

**AUGUSTO MARASCA DE CONTO**

**ALOCAÇÃO DE GÁS DE INJEÇÃO EM POÇOS  
DE PETRÓLEO SOB RESTRIÇÕES DE  
PRECEDÊNCIA: LINEARIZAÇÃO POR PARTES E  
PROGRAMAÇÃO INTEIRA**

**FLORIANÓPOLIS  
2006**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO**  
**EM ENGENHARIA ELÉTRICA**

**ALOCAÇÃO DE GÁS DE INJEÇÃO EM POÇOS**  
**DE PETRÓLEO SOB RESTRIÇÕES DE**  
**PRECEDÊNCIA: LINEARIZAÇÃO POR PARTES E**  
**PROGRAMAÇÃO INTEIRA**

Dissertação submetida à Universidade Federal de Santa Catarina  
como requisito parcial à obtenção do grau de

**MESTRE EM ENGENHARIA ELÉTRICA**

por

**AUGUSTO MARASCA DE CONTO**

Florianópolis, setembro de 2006.

# **ALOCAÇÃO DE GÁS DE INJEÇÃO EM POÇOS DE PETRÓLEO SOB RESTRIÇÕES DE PRECEDÊNCIA: LINEARIZAÇÃO POR PARTES E PROGRAMAÇÃO INTEIRA**

Augusto Marasca de Conto

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia na especialidade Engenharia Elétrica, área de concentração Automação e Sistemas, e aprovada em sua forma final pelo Programa de Pós-Graduação.  
Florianópolis, Setembro de 2006.

---

Prof. Eduardo Camponogara, Ph.D. (Orientador)

---

Prof. Nelson Sadowski, Dr.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica  
da Universidade Federal de Santa Catarina

Banca Examinadora:

---

Prof. Eduardo Camponogara, Ph.D. (Orientador)

---

Prof. Mário César Zambaldi, Dr.

---

Prof. José Eduardo Ribeiro Cury, Dr.

---

Prof. Rômulo Silva de Oliveira, Dr.

*Dedico aos meus pais, Augusto e Ivone,  
e à minha irmã Cláudia.*

Resumo da dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

# **ALOCAÇÃO DE GÁS DE INJEÇÃO EM POÇOS DE PETRÓLEO SOB RESTRIÇÕES DE PRECEDÊNCIA: LINEARIZAÇÃO POR PARTES E PROGRAMAÇÃO INTEIRA**

**Augusto Marasca de Conto**

Setembro/2006

Orientador: Eduardo Camponogara

Área de Concentração: Automação e Sistemas

Palavras-chave: Campos de petróleo, alocação de gás de injeção, formulação linear por partes, programação inteira mista, teoria poliedral, restrições de precedência

Número de Páginas: xi + 94

A alocação ótima de gás de injeção busca taxas de injeção para poços de petróleo operados por *gas-lift* que otimize uma função objetivo, normalmente que maximize o lucro, em um campo de petróleo. Apesar do grande interesse nesta classe de problemas, a maioria dos trabalhos nesta área não os trata adequadamente, produzindo soluções sub-ótimas e não considerando explicitamente decisões discretas de ativação e desativação dos poços. Neste trabalho buscamos obter taxas ótimas de injeção de gás considerando restrições de capacidade máxima de gás disponível e restrições de precedência de ativação dos poços. Como a curva de performance do poço, que caracteriza sua resposta, é não-linear, aplicamos a ela um procedimento de linearização por partes. Então utilizamos programação linear inteira mista, que possui um grande ferramental teórico e algorítmico. Este problema pertence a classe *NP-Difícil*, então apresentamos um procedimento para obtenção de cortes que acelera a busca da resposta ótima. Experimentos computacionais mostraram que estes cortes podem reduzir o número de iterações do algoritmo de otimização. A programação inteira mista também nos garante que a solução ótima global seja obtida, e permite uma medida da qualidade da solução pelos limites primal-dual, caso a execução do algoritmo seja interrompida. Desenvolvemos também uma interface de otimização para o usuário, que permite que ele especifique uma instância de um problema de alocação de gás de injeção e obtenha a solução ótima, sem que para isto ele necessite conhecer detalhes do modelo e do algoritmo de resolução do problema.

Abstract of dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

# **LIFT-GAS ALLOCATION TO OIL WELLS UNDER PRECEDENCE CONSTRAINTS: PIECEWISE LINEARIZATION AND INTEGER PROGRAMMING**

**Augusto Marasca de Conto**

September/2006

Advisor: Eduardo Camponogara

Area of Concentration: Automation and Systems Engineering

Key words: Oil fields, lift-gas allocation, piecewise linear formulation, mixed-integer programming, polyhedral theory, precedence constraints

Number of Pages: xi + 94

Optimum lift-gas allocation searches gas injection rates for wells in gas-lifted oil fields that optimizes an objective, generally profit. Despite the long interest, most of the literature lack rigor, producing sub-optimal solutions and not treating explicitly discrete decisions and well activation and deactivation. In this work, we present a formulation for optimum lift-gas allocation with maximum available gas and activation precedence constraints. We use mixed integer linear programming with piecewise linearization of the well performance curves, because of its great theoretical and algorithmic possibilities. This problem belongs to the *NP*-Hard class, thus we propose a procedure to obtain cutting planes that accelerates the search for the optimum allocation. Computational experiments showed that these cuts can reduce the number of algorithmic iterations. Mixed integer programming also produces globally optimum solutions and gives a measure of the solution quality by the primal-dual bounds, in case of the algorithmic execution is interrupted. We also have developed an optimization interface that allows the user to specify an instance of the lift-gas allocation problem and obtain its solution, without necessarily knowing details of the model and the resolution algorithm.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problemática . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Visão geral da dissertação . . . . .	2
<b>2</b>	<b>O problema de alocação de gás de injeção</b>	<b>3</b>
2.1	Ciclo de vida de campos de petróleo . . . . .	3
2.2	Elevação artificial . . . . .	5
2.3	Alocação de gás de injeção . . . . .	9
2.4	Revisão da literatura . . . . .	11
2.5	Sumário . . . . .	13
<b>3</b>	<b>Fundamentos de otimização inteira</b>	<b>14</b>
3.1	Introdução à otimização . . . . .	14
3.2	Programação inteira . . . . .	18
3.2.1	Formulações . . . . .	18
3.2.2	Introdução a algoritmos . . . . .	20
3.3	Métodos de planos de corte . . . . .	26
3.3.1	Desigualdades válidas . . . . .	26
3.3.2	Coberturas para mochila . . . . .	27
3.3.3	<i>Lifting</i> . . . . .	29
3.3.4	<i>Down-lifting</i> . . . . .	30

3.4	O problema da mochila com restrições de precedência . . . . .	32
3.4.1	Definição do problema . . . . .	33
3.4.2	$K$ -coberturas . . . . .	34
3.4.3	<i>Lifting</i> . . . . .	35
3.5	Linearização por partes . . . . .	36
3.5.1	Modelo com pesos por ponto (Modelo I) . . . . .	36
3.5.2	Modelo seqüencial de pesos (Modelo II) . . . . .	38
3.5.3	Modelo de Sherali (Modelo III) . . . . .	40
3.6	Sumário . . . . .	42
<b>4</b>	<b>Modelo linear por partes</b>	<b>44</b>
4.1	Formulação do problema . . . . .	44
4.2	Desigualdades válidas . . . . .	47
4.2.1	Coberturas . . . . .	47
4.2.2	$K$ -Coberturas . . . . .	49
4.3	Fatores de <i>lifting</i> para $K$ -coberturas . . . . .	53
4.4	Fatores de <i>pseudo-lifting</i> para $K$ -coberturas . . . . .	56
4.5	Resultados computacionais . . . . .	62
4.6	Sumário . . . . .	66
<b>5</b>	<b>Interface de otimização</b>	<b>67</b>
5.1	Ambiente de especificação do problema . . . . .	68
5.1.1	Estrutura de dados . . . . .	70
5.1.2	Ajuste de curvas . . . . .	71
5.1.3	Padrão dos arquivos . . . . .	74
5.2	Servidor de otimização . . . . .	76
5.2.1	Gerenciador de instâncias . . . . .	77
5.2.2	Interface . . . . .	78
5.3	Estudo de caso . . . . .	79
5.4	Sumário . . . . .	81



<b>6</b>	<b>Conclusões e trabalhos futuros</b>	<b>83</b>
<b>A</b>	<b>Fatores de <i>lifting</i> exatos para uma floresta</b>	<b>85</b>
<b>B</b>	<b>Modelo linear por partes considerando concavidade</b>	<b>88</b>
B.1	Exemplo ilustrativo . . . . .	89

# Lista de Figuras

2.1	Etapas de produção de um campo de petróleo . . . . .	4
2.2	Poço operado por <i>gas-lift</i> . . . . .	6
2.3	Curvas típicas de <i>gas-lift</i> . . . . .	7
2.4	Linearização por partes da curva de performance do poço . . . . .	8
2.5	Conjunto de poços de petróleo produzindo por <i>gas-lift</i> . . . . .	8
2.6	Exemplo de grafo de precedência de ativação . . . . .	11
3.1	Formulações para $X$ . . . . .	19
3.2	Exemplo ilustrativo do algoritmo de <i>Branch and Bound</i> . . . . .	23
3.3	Ilustração do corte no problema da mochila com $x_1 = x_3 = x_4 = 0$ e $x_6 = x_7 = 1$ . . . . .	25
3.4	Exemplo de restrição de precedência . . . . .	33
3.5	Linearização por partes para o modelo por pesos . . . . .	37
3.6	Linearização por partes para o modelo seqüencial . . . . .	39
3.7	Linearização por partes para o modelo de Sherali . . . . .	41
4.1	Linearização por partes da CPP de um poço $n$ . . . . .	45
4.2	Grafo de restrição de precedência $G = (V, E)$ . . . . .	48
5.1	Ambiente de especificação do problema . . . . .	68
5.2	Editor gráfico . . . . .	69
5.3	Diagrama de classes da estrutura de dados da interface . . . . .	70
5.4	Guia de caracterização de poço com exemplo de ajuste de curva . . . . .	74
5.5	Exemplo do formato de gravação dos dados . . . . .	75

5.6	Exemplo de conexão com o gerenciador de instâncias . . . . .	79
5.7	Interface para o usuário do servidor de otimização . . . . .	80
5.8	Apresentação da solução de um problema . . . . .	81
B.1	Exemplo de CPP considerando concavidade nas variáveis de decisão. . . . .	90

# Lista de Tabelas

2.1	Comparativo entre alguns métodos de elevação artificial . . . . .	5
3.1	Exemplo de instância para o problema da mochila . . . . .	16
3.2	Solução para uma instância do problema da mochila sem restrição de precedência . . . . .	33
4.1	Taxa de injeção para os poços de petróleo . . . . .	48
4.2	Fatores de <i>lifting</i> para o exemplo da mochila . . . . .	55
4.3	Recursos mínimos de ativação . . . . .	60
4.4	Subconjuntos para ativação de $H(C)$ . . . . .	60
4.5	Coefficientes de <i>pseudo-lifting</i> . . . . .	61
4.6	Impacto dos planos de corte por taxa de injeção de gás . . . . .	63
4.7	Impacto dos planos de corte por densidade do grafo . . . . .	64
5.1	Descrição dos campos de petróleo para o estudo de caso . . . . .	80

# Capítulo 1

## Introdução

### 1.1 Problemática

Nos dias atuais o petróleo ainda é a principal fonte de energia em nosso planeta. Apesar de estarem surgindo novas tecnologias, ainda somos bastante dependentes da queima destes hidrocarbonetos. Este recurso ainda existe em grande quantidade em nosso planeta, mas é uma fonte de energia não-renovável e, em um horizonte de tempo não muito longo, este recurso se esgotará. Uma consequência natural é que conforme o extraímos, sua obtenção se torna mais difícil e cara, pois cada vez temos que buscá-lo em regiões mais profundas (poços mais longos) e menos acessíveis (como o fundo do oceano). A necessidade de novas fontes faz com que seja considerada a produção de óleos de difícil extração devido à sua densidade e viscosidade.

É cada vez mais necessário técnicas eficientes para a extração e processamento do petróleo. A otimização entra neste contexto com o objetivo de auxiliar a atingir o desempenho ótimo dos processos existentes.

Várias técnicas de elevação artificial podem ser empregadas para trazer o petróleo de um reservatório à superfície. Uma técnica bastante utilizada no Brasil e no mundo é chamada de *gas-lift*, em que gás é injetado na base do poço, diminuindo a pressão da coluna de fluido que auxilia sua produção natural. O gás utilizado geralmente é o gás produzido pelo poço junto com o petróleo, que é comprimido e reinjetado. O poço possui uma taxa de produção de acordo com a quantidade de gás injetado. A relação entre a quantidade de gás injetada em um poço e sua produção é conhecida como *Curva de Performance do Poço*. Cada poço possui uma taxa de injeção de gás que o levará a uma produção mais lucrativa, assim como existem taxas de injeção para cada um dos poços que levarão o campo de petróleo a obter seu maior lucro. Não necessariamente as taxas que levam o campo a sua condição de maior lucro são as mesmas de cada poço separadamente.

Um dos custos que envolvem a operação de um campo de *gas-lift* é o da compressão do gás. O equipamento de compressão representa o maior custo na operação do campo, portanto é importante que o gás de injeção seja utilizado da maneira mais eficiente possível [1]. Outra situação ocorre em reservatórios maduros, em que o petróleo está terminando e o número de poços é grande. Nestes

campos, não apenas é necessário determinar em que ponto de produção os poços irão operar, mas também deve ser decidido quais poços irão operar. O operador pode desejar priorizar alguns poços, fazendo com que outros só sejam ativados (entrem em operação) se os de maior prioridade já estiverem produzindo. A esta ordem de ativação dos poços, damos o nome de *Restrição de Precedência*.

## 1.2 Objetivos

Nesta dissertação apresentamos uma maneira de obter as taxas de injeção de gás que levem um campo de petróleo com poços operados por *gas-lift* ao ótimo econômico. Consideramos que a capacidade de gás é limitada, não necessariamente existe gás disponível para que todos os poços operem em suas regiões de maior lucratividade. Consideramos também as restrições de precedência, limites mínimos e máximos de operação dos poços e que eles possam ser desativados.

Para o desenvolvimento deste trabalho foi utilizada uma modelagem de otimização baseada em programação linear inteira. Existem algoritmos para modelos desta categoria que em princípio são capazes de gerar uma solução ótima global. Como a curva de performance do poço possui característica não linear, foi feita uma linearização por partes, aproximando a curva por vários segmentos de reta.

Também desenvolvemos uma interface que permite que o usuário caracterize seu campo de petróleo, e invoque automaticamente algoritmos de otimização; não apenas o apresentado neste trabalho como outros. A interface tem como objetivo abstrair do usuário a compreensão do funcionamento dos algoritmos, fazendo com que ele se preocupe apenas com a operação de seu campo de petróleo.

## 1.3 Visão geral da dissertação

Esta dissertação está dividida em seis capítulos. No próximo capítulo apresentamos mais detalhadamente o cenário que pretendemos abordar, com um detalhamento da operação por *gas-lift*, uma formulação geral para o problema e um resumo dos trabalhos existentes nesta área. No Capítulo 3 fazemos uma breve revisão dos fundamentos de otimização inteira, ferramenta que foi utilizada para o desenvolvimento do trabalho. O emprego da teoria de otimização apresentada no problema de alocação ótima de gás de injeção é mostrado no Capítulo 4. No Capítulo 5 apresentamos a ferramenta desenvolvida para auxiliar o operador a resolver problemas desta classe. Por fim, no Capítulo 6 apresentamos as conclusões deste trabalho e perspectivas para trabalhos futuro.

## Capítulo 2

# O problema de alocação de gás de injeção

### 2.1 Ciclo de vida de campos de petróleo

Há vários séculos o petróleo é conhecido e utilizado. Povos antigos, como os Babilônios, Egípcios, Gregos e Romanos o utilizavam para desde a pavimentação de estradas até fins bélicos. Na América, também era conhecido pelos povos Incas e Maias. Naquela época, o petróleo era retirado de exsudações naturais, não sendo necessário a escavação para sua obtenção.

O início da era moderna do petróleo se dá em 1859, quando o primeiro poço foi perfurado. Descobriu-se que este era um negócio bastante lucrativo, pois seus derivados substituíam o óleo de baleia, utilizados na época para iluminação, com custos de produção bastante inferiores. Em seguida a gasolina e o diesel, que eram derivados até então desprezados, passaram a alimentar motores à combustão, adicionando expressivos lucros à atividade.

Em 1900 o método de percussão para perfuração de poços foi substituído pelo rotativo. Neste método uma broca rotativa é utilizada para escavar e perfurar as rochas até o reservatório. Ele permitiu a escavação de poços mais profundos e longos, e é empregado até hoje com algumas variações, podendo chegar a mais de 10.000 metros.

O petróleo tem origem a partir de matéria orgânica depositada junto com sedimentos. A interação destes elementos em condições apropriadas de temperatura e pressão são essenciais para a formação de hidrocarbonetos. O tipo do hidrocarboneto gerado, óleo ou gás, depende da constituição da matéria orgânica original e das condições termoquímicas presentes no processo.

Após serem gerados, os hidrocarbonetos migram de sua rocha geradora por falhas geológicas, até encontrar uma armadilha geológica, uma rocha selante impermeável. A rocha reservatório deve possuir espaços vazios (porosidade) e que eles estejam conectados (permeabilidade).

A descoberta de uma jazida de petróleo envolve um longo período de estudo de dados geofísicos e geológicos. Apenas após estes estudos é proposta a perfuração de um poço pioneiro, que é a etapa que exige mais investimentos do processo de prospecção. Não se pode prever a existência de petróleo, apenas locais mais favoráveis para sua ocorrência.

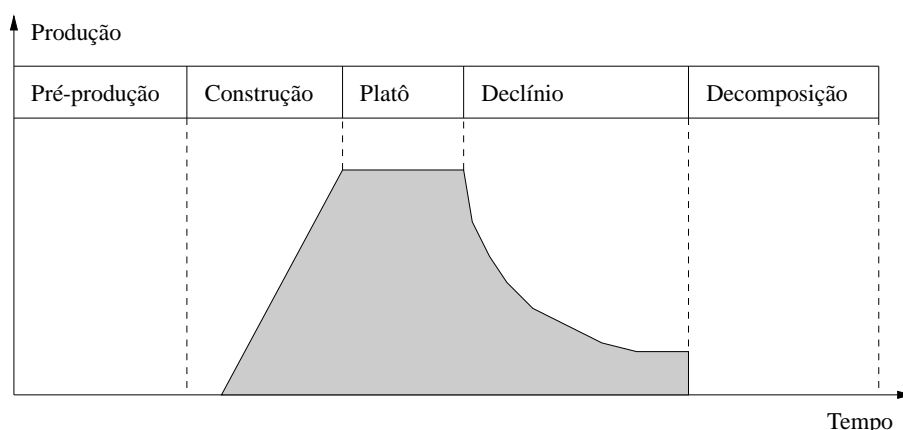


Figura 2.1: Etapas de produção de um campo de petróleo

Os estudos geológicos são realizados por técnicas como a geologia de superfície, que mapeia as rochas da superfície da área em estudo; a aerofotometria e fotogeologia, que constrói mapas geológicos a partir de imagens do terreno; e a geologia de sub-superfície, que consiste no estudo de dados geológicos obtidos em um poço exploratório. Também são empregados métodos sísmicos, que consistem em gerar ondas sísmicas e mapear sua reflexão no tempo e espaço, para o levantamento da estrutura geológica do interior da área em estudo. Mais de 90% dos investimentos em prospecção são aplicados à sísmica de reflexão [2].

Uma vez encontrado o petróleo, ainda é necessário estudos para se levantar o tamanho, forma e produtividade do acúmulo. Conhecendo-se o campo, é feito um estudo de viabilidade de exploração do reservatório. Neste estudo consideram-se a viabilidade econômica, opções de desenvolvimento do campo, equipamentos de produção e logística de evacuação da produção. Em seguida, é realizado um plano de desenvolvimento que serve de base para as ações tomadas sobre o campo. Seu propósito é um projeto conceitual do equipamento a ser empregado e da política de operação para que os objetivos de produção sejam atingidos.

A fase de produção é iniciada quando as primeiras quantidades comerciais de hidrocarbonetos são extraídas. É o momento em que os investimentos começam a ser recuperados. A Figura 2.1 mostra a produção em cada uma das etapas desta fase:

- 1. Construção:** neste período os poços recém perfurados são progressivamente colocados em seus pontos de operação;
- 2. Platô:** novos poços são trazidos aos seus pontos de operação e os antigos começam a declinar, mantendo a produção total constante. Este período é tipicamente de 2 a 5 anos para um campo de óleo e maior para campos de gás [3];
- 3. Declínio:** todos os produtores declinam sua produção.

A vida de um campo normalmente termina quando ele permanentemente gera prejuízos. Ainda é tecnicamente possível continuar a produção, mas economicamente inviável. Neste ponto, a redução do custo de operação pode ser utilizada para aumentar o tempo de vida do campo.



	Haste Flexível	Bomba Elétrica Submersível	Gas-Lift
Instalação	\$	\$\$	\$\$\$
Operação	\$	\$\$	\$\$
Produção (BPD <sup>1</sup> )	1 - 5.000	100 - 60.000	80 - 10.000
Flexibilidade	++	-	++
Viscosidade	+	-	-
Sólidos	-	-	+
Gás	-	-	++
Alta temperatura	+	-	+
Profundidade	-	-	-
Curvas	-	-	+
<i>offshore</i>	--	+	++

Tabela 2.1: Comparativo entre alguns métodos de elevação artificial

## 2.2 Elevação artificial

Quando um novo poço de petróleo é perfurado inicialmente a pressão interna do reservatório é suficiente para vencer a coluna de fluido e o óleo flui naturalmente para a superfície, chamamos estes poços de surgentes. Em decorrência desta extração natural, com o tempo, a pressão interna diminui e se torna necessário algum método de elevação artificial para extrair o óleo para a superfície.

A elevação artificial não é empregada apenas quando o poço deixa de fluir naturalmente. Ela também pode ser usada para aumentar a produção de um poço de baixa vazão natural para um ponto determinado pelo plano de produção. Podem ainda existir casos em que é necessário apenas injetar energia em uma fase inicial, para ativar o poço, que em seguida passa a produzir naturalmente.

Existem vários métodos de elevação artificial. A escolha do método a ser empregado depende de fatores como: viscosidade, presença de sólidos e composição do fluido; temperatura, profundidade e forma do poço (se possui curvas); se o poço está localizado em terra (*onshore*) ou no mar (*offshore*); produtividade, custo de instalação, equipamento auxiliar necessário e presença de infra-estrutura. Pode-se ainda iniciar a extração com um método para poços de maior produtividade e conforme a produtividade do reservatório diminui, fazer a substituição por outro método. Alguns dos métodos disponíveis hoje são mostrados a seguir. A Tabela 2.1 extraída de [3] mostra um comparativo entre eles. O símbolo “\$\$\$” indica que o método possui um custo maior que “\$\$”, que por sua vez é maior que “\$”. Os símbolos “-” à “++” indicam de maneira crescente a adequação do método às características apresentadas.

**Haste Flexível:** uma bomba na forma de um cilindro hidráulico é posicionada no fundo do poço. Seu êmbolo é movido para cima e para baixo por uma haste, que é acoplada a um motor elétrico localizado na superfície. Válvulas no cilindro controlam a entrada e saída do fluido, o bombeando para a superfície.

**Bomba Elétrica Submersível:** é utilizada uma bomba centrífuga de vários estágios, alimentada

<sup>1</sup>BPD - *Barrels Per Day*, Barris por dia.

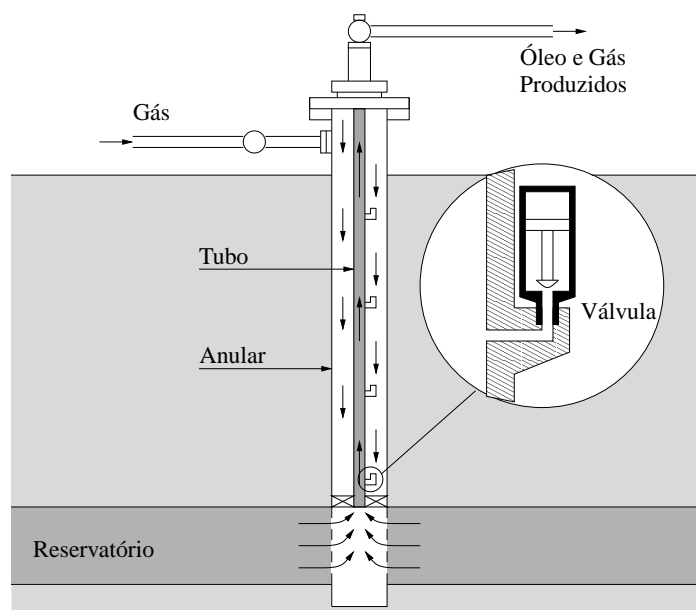


Figura 2.2: Poço operado por *gas-lift*

por um motor elétrico. A produção por este método é também determinada pelo número de estágios que a compõem. Pode-se ainda instalar um variador da velocidade do motor que permite um ajuste na produção do poço, mas isto representa um custo adicional e nem sempre é empregado.

**Gas-Lift:** um dos métodos de elevação artificial mais empregados é o *gas-lift* contínuo [4]. Neste método gás em alta pressão é injetado continuamente no tubo de produção, resultando na redução da densidade da coluna de fluido, e assim reduzindo a diferença de pressão entre o topo e o fundo do poço, estimulando a produção natural.

A Figura 2.2 (retirada de [5]) mostra um poço operando por *gas-lift*. As válvulas localizadas ao longo do tubo de produção são necessárias apenas para o início da operação do poço. A descarga do poço consiste em retirar o fluido presente na coluna de produção e/ou anular para colocar o poço em produção. Inicialmente todas as válvulas estão abertas. Conforme o gás é injetado no anular há uma transferência de fluido dele para dentro do tubo de produção. Quando o nível do fluido no anular baixar da altura da válvula, esta deve ser fechada para que o processo se mantenha nas outras válvulas e o gás não entre diretamente no tubo de produção. O fechamento das válvulas ocorre em seqüência. Quando todo o fluido for expulso da região anular apenas a última válvula se mantém aberta, iniciando a produção do poço. Este procedimento deve ser realizado lentamente para não danificar as válvulas.

Um poço em operação por *gas-lift* tem sua produção controlada pela taxa de injeção de gás aplicada em sua base. A relação entre a vazão de gás injetado e a vazão de saída de fluido  $q_{in} \times q_{out}$  é dada pela curva de performance do poço (CPP).

A forma da curva depende da resposta do poço ao gás injetado. Na Figura 2.3(a), a curva A representa um poço surgente. A injeção de gás em um poço desta categoria aumenta sua produção. A curva B representa um poço que não flui naturalmente, entretanto qualquer que seja a quantidade de gás injetado já coloca o poço em operação. No caso da curva C, é necessário um mínimo de injeção

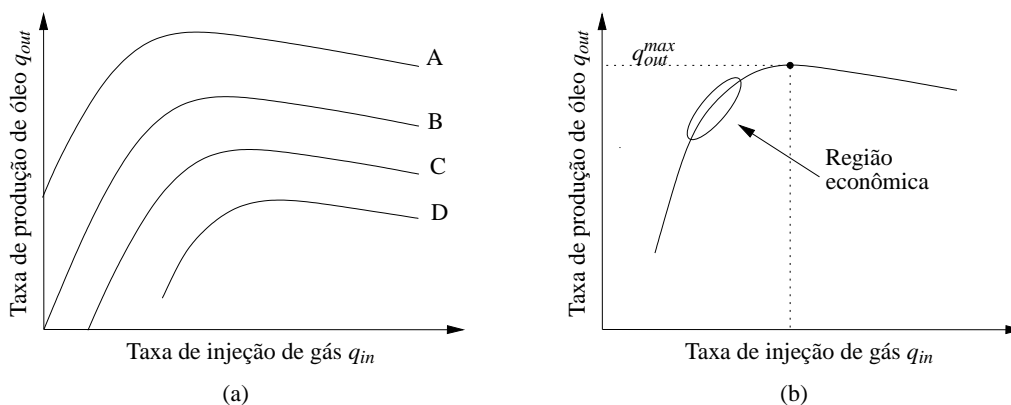


Figura 2.3: Curvas típicas de *gas-lift*

de gás para iniciar a produção. Por fim, a curva D é semelhante a C, mas há um salto na produção inicial para um valor maior que zero.

