 2, 3 e 4.

Após alguns estudos sobre funções de pressão  $p_j^c(q_j^c)$ , pode-se observar que estas seguem um determinado padrão, conforme mostra a Figura 10. Por este motivo a função de pressão será definida como  $p_j^c(q_j^c) = \alpha_{0,j} + \alpha_{1,j}q_j^c + \alpha_{2,j}(q_j^c)^2 + \alpha_{3,j}(q_j^c)^3 + \alpha_{4,j}\ln(1 + q_j^c)$ , já que esta função provou ser um bom ajuste para os dados dos compressores. A curva de pressão  $p_j^c(q_j^c)$  induzida pelos parâmetros da Tabela 3 é côncava no domínio de  $q_j^c$  como requer a formulação antiga.

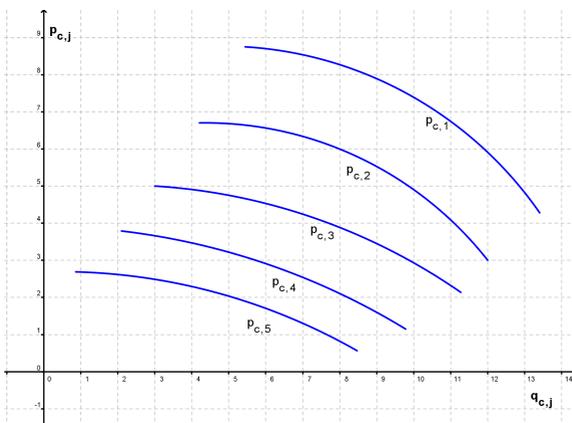


Figura 10 – Curva de Pressão de Cinco Compressores Típicos.

Na Tabela 5 apresenta-se a perda de pressão e custo de manutenção das linhas.

A função custo de operação para o compressor  $j$  será definida como  $h_j^c(q_j^c) = d_j q_j^c p_j^c(q_j^c)$  com  $d_j$  constante e positivo. O custo operacional dado pelo produto do fluxo de gás comprimido pela pressão, expressa a energia consumida por um compressor, sendo análogo ao produto da corrente elétrica pela voltagem.

Tabela 2 – Dados do Compressor

$j$	$c_j$	$d_j$	$q_j^{c,\min}$	$q_j^{c,\max}$
1	8	3	5,3	13,0
2	6	5	4,2	12,0
3	10	1	3,8	10,8
4	4	7	2,0	9,8
5	6	4	1,0	8,5

Tabela 3 – Parâmetros da Curva de Pressão

$j$	$\alpha_{0,j}$	$\alpha_{1,j}$	$\alpha_{2,j}$	$\alpha_{3,j}$	$\alpha_{4,j}$
1	8,36900	-0,502034	0,0651919	-0,00497000	0,814770
2	7,11148	-0,602903	0,0680301	-0,00507334	0,766603
3	5,30343	-0,309098	0,0401807	-0,00427144	0,324494
4	3,49652	-1,215010	0,1236120	-0,00741996	1,978650
5	2,45413	-0,356522	0,0361532	-0,00363706	0,347968

Tabela 4 – Dados dos Poços

$i$	$q_i^w$	$p_i^w$	$N_i$
1	3	5	{1, 2}
2	4	5	{1, 2}
3	2	4	{1, 2, 3}
4	4	3	{1, 2, 3, 4}
5	5	1,5	{1, 2, 3, 4, 5}
6	2	1	{1, 2, 3, 4, 5}

Tabela 5 – Dados das Linhas

$i \setminus j$	$l_{i,j}/c_{i,j}$				
	1	2	3	4	5
1	0,1/8	0,4 /7			
2	0,2/9	0,5 /5			
3	0,2/6	0,2 /4	0,05/6		
4	0,1/10	0,12/8	0,15/7	0,2/1	
5	0,1/6	0,1 /4	0,2 /3	0,1/1	0,1/1
6	0,1/7	0,1 /3	0,1 /2	0,1/1	0,1/1

Tabela 6 – Taxa Máxima de Saída dos Compressores para o Atendimento dos Poços

$i \setminus j$	$q_{i,j}^{c,\max}$				
	1	2	3	4	5
1	12,1291	8,73323			
2	12,0331	8,52604			
3	12,9134	10,6014	7,71841		
4	13,0	11,7701	9,41583	4,64912	
5	13,0	12,0	10,8	9,19731	5,63834
6	13,0	12,0	10,8	9,8	7,29901

### 3.3 SUMÁRIO

A nova forma de se abordar a restrição de pressão,  $p_i^c(q_i^c) \geq p_i^w + l_{i,j}$ , no problema de escalonamento de compressores nos proporcionou uma nova formulação para o problema. A nova formulação exige menos suposições sobre a curva  $p_i^c(q_i^c)$  e, como provado no capítulo, é superior a formulação antiga no sentido de ser mais próxima do fecho convexo do problema.

## 4 MODELOS E ALGORITMOS PARA LINEARIZAÇÃO POR PARTES

O processo de linearização por partes baseia-se na aproximação de uma função não-linear por uma função seccionalmente linear. Este processo é muito utilizado em problemas de otimização que contêm funções não lineares em seu modelo. O PEC, como comentado anteriormente, é um problema não-linear, não convexo inteiro misto. Os algoritmos desenvolvidos até a atualidade para este tipo de problema, por mais desenvolvidos que estejam, podem resultar em ótimos locais, que por sua vez podem estar muito aquém de um ótimo global do problema.

Na literatura, a forma mais utilizada para contornar este tipo de problema é a linearização por partes da função não-linear. A linearização desta função torna possível transformar o problema atual em um problema linear inteiro misto (MILP), que pode ser resolvido utilizando algoritmos MILP eficientes.

Embora muitos trabalhos utilizem este processo (GUNNERUD; FOSS, 2010; CAMPONOVARA; CONTO, 2009; REJOWSKI; PINTO, 2002; LEE et al., 1996; CAMPONOVARA et al., 2011), pouca atenção é dada ao problema de se encontrar uma linearização ótima para a função não linear. O número de segmentos de retas para a aproximação é de suma importância. Um número muito alto de seções pode ser desnecessário e tornar o modelo muito complexo, já um número muito pequeno pode não retratar as características principais da função.

Algumas técnicas para gerar o modelo linear por partes são baseados em agrupamentos de pontos, como K-means (FERRARI-TRECCATE et al., 2003), agrupamentos Fuzzy (GHOSH; MAKI, 2010) e uma técnica de otimização para agrupamentos que utiliza algoritmos genéticos (GHOSH et al., 2011).

A seguir serão propostos algoritmos baseados em programação dinâmica, em que a aproximação linear ótima é obtida através da minimização de erros quadráticos. Os modelos proporcionam ao operador optar por uma linearização com um número fixo ou máximo de segmentos, exigir ou não que a linearização seja feita de forma a obter um envelope superior ou inferior, e ainda garantir a continuidade da linearização.

### 4.1 INTRODUÇÃO AO PROBLEMA DE LINEARIZAÇÃO POR PARTES

Considere uma função não-linear qualquer  $\Phi : D \rightarrow \mathbb{R}$ , com  $D \subseteq \mathbb{R}$  e um conjunto de pontos  $P = \{p_1, \dots, p_n\}$ , tal que  $p_i = (x_i, y_i)$ ,  $y_i = \Phi(x_i)$  e  $x_1 < x_2 < \dots < x_n$ . Deseja-se encontrar um conjunto de segmentos de

reta  $R = \{r_1, \dots, r_T\}$  que aproxime a função geradora de  $P$ , ou seja,  $\Phi$ .

Assumindo que os pontos de quebra dos segmentos serão os próprios pontos de  $P$ , uma aproximação pode ser definida em função de uma sequência de índices  $I = \langle i(1), i(2), \dots, i(T+1) \rangle$  tal que:

- $i(1) = 1 < i(2) < \dots < i(T+1) = n$ ; e
- cada subconjunto de pontos  $P_t = \{x_{i(t)}, x_{i(t)+1}, \dots, x_{i(t+1)}\} \subseteq P$ , é aproximado pelo segmento de reta  $r_t = \{(x, y) : y = a_t x + b_t, x_{i(t)} \leq x \leq x_{i(t+1)}\}$  para  $t = 1, \dots, T$ .

Uma aproximação de  $P_t$  pode ser definida pela minimização do erro quadrático entre a reta  $r_t$  e os pontos de  $P_t$ :

$$\min_{a_t, b_t} \sum_{k=i(t)}^{i(t+1)} (a_t x_k + b_t - y_k)^2 \quad (4.1)$$

sendo  $a_t$  e  $b_t$  os parâmetros que definem a aproximação linear.

O problema de mínimos quadrados acima é facilmente resolvido calculando os parâmetros  $a_t$  e  $b_t$  que anulam a derivada da função objetivo. Tem-se que:

$$a_t = \frac{|P_t| \sum_{k=i(t)}^{i(t+1)} x_k y_k - (\sum_{k=i(t)}^{i(t+1)} x_k) (\sum_{k=i(t)}^{i(t+1)} y_k)}{|P_t| \sum_{k=i(t)}^{i(t+1)} x_k^2 - (\sum_{k=i(t)}^{i(t+1)} x_k)^2} \quad (4.2a)$$

$$b_t = \frac{\sum_{k=i(t)}^{i(t+1)} y_k - a_t \sum_{k=i(t)}^{i(t+1)} x_k}{|P_t|}. \quad (4.2b)$$

## 4.2 MODELO COM NÚMERO FIXO DE SEGMENTOS

O primeiro modelo proposto para linearização por partes é o modelo com um número fixo de segmentos  $T$ , onde os segmentos serão determinados de forma a minimizar o erro quadrático total da aproximação linear por partes. Formalmente, o problema é dado por:

$$\min_I E = \sum_{t=1}^T E_t = \sum_{t=1}^T \sum_{k=i(t)}^{i(t+1)} (y_k - a_t x_k - b_t)^2 \quad (4.3)$$

tendo como variáveis de decisão a sequência  $I = \langle i(1), i(2), \dots, i(T+1) \rangle$  e os parâmetros  $a_t$  e  $b_t$  para  $t = 1, \dots, T$ .

Baseado em uma proposta similar de (KLEINBERG; TARDOS, 2005), este problema pode ser colocado em uma forma recursiva que leva a um algoritmo de programação dinâmica (BERTSEKAS, 2000). Para tanto, seja  $F(j, t)$  o custo mínimo de se aproximar os pontos  $\{p_1, p_2, \dots, p_j\}$  com exatamente  $t$  segmentos de reta. Então,

$$F(j, t) = \min_{t \leq i \leq j} F(i, t-1) + E(i, j) \quad (4.4)$$

onde  $E(i, j)$  é o erro de se aproximar os pontos  $\{p_i, p_{i+1}, \dots, p_j\}$  com um segmento obtido via mínimos quadrados (4.1). Note que  $E(i, i) = \infty$ ,  $F(j, t) = \infty$  para  $j \leq t$  e  $F(j, 1) = E(1, j)$  para todo  $j$ . Pela estrutura da equação recursiva (4.4), é possível observar que  $F(n, T) = \min E$ .

**Teorema 1** *A solução recursiva dada pela equação (4.4) leva à aproximação linear por partes ótima de um conjunto de pontos  $P$  no plano com exatamente  $T$  segmentos de reta, tendo como função objetivo o desvio quadrático do segmento de reta.*

**Prova:** Para a demonstração será utilizado o método de indução matemática em  $j$ .

- (Caso base): Para o caso base  $j = 2$  obtemos dois casos,  $t = 1$  ou  $t > 1$ . Caso  $t = 1$ , temos  $F(j, t) = 0$  que é a solução ótima, pois devemos aproximar os dois pontos pela reta que os intersectam. Caso contrário  $F(j, t) = \infty$  que também resulta na solução ótima, por definição.
- (Hipótese de Indução): Para o passo de indução, considere  $j > 2$ , e  $t$  qualquer. H.I. =  $F(j, t)$  leva à aproximação linear por partes ótima de um conjunto de pontos  $P_j$  no plano com exatamente  $t$  segmentos, tendo como função objetivo o desvio quadrático do segmento de reta.
- (Caso  $j + 1$ ):  $F((j+1), t) = \min_{t \leq i \leq (j+1)} F(i, t-1) + E(i, (j+1))$ . O ponto  $p_i$  deve ser tomado como aquele que minimiza o custo agregado, ou seja, custo de aproximar  $\{p_1, \dots, p_i\}$  com  $t-1$  segmentos, dado por  $F(i, t-1)$ , e o custo de aproximar  $\{p_i, \dots, p_{(j+1)}\}$  com apenas um segmento, dado por  $E(i, (j+1))$ . Uma vez que  $F(i, t-1)$  é, por indução, o custo mínimo de aproximar  $\{p_1, \dots, p_i\}$  com  $t-1$  segmentos, a solução recursiva produz a solução ótima para  $F((j+1), t)$ .  $\square$

A sequência ótima de índices  $\langle i(1), i(2), \dots, i(T+1) \rangle$  é obtida recursivamente, onde,  $i(T+1) = n$ ,  $i(T)$  é o índice  $i$  que minimiza a função

$F(i(T+1), T)$  na equação recursiva (4.4);  $i(T-1)$  é o índice  $i$  que minimiza  $F(i(T), T-1)$  na recursão, e assim sucessivamente até que  $F(i(2), 1)$  seja alcançado, quando se define  $i(1) = 1$ .