Destacamos nas curvas C e D a presença de uma quantidade mínima de gás injetado, que ocorre devido a necessidade de vencer a coluna hidrostática que se forma no poço, caracterizando um limite inferior na taxa de injeção. Também observamos que todas as curvas possuem um limite superior de gás que pode ser injetado. Ele não deve ser ultrapassado pois a altas vazões o fluido produzido passa a carregar sólidos junto, que são prejudiciais aos equipamentos.

As CPPs possuem uma forma característica que pode ser observada na Figura 2.3(b). A partir do ponto em que existe produção de fluido há um crescimento rápido até uma vazão máxima. Após, a curva cai lentamente até o limite superior de injeção. A região econômica do gráfico é onde o poço produz com maior lucro. Considerando que não existe restrição de produção, o ponto ótimo é quando o lucro gerado por um incremento de produção seja igual ao incremento no custo de operação [6].

A curva de performance do poço pode ser obtida numericamente por simulações a partir de métodos apropriados de cálculo, como por exemplo a análise nodal [7]. Quando apenas pontos destas curvas são conhecidos ou podem ser obtidos a partir de experimentos de campo, são empregados métodos de ajuste de curvas para o levantamento delas. Para problemas de otimização clássica modelam-se as CPPs como polinômios de terceira ordem. Entretanto, apesar de serem mais facilmente tratáveis, o ajuste em curvas deste tipo não apresenta uma boa correlação. Alarcón *et al.* [8] propõem, em seu trabalho de alocação ótima de *lift-gas* utilizando programação não-linear, que a curva seja representada por um polinômio de segunda ordem, acrescido de um termo logarítmico:

$$q_{out} = c_1 + c_2 q_{in} + c_3 q_{in}^2 + c_4 \ln(q_{in} + 1)$$

onde  $c_1$ ,  $c_2$ ,  $c_3$  e  $c_4$  são as constantes que caracterizam o poço. Outra representação desta curva foi proposta por Nakashima e Camponogara [9], utilizando a composição de duas exponenciais:

$$q_{out} = \alpha_1 (2 - e^{-\beta_1 q_{in}}) - \alpha_2 e^{\beta_2 q_{in}}$$

Estas curvas apresentam melhor correlação que a polinomial pura, sendo que a baseada em exponenciais se mantém fiel também na extrapolação. Uma propriedade interessante da curva exponencial é

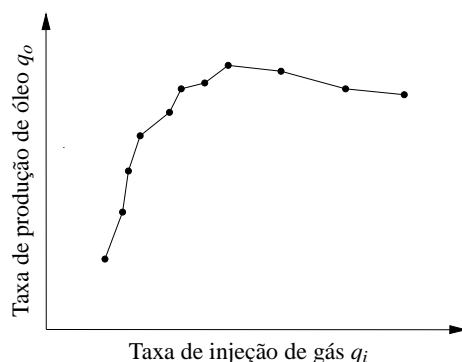


Figura 2.4: Linearização por partes da curva de performance do poço

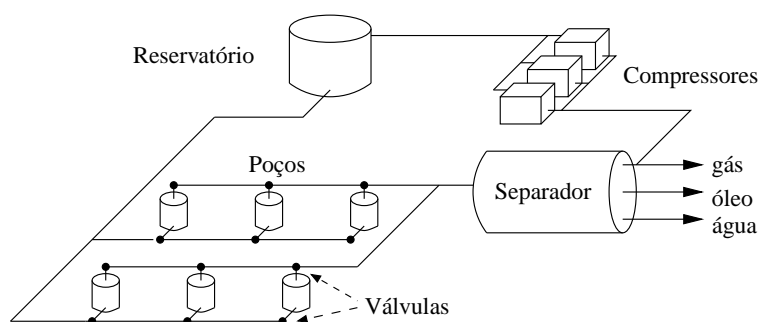


Figura 2.5: Conjunto de poços de petróleo produzindo por *gas-lift*

que  $\frac{d^2 q_{out}}{dq_{in}^2} = -\alpha_1 \beta_1^2 e^{-\beta_1 q_{in}} - \alpha_2 \beta_2^2 e^{-\beta_2 q_{in}}$ . Para que a CPP seja côncava, basta garantir que  $\alpha_1 \geq 0$  e  $\alpha_2 \geq 0$ .

Neste trabalho utilizamos uma abordagem linear por partes. Cada par de pontos consecutivos da curva é interpolado por uma reta como pode ser visto na Figura 2.4. Esta técnica permite que sejam empregados métodos de programação linear. Em contrapartida, faz-se necessário o uso de variáveis de decisão para selecionar a região da curva que está sendo trabalhada. Um aspecto relevante das curvas lineares por partes é que elas permitem utilizar os pontos de teste diretamente, sem a necessidade de ajustar uma curva, o que pode ser tedioso e envolver problemas de otimização não-linear sob restrições. Os testes para obtenção destes pontos são realizados na fase inicial de produção do poço, antes da extração das quantidades comerciais, quando está sendo levantado o perfil de produção do poço e do reservatório. São medidas as quantidades de óleo, gás e água produzidas para um conjunto adequado de taxas de injeção. Uma formulação mais detalhada da linearização por partes para descrever as CPPs será apresentada no Capítulo 4.

Na Figura 2.5 apresentamos um esquema de um campo de petróleo com poços operados por *gas-lift*. Um banco de compressores produz gás a alta pressão, que é armazenado em um reservatório. Válvulas controlam a quantidade de gás injetado em cada poço. O petróleo produzido é levado a um separador que divide o fluido em água, óleo e gás. A água extraída sofre um processo de tratamento para atender determinadas especificações, regulamentadas por órgãos de controle do meio ambiente, limitando a quantidade de poluentes como o teor de óleo e sulfato de hidrogênio antes de ser descartada [2]. O óleo produzido é armazenado e vendido. Uma parcela do gás é comprimida novamente para ser reinjetada nos poços. O excedente pode ser armazenado ou escoado, ou se não existir maneira

de fazê-lo, é queimado.

Uma restrição tipicamente tratada nos problemas de alocação ótima de *gas-lift* é a capacidade total de injeção de gás dos compressores. Nem sempre a planta possui gás a alta pressão suficiente para que todos os poços operem em seus pontos de maior lucratividade. Isto pode ocorrer, por exemplo, porque um compressor do conjunto está desligado para reparos ou manutenção. Outra possibilidade é de um campo com muitos poços, em que o petróleo no reservatório está terminando, devendo-se escolher quais os poços serão ativados.

Na realidade, o cálculo das taxas de injeção ótimas não é a única ferramenta utilizada na determinação dos pontos de operação dos poços. São considerados outros aspectos como, por exemplo, o plano de recuperação do reservatório. O comportamento do acúmulo em longo prazo, sua dinâmica, deve ser considerado para que uma extração inapropriada de alguns poços não deixem regiões do reservatório com óleo acumulado que não possa ser extraído. Assim, restrições de precedência de ativação podem ser utilizadas para representar estes aspectos subjetivos. Elas significam que um dado poço só pode ser ativado se um outro, seu predecessor na restrição, já esteja produzindo pelo menos seu mínimo. É claro que estas restrições só fazem sentido em poços não surgentes.

### 2.3 Alocação de gás de injeção

A alocação ótima de *gas-lift* busca taxas de injeção de gás para cada um dos poços de um campo que maximize ou minimize algum objetivo. Duas abordagens são mais comumente utilizadas: a primeira é baseada na performance hidrodinâmica, onde se buscam pontos de operação para os poços que maximizem a produção total do campo; outra abordagem é baseada em um critério econômico, a qual busca o maior lucro que um conjunto de poços pode fornecer. Ainda são possíveis outras abordagens como, por exemplo, a minimização do custo de operação sujeita a uma quota mínima produção.

Embora neste trabalho utilizamos a abordagem baseada no critério econômico, ela pode ser adaptada facilmente para minimizar os custos de produção da quota mínima. Assim, nossa formulação necessita de informações como custos de produção e lucros obtidos. É importante salientar que estamos interessados na operação do campo de petróleo, por isso questões externas a ele, como situação do mercado, não são consideradas.

Como foi visto na seção anterior, dois tipos de restrição podem ser considerados quando estamos tratando este problema: capacidade máxima de injeção de gás, que é uma restrição decorrente das capacidades dos compressores; e restrições de precedência, que restringe a ativação de um dado poço em decorrência de outros. Considerando nosso objetivo econômico e estas restrições do sistema,

chegamos à seguinte formulação:

$$\begin{aligned}
 P(G) : \text{Maximize } f &= \sum_{n=1}^N (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) q_{out}^n - \sum_{n=1}^N p_{in} q_{in}^n \\
 \text{Sujeito a: } q_{out}^n &= \tilde{q}_{out}^n(q_{in}^n) & n = 1, \dots, N \\
 \sum_{n=1}^N q_{in}^n &\leq q_{in}^{max} \\
 y_n &\leq y_m & \forall (m, n) \in E(G) \\
 l_n y_n &\leq q_{in}^n \leq u_n y_n & n = 1, \dots, N \\
 y_n &\in \{0, 1\} & n = 1, \dots, N
 \end{aligned}$$

onde:

- $N$  é o número de poços que produzem via *gas-lift*;
- $q_{in}^n$  é a taxa de injeção de gás alocada para o poço  $n$ ;
- $q_{out}^n$  é a taxa de produção de fluidos do poço  $n$ , dada pela função  $\tilde{q}_{out}^n(q_{in}^n)$ ;
- $p_o$  e  $p_g$  representam o lucro (preço de venda menos os custos de processamento) obtido por barril de óleo e gás vendido, respectivamente;
- $p_w$  é o custo de tratamento do barril de água a ser descartada;
- $p_{in}$  representa o custo de compressão de gás;
- $\gamma_o^n$ ,  $\gamma_g^n$  e  $\gamma_w^n$  são respectivamente as frações de óleo, gás e água produzidas pelo poço  $n$ , sendo que  $\gamma_o^n + \gamma_g^n + \gamma_w^n = 1$ ;
- $\tilde{q}_{out}^n(q_{in}^n)$  é curva de performance do poço  $n$ , que relaciona a quantidade de gás injetado com a quantidade de fluido produzido no poço;
- $q_{in}^{max}$  é a capacidade total de injeção de gás, sendo que a soma das taxas de injeção de gás dos  $N$  poços não deve ultrapassar esta taxa;
- $y_n$  assume valor 1 se o poço  $n$  está ativo, e 0 caso contrário;
- $G$  é um grafo direcionado acíclico que representa as restrições de precedência. Seus vértices são o conjunto dos  $N$  poços do campo e suas arestas são dadas por  $E(G)$ . A Figura 2.6 apresenta um exemplo de um grafo de precedência. Nela, os poços 1 e 3 podem ser ativados livremente, o poço 2 requer que o de número 1 já esteja ativado pelo menos em seu mínimo, e assim sucessivamente;
- $l_n$  e  $u_n$  são respectivamente as taxas de injeção mínima e máxima para o poço  $n$ .

**Proposição 2.1**  $P(G)$  é NP-Difícil no sentido forte [9].

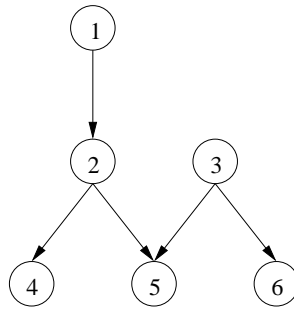


Figura 2.6: Exemplo de grafo de precedência de ativação

A Proposição 2.1 evidencia o grau de dificuldade do problema em consideração. Ou seja, existe um algoritmo de tempo pseudo-polinomial se e somente se a classe  $P$  equivale a classe  $NP$  [10];

## 2.4 Revisão da literatura

Mayhill [11] analisou em seu trabalho a relação entre a taxa de injeção de gás e a produção do poço, chamando a curva resultante de *curva de performance do poço* (CPP).

Desde a década de 70, problemas de injeção ótima de gás vêm sendo abordados. Redden *et al.* [12] reconheceu a importância de uma operação de um campo de *gas-lift* de acordo com critérios econômicos, em que leva em consideração o custo de compressão, capacidade limitada de gás e a possibilidade de compressores estarem inoperantes. Também propuseram um procedimento para encontrar a distribuição de gás de injeção que resulta em um maior lucro baseado na perturbação iterativa da distribuição corrente. Apesar da qualidade da proposta iterativa, o procedimento pode produzir soluções sub-ótimas e não pode ser estendido para tratar outras restrições, como limites inferior e superior de gás de injeção.

Mach *et al.* [7] redefiniram o método de [12] com uma abordagem nodal que considera perdas de pressão em toda a coluna de *gas-lift*. Seu procedimento nodal leva a uma curva de performance do poço mais precisa.

Kanu *et al.* [6] compilou os métodos e princípios anteriores de alocação de *gas-lift* de acordo com critérios econômicos em um método chamado *equal slope*. O método pode obter a alocação de gás que produz maior lucro para um conjunto de poços modelados por CPPs com capacidade de gás limitada. Este método pode ser visto como a aplicação da dualidade Lagrangeana para eliminar a restrição de capacidade de gás. O método possui limitações, incluindo a incapacidade de tratar poços que não respondam instantaneamente à injeção de gás (limite inferior) e a dificuldade de incorporar restrições adicionais.

Mais tarde, Nishikiori *et al.* [13, 14] apresentaram uma formulação não-linear sob restrições para alocação de gás de injeção, com um método quasi-Newton combinado com um descenso pela projeção do gradiente para buscar a solução ótima. O método de Nishikiori *et al.* provou ser melhor que o *equal slope*, obtendo alocação de maior lucro em tempo menor. Apesar da proposta ter imposto

limites não-lineares às taxas de injeção, o método pode divergir e não trata de poços que não produzem espontaneamente.

Buitrago *et al.* [4] estão entre os pioneiros em tratar de cenários com poços que não respondem instantaneamente à injeção de gás. A abordagem, chamada de *Ex-In*, combina uma busca estocástica e uma heurística de cálculo de direção de descenso. Por um lado, o método *Ex-In* utiliza um algoritmo genético, no sentido que uma população de soluções evolui no tempo, mas por outro lado, utiliza uma direção obtida de forma heurística para produzir uma solução candidata. A busca termina quando a população de soluções não produz melhorias na solução objetivo ou após um número máximo de iterações. Apesar da busca *Ex-In* ocorrer sobre o espaço de soluções factíveis, ela não garante que o ótimo global seja alcançado, e não possui um procedimento de obtenção de um limite superior, que poderia ser utilizado para estimar a qualidade de sua solução. Em um sistema com 56 poços, uma comparação computacional com o método *equal slope* mostrou que o *Ex-In* obteve soluções melhores.

Alarcón *et al.* [8] introduziram curvas de performance de poço com termos logarítmicos e polinômios de segunda ordem como um ajuste melhor que os polinômios puros para representar os dados. Eles melhoraram o método de Nishikiori *et al.* [14] substituindo o método quasi-Newton pelo algoritmo SQP (*Sequential Quadratic Programming* - Programação Seqüencial Quadrática) para garantir convergência para uma solução ótima local. O algoritmo SQP é mais robusto e pode operar com restrições adicionais, entretanto, a abordagem requer intervenção dos engenheiros para tratar de poços que não respondem imediatamente à injeção de gás. Assim, os autores propuseram regras *ad hoc* para guiar o engenheiro na decisão de quais poços devem ser ativados e desativados.

Fang e Lo [15] abandonaram os procedimentos baseados em programação não linear e sugeriram a linearização por partes das CPPs, reformulando o problema utilizando programação linear. Esta visão trouxe uma série de vantagens e abriu várias possibilidades. Primeiro, podemos obter soluções ótimas globais com algoritmos poderosos de programação linear, facilmente introduzir restrições adicionais e tratar de problemas muito grandes. Segundo, nos permite utilizar a teoria e algoritmos de programação linear inteira mista (MILP) para tratar de decisões discretas. O método de Fang e Lo superou os métodos anteriores, mas eles não fizeram uso de toda a teoria para tratar problemas de programação inteira mista de maneira sistemática, sugerindo regras heurísticas, baseadas em arredondamento para tratar o problema de ativação de poços.

Wang *et al.* [16] sugeriram um modelo em programação não linear inteira mista (MINLP) para generalizar a abordagem anterior. O modelo proposto incorpora as decisões de alocação de gás de injeção, decisões sobre as taxas de produção e restrições da planta de produção. Como os problemas de programação não linear inteira mista são inerentemente difíceis de serem resolvidos, eles propuseram uma combinação de programação linear, programação separável e algoritmos genéticos. Apesar dos resultados experimentais terem sido satisfatórios, o método pode produzir soluções sub-ótimas e não produz limites superiores/inferiores que permitem avaliar a qualidade das soluções.

Nakashima e Camponogara [9, 17] desenvolveram algoritmos utilizando programação dinâmica (DP) para o problema de alocação de gás de injeção. Seus métodos tratam as decisões discretas de ativação dos poços de maneira sistemática, e consideram incertezas nas CPPs, aceitando múltiplas curvas e resolvendo problemas max-min, resultando em soluções próximas à ótima com garantias de



performance. Junto com o método de MINLP proposto em [16], foi o primeiro a tratar rigorosamente as descontinuidades nas CPPs. O algoritmo de programação dinâmica pode tratar de restrições de precedência de ativação, com uma execução em tempo pseudo-polinomial se as restrições possuírem a forma de uma floresta. Além da inexistência de um algoritmo pseudo-polinomial para tratar de grafos de precedência gerais, a limitação principal da abordagem em programação dinâmica é a dificuldade de adicionar restrições.

## 2.5 Sumário

Neste capítulo apresentamos uma visão geral sobre a formação e operação de um campo de petróleo. Apresentamos alguns dos métodos utilizados para extrair o petróleo para a superfície e, em seguida, nos aprofundamos no método de extração por *gas-lift*, em que gás é injetado na base do poço para diminuir a pressão da coluna e possibilitar que o petróleo seja retirado. Em seguida apresentamos o problema geral de alocação ótima de gás de injeção segundo critérios econômicos, sob restrição de limite de capacidade de injeção e precedência na ativação dos poços. Por fim, apresentamos uma revisão bibliográfica, que contextualiza esta dissertação frente a outros trabalhos já publicados sobre o assunto.

## Capítulo 3

# Fundamentos de otimização inteira

### 3.1 Introdução à otimização

Otimização é a área da matemática aplicada que busca obter valores para variáveis de decisão com o objetivo de obter um comportamento ótimo, sujeito a um conjunto de restrições do modelo matemático. As variáveis de decisão são avaliadas a partir de algum critério, que pode ser a maximização ou minimização de uma função. Os pontos que atendem as restrições do problema são *pontos factíveis*, e seu conjunto é denominado de *região factível* do problema.

Modelos são estruturas que buscam representar alguma característica de uma realidade. As razões que levam ao uso de modelos são as mais variadas. Dentre elas podemos citar: ajudam a compreender melhor a natureza do objeto que está sendo estudado; experimentos realizados sobre o objeto em estudo possuem custos proibitivos ou até são impossíveis; podem-se realizar testes antes que o objeto em estudo seja construído. Modelos podem ser *concretos*, como uma miniatura de avião utilizada em testes em um túnel de vento. Modelos *abstratos* usualmente são representações matemáticas da realidade em estudo. A modelagem de um problema não é uma tarefa trivial, dependendo de fatores subjetivos como intuição, experiência, criatividade e poder de síntese [18, 19].

Para modelar problemas de otimização é utilizada uma linguagem conhecida como *programação matemática*. É importante ressaltar que programação matemática é bastante diferente de programação de computadores. Programação matemática é utilizada no sentido de “planejamento” [18]. Inevitavelmente programação matemática envolve computação, pois problemas práticos normalmente possuem grandes quantidades de dados e cálculos que tornam necessário o emprego dos computadores. Os elementos que a compõem são:

**Variáveis de decisão:** Variáveis cujos valores definem uma solução para o problema;

**Função objetivo:** Expressão que relaciona as variáveis de decisão com o comportamento do modelo.  
É a função que se deseja maximizar ou minimizar;

**Restrições:** Conjunto de funções das variáveis de decisão que definem o espaço de soluções que pode ser aplicado ao problema.

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  $x$  é 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.

Ainda é possível problemas sem função objetivo, onde se deseja apenas encontrar uma solução que atenda as restrições. Por exemplo, montar a tabela de um campeonato de futebol obedecendo a restrições como o limite de partidas consecutivas em um campo. E problemas com múltiplas funções objetivo. Por exemplo, em um problema de manufatura que se deseja maximizar o lucro e minimizar o tempo de produção.

#### Exemplo: O problema da mochila

Um alpinista deseja escolher entre diversos itens quais levar para sua expedição. Os objetos devem ser escolhidos de forma que atenda ao máximo suas necessidades na escalada, sendo que eles não devem exceder a capacidade de sua mochila. Os dados do problema são:

- $N$  é o número de itens disponíveis;
- $c_n$  é um valor que representa a utilidade associada ao item  $n$  para a escalada;
- $b$  é a capacidade total da mochila;
- $a_n$  é o espaço ocupado pelo item  $n$ ;

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 item  $n$  será carregado na mochila, e 0 caso contrário;

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

**Restrições:** O espaço ocupado pelos itens selecionados não deve exceder a capacidade da mochila e as variáveis de decisão devem assumir valores binários do conjunto  $\mathbb{B} = \{0, 1\}$ .

Em programação matemática, o problema fica:

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

$n$	item	utilidade	tamanho
1	Fogareiro	2	5
2	Corda	7	5
3	Colchonete	2	4
4	Ração	4	3
5	Primeiros Socorros	3	2
6	Grampos	7	2
7	Aparelho de GPS <sup>1</sup>	6	1

Tabela 3.1: Exemplo de instância para o problema da mochila

Dado o modelo, chamamos de *instância* o conjunto de dados que caracterizam um problema. Para o exemplo da mochila uma instância é dada na Tabela 3.1. Considerando a capacidade da mochila  $b = 6$ , obtemos a seguinte formulação em programação matemática:

$$\begin{aligned}
 &\text{Maximize } 2x_1 + 7x_2 + 2x_3 + 4x_4 + 3x_5 + 7x_6 + 6x_7 & (3.1) \\
 &\text{Sujeito a: } 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 \leq 6 \\
 & & x \in \mathbb{B}^7
 \end{aligned}$$

Dependendo da natureza da função objetivo e das restrições, os problemas de otimização são classificados em subdomínios.

**Programação Linear:** é uma classe de problemas de otimização onde tanto a função objetivo quanto as restrições são lineares. Matematicamente, esta classe de problemas é representada por:

$$\begin{aligned}
 P_L : & \text{Maximize } c^T x \\
 & \text{Sujeito a: } Ax \leq b \\
 & & x \in \mathbb{R}_+^n
 \end{aligned}$$

onde  $b$  e  $c$  são vetores<sup>2</sup> e  $A$  é uma matriz. Problemas desta natureza são bastante empregados e bem resolvidos algoritmicamente, por meio do algoritmo Simplex e métodos de ponto interior [20].

**Programação Linear Inteira:** possui formulação semelhante à Programação Linear pois a função objetivo e as restrições devem ser lineares. Entretanto, neste sub-domínio da otimização as variáveis de decisão pertencem a um domínio discreto.

$$\begin{aligned}
 P_I : & \text{Maximize } c^T x & (3.2) \\
 & \text{Sujeito a: } Ax \leq b \\
 & & x \in \mathbb{Z}_+^n
 \end{aligned}$$

<sup>1</sup>Global Positioning System, Sistema de Posicionamento Global.

<sup>2</sup>Nesta dissertação utilizamos a notação de matrizes coluna para os vetores.

Quando as variáveis de decisão pertencem ao conjunto binário  $\mathbb{B}^n$ , pode-se dizer que o problema é de Programação Binária. É empregada quando se deseja modelar problemas em que se deve escolher entre uma ou outra alternativa, como no problema da mochila apresentado, tendo que decidir em carregar ou não o objeto; ou em problemas em que as variáveis de decisão devem ser necessariamente inteiras, não tendo sentido assumirem valores fracionários como, por exemplo, em uma montadora de veículos que se deseja determinar a quantidade ótima a ser produzida.

Programação Linear é NP-Difícil, pois qualquer problema da classe NP-Completa pode ser reduzido a  $P_I$  em tempo e espaço polinomial, por exemplo, o problema de satisfatibilidade.

**Programação Linear Inteira Mista:** esta sub-classe dos problemas é uma união das duas apresentadas anteriormente. Nela existem variáveis de decisão contínuas e inteiras.

$$\begin{aligned} P_{IM} : & \text{Maximize } c^T x + h^T y \\ & \text{Sujeito a: } Ax + Gy \leq b \\ & x \in \mathbb{R}_+^n \\ & y \in \mathbb{Z}_+^m \end{aligned}$$

Emprega-se a programação linear inteira mista, ou apenas programação inteira mista, normalmente quando são utilizadas variáveis de decisão sobre variáveis contínuas. Problemas desta subclasse são tão complexos quanto os apresentados anteriormente.

**Programação Não-linear com Restrições:** é o caso mais geral de problemas de otimização.

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

onde  $f(x)$  e  $g(x)$  são não lineares e não precisam ser necessariamente contínuas. Não existe uma forma geral para resolver problemas desta natureza, devendo-se analisar cada caso em busca de um algoritmo eficiente, quando existir.

Existem ainda dois casos particulares: na programação quadrática a função objetivo é uma função quadrática das variáveis de decisão  $f(x) = \frac{1}{2}x^T Qx + c^T x$ , sendo  $Q$  uma matriz simétrica, e as restrições são lineares  $Ax \leq b$ . Se a matriz  $Q$  for positiva definida ou positiva semi-definida encontrar a solução global é relativamente fácil. Mas se  $Q$  for indefinida, negativa definida ou negativa semi-definida o problema já se torna difícil.

Na programação não-linear irrestrita, todas as variáveis de decisão podem assumir qualquer valor. Caso  $f(x)$  seja contínua, pode-se empregar o método de descenso ou o método de Newton. A dificuldade desta subclasse é garantir que o problema convirja para um ponto ótimo global, não apenas sendo ótimo para uma região próxima (ótimo local).

## 3.2 Programação inteira

Nesta seção apresentamos os conceitos básicos para resolução de problemas modelados com Programação Inteira baseados no livro de Wolsey [21]. Tendo em vista que o modelo a ser desenvolvido para  $P(G)$  é tipo inteiro misto, faz-se necessário um maior aprofundamento desta teoria. A teoria aplicada à programação inteira pura também pode ser utilizada na programação inteira mista, tomando alguns cuidados. Podem-se ignorar as variáveis de decisão reais ao aplicar métodos que levam em conta que os números são inteiros, e tratar o problema completo quando são consideradas as relaxações.

### 3.2.1 Formulações

Consideramos agora o problema geral de programação inteira dado por  $P_I$ . Suponha que  $X \subseteq \mathbb{Z}_+^n$  é o conjunto das soluções factíveis. Tipicamente, existem muitas (possivelmente infinitas) formulações para  $X$  dadas por meio de poliedros. Esta equivalência entre elementos de  $X$  e os elementos da sua formulação como um poliedro  $P$  são fundamentais, tanto ao processo de formulação quanto aos procedimentos algorítmicos de solução. Abaixo, formalizamos estas noções e apresentamos conceitos-chave para o desenvolvimento desta dissertação.