O pseudo-código do algoritmo de programação dinâmica DPPL-Fixo para resolução recursiva de  $F(j, t)$  é apresentado em Algoritmo 1. O algoritmo registra na tabela  $I(j, t)$  os índices  $i$  que minimizam a equação recursiva (4.4), índices estes que representam os pontos de quebra ótimos da linearização por partes.

---

**Algorithm 1:** DPPL-Fixo( $P, T$ )

---

```

1 for  $i = 1$  to  $n$  do
2    $E(i, i) := \infty$ ;
3   for  $j = i + 1$  to  $n$  do
4      $a_{i,j} := \frac{(j-i+1) \sum_{k=i}^{k=j} x_k y_k - (\sum_{k=i}^{k=j} x_k)(\sum_{k=i}^{k=j} y_k)}{(j-i+1) \sum_{k=i}^{k=j} x_k^2 - (\sum_{k=i}^{k=j} x_k)^2}$ ;
5      $b_{i,j} := \frac{\sum_{k=i}^{k=j} y_k - a_j \sum_{k=i}^{k=j} x_k}{(j-i+1)}$ ;
6      $A_{i,j} := a_{i,j}$ ;  $B_{i,j} := b_{i,j}$ ;
7      $E(i, j) := \sum_{k=i}^j (y_k - a_{i,j} x_k - b_{i,j})^2$ ;
8 for  $j = 1$  to  $n$  do
9   for  $t = j$  to  $T$  do
10     $F(j, t) := \infty$ ;  $I(j, t) := 0$ ;
11    $F(j, 1) := E(1, j)$ ;  $I(j, 1) := 1$ ;
12   for  $t = 2$  to  $\min\{j-1, T\}$  do
13      $F(j, t) := \infty$ ;  $I(j, t) := 0$ ;
14     for  $i = t$  to  $j$  do
15       if  $F(j, t) > F(i, t-1) + E(i, j)$  then
16          $F(j, t) := F(i, t-1) + E(i, j)$ ;
17          $I(j, t) := i$ ;

```

---

O algoritmo de programação dinâmica DPPL-Fixo segue a estrutura recursiva da equação (4.4), portanto, é correto segundo o Teorema 1. Uma questão relevante refere-se ao desempenho do algoritmo. O loop-for das linhas 1-7 executa em tempo  $\mathcal{O}(n^3)$ , enquanto o loop-for nas linhas 8-17 executa em tempo  $\mathcal{O}(n^2T)$ . Um vez que  $T < n$ , conclui-se que o algoritmo tem tempo de execução  $\mathcal{O}(n^3)$ .

Para eventuais ilustrações deste, e dos demais algoritmos, a Tabela 7 descreve um conjunto de pontos  $P$  a serem aproximados por funções lineares por partes. Os algoritmos deste capítulo foram implementados no software

Tabela 7 – Conjunto de Pontos para Linearização por Partes.

$i$	$x_i$	$y_i$
1	0,011420	0,185752
2	0,086119	0,450287
3	0,172113	0,558454
4	0,292823	0,585790
5	0,313050	0,702052
6	0,321203	0,833851
7	0,489927	0,871395
8	0,880797	0,910477

Tabela 8 – Valores de  $F(j, t)$  e  $I(j, t)$  para a Instância Exemplo.

$j \backslash t$	$F(j, t) : I(j, t)$		
	1	2	3
1	$\infty : 1$	$\infty : 0$	$\infty : 0$
2	<b>0 : 1</b>	$\infty : 0$	$\infty : 0$
3	5,5459e-03 : 1	0 : 2	$\infty : 0$
4	2,2611e-02 : 1	1,7718e-03 : 2	0 : 3
5	2,3016e-02 : 1	<b>5,5550e-03 : 2</b>	1,7718e-03 : 4
6	3,5038e-02 : 1	2,3016e-02 : 5	4,0758e-03 : 4
6	4,4083e-02 : 1	2,4617e-02 : 2	1,3596e-02 : 5
8	1,2685e-01 : 1	3,4109e-02 : 5	<b>1,6648e-02 : 5</b>

Matlab versão 2007.

A aplicação do algoritmo DPPL-Fixo à instância exemplo com  $T = 3$  segmentos produz os valores  $F(i, j)$  e  $I(i, j)$  descritos na Tabela 8. A sequência ótima é obtida recursivamente da tabela como segue:  $i(4) = 8$ ;  $i(3) = I(8, 3) = 5$ ;  $i(2) = I(5, 2) = 2$ ; e  $i(1) = I(2, 1) = 1$ , conforme indicado em negrito. Logo, a sequência de índices ótima é  $\langle 1, 2, 5, 8 \rangle$ , indicando que o primeiro segmento de reta aproxima os pontos  $\{p_1, p_2\}$ , o segundo segmento aproxima os pontos  $\{p_2, \dots, p_5\}$  e o terceiro segmento aproxima os pontos  $\{p_5, \dots, p_8\}$ . O custo total da aproximação é  $F(8, 3) = 1,6648 \times 10^{-2}$ . Esta aproximação linear por partes é ilustrada na Figura 11.

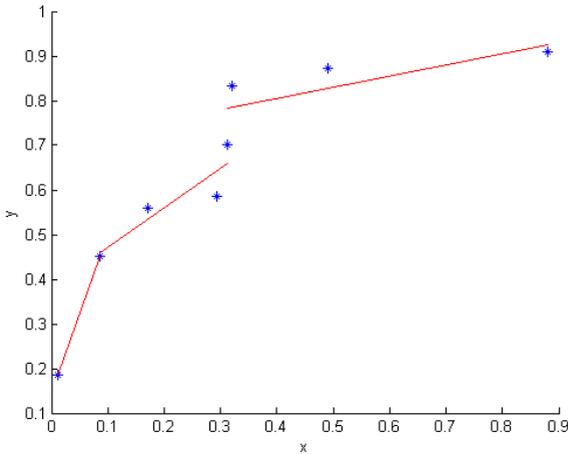


Figura 11 – Aproximação Linear por Partes para a Instância Exemplo com 3 Segmentos.

### 4.3 MODELO COM NÚMERO MÁXIMO DE SEGMENTOS

Uma variação do modelo anterior surge quando se especifica um número máximo de segmentos de reta, cada um dos quais incorre um custo  $c$ , que penalizará a construção da linearização cada vez que o conjunto  $P$  seja particionado para incluir um segmento na mesma. O problema pode também ser expresso em uma forma recursiva. Nesta versão do problema,  $F(j, t)$  passa a ser o custo mínimo de se aproximar os pontos  $\{p_1, p_2, \dots, p_j\}$  com no máximo  $t$  segmentos de retas, que consiste do erro de aproximação e do custo de cada segmento de reta. Mais precisamente,

$$F(j, t) = \min_{1 \leq i \leq j} F(i, t - 1) + E(i, j) + c \quad (4.5)$$

onde  $E(i, j)$  é definido como anteriormente e as condições iniciais são  $F(1, t) = \infty$  e  $F(2, t) = E(1, 2) + c$  para todo  $t$ .

**Teorema 2** *A solução recursiva dada pela equação (4.5) produz a aproximação linear por partes ótima de um conjunto de pontos  $P$  no plano com no máximo  $T$  segmentos de reta, tendo como função objetivo o desvio quadrático do segmento de reta combinado com o custo  $c$  por segmento adicionado à li-*

nearização.

A prova deste teorema será omitida por ser análoga à prova do Teorema 1.

O número de segmentos e a sequência de índices ótima é obtida através da função recursiva  $F(j, T)$ . Para determinar o número ótimo de segmentos,  $T^*$ , segue-se a solução recursiva:  $i(T+1) = n$ ;  $i(T)$  é o índice que minimiza  $F(n, T)$  na equação recursiva (4.5);  $i(T-1)$  é o índice que minimiza  $F(i(T), T-1)$  na recursão; e assim sucessivamente até se chegar ao primeiro índice  $i(T+1-k) = 1$ . Portanto,  $T^* = k$  é o número ótimo de segmentos e  $\langle i(T-k+1), i(T-k+2), \dots, i(T+1) \rangle$  é a sequência ótima de pontos de quebra.

O pseudo-código do algoritmo de programação dinâmica DPPL-Max para resolução recursiva (4.5) é apresentado em Algoritmo 2. O algoritmo, da mesma forma que DPPL-Fixo, registra os índices  $i$  que minimizam a equação recursiva (4.5) na tabela  $I(j, t)$ .

---

**Algorithm 2:** DPPL-Max( $P, T$ )

---

```

1 for  $i = 1$  to  $n$  do
2    $E(i, i) := \infty$ 
3   for  $j = i + 1$  to  $n$  do
4      $a_{i,j} := \frac{(j-i+1) \sum_{k=i}^{k=j} x_k y_k - (\sum_{k=i}^{k=j} x_k)(\sum_{k=i}^{k=j} y_k)}{(j-i+1) \sum_{k=i}^{k=j} x_k^2 - (\sum_{k=i}^{k=j} x_k)^2}$ 
5      $b_{i,j} := \frac{\sum_{k=i}^{k=j} y_k - a_{i,j} \sum_{k=i}^{k=j} x_k}{(j-i+1)}$ 
6      $E(i, j) := \sum_{k=i}^j (y_k - a_{i,j} x_k - b_{i,j})^2$ 
7 for  $t = 1$  to  $T$  do
8    $F(1, t) := \infty, I(1, t) := 0$ 
9    $F(2, t) := E(1, 2) + c, I(2, t) := 1$ 
10 for  $j = 3$  to  $n$  do
11    $F(j, 1) := E(1, j) + c, I(j, 1) := 1$ 
12   for  $t = 2$  to  $T$  do
13      $F(j, t) := E(1, j) + c, I(j, t) := 1$ 
14     for  $i = 2$  to  $j$  do
15       if  $F(j, t) > F(i, t-1) + E(i, j) + c$  then
16          $F(j, t) := F(i, t-1) + E(i, j) + c$ 
17          $I(j, t) := i$ 

```

---

Tabela 9 – Valores de  $F(j, t)$  e  $I(j, t)$  para a Instância Exemplo.

$j \setminus t$	$F(j, t) : I(j, t)$				
	1	2	3	4	5
1	$\infty : 0$				
2	0,010000 : 1	<b>0,010000 : 1</b>	0,010000 : 1	0,010000 : 1	0,010000 : 1
3	0,015546 : 1	0,015546 : 1	0,015546 : 1	0,015546 : 1	0,015546 : 1
4	0,032611 : 1	0,021772 : 2	<b>0,021772 : 2</b>	0,021772 : 2	0,021772 : 2
5	0,033016 : 1	0,025555 : 2	0,025555 : 2	0,025555 : 2	0,025555 : 2
6	0,045038 : 1	0,043016 : 5	0,034076 : 4	<b>0,034076 : 4</b>	0,034076 : 4
7	0,054083 : 1	0,044617 : 2	0,043596 : 5	0,043596 : 5	0,043596 : 5
8	0,136850 : 1	0,054109 : 5	0,046648 : 5	0,044208 : 6	<b>0,044208 : 6</b>

Seguindo a mesma estrutura de análise de execução do algoritmo DPPL-Fixo, pode-se concluir que o algoritmo de programação dinâmica DPPL-Máximo executa em tempo  $\mathcal{O}(n^3)$ .

Ao aplicar o algoritmo DPPL-Max sobre a instância exemplo com um limite de  $T = 5$  segmentos de retas e custo  $c = 0.01$  por segmento de reta, o algoritmo de programação dinâmica gera os valores  $F(j, t)$  e índices  $I(j, t)$  apresentados na Tabela 9. A partir dos valores  $I(j, t)$ , verifica-se que  $I(8, 5) = 6$ ,  $I(6, 4) = 4$ ,  $I(4, 3) = 2$ ,  $I(2, 2) = 1$  e portanto  $T^* = 4$ .

A aproximação linear por partes ótima utiliza 4 segmentos dos 5 segmentos permitidos em função do custo por segmento. O custo total da aproximação é  $F(8, 5) = 0,044208$ .

Uma vez que a sequência de índices ótima é  $\langle i(1), \dots, i(5) \rangle = \langle 1, 2, 4, 6, 8 \rangle$ , o primeiro segmento de reta aproxima os pontos  $\{p_1, p_2\}$ , o segundo aproxima  $\{p_2, p_3, p_4\}$ , o terceiro  $\{p_4, p_5, p_6\}$  e, por fim, o quarto segmento aproxima os pontos  $\{p_6, p_7, p_8\}$ . A aproximação linear por partes obtida é ilustrada na Figura 12.

#### 4.4 APROXIMAÇÃO LINEAR POR PARTES CONTÍNUA

Os modelos de linearização por partes apresentados até o momento podem produzir funções aproximativas descontínuas, conforme ilustra a Figura 11. Se as funções aproximativas forem semicontínuas inferiormente, o envelope inferior de tais funções será uma função convexa que não comprometerá a convexidade de um problema de otimização (VIELMA; NEMHAUSER, 2010). No entanto, o usuário poderá exigir que as funções aproximativas por partes sejam contínuas, seja porque a função real é de fato contínua, ou

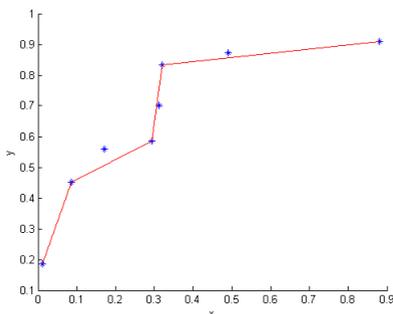


Figura 12 – Aproximação Linear por Partes para a Instância Exemplo com 4 Segmentos.

porque a continuidade oferece propriedades que facilitam a solução numérica ou modelagem em programação matemática. Surge assim o interesse em impor condições que garantam continuidade.

#### 4.4.1 Assegurando Continuidade com os Pontos de Quebra

Uma alternativa para se garantir continuidade é fazer com que cada conjunto de pontos  $P_j$  seja aproximado pelo segmento de reta que passa pelos pontos  $p_{i(j)}$  e  $p_{i(j+1)}$ . Para se obter um algoritmo que produz a aproximação linear por partes que minimiza o erro quadrático, tanto com um número fixo quanto um número máximo de segmentos, basta calcular os coeficientes  $a_{i,j}$  e  $b_{i,j}$  nos algoritmos DPPL-Fixo e DPPL-Max segundo o equacionamento abaixo:

$$a_{i,j} = \frac{y_j - y_i}{x_j - x_i} \quad (4.6a)$$

$$b_{i,j} = y_j - a_{i,j}x_j \quad (4.6b)$$

**Teorema 3** *Definindo os coeficientes do segmento de reta que aproxima os pontos  $\{p_i, \dots, p_j\}$  conforme o equacionamento (4.6), as soluções recursivas (4.4) e (4.5) produzem a aproximação linear por partes contínua ótima de um conjunto de pontos  $P$  no plano com exatamente e no máximo  $T$  que intersectam os pontos  $p_i$  e  $p_j$ , respectivamente, tendo como função objetivo o desvio quadrático do segmento de reta e o custo por segmento, respectivamente.*

**Prova:** A continuidade do modelo pode ser assegurada, pois os coeficientes das retas de aproximação são determinados de forma a traçar apenas retas que intersectam os extremos de um conjunto  $P_j$  de pontos. Já a questão de otimalidade pode ser provada de forma análoga ao do Teorema 1  $\square$

A execução do algoritmo DPPL-Max com um limite de  $T = 5$  segmentos de retas e custo  $c = 0,01$ , porém com os segmentos de reta definidos pelas equações (4.6), produz a linearização por partes ilustrada na Figura 13. Note que a aproximação linear por partes é contínua nos pontos de quebra, sendo a aproximação definida pela sequência de índices  $\langle i(1), \dots, i(5) \rangle = \langle 1, 2, 4, 6, 8 \rangle$  que, coincidentemente, é muito semelhante à sequência obtida sem a garantia de continuidade. A solução obtida tem custo de aproximação  $F(8, 5) = 0,046556$ , superando o valor ótimo  $0,044208$  da aproximação sem garantia de continuidade. Esta condição é sempre verificada devido à convexidade de (4.1).

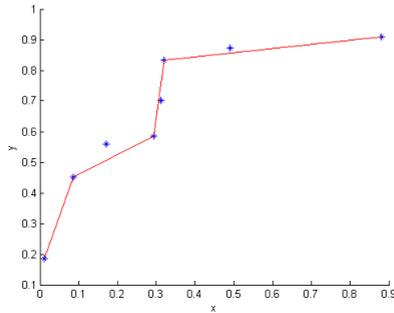


Figura 13 – Linearização por Partes da Instância Exemplo com 4 Segmentos e Garantia de Continuidade nos Pontos de  $P$ .

#### 4.4.2 Garantindo Continuidade Através de Restrições

Outra forma de se assegurar a continuidade consiste em modificar as funções recursivas (4.4) e (4.5) para que se rejeite aproximações que não sejam contínuas. Para o modelo em que o número de segmentos de reta é fixo, com  $t > 1$ , a função recursiva assume a forma:

$$F(j, t) = \min_{i \in S_{j,t}} F(i, t-1) + E(i, j) \quad (4.7)$$

onde  $S_{j,t} = \{i : t \leq i \leq j\}$  e a reta que aproxima  $P_j$  e a reta que aproxima  $P_{j-1}$  se interceptam no intervalo  $x_{\max\{1,i-1\}} \leq x \leq x_{\min\{n,i+1\}}$ , sendo  $F(j,t) = \infty$  se  $S_{j,t} = \emptyset$ . Além disso,  $E(i,i) = \infty$ ,  $F(j,t) = \infty$  para  $j \leq t$  e  $F(j,1) = E(1,j)$ .

O pseudo-código do algoritmo de programação dinâmica que determina a linearização por partes ótima utilizando um número fixo de segmentos de reta,  $T$ , sujeito à restrição de continuidade dos segmentos para resolução recursiva (4.7), é apresentado em Algoritmo 3. Como pode-se observar, DPPL-Fixo-Cont-Restrição é uma versão modificada de DPPL-Fixo.

---

**Algorithm 3:** DPPL-Fixo-Cont-Restrição( $P, T$ )

---

```

1 Linhas 1 - 7 iguais de DPPL-Fixo
2 for j = 1 to n do
3   for t = j to T do
4     F(j,t) := ∞; I(j,t) := 0; X(j,t) := 0;
5     F(j,1) := E(1,j); I(j,1) := 1; X(j,1) := 0;
6     for t = 2 to min{j - 1, T} do
7       F(j,t) := ∞; I(j,t) := 0; X(j,1) := 0;
8       for i = t to j - 1 do
9         k := I(i, t - 1);
10        a := A(k, i); b := B(k, i);
11        ā := A(i, j); b̄ := B(i, j);
12        if a ≠ ā then
13          x := (b - b̄)/(a - ā);
14          if  $x_{\max\{1,i-1\}} \leq x \leq x_{\min\{n,i+1\}}$  and
15             F(j,t) > F(i, t - 1) + E(i, j) then
16            F(j,t) := F(i, t - 1) + E(i, j); I(j,t) := i;
17            X(j,t) := x;

```

---

A execução do algoritmo DPPL-Fixo-Cont-Restrição à instância exemplo com  $T = 3$  segmentos de reta produz a linearização por partes ilustrada na Figura 14. Note que a aproximação linear por partes é contínua, sendo a aproximação definida pela sequência de índices ótima  $\langle i(1), \dots, i(4) \rangle = \langle 1, 5, 6, 8 \rangle$ , a qual difere da sequência ótima sem restrição de continuidade. O custo da aproximação linear é  $F(8, 3) = 0,023148$  que excede o custo  $0,016648$  de aproximação sem restrição de continuidade.

Note, no entanto, que o primeiro segmento de reta terá como domínio  $[x_1, 0, 31207] = [0, 011420, 0, 31207]$ , o segundo,  $[0, 31207, 0, 32160]$ , e por

fim o terceiro terá como domínio  $[0, 32160, x_8] = [0, 32160, 0, 910477]$ .

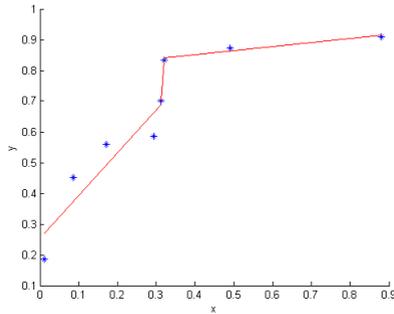


Figura 14 – Linearização por Partes da Instância Exemplo com 3 Segmentos de Retas e Continuidade Assegurada por Restrições.

#### 4.5 INDUZINDO ENVELOPE SUPERIOR E INFERIOR

Será apresentado nesta seção como se obtém o envelope superior de aproximação linear por partes. O envelope inferior é construído de forma análoga. Para tanto, basta calcular os coeficientes  $a_{i,j}$  e  $b_{i,j}$  que induzem a reta  $y = a_{i,j}x + b_{i,j}$  que minimiza o erro quadrático de aproximação dos pontos  $\{p_i, \dots, p_j\}$ , enquanto garante-se que a reta está acima destes pontos. Formalmente, calcula-se os coeficientes resolvendo o seguinte problema de programação quadrática:

$$L_{i,j} : \min_{a_{i,j}, b_{i,j}} \sum_{k=i}^j (a_{i,j}x_k + b_{i,j} - y_k)^2 \quad (4.8a)$$

$$s.a : a_{i,j}x_k + b_{i,j} \geq y_k, p_k \in \{p_i, \dots, p_j\} \quad (4.8b)$$

Os algoritmos que geram a linearização por partes ótima com exatamente e no máximo  $T$  segmentos são facilmente modificados para resolver o problema em questão. Basta trocar o procedimento de cálculo da reta que aproxima os pontos  $\{p_i, p_{i+1}, \dots, p_j\}$  pela resolução do problema  $L_{i,j}$ , o qual consiste de um problema de programação quadrática. O tempo de execução do algoritmo resultante será  $\mathcal{O}(n^3g(n))$  onde  $g(n)$  define um limite para o tempo de solução do problema de programação quadrática.

A execução do algoritmo DPPL-Fixo, com coeficientes computados a

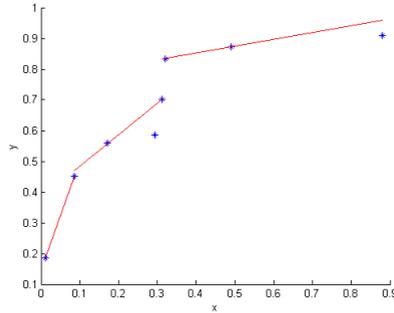


Figura 15 – Linearização por Partes da Instância Exemplo com 3 Segmentos de Retas que Definem o Envelope Superior dos Pontos.

partir da solução dos problemas  $L_{i,j}$ , à instância exemplo com  $T = 3$  segmentos de reta gerou a linearização por partes ilustrada na Figura 15. O custo da aproximação é  $F(8, 3) = 0,028762$ , sendo induzido pela sequência ótima  $\langle i(1), \dots, i(T+1) \rangle = \langle 1, 2, 5, 8 \rangle$ .

Vale ressaltar que as linearizações que induzem envelopes superior e inferior podem ser sujeitas a restrições de continuidade. Os modelos e algoritmos que asseguram a continuidade da aproximação com o emprego de restrições podem ser aplicados de maneira direta, sendo suficiente o uso dos coeficientes  $a_{i,j}$  e  $b_{i,j}$  do segmento de reta obtidos através da solução do problema  $L_{i,j}$ .

O algoritmo DPPL-Fixo-Cont-Restrição, aplicado à instância exemplo com  $T = 3$  segmentos de reta que definem o envelope superior, produz a linearização por partes apresentada na Figura 16. O custo desta aproximação é  $F(8, 3) = 0,04609$ , estando acima do custo sem a imposição de restrição de continuidade, como era de se esperar.

A sequência ótima é  $\langle i(1), \dots, i(T+1) \rangle = \langle 1, 5, 6, 8 \rangle$ . Já o domínio dos segmentos de reta são  $[x_1, 0, 31531] = [0, 011420, 0, 31531]$  para o primeiro segmento,  $[0, 31531, 0, 32229]$  para os segundo e  $[0, 32229, x_8] = [0, 32229, 0, 910477]$  para o terceiro.

#### 4.6 SUMÁRIO

Aplicações de modelos lineares por partes são inúmeras na ciência e na engenharia. Em nosso contexto, a engenharia de petróleo, a curva de

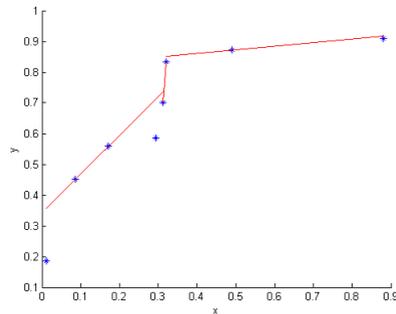


Figura 16 – Linearização por Partes da Instância Exemplo com 3 Segmentos de Retas que Definem o Envelope Superior dos Pontos e com Restrição de Continuidade.

pressão de descarga dos compressores pode ser aproximada por uma função linear por partes. Na otimização, problemas não-lineares são convertidos em problemas de programação inteira-mista através da aproximação das funções não-lineares por modelos lineares por partes. Dada a sua importância, neste capítulo foram apresentados algoritmos desenvolvidos para a linearização por partes de funções não-lineares com o compromisso da qualidade da aproximação. Os modelos e algoritmos servem como ferramentas para auxiliar engenheiros à encontrar modelos lineares por partes que se adequam melhor às necessidades e características das aplicações.

Os algoritmos aqui desenvolvidos consideravam uma variedade de condições, incluindo um número fixo ou máximo de segmentos lineares, procedimentos e restrições para aplicação de continuidade e restrições para envelope inferior e superior. Estas condições podem ser combinadas criando modelos e algoritmos específicos, tais como uma função linear por partes aproximada pelo envelope superior, com continuidade.

Um trabalho futuro ligado a estes algoritmos é a extensão dos mesmos para funções multidimensionais.