**Definição 3.1** *Um subconjunto do  $\mathbb{R}^n$  descrito por um conjunto de restrições lineares  $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$  é um poliedro.*

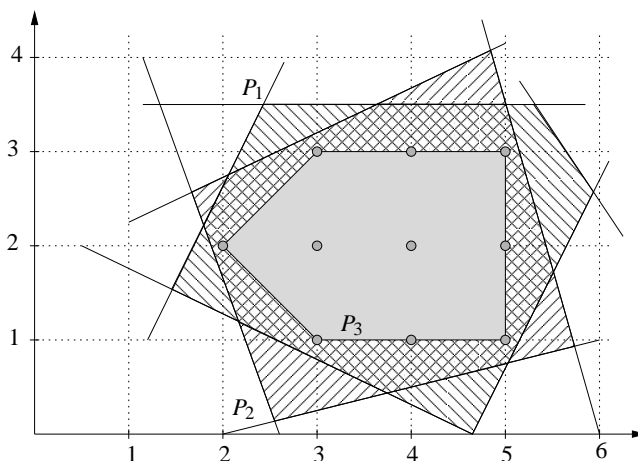
**Definição 3.2** *O poliedro  $P \subseteq \mathbb{R}_+^n$  é uma formulação para o conjunto  $X \subseteq \mathbb{Z}_+^n$  se, e somente se,  $X = P \cap \mathbb{Z}_+^n$ .*

Seja  $X$  o conjunto das soluções factíveis de um problema inteiro dado pelos pontos do conjunto  $X = \{(2,2), (3,1), (3,2), (3,3), (4,1), (4,2), (4,3), (5,1), (5,2), (5,3)\}$ . Então é fácil verificar que os poliedros:

$$P_1 = \left\{ x \in \mathbb{R}_+^2 : \begin{array}{l} -2x_1 + x_2 \leq -\frac{3}{2}, \quad x_2 \leq \frac{7}{2}, \\ \frac{3}{2}x_1 + x_2 \leq \frac{23}{2}, \quad 2x_1 - x_2 \leq \frac{28}{3}, \\ -\frac{1}{2}x_1 - x_2 \leq -\frac{7}{3} \end{array} \right\},$$

$$P_2 = \left\{ x \in \mathbb{R}_+^2 : \begin{array}{l} -\frac{1}{2}x_1 + x_2 \leq \frac{5}{3}, \quad \frac{7}{2}x_1 + x_2 \leq 21, \\ \frac{1}{4}x_1 - x_2 \leq \frac{1}{2}, \quad -\frac{8}{3}x_1 - x_2 \leq -7 \end{array} \right\},$$

$$P_3 = \left\{ x \in \mathbb{R}_+^2 : \begin{array}{l} -x_1 + x_2 \leq 0, \quad x_2 \leq 3, \\ x_1 \leq 5, \quad -x_2 \leq -1, \\ -x_1 - x_2 \leq -4 \end{array} \right\},$$

Figura 3.1: Formulações para  $X$ 

assim como  $P_1 \cap P_2$ , são formulações para  $X$ , como pode ser visto na Figura 3.1.

No exemplo apresentamos três formulações para o mesmo conjunto de pontos  $X$ . Observamos que pode existir um número infinito de formulações para  $X$ . Então, como escolher entre duas formulações? Pode-se dizer que uma formulação é *melhor* que outra? Chegamos então a seguinte definição:

**Definição 3.3** Dado um conjunto de pontos  $X \subset \mathbb{R}^n$ , e duas formulações  $P_a$  e  $P_b$  para  $X$ , dizemos que  $P_a$  é melhor que  $P_b$  se  $P_a \subset P_b$ .

Podemos dizer ainda que uma formulação melhor é *mais forte* ou *apertada* que outra. Evidentemente, nem todas as formulações podem ser comparadas. Como vemos as formulações  $P_1$  e  $P_2$  são *não comparáveis*, pois  $P_1 \not\subset P_2$  nem  $P_2 \not\subset P_1$ .

A formulação  $P_3$  é dita *ideal* porque não existe uma formulação  $P_4$  para o conjunto  $X$  tal que  $P_4 \subset P_3$ . Também porque se resolvermos a relaxação linear de  $P_3$  a solução do problema está localizada sobre um dos vértices de  $P_3$ . A seguir, formalizamos estes conceitos.

**Definição 3.4** Dado um conjunto de pontos  $X$ ,  $x$  é combinação convexa de  $X$  se existe um conjunto finito de pontos  $x_j \in X$  e  $\lambda_j \in \mathbb{R}_+$ ,  $j = 1, \dots, n$ , com  $\sum_{j=1}^n \lambda_j = 1$  tal que  $x = \sum_{j=1}^n \lambda_j x_j$ .

**Definição 3.5** Dado um conjunto de pontos  $X$ ,  $\text{conv}(X) = \{x : x \text{ é combinação convexa de } X\}$  é o fecho convexo do conjunto  $X$ .

**Proposição 3.1**  $\text{conv}(X)$  é um poliedro [22, página 106, Teorema 6.2].

**Proposição 3.2** Os vértices de  $\text{conv}(X)$  são pontos de  $X$  [21, página 15, Proposição 1.2].

Como consequência temos que  $\text{conv}(X)$  pode ser descrito por um conjunto finito de desigualdades lineares, e  $P_I : \{\max c^T x : x \in X\}$  pode ser reescrito como  $\{\max c^T x : x \in \text{conv}(X)\}$ . Entretanto, encontrar todas as desigualdades que descrevem  $\text{conv}(X)$  nem sempre é fácil, pois pode existir um número muito grande (exponencial) de desigualdades. No exemplo apresentado na Figura 3.1 podemos verificar que para o conjunto de pontos  $X$ , a formulação  $P_3 = \text{conv}(X)$ .

### 3.2.2 Introdução a algoritmos

Formulado o problema, o passo seguinte é encontrar uma maneira eficiente de encontrar a solução ótima. Como já foi citado, problemas de programação linear não inteira possuem algoritmos eficientes para obtenção da solução ótima, como os métodos Simplex [20] e ponto interior [23]. Porém no caso da programação inteira, por pertencerem à classe NP-difícil, não existem algoritmos gerais capazes de encontrar a solução ótima global de maneira eficiente (a não ser que NP=P).

Algoritmos exatos são métodos capazes de encontrar a solução ótima global para um problema de otimização. Nem sempre o seu uso é possível, geralmente devido ao elevado tempo e/ou grande quantidade de recursos computacionais demandados pelo algoritmo. Em casos onde uma solução relativamente boa é admissível, podem-se empregar heurísticas e meta-heurísticas. Estes métodos não garantem uma solução ótima para o problema, mas são capazes de encontrar uma solução de qualidade em um tempo adequado para aplicação.

#### 3.2.2.1 Otimalidade e relaxação

Dado um problema em programação linear inteira  $z = \max\{c^T x : x \in X\}$ , como provar que um ponto  $x^*$  é ótimo? Uma maneira é encontrar um limite inferior à solução  $\underline{z} \leq z$  e um limite superior  $\bar{z} \geq z$  de tal forma que  $\underline{z} = \bar{z} = z$ . Esta abordagem propõe um algoritmo para a solução desta classe de problemas. Deve-se buscar uma seqüência decrescente de limites superiores:

$$\bar{z}_1 > \bar{z}_2 > \bar{z}_3 > \dots > \bar{z}_s \geq z$$

e uma seqüência crescente de limites inferiores:

$$\underline{z}_1 < \underline{z}_2 < \underline{z}_3 < \dots < \underline{z}_t \leq z$$

até que seja atingido:

$$\bar{z}_s - \underline{z}_t \leq \epsilon$$

onde  $\epsilon$  é um valor positivo muito pequeno. Então temos que encontrar maneiras de obter estes limites.

**Limite Primal:** Toda solução factível  $x^* \in X$  induz um limite inferior  $\underline{z} = c^T x^* \leq z$ . Geralmente encontrar uma solução factível para um problema não é muito difícil. O desafio é encontrar uma boa solução, isto é, que seja próxima da solução ótima.

**Limite Dual:** O limite dual procura um limite superior para os problemas de maximização. A abordagem mais importante é a *relaxação*, que busca tornar o problema mais fácil de ser resolvido em que a solução ótima seja no mínimo tão grande quanto  $z$ . Na relaxação deve-se balancear a dificuldade de obtê-la com os benefícios que ela trará. Uma relaxação muito apertada pode requerer um esforço computacional grande, enquanto uma relaxação mais facilmente computável pode resultar em valores muito distantes. Uma relaxação bastante empregada é a *relaxação linear*.



**Definição 3.6** Para o problema em programação linear inteira  $P_I = \max\{c^T x : x \in P \cap \mathbb{Z}^n\}$  com formulação  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ , sua relaxação linear é  $P_R = \max\{c^T x : x \in P\}$ .

Ou seja, a relaxação linear de um problema em programação linear inteira consiste em transformar suas variáveis de decisão inteiras em reais, transformando o problema para programação linear apenas, que pode ser facilmente resolvido.

Nos problemas de minimização o limite primal representa um limite superior na função objetivo, e o limite dual um limite inferior. A seguir, vamos apresentar alguns algoritmos exatos para resolução de problemas de otimização.

### 3.2.2.2 Branch and Bound

O algoritmo de *Branch and Bound* baseia-se em dividir o problema em problemas menores e mais fáceis de serem resolvidos, e depois unir as soluções para formar a solução do problema original.

Uma maneira de se representar esta divisão seria fixar os valores de todas as variáveis inteiras, enumerando todas as possíveis soluções para o problema. Em seguida obter o valor da função objetivo para cada uma das soluções e escolher a ótima. Esta abordagem é chamada de *enumeração explícita* e não é uma boa maneira de se resolver o problema, pois o número de soluções possíveis cresce exponencialmente com o número de variáveis de decisão, tornando impraticável já para problemas de tamanho médio.

No algoritmo de *Branch and Bound*, para cada decomposição do problema é avaliado se a árvore de soluções factíveis pode ser reduzida. Esta abordagem é chamada de *enumeração implícita*, pois descarta maus ramos da árvore de soluções. A seguir é apresentado o algoritmo em pseudocódigo para um problema de maximização:

```

Branch-And-Bound(  $c, P$  )
1    $L \leftarrow \{P\}$            ▷ lista de nós ativos
2    $T \leftarrow (\{P\}, \emptyset)$    ▷ árvore inicial de Branch and Bound
3    $\underline{z} \leftarrow -\infty$      ▷ limite inferior
4    $x^* \leftarrow \text{NIL}$          ▷ a solução do problema
5   enquanto  $L \neq \emptyset$ 
6     faça escolha um  $P_k$  de  $L$ 
7       resolver relaxação  $RL_k \leftarrow \{\max c^T x : x \in P_k\}$ 
8        $\bar{z}^k \leftarrow \text{valorObjetivo}[RL_k]$ 
9        $x^k \leftarrow \text{solução}[RL_k]$ 
10      se  $\bar{z}^k = -\infty$            ▷  $P_k = \emptyset$ 
11          então descarte  $P_k$  por infactibilidade
12      senão se  $\bar{z}^k \leq \underline{z}$ 
13          então descarte  $P_k$  por limite

```

14           **senão**  
 15            **se**  $x^k \in \mathbb{Z}^{|x^k|}$  e  $\bar{z}^k > \underline{z}$   
 16            **então**  $\underline{z} \leftarrow \bar{z}^k$             $\triangleright$  novo limite inferior  
 17             $x^* \leftarrow x^k$             $\triangleright$  nova solução  
 18           **senão**            $\triangleright x^k \notin \mathbb{Z}^{|x^k|}$   
 19            escolha uma variável fracionária  $x_j^k$  da solução  
 20             $P_{k1} \leftarrow P_k \cap \{x_j \leq \lfloor x_j^k \rfloor\}$   
 21             $P_{k2} \leftarrow P_k \cap \{x_j \geq \lceil x_j^k \rceil\}$   
 22            adicione  $P_{k1}$  e  $P_{k2}$  a  $T$  como folhas de  $P_k$   
 23            insira  $P_{k1}$  e  $P_{k2}$  em  $L$   
 24            descarte  $P_k$

Nas primeiras quatro linhas do algoritmo é criada uma lista de nós ativos, uma árvore de soluções, um vetor para armazenar o valor das variáveis de decisão da melhor solução encontrada e uma variável para armazenar o valor da função objetivo correspondente. Na prática, não é necessário armazenar a árvore de soluções, apenas a lista dos nós ativos, mas para compreensão do algoritmo esta informação é mantida. Em seguida, o algoritmo entra em um laço que é executado para cada um dos elementos da lista de nós ativos. Na primeira iteração existe apenas o nó que representa o problema completo.

A escolha do nó ativo para ser processado pode ser feita de várias maneiras. Heurísticas buscam o nó que seja o melhor candidato a levar à solução do problema o mais rápido possível. Deseja-se o nó que leve a um alto limite inferior, que reduza ao máximo a árvore de soluções e por conseqüência o tempo de processamento. Outra abordagem pode ser uma busca em profundidade. Esta heurística é menos elaborada que as adotadas como padrão em implementações comerciais, implicando em uma execução mais lenta (maior número de iterações), mas ela mantém o número de nós ativos pequeno, sendo indicada quando há pouca memória para executar o algoritmo. Uma abordagem comercial consiste em inicialmente utilizar uma busca em profundidade, que leva rapidamente a uma solução inteira (não necessariamente boa), e em seguida partir para uma heurística mais elaborada. A presença de uma solução inteira permite que nós sejam descartados rapidamente por limite.

Escolhido um nó é resolvida sua relaxação, que como vimos geralmente é utilizada a relaxação linear. O nó pode ser descartado por infactibilidade se seu conjunto de soluções é vazio, ou por limite se seu limite superior apresentar valor na relaxação menor que o melhor limite inferior já obtido. Então são verificados os valores obtidos nas variáveis de decisão. Se elas obtiveram valores inteiros, o limite superior do nó também é seu limite inferior, e uma nova melhor solução pode ter sido encontrada. Caso contrário é escolhida uma das variáveis de decisão e o problema é dividido em dois sub-problemas (ou até em mais, em implementações mais avançadas), que são adicionados à lista de nós ativos e à árvore de soluções.

Da mesma forma, não existe uma melhor forma de escolher a variável de decisão que será usada para dividir o problema. Geralmente é empregada a *mais fracionária* das variáveis, ou seja, a variável que possui valor mais distante do seu inteiro mais próximo. Por exemplo, considere duas variáveis de decisão  $y_1 = 1,4$  e  $y_2 = 2,8$ , então é escolhida a variável  $y_1$ , pois 1,4 está mais distante de 1 (seu inteiro mais próximo) que 2,8 está de 3.

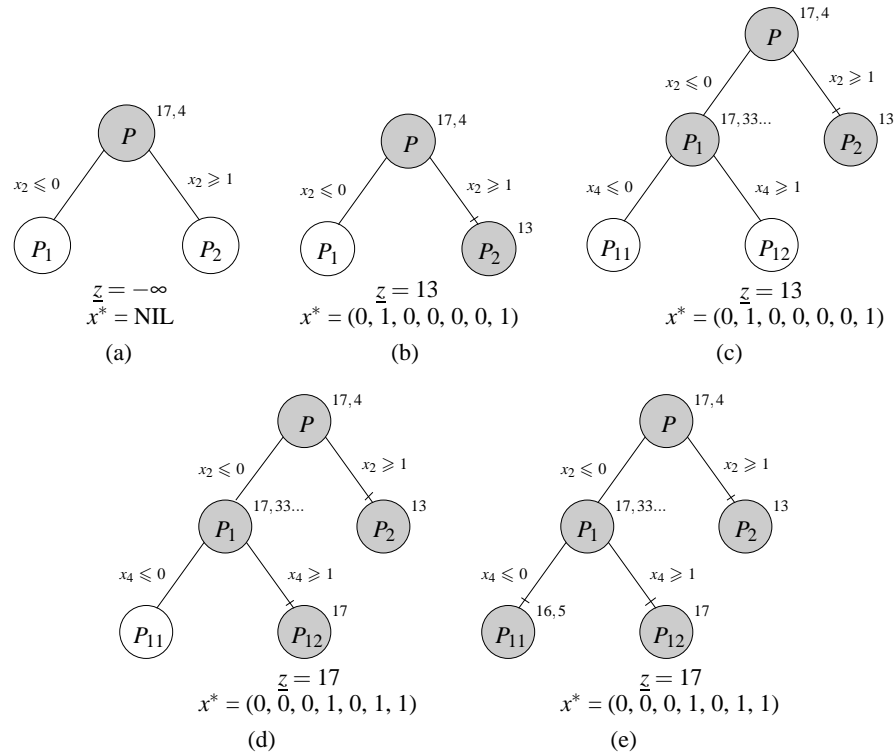


Figura 3.2: Exemplo ilustrativo do algoritmo de *Branch and Bound*

Dado o exemplo do problema da mochila apresentado na Equação 3.1, a Figura 3.2 apresenta a aplicação do algoritmo. Na figura, os nós escuros já foram processados e os nós claros compõem a lista dos nós ativos.

1. Em (a) foi resolvida a relaxação linear do problema original, chegando a solução  $x = (0, \frac{1}{5}, 0, 0, 1, 1, 1)$  com limite superior igual a 17,4. Como  $x_2$  apresentou valor fracionário, o problema foi dividido em dois conjuntos de soluções sobre  $x_2$ .
2. Em (b) foi escolhido o nó do conjunto de soluções  $P_2$ , resolvida sua relaxação obtendo a solução inteira  $x^2 = (0, 1, 0, 0, 0, 0, 1)$  com limite superior 13. Note que o limite inferior passa a ser 13, e o nó é fechado por otimalidade.
3. A parte (c) da figura mostra a resolução do nó  $P_1$ , que obteve solução  $x^1 = (0, 0, 0, \frac{1}{3}, 1, 1, 1)$  e valor da relaxação  $\bar{z}^1 = 17\frac{1}{3}$ . A variável  $x_4^1 = \frac{1}{3}$  apresentou valor fracionário neste subproblema. Então este foi dividido em torno de  $x_4$ , gerando novos subconjuntos de solução  $P_{11}$  e  $P_{12}$ .
4. (d) mostra a resolução da relaxação de  $P_{12}$ , com  $x^{12} = (0, 0, 0, 1, 0, 1, 1)$  e  $\bar{z}^{12} = 17$ . Observe que novamente a solução foi inteira. Como sua solução é maior que a até então obtida, esta passa ser a solução do problema.
5. Em (e) resolvemos  $RL_{11}$ , com solução  $x^{11} = (0, 0, \frac{1}{4}, 0, 1, 1, 1)$  e  $\bar{z}^{11} = 16,5$ . Como o limite superior é menor que  $\bar{z}$ , isto é, a solução deste ramo da árvore não pode ser melhor que a já obtida, o nó é cortado por limite.

6. Como não existem mais nós ativos, a solução  $z^* = 17$  e  $x^* = (0, 0, 0, 1, 0, 1, 1)$  é o ótimo global do problema.

### 3.2.2.3 Planos de corte

Considere a restrição do problema da mochila em (3.1),

$$5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 \leq 6 \quad (3.3)$$

$x \in \mathbb{B}^7$ . Podemos observar que  $x_4, x_5$  e  $x_6$  não podem assumir valor 1 simultaneamente, pois se  $x_4 = x_5 = x_6 = 1$ , temos que  $3x_4 + 2x_5 + 2x_6 = 7 > 6$ , que excede a capacidade da mochila. Mas se eles forem agrupados dois a dois, a restrição continua válida. Isto sugere a restrição  $x_4 + x_5 + x_6 \leq 2$ . Na mesma linha de raciocínio, observamos ainda que se  $x_2 = 1$ ,  $x_4, x_5$  e  $x_6$  devem ser 0. Assim, podemos complementar a restrição obtendo

$$2x_2 + x_4 + x_5 + x_6 \leq 2. \quad (3.4)$$

Esta desigualdade é válida para a instância apresentada, pois ela é verdadeira para todos os pontos que compõem o conjunto de soluções  $X$  da instância.

**Definição 3.7** Uma desigualdade  $\pi^T x \leq \pi_0$  é uma desigualdade válida para um conjunto  $X \subseteq \mathbb{R}^n$  se  $\pi^T x \leq \pi_0$  para todo  $x \in X$ .

Em um problema em programação inteira de espaço de soluções  $X$ , a inserção de desigualdades válidas pode gerar uma redução no poliedro da formulação, ou seja,  $P \cap \{x : \pi^T x \leq \pi_0\} \subset P$ . O algoritmo de planos de corte consiste em inserir cortes válidos ao poliedro até que a resolução da relaxação linear do problema seja inteira, que implica na obtenção do ótimo global. Este procedimento não requer a obtenção das expressões que produzem o fecho convexo de todo o poliedro, apenas a sua redução de maneira apropriada. A seguir, apresentamos o algoritmo de planos de corte.

Planos-De-Corte(  $c, P$  )

- 1  $t \leftarrow 0$     ▷ contador
- 2  $P^0 \leftarrow P$     ▷ poliedro inicial
- 3 **faça sempre**
- 4     resolver relaxação  $RL_t \leftarrow \{\max c^T x : x \in P^t\}$
- 5      $x^t \leftarrow \text{solução}[RL_t]$
- 6     se  $x^t \in \mathbb{Z}^{|x^t|}$
- 7         **então retorne**  $x^t$     ▷  $x^t$  é solução ótima
- 8     busque um corte  $(\pi^t, \pi_0^t)$  para  $x^t$
- 9     se  $(\pi^t, \pi_0^t)$  corta  $x^t$
- 10         **então**  $P^{t+1} \leftarrow P^t \cap \{x : (\pi^t)^T x \leq \pi_0^t\}$
- 11          $t \leftarrow t + 1$

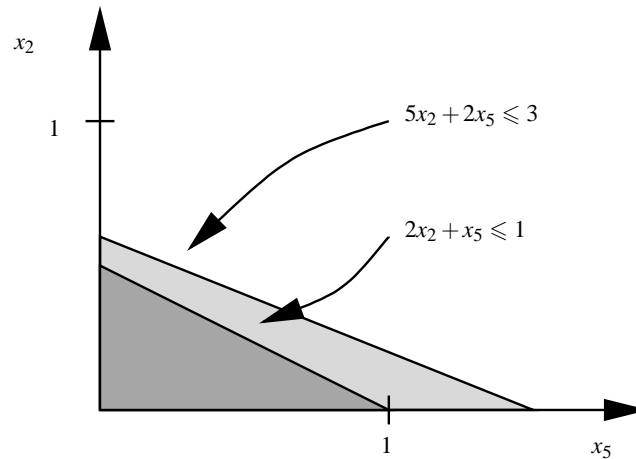


Figura 3.3: Ilustração do corte no problema da mochila com  $x_1 = x_3 = x_4 = 0$  e  $x_6 = x_7 = 1$

12            **senão retorne NIL**            ▷ não encontrou solução

Podemos observar que este algoritmo busca cortes para o poliedro até que seja encontrada uma solução inteira, que é o ótimo global. A obtenção dos cortes (linha 8) é a base do algoritmo. Como é uma tarefa difícil, nem sempre são encontrados cortes até resolver o problema, assim o algoritmo pode retornar uma falha (linha 12).

Existem procedimentos gerais para a obtenção de desigualdades válidas para problemas inteiros, como o de Chavátal-Gomory [21, 22]. Para a programação inteira pura os cortes de Chavátal-Gomory fazem que o algoritmo de planos de cortes convirja em um número finito de iterações. Para problemas binários com formulação  $P = \{x \in \mathbb{R}_+^n : Ax \leq b, x \leq 1\}$  e  $S = P \cap \mathbb{Z}^n$ , o procedimento de cortes disjuntivos pode ser utilizado para obter  $\text{conv}(S)$  após  $n$  iterações.

A obtenção dos cortes geralmente faz uso da estrutura da formulação. No exemplo acima aplicamos parcialmente um procedimento chamado de *coberturas* que busca cortes para o problema da mochila, e será visto mais detalhadamente na próxima seção. Com a adição desta nova restrição convenientemente escolhida ao problema, sua relaxação linear passa a apresentar  $x^T = (0, 0, 0, 1, 0, 1, 1)$  em suas variáveis de decisão, contra  $x^T = (0, \frac{1}{5}, 0, 0, 1, 1, 1)$  que é o valor obtido na formulação original. Uma vez que  $x \in \mathbb{Z}_+^7$  na nova formulação, esta solução é ótima global. Note que ao introduzirmos a restrição (3.4) a  $P = \{x \in \mathbb{B}^7 : \text{sujeito a (3.3)}\}$ , não obtemos o fecho convexo  $\text{conv}(S)$ ,  $S = P \cap \mathbb{Z}^7$ , mas uma configuração exata em torno da solução ótima.

A Figura 3.3 ilustra a inserção do corte no subespaço  $x_5 \times x_2$ , fixando  $x_1 = x_3 = x_4 = 0$  e  $x_6 = x_7 = 1$ . Neste subespaço (3.3) fica  $5x_2 + 2x_5 \leq 3$  e o corte (3.4) fica  $2x_2 + x_5 \leq 1$ . Podemos observar que a inserção do corte reduz o poliedro, mas que não gera o poliedro mínimo ( $x_2 \leq 0$  e  $x_5 \leq 1$  para este sub-espço). Mesmo assim, este corte leva à solução problema.

### 3.2.2.4 Branch and Cut

A inserção pura de cortes não é a melhor maneira de se resolver um problema de otimização. A obtenção eficiente de cortes é uma tarefa difícil. Uma maneira alternativa de resolver os problemas de otimização é combinar algoritmos conhecidos. O algoritmo de *Branch and Cut* é a combinação dos dois algoritmos vistos anteriormente. Para cada nó da árvore de soluções do *Branch and Bound* são inseridos cortes. Em cada nó busca-se o máximo de cortes que podem ser inseridos em um tempo razoável, reduzindo assim o limite superior do nó e promovendo uma redução no tamanho da árvore *Branch and Bound*.

É necessário, entretanto limitar o tempo de busca por cortes em cada nó. Um excesso de empenho do algoritmo na busca de cortes pode representar uma redução pequena na relaxação, de maneira que este tempo não seja compensado na redução do número de nós.

## 3.3 Métodos de planos de corte

Nesta seção formalizaremos os conceitos de aplicação de cortes a uma formulação e como gerar cortes para o problema da mochila.

### 3.3.1 Desigualdades válidas

Vimos na Definição 3.7 o que é uma desigualdade válida. Um poliedro possui infinitas desigualdades válidas, surgindo a questão de quais desigualdades são relevantes e quais podem ser descartadas.

**Definição 3.8** *Dadas duas desigualdades válidas  $\pi^T x \leq \pi_0$  e  $\mu^T x \leq \mu_0$  para um poliedro  $P \subseteq \mathbb{R}_+^n$ , também notadas por  $(\pi, \pi_0)$  e  $(\mu, \mu_0)$ , dizemos que  $\pi^T x \leq \pi_0$  domina  $\mu^T x \leq \mu_0$  se existe  $u > 0$  tal que  $\pi \geq u\mu$  e  $\pi_0 \leq u\mu_0$ , e  $(\pi, \pi_0) \neq (\mu, \mu_0)$ .*

Note que se  $(\pi, \pi_0)$  domina  $(\mu, \mu_0)$ , denotado por  $(\pi, \pi_0) \prec (\mu, \mu_0)$ , então  $(\mu, \mu_0)$  é redundante e pode ser descartada da formulação.

**Definição 3.9** *Considerando um poliedro  $P \subseteq \mathbb{R}_+^n$ , uma desigualdade válida  $\pi^T x \leq \pi_0$  é redundante para  $P$  se existirem desigualdades  $(\mu^i)^T x \leq \mu_0^i$ ,  $i = 1, \dots, k$ , de  $P$  e pesos  $u_i > 0$  tal que  $(\sum_{i=1}^k u_i \mu^i)^T x \leq \sum_{i=1}^k u_i \mu_0^i$  domine a desigualdade  $\pi^T x \leq \pi_0$ .*

Desejamos trabalhar com as desigualdades necessárias e suficientes para definição do poliedro, chamadas de *desigualdades válidas fortes*, ou apenas *desigualdades fortes*. São fortes no sentido que levam a uma formulação mais apertada. Para o entendimento mais preciso de desigualdades fortes, alguns conceitos serão apresentados.