## 5 FORMULAÇÕES LINEARES POR PARTES

A não-convexidade da função custo de operação  $h_j^c$  em (3.2a) torna o problema consideravelmente difícil, além do problema ser NP-Difícil. Enquanto algoritmos de programação inteira (PI) podem lidar com problemas de alocação de instalações, algoritmos e teorias de otimização global não são efetivos quando não se tem a convexidade da função objetivo e das restrições. Uma alternativa viável é a convexificação do PEC.

Através da linearização por partes da função  $h_j^c$ , a formulação não-linear inteira é aproximada por um programa linear inteiro misto (PILM). Diferentes formulações inteiras-mistas para funções lineares por partes pode ser encontradas na literatura, algumas que explicitam as restrições por meio de variáveis binárias (PADBERG, 2000; SHERALI, 2001; FLOUDAS, 1995) e outras que aplicam as restrições de forma implícita por meio de variáveis SOS2 (*specialy ordered set variables of type II*) durante o processo de enumeração implícita tipo *branch-and-bound* (BEALE; TOMLIN, 1970; TOMLIN, 1988; KEHA; FARIAS; NEMHAUSER, 2006).

Serão aplicadas a seguir no problema em questão modelos lineares por partes com e sem variáveis inteiras para linearizar a função custo  $h_j^c$ . Estas formulações serão então avaliadas e comparadas mais adiante na resolução do PEC.

### 5.1 FORMULAÇÃO LINEAR POR PARTES UTILIZANDO VARIÁVEIS BINÁRIAS

Para a formulação do problema linear por partes com variáveis binárias, será utilizada a formulação proposta por Sherali (SHERALI, 2001) devido a sua caracterização intuitiva e propriedades quanto à relaxação contínua.

**Definição 9**  $Q_j = \{(q_j^{c,0}, h_j^{c,0}), \dots, (q_j^{c,\kappa(j)}, h_j^{c,\kappa(j)})\}$  é um conjunto de pontos, onde  $q_j^{c,i}$  é a vazão de gás do compressor  $j$  e  $h_j^{c,i}$  seu respectivo custo de operação para a linearização por partes, com:

1.  $q_j^{c,k-1} < q_j^{c,k}$  para todo  $k \in K(j)$ , com  $K(j) = \{1, \dots, \kappa(j)\}$ ;
2.  $q_j^{c,0} = q_j^{c,\min}$  e  $q_j^{c,\kappa(j)} = q_j^{c,\max}$ .

Neste trabalho assume-se que os  $n$  pontos utilizados para a linearização por partes são encontrados e avaliados por um engenheiro de campo. Tipicamente  $n = \kappa(j)$ ,  $K(j) = \{1, \dots, n\}$  e  $h_j^{c,k} = h_j^{c,k}(q_j^{c,k})$ , mas tanto

o número de pontos a serem utilizados, como o possível desejo de se obter um envelope superior para a aproximação, podem ser realizados utilizando os algoritmos apresentados no capítulo anterior.

Seguindo os passos descritos por Sherali (SHERALI, 2001), a aproximação linear por partes do problema PEC é dado por:

$$P_{\text{pl}}^{01} : \min f_{\text{pl}}^{01} = \sum_{j \in N} c_j y_j + \sum_{i \in M} \sum_{j \in N_i} c_{i,j} x_{i,j} + \sum_{j \in N} \sum_{k \in K(j)} \left( h_j^{c,k-1} \lambda_j^{k,L} + h_j^{c,k} \lambda_j^{k,R} \right) \quad (5.1a)$$

sujeito a:

$$\text{Eqs. (3.2b)–(3.2d), (3.2f)–(3.2j)} \quad (5.1b)$$

$$q_j^c = \sum_{k \in K(j)} \left( q_j^{c,k-1} \lambda_j^{k,L} + q_j^{c,k} \lambda_j^{k,R} \right), \quad j \in N \quad (5.1c)$$

$$\sum_{k \in K(j)} z_j^k = y_j, \quad j \in N \quad (5.1d)$$

$$\lambda_j^{k,L} + \lambda_j^{k,R} = z_j^k, \quad j \in N, \quad k \in K(j) \quad (5.1e)$$

$$\lambda_j^{k,L}, \lambda_j^{k,R} \geq 0, \quad j \in N, \quad k \in K(j) \quad (5.1f)$$

$$z_j^k \in \mathbb{Z}, \quad j \in N, \quad k \in K(j) \quad (5.1g)$$

com as seguintes variáveis:

- $\lambda_j^{k,L}$  ( $\lambda_j^{k,R}$ ) representa o peso do ponto da esquerda (direita) do intervalo de  $q_j^{c,k-1}$  a  $q_j^{c,k}$ ;
- $z_j^k$  assume o valor 1, se e somente se,  $q_j^c$  é aproximado linearmente pelos pontos  $q_j^{c,k-1}$  e  $q_j^{c,k}$ , ou seja,  $q_j^c = q_j^{c,k-1} \lambda_j^{k,L} + q_j^{c,k} \lambda_j^{k,R}$ ;
- $y_j$ ,  $q_j^c$  e  $x_{i,j}$  são definidos da mesma forma que em  $P$ , assim como os demais parâmetros.

Buscando facilitar a compreensão do modelo de Sherali a linearização por partes de uma curva  $h(q)$  é ilustrada na Figura 17.

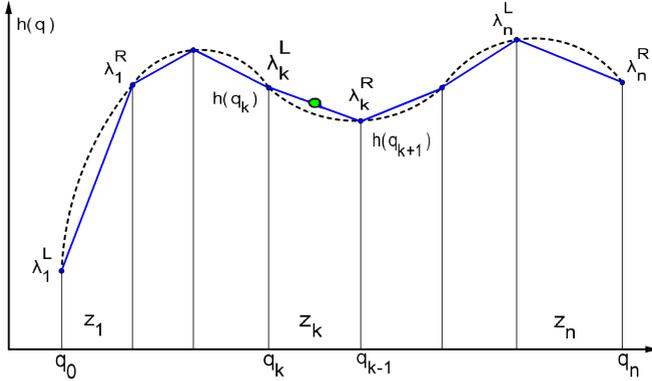


Figura 17 – Modelo Linear por Partes de Sherali.

## 5.2 FORMULAÇÃO LINEAR POR PARTES SEM A UTILIZAÇÃO DE VARIÁVEIS BINÁRIAS

Em um trabalho pioneiro, Beale e Tomlin (BEALE; TOMLIN, 1970) sugerem representar funções lineares por partes utilizando variáveis  $\lambda$  como peso para a combinação convexa enquanto aplicam a restrição de que apenas duas destas variáveis devem ser diferentes de zero e ainda consecutivas durante a busca *Branch-and-bound*. Tal restrição foi denominada como *specialy ordered set variables of type II* (SOS2).

Como exemplo, suponha que  $\{\lambda_j^0, \dots, \lambda_j^{\kappa(j)}\}$  sejam SOS2 e seja  $\{\tilde{\lambda}_j^0, \dots, \tilde{\lambda}_j^{\kappa(j)}\}$  a solução atual, onde  $\tilde{\lambda}_j^{k_1}, \tilde{\lambda}_j^{k_2} > 0$  para  $k_1, k_2 \in \{0, \dots, \kappa(j)\}$ , com  $k_1 < k_2$  e  $k_2 - k_1 \geq 2$ . Então, esta infactibilidade pode ser descartada adicionando a restrição  $\lambda_j^0 = \dots = \lambda_j^{k_1} = 0$  em um *branch*, e adicionando a restrição  $\lambda_j^{k_1+2} = \dots = \lambda_j^{\kappa(j)} = 0$  no outro *branch*.

A aproximação linear inteira mista do problema PEC utilizando variáveis SOS2 e descartando a utilização de variáveis binárias para a linearização por partes da função custo é dada por:

$$P_{pl} : \min f_{pl} = \sum_{j \in N} c_j y_j + \sum_{i \in M} \sum_{j \in N_i} c_{i,j} x_{i,j} + \sum_{j \in N} \sum_{k \in \tilde{K}(j)} h_j^{c,k} \lambda_j^k \quad (5.2a)$$

sujeito a:

$$\text{Eq.s. (3.2b)–(3.2d), (3.2f)–(3.2j)} \quad (5.2b)$$

$$q_j^c = \sum_{k \in \widehat{K}(j)} \lambda_j^k q_j^{c,k}, \quad j \in N \quad (5.2c)$$

$$\sum_{k \in \widehat{K}(j)} \lambda_j^k = y_j, \quad j \in N \quad (5.2d)$$

$$\lambda_j^k \geq 0, \quad j \in N, \quad k \in \widehat{K}(j) \quad (5.2e)$$

$$\{\lambda_j^k\}_{k \in \widehat{K}(j)} \text{ is SOS2}, \quad j \in N \quad (5.2f)$$

onde  $\widehat{K}(j) = K(j) \cup \{0\}$ . Como definido anteriormente, a restrição SOS2 sobre  $\{\lambda_j^k\}_{k \in \widehat{K}(j)}$  significa que não mais do que duas variáveis são positivas e, caso duas delas forem positivas, então estas devem ser consecutivas.

Buscando facilitar a compreensão do modelo SOS2 a linearização por partes de uma curva  $h(q)$  é ilustrada na Figura 18.

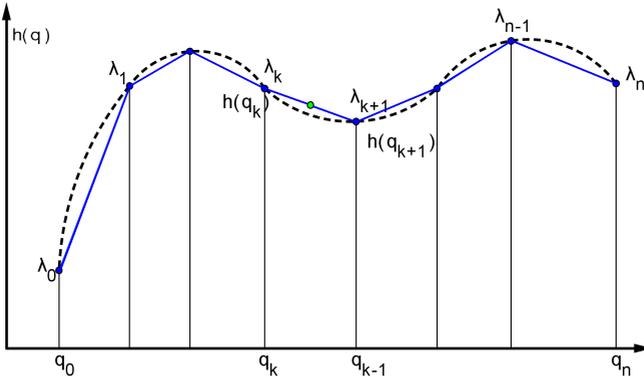


Figura 18 – Modelo Linear por Partes SOS2.

### 5.3 PLANOS DE CORTE

Nas seções a seguir, será apresentado uma família de desigualdades válidas para o PEC buscando desigualdades fortes que auxiliarão a resolução do problema. Como já comentado, estas desigualdades serão baseadas nas desigualdades de cobertura para o problema da mochila (*cover inequalities*)

(WOLSEY, 1998).

### 5.3.1 Desigualdades de Cobertura de Instalações

As desigualdades de cobertura foram descobertas no contexto do problema da mochila. Intensas pesquisas foram realizadas até serem desenvolvidas teorias consistentes para a definição de facetas, procedimentos de *lifting* eficientes e extensões para problemas relacionados (BALAS; ZEMEL, 1978). Estas desigualdades provaram ser efetivas na resolução deste problema e também em problemas semelhantes. Buscando os mesmos ganhos, utilizaremos a técnica de desigualdade de cobertura para propor desigualdades válidas para o fecho convexo do problema de escalonamento de compressores.

Antes de serem apresentadas a definição de cobertura e a desigualdade de cobertura, considere que:

1.  $M_{i,j} = \{l \in M_j : q_{l,j}^{c,\max} \geq q_{i,j}^{c,\max}\}, \forall j \in N, i \in M_j, \text{ e}$
2.  $q(C) = \sum_{i \in C} q_i^w$ , para  $C_{i,j} \subseteq M_{i,j}$ .

Observe que o conjunto  $M_{i,j}$  representa o conjunto de poços para os quais a taxa máxima de saída do compressor excede a taxa máxima de saída do compressor para o poço  $i$ . O valor  $q(C)$  expressa o total de gás necessário para atender os poços em  $C$ .

Claramente,

$$p_l^w + l_{i,j} \leq p_i^w + l_{i,j} \implies l \in M_{i,j} \text{ e}$$

$$q_{l,j}^{c,\max} \geq q_{i,j}^{c,\max} \implies M_{l,j} \subseteq M_{i,j}.$$

**Definição 10** Um conjunto  $C_{i,j} \subseteq M_{i,j}$  é uma cobertura para a instalação  $j$  que serve o cliente  $i$ , se  $q(C_{i,j}) > q_{i,j}^{c,\max}$ .

**Definição 11**  $C_{i,j}$  é uma cobertura mínima se  $q(C_{i,j} - \{l\}) \leq q_{i,j}^{c,\max}$  para todo  $l \in C_{i,j}$ .

Como ilustração, observe a instância exemplo. Note que  $M_{3,2} = \{3, 4, 5, 6\}$  e  $C_{3,2} = \{3, 4, 5\}$  defini uma cobertura, pois  $q(C_{3,2}) = 2 + 4 + 5 = 11 > 10,6014 = q_{3,2}^{c,\max}$ . Esta cobertura é mínima dado que

$$q(C_{3,2} - \{3\}) = 9 < q_{3,2}^{c,\max},$$

$$q(C_{3,2} - \{4\}) = 7 < q_{3,2}^{c,\max},$$

$$q(C_{3,2} - \{5\}) = 6 < q_{3,2}^{c,\max}.$$

**Proposição 5** A desigualdade de cobertura induzida pela cobertura  $C_{i,j}$  para a instalação  $j$  que serve o poço  $i$ ,

$$\sum_{l \in C_{i,j}} x_{l,j} \leq |C_{i,j}| - 1, \quad (5.3)$$

é uma desigualdade válida para  $\text{conv}(\mathcal{X})$ .

**Prova:** Qualquer solução candidata a violar a desigualdade deverá ter  $x_{l,j} = 1$  para todo  $l \in C_{i,j}$  e, assim,  $q(C_{i,j}) > q_{i,j}^{c,\max}$ .

A prova desta proposição será dividida em dois casos. Primeiro, se  $q_{i,j}^{c,\max} < q_j^{c,\max}$ , então a pressão de descarga  $p_j^c \leq p_j^c(q(C_{i,j})) < p_j^c(q_{i,j}^{c,\max}) = p_i^w$ , tornando a solução infactível uma vez que o cliente  $i$  é atendido. No segundo caso, se  $q_{i,j}^{c,\max} = q_j^{c,\max}$ , então  $q(C_{i,j}) > q_{i,j}^{c,\max} = q_j^{c,\max}$ , o que torna a solução infactível. Sendo assim, qualquer que seja o caso, toda solução candidata que viola a desigualdade é infactível.  $\square$

Como demonstrado acima, a desigualdade (5.3) é uma desigualdade válida para o PEC. Precisa-se agora obter uma forma de utilizá-la e ainda, se possível, estendê-la a outras variáveis.

### 5.3.2 Lifting para a Desigualdade de Cobertura de Instalações

Seja  $V = M_j \setminus C_{i,j}$  e  $T = |M_j \setminus C_{i,j}|$ . Por definição,  $V$  contém todos os poços que não foram relacionados na cobertura  $C_{i,j}$  e que podem ser atendidos pelo compressor  $j$ . O objetivo agora é estender a cobertura às variáveis de decisão relacionadas a estes poços contidos em  $V$ , para que possamos tornar a desigualdade ainda mais forte.

Aplicando o procedimento de *lifting* descrito em (NEMHAUSER; WOLSEY, 1988) para a desigualdade de cobertura (5.3), a seguinte desigualdade é obtida:

$$\sum_{l \in C_{i,j}} x_{l,j} + \sum_{t=1}^T \alpha_{i(t),j}(C_{i,j}) x_{i(t),j} \leq |C_{i,j}| - 1 \quad (5.4)$$

onde  $\alpha_{i(t),j}(C_{i,j})$  são constantes não negativas. Para a determinação destas constantes é necessário uma ordenação  $\bar{C}_{i,j} = \langle i(1), i(2), \dots, i(T) \rangle$  em que as variáveis  $x_{l,j}$  serão sujeitas ao *lifting*, com  $l \in V$ . Só assim, o coeficiente  $\alpha_{i(t),j}(C_{i,j})$  é determinado pela solução do problema:

$$L(t) : \xi_{i(t),j}(C_{i,j}) = \max \sum_{l \in C_{i,j}} x_{l,j} + \sum_{l=1}^{t-1} \alpha_{i(l),j}(C_{i,j}) x_{i(l),j} \quad (5.5a)$$

sujeito a:

$$q_j^c \leq q_{i(t),j}^{c,\max} \quad (5.5b)$$

$$q_j^c \leq q_{i(t),j}^{c,\max} + (q_{l,j}^{c,\max} - q_{i(t),j}^{c,\max}) x_{l,j},$$

$$l \in C_{i,j} \cup \{i(1), \dots, i(t-1)\},$$

dado que  $q_{l,j}^{c,\max} < q_{i(t),j}^{c,\max}$  (5.5c)

$$\sum_{l \in C_{i,j} \cup \{i(1), \dots, i(t-1)\}} q_l^w x_{l,j} \leq q_j^c - q_{i(t),j}^w, \quad (5.5d)$$

$$x_{l,j} \in \{0, 1\}, l \in C_{i,j} \cup \{i(1), \dots, i(t-1)\} \quad (5.5e)$$

e recursivamente definindo  $\alpha_{i(k),j}(C_{i,j}) = |C_{i,j}| - 1 - \xi_{i(k),j}(C_{i,j})$  para  $k = 1, \dots, t$ .

O problema  $L(t)$  torna-se um problema padrão da mochila para todo  $t$ , se as variáveis  $x_{l,j}$ , com  $l \in V$ , são submetidas ao *lifiting* em ordem decrescente de  $q_{l,j}^{c,\max}$ . Tal ordem implica que  $q_{i(t),j}^{c,\max} \leq q_{l,j}^{c,\max}$  para todo  $l \in C_{i,j} \cup \{i(1), \dots, i(t-1)\}$  e assim  $L(t)$  terá apenas a restrição

$$\sum_{l \in C_{i,j} \cup \{i(1), \dots, i(t-1)\}} q_l^w x_{l,j} \leq q_{i(t),j}^{c,\max} - q_{i(t),j}^w$$

além da restrição 0-1.

**Proposição 6** *A desigualdade de cobertura com o lifiting (5.4) é válida para o conv( $\mathcal{X}$ ).*

A prova desta proposição será omitida por ser similar à prova de lifiting padrão descrita em (NEMHAUSER; WOLSEY, 1988) e seguindo do fato que

$$\begin{aligned} \alpha_{i(t),j}(C_{i,j}) &= |C_{i,j}| - 1 - \xi_{i(t),j}(C_{i,j}) \\ &\leq |C_{i,j}| - 1 - \max \left\{ \sum_{l \in C_{i,j}} x_{l,j} + \sum_{l=1}^{t-1} \alpha_{i(l),j}(C_{i,j}) x_{i(l),j} : \mathbf{u} \in \mathcal{X} \right\}, \end{aligned}$$

pois o espaço solução das variáveis  $x_{l,j}$ ,  $l \in V$ , é menos restrito em  $L(t)$  do

que em  $\mathcal{X}$ .

De forma ilustrativa, a aplicação do procedimento de *lifting* à cobertura  $C_{3,2} = \{3, 4, 5\}$ , da instância exemplo, produz a desigualdade  $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} \leq 2$ , independente da ordem em que as variáveis  $x_{i,2}$  sejam submetidas ao *lifting*,  $i \in M_{3,2} \setminus C_{3,2} = \{1, 2, 6\}$ .

Ao invés de um *lifting* exato como foi feito acima, foi sugerido por Crowder em (CROWDER; JOHNSON; PADBERG, 1983) a relaxação contínua do problema de *lifting* para reduzir o seu custo computacional. Para entender esta opção, seja  $\tilde{L}(1)$  a relaxação de  $L(1)$  em que  $x_{l,j} \in [0, 1]$  substitui a restrição (5.5e). O *lifting* aproximado produz a desigualdade  $\sum_{l \in C_{i,j}} x_{l,j} + \tilde{\alpha}_{i(1),j}(C_{i,j})x_{i(1),j} \leq |C_{i,j}| - 1$ , onde  $\tilde{\alpha}_{i(1),j}(C_{i,j}) = |C_{i,j}| - 1 - \lfloor \tilde{\xi}_{i(1),j}(C_{i,j}) \rfloor$  e  $\tilde{\xi}_{i(1),j}(C_{i,j})$  é o valor ótimo do objetivo da relaxação. Continuando o processo para  $t = 2, \dots, T$  obtém-se uma desigualdade válida  $\sum_{l \in C_{i,j}} x_{l,j} + \sum_{t=1}^T \tilde{\alpha}_{i(t),j}(C_{i,j})x_{i(t),j} \leq |C_{i,j}| - 1$  para  $\text{conv}(\mathcal{X})$ , onde  $\tilde{\alpha}_{i(t),j}(C_{i,j}) = |C_{i,j}| - 1 - \lfloor \tilde{\xi}_{i(t),j}(C_{i,j}) \rfloor$ ,  $\tilde{\xi}_{i(t),j}(C_{i,j})$  é o objetivo de  $\tilde{L}(t)$  e  $\tilde{\alpha}_{i(l),j}(C_{i,j})$  substitui  $\alpha_{i(l),j}(C_{i,j})$  no objetivo (5.5a) para  $l = 1, \dots, t - 1$ .

A aplicação do procedimento de *lifting* aproximado para a cobertura  $C_{3,2} = \{3, 4, 5\}$  da instância exemplo com  $\bar{C}_{i,j} = \langle (1, 2), (2, 2), (6, 2) \rangle$ , resulta em  $\tilde{\xi}_{i(1),j}(C_{i,j}) = 1,9333$ , o que implica em  $\tilde{\alpha}_{i(1),j}(C_{i,j}) = 1$ ,  $\tilde{\xi}_{i(2),j}(C_{i,j}) = 1,8420$  e  $\tilde{\alpha}_{i(2),j}(C_{i,j}) = 1$ , e finalmente  $\tilde{\xi}_{i(3),j}(C_{i,j}) = 2,7574$  e  $\tilde{\alpha}_{i(3),j}(C_{i,j}) = 0$ . Portanto, a desigualdade obtida é  $x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} \leq 2$ .

### 5.3.3 Problema de Separação das Desigualdades de Cobertura

Dado uma solução fracionária para um problema inteiro misto e uma família de desigualdades válidas, o problema de separação é definido como a procura por uma desigualdade que é violada por esta solução fracionária. A capacidade de realizar a separação é chave para o uso prático das desigualdades válidas em algoritmos de planos de cortes e em software de otimização especializados. Para muitas famílias de desigualdades, um procedimento de separação eficiente não é conhecido ou tem-se mostrado um problema NP-Difícil.

A seguir será apresentado um método de separação para a família de desigualdades de cobertura de instalações. Este método é baseado em uma estratégia de separação para as desigualdades de cobertura da mochila.

Suponha que  $\bar{\mathbf{u}}$  é uma solução fracionária da relaxação contínua  $\bar{P}$  de  $P$ . Deseja-se encontrar uma cobertura  $C_{i,j}$  para uma instalação  $j$  que atenda o cliente  $i$  e induza uma desigualdade violada por  $\bar{\mathbf{u}}$ . Note, antes de mais

nada, que a desigualdade de cobertura apresentada na inequação (5.3) pode ser escrita como

$$\sum_{l \in C_{i,j}} (1 - x_{l,j}) \geq 1 \quad (5.6)$$

tal que  $q(C_{i,j}) \geq q_{i,j}^{c,\max}$ .

Então, para  $j$  e  $i$  fixos, com  $i \in M_j$ , a procura pela desigualdade mais violada, ou pelo menos viável induzida, por uma cobertura  $C_{i,j}$  é representada por um programa inteiro baseado na formulação de separação padrão (CROWDER; JOHNSON; PADBERG, 1983):

$$S_{i,j} : v_{i,j} = \min \sum_{l \in M_{i,j} \setminus \{i\}} (1 - \bar{x}_{l,j}) s_{l,j} - \bar{x}_{i,j} \quad (5.7a)$$

sujeito a:

$$\sum_{l \in M_{i,j} \setminus \{i\}} q_l^w s_{l,j} > q_{i,j}^{c,\max} - q_i^w \quad (5.7b)$$

$$s_{l,j} \in \{0, 1\}, l \in M_{i,j} \setminus \{i\} \quad (5.7c)$$

onde a cobertura de instalações é  $C_{i,j} = \{l : s_{l,j} = 1\} \cup \{i\}$ . Observe que  $C_{i,j}$  induz um plano de corte para  $\bar{u}$  se  $v_{i,j} < 0$  e, que pela forma com que é descrito o objetivo, a cobertura produzida é a cobertura mais violada pela solução  $\bar{u}$ . Caso  $v_{i,j} \geq 0$ , então não existe cobertura que é violada pela solução com os respectivos  $j$  e  $i$  definidos.

Para dimensões pequenas,  $S_{i,j}$  pode ser resolvido usando algoritmos de programação inteira e programação dinâmica com a solução produzida por  $P$ . Uma alternativa computacional é a solução da relaxação contínua  $\bar{S}_{i,j}$  de  $S_{i,j}$ , obtida pela representação de  $s_{l,j}$  por  $\bar{s}_{l,j} \in [0, 1]$ , seguido pelo arredondamento para o número superior mais próximo das variáveis fracionárias  $\bar{s}_{l,j}$  para se obter uma cobertura  $C_{i,j}$ .

Indiferentemente da forma com que a cobertura  $C_{i,j}$  é produzida, uma cobertura mínima  $C'_{i,j} \subseteq C_{i,j}$  pode ser encontrada pela retirada de elementos de  $C_{i,j}$ , e uma desigualdade de cobertura válida (5.3) é obtida. Aplicar o *lifting* à desigualdade produz uma desigualdade de cobertura aumentada que pode cortar a solução fracionária  $\bar{u}$ . Se assim for, o plano de corte resultante dado pela inequação (5.4) pode ser adicionado ao conjunto de restrições.