**Definição 3.10** *Os pontos  $x^1, \dots, x^k \in \mathbb{R}^n$  são ditos afim independentes se a única solução  $u_i$ ,  $i = 1, \dots, k$ , para  $\sum_{i=1}^k u_i x^i = 0$  e  $\sum_{i=1}^k u_i = 0$  for nula.*

Dizer que pontos  $x^1, \dots, x^k \in \mathbb{R}^n$  são afim independentes é equivalente a dizer que os pontos  $(x^2 - x^1), \dots, (x^k - x^1)$  são linearmente independentes. É fácil observar que o maior número de pontos afim independentes em  $\mathbb{R}^n$  é  $n + 1$ , por exemplo,  $n$  pontos linearmente independentes e o vetor zero.

**Definição 3.11** *Se para um poliedro  $P$  o número máximo de pontos afim independentes é  $k$ , então a dimensão do poliedro  $P$ , denotada por  $\dim(P)$ , é  $k - 1$ .*

**Definição 3.12** *Um poliedro  $P \subseteq \mathbb{R}^n$  é de dimensão cheia se, e somente se,  $\dim(P) = n$ .*

Dizer que um poliedro  $P \subseteq \mathbb{R}^n$  tem dimensão cheia significa dizer que não existem variáveis de decisão que podem ser expressas como função de outras variáveis. Não existem igualdades que são satisfeitas por todo  $x \in P$ .

**Definição 3.13** *Uma face de um poliedro  $P$  induzida por uma desigualdade válida  $\pi^T x \leq \pi_0$  é  $F = \{x \in P : \pi^T x = \pi_0\}$ . A desigualdade  $\pi^T x \leq \pi_0$  define ou representa a face.*

**Definição 3.14** *Uma face  $F$  é uma faceta de  $P$  se  $\dim(F) = \dim(P) - 1$ .*

Considerando o poliedro do problema da mochila em (3.1)

$$P = \{x \in \mathbb{R}_+^7 : 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 \leq 6, \\ x_j \leq 1, j = 1, \dots, 7\}. \quad (3.5)$$

Seja  $e_j \in \mathbb{R}_+^7$  o vetor com todas as entradas nulas a exceção da  $j$ -ésima entrada, cujo valor é unitário. Note que  $e_j \in P$  para  $j = 1, \dots, 7$ . Logo  $\{e_j\}_{j=1}^7 \cup \{0\}$  constitui um conjunto de pontos afim independentes de  $P$  e, portanto,  $\dim(P) = 7$ .

Considere agora a face representada pela desigualdade da mochila  $F = \{x \in P : 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 + x_7 = 6\}$ . Tomando os pontos  $x^1 = (1, 0, 0, 0, 0, 0, 1)$ ,  $x^2 = (0, 1, 0, 0, 0, 0, 1)$ ,  $x^3 = (0, 0, 1, 0, 1, 0, 0)$ ,  $x^4 = (0, 0, 1, 0, 0, 1, 0)$ ,  $x^5 = (0, 0, 0, 1, 1, 0, 1)$ ,  $x^6 = (\frac{3}{5}, \frac{3}{5}, 0, 0, 0, 0, 0)$  e  $x^7 = (0, 0, \frac{3}{4}, 1, 0, 0, 0)$ , podemos verificar que  $x^j \in F$ ,  $j = 1, \dots, 7$ , são linearmente independentes e que estes pontos formam um conjunto de pontos afim independentes (independência linear implica em independência afim, mas o contrário não é verdade). Concluimos que  $\dim(F) = 7 - 1 = 6$  e como  $\dim(F) = \dim(P) - 1$  verificamos que  $F$  é uma faceta de  $P$ , ou seja, face de dimensão máxima.

### 3.3.2 Coberturas para mochila

Na Seção 3.2.2.3 vimos informalmente a aplicação de coberturas para a geração de cortes no problema da mochila. Nesta seção e nas próximas estes conceitos serão generalizados. Iniciamos apresentando algumas definições.

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 3.15** *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.*

Uma cobertura assinala um conjunto de elementos que excede a capacidade da mochila. Se o conjunto obtido pela remoção de qualquer elemento da cobertura pode ser armazenado na mochila, dizemos que a cobertura é mínima. Para (3.3),  $C = \{4, 5, 6\}$  é uma cobertura, pois  $a_4 + a_5 + a_6 = 3 + 2 + 2 = 7 > 6$ . Esta cobertura também é mínima, pois  $a_4 + a_5 = 5 \leq 6$ ,  $a_4 + a_6 = 5 \leq 6$  e  $a_5 + a_6 = 4 \leq 6$ .

**Proposição 3.3** *Dado uma cobertura  $C \subseteq N$ , a desigualdade*

$$\sum_{j \in C} x_j \leq |C| - 1 \quad (3.6)$$

*é válida para  $X$ .*

Temos então para a cobertura  $C = \{4, 5, 6\}$  a desigualdade válida  $x_4 + x_5 + x_6 \leq 2$ . Definimos  $P_C = \text{conv}(X_C)$ , de  $X_C = \{x \in \mathbb{B}^{|C|} : \sum_{j \in C} a_j x_j \leq b\}$ , como o poliedro obtido a partir da projeção de  $\text{conv}(X)$  sobre o espaço das variáveis da cobertura. Podemos facilmente verificar que  $\dim(P_C) = |C|$ , ou seja,  $P_C$  tem dimensão cheia. Definimos  $F_C = \{x \in P_C : \sum_{j \in C} x_j = |C| - 1\}$  como a face de  $P_C$  induzida por (3.6), podemos verificar que  $F_C$  é uma faceta, ou seja,  $\dim(F_C) = \dim(P_C) - 1$ . Para o exemplo dado acima,  $\dim(P_C) = 3$  e  $\{(1, 1, 0), (1, 0, 1), (0, 1, 1)\} \subseteq F_C$  formam um conjunto afim independente que nos leva a concluir que  $\dim(F_C) = 2$ .

Podemos ainda estender a cobertura para outras variáveis de decisão, para torná-la mais forte.

**Definição 3.16** *A cobertura estendida de uma cobertura  $C$  é  $E(C) = C \cup \{j : a_j \geq a_i, \forall i \in C\}$ .*

Da mesma maneira, a extensão da cobertura implica na desigualdade também válida para um conjunto  $X \subseteq N$ :

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

Em nosso exemplo  $E(C) = \{1, 2, 3, 4, 5, 6\}$ , pois  $a_1, a_2, a_3 \geq \max\{a_j : j \in C\} = 3$ , mas  $7 \notin E(C)$  porque  $a_7 < a_4$ . A desigualdade para a extensão da cobertura é então

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 2 \quad (3.7)$$

Considerando simultaneamente as desigualdades (3.3) e (3.7), observamos que se  $x_1 = 1$ , não existe em (3.7) nenhuma outra variável de decisão que possa assumir valor 1, pois para (3.3) apenas  $x_7$  poderia ser selecionado sem que a capacidade da mochila seja excedida, e  $x_7$  não pertence à (3.7). Mas se considerarmos apenas a desigualdade da cobertura estendida, ainda é possível que outra variável de decisão seja diferente de zero. Isto sugere que a desigualdade (3.7) ainda possui folga, e que esta folga está nas variáveis de decisão adicionadas na extensão da cobertura, pois para a cobertura apenas, a desigualdade é mínima. Em síntese, poderíamos buscar valores  $\alpha_1, \alpha_2$  e  $\alpha_3$  que tornam a desigualdade

$$\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + x_4 + x_5 + x_6 \leq 2$$

o mais apertada possível.



### 3.3.3 Lifting

Dado uma cobertura  $C \subseteq N = \{1, \dots, n\}$ , desejamos encontrar valores para  $\alpha_j$ ,  $j \in N \setminus C$ , de maneira que a desigualdade

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

seja válida para  $X = \{x \in \mathbb{B}^n : \sum_{j=1}^n a_j x_j \leq b\}$ . A extensão  $E(C)$  de uma cobertura  $C$ , conforme visto acima, nos dá uma forma de obter valores  $\alpha_j$ , onde  $\alpha_j = 1$  para  $j \in E(C) \setminus C$  enquanto  $\alpha_j = 0$  para  $j \in N \setminus E(C)$ . Abaixo apresentamos um procedimento alternativo para o cômputo dos fatores  $\alpha_j$ . Este procedimento é conhecido por *lifting* e os parâmetros  $\alpha_j$  por *fatores de lifting*.

Seja  $\langle j_1, \dots, j_r \rangle$  uma ordenação de  $N \setminus C$ . Faça para  $t = 1$  até  $r$ .

Para calcular o maior valor possível de  $\alpha_{j_t}$  no qual  $\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$  seja válida, devemos resolver o problema

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

assim chegamos a  $\alpha_{j_t} = |C| - 1 - \zeta_t$ .

Se a cobertura  $C$  for mínima, a desigualdade gerada utilizando este procedimento define uma faceta de  $\text{conv}(X)$  [24, Teorema 1]. De fato, a desigualdade (3.6) define uma faceta, portanto uma face máxima, para o subconjunto das variáveis de decisão derivadas da cobertura. O procedimento de *lifting* busca obter os pesos das variáveis de decisão que não aparecem na cobertura, de maneira que a restrição obtida no espaço completo das variáveis continue definindo uma faceta.

Podemos observar, entretanto, que cada iteração deste procedimento requer que seja resolvida uma instância de um problema da mochila, cuja dificuldade de resolução é comparável a do problema. Assim, são utilizadas heurísticas para a obtenção de cortes que podem não ser ótimos, mas são bons o suficiente para reduzir o poliedro de maneira a auxiliar na resolução do problema original com algoritmos de planos de corte ou *Branch and Cut*.

Sabemos que  $C = \{4, 5, 6\}$  é uma cobertura mínima para a restrição da mochila (3.3). Vamos agora ilustrar o procedimento para obter os fatores de *lifting* das variáveis de decisão que não pertencem à cobertura. Primeiro construímos uma ordenação de  $N \setminus C = \langle j_1, j_2, j_3, j_4 \rangle = \langle 1, 2, 3, 7 \rangle$ ,  $r = 4$ . Para este procedimento, considere implícita a restrição  $x \in \mathbb{B}^{|C|+t-1}$ . Efetuamos então o processo iterativo, como segue:

- $$\begin{array}{ll}
t = 1 & j_1 = 1 \\
1) & \zeta_1 = \max x_4 + x_5 + x_6 \\
& \text{s.a. } 3x_4 + 2x_5 + 2x_6 \leq 6 - a_1 = 6 - 5 = 1 \\
2) & \zeta_1 = 0 \\
3) & \alpha_1 = |C| - 1 - \zeta_1 = 2 \\
t = 2 & j_2 = 2 \\
4) & \zeta_2 = \max 2x_1 + x_4 + x_5 + x_6 \\
& \text{s.a. } 5x_1 + 3x_4 + 2x_5 + 2x_6 \leq 1 \\
5) & \zeta_2 = 0 \\
6) & \alpha_2 = |C| - 1 - \zeta_2 = 2 \\
t = 3 & j_3 = 3 \\
7) & \zeta_3 = \max 2x_1 + 2x_2 + x_4 + x_5 + x_6 \\
& \text{s.a. } 5x_1 + 5x_2 + 3x_4 + 2x_5 + 2x_6 \leq 2 \\
8) & \zeta_3 = 1 \\
9) & \alpha_3 = |C| - 1 - \zeta_3 = 1 \\
t = 4 & j_4 = 7 \\
10) & \zeta_4 = \max 2x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 \\
& \text{s.a. } 5x_1 + 5x_2 + 4x_3 + 3x_4 + 2x_5 + 2x_6 \leq 5 \\
11) & \zeta_4 = 2 \\
12) & \alpha_7 = |C| - 1 - \zeta_4 = 0
\end{array}$$

Chegamos então à desigualdade:

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

válida para  $X$ . Esta restrição nos define a face  $F = \{x \in \text{conv}(P \cap \mathbb{Z}^7) : 2x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 = 2\}$ . Considere os pontos  $x^1 = (1, 0, 0, 0, 0, 0, 0)$ ,  $x^2 = (0, 1, 0, 0, 0, 0, 0)$ ,  $x^3 = (0, 0, 1, 0, 1, 0, 0)$ ,  $x^4 = (0, 0, 0, 1, 1, 0, 0)$ ,  $x^5 = (0, 0, 0, 1, 0, 1, 0)$ ,  $x^6 = (0, 0, 0, 0, 1, 1, 0)$ ,  $x^7 = (0, 0, 1, 0, 0, 1, 1)$ . Notamos que  $x^j \in F$ ,  $j = 1, \dots, 7$ , e que estes pontos são afim independentes. Então  $\dim(F) = 6$ , e como  $\dim(F) = \dim(\tilde{P}) - 1$ ,  $\tilde{P} = \text{conv}(P \cap \mathbb{Z}^7)$ ,  $F$  é uma faceta do poliedro  $\tilde{P}$ .

### 3.3.4 Down-lifting

*Lifting* pode ser entendido como o procedimento onde tomamos a face induzida pela desigualdade da cobertura,  $\sum_{j \in C} x_j \leq |C| - 1$ , do poliedro da formulação do problema da mochila acrescido das equações  $x_j = 0$  para  $j \in N \setminus C$ . Então rotacionamos a desigualdade de cobertura de forma a se obter uma face de maior dimensão para o poliedro original da mochila.

Um procedimento análogo pode ser realizado com uma face definida por equações da forma  $x_j = 1$  [22, 25]. Para qualquer cobertura  $C$  e qualquer subconjunto  $D \subset C$ , a desigualdade:

$$\sum_{j \in C \setminus D} x_j \leq |C| - |D| - 1$$

é válida para o poliedro restrito:

$$\text{conv}(\{x \in \mathbb{B}^{|C|-|D|} : \sum_{j \in N \setminus D} a_j x_j \leq b - \sum_{j \in D} a_j\}).$$

O *lifting* da seção anterior pode ser utilizado sobre  $C \setminus D$  para produzir uma desigualdade:

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

diminuindo a capacidade da mochila para  $b - \sum_{j \in D} a_j$ , pois  $x_j = 1$ ,  $j \in D$ , e considerando os somatórios sobre  $C \setminus D$ , assim como  $\alpha_j = |C| - |D| - 1 - \zeta_t$ . O cômputo destes fatores ( $N \setminus C$ ) algumas vezes é chamado de *up-lifting*, para diferenciá-los dos fatores das variáveis fixadas em 1 ( $D$ ).

Finalmente, a desigualdade acima pode ser submetida a um processo de *down-lifting* dos fatores de  $D$  para se obter uma faceta da forma:

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

onde  $\beta_j = \zeta_j - (|C| - |D|) - 1$ , e os fatores  $\zeta_t$ ,  $j_t \in \langle j_1, \dots, j_r \rangle$  é uma ordenação de  $D$ , são obtidos resolvendo o problema:

$$\begin{aligned} \zeta_t = \text{Maximize} \quad & \sum_{j \in C \setminus D} x_j + \sum_{j \in N \setminus C} \alpha_j x_j + \sum_{i=1}^{t-1} \beta_{j_i} x_{j_i} \\ \text{Sujeito a:} \quad & \sum_{i \in N \setminus D} a_i x_i \leq b - \sum_{j \in D} a_j + \sum_{i=1}^t a_{j_i} \\ & x \in \mathbb{B}^{|N|-|D|+t-1} \end{aligned}$$

Temos a desigualdade da cobertura correspondente a  $C = \{4, 5, 6\}$ , dada por  $x_4 + x_5 + x_6 \leq 2$ . Fazendo  $D = \{5\}$ , obtemos a desigualdade  $x_4 + x_6 \leq 1$  para  $\text{conv}(P \cap \{x \in \mathbb{B}^{|N|} : x_5 = 1\})$ . Executando *lifting* para  $x_3$ , calculamos:

$$\zeta_3 = \max\{x_4 + x_6 : 3x_4 + 2x_6 \leq 0\} = 0$$

e portanto,  $\alpha_3 = |C| - |D| - 1 - \zeta_3 = 1$ , obtendo a desigualdade  $x_3 + x_4 + x_6 \leq 1$ .

Realizando *lifting* para  $x_7$ , calculamos primeiramente:

$$\zeta_7 = \max\{x_3 + x_4 + x_6 : 4x_3 + 3x_4 + 2x_6 \leq 3\} = 1$$

que nos leva a deduzir que  $\alpha_7 = 0$ . Daí resulta a desigualdade  $x_3 + x_4 + x_6 \leq 1$ .

Executando *down-lifting* na variável  $x_5$ , desejamos calcular  $\beta_5$  tal que:

$$\beta_5 x_5 + x_3 + x_4 + x_6 \leq 1 + \beta_5 \tag{3.8}$$

permaneça válida quando  $x_5 = 0$ . Note que, qualquer que seja o valor de  $\beta_5$ , (3.8) é válida quando  $x_5 = 1$ . Para isto,

$$\beta_5 \geq \max\{x_3 + x_4 + x_6 : 4x_3 + 3x_4 + 2x_6 + x_7 \leq 6\} - 1 = \zeta_5 - 1$$

Calculando  $\zeta_5$ , descobrimos que  $\zeta_5 = 2$  e portanto  $\beta_5 \geq 2 - 1 = 1$  que nos permite obter a desigualdade  $x_3 + x_4 + x_5 + x_6 \leq 2$ .

Continuamos agora com o *lifting* de  $x_1$ . Devemos calcular  $\zeta_1 = \max\{x_3 + x_4 + x_5 + x_6 : 4x_3 + 3x_4 + 2x_6 + x_7 \leq 1\} = 0$ . Concluimos que  $\alpha_1 = 2 - 0 = 2$ . Para a última variável,  $x_2$ , computamos  $\zeta_2 = \max\{2x_1 + x_3 + x_4 + x_5 + x_6 : 5x_1 + 4x_3 + 3x_4 + 2x_6 + x_7 \leq 1\} = 0$ , permitindo calcular que  $\alpha_2 = 2$  e obter a desigualdade válida

$$2x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 \leq 2.$$

Coincidentemente chegamos à mesma desigualdade obtida na seção anterior. Já foi demonstrado que esta desigualdade induz uma faceta para o poliedro  $\tilde{P} = \text{conv}(P \cap \mathbb{Z}^7)$ .

A ordem em que os fatores de *lifting* são calculados é importante na construção da faceta. É possível obter desigualdades diferentes para uma mesma cobertura modificando apenas a ordem de cálculo. Outro ponto a considerar na ordenação é que os subproblemas gerados sejam sempre factíveis. Por isto, em nosso exemplo alternamos o cômputo dos fatores de *up-lifting* com os de *down-lifting*.

### 3.4 O problema da mochila com restrições de precedência

Na seção anterior vimos que para a instância apresentada do nosso problema da mochila, as variáveis de decisão assumiram valor  $x^T = (0, 0, 0, 1, 0, 1, 1)$  para uma solução ótima. Portanto para esta instância o alpinista deveria levar a ração, os grampos e o aparelho de GPS (Tabela 3.2). Entretanto, não faz sentido que o alpinista leve os grampos sem que ele também leve a corda, mas a corda pode ser útil sem os grampos. Assim em nosso problema hipotético temos uma restrição de precedência de que os grampos só podem ser levados se a corda também for. Na Figura 3.4 apresentamos uma representação para esta restrição e a estendemos aos outros elementos da mochila. Para representar as restrições de precedência utilizamos um grafo acíclico direcionado  $G = (V, E)$  em que seus vértices  $V = \{1, \dots, 7\}$  são os próprios elementos da mochila e suas arestas  $E = \{(2, 6), (3, 6), (4, 1)\}$  indicam a precedência.

A seguir, apresentamos algumas definições que serão necessárias para a obtenção de desigualdades válidas para o problema da mochila com restrições de precedência.

**Definição 3.17** Dizemos que  $i$  precede  $j$ , denotado por  $i \preceq j$ , se existe um caminho de  $i$  até  $j$  em  $G$ .

Esta definição significa que  $i$  deve ser carregado na mochila antes que  $j$  seja considerado.

**Definição 3.18** Dizemos que  $i$  precede propriamente  $j$ , denotado por  $i \prec j$ , se  $i \preceq j$  e  $i \neq j$ .

$n$	item	utilidade	tamanho	levar
1	Fogareiro	2	5	não
2	Corda	7	5	não
3	Colchonete	2	4	não
4	Ração	4	3	sim
5	Primeiros Socorros	3	2	não
6	Grampos	7	2	sim
7	Aparelho de GPS	6	1	sim

Tabela 3.2: Solução para uma instância do problema da mochila sem restrição de precedência

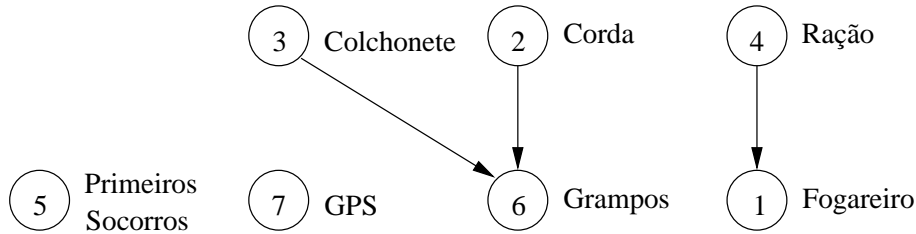


Figura 3.4: Exemplo de restrição de precedência

**Definição 3.19** Dado um conjunto  $S \subseteq V$ ,  $l(S) = \{i : i \preceq j \text{ para algum } j \in S\}$  é o conjunto suporte de  $S$ . Para apenas um elemento,  $l(i) = l(\{i\})$ .

O conjunto suporte contém todos os predecessores de  $S$ . Note que  $S \subseteq l(S)$ .

**Definição 3.20** Dado um conjunto  $S \subseteq V$ ,  $H(S) = \{i \in S : \nexists j \in S | i \prec j\}$  é o conjunto das folhas de  $S$ , e  $R(S) = \{i \in S : \nexists j \in S | j \prec i\}$  é o conjunto das raízes de  $S$ .

Considere o grafo da Figura 3.4 e  $S = \{1, 2, 6\}$ . Então  $l(S) = \{1, 2, 3, 4, 6\}$ ,  $H(l(S)) = \{1, 6\}$  e  $R(l(S)) = \{2, 3, 4\}$ .

### 3.4.1 Definição do problema

Formalmente, o problema da mochila com restrições de precedência pode ser definido como: dado um grafo de precedência acíclico direcionado  $G = (V, E)$  de vértices  $V = \{1, \dots, n\}$  e arestas  $E \subseteq V \times V$ , um peso  $c_i$  e um volume  $a_i$ ,  $i \in V$ , para cada vértice e uma capacidade  $b$  da mochila,

$$\begin{aligned}
 P_{MP} : \text{Maximize } & \sum_{j \in V} c_j x_j \\
 \text{Sujeito a: } & x_j \geq 0 && j \in H(V) \\
 & x_j \leq 1 && j \in R(V) \\
 & x_j \leq x_i && \forall (i, j) \in E \\
 & \sum_{j \in V} a_j x_j \leq b \\
 & x_j \in \mathbb{B} && j \in V
 \end{aligned}$$

**Definição 3.21**  $\sigma_j = \sum_{i \in l(j)} a_i$  é a capacidade mínima da mochila necessária para selecionar o elemento  $j$  e seus predecessores.

**Proposição 3.4** Se  $\max\{\sigma_j : j \in H(V)\} \leq b$ , então  $\dim(\tilde{P}) = n$ , onde  $\tilde{P} = \text{conv}(P \cap \mathbb{Z}^n)$  [26].

### 3.4.2 $K$ -coberturas

Aqui vamos generalizar o conceito de cobertura ao problema da mochila com restrições de precedência e estendê-lo com a noção de  $K$ -cobertura.

**Definição 3.22**  $b(S) = \sum_{i \in S} a_i$ ,  $S \subseteq V$  é o volume ocupado pelos elementos de  $S$ .

**Definição 3.23**  $C \subseteq V$  é uma cobertura se:

- i.  $b(C) > b$ ,
- ii.  $l(C) = C$ .

**Definição 3.24**  $C \subseteq V$  é uma  $K$ -cobertura se:

- i.  $C$  é uma cobertura,
- ii. Para todo conjunto  $S \subseteq H(C)$  com  $|S| = K$ ,  $b(l(S)) > b$ ,
- iii. Para todo  $i \in S$ ,  $b(l(S) \setminus \{i\}) \leq b$ .

A partir de agora, tomemos uma nova capacidade  $b = 11$  para a mochila em nossos exemplos.

Considere o conjunto  $C = \{1, 3, 4\}$  para o exemplo do problema da mochila.  $C$  é uma cobertura, pois  $l(C) = \{1, 3, 4\}$  e  $b(C) = 5 + 4 + 3 = 12 > 11$ , que atende à Definição 3.23. Note que  $H(C) = \{1, 3\}$ . Se tomarmos o conjunto  $S = \{1, 3\}$ ,  $S \subseteq H(C)$ , temos  $b(l(S)) = b(\{1, 3, 4\}) = 12 > b$ .  $S$  também atende à condição 3 da Definição 3.24, pois para cada  $i \in S$ , se  $i = 1$ ,  $b(l(S) \setminus \{1\}) = b(\{3, 4\}) = 7 \leq 11$ ; e se  $i = 3$ ,  $b(\{1, 4\}) = 8 \leq 11$ . Portanto,  $C$  é uma 2-cobertura.

**Proposição 3.5** Dado uma  $K$ -cobertura  $C \subseteq V$ , a desigualdade

$$\sum_{i \in H(C)} x_i \leq K - 1 \quad (3.9)$$

é uma faceta para  $P_C$  se, e somente se,

$$\bigcap_{\{S \subseteq H(C) : |S| = K-1\}} l(S) = \emptyset$$

Note na 2-cobertura  $C = \{1, 3, 4\}$  que  $\bigcap_{\{S \subseteq \{1,3\} : |S|=1\}} l(S) = l(\{1\}) \cap l(\{3\}) = \{1, 4\} \cap \{3\} = \emptyset$ , portanto a desigualdade induzida pela cobertura  $x_1 + x_3 \leq 2$  é uma faceta para  $P_C$ .

### 3.4.3 Lifting

Considerando a desigualdade de  $K$ -cobertura em (4.6), podemos obter através de um processo de *lifting* uma desigualdade mais forte que englobe as variáveis não contempladas na  $K$ -cobertura, ou seja,

$$\sum_{i \in H(C)} x_i + \sum_{i \in V \setminus C} \alpha_i x_i \leq K - 1$$

sendo  $\alpha_i$ ,  $i \in V \setminus C$ , os *fatores de lifting*. Da mesma forma que no problema clássico da mochila, os fatores de *lifting* dependem da cobertura e a ordem na qual o procedimento é aplicado. Em virtude das restrições de precedência, temos ainda que impor uma ordenação topológica na ordem de *lifting*, para garantir que o procedimento seja bem definido. Seja  $C' = \langle i_1, \dots, i_T \rangle$  uma ordenação topológica dos elementos de  $V \setminus C$ , ou seja,  $i_k \not\prec i_h$  para todo  $1 \leq h < k \leq T$ . Então o fator de *lifting*  $\alpha_{i_t}$  pode ser calculado resolvendo o problema

$$\zeta_t = \text{Maximize} \sum_{j \in H(C)} x_j + \sum_{j=1}^{t-1} \alpha_{i_j} x_{i_j}$$

$$\text{Sujeito a: } x \in X_{U_t|W_t}$$

onde  $X_{U_t|W_t} = \{x \in X : x_i = 0 \text{ para todo } i \in U, \text{ enquanto } x_i = 1 \text{ para todo } i \in W\}$ ,  $U_t = \{i_{t+1}, \dots, i_T\}$ ,  $W_t = \{i_t\}$  e  $\alpha_{i_j} = K - 1 - \zeta_j$ ,  $j = 1, \dots, t-1$ , são obtidos recursivamente, e  $\alpha_{i_t} = K - 1 - \zeta_t$ .

Para aplicarmos programação inteira no cômputo dos fatores de *lifting*, precisamos encontrar uma formulação do conjunto  $X_{U_t|W_t}$ . O problema dado a seguir corresponde a uma formulação em programação matemática que pode ser utilizada para obter  $\zeta_t$ :

$$\zeta_t = \text{Maximize} \sum_{j \in H(C)} x_j + \sum_{j=1}^{t-1} \alpha_{i_j} x_{i_j}$$

$$\text{Sujeito a: } \sum_{j \in V_t} a_j x_j \leq b$$

$$x_i \geq x_j \quad (i, j) \in E_t$$

$$x_{i_t} = 1$$

$$x_i \in \mathbb{B} \quad i \in V_t$$

onde  $G_t = (V_t, E_t)$  é o grafo induzido de  $G$  por  $V_t = C \cup \{i_1, \dots, i_t\}$ , ou seja,  $G_t = G[V_t]$ .

Considere a cobertura  $C = \{1, 3, 4\}$ , e uma ordenação topológica  $C' = \langle 2, 6, 5, 7 \rangle$  dos elementos que não pertencem à cobertura. Para obter os fatores de *lifting* inicialmente temos  $V_t = \{1, 2, 3, 4\}$  e resolvemos  $\zeta_1 = \{\max x_1 + x_3 : 5x_1 + 5x_2 + 4x_3 + 4x_4 \leq 11, x_4 \geq x_1, x_2 = 1\} = 1$ , com solução  $(x_1, \dots, x_7) = (0, 1, 1, 0, 0, 0, 0)$ . Assim obtemos  $\alpha_2 = K - 1 - \zeta_1 = 0$ . Para o elemento seguinte de  $C'$ ,  $\zeta_2 = \{\max x_1 + x_3 : 5x_1 + 5x_2 + 4x_3 + 4x_4 + 2x_6 \leq 11, x_4 \geq x_1, x_3 \geq x_6, x_2 \geq x_6, x_6 = 1\} = 1$ , com  $(x_1, \dots, x_7) = (0, 1, 1, 0, 0, 1, 0)$ . Portanto  $\alpha_6 = 0$ .

O cômputo de  $\alpha_5$  é feito por  $\zeta_3 = \{\max x_1 + x_3 : 5x_1 + 5x_2 + 4x_3 + 4x_4 + 2x_5 + 2x_6 \leq 11, x_4 \geq x_1, x_3 \geq x_6, x_2 \geq x_6, x_5 = 1\} = 1$ , obtendo  $(x_1, \dots, x_7) = (0, 0, 1, 1, 1, 0, 0)$  que implica em  $\alpha_5 = 0$ .

E por fim  $\zeta_4 = \{\max x_1 + x_3 : 5x_1 + 5x_2 + 4x_3 + 4x_4 + 2x_5 + 2x_6 + x_7 \leq 11, x_4 \geq x_1, x_3 \geq x_6, x_2 \geq x_6, x_7 = 1\} = 1$ , com  $(x_1, \dots, x_7) = (0, 0, 1, 1, 0, 0, 1)$  que resulta em  $\alpha_7 = 0$ . Todos os fatores de *lifting* assumiram valor zero, então a desigualdade

$$x_1 + x_3 \leq 2$$

também é uma faceta para  $\text{conv}(P \cap \mathbb{Z}^7)$ .

### 3.5 Linearização por partes

Consideremos o problema de otimização separável:

$$P_{sep} : \text{Maximize } F = \sum_{j=1}^n f_j(x_j)$$

onde  $x \in X \subseteq \mathbb{R}^n$  são as possíveis soluções para o problema e  $f_j$  é uma função não-linear qualquer.  $F$  é dita separável porque seus elementos  $f_j$  são funções apenas de uma variável  $x_j$ .

Tomaremos agora a função  $f_j(x_j)$  e para simplificar a notação eliminaremos o índice  $j$ . A linearização por partes consiste em selecionar  $k+1$  pontos  $P = \{p_i : p_i = (a_i, f(a_i)), a_0 = l, a_k = u, a_{i-1} < a_i\}$  que pertençam a uma região  $[l, u]$  do domínio de  $x$  e interpolá-los por retas, formando a função linear por partes  $\tilde{f}(x)$ . A Figura 3.5 ilustra a função não-linear  $f(x)$  e sua linearização por partes  $\tilde{f}(x)$ . Cada par de pontos  $r_i = (p_{i-1}, p_i)$ ,  $i = 1, \dots, k$ , forma uma região da função linear  $\tilde{f}(x)$ . Apresentaremos a seguir três modelos de aproximação de  $f(x)$  em funções lineares por partes utilizando programação linear inteira mista.

#### 3.5.1 Modelo com pesos por ponto (Modelo I)

Nesta primeira abordagem utilizamos variáveis de decisão reais  $\lambda_i$  associadas ao peso de cada ponto  $p_i$ , e variáveis de decisão inteiras  $y_i$  que selecionam a região de  $\tilde{f}(x)$ . Temos então a seguinte formulação:

$$x = \sum_{i=0}^k a_i \lambda_i \tag{3.10a}$$

$$\tilde{f}(x) = \sum_{i=0}^k f(a_i) \lambda_i \tag{3.10b}$$



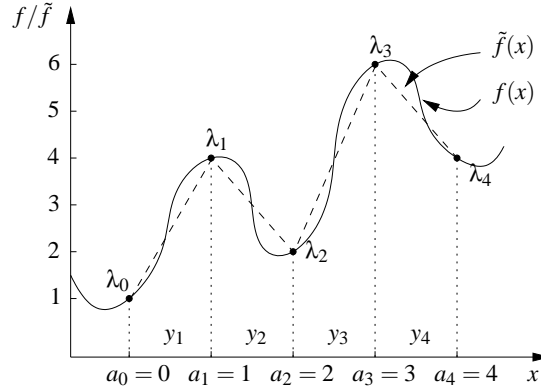


Figura 3.5: Linearização por partes para o modelo por pesos

com restrições:

$$\lambda_0 \leq y_1 \quad (3.11a)$$

$$\lambda_i \leq y_i + y_{i+1} \quad i = 1, \dots, k-1 \quad (3.11b)$$

$$\lambda_k \leq y_k \quad (3.11c)$$

$$\sum_{i=0}^k \lambda_i = 1 \quad (3.11d)$$

$$\sum_{i=1}^k y_i = 1 \quad (3.11e)$$

$$\lambda_i \geq 0 \quad i = 0, \dots, k \quad (3.11f)$$

$$\lambda_i \in \mathbb{R} \quad i = 0, \dots, k \quad (3.11g)$$

$$y_i \in \mathbb{B} \quad i = 1, \dots, k \quad (3.11h)$$

Tomemos como exemplo a Figura 3.5, com os valores dos pontos conforme indicado. Desejamos representar de acordo com o modelo  $x = 2,7$ . Assim,  $y_3 = 1$  e  $y_i = 0$ ,  $i \neq 3$ , seleciona apenas a região  $r_3$  (restrições (3.11e) e (3.11h)). As variáveis  $(\lambda_0, \dots, \lambda_4)$  assumem valores  $(0, 0, 0,3, 0,7, 0)$  (restrições (3.11d), (3.11f) e (3.11g)). Para as restrições (3.11a) a (3.11c) temos:

$$\lambda_0 \leq y_1 \quad \implies \lambda_0 \leq 0 \quad (3.11a)$$

$$\lambda_1 \leq y_1 + y_2 \quad \implies \lambda_1 \leq 0 + 0 = 0 \quad (3.11b)$$

$$\lambda_2 \leq y_2 + y_3 \quad \implies \lambda_2 \leq 0 + 1 = 1$$

$$\lambda_3 \leq y_3 + y_4 \quad \implies \lambda_3 \leq 1 + 0 = 1$$

$$\lambda_4 \leq y_4 \quad \implies \lambda_4 \leq 0 \quad (3.11c)$$

Os valores de  $x$  e  $\tilde{f}(x)$  são calculados por (3.10a) e (3.10a) respectivamente:

$$\begin{aligned}
x &= 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0,3 + 3 \cdot 0,7 + 4 \cdot 0 \\
&= 0,6 + 2,1 = 2,7 \\
\tilde{f}(x) &= 1 \cdot 0 + 4 \cdot 0 + 2 \cdot 0,3 + 6 \cdot 0,7 + 4 \cdot 0 \\
&= 0,6 + 4,2 = 4,8
\end{aligned}$$

Observamos que o ponto desejado  $x = 2,7$  está entre  $a_2$  e  $a_3$ . Portanto,  $y_3 = 1$  seleciona apenas aquela região. Observamos também que  $\lambda_3$  assumiu um valor maior que  $\lambda_2$ , pois  $x$  está mais próximo de  $a_3$ , devendo sua variável de peso ser mais expressiva.

### 3.5.2 Modelo seqüencial de pesos (Modelo II)

Neste modelo, escrevemos  $x$  como:

$$x = a_0 + \lambda_1 + \dots + \lambda_k$$

onde  $0 \leq \lambda_i \leq a_i - a_{i-1}$ ,  $i = 1, \dots, k$ .

Desejamos representar um dado  $x$  que pertence a uma região  $r$  da função. Então  $\lambda_i$  assume:

$$\lambda_i = \begin{cases} a_i - a_{i-1} & \text{se } i < r \\ x - a_i & \text{se } i = r \\ 0 & \text{se } i > r \end{cases}$$

Assim chegamos a formulação:

$$x = a_0 + \sum_{i=1}^k \lambda_i \quad (3.12a)$$

$$\tilde{f}(x) = f(a_0) + \sum_{i=1}^k \frac{f(a_i) - f(a_{i-1})}{a_i - a_{i-1}} \lambda_i \quad (3.12b)$$

com restrições:

$$\lambda_1 \leq a_1 - a_0 \quad (3.13a)$$

$$\lambda_i \leq (a_i - a_{i-1})y_{i-1} \quad i = 2, \dots, k \quad (3.13b)$$

$$\lambda_i \geq (a_i - a_{i-1})y_i \quad i = 1, \dots, k-1 \quad (3.13c)$$

$$\lambda_k \geq 0 \quad (3.13d)$$

$$\lambda_i \in \mathbb{R} \quad i = 1, \dots, k \quad (3.13e)$$

$$y_i \in \mathbb{B} \quad i = 1, \dots, k-1 \quad (3.13f)$$

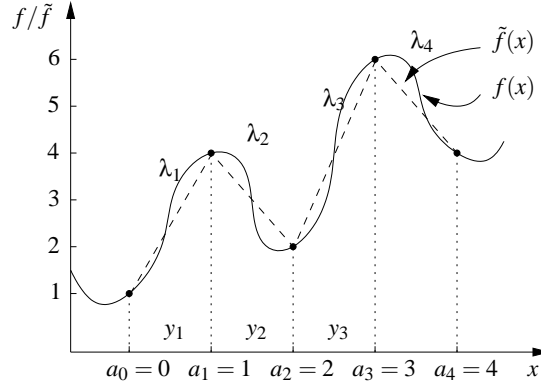


Figura 3.6: Linearização por partes para o modelo seqüencial

Note que em (3.13c) está implícito  $\lambda_i \geq 0$ , pois  $\lambda_i \geq (a_i - a_{i-1})y_i \geq 0$ , sendo  $y_i \in \{0, 1\}$  e  $a_i > a_{i-1}$ . As variáveis inteiras  $y_i$  selecionam as regiões anteriores a  $x$ . Assim, se considerarmos que  $x$  pertence a uma região  $r$ ,

$$y_i = \begin{cases} 1 & \text{se } i < r \\ 0 & \text{se } i \geq r \end{cases}$$

Observamos em (3.13f) que  $y_i$  é definido apenas até  $i = k - 1$ , sendo desnecessário  $y_k$ , pois se  $x$  pertencer à última região,  $y_i = 1, \forall i \in \{1, \dots, k - 1\}$ .

Para o mesmo exemplo do modelo anterior, buscamos  $x = 2,7$ . Como  $a_2 \leq x \leq a_3$ ,  $r = 3$  e  $(y_1, y_2, y_3) = (1, 1, 0)$ . As variáveis reais  $(\lambda_1, \dots, \lambda_4) = (1, 1, 0, 7, 0)$ . A Figura 3.6 ilustra este modelo e as restrições ficam:

$$\lambda_1 \leq a_1 - a_0 \quad \implies \lambda_1 \leq 1 - 0 = 1 \quad (3.13a)$$

$$\lambda_1 \geq (a_1 - a_0)y_1 \quad \implies \lambda_1 \geq (1 - 0)1 = 1 \quad (3.13c)$$

$$\therefore \lambda_1 = 1$$

$$\lambda_2 \leq (a_2 - a_1)y_1 \quad \implies \lambda_2 \leq (2 - 1)1 = 1 \quad (3.13b)$$

$$\lambda_2 \geq (a_2 - a_1)y_2 \quad \implies \lambda_2 \geq (2 - 1)1 = 1 \quad (3.13c)$$

$$\therefore \lambda_2 = 1$$

$$\lambda_3 \leq (a_3 - a_2)y_2 \quad \implies \lambda_3 \leq (3 - 2)1 = 1 \quad (3.13b)$$

$$\lambda_3 \geq (a_3 - a_2)y_3 \quad \implies \lambda_3 \geq (3 - 2)0 = 0 \quad (3.13c)$$

$$\therefore 0 \leq \lambda_3 \leq 1$$

$$\lambda_4 \leq (a_4 - a_3)y_3 \quad \implies \lambda_4 \leq (4 - 3)0 = 0 \quad (3.13b)$$

$$\lambda_4 \geq 0 \quad (3.13d)$$

$$\therefore \lambda_4 = 0$$

Os valores de  $x$  e  $\tilde{f}(x)$  são calculados por (3.12a) e (3.12b) respectivamente:

$$\begin{aligned}
x &= a_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \\
&= 0 + 1 + 1 + 0,7 + 0 = 2,7 \\
\tilde{f}(x) &= f(a_0) + \frac{f(a_1) - f(a_0)}{a_1 - a_0} \lambda_1 + \frac{f(a_2) - f(a_1)}{a_2 - a_1} \lambda_2 + \\
&\quad + \frac{f(a_3) - f(a_2)}{a_3 - a_2} \lambda_3 + \frac{f(a_4) - f(a_3)}{a_4 - a_3} \lambda_4 \\
&= 1 + 3 \cdot 1 + (-2) \cdot 1 + 4 \cdot 0,7 + (-2) \cdot 0 \\
&= 4,8
\end{aligned}$$

Padberg apresenta em [27] uma comparação entre este modelo (Modelo II) e o anterior (Modelo I). Ele faz uma transformação do Modelo I para o espaço de variáveis de decisão do Modelo II, e mostra que o Modelo II está estritamente contido nesta transformação do Modelo I. Mostra ainda que os vértices do poliedro correspondente à relaxação linear do Modelo II são todos inteiros com respeito às variáveis  $y$ . Isto implica que a solução de um problema de otimizar uma função linear por partes pode ser obtida por meio de programação linear pura. Neste aspecto, o Modelo II é considerado localmente muito melhor que o Modelo I. Padberg propõe ainda que o Modelo I não seja utilizado, pois esta formulação *é sempre igual ou melhor que a anterior, mesmo no pior caso* e que o Modelo I *deveria ser definitivamente abandonado, apesar de sua popularidade nos livros texto*.

Entretanto sob o aspecto didático, o Modelo I é mais intuitivo que o II, devido ao significado de suas variáveis de decisão e simplicidade das restrições. Em um exercício algébrico, Padberg propõe uma formulação que transforma o Modelo II no espaço de variáveis de decisão de I, mas suas restrições continuam pouco intuitivas. Então Sherali propõe em [28] uma formulação simples com variáveis de decisão semelhantes às do Modelo I e com as características poliedrais do Modelo II.

### 3.5.3 Modelo de Sherali (Modelo III)

Sherali propõe em [28] uma “formulação didática” para a linearização por partes. Nesta formulação, são utilizadas duas variáveis de peso para cada região da linearização:  $\lambda_i^L$  é o peso do ponto à esquerda da região e  $\lambda_i^R$  é o peso do ponto à direita da região. Também são empregadas variáveis  $y_i$  inteiras que selecionam a região. Esta formulação é mais geral que as anteriores, pois permite também linearizar funções descontínuas.

$$x = \sum_{i=1}^k [a_{i-1} \lambda_i^L + a_i \lambda_i^R] \quad (3.14a)$$

$$\tilde{f}(x) = \sum_{i=1}^k [f(a_{i-1}) \lambda_i^L + f(a_i) \lambda_i^R] \quad (3.14b)$$

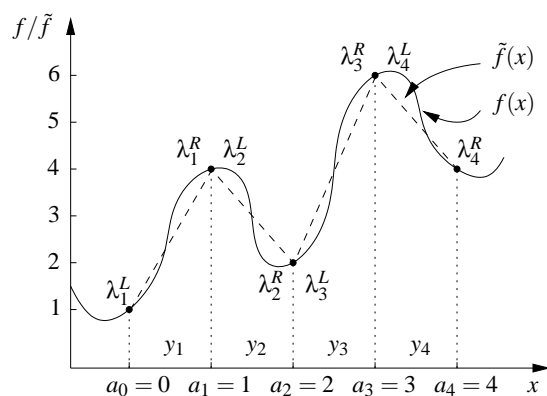


Figura 3.7: Linearização por partes para o modelo de Sherali

com restrições

$$\lambda_i^L + \lambda_i^R = y_i \quad i = 1, \dots, k \quad (3.15a)$$

$$\sum_{i=1}^k y_i = 1 \quad (3.15b)$$

$$\lambda_i \geq 0 \quad i = 1, \dots, k \quad (3.15c)$$

$$\lambda_i \in \mathbb{R} \quad i = 1, \dots, k \quad (3.15d)$$

$$y_i \in \mathbb{B} \quad i = 1, \dots, k \quad (3.15e)$$

Na Figura 3.7 vemos uma ilustração deste modelo. Para exemplificar, representamos  $x = 2,7$ . Para selecionar a região  $r_3$  de  $x$ , as variáveis inteiras  $y_i$  assumem valores  $(y_1, \dots, y_4) = (0, 0, 1, 0)$ , atendendo as restrições (3.15b) e (3.15e). Devido às restrições (3.15a), (3.15c) e (3.15d),  $\lambda_i^L$  e  $\lambda_i^R = 0$ ,  $i \in \{1, 2, 4\}$ ; e  $\lambda_3^L = 0,3$  e  $\lambda_3^R = 0,7$ .

Com (3.14a) e (3.14b) calculamos  $x$  e  $\tilde{f}(x)$ , como apresentado a seguir. Como  $\lambda_i^L$  e  $\lambda_i^R$  são nulos para  $i \in \{1, 2, 4\}$ , estes termos serão omitidos.

$$\begin{aligned} x &= a_2 \lambda_3^L + a_3 \lambda_3^R \\ &= 2 \cdot 0,3 + 3 \cdot 0,7 = 2,7 \\ \tilde{f}(x) &= f(a_2) \lambda_3^L + f(a_3) \lambda_3^R \\ &= 2 \cdot 0,3 + 6 \cdot 0,7 = 4,8 \end{aligned}$$

Do exemplo apresentado, (3.15a) e (3.15b) são escritas matricialmente como  $\hat{A}x = \hat{b}$ ,

$$\begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$x^T = \left[ \lambda_1^L \quad \lambda_1^R \quad y_1 \quad \lambda_2^L \quad \lambda_2^R \quad y_2 \quad \lambda_3^L \quad \lambda_3^R \quad y_3 \quad \lambda_4^L \quad \lambda_4^R \quad y_4 \right]$$

Desejamos mostrar que  $\hat{A}$  é totalmente unimodular. As colunas das variáveis  $\lambda$  possuem apenas uma linha diferente de zero. De acordo com Newhauser [22], estas colunas podem ser desconsideradas. Note que a matriz resultante (colunas de  $y$ ) corresponde a uma matriz de incidência [21, página 40], que é totalmente unimodular. Assim, a matriz  $\hat{A}$  também é totalmente unimodular.

As restrições (3.15c) a (3.15e), na forma  $y_i \geq 0$  e  $y_i \leq 1$  para todo  $i = 1, \dots, k$ , implicam em identidades que são concatenadas à matriz  $\hat{A}$ , e não afetam sua unimodularidade. Note também que as restrições de igualdade podem ser transformadas em desigualdades ( $\pi^T x = \pi_0 \implies \pi^T x \geq \pi_0$  e  $\pi^T x \leq \pi_0$ ). Assim, a matriz de restrições para esta formulação é totalmente unimodular.

A consequência é que o poliedro desta formulação é mínimo, ou seja, ele corresponde ao poliedro do fecho convexo das soluções, e todos seus vértices estão sobre pontos inteiros. Assim, apenas programação linear é suficiente para resolver um problema de linearização por partes pura. Em [28] Sherali mostra que se  $f(x)$  for contínua, sua formulação é equivalente à do Modelo II.

Em [29] apresentamos uma comparação entre o modelo com pesos por ponto (Modelo I) com o modelo de Sherali (Modelo III). Aplicamos os modelos ao problema de alocação ótima de gás de injeção com restrição de capacidade máxima de injeção, mas sem a restrição de precedência de ativação. Executamos experimentos numéricos em instâncias com 32 e 64 poços de petróleo, aplicando o algoritmo de *Branch and Bound* puro, sem o uso de geração automática de cortes. Os experimentos mostraram que as formulações baseadas no modelo de Sherali apresentaram sempre um número menor de iterações do algoritmo em relação às formulações baseadas no Modelo I, e portanto, que o modelo apresenta resultados melhores também quando não é aplicado de maneira pura, isto é, com outras restrições além das do modelo (no caso, restrições de capacidade máxima de injeção de gás).

### 3.6 Sumário

Neste capítulo vimos que a otimização é a área da matemática aplicada que busca uma solução ótima (mínima ou máxima) para um problema atendendo a restrições. Vimos que ela é dividida em vários domínios, e nos aprofundamos no domínio que envolve variáveis inteiras. Estudamos o problema da mochila e vimos mais detalhadamente o problema com restrições de precedência, que será aplicado ao problema de otimização de gás de injeção. Mostramos como obter desigualdades fortes para esta formulação, que são importantes para agilizar o processo de resolução do problema. Por fim,

---

apresentamos três modelos de linearização por partes de funções separáveis utilizando programação linear inteira-mista.

## Capítulo 4

# Modelo linear por partes

Na Seção 2.3 apresentamos uma formulação geral para o problema de alocação ótima de gás de injeção sob restrições da capacidade máxima de injeção de gás e de precedência de ativação. Neste capítulo a curva de performance do poço (CPP) será linearizada por partes e aplicaremos programação linear inteira-mista sobre  $P(G)$ .

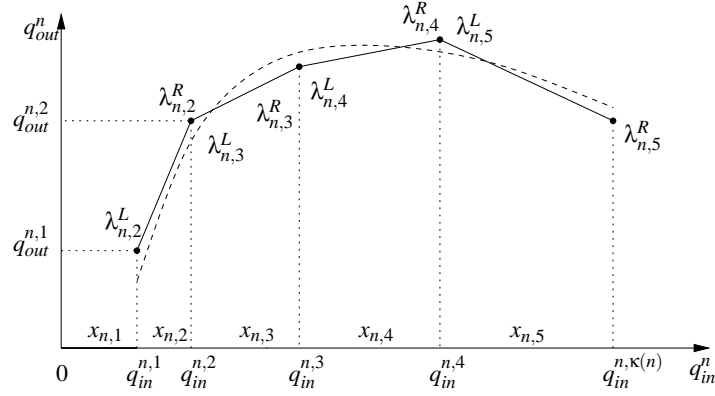
Em seguida apresentaremos desigualdades válidas para o poliedro da formulação, derivadas a partir das desigualdades de cobertura para o problema da mochila sob restrições de precedência, as quais podem ser empregadas na geração de cortes em um algoritmo de planos de corte. Os conceitos introduzidos serão elucidados por meio de exemplos. As formulações e demonstrações apresentadas neste capítulo foram desenvolvidas por Nakashima[19]. Camponogara e Conto apresentam sinteticamente em [30] o desenvolvimento mostrado neste capítulo.

### 4.1 Formulação do problema

Desejamos uma formulação linear por partes para a formulação apresentada na Seção 2.3. Para isto, aplicamos o modelo de linearização proposto por Sherali [28] (Seção 3.5.3) devido a sua simplicidade e características poliedrais.

A Figura 4.1 apresenta uma aproximação linear por partes de um poço  $n$  para uma curva contínua. A aproximação é obtida diretamente de pontos amostrados da CPP  $Q^n = \{(q_{in}^{n,1}, q_{out}^{n,1}), \dots, (q_{in}^{n,\kappa(n)}, q_{out}^{n,\kappa(n)})\}$ , onde  $\kappa(n)$  é o número de pontos de amostra. Alternativamente, se a CPP está na forma de uma curva contínua  $\tilde{q}_{out}^n$ , a aproximação linear por partes pode ser produzida tomando uma quantidade arbitrária de pontos desta curva, que refletem a qualidade da linearização. Variáveis binárias  $x_{n,k}$  são necessárias para indicar qual segmento linear está representado na função objetivo:  $x_{n,1} = 1$  se o poço  $n$  está desativado, neste caso  $q_{in}^n = q_{out}^n = 0$ ; note que  $y_n = 1 - x_{n,1}$ . Para  $k \geq 2$ ,  $x_{n,k} = 1$  se, e somente se,  $(q_{in}^n, q_{out}^n)$  é uma combinação convexa dos pontos extremos do  $k$ -ésimo intervalo, isto é,  $(q_{in}^n, q_{out}^n) = \lambda_{n,k}^L (q_{in}^{n,k-1}, q_{out}^{n,k-1}) + \lambda_{n,k}^R (q_{in}^{n,k}, q_{out}^{n,k})$  com  $\lambda_{n,k}^L + \lambda_{n,k}^R = 1$  e  $\lambda_{n,k}^L, \lambda_{n,k}^R \geq 0$ .



Figura 4.1: Linearização por partes da CPP de um poço  $n$ .

Para cada  $n \in \mathcal{N} = \{1, \dots, N\}$ , temos as seguintes restrições:

$$\sum_{k=1}^{\kappa(n)} x_{n,k} = 1 \quad (4.1a)$$

$$y_n = 1 - x_{n,1} \quad (4.1b)$$

$$\lambda_{n,k}^L, \lambda_{n,k}^R \geq 0, \quad k = 2, \dots, \kappa(n) \quad (4.1c)$$

$$\lambda_{n,k}^L + \lambda_{n,k}^R = x_{n,k}, \quad k = 2, \dots, \kappa(n) \quad (4.1d)$$

$$q_{in}^n = \sum_{k=2}^{\kappa(n)} (q_{in}^{n,k-1} \lambda_{n,k}^L + q_{in}^{n,k} \lambda_{n,k}^R) \quad (4.1e)$$

$$q_{out}^n = \sum_{k=2}^{\kappa(n)} (q_{out}^{n,k-1} \lambda_{n,k}^L + q_{out}^{n,k} \lambda_{n,k}^R) \quad (4.1f)$$

$$x_{n,k} \in \{0, 1\} \quad k = 1, \dots, \kappa(n) \quad (4.1g)$$

Para o conjunto de pontos  $Q^n$ , obtidos diretamente por testes no poço ou da linearização da aproximação  $\tilde{q}_{out}^n$  da CPP, assumimos algumas hipóteses que não implicam em perda da generalidade do problema:

**Hipótese 4.1** Para cada poço  $n \in \mathcal{N}$ :

1.  $0 < q_{in}^{n,k} < q_{in}^{n,k+1}$ , para todo  $1 \leq k < \kappa(n)$ , implica que as taxas de injeção são distintas e consecutivas;
2.  $q_{in}^{n,1} = l_n$  e  $q_{in}^{n,\kappa(n)} = u_n \leq q_{in}^{max}$  para eliminar pontos desnecessários; e
3. cada ponto  $q^{n,k} = (q_{in}^{n,k}, q_{out}^{n,k}) \in Q^n$ ,  $2 < k < \kappa(n)$ , não é uma combinação convexa de  $(q_{in}^{n,k-1}, q_{out}^{n,k-1})$  e  $(q_{in}^{n,k+1}, q_{out}^{n,k+1})$ , para descartar pontos redundantes.