Considere uma solução fracionária  $\bar{u}$  para a instância exemplo do problema, onde  $\bar{x}_{1,1} = 1$ ,  $\bar{x}_{2,1} = 1$ ,  $\bar{x}_{3,1} = 0$ ,  $550827$ ,  $\bar{x}_{3,3} = 0$ ,  $449173$ ,  $\bar{x}_{4,3} = 1$ ,  $\bar{x}_{5,3} = 0$ ,  $665383$ ,  $\bar{x}_{5,5} = 0$ ,  $334617$ ,  $\bar{x}_{6,3} = 0$ ,  $595285$ ,  $\bar{x}_{6,5} = 0$ ,  $404715$ . Procurando por uma cobertura para instalação 3 que serve o cliente 3, a re-

laxação  $\bar{S}_{3,3}$  produziu a solução  $v_{3,3} = -0,114556$  com  $s_{4,3} = 1$ ,  $s_{5,3} = 0,34682$  e  $s_{6,3} = 0$ , onde  $M_{3,3} = \{3, 4, 5, 6\}$ . O conjunto  $C_{3,3} = \{3, 4, 5\}$  é obtido após o arredondamento para o número superior mais próximo das variáveis fracionárias, que como pode ser visto é uma cobertura, desde que  $q(C_{3,3}) = 2 + 4 + 5 = 11 > 7,71841 = q_{3,3}^{c;\max}$ . Note que a desigualdade resultante  $x_{3,3} + x_{4,3} + x_{5,3} \leq 2$  elimina a solução  $\bar{u}$ , pois  $\bar{x}_{3,3} + \bar{x}_{4,3} + \bar{x}_{5,3} = 0,449173 + 1 + 0,665383 = 2,114556 > 2$ .

A solução do problema inteiro  $S_{3,3}$  produz  $v_{3,3} = -0,114556$  com  $s_{4,3} = 1$ ,  $s_{5,3} = 1$  e  $s_{6,3} = 0$ , induzindo a mesma cobertura.

## 5.4 SUMÁRIO

Devido a não-convexidade da função objetivo do PEC, algoritmos e teorias de otimização global não são efetivos. Buscando contornar este problema e encontrar um ótimo global, a função objetivo do PEC foi linearizada por partes. Esta abordagem transforma o problema inicial em um problema de programação linear inteira-mista, que possui algoritmos eficientes para otimização global.

Motivado por Beale e Tomlin (BEALE; TOMLIN, 1970), abordou-se duas formas de modelagem de funções lineares por partes, uma utilizando variáveis binárias e a outra utilizando variáveis tipo SOS2. Buscando analisar qual das representações é a melhor, as mesmas serão comparadas durante a resolução do PEC no capítulo seguinte.

Ainda buscando melhorias para a resolução do problema em questão, apresentou-se neste capítulo um algoritmo de planos de corte baseados em desigualdades de cobertura. Deseja-se com o algoritmo, obter ganhos computacionais durante a resolução do problema. Conforme citado anteriormente, existem uma infinidade de desigualdades válidas para um problema. O algoritmo de planos de corte tem como objetivo a obtenção de desigualdades válidas de alta dimensionalidade que devem ser adicionadas à formulação inicial, diminuindo o poliedro das soluções factíveis. O algoritmo aqui apresentado será analisado quanto a sua efetividade no capítulo seguinte.

## 6 ANÁLISE NUMÉRICA

Neste capítulo serão apresentados os resultados obtidos durante a realização de alguns experimentos. Experimentos estes que visam comparar as formulações antiga e revisada do PEC, os ganhos obtidos com aplicação dos planos de corte, desenvolvidos na seção 4.3, e também a comparação das formulações lineares por partes com e sem a utilização de variáveis binárias.

### 6.1 COMPARAÇÃO ENTRE AS FORMULAÇÕES ANTIGA E REVISADA

O modelo revisado dado pelas equações (3.2a)–(3.2j) é melhor do que a formulação antiga, apresentada nas equações (3.3a)–(3.3j), no sentido de que o poliedro resultante pela formulação revisada é mais próximo do fecho convexo dentre as duas formulações, conforme estabelece a Proposição 4. O objetivo agora é comparar as formulações computacionalmente.

O programa não-linear inteiro-misto  $P$  e a formulação antiga  $\tilde{P}$  não são passíveis de otimização direta. Por este motivo, a comparação computacional será baseada em suas formas lineares por partes  $P_{pl}^{01}$  e  $\tilde{P}_{pl}^{01}$ , respectivamente. Conforme comentado no Capítulo 2, a formulação de  $\tilde{P}$  é obtida a partir de  $P$ , removendo a restrição (3.2d) e introduzindo as restrições (3.3d) e (3.3e). Já sua forma linear por partes,  $\tilde{P}_{pl}^{01}$ , pode ser obtida com o modelo de Sherali (SHERALI, 2001), de forma semelhante à formulação  $P_{pl}^{01}$ .

As instâncias utilizadas para os experimentos do problema foram obtidas combinando curvas de performance de compressores de fornecedores com demandas de poços plausíveis. Conforme a formulação antiga requer, todas as curvas de performance são côncavas. Três casos que serão utilizados para a comparação das formulações são idênticos aos usados em (CAMPONOGARA et al., 2011), as instâncias com  $n = 7$  compressores e  $m = 16$  poços,  $n = 8$  e  $m = 18$ , e  $n = 9$  e  $m = 14$ . Além destes casos, foram projetados também, instâncias maiores para os experimentos, consistindo de  $n = 14$  compressores e  $m = 20$  poços,  $n = 14$  e  $m = 32$ , e  $n = 16$  compressores e  $m = 34$  poços. Para mais detalhes sobre tais instâncias, ver os Anexos A, B, C.

Os experimentos aplicados ao problema utilizam um procedimento de *cut-and-branch* (WOLSEY, 1998), consistindo de duas fases. Na primeira fase, a relaxação contínua será resolvida iterativamente através da introdução de desigualdades válidas até que não seja mais possível encontrar um plano de corte para a solução fracionária. Para a formulação antiga, as desigualdades válidas utilizadas foram as desigualdades fortes *flow cover*, obtidas com um procedimento de *lifting* aproximado (CAMPONOGARA et al., 2011). Já para a

formulação revisada, a geração de cortes consistiu em executar as seguintes etapas iterativamente:

1. resolver a relaxação em programação linear (PL);
2. resolver o problema de separação exatamente pela solução do problema  $S_{i,j}$  para todo  $i \in M$  e  $j \in N_i$ , se  $y_j > 0$ , obtendo a cobertura mais violada ou a cobertura menos satisfeita  $C_{i,j}$ ;
3. realizar o procedimento de *lifting* da desigualdade de cobertura (5.3) seguindo uma ordem aleatória  $\bar{C}_{i,j}$  e resolver os problemas  $L(t)$  com o software CPLEX; e
4. adicionar a desigualdade (5.4) se esta eliminar a solução PL fracionária atual.

Na segunda fase, as formulações aumentadas com os planos de corte são enviadas a um solver de PLIM (programação linear inteira-mista) que contém estratégias de *branch-and-cut*. O solver utilizado é o solver CPLEX versão 11 com a geração de planos de cortes padrão ativada. A estação de trabalho utilizada pelo solver é equipada com um processador Intel Xeon, com 2.50 GHz, 3.5 Gb of RAM e GNU Linux kernel 2.6.18-5-686.

Tabela 10 – Comparação Entre as Formulações I

$n$	$m$	Formulação Antiga		
		nós	CPU (s)	iterações
7	16	96 845	120,52	806 831
8	18	53 069	51,76	576 455
9	14	11 357	12,22	93 548
14	20	139 027	249,93	1 388 901
14	32	*10 573 690	*42 203,72	*190 492 932
		10 873 990	42 638,15	19 3358 668

Os resultados dos experimentos podem ser observados nas Tabelas 10 e 11. As estatísticas (nós do algoritmo *branch-and-bound*, iterações e tempo de CPU) são médias sobre 7 experimentos independentes das instâncias do PEC com um número variado de intervalos para a linearização por partes. A função custo em ambas formulações e a curva de performance de poços na formulação antiga foram linearizadas com 10, 15, 20, 25, 30, 35 e 40 intervalos.

A estatística computacional em termos de ciclos de CPU, nós e interações não levam em conta a geração de planos de corte, pois muitas das instâncias da antiga formulação estavam apenas disponíveis no formato CPLEX,

Tabela 11 – Comparação Entre as Formulações II

$n$	$m$	Formulação Revisada			Ganhos (%)		
		nós	CPU (s)	iterações	nós	CPU (s)	iterações
7	16	4 552	6,84	44 887	95,30	94,32	94,44
8	18	1 295	4,23	15 403	97,56	91,82	97,33
9	14	1 469	4,12	11 458	87,07	66,25	87,75
14	20	3 541	12,2	51 242	97,45	95,12	96,31
14	32	94 811	1 221,05	2 694 790	99,10	97,11	98,59
		105 668	1 248,45	2 817 780	99,03	97,07	98,54

com cortes já dados. O tempo de CPU para a geração de cortes na formulação revisada tem na média 1.5s, 3.1s, 1s, 2.7s e 7.1s para as instâncias da tabela, respectivamente. As instâncias com  $n = 14$  e  $m = 32$  marcadas com “\*” não puderam ser resolvidas usando a formulação antiga: os resultados apresentados na tabela são somente para a linearização com 10 intervalos, que foi interrompida devido ao *overflow* de memória. O ganho global, dado na última linha das Tabelas 10 e 11, refere-se às estatísticas globais (CPU, iteração e nós gerados) ao longo de todas as cinco instâncias. Não se refere à melhoria percentual média sobre estas instâncias.

Com base no subconjunto de instâncias do problema considerado nos experimentos, os resultados computacionais mostram que a formulação revisada é muito melhor que a anterior. As instâncias do problema revisado foram resolvidas em torno de 10 vezes mais rápido que as instâncias antigas correspondentes. Além disso, apenas a formulação revisada possibilitou tratar maiores instâncias com o solver de PLIM.

## 6.2 ANÁLISE COMPUTACIONAL DAS DESIGUALDADES DE COBERTURA DE INSTALAÇÕES

As experiências anteriores com a formulação antiga mostraram que os planos de corte *flow cover* podem reduzir o tempo de CPU em mais de 30% quando comparado à resolução do problema sem a utilização dos planos de corte. Assim, uma questão natural é se a desigualdade de cobertura de instalações pode ou não reduzir o tempo de solução da formulação revisada. As mesmas instâncias do experimento anterior, juntamente com outra instância de maior escala foram resolvidas usando a geração de planos de corte com as desigualdades de coberturas de instalações e sem a geração de planos de corte usando diretamente o CPLEX na formulação  $P_{pl}^{01}$ . O solver CPLEX

Tabela 12 – Análise da Desigualdade de Cobertura I

<i>n</i>	<i>m</i>	Sem Planos de Corte		
		nós	CPU (s)	iterações
7	16	4 679	6,73	39 803
8	18	1 588	4,81	22 409
9	14	1 879	4,84	13 712
14	20	4 616	16,3	79 194
14	32	132 273	1 878,57	3 866 940
16	34	283 691	2 206,94	5 298 905
		428 726	4 118,13	9 320 963

Tabela 13 – Análise da Desigualdade de Cobertura II

<i>n</i>	<i>m</i>	Com Planos de Corte			Ganho (%)			
		nós	CPU (s)	iter	cortes	nós	CPU (s)	iter
7	16	4 552	8,34	44 888	14	2,72	-23,98	-12,77
8	18	1 296	7,33	15 403	12	18,44	-52,26	31,26
9	14	1 469	5,12	11 458	15	21,81	-5,72	16,44
14	20	3 541	14,89	51 242	17	23,29	8,26	35,3
14	32	94 811	1 228,15	2 694 790	6	28,32	34,62	30,31
16	34	226 275	1 891,33	4 588 610	33	20,24	14,30	13,40
		331 944	3 155,16	7 406 390	97	22,57	23,38	20,54

foi configurado com sua geração de planos de corte padrão na resolução de ambas formulações, com e sem cortes de cobertura de instalações.

Os resultados destes experimentos estão expostos nas Tabelas 12 e 13. Note que na Tabela 13, referente à resolução do problema com a utilização dos planos de corte, também é dado o número de cortes utilizados e, neste caso, o tempo para gerá-los é adicionado às estatísticas no tempo de CPU. Estes resultados indicam que os planos de cortes não ocasionam um grande impacto na eficiência da resolução, com um ganho na ordem de 20% para as instâncias utilizadas na análise experimental. Novamente, o ganho global refere-se às estatísticas globais geradas pelo solver PLIM sobre todos os seis casos, não se referindo à melhoria percentual média sobre esses casos.

### 6.3 COMPARAÇÃO ENTRE AS FORMULAÇÕES LINEARES POR PARTES COM E SEM A UTILIZAÇÃO DE VARIÁVEIS BINÁRIAS

A comparação entre as formulações lineares por partes baseadas na utilização de variáveis binárias e na utilização da restrição SOS2, foi motivada por bons resultados obtidos por Misener (MISENER; GOUNARIS; FLOUDAS, 2009) em um problema semelhante. As formulações  $P_{pl}^{01}$  e  $P_{pl}$  foram resolvidas utilizando novamente o solver CPLEX versão 11 para as mesmas instâncias utilizadas nos experimentos anteriores. Visando avaliar neste momento somente a utilização de variáveis binárias nas formulações, não foram aplicados a elas os planos de corte desenvolvidos.

Tabela 14 – Comparação Entre as Formulações Com/Sem Variáveis Binárias

$n$	$m$	Formulação SOS2		
		nós	CPU (s)	iter
7	16	70 000	62,21	952 817
8	18	7 698	7,12	99 612
9	14	15 763	8,84	108 427
14	20	19 837	32,7	391 499
14	32	688 837	7 520,47	29 685 420
16	34	*5 730 714	*68 315,02	*393 261 150
		6 532 849	75 946,34	42 4498 924

Tabela 15 – Comparação Entre as Formulações Com/Sem Variáveis Binárias

$n$	$m$	Com Variáveis Binárias			Ganhos (%)		
		nós	CPU (s)	iter	nós	CPU (s)	iter
7	16	4 679	6,73	39 803	93,32	89,19	95,82
8	18	1 588	4,81	22 409	79,37	32,44	77,50
9	14	1 879	4,84	13 712	88,08	45,30	87,35
14	20	4 616	16,3	79 194	76,73	50,15	79,77
14	32	132 273	1 878,57	3 866 940	80,80	75,02	86,97
16	34	283 691	2 206,94	5 298 905	95,05	96,77	98,65
		331 944	3 155,16	7 406 390	93,44	94,58	97,80

As Tabelas 14 e 15 apresentam os resultados obtidos nos experimentos computacionais. A instância da formulação SOS2 marcada com uma estrela indica que esta foi interrompida após várias horas de computação com  $\kappa(j) =$

10 para todo  $j \in N$ . Levando em conta este resultado, nenhuma tentativa foi realizada para resolver instâncias com um número maior de intervalos lineares por partes.

Ao contrário de Misener (MISENER; GOUNARIS; FLOUDAS, 2009), os resultados aqui obtidos indicam que a linearização por partes baseada em variáveis binárias é superior à formulação baseada em um conjunto especialmente ordenado de variáveis de tipo II, SOS2.

## 6.4 SUMÁRIO

Após os testes aplicados neste Capítulo pode-se dar ênfase a alguns resultados. A comparação entre as formulações antiga e revisada, através de suas respectivas formulações linearizadas, expressa que a modelagem proposta na formulação revisada reduziu o tempo de solução em torno de dez vezes quando comparado com o modelo anterior. A partir do momento em que o problema faz parte de um sistema em tempo real esta melhoria pode ser significativa.

Com base nas desigualdades de cobertura, a proposta era de fortalecer a formulação inicial aplicando os planos de cortes. Mostra-se que tal proposta rendeu algumas economias em termos de recursos computacionais.

Já a comparação entre o modelo linear por partes com e sem a utilização de variáveis binárias, mostrou-se que, para o problema em questão, a formulação usando variáveis binárias é mais adequada para sua resolução.

## 7 CONCLUSÃO E TRABALHOS FUTUROS

A tecnologia de campos inteligentes pode ser vista como a integração das instalações horizontalmente no espaço e verticalmente no nível de tomada de decisões. No espaço, simuladores e modelos para instalações como reservatórios, poços e equipamentos de superfície, estão sendo integrados para criar um modelo completo de um campo de petróleo. Já no nível de tomada de decisões, o sistema de automação responsável pela operação em tempo-real está sendo integrado com o sistema de planejamento de produção, visando a redução de custos enquanto procura-se a maximizar a produção.

Independentemente da localização geográfica e do nível de tomada de decisão, os problemas para se operar campos de petróleo modernos estão aumentando em complexidade e precisam ser resolvidos rapidamente para aplicações em tempo-real. Para este fim, este trabalho melhorou a modelagem de um problema de escalonamento de compressores, que pode servir como ferramenta para otimização em tempo-real de operações em um campo de petróleo operado por injeção contínua de gás, e também auxiliar no projeto de instalações de compressores durante a fase de planejamento. Através da nova abordagem das restrições sobre a pressão de descarga do compressor, o modelo revisado reduziu o tempo de resolução em torno de dez vezes quando comparado com o modelo anterior. Já com base nas desigualdades de cobertura, a proposta era de criar planos de cortes para fortalecer a formulação inicial, proposta esta que nos rendeu algumas economias em termos de recursos computacionais.

A programação inteira mista é uma ferramenta poderosa à síntese de soluções ótimas globais. Para esse fim, foram propostas formulações desta classe para o problema de escalonamento de compressores. Como um problema de programação inteira mista necessita que as funções, tanto objetivo como as de restrições sejam lineares, algumas funções não-lineares da formulação revisada deveriam ser linearizadas. Para linearização por partes das funções não-lineares, os algoritmos propostos no trabalho demonstraram-se boas ferramentas para problemas de otimização, levando em conta que o número de politopos, retas, influencia no tempo de resolução do problema. Caso conheça-se a função a ser linearizada é possível definir o número desejado ou máximo de retas a serem utilizadas e, ainda, obter uma garantia de otimalidade da linearização.

Através da comparação entre os modelos lineares por partes com e sem a utilização de variáveis binárias, mostrou-se que, para o problema em questão, a formulação usando variáveis binárias é mais adequada para sua resolução.

Como sugestão de trabalhos futuros, deseja-se aplicar um procedimento de *branch-and-cut*, mudando a forma de aplicarmos os planos de cortes. Não mais será aplicando um procedimento de *cut-and-branch*, assim, será enviando a formulação diretamente para o solver, aplicando os cortes baseados em desigualdades de coberturas somente quando necessário. Esta proposta é um pouco audaciosa, pois o solver CPLEX é o estado da arte até o momento e interferir em sua resolução visando ganhos não é tarefa fácil. No algoritmo de planos de corte, deseja-se ainda verificar se é sempre possível encontrar uma cobertura mínima que ainda corte o ponto fracionário fornecido pela relaxação linear. Outra proposta é a de criar novas famílias de planos de corte para o problema de escalonamento de compressores que sejam mais eficientes do que os cortes baseados em desigualdades de cobertura. A fim de avaliar as aproximações lineares inteiras-mista do PEC, deseja-se resolver o problema não-linear inteiro-misto  $P$  fornecendo como ponto inicial, ao algoritmo de resolução, o resultado ótimo encontrado pelo problema linear inteiro-misto  $P_{pl}^{01}$ .

## REFERÊNCIAS

- ANDRADE, E. L. *Introdução à Pesquisa Operacional*. [S.l.]: LTC, 1998.
- BALAS, E.; ZEMEL, E. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, v. 34, n. 1, p. 119–148, 1978.
- BEALE, E. M. L.; TOMLIN, J. A. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In: *Proceedings of the Fifth International Conference on Operations Research*. [S.l.: s.n.], 1970. p. 447–454.
- BERTSEKAS, D. P. *Dynamic Programming and Optimal Control*. [S.l.]: Athena Scientific, 2000.
- CAMPONOVARA, E. et al. Compressor scheduling in oil fields: piecewise-linear formulation, valid inequalities, and computational analysis. *Optimization and Engineering*, v. 12, p. 153–174, 2011.
- CAMPONOVARA, E.; CASTRO, M. P. de. Cover-based cutting planes for a compressor scheduling problem: a computational study. In: *Proceedings of the International Conference on Engineering Optimization*. Rio de Janeiro, Brazil: [s.n.], 2008.
- CAMPONOVARA, E.; CASTRO, M. P. de; PLUCENIO, A. Compressor scheduling in oil fields: a piecewise-linear formulation. In: *Proceedings of the IEEE Conference on Automation Science and Engineering*. Scottsdale, AZ: [s.n.], 2007. p. 436–441.
- CAMPONOVARA, E. et al. Compressor scheduling in oil fields: piecewise-linear formulation, valid inequalities, and computational analysis. *Optimization and Engineering*, v. 12, n. 1-2, p. 153–174, 2011.
- CAMPONOVARA, E.; CONTO, A. M. de. Lift-gas allocation under precedence constraints: MILP formulation and computational analysis. *IEEE Transactions on Automation Science and Engineering*, v. 6, n. 3, p. 544–551, 2009.
- CAMPONOVARA, E.; NAKASHIMA, P. H. R. Optimizing gas-lift production of oil wells: Piecewise linear formulation and computational analysis. *IIE Transactions*, v. 38, n. 2, p. 173–182, 2006.

CAMPONOGARA, E. et al. An automation system for gas-lifted oil wells: model identification, control, and optimization. *Journal of Petroleum Science and Engineering*, v. 70, p. 157–167, 2010.

CORNUEJOLS, G. The uncapacitated facility location problem. 1983.

CROWDER, H.; JOHNSON, E. L.; PADBERG, M. W. Solving large-scale zero-one linear programming problems. *Operations Research*, v. 31, p. 803–834, 1983.

DEVRIES, S. Production management information challenges in the digital oil field. In: *Proc. of the SPE Annual Technical Conference and Exhibition*. Dallas, TX: [s.n.], 2005.

DUTTA-ROY, K.; BARUA, S.; HEIBA, A. Computer-aided gas field planning and optimization. In: *Proceedings of the SPE Production Operations Symposium*. Oklahoma City: [s.n.], 1997.

FERRARI-TRECCATE, G. et al. A clustering technique for the identification of piecewise affine systems. *Automatica*, v. 39, n. 2, p. 205–217, 2003.

FLOUDAS, C. A. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York: Oxford University Press, 1995.

GHOSH, S.; MAKHA, S. A fuzzy clustering based technique for piecewise affine approximation of a class of nonlinear systems. *Communications in Nonlinear Science and Numerical Simulation*, v. 15, n. 9, p. 2235–2244, 2010.

GHOSH, S. et al. A genetic algorithm based clustering approach for piecewise linearization of nonlinear functions. In: *International Conference on Devices and Communications*. [S.l.: s.n.], 2011. p. 1–4.

GUNNERUD, V.; FOSS, B. Oil production optimization – a piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering*, v. 34, n. 11, p. 1803–1812, 2010.

HUPPLER, J. D. Scheduling gas field production for maximum production. *SPE Journal*, v. 14, n. 3, p. 279–294, 1974.

KEHA, A. B.; FARIAS, I. R. de; NEMHAUSER, G. L. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, v. 54, n. 5, p. 847–858, 2006.

KLEINBERG, J.; TARDOS, E. *Algorithm Design*. [S.l.]: Addison Wesley, 2005.

- LEE, H. et al. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial & Engineering Chemistry Research*, v. 35, n. 5, p. 1630–1641, 1996.
- MISENER, R.; GOUNARIS, C. E.; FLOUDAS, C. A. Global optimization of gas lifting operations: a comparative study of piecewise linear formulations. *Industrial & Engineering Chemistry Research*, v. 48, p. 6098–5104, 2009.
- MURRAY, R. et al. Making our mature fields smarter – an industry wide position paper from the 2005 SPE Forum. In: *Proc. of the SPE Intelligent Energy Conference and Exhibition*. Amsterdam: [s.n.], 2006.
- NEMHAUSER, G. L.; WOLSEY, L. A. *Integer and Combinatorial Optimization*. [S.l.]: John Wiley & Sons, 1988.
- NGUYEN, H. H. et al. Aa comparison of automation techniques for optimization of compressor scheduling. *Advances in Engineering Software*, v. 39, n. 3, p. 178–188, 2008.
- PADBERG, M. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, v. 27, n. 1, p. 1–5, 2000.
- REJOWSKI, R.; PINTO, J. M. An MILP formulation for the scheduling of multiproduct pipeline systems. *Brazilian Journal of Chemical Engineering*, v. 19, p. 467–474, 12 2002.
- SAEPUDIN, D. et al. Oil production optimization in a cluster of gas lift wells system. *Journal of Applied Sciences*, v. 10, n. 16, p. 1705–1713, 2010.
- SHERALI, H. D. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Operations Research Letters*, v. 28, n. 4, p. 155–160, 2001.
- TOMLIN, J. A. Special ordered sets and an application to gas supply operations planning. *Mathematical Programming*, v. 42, n. 1–3, p. 69–84, 1988.
- VANDERBEI, R. J. *Linear Programming: Foundations and Extensions*. Boston: Kluwer Academic Publishers, 1996.  
<citeseer.ist.psu.edu/vanderbei96linear.html>.
- VIELMA, S. A. J.; NEMHAUSER, G. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations research*, v. 58, n. 2, p. 303–315, 2010.

WOLSEY, L. *Integer programming*. [S.l.]: Wiley, 1998. (Wiley-Interscience series in discrete mathematics and optimization).