Seja  $f^{n,k} = f_n(q_{in}^{n,k}, q_{out}^{n,k}) = (p_o \gamma_o^n + p_g \gamma_g^n - p_w \gamma_w^n) q_{out}^{n,k} - p_{in} q_{in}^{n,k}$  a contribuição na função lucro do poço  $n$  operando no ponto  $(q_{in}^{n,k}, q_{out}^{n,k})$ , podemos levar a linearização por partes para a função objetivo:

$$\text{Maximize } f = \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} (\lambda_{n,k}^L f^{n,k-1} + \lambda_{n,k}^R f^{n,k}) \quad (4.2)$$

Portanto, a formulação linear inteira por partes  $P_{pl}(G)$  é composta pela função objetivo (4.2), pelas restrições (4.1a) a (4.1g), e pelas restrições de capacidade máxima de injeção de gás e precedência. No Apêndice B apresentamos uma formulação linear por partes para  $P(G)$  que considera a concavidade da curva de performance do poço. Esta formulação utiliza uma propriedade do modelo de Sherali [28] que permite uma redução no número de variáveis de decisão inteiras caso a curva linear por partes possua regiões côncavas. O modelo apresentado não implica em uma redução da complexidade computacional da resolução do problema, mas o menor número de variáveis inteiras pode significar um número menor de nós na árvore de busca de *Branch and Bound*, e conseqüentemente um menor tempo de processamento.

Baseado na análise das restrições de  $P_{pl}(G)$ , algumas variáveis de decisão podem ser descartadas, e as restrições reformuladas para uma formulação mais compacta. Para cada poço  $n$ , as variáveis que podem ser expressas como função de outras são:

$$\lambda_{n,k}^R = x_{n,k} - \lambda_{n,k}^L \quad k = 2, \dots, \kappa(n) \quad (4.3a)$$

$$\begin{aligned} q_{in}^n &= \sum_{k=2}^{\kappa(n)} [q_{in}^{n,k-1} \lambda_{n,k}^L + q_{in}^{n,k} \lambda_{n,k}^R] \\ &= \sum_{k=2}^{\kappa(n)} [q_{in}^{n,k} x_{n,k} - (q_{in}^{n,k} - q_{in}^{n,k-1}) \lambda_{n,k}^L] \end{aligned} \quad (4.3b)$$

$$x_{n,1} = 1 - \sum_{k=2}^{\kappa(n)} x_{n,k} \quad (4.3c)$$

$$y_n = \sum_{k=2}^{\kappa(n)} x_{n,k} \quad (4.3d)$$

Aplicando à  $P_{pl}(G)$  estas substituições e algumas outras que serão apresentadas a seguir, obtemos a formulação linear por partes compacta:

$$P_{cpl}(G) : \text{Maximize } f = \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [f^{n,k} x_{n,k} - (f^{n,k} - f^{n,k-1}) \lambda_{n,k}] \quad (4.4a)$$

$$\text{Sujeito a: } \sum_{n=1}^N \sum_{k=2}^{\kappa(n)} [q_{in}^{n,k} x_{n,k} - (q_{in}^{n,k} - q_{in}^{n,k-1}) \lambda_{n,k}] \leq q_{in}^{max} \quad (4.4b)$$

$$\sum_{k=2}^{\kappa(n)} x_{n,k} \leq \sum_{k=2}^{\kappa(m)} x_{m,k} \quad \forall (m,n) \in E(G) \quad (4.4c)$$

$$\sum_{k=2}^{\kappa(n)} x_{n,k} \leq 1 \quad \forall n \in R(G) \quad (4.4d)$$

$$\lambda_{n,k} \geq 0, \quad \forall (n,k) \in \Omega \quad (4.4e)$$

$$x_{n,k} - \lambda_{n,k} \geq 0 \quad \forall (n,k) \in \Omega \quad (4.4f)$$

$$x_{n,k} \in \{0, 1\} \quad \forall (n,k) \in \Omega \quad (4.4g)$$

onde  $\lambda_{n,k} = \lambda_{n,k}^L$ ,  $R(G) = \{j \in V(G) : \nexists (i,j) \in E(G)\}$  é o conjunto das raízes de  $G$  e  $\Omega = \{(n,k) : n = 1, \dots, N, k = 2, \dots, \kappa(n)\}$  é o conjunto de pares poços e níveis de ativação, isto é, o conjunto de índices dos pontos da representação linear por partes da CPP. Note que as restrições (4.4e) e (4.4f)

garantem que  $x_{n,k} \geq 0$ . Ainda, as restrições (4.4c) e (4.4d) implicam que  $x_{n,k} \leq 1$  para qualquer caminho  $\langle i, j, \dots \rangle$  de  $G$  com  $i \in R(G)$ , assim, é suficiente impor a restrição  $\sum_{k=2}^{\kappa(i)} x_{i,k} = y_i \leq 1$ .

Sejam  $\lambda_n = \{\lambda_{n,k} : k = 2, \dots, \kappa(n)\}$  os fatores que induzem a combinação convexa dos elementos de  $\mathcal{Q}^n$ , e  $\lambda = \cup_{n=1}^N \lambda_n$ . Sejam  $x_n = \{x_{n,k} : k = 2, \dots, \kappa(n)\}$  as variáveis discretas que forçam a combinação convexa a usar apenas dois elementos consecutivos de  $\mathcal{Q}^n$ , e  $x = \cup_{n=1}^N x_n$ . Chamemos o conjunto das soluções factíveis do problema de alocação de gás sob restrições de precedência de  $x$ . Definimos o poliedro  $\mathcal{P} = \{(x, \lambda) \in \mathbb{R}^{K-N} \times \mathbb{R}^{K-N} : (x, \lambda) \text{ que satisfaça as restrições (4.4b) a (4.4f)}\}$ , onde  $K = \sum_{n=1}^N \kappa(n)$ . Então  $\mathcal{P}$  é uma formulação para  $x$ , pois  $\mathcal{X} = \mathcal{P} \cap (\mathbb{Z}^{K-N} \times \mathbb{R}^{K-N})$ .

## 4.2 Desigualdades válidas

No capítulo anterior apresentamos de maneira geral como obter desigualdades válidas para o problema da mochila sob restrição de precedência. Podemos observar que  $\mathcal{P}_{cpl}(G)$  se enquadra nesta classe de problemas, pois (4.4b) é uma restrição linear do tipo da mochila e (4.4c) é a restrição de precedência. Nesta seção, apresentaremos famílias de desigualdades válidas de  $\text{conv}(x)$  que serão adicionadas a  $\mathcal{P}$  para torná-lo mais forte. Com estas desigualdades válidas podemos utilizar o algoritmo de *Branch and Cut* apresentado no capítulo anterior.

Começamos apresentando as condições para as quais  $\text{conv}(x)$  possui dimensão cheia. Estas condições garantem que o poliedro não possui variáveis redundantes e dependências lineares, simplificando as provas de maximalidade das faces induzidas pelas desigualdades. Seja  $A_n = \{m \in V : m \prec n\} \setminus \{n\}$  o conjunto de predecessores do nó  $n$ . Seja  $\sigma_n = \sum_{m \in A_n} q_{in}^{m,1}$  é a quantia mínima de gás necessária para ativar os poços que precedem  $n$ .

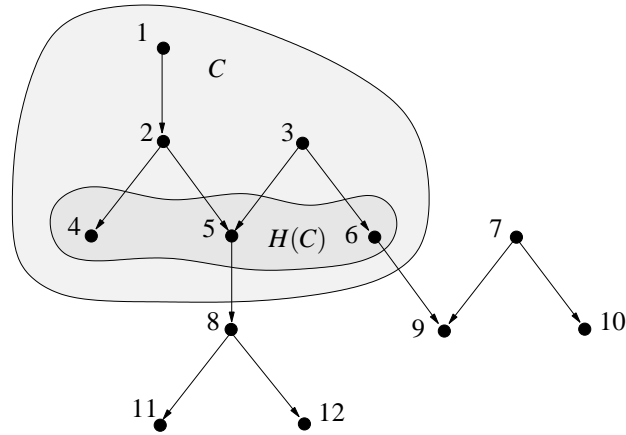
**Proposição 4.1**  *$\text{conv}(x)$  possui dimensão cheia se  $\max\{\sigma_n + q_{in}^{n,\kappa(n)} : n \in \mathcal{N}\} \leq q_{in}^{max}$ .*

**Prova:** Seja  $T = \langle n_1, n_2, \dots, n_N \rangle$  uma ordenação topológica de  $G$ . Seja  $Z$  uma matriz  $2(K-N) \times 2(K-N)$  em que as colunas correspondem as variáveis  $x_{n_1,2}, \lambda_{n_1,2}, \dots, x_{n_1,\kappa(n_1)}, \lambda_{n_1,\kappa(n_1)}, \dots, x_{n_N,2}, \lambda_{n_N,2}, \dots, x_{n_N,\kappa(n_N)}, \lambda_{n_N,\kappa(n_N)}$  nesta ordem, da esquerda para a direita. Para  $i = 1, \dots, N$  e  $k = 2, \dots, \kappa(n_i)$ : seja  $z^{i,k} = (x^{i,k}, \lambda^{i,k})$  o vetor de incidência obtido definindo  $x_{l,2}^{i,k} = \lambda_{l,2}^{i,k} = 1$  para todo  $l \in A_{n_i}$ , e  $x_{n_i,k}^{i,k} = 1$ , enquanto todas as outras entradas são nulas. Seja  $\hat{z}^{i,k} = (\hat{x}^{i,k}, \hat{\lambda}^{i,k})$  idêntico a  $z^{i,k}$ , exceto que  $\hat{\lambda}_{n_i,k}^{i,k} = 1$ ; como as restrições de precedência são respeitadas e  $\max\{\sigma_n + q_{in}^{n,\kappa(n)} : n \in \mathcal{N}\} \leq q_{in}^{max}$ ,  $z^{i,k}$  e  $\hat{z}^{i,k}$  pertencem a  $\mathcal{P}_{cpl}(G)$ .

Sejam  $z^{i,k}$  e  $\hat{z}^{i,k}$  as linhas de  $Z$  ordenadas como foram inicializadas no procedimento acima. Então  $Z$  é uma matriz triangular inferior e sua diagonal possui apenas uns. Ainda,  $\text{rank}(Z) = 2(K-N)$  e como o vetor nulo também pertence a  $\mathcal{P}_{cpl}(G)$ , concluímos que  $\dim(\mathcal{P}_{cpl}(G)) = 2(K-N)$ . ■

### 4.2.1 Coberturas

No capítulo anterior apresentamos o conceito de coberturas para o problema da mochila com restrições de precedência. Nesta seção apresentaremos a aplicação deste conceito aplicado ao problema

Figura 4.2: Grafo de restrição de precedência  $G = (V, E)$ 

$n \setminus k$	Níveis de injeção $q_{in}^{n,k}$						$\kappa(n)$	$\sigma_n$
	1	2	3	4	5	6		
1	0,5	2	3	4	5	6	6	0
2	1	2	3	4	5	5,5	6	0,5
3	1	2	3	4	5	6	6	0
4	1	2	3	4	4,5		5	1,5
5	1	2	3	3,5			4	2,5
6	1	2	3	4	5		5	1
7	1	2	3	4	5	6	6	0
8	1	2	2,5				3	3,5
9	1	2	3				3	3
10	1	2	3	4	5		5	1
11	1	1,5					2	4,5
12	1	1,5					2	4,5

Tabela 4.1: Taxa de injeção para os poços de petróleo

que estamos estudando. Iniciaremos com um exemplo ilustrativo, e em seguida o generalizaremos apresentando a definição formal.

Considere um cenário simples com 12 poços e o grafo de precedência descrito da Figura 4.2. Os valores de  $q_{in}^{n,k}$  da CPP linear por partes são apresentados na Tabela 4.1. A quantia total de gás de injeção disponível é  $q_{in}^{max} = 6$ . Os valores  $q_{out}^{n,k}$  não são necessários para a análise do poliedro  $conv(x)$ , e estão implícitos na forma linear por partes da função objetivo.

Dado o conjunto  $\Omega$  definido na seção anterior, tomamos o conjunto  $C \subset \Omega$ , com  $C = \{(1,2), (2,2), (3,2), (4,3), (5,3), (6,3)\}$ . Considere:

- i) Seja  $N(S) = \{n : (n,k) \in S\}$  o conjunto de poços com pares no subconjunto  $S \subseteq \Omega$ . Como  $N(C) = \{1, 2, 3, 4, 5, 6\}$ , todos os pares em  $C$  são de poços distintos.
- ii) Dado qualquer  $S \subseteq \Omega$ , seja  $I(S) = \cup_{n \in N(S)} (A_n \cup \{n\})$  o conjunto de todos os predecessores de  $N(S)$  incluindo os elementos de  $N(S)$ . Observe que  $I(C) = N(C)$  significa que o subgrafo

induzido  $G[N(C)]$  contém todos os nós que precedem nós de  $N(C)$ .

- iii) Para  $S \subseteq \Omega$ , seja  $H(S) = \{(n, k) \in S : \nexists (m, j) \in S \text{ tal que } n \prec m\}$  os pares de  $S$  em que os nós não precedam nenhum outro elemento de  $S$ . Note que  $H(C) = \{(4,3), (5,3), (6,3)\}$  e que para todo  $(n, k) \in C \setminus H(C) = \{(1,2), (2,2), (3,2)\}$  temos  $k = 2$ .
- iv) Para  $S \subseteq \Omega$ , seja  $\gamma(S) = \sum_{(n,k) \in S} q_{in}^{n,k-1}$  a quantidade mínima de gás necessário para ativar o poço  $n$  em um nível  $k$  ( $x_{n,k} = 1$ ) para todo  $(n, k) \in S$ . Observe que  $\gamma(C) = 0,5 + 1 + 1 + 2 + 2 + 2 = 8,5 > q_{in}^{max}$ .

Das considerações (i) a (iv), concluímos que os elementos de  $H(C)$  não podem ser ativados simultaneamente porque demandaria mais gás de injeção que o disponível. Isto significa que  $C$  é uma cobertura que induz a desigualdade válida  $\sum_{(n,k) \in H(C)} x_{n,k} \leq |H(C)| - 1$ . No exemplo, a desigualdade da cobertura é  $x_{4,3} + x_{5,3} + x_{6,3} \leq 2$ .

**Definição 4.1**  $C \subseteq \Omega$  é uma cobertura se:

- i.  $C$  possui no máximo um elemento  $(n, k) \in \Omega$  para cada  $n \in \mathcal{N}$ ;
- ii.  $l(C) = N(C)$ ;
- iii.  $k = 2$  para todo  $(n, k) \in C \setminus H(C)$ ;
- iv.  $\gamma(C) > q_{in}^{max}$ .

**Definição 4.2** Uma cobertura  $C$  é mínima se  $\gamma(C \setminus \{(n, k)\}) \leq q_{in}^{max}$  para cada  $(n, k) \in H(C)$ .

Uma cobertura é um conjunto de pontos de operação de poços em um campo de petróleo que seja realizável (condição i) e respeite a restrição de precedência (condições ii e iii), e que não exista recurso suficiente para que o campo opere nestes pontos de operação (iv). Uma cobertura é mínima se excluindo qualquer um dos pontos de operação de  $C$ , exista recurso suficiente para operar o conjunto restante.

#### 4.2.2 K-Coberturas

Dado um subconjunto de nós  $U \subseteq \mathcal{N}$ , seja  $\Omega(U) = \{(n, k) \in \Omega : n \in U\}$  o conjunto de pares “poço/nível de ativação” em que os poços estão em  $U$ . Para  $K = 2$ ,  $H(C)$  possui três conjuntos de cardinalidade  $K$ :  $S_{4,5} = \{(4,3), (5,3)\}$ ,  $S_{4,6} = \{(4,3), (6,3)\}$  e  $S_{5,6} = \{(5,3), (6,3)\}$ ; que induzem os conjuntos de precedência  $l(S_{4,5}) = \{1, 2, 3, 4, 5\}$ ,  $l(S_{4,6}) = \{1, 2, 3, 4, 6\}$  e  $l(S_{5,6}) = \{1, 2, 3, 5, 6\}$ . Estes conjuntos de precedência induzem subconjuntos de pares  $\Phi_{4,5}$ ,  $\Phi_{4,6}$  e  $\Phi_{5,6}$  dados por  $\Phi_{4,5} = \Omega(l(S_{4,5})) \cap C = \{(1,2), (2,2), (3,2), (4,3), (5,3)\}$ ,  $\Phi_{4,6} = \Omega(l(S_{4,6})) \cap C = \{(1,2), (2,2), (3,2), (4,3), (6,3)\}$  e  $\Phi_{5,6} = \Omega(l(S_{5,6})) \cap C = \{(1,2), (2,2), (3,2), (5,3), (6,3)\}$ . Observe que  $\gamma(\Phi_{4,5}) = \gamma(\Phi_{4,6}) = \gamma(\Phi_{5,6}) = 6,5 > 6 = q_{in}^{max}$ , implicando que  $\Phi_{4,5}$ ,  $\Phi_{4,6}$  e  $\Phi_{5,6}$  são cada um uma cobertura, conforme definido anteriormente. Como  $\gamma(\Phi_{m,n} \setminus \{(i, j)\}) = 4,5 < q_{in}^{max}$  para todo  $(m, n) \in \{(4,5),$

$(4,6), (5,6)\}$  e  $(i, j) \in S_{m,n}$ , verificamos que  $\Phi_{4,5}$ ,  $\Phi_{4,6}$  e  $\Phi_{5,6}$  são coberturas mínimas. Como qualquer subconjunto de  $H(C)$  com cardinalidade  $K = 2$  induz uma cobertura mínima,  $C$  é chamado de  $K$ -cobertura e implica na desigualdade válida:

$$\sum_{(n,k) \in H(C)} x_{n,k} = x_{4,3} + x_{5,3} + x_{6,3} \leq 1 = K - 1, \quad (4.5)$$

que é mais forte que a desigualdade anterior. O conceito  $K$ -coberturas é formalizado a seguir. Dado uma cobertura  $C$  e um subconjunto  $S \subseteq \Omega$ ,  $\Phi_S = \Omega(l(S)) \cap C$  denota pares de  $C$  cujos poços aparecem na lista de predecessores de  $S$ .

**Definição 4.3** *Uma cobertura  $C$  é uma  $K$ -cobertura se:*

- i.  $\gamma(\Phi_S) > q_{in}^{max}$  para todo  $S \subseteq H(C)$  com  $|S| = K$ , onde  $\Phi_S = \Omega(l(S)) \cap C$ ; mas
- ii.  $\gamma(\Phi_S \setminus \{(n,k)\}) \leq q_{in}^{max}$  para todo  $(n,k) \in S$ .

$C$  é uma  $K$ -cobertura se for uma cobertura e todo subconjunto  $S$  de  $C$  que respeite as restrições de precedência, e com cardinalidade  $K$ , for uma cobertura mínima de acordo com as Definições 4.1 e 4.2.

Em seguida, trataremos de um conjunto de pares expandidos de uma  $K$ -cobertura  $S$ , definido por  $\Gamma(S) = \{(n,i) : (n,k) \in S \text{ e } 2 \leq i \leq k\}$  que adiciona a  $S$  pares cujo poço está em  $S$  e cujo nível de ativação é menor que o de  $S$ . Para a cobertura  $C$  dada anteriormente,  $\Gamma(C) = \{(1,2), (2,2), (3,2), (4,2), (4,3), (5,2), (5,3), (6,2), (6,3)\}$ . Note que a desigualdade (4.5) é válida porque não existe recurso de injeção suficiente para simultaneamente ativar todos os pares de qualquer subconjunto de  $H(C)$  com cardinalidade  $K$ . A proposição a seguir formaliza esta desigualdade.

**Proposição 4.2** *Se  $C$  é uma  $K$ -cobertura então*

$$\sum_{(n,k) \in H(C)} x_{n,k} \leq K - 1 \quad (4.6)$$

é válida para o poliedro  $\text{conv}(X_C)$  onde  $X_C = \{(x, \lambda) \in X : x_{n,k} = \lambda_{n,k} = 0 \text{ para todo } (n,k) \in \Omega \setminus \Gamma(C)\}$  é o conjunto de soluções que podem ativar apenas pares de  $\Gamma(C)$ .

Esta proposição mostra que a desigualdade da  $K$ -cobertura é válida para a projeção de  $\text{conv}(X)$  no espaço expandido pelas variáveis de índices  $\Gamma(C)$ , que é o poliedro  $\text{conv}(X_C)$ . Obviamente, a desigualdade é válida para  $\text{conv}(X)$ .

Continuando o exemplo, considere  $\Phi_{4,5}$  de maneira que  $\gamma((\Phi_{4,5} \setminus \{(4,3)\}) \cup \{(4,2)\}) = 5,5 < q_{in}^{max}$  e  $\gamma((\Phi_{4,5} \setminus \{(5,3)\}) \cup \{(5,2)\}) = 5,5 < q_{in}^{max}$ . Isto significa que  $\Phi_{4,5}$  não é mais uma cobertura se qualquer elemento  $(n,k) \in H(\Phi_{4,5})$  for substituído por  $(n,k-1)$ , ou simplesmente removido quando  $k = 2$ . Esta propriedade também pode ser verificada para  $\Phi_{4,6}$  e  $\Phi_{5,6}$ . Uma  $K$ -cobertura é *estrita* se esta propriedade é satisfeita por cada cobertura induzida  $\Phi_S$ , onde  $S \subseteq H(C)$ ,  $|S| = K$ , e  $\Phi_S = \Omega(l(S)) \cap C$ .

**Definição 4.4** Uma  $K$ -cobertura  $C$  é estrita se, para todo  $S \subseteq H(C)$ ,  $|S| = K$ , e todo  $(n, k) \in S$  valer:

- i.  $\gamma(\Phi_S \setminus \{(n, k)\}) < q_{in}^{max}$  se  $k = 2$ ; ou
- ii.  $\gamma((\Phi_S \setminus \{(n, k)\}) \cup \{(n, k-1)\}) < q_{in}^{max}$  quando  $k > 2$ .

$K$ -coberturas estritas possuem a propriedade de serem máximas ou facetas de  $conv(X_C)$ . Neste sentido elas são o mais forte possível, pois um poliedro é representado unicamente por facetas. Antes de apresentarmos as condições de maximalidade, definimos uma solução  $z^S = (x^S, \lambda^S) \in X$  induzida por  $S \subseteq \Omega$  como: para todo  $(n, k) \in \Omega$ ,  $x_{n,k}^S = 1$  e  $\lambda_{n,k}^S = 1$  se  $(n, k) \in S$ , caso contrário  $x_{n,k}^S = 0$  e  $\lambda_{n,k}^S = 0$ .

**Proposição 4.3** Dado uma  $K$ -cobertura estrita  $C$ , a desigualdade (4.6) induz uma face máxima  $\mathcal{F}_C = \{(x, \lambda) \in conv(X_C) : \sum_{(n,k) \in H(C)} x_{n,k} = K-1\}$  de  $conv(X_C)$  se, e somente se:

$$\bigcap_{\{S \subseteq H(C) : |S|=K-1\}} \Omega(I(S)) \cap C = \emptyset \quad (4.7)$$

**Prova:** (Necessidade) Suponha que (4.7) não é obedecida e seja  $(m, i) \in \bigcap_{\{S \subseteq H(C) : |S|=K-1\}} \Omega(I(S)) \cap C$ . Para qualquer  $T \subseteq C$  tal que  $z(T) \in \mathcal{F}_C$ ,  $(m, i)$  deve aparecer em  $T$ . Então todo  $z = (x, \lambda) \in \mathcal{F}_C$  deve satisfazer  $x_{m,i} = 1$ . Como  $\mathcal{F}_C \subseteq \mathcal{F} = \{(x, \lambda) \in conv(X_C) : x_{m,i} = 1\}$ , concluímos que para  $\mathcal{F}_C$  ser máxima,  $\mathcal{F}_C = \mathcal{F}$ , mas (4.6) não é um múltiplo escalar de  $x_{m,i} \leq 1$ , contradizendo a suposição.

(Suficiência) Para mostrar que (4.6) induz uma faceta, seja  $F_\pi$  uma face máxima de  $\mathcal{P}_C$  contendo  $F_C$  e induzida por:

$$\sum_{(n,k) \in \Gamma(C)} \pi_{n,k} x_{n,k} + \sum_{(n,k) \in \Gamma(C)} \mu_{n,k} \lambda_{n,k}^L \leq \pi_0 \quad (4.8a)$$

Mostraremos que (4.6) e (4.8a) diferem apenas de uma constante multiplicativa, que prova a proposição.

[Parte I] Considere qualquer  $(n, k) \in A(C) = C \setminus H(C)$ . Tem que existir  $T \subseteq H(C)$  com  $|T| = K-1$  tal que  $n \in I(T)$ . Seja  $\Phi_T = \Omega(I(T)) \cap C$ . Claramente  $z^{\Phi_T} \in \mathcal{F}_C$ . Seja  $z^{\Phi'_T}$  idêntico a  $z^{\Phi_T}$  exceto que  $\lambda_{n,k}^{\Phi'_T} = 1 - \varepsilon$ . Para um  $\varepsilon > 0$  suficientemente pequeno,  $z^{\Phi'_T} \in \mathcal{F}_C$  pois  $C$  é uma  $K$ -cobertura estrita. Portanto, para  $z^{\Phi_T}$  e  $z^{\Phi'_T}$  pertencerem a  $\mathcal{F}_\pi$ ,  $z^{\Phi_T}$  e  $z^{\Phi'_T}$  devem satisfazer:

$$\sum_{(n,k) \in \Gamma(C)} \pi_{n,k} x_{n,k} + \sum_{(n,k) \in \Gamma(C)} \mu_{n,k} \lambda_{n,k} = \pi_0 \quad (4.8b)$$

Subtraindo (4.8b) com  $z^{\Phi'_T}$  de (4.8b) com  $z^{\Phi_T}$  concluímos que  $\mu_{n,k} = 0$ .

[Parte II] Considere qualquer  $(n, k) \in H(C)$  e faça  $T \subseteq H(C)$  de maneira que  $(n, k) \in T$  e  $|T| = K-1$ . Claramente  $z^{\Phi_T} \in \mathcal{F}_C$  para  $\Phi_T = \Omega(I(T)) \cap C$ . Seja  $z^{\Phi'_T}$  idêntico a  $z^{\Phi_T}$  exceto que  $\lambda_{n,k}^{\Phi'_T} = 1 - \varepsilon$ . Com  $\varepsilon > 0$  pequeno o suficiente,  $z^{\Phi'_T} \in \mathcal{F}_C$ . Como  $z^{\Phi_T}$  e  $z^{\Phi'_T}$  devem satisfazer (4.8b), concluímos que  $\mu_{n,k} = 0$ . Agora considere algum  $k' < k$ . Seja  $T \subseteq H(C) \setminus \{(n, k)\}$  um sub-conjunto de maneira que  $|T| = K-1$ . Claramente,  $z^\Phi \in \mathcal{F}_C$  para  $\Phi = (\Omega(I(T) \cup A_n) \cap C) \cup \{(n, k')\}$  porque  $C$  é uma

$K$ -cobertura estrita. Seja  $z^{\Phi'}$  idêntico a  $z^{\Phi}$  exceto que  $\lambda_{n,k}^{\Phi'} = 1 - \varepsilon$ . Para um  $\varepsilon > 0$  suficientemente pequeno,  $z^{\Phi'} \in \mathcal{F}_C$ . Para  $z^{\Phi}, z^{\Phi'}$  pertencerem a  $\mathcal{F}_C$ ,  $z^{\Phi}$  e  $z^{\Phi'}$  devem satisfazer (4.8b), que nos leva a deduzir que  $\mu_{n,k} = 0$ . Do desenvolvimento acima, concluímos que  $\mu_{n,k} = 0$  para todo  $(n,k) \in \Gamma(C)$  e, então, (4.8a) se reduz a:

$$\sum_{(n,k) \in \Gamma(C)} \pi_{n,k} x_{n,k} \leq \pi_0 \quad (4.8c)$$

[Parte III] Considere qualquer  $(n,k) \in A(C)$ . Como (4.7) é válido, deve existir  $T \subseteq H(C)$  com  $|T| = K - 1$  de maneira que  $n \notin l(T)$  e  $(m,j) \in H(C) \setminus T$  com  $n \in A_m$ . Pela definição de  $K$ -cobertura,  $\gamma(\Phi) \leq q_{in}^{max}$  para  $\Phi = \Omega(l(T) \cup A_m) \cap C$ , e uma vez que  $(A_n \cup \{n\}) \subseteq A_m$ , temos: (i)  $\gamma(\Phi') \leq q_{in}^{max}$  para  $\Phi' = \Omega(l(T) \cup A_n \cup \{n\}) \cap C$ ; e (ii)  $\gamma(\Phi'') \leq q_{in}^{max}$  para  $\Phi'' = \Phi' \setminus \{(n,k)\}$ . Portanto,  $z(\Phi')$  e  $z(\Phi'')$  pertencem a  $\mathcal{F}_C$ , e para tê-los em  $\mathcal{F}_\pi$ , devem satisfazer (4.8c) na igualdade. Que nos leva a concluir que  $\pi_{n,k} = 0$ .