**ANEXO A - Instância com 7 Compressores e 16 poços**



Tabela 16 – Dados do Compressor I

$j$	$c_j$	$d_j$	$q_j^{c,\min}$	$q_j^{c,\max}$
1	8,00	3,00	5,3	13
2	6,00	5,00	4,2	12
3	10,00	1,00	3,8	10,8
4	4,00	7,00	2,0	9,8
5	6,00	4,00	1,0	8,5
6	4,20	2,03	2,64	8,65
7	4,19	3,48	1,83	5,43

Tabela 17 – Parâmetros da Curva de Pressão I

$j$	$\alpha_{0,j}$	$\alpha_{1,j}$	$\alpha_{2,j}$	$\alpha_{3,j}$	$\alpha_{4,j}$
1	8,36900	-0,502034	0,0651919	-0,00497000	0,814770
2	7,11148	-0,602903	0,0680301	-0,00507334	0,766603
3	5,30343	-0,309098	0,0401807	-0,00427144	0,324494
4	3,49652	-1,215010	0,1236120	-0,00741996	1,978650
5	2,45413	-0,356522	0,0361532	-0,00363706	0,347968
6	4,1899	-0,3789	0,0375	-0,0028	0,5832
7	2,8493	-0,2794	0,0295	-0,0021	0,4508

Tabela 18 – Dados dos Poços I

$i$	$q_i^w$	$p_i^w$	$N_i$
1	3	5	{1, 2}
2	4	5	{1, 2}
3	2	4	{1, 2, 3}
4	4	3	{1, 2, 3, 4}
5	5	1,5	{1, 2, 3, 4}
6	2	1	{1, 2, 3, 4}
7	2,03	2,05	{1, 2, 3, 4, 5, 6, 7}
8	1,01	1,63	{1, 2, 3, 4, 5, 6, 7}
9	2,18	1,79	{1, 2, 3, 4, 5, 6, 7}
10	2,78	1,48	{1, 2, 3, 4, 5, 6, 7}
11	3,06	1,87	{1, 2, 3, 4, 5, 6, 7}
12	2,01	0,92	{1, 2, 3, 4, 5, 6, 7}
13	2,57	1,94	{1, 2, 3, 4, 5, 6, 7}
14	1,69	1,07	{1, 2, 3, 4, 5, 6, 7}
15	2,49	1,54	{1, 2, 3, 4, 5, 6, 7}
16	2,17	1,35	{1, 2, 3, 4, 5, 6, 7}

Tabela 19 – Dados da Perda de Pressão nas Linhas I

$i \setminus j$	$l_{i,j}$						
	1	2	3	4	5	6	7
1	0,03	0,01					
2	0,08	0,20					
3	0,15	0,31	0,15				
4	0,27	0,22	0,11	0,09			
5	0,22	0,09	0,15	0,02			
6	0,03	0,01	0,2	0,2			
7	0,3	0,08	0,08	0,01	0,02	0,21	0,06
8	0,27	0,14	0,3	0,24	0,23	0,11	0,05
9	0,25	0,26	0,3	0,28	0,12	0,21	0,24
10	0,23	0,33	0,27	0,23	0,16	0,04	0,22
11	0,08	0,29	0,17	0,24	0	0,32	0
12	0,16	0,3	0,2	0,11	0,16	0,2	0,05
13	0,16	0,13	0,15	0,2	0,02	0,1	0,2
14	0,23	0,26	0,33	0,16	0,3	0,15	0,27
15	0,07	0,15	0,06	0,32	0,12	0,01	0,25
16	0,12	0,27	0,01	0,2	0	0,09	0,29

Tabela 20 – Dados dos Custos das Linhas I

$i \backslash j$	$c_{i,j}$						
	1	2	3	4	5	6	7
1	0,24	0,22					
2	0,28	0,08					
3	0,26	0,24	0,23				
4	0,23	0,15	0,21	0,3			
5	0	0,15	0,18	0,11			
6	0,18	0,14	0,15	0,1			
7	0,16	0,01	0,07	0,12	0,04	0,13	0,02
8	0,21	0,23	0,29	0,12	0,02	0,11	0,2
9	0,29	0,17	0,29	0,15	0,26	0,07	0,07
10	0,19	0,22	0,04	0,12	0,02	0,27	0,25
11	0,29	0,15	0,12	0,3	0	0,23	0
12	0,06	0,26	0,07	0,21	0,29	0,13	0,05
13	0,01	0	0,25	0,27	0,22	0,06	0,07
14	0,2	0,12	0,03	0,14	0,17	0,09	0,19
15	0,22	0,27	0,28	0,28	0,23	0,11	0,25
16	0,03	0,21	0,2	0,14	0	0,16	0,09



**ANEXO B - Instância com 8 Compressores e 18 poços**



Tabela 21 – Dados do Compressor II

$j$	$c_j$	$d_j$	$q_j^{c,\min}$	$q_j^{c,\max}$
1	8,00	3,00	5,3	13
2	6,00	5,00	4,2	12
3	10,00	1,00	3,8	10,8
4	4,00	7,00	2,0	9,8
5	6,00	4,00	1,0	8,5
6	4,20	2,03	2,64	8,65
7	4,19	3,48	1,83	5,43
8	3,89	2,93	1,80	5,5

Tabela 22 – Parâmetros da Curva de Pressão II

$j$	$\alpha_{0,j}$	$\alpha_{1,j}$	$\alpha_{2,j}$	$\alpha_{3,j}$	$\alpha_{4,j}$
1	8,36900	-0,502034	0,0651919	-0,00497000	0,814770
2	7,11148	-0,602903	0,0680301	-0,00507334	0,766603
3	5,30343	-0,309098	0,0401807	-0,00427144	0,324494
4	3,49652	-1,215010	0,1236120	-0,00741996	1,978650
5	2,45413	-0,356522	0,0361532	-0,00363706	0,347968
6	4,1899	-0,3789	0,0375	-0,0028	0,5832
7	2,8493	-0,2794	0,0295	-0,0021	0,4508
8	2,633	-0,2750	0,0315	-0,0025	0,3879

Tabela 23 – Dados dos Poços II

$i$	$q_i^w$	$p_i^w$	$N_i$
1	3	5	{1, 2}
2	4	5	{1, 2}
3	2	4	{1, 2, 3}
4	4	3	{1, 2, 3, 4}
5	5	1.5	{1, 2, 3, 4}
6	2	1	{1, 2, 3, 4, 5, 6}
7	2,03	2,05	{1, 2, 3, 4, 5, 6, 7}
8	1,01	1,63	{1, 2, 3, 4, 5, 6, 7}
9	2,8	1,79	{1, 2, 3, 4, 5, 6, 7}
10	2,78	1,48	{1, 2, 3, 4, 5, 6, 7}
11	3,06	1,87	{1, 2, 3, 4, 5, 6, 7}
12	2,01	0,92	{1, 2, 3, 4, 5, 6, 7}
13	2,57	1,94	{1, 2, 3, 4, 5, 6, 7}
14	1,69	1,07	{1, 2, 3, 4, 5, 6, 7}
15	2,49	1,54	{1, 2, 3, 4, 5, 6, 7}
16	2,17	1,35	{1, 2, 3, 4, 6, 7}
17	2,14	1,58	{1, 2, 3, 4, 5, 6, 7}
18	2,49	1,86	{1, 2, 3, 4, 5, 6, 7}

Tabela 24 – Dados da Perda de Pressão nas Linhas II

$i \backslash j$	$l_{i,j}$							
	1	2	3	4	5	6	7	8
1	0,03	0,01						
2	0,08	0,20						
3	0,15	0,31	0,15					
4	0,27	0,22	0,11	0,09				
5	0,22	0,09	0,15	0,02				
6	0,03	0,01	0,2	0,2	0	0	0,06	0,02
7	0,3	0,08	0,08	0,01	0,02	0,21	0,06	0,06
8	0,27	0,14	0,3	0,24	0,23	0,11	0,05	0,06
9	0,25	0,26	0,3	0,28	0,12	0,21	0,24	0,3
10	0,23	0,33	0,27	0,23	0,16	0,04	0,22	0,05
11	0,08	0,29	0,17	0,24	0	0,32	0	0
12	0,16	0,3	0,2	0,11	0,16	0,2	0,05	0,32
13	0,16	0,13	0,15	0,2	0,02	0,1	0,2	0,21
14	0,23	0,26	0,33	0,16	0,3	0,15	0,27	0,05
15	0,07	0,15	0,06	0,32	0,12	0,01	0,25	0,09
16	0,12	0,27	0,01	0,2	0	0,09	0,29	0,02
17	0,27	0,09	0,25	0,22	0,07	0,2	0,2	0,06
18	0,06	0,12	0,09	0,26	0,1	0,21	0,33	0,31

Tabela 25 – Dados dos Custos das Linhas II

$i \backslash j$	$c_{i,j}$							
	1	2	3	4	5	6	7	8
1	0,24	0,22						
2	0,28	0,08						
3	0,26	0,24	0,23					
4	0,23	0,15	0,21	0,3				
5	0	0,15	0,18	0,11				
6	0,18	0,14	0,15	0,1	0,03	0,08	0,16	0,4
7	0,16	0,01	0,07	0,12	0,04	0,13	0,02	0,3
8	0,21	0,23	0,29	0,12	0,02	0,11	0,2	0,21
9	0,29	0,17	0,29	0,15	0,26	0,07	0,07	0,27
10	0,19	0,22	0,04	0,12	0,02	0,27	0,25	0,09
11	0,29	0,15	0,12	0,3	0	0,23	0	0
12	0,06	0,26	0,07	0,21	0,29	0,13	0,05	0,2
13	0,01	0	0,25	0,27	0,22	0,06	0,07	0,7
14	0,2	0,12	0,03	0,14	0,17	0,09	0,19	0,09
15	0,22	0,27	0,28	0,28	0,23	0,11	0,25	0,21
16	0,03	0,21	0,2	0,14	0	0,16	0,09	0,12
17	0,17	0,08	0,03	0,22	0,25	0,25	0,16	0,11
18	0,06	0,19	0,04	0,15	0,03	0,01	0,02	0,05

**ANEXO C - Instância com 9 Compressores e 14 poços**



Tabela 26 – Dados do Compressor III

$j$	$c_j$	$d_j$	$q_j^{c,\min}$	$q_j^{c,\max}$
1	8,00	3,00	5,3	13
2	6,00	5,00	4,2	12
3	10,00	1,00	3,8	10,8
4	4,00	7,00	2,0	9,8
5	6,00	4,00	1,0	8,5
6	4,20	2,03	2,64	8,65
7	4,19	3,48	1,83	5,43
8	3,89	2,93	1,80	5,5
8	3,25	2,07	1,00	3,12

Tabela 27 – Parâmetros da Curva de Pressão III

$j$	$\alpha_{0,j}$	$\alpha_{1,j}$	$\alpha_{2,j}$	$\alpha_{3,j}$	$\alpha_{4,j}$
1	8,36900	-0,502034	0,0651919	-0,00497000	0,814770
2	7,11148	-0,602903	0,0680301	-0,00507334	0,766603
3	5,30343	-0,309098	0,0401807	-0,00427144	0,324494
4	3,49652	-1,215010	0,1236120	-0,00741996	1,978650
5	2,45413	-0,356522	0,0361532	-0,00363706	0,347968
6	4,1899	-0,3789	0,0375	-0,0028	0,5832
7	2,8493	-0,2794	0,0295	-0,0021	0,4508
8	2,8633	-0,2750	0,0315	-0,0025	0,3879
9	1,5440	-0,1402	0,0153	-0,0012	0,1984

Tabela 28 – Dados dos Poços III

$i$	$q_i^w$	$p_i^w$	$N_i$
1	3	5	{1, 2}
2	4	5	{1, 2}
3	2	4	{1, 2, 3}
4	4	3	{1, 2, 3, 4}
5	5	1,5	{1, 2, 3, 4}
6	2	1	{1, 2, 3, 4, 5, 6, 7}
7	2,03	2,05	{1, 2, 3, 4, 5, 6, 7}
8	1,01	1,63	{1, 2, 3, 4, 5, 6, 7}
9	2,18	1,79	{1, 2, 3, 4, 5, 6, 7}
10	2,78	1,48	{1, 2, 3, 4, 5, 6, 7}
11	3,06	1,87	{1, 2, 3, 4, 6}
12	2,01	0,92	{1, 2, 3, 4, 5, 6, 7, 8, 9}
13	2,57	1,94	{1, 2, 3, 4, 5, 6, 7, 9}
14	1,69	1,07	{1, 2, 3, 4, 5, 6, 7, 8, 9}

Tabela 29 – Dados da Perda de Pressão nas Linhas III

$i \setminus j$	1	2	3	4	5	6	7	8	9
1	0,03	0,01							
2	0,08	0,20							
3	0,15	0,31	0,15						
4	0,27	0,22	0,11	0,09					
5	0,22	0,09	0,15	0,02					
6	0,03	0,01	0,2	0,2	0	0	0,06	0,02	
7	0,3	0,08	0,08	0,01	0,02	0,21	0,06	0,06	
8	0,27	0,14	0,3	0,24	0,23	0,11	0,05	0,06	
9	0,25	0,26	0,3	0,28	0,12	0,21	0,24	0,3	
10	0,23	0,33	0,27	0,23	0,16	0,04	0,22	0,05	
11	0,08	0,29	0,17	0,24	0	0,32	0	0	
12	0,16	0,3	0,2	0,11	0,16	0,2	0,05	0,27	0,32
13	0,16	0,13	0,15	0,2	0,02	0,1	0,2		0,21
14	0,23	0,26	0,33	0,16	0,3	0,15	0,27	0,27	0,05

Tabela 30 – Dados dos Custos das Linhas III

$i \backslash j$	$c_{i,j}$								
	1	2	3	4	5	6	7	8	9
1	0,24	0,22							
2	0,28	0,08							
3	0,26	0,24	0,23						
4	0,23	0,15	0,21	0,3					
5	0	0,15	0,18	0,11					
6	0,18	0,14	0,15	0,1	0,03	0,08	0,16		
7	0,16	0,01	0,07	0,12	0,04	0,13	0,02		
8	0,21	0,23	0,29	0,12	0,02	0,11	0,2		
9	0,29	0,17	0,29	0,15	0,26	0,07	0,07		
10	0,19	0,22	0,04	0,12	0,02	0,27	0,25		
11	0,29	0,15	0,12	0,3	0	0,23	0		
12	0,06	0,26	0,07	0,21	0,29	0,13	0,05	0,2	
13	0,01	0	0,25	0,27	0,22	0,06	0,07	0,27	
14	0,2	0,12	0,03	0,14	0,17	0,09	0,19	0,09	0,16