[Parte IV] Seja  $(n,k) \in \Gamma(C) \setminus (A(C) \cup H(C))$ . Como  $C$  é uma  $K$ -cobertura estrita, deve existir  $T \subseteq H(C)$  com  $|T| = K - 1$  de maneira que: (i)  $n \notin N(T)$ ; (ii)  $\gamma(\Phi') \leq q_{in}^{max}$  para  $\Phi' = \Omega(l(T) \cup A_n) \cap C$ ; e (iii)  $\gamma(\Phi'') \leq q_{in}^{max}$  para  $\Phi'' = \Phi' \cup \{(n,k)\}$ . Claramente,  $z^{\Phi'}$  e  $z^{\Phi''}$  pertencem a  $\mathcal{F}_C$ , e assim (4.8c) deve ser satisfeita na igualdade, implicando que  $\pi_{n,k} = 0$ . Deduzimos então que (4.8c) se reduz a:

$$\sum_{(n,k) \in H(C)} \pi_{n,k} x_{n,k} \leq \pi_0 \quad (4.8d)$$

[Parte V] Considere qualquer par de pontos distintos  $(n,i), (m,j) \in H(C)$  e faça  $S \subseteq H(C) \setminus \{(n,i), (m,j)\}$  de maneira que  $|S| = K - 2$ . Pela definição da  $K$ -cobertura,  $\gamma(\Phi') \leq q_{in}^{max}$  para  $\Phi' = \Omega(l(S \cup \{(n,i)\})) \cap C$  e  $\gamma(\Phi'') \leq q_{in}^{max}$  para  $\Phi'' = \Omega(l(S \cup \{(m,j)\})) \cap C$ . Claramente,  $z^{\Phi'}$  e  $z^{\Phi''}$  pertencem a  $\mathcal{F}_C$ . Uma vez que  $z^{\Phi'}$  e  $z^{\Phi''}$  devem pertencer a  $\mathcal{F}_\pi$ , eles devem satisfazer (4.8d) na igualdade, que nos leva a deduzir que  $\pi_{n,i} = \pi_{m,j}$ . Assumindo, sem perda de generalidade,  $\pi_{n,k} = 1$  para todo  $(n,k) \in H(C)$ , temos que  $\pi_0 = K - 1$ . Consequentemente, (4.6) difere de (4.8a) apenas de uma constante multiplicativa, demonstrando que (4.6) é máxima. ■

Para a  $K$ -cobertura estrita  $C$  do exemplo ilustrativo, note que  $\bigcap_{\{S \subseteq H(C): |S|=K-1\}} \Omega(l(S)) \cap C = \bigcap_{\{S \subseteq \{(4,3), (5,3), (6,3)\}: |S|=1\}} \Omega(l(S)) \cap C = \Omega(l(\{(4,3)\})) \cap \Omega(l(\{(5,3)\})) \cap \Omega(l(\{(6,3)\})) \cap C = \emptyset$ . Como  $C$  satisfaz a condição (4.7), temos que (4.5) induz uma faceta de  $\text{conv}(X_C)$ .

Uma  $K$ -cobertura estrita  $C$  é chamada de *cobertura (estritamente) mínima* se  $|H(C)| = K$ . É evidente da definição de  $K$ -cobertura que todo  $S \subseteq H(C)$  com  $|S| = K$  induz uma cobertura mínima  $\Phi_S = \Omega(l(S)) \cap C$ . No exemplo ilustrativo, a  $K$ -cobertura  $C$  não é mínima, pois  $H(C) = 3 > 2 = K$ . Com  $S = \{(4,3), (5,3)\}$  a cobertura  $\Phi_S$  é mínima, pois  $|H(\Phi_S)| = K$ .

Uma  $K$ -cobertura  $C$  não mínima é mais forte que qualquer cobertura mínima induzida  $\Phi_S$  no sentido que  $\sum_{(n,k) \in H(C)} x_{n,k} \leq K - 1 \implies \sum_{(n,k) \in S} x_{n,k} \leq K - 1$ . No exemplo, qualquer desigualdade  $x_{4,3} + x_{5,3} \leq 1$ ,  $x_{4,3} + x_{6,3} \leq 1$  e  $x_{5,3} + x_{6,3} \leq 1$ , obtida de uma cobertura mínima  $\Phi_S$  é claramente mais



fraca que (4.5). É necessário aplicar um procedimento de arredondamento  $(2x_{4,3} + 2x_{5,3} + 2x_{6,3} \leq 3)/2 \implies x_{4,3} + x_{5,3} + x_{6,3} \leq \lfloor 3/2 \rfloor \implies x_{4,3} + x_{5,3} + x_{6,3} \leq 1$  para obter (4.5).

### 4.3 Fatores de *lifting* para $K$ -coberturas

Mostramos que uma  $K$ -cobertura estrita induz uma face máxima (faceta) de  $\text{conv}(X_C)$ , que é a projeção de  $\text{conv}(X)$  no espaço de variáveis com índices em  $\Gamma(C)$ . Esta desigualdade também é válida para  $\text{conv}(X)$ , entretanto podemos torná-la mais forte, estendendo-a para todos os pares de  $\Omega$ :

$$\sum_{(n,k) \in H(C)} x_{n,k} + \sum_{(n,k) \in \Omega \setminus \Gamma(C)} \alpha_{n,k} x_{n,k} \leq K - 1 \quad (4.9)$$

onde  $\alpha_{n,k}$  são os fatores de *lifting*, que dependem da cobertura e da ordem na qual são calculados.

Para exemplificar, assuma o exemplo ilustrativo e a desigualdade da  $K$ -cobertura (4.5) induzida por  $C = \{ (1,2), (2,2), (3,2), (4,3), (5,3), (6,3) \}$ . O par  $(n,k') = (4,4)$  requer ao menos  $\min\{q_{in}^{n,k-1}, q_{in}^{n,k}\} = \min\{q_{in}^{4,3}, q_{in}^{4,4}\} = \min\{3,4\} = 3$  unidades de gás para ser ativado. Como esta quantia é maior que o requerido por  $(n,k) = (4,3) \in H(C)$ , e no máximo um destes pares pode ser ativado, a variável  $x_{4,4}$  pode ser adicionado ao lado esquerdo da desigualdade (4.5). Neste raciocínio, podemos adicionar as variáveis  $x_{4,5}$ ,  $x_{5,4}$  e  $x_{6,5}$  à desigualdade (4.5). Continuando a análise, o par  $(n,k') = (1,6)$  requer que ao menos  $\min\{q_{in}^{n,k-1}, q_{in}^{n,k}\} = \min\{q_{in}^{1,5}, q_{in}^{1,6}\} = 5$ . Esta quantidade é suficiente para ativar o poço  $n$  no nível  $k = 2$ , que requer 1 unidade, enquanto as 4 unidades restantes podem ativar os pares  $(2,2)$ ,  $(3,2)$  e  $(5,3)$  que requerem 1, 1 e 2 unidades de gás de injeção respectivamente. Portanto, a ativação do par  $(1,6)$  requer tanto recurso quanto ativar  $(5,3) \in H(C)$ , e assim, a variável  $x_{1,6}$  é adicionada ao lado esquerdo de (4.5).

Os fatores de *lifting* são obtidos resolvendo uma seqüência  $\mathcal{X} = \langle \mathcal{X}_t : t = 1, \dots, T \rangle$  de problemas  $P_{cpl}(G)$  definidos de forma recursiva por:

$$\mathcal{X}_t : \zeta_t = \text{Maximize} \quad \sum_{(n,k) \in H(C)} x_{n,k} + \sum_{j=1}^{t-1} \alpha_{n_j, k_j} x_{n_j, k_j}$$

Sujeito a:  $x \in X_{U_t | W_t}$

para  $t = 1, \dots, T$ , onde:

- $X_{U|W} = \{x : (x, \lambda) \in X, x_{n,k} = 0 \text{ para todo } (n,k) \in U, \text{ enquanto } x_{n,k} = 1 \text{ para todo } (n,k) \in W\}$  é o conjunto restrito de soluções de  $X$ ;
- $U_t = \{(n_{t+1}, k_{t+1}), \dots, (n_T, k_T)\}$  é o conjunto de elementos a serem submetidos ao procedimento de *lifting*; e
- $W_t = \{(n_t, k_t)\}$  é o conjunto com o elemento que está sendo executado o *lifting*.

Os coeficientes de *lifting* são definidos recursivamente por  $\alpha_{n_j, k_j} = K - 1 - \zeta_t$ ,  $j = 1, \dots, t - 1$ . Para que este procedimento seja bem definido é necessário que  $X_{U|W}$  não seja vazio para todo  $t = 1, \dots, T$ . Para isto os fatores devem ser calculados segundo uma ordem topológica  $C' = \{(n_1, k_1), \dots, (n_T, k_T)\}$  de  $\Omega \setminus \Gamma(C)$ . Dizemos que  $C'$  está ordenado topologicamente se  $n_j \not\leq n_i$  para todo  $i, j$ , com  $1 \leq i < j \leq T$ .

Seja  $(C'_<, C'_\leq)$  uma partição de  $C'$ ,  $C'_< \cup C'_\leq = C'$  e  $C'_< \cap C'_\leq = \emptyset$ , tal que  $(n_t, k_t) \in C'$  pertence a  $C'_<$  se e somente se existe  $x_t \in X_{U_t|W_t}$  que resolva  $\mathcal{X}_t$  e tal que:

$$\sum_{(n,k) \in \Omega \setminus U_t} q_{in}^{n,k-1} x_{n,k}^t < q_{in}^{max}$$

**Proposição 4.4** *Dado uma  $K$ -cobertura estrita e uma seqüência de lifting topologicamente ordenada  $C'$ , a desigualdade de lifting (4.9) é válida para  $conv(X)$ .*

**Prova:** Por indução em  $t$ . (*Caso base*) Para  $t = 0$ , (4.9) é reduzida à desigualdade da  $K$ -cobertura (4.3) que é válida por definição. (*Passo de indução*) Considere  $t > 0$  e qualquer solução  $(x, \lambda) \in conv(X)$ . Se  $x_{n_t, k_t} = 0$  então (4.9) é válida por indução. Se  $x_{n_t, k_t} = 1$  então (4.9) é válida apenas se:

$$\sum_{(n,k) \in H(C)} x_{n,k} + \sum_{j=1}^{t-1} \alpha_{n_j, k_j} x_{n_j, k_j} + \alpha_{n_t, k_t} \leq K - 1$$

que, por sua vez, é válida apenas se:

$$\begin{aligned} \alpha_{n_t, k_t} &\leq K - 1 - \sum_{(n,k) \in H(C)} x_{n,k} - \sum_{j=1}^{t-1} \alpha_{n_j, k_j} x_{n_j, k_j} \\ &\leq K - 1 - \max \left\{ \sum_{(n,k) \in H(C)} x_{n,k}^t + \sum_{j=1}^{t-1} \alpha_{n_j, k_j} x_{n_j, k_j}^t : x^t \in X_{U_t|W_t} \right\} \\ &= K - 1 - \zeta_t = \alpha_{n_t, k_t} \end{aligned}$$

Portanto (4.9) é válida. ■

**Corolário 4.1** *Se a  $K$ -cobertura  $C$  é estrita, então a desigualdade (4.9) induz uma face  $\mathcal{F}_C = \{(x, \lambda) \in conv(X) : (x, \lambda) \text{ respeita (4.9) na igualdade}\}$  de  $conv(X)$  com  $dim(\mathcal{F}_C) = 2|\Omega| - |C'_\leq| - 1$ .*

**Prova:** Seja  $Z \in \mathbb{R}^{(2|\Omega| - |C'_\leq|) \times 2|\Omega|}$  uma matriz em que as colunas estão relacionadas a  $x_{n,k}$  e  $\lambda_{n,k}$ ,  $(n,k) \in \Omega$ . Sejam as colunas mais à esquerda as correspondentes às variáveis  $x_{n,k}$  e  $\lambda_{n,k}$ ,  $(n,k) \in \Gamma(C)$ , e as demais correspondentes a  $x_{n,k}$  e  $\lambda_{n,k}$ ,  $(n,k) \in \Omega \setminus \Gamma(C)$ , posicionadas da esquerda para a direita em uma ordem topológica dada por  $C'$ .

Sejam as  $2|\Gamma(C)|$  linhas mais acima de  $Z$  correspondentes às  $2|\Gamma(C)|$  soluções afim independentes pertencentes a  $\mathcal{F}_C$  de maneira que  $x_{n,k} = 0$  para todo  $(n,k) \in \Omega \setminus \Gamma(C)$  que existem de acordo com a Proposição 4.3. Faça as demais linhas do topo à base serem obtidas como segue. Para  $t = 1, \dots, T$ , a próxima linha de  $Z$  é a solução  $x^t$  para  $\mathcal{X}_t$ ; se  $(n_t, k_t) \in C'_<$ , então adicione outra linha abaixo, idêntica

$n \backslash k$	Fatores de <i>lifting</i> $\alpha_{n,k}$				
	2	3	4	5	6
1		0	0*	1	1
2		0	0	1	1
3		0	0	0*	1
4			1	1	
5			1		
6			1	1	
7	0	0	0*	1	1
8	0	1			
9	0	0*			
10	0	0*	1	1	
11	1				
12	1				

Tabela 4.2: Fatores de *lifting* para o exemplo da mochila

à linha de  $x^t$ , exceto que  $\lambda_{n_t, k_t} = 1 - \varepsilon$  com  $\varepsilon > 0$  suficientemente pequeno para que a linha seja um elemento de  $\mathcal{F}_C$ . Note que  $x_{n_t, k_t} = \lambda_{n_t, k_t}$  para todo  $(x, \lambda) \in \mathcal{F}_C$  se  $(n_t, k_t) \in C'_=$ .

Claramente, as linhas  $2|\Gamma(C)| + 1, \dots, 2|\Gamma(C)| + T + |C'_<|$  de  $Z$  formam uma matriz bloco-diagonal, com as  $2|\Gamma(C)|$  linhas mais acima linearmente independentes. Portanto,  $\dim(\mathcal{F}_C) \geq 2|\Gamma(C)| + T + |C'_<| - 1 = 2|\Gamma(C)| + |\Omega \setminus \Gamma(C)| + |C'_<| - 1 = |\Omega| + |\Gamma(C)| + |C'_<| - 1 = |\Omega| + |\Omega| - |C'_=| - 1 = 2|\Omega| - |C'_=| - 1$ . Como  $x_{n,k} = \lambda_{n,k}$  para todo  $(x, \lambda) \in \mathcal{F}_C$  se  $(n, k) \in C'_=$ , temos que  $\dim(\mathcal{F}_C) \leq 2|\Omega| - |C'_=| - 1$ , que prova a proposição. ■

Por conveniência, o problema de *lifting*  $\mathcal{X}_t$  pode ser reescrito como um problema de programação inteira tornando  $\mathcal{X}_{U_t|W_t}$  explícito:

$$\mathcal{X}_t' : \zeta_t = \text{Maximize} \quad \sum_{(n,k) \in H(C)} x_{n,k} + \sum_{j=1}^{t-1} \alpha_{n_j, k_j} x_{n_j, k_j} \quad (4.10a)$$

$$\text{Sujeito a} \quad \sum_{n \in N_t} \sum_{(n,k) \in S_{t,n}} q_{in}^{n,k-1} x_{n,k} \leq q_{in}^{\max} \quad (4.10b)$$

$$\sum_{(m,i) \in S_{t,m}} x_{m,i} \geq \sum_{(n,j) \in S_{t,n}} x_{n,j} \quad \forall (m,n) \in E(G[N_t]) \quad (4.10c)$$

$$\sum_{(n,k) \in S_{t,n}} x_{n,k} \leq 1 \quad \forall n \in R(G[N_t]) \quad (4.10d)$$

$$x_{n_t, k_t} = 1 \quad (4.10e)$$

$$x_{n,k} \in \{0, 1\} \quad \forall (n,k) \in \cup_{n \in N_t} S_{t,n} \quad (4.10f)$$

onde  $N_t = N(C) \cup \{n_1, \dots, n_t\}$  e  $S_{t,n} = (\Gamma(C) \cup \{(n_1, k_1), \dots, (n_t, k_t)\}) \cap \Omega(\{n\})$ .

*Exemplo Ilustrativo:* Para aplicamos o procedimento de *lifting* sobre a desigualdade (4.5),  $C'$  é obtida arranjando os elementos de  $\Omega \setminus \Gamma(C)$  em ordem lexicográfica: para todo  $(m, i), (n, j) \in C'$ ,  $(m, i) \prec (n, j)$  se  $m < n$ , e caso  $m = n$  então  $i < j$ , induzindo uma ordem topológica. Os fatores de *lifting* obtidos são apresentados na Tabela 4.2. As posições da tabela assinaladas com uma estrela

pertencem a  $C'_=$ , e os demais a  $C'_<$ , portanto  $|C'_|= 5$  e  $|C'_<|= 27$ . A dimensão da face  $\mathcal{F}_C$  definida pela desigualdade (4.9), considerando os fatores da tabela, é  $\dim(\mathcal{F}_C) = 2 \times 41 - 5 - 1 = 76$ , que é um pouco menor que 81, a dimensão da face máxima de  $\text{conv}(\mathcal{X})$ .

Para se obter cada fator de *lifting* exato  $\alpha_{n,k}$ , é necessário resolver um problema da mochila  $\mathcal{X}_G$ , que é NP-Difícil. Ou seja, para o caso geral, resolver  $\mathcal{X}_G$  é tão difícil quanto resolver  $P_{cpl}(G)$ . Entretanto, se o grafo de precedência é uma floresta,  $\mathcal{X}_G$  pode ser resolvido em tempo pseudo-polinomial utilizando programação dinâmica [9]. No Apêndice A apresentamos este algoritmo de programação dinâmica.

#### 4.4 Fatores de *pseudo-lifting* para $K$ -coberturas

A dificuldade de se encontrar fatores exatos de *lifting* pode ser contornada buscando fatores de *lifting* aproximados  $\beta_{n,k}$ , que chamaremos de *fatores de pseudo-lifting*. Primeiro introduziremos alguns conceitos utilizando o exemplo dado na Figura 4.2 e Tabela 4.1. Seja  $\bar{\Gamma}(C) = \Omega - \Gamma(C)$  o conjunto de índices das variáveis em que os fatores de *pseudo-lifting* serão obtidos. Este conjunto é dividido em três subconjuntos, dependendo se os poços pertencem a  $N(H(C))$ ,  $N(C \setminus H(C))$  ou  $\mathcal{X} \setminus N(C)$ :

- $\bar{\Gamma}_H(C) = \{(n, j) : (n, k) \in H(C), j = k + 1, \dots, \kappa(n)\}$  são os índices das variáveis relativos a  $H(C)$ ; no exemplo,  $C = \{(1,2), (2,2), (3,2), (4,3), (5,3), (6,3)\}$  implica  $\bar{\Gamma}_H(C) = \{(4,4), (4,5), (5,4), (6,4), (6,5)\}$ .
- $\bar{\Gamma}_A(C) = \{(n, j) : (n, k) \in C \setminus H(C), j = k + 1, \dots, \kappa(n)\}$  é o conjunto dos índices dos fatores de *pseudo-lifting* correspondentes aos predecessores  $C \setminus H(C)$ ; no exemplo ilustrativo,  $\bar{\Gamma}_A(C) = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,3), (3,4), (3,5), (3,6)\}$ .
- $\bar{\Gamma}_B(C) = \bar{\Gamma}(C) \setminus (\bar{\Gamma}_H(C) \cup \bar{\Gamma}_A(C))$  é o conjunto dos índices restantes; no exemplo,  $\bar{\Gamma}_B(C) = \Omega(\{7, 8, 9, 10, 11, 12\})$ .

No início da Seção 4.3, o exemplo sobre  $x_{4,4}$  ilustra precisamente a obtenção do fator de *pseudo-lifting*  $\beta_{n,k}$  para  $(n, k) \in \bar{\Gamma}_H(C)$ : a quantidade mínima de gás para ativar o poço  $n$  no nível  $k$  é  $q_{in}^{n,k-1}$ ; como  $q_{in}^{n,k-1} = q_{in}^{4,3} = 3 > 2 = q_{in}^{4,2} = q_{in}^{n,k'-1}$  com  $(n, k') = (4, 3) \in H(C)$ ,  $\beta_{4,4} = 1$  e a variável  $x_{4,4}$  pode ser adicionada à desigualdade da  $K$ -cobertura.

Similarmente, a discussão sobre  $x_{1,6}$  mostra o cômputo de  $\beta_{n,k}$  para  $(n, k) \in \bar{\Gamma}_A(C)$ : note que a quantidade mínima de gás para ativar o poço 1 no nível 6 é  $q_{in}^{n,k-1} = q_{in}^{1,5} = 5$ , que é suficiente para ativar este poço no nível 2 e o par de maior demanda de recursos  $(m, j) \in H(C)$  com seus predecessores, que são os pares  $(5,3)$ ,  $(2,2)$  e  $(3,2)$  que requerem 4 unidades; portanto  $\beta_{1,6} = 1$  permitindo que  $x_{1,6}$  seja adicionada a desigualdade da  $K$ -cobertura.

Resta ilustrar o procedimento de *pseudo-lifting* para os elementos de  $(n, k) \in \bar{\Gamma}_B(C)$ . Para este conjunto, o cômputo é mais complicado pois as restrições de precedência devem ser levadas em conta devido a uma liberdade de escolha adicional. Considere um caso simples onde  $(n, k) = (7, 6) \in \bar{\Gamma}_B(C)$ .

Como o nó 7 é uma folha de  $G$ , a desativação do poço 7 não viola restrições de precedência. Note que  $q_{in}^{n,k-1} = q_{in}^{7,5} = 5$ , que é suficiente para ativar o maior elemento de  $H(C)$  e seus predecessores:  $(5,3) \in H(C)$  e  $(1,2)$ ,  $(2,2)$  e  $(3,2)$ , consumindo juntos  $2 + 1 + 1 + 1 = 5$  unidades de gás de injeção. Consequentemente, o fator de *pseudo-lifting*  $\beta_{7,5} = 1$  e  $x_{7,5}$  pode ser adicionada a desigualdade da  $K$ -cobertura.

Algumas definições serão introduzidas a seguir com o objetivo de formalizar as noções dadas.

- para qualquer  $n \in \mathcal{N}$  e  $(m, j) \in \Omega \setminus \Omega(\{n\})$ ,

$$\sigma(n)_{m,j} = \begin{cases} q_{in}^{m,j-1} + \sum_{l \in A_m \setminus (A_n \cup \{n\})} q_{in}^{l,1} & \text{se } m \not\prec n \\ q_{in}^{m,j-1} - q_{in}^{m,1} & \text{se } m \prec n \end{cases}$$

é a quantidade mínima de recursos necessários para ativar o poço  $m$  em um nível  $j$ , sendo que o poço  $n$  foi ativado;

- $H(C, n, k) = H(C) \setminus \{(n, k)\}$ , com  $(n, k) \in H(C)$ ;
- $H(C, n, k)_h \subseteq H(C, n, k)$  é o conjunto onde:

(i)  $|H(C, n, k)_h| = h$ ; e

(ii)  $\min\{\sigma(n)_{m,j} : (m, j) \in H(C, n, k)_h\} \geq \max\{\sigma(n)_{m,j} : (m, j) \in H(C, n, k) \setminus H(C, n, k)_h\}$ ,

isto é,  $H(C, n, k)_h$  é o subconjunto de  $H(C, n, k)$  com  $h$  pares que requerem a maior quantidade de recursos para ser ativado, dado que o poço  $n \in N(H(C))$ , está ativo; e

- $H(C, n)_h \subseteq H(C)$  é tal que:

(i)  $|H(C, n)_h| = h$  e

(ii)  $\min\{\sigma(n)_{m,j} : (m, j) \in H(C, n)_h\} \geq \max\{\sigma(n)_{m,j} : (m, j) \in H(C) \setminus H(C, n)_h\}$ ,

isto é,  $H(C, n)_h$  é o subconjunto de  $H(C)$  com os  $h$  pares que requerem a maior quantidade de recursos para ser ativado, dado que o par  $(n, j)$  está ativo.

Para os subconjuntos definidos acima,  $\bar{\Gamma}_H(C)$ ,  $\bar{\Gamma}_A(C)$  e  $\bar{\Gamma}_B(C)$ , apresentamos a seguir fatores de *pseudo-lift*. Para um poço  $n$  de uma cobertura  $C$ , seja  $\delta(n)$  o nível de ativação correspondente: se  $n \in N(C)$  então  $\delta(n) = k$  onde  $(n, k) \in C$ .

**Definição 4.5** Para  $(n, k) \in \bar{\Gamma}_H(C)$ , o fator de *pseudo-lift* é:

$$\beta_{n,k} = 1 + \max\{h : \sum_{(m,j) \in H(C, n, \delta(n))_h} \sigma(n)_{m,j} \leq q_{in}^{n,k-1} - q_{in}^{n, \delta(n)-1}\}$$

onde  $(n, \delta(n)) \in H(C)$ .

**Definição 4.6** Para  $(n, k) \in \bar{\Gamma}_A(C)$ , o fator de pseudo-lift é:

$$\beta_{n,k} = \max\{h : \sum_{(m,j) \in H(C,n)_h} \sigma(n)_{m,j} \leq q_{in}^{n,k-1} - q_{in}^{n,1}\}$$

Com respeito aos elementos de  $\bar{\Gamma}_B(C)$ , existe um procedimento para o cômputo de seus fatores aproximados. Seja  $\bar{\Theta}_0, \bar{\Theta}_1, \dots, \bar{\Theta}_R$  uma partição de  $\bar{\Gamma}_B(C)$  tal que para cada  $r = 1, 2, \dots, R$  temos:

- (i)  $H(\bar{\Theta}_r) = \{(n_r, 2), \dots, (n_r, \kappa(n_r))\}$  para algum  $n_r \in \mathcal{X} \setminus N(C)$ ;
- (ii)  $m \prec n_r$  e  $j = 2$  para todo  $(m, j) \in \bar{\Theta}_r \setminus H(\bar{\Theta}_r)$ .

**Definição 4.7** Dado uma partição  $\{\bar{\Theta}_r\}_0^R$  de  $\bar{\Gamma}_B(C)$ , os fatores de pseudo-lifting  $\beta_{n,k}$  para  $(n, k) \in \bar{\Theta}_r$  são definidos como:

- $\beta_{n,k} = \max\{h : \sum_{(m,j) \in H(C,n)_h} \sigma(n)_{m,j} \leq q_{in}^{n,k-1} - q_{in}^{n,1}\}$  se  $r = 0$ ;
- $\beta_{n,k} = \max\{h : \sum_{(m,j) \in H(C,n)_h} \sigma(n)_{m,j} \leq q_{in}^{n,k-1} + \sum_{(m,j) \in \bar{\Theta}_r \setminus H(\bar{\Theta}_r)} q_{in}^{m,j-1}\}$  se  $r \geq 1$  e  $(n, k) \in H(\bar{\Theta}_r)$ ;
- $\beta_{n,k} = 0$  se  $r \geq 1$  e  $(n, k) \in \bar{\Theta}_r \setminus H(\bar{\Theta}_r)$ .

**Proposição 4.5** Dado uma  $K$ -cobertura  $C$  e os fatores de pseudo-lifting  $\beta_{n,k}$ , a desigualdade de pseudo-lifting:

$$\sum_{(n,k) \in H(C)} x_{n,k} + \sum_{(n,k) \in \bar{\Gamma}(C)} \beta_{n,k} x_{n,k} \leq K - 1 \quad (4.11)$$

é válida para  $\text{conv}(X)$ .

**Prova:** (Por contradição) Suponha que existe um  $(x, \lambda) \in \text{conv}(X)$  que viola (4.11), e seja:

- a)  $H(x) = \{(n, k) \in H(C) : x_{n,k} = 1\}$ ;
- b)  $\bar{H}(x) = \{(n, k) \in \bar{\Gamma}_H(C) : x_{n,k} = 1\}$  e  $\bar{H}(x)_t = \{(n, k) \in \bar{H}(x) : \beta_{n,k} = t\}$ ;
- c)  $\bar{A}(x) = \{(n, k) \in \bar{\Gamma}_A(C) : x_{n,k} = 1\}$  e  $\bar{A}(x)_t = \{(n, k) \in \bar{A}(x) : \beta_{n,k} = t\}$ ; e
- d)  $\bar{B}(x) = \{(n, k) \in \bar{\Gamma}_B(C) : x_{n,k} = 1\}$  e  $\bar{B}(x)_t = \{(n, k) \in \bar{B}(x) : \beta_{n,k} = t\}$ .

Note que como  $(x, \lambda)$  viola (4.11),

$$|H(x)| + \sum_{t=1}^{K-1} t (|\bar{H}(x)_t| + |\bar{A}(x)_t| + |\bar{B}(x)_t|) > K - 1$$

Seja  $H_0(x) = H(C) \setminus (H(x) \cup \{(n, \delta(n)) : (n, k) \in \overline{H}(x)\})$ . E considere  $(\tilde{x}, \tilde{\lambda})$  idêntica a  $(x, \lambda)$ , com exceção apenas do redefinido nas Partes I, II, III e IV a seguir.

(Parte I) Seja  $\Phi \subseteq H_0(x)$  um subconjunto máximo com  $|\Phi| \leq \sum_{t=1}^{K-1} (t-1) |\overline{H}(x)_t|$ . Modifique  $(\tilde{x}, \tilde{\lambda})$  fazendo:

- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 0$  e  $\tilde{x}_{n,\delta(n)} = \tilde{\lambda}_{n,\delta(n)} = 1$  para todo  $(n, k) \in \overline{H}(x)$ ;
- $\tilde{x}_{n,j} = \tilde{\lambda}_{n,j} = 0$  para todo  $(n, k) \in \Phi$ ,  $j = 1, \dots, k-1$ ;
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \Phi$ ; e
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \{(m, j) \in C \setminus H(C) : \tilde{x}_{m,t} = 0 \text{ para todo } t\} \cap \Omega(l(\Phi))$ .

Como  $q_{in}^{n,k-1} - q_{in}^{n,\delta(n)-1} \geq \sum_{(m,j) \in H(C,n,\delta(n))_{t-1}} \sigma(n)_{m,j}$  para todo  $(n, k) \in \overline{H}(x)_t$  e todo  $t$ , temos que  $(\tilde{x}, \tilde{\lambda}) \in \text{conv}(X)$ . Note que  $|H(\tilde{x})| = |H(x)| + \sum_{t=1}^{K-1} t |\overline{H}(x)_t|$  porque, caso contrário,  $(\tilde{x}, \tilde{\lambda}) \notin \text{conv}(X) \Rightarrow (x, \lambda) \notin \text{conv}(X)$ .

(Parte II) Seja  $\Phi \subseteq H_0(\tilde{x})$  um subconjunto máximo onde  $|\Phi| \leq \sum_{t=1}^{K-1} t |\overline{A}(x)_t|$ . Modifique  $(\tilde{x}, \tilde{\lambda})$  da seguinte maneira:

- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 0$  e  $\tilde{x}_{n,2} = \tilde{\lambda}_{n,2} = 1$  para todo  $(n, k) \in \cup_{t=1}^{K-1} \overline{A}(x)_t$ ;
- $\tilde{x}_{n,j} = \tilde{\lambda}_{n,j} = 0$  para todo  $(n, k) \in \Phi$ ,  $j = 1, \dots, k-1$ ;
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \Phi$ ; e
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \{(m, j) \in C \setminus H(C) : \tilde{x}_{m,t} = 0 \text{ para todo } t\} \cap \Omega(l(\Phi))$ .

Como na Parte I,  $(\tilde{x}, \tilde{\lambda}) \in \text{conv}(X)$  pois  $q_{in}^{n,k-1} - q_{in}^{n,1} \geq \sum_{(m,j) \in H(C,n)_t} \sigma(n)_{m,j}$  para todo  $(n, k) \in \overline{A}(x)_t$  e todo  $t$ . Ainda,  $\Phi$  deve ser máximo porque, caso contrário,  $(\tilde{x}, \tilde{\lambda}) \notin \text{conv}(X) \Rightarrow (x, \lambda) \notin \text{conv}(X)$ . Consequentemente,  $|H(\tilde{x})| = |H(x)| + \sum_{t=1}^{K-1} t (|\overline{H}(x)_t| + |\overline{A}(x)_t|)$

(Parte III) Seja  $\Phi \subseteq H_0(\tilde{x})$  um subconjunto máximo com cardinalidade  $|\Phi| \leq \sum_{t=1}^{K-1} t |\overline{B}(x)_t \cap \overline{\Theta}_0|$ . Modifique  $(\tilde{x}, \tilde{\lambda})$  fazendo:

- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 0$  e  $\tilde{x}_{n,2} = \tilde{\lambda}_{n,2} = 1$  para todo  $(n, k) \in \cup_{t=1}^{K-1} (\overline{B}(x)_t \cap \overline{\Theta}_0)$ ;
- $\tilde{x}_{n,j} = \tilde{\lambda}_{n,j} = 0$  para todo  $(n, k) \in \Phi$ ,  $j = 1, \dots, k-1$ ;
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \Phi$ ; e
- $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \{(m, j) \in C \setminus H(C) : \tilde{x}_{m,t} = 0 \text{ para todo } t\} \cap \Omega(l(\Phi))$ .

Como  $q_{in}^{n,k-1} - q_{in}^{n,1} \geq \sum_{(m,j) \in H(C,n)_t} \sigma(n)_{m,j}$  para todo  $(n, k) \in \overline{B}(x)_t \cap \overline{\Theta}_0$  e todo  $t$ , temos que  $(\tilde{x}, \tilde{\lambda}) \in \text{conv}(X)$ . Note que  $|H(\tilde{x})| = |H(x)| + \sum_{t=1}^{K-1} t (|\overline{H}(x)_t| + |\overline{A}(x)_t| + |\overline{B}(x)_t \cap \overline{\Theta}_0|)$  porque, caso contrário,  $(\tilde{x}, \tilde{\lambda}) \notin \text{conv}(X) \Rightarrow (x, \lambda^L) \notin \text{conv}(X)$ .

(Parte IV) Seja  $\Phi \subseteq H_0(\tilde{x})$  um subconjunto máximo de cardinalidade  $|\Phi| \leq \sum_{t=1}^{K-1} t |\overline{B}(x)_t \cap \overline{\Theta}|$  onde  $\overline{\Theta} = \cup_{r=1}^R \overline{\Theta}_r$ . Modifique  $(\tilde{x}, \tilde{\lambda})$  fazendo:

$n \setminus (m, j)$	Valores $\sigma(n)_{m,j}$		
	(4,3)	(5,3)	(6,3)
1	3	4	3
2	2	3	3
3	3.5	3.5	2
4		3	3
5	2		2
6	3.5	3.5	
7	3.5	4.5	3
8	2	1	2
9	3.5	3.5	1
10	3.5	4.5	3
11	2	1	2
12	2	1	2

Tabela 4.3: Recursos mínimos de ativação

$n : (m, j)$	$H(C, n)_h$ e $H(C, n, k)_h$		
	$h = 1$	$h = 2$	$h = 3$
1	{(5,3)}	{(5,3),(4,3)}	{(5,3),(4,3),(6,3)}
2	{(6,3)}	{(6,3),(5,3)}	{(6,3),(5,3),(4,3)}
3	{(4,3)}	{(4,3),(5,3)}	{(4,3),(5,3),(6,3)}
(4,3)	{(5,3)}	{(5,3),(6,3)}	
(5,3)	{(4,3)}	{(4,3),(6,3)}	
(6,3)	{(4,3)}	{(4,3),(5,3)}	
7	{(5,3)}	{(5,3),(4,3)}	{(5,3),(4,3),(6,3)}
8	{(4,3)}	{(4,3),(6,3)}	{(4,3),(6,3),(5,3)}
9	{(4,3)}	{(4,3),(5,3)}	{(4,3),(5,3),(6,3)}
10	{(5,3)}	{(5,3),(4,3)}	{(5,3),(4,3),(6,3)}
11	{(4,3)}	{(4,3),(6,3)}	{(4,3),(6,3),(5,3)}
12	{(6,3)}	{(6,3),(4,3)}	{(6,3),(4,3),(5,3)}

Tabela 4.4: Subconjuntos para ativação de  $H(C)$ 

- a)  $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 0$  para todo  $(n, k) \in \Omega - \Omega(N(C))$ ;
- b)  $\tilde{x}_{n,j} = \tilde{\lambda}_{n,j} = 0$  para todo  $(n, k) \in \Phi, j = 1, \dots, k - 1$ ;
- c)  $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \Phi$ ; e
- d)  $\tilde{x}_{n,k} = \tilde{\lambda}_{n,k} = 1$  para todo  $(n, k) \in \{(m, j) \in C \setminus H(C) : \tilde{x}_{m,t} = 0 \text{ para todo } t\} \cap \Omega(l(\Phi))$ .

Considerando que  $q_{in}^{n,k-1} + \sum_{(m,j) \in \bar{\Theta}_r \setminus H(\bar{\Theta}_r)} q_{in}^{m,j-1} \geq \sum_{(m,j) \in H(C, n)_t} \sigma(n)_{m,j}$  para todo  $(n, k) \in \bar{\Theta}_r \cap \bar{B}(x)_t$ ,  $r$ , e  $t$ , temos  $(\tilde{x}, \tilde{\lambda}) \in \text{conv}(x)$ . Note que  $\sum_{(n,k) \in H(C)} \tilde{x}_{n,k} = H(\tilde{x}) \geq |H(x)| + \sum_{t=1}^{K-1} t(|\bar{H}(x)_t| + |\bar{A}(x)_t| + |\bar{B}(x)_t|) \geq K$  pois  $(x, \lambda)$  viola (4.11). Isto implica que  $(\tilde{x}, \tilde{\lambda}) \notin \text{conv}(x) \Rightarrow (x, \lambda) \notin \text{conv}(x)$ , contradizendo a hipótese inicial. ■

*Exemplo ilustrativo:* Continuando o exemplo anterior, iniciamos calculando as quantidades mínimas de recursos de ativação  $\sigma(n)_{m,j}$  que estão na Tabela 4.3. Os conjuntos  $H(C, n, k)_h$  e  $H(C, n)$  estão



$n \setminus k$	Coeficientes $\beta_{n,k}$ para $\{\overline{\Theta}'_r\}$					Coeficientes $\beta_{n,k}$ para $\{\overline{\Theta}''_r\}$				
	2	3	4	5	6	2	3	4	5	6
1		0	0	0*	1	0	0	0*	1	
2		0	0	1	1	0	0	1	1	
3		0	0	0	1	0	0	0	1	
4			1	1				1	1	
5			1					1		
6			1	1				1	1	
7	0	0	0	0*	1	0	0	0	0*	0*
8	0	1				0	0*			
9	0	0				0	0			
10	0	0	0*	0*		0	0	0*	1	
11	0*					1				
12	0*					0*				

Tabela 4.5: Coeficientes de *pseudo-lifting*

na Tabela 4.4. Os parâmetros destas tabelas serão necessários para o cálculo dos fatores de *pseudo-lifting*. Para a  $K$ -cobertura  $C = \{(1,2), (2,2), (3,2), (4,3), (5,3), (6,3)\}$ , os conjuntos de índices das variáveis cujos fatores de *lifting*  $\beta_{n,k}$  serão calculados são:

- $\overline{\Gamma}_H(C) = \{(4,4), (4,5), (5,4), (6,4), (6,5)\}$ ;
- $\overline{\Gamma}_A(C) = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,3), (3,4), (3,5), (3,6)\}$ ; e
- $\overline{\Gamma}_B(C) = \cup_{n=7}^{12} \Omega_n$ .

Considere a partição  $\{\overline{\Theta}'_r\}$  de  $\overline{\Gamma}_B(C)$  dada por  $\overline{\Theta}'_0 = \emptyset$ ,  $\overline{\Theta}'_1 = \Omega(\{7\})$  com  $n_1 = 7$ ,  $\overline{\Theta}'_2 = \Omega(\{8\})$  com  $n_2 = 8$ ,  $\overline{\Theta}'_3 = \Omega(\{9\})$  com  $n_3 = 9$ ,  $\overline{\Theta}'_4 = \Omega(\{10\})$  com  $n_4 = 10$ ,  $\overline{\Theta}'_5 = \Omega(\{11\})$  com  $n_5 = 11$ , e  $\overline{\Theta}'_6 = \Omega(\{12\})$  com  $n_6 = 12$ .

Considere também a partição  $\{\overline{\Theta}''_r\}$  dada por  $\overline{\Theta}''_0 = \{(7,3), (7,4), (7,5), (7,6), (8,3)\}$ ,  $\overline{\Theta}''_1 = \{(9,2), (9,3), (9,4)\}$  com  $n_1 = 9$ ,  $\overline{\Theta}''_2 = \{(7,2), (10,2), (10,3), (10,4), (10,5)\}$  com  $n_2 = 10$ ,  $\overline{\Theta}''_3 = \{(8,2), (11,2)\}$  com  $n_3 = 11$ , e  $\overline{\Theta}''_4 = \{(12,2)\}$  com  $n_4 = 12$ .

Os fatores de *pseudo-lifting* para os conjuntos  $\overline{\Gamma}_A(C)$  e  $\overline{\Gamma}_H(C)$  estão na Tabela 4.5 junto com os fatores para  $\overline{\Gamma}_B(C)$ , que variam de acordo com as partições  $\{\overline{\Theta}'_r\}$  e  $\{\overline{\Theta}''_r\}$ . As posições da tabela marcadas com uma estrela correspondem às instâncias cujos fatores de *pseudo-lifting*  $\beta_{n,k}$  são diferentes dos fatores exatos  $\alpha_{n,k}$ .

Podemos observar que os fatores de *pseudo-lifting* são bastante próximos dos fatores exatos, divergindo apenas em alguns pontos. Note também que a escolha das partições  $\{\overline{\Theta}'_r\}$  influencia nos resultados obtidos.

## 4.5 Resultados computacionais

Foram realizados testes computacionais para avaliar o impacto da aplicação de cortes sobre o poliedro  $\mathcal{P}_{cpl}(G)$  no desempenho do algoritmo de *Branch and Bound*. As curvas de performance dos poços utilizadas são baseadas nas de Buitrago *et al.* [4], em que o campo em estudo possui 56 poços, com várias configurações de CPPs (uma CPP para cada poço), algumas com resposta imediata (representando poços surgentes) enquanto outras com a necessidade de uma injeção mínima de gás para iniciar a produção. A forma linear por partes destas curvas consiste em 20 regiões lineares para cada poço  $n$ . Realizamos testes com instâncias de 32, 64 e 85 poços. Nas instâncias de 32 poços, utilizamos as primeiras curvas de [4]. As instâncias de 64 e 85 poços possuem as 56 curvas de Buitrago *et al.*, e as restantes foram obtidas aplicando-se perturbações às primeiras.

Os grafos de restrição de precedência  $G = (V, E)$  foram gerados aleatoriamente, com densidades que variam desde o caso irrestrito ( $E = \emptyset$ ) até grafos densos ( $|E| \approx \lfloor |V|^2/2 \rfloor$ ), formando 13 densidades de grafo para cada grupo de poços. A quantidade total de gás disponível  $q_{in}^{max}$  também variou de pouco gás de injeção até grande capacidade de compressão, formando 5 níveis de injeção para as instâncias com 32 poços, 4 níveis para as de 64 poços e 5 níveis para as de 85 poços.

Foram realizados testes comparando o desempenho do algoritmo sobre as instâncias puras, com as instâncias após a aplicação de cortes, que são apresentados em duas tabelas. As instâncias foram resolvidas em uma estação rodando o sistema operacional GNU/Linux, com 1GB de RAM e 2GB de espaço de disco para memória virtual, e equipado com um processador Intel Pentium IV de 3GHz. Na Tabela 4.6 podemos avaliar o impacto da geração de cortes em cada uma das taxas de injeção de gás, com a média dos resultados obtidos variando o tamanho do grafo de precedência. A Tabela 4.7 apresenta o impacto da geração de cortes de acordo com a densidade do grafo de restrições de precedência, com a média obtida nas instâncias que diferem pela capacidade de compressão de gás de injeção.

Os resultados apresentados nas colunas “Sem cortes” foram obtidos resolvendo as instâncias de  $\mathcal{P}_{cpl}(G)$  correspondentes utilizando o programa ILOG CPLEX [31], que é tido como uma das ferramentas mais poderosas para resolução de problemas de programação inteira-mista das existentes no mercado. As tabelas apresentam o número de iterações realizadas, o número de nós gerados, e o tempo em segundos para resolução do algoritmo de *Branch and Bound*.

Também foram aplicados cortes através de coberturas às instâncias, e em seguida foram submetidas ao mesmo programa de resolução de problemas de programação inteira-mista. Os resultados obtidos destas instâncias são apresentados nas colunas “Com cortes” das Tabelas 4.6 e 4.7. A fase de geração de cortes resolve iterativamente a relaxação linear de cada instância, busca uma cobertura cuja desigualdade obtida a partir do processo de *pseudo-lifting* corte a solução fracionária, e a adiciona ao conjunto de restrições. Este processo finaliza após produzir certo número de planos de corte, ou atingir um número de tentativas mal sucedidas. Como o problema de separação é difícil, isto é, não é fácil encontrar uma cobertura cuja desigualdade corte a solução fracionária, foi desenvolvida uma heurística simples, porém rápida, que sistematicamente busca por coberturas mínimas, aplica o processo de *pseudo-lifting*, e verifica se a desigualdade resultante é um plano de corte. O procedimento

$q_i^{max}$	Sem cortes			Com cortes			Redução (%)			
	Iterações	Nós	CPU	Iterações	Nós	CPU	Iterações	Nós	CPU	Relax.
$N = 32$										
300	25063	12271	5.7	310	13	0.28	98.76	99.89	95.09	16.06
500	2590	1409	3.40	375	13	1.00	85.52	99.08	70.59	4.77
700	46389	24476	19.21	19190	9309	10.65	58.63	61.97	44.56	4.99
1100	67905	32907	28.55	45236	22207	22.94	33.38	32.52	19.65	2.38
1500	28672	10845	13.16	25001	10906	11.30	12.80	-0.56	14.13	1.30
Total	170619	81908	70.02	90112	42448	46.17	47.19	48.18	34.06	5.90
$N = 64$										
700	1159402	553323	694.26	37293	8611	18.88	96.78	98.44	97.28	4.20
2300	7653030	3552912	6131.32	5981043	2708404	5079.69	21.85	23.77	17.15	1.23
2700	3464600	1552803	2698.16	3022232	1330704	2545.21	12.77	14.30	5.67	1.09
3500	377212	163308	304.78	259068	105303	220.64	31.32	35.52	27.61	0.86
Total	12654244	5822346	9828.52	9299636	4153022	7864.42	26.51	28.67	19.98	1.85
$N = 85$										
730	13879	8207	28.80	2134	712	8.44	84.62	91.32	70.69	1.51
996	1565583	828402	1618.51	260601	81406	205.68	83.35	90.17	87.29	1.56
1261	17398974	8915702	19555.33	8074359	3183605	10381.58	53.59	64.29	46.91	3.35
2058	15704801	7406702	15340.51	5221739	1803605	4455.65	66.75	75.65	70.96	1.16
2324	3570	1019	24.25	3501	1017	16.15	1.93	0.20	33.40	0.46
4648	4591	1208	23.64	2828	438	11.28	38.40	63.74	52.28	0.46
Total	34691398	17161240	36591.04	13565162	5070783	15078.78	60.90	70.45	58.79	1.42
Total	47516261	23065494	46489.58	22954910	9266253	22989.37	51.69	59.83	50.55	3.03

Tabela 4.6: Impacto dos planos de corte por taxa de injeção de gás

de geração de cortes foi implementado em linguagem C++ [32], utilizando as bibliotecas ILOG [31] para resolver as relaxações lineares e as estruturas de dados avançadas suportadas pelo pacote LEDA [33].

A coluna “Redução” apresenta o impacto da aplicação dos cortes em relação às instâncias puras no número de iterações e de nós do algoritmo de *Branch and Bound*. A coluna “Relax.” apresenta a redução relativa no valor da relaxação linear devido à aplicação dos planos de corte.

Apresentamos a seguir uma visão geral da heurística de busca dos cortes. Inicialmente são computados os conjuntos  $A_n$  e os parâmetros  $\sigma(n, m, j)$  que não dependem da cobertura. Seja  $\hat{\kappa} = \max\{\kappa(n) : n \in \mathcal{N}\}$ . Este pré-processamento é executado em tempo  $O(N^3\hat{\kappa})$ , ou  $O(N^4)$  dado que  $\hat{\kappa} \in O(N)$ . Para a busca de cada cobertura, os seguintes passos são executados:

1. Inicie com uma cobertura vazia,  $C = \emptyset$  e  $H(C) = \emptyset$ .
2. Escolha aleatoriamente pares de  $(n, k) \in \Omega$ , para o qual  $x_{n,k} \neq 0$  no par  $(x, \lambda)$  e  $n \notin N(C)$ , atualizando  $C$ ,  $H(C)$ , e  $\gamma(C)$ , até que uma cobertura seja obtida ou sejam realizadas no máximo  $\varepsilon_1$  escolhas aleatórias. Este passo é executado em tempo  $O(\varepsilon_1 N(\log N + \hat{\kappa})) = O(N^4)$  para  $\varepsilon_1 = N\hat{\kappa}$  e  $\hat{\kappa} \in O(N)$ .
3. Se o passo 2 falhar na busca de uma cobertura, então busque nos pares  $(n, k) \in \Omega$  em ordem arbitrária, não levando em conta o valor de  $x_{n,k}$ , e execute as mesmas ações para gerar a cobertura  $C$ . Este passo possui a mesma complexidade.
4. Se  $C$  não é uma cobertura, então aborte a heurística e notifique a falha. Caso contrário, remova os elementos de  $H(C)$  e  $C$  até que seja obtida uma cobertura mínima. Este passo é executado em tempo  $O(N^3)$ .

$ E(G) $	Sem cortes			Com cortes			Redução (%)			
	Iterações	Nós	CPU	Iterações	Nós	CPU	Iterações	Nós	CPU	Relax.
$N = 32$										
0	30495	20300	13.34	82	5	0.11	99.73	99.98	99.18	3.57
$\lfloor N/8 \rfloor$	34755	18300	12.34	21280	10903	9.92	38.77	40.42	19.61	3.50
$\lfloor N/4 \rfloor$	36700	19000	14.02	33911	18302	17.28	7.60	3.67	-23.25	3.21
$\lfloor N/3 \rfloor$	23309	10200	7.33	4869	1904	1.96	79.11	81.33	73.26	3.31
$\lfloor N/2 \rfloor$	10344	3601	3.72	8067	3802	3.65	22.01	-5.58	1.88	2.70
$\lfloor 2N/3 \rfloor$	17536	6101	5.36	10713	4202	5.06	38.91	31.13	5.60	3.13
$N$	13510	3926	6.15	8572	3102	4.53	36.55	20.99	26.34	2.37
$2N$	1484	302	1.54	1148	203	1.3	22.64	32.78	15.58	3.14
$3N$	799	70	0.95	316	5	0.78	60.45	92.86	17.89	3.26
$4N$	576	69	0.99	320	5	0.48	44.44	92.75	51.52	3.99
$5N$	396	27	1.08	320	5	0.48	19.19	81.48	55.56	6.74
$\lfloor N^2/3 \rfloor$	365	5	1.45	257	5	0.31	29.59	0	78.62	5.65
$\lfloor N^2/2 \rfloor$	350	7	1.75	257	5	0.31	26.57	28.57	82.29	10.18
Total	170619	81908	70.02	90112	42448	46.17	47.19	48.18	34.06	5.90
$N = 64$										
0	2492947	1318200	1978.32	114	4	0.21	100.00	100.00	99.99	1.86
$\lfloor N/8 \rfloor$	5067958	2577200	4297.35	4797594	2443601	4271.98	5.33	5.18	0.59	1.26
$\lfloor N/4 \rfloor$	1673498	708800	1258.03	1528423	645101	1352.08	8.67	8.99	-7.48	1.27
$\lfloor N/3 \rfloor$	1931502	793100	1394.37	1752252	707301	1354.86	9.28	10.82	2.83	1.34
$\lfloor N/2 \rfloor$	569639	176100	338.66	485924	153501	344.51	14.70	12.83	-1.73	1.27
$\lfloor 2N/3 \rfloor$	427275	128900	233.11	271137	89001	226.08	36.54	30.95	3.02	1.29
$N$	418611	109800	254.93	414835	106400	265.90	0.90	3.10	-4.30	0.10
$2N$	31470	5200	18.76	16532	3301	13.33	47.47	36.52	28.94	0.37
$3N$	12886	1900	10.47	12590	1800	12.04	2.30	5.26	-15.00	1.30
$4N$	25509	3001	19.50	17282	2901	17.42	32.25	3.33	10.67	0.84
$5N$	1003	103	5.99	1713	103	4.05	-70.79	0.00	32.39	3.26
$\lfloor N^2/3 \rfloor$	1196	38	5.72	535	4	0.77	55.27	89.47	86.54	3.06
$\lfloor N^2/2 \rfloor$	750	4	13.31	705	4	1.19	6.00	0.00	91.06	6.74
Total	12654244	5822346	9828.52	9299636	4153022	7864.42	26.51	28.67	19.98	1.85
$N = 85$										
0	7014551	4607700	9366.13	87	4	0.28	100.00	100.00	100.00	1.95
$\lfloor N/8 \rfloor$	7159944	4125100	8660.61	87	4	0.28	100.00	100.00	100.00	1.15
$\lfloor N/4 \rfloor$	6041502	3048500	6297.51	82	4	0.29	100.00	100.00	100.00	1.11
$\lfloor N/3 \rfloor$	8740144	3816200	8235.11	7932336	3504502	10688.01	9.24	8.17	-29.79	0.51
$\lfloor N/2 \rfloor$	2776802	823500	1920.01	2746963	828601	2062.87	1.07	-0.62	-7.44	0.16
$\lfloor 2N/3 \rfloor$	1420430	369300	976.18	1426351	363800	1084.78	-0.42	1.49	-11.12	0.08
$N$	1336313	343601	916.20	1300825	349701	1094.43	2.66	-1.78	-19.45	3.16
$2N$	134951	19401	80.24	115668	17901	69.26	14.29	7.73	13.68	0.16
$3N$	16013	1601	15.30	13983	1601	16.27	12.68	0.00	-6.34	0.31
$4N$	24479	2401	18.88	10407	1501	13.74	57.49	37.48	27.22	0.23
$5N$	16533	1701	17.07	10621	1701	16.73	35.76	0.00	1.99	0.91
$\lfloor N^2/3 \rfloor$	659	4	14.75	736	4	1.62	-11.68	0.00	89.02	4.64
$\lfloor N^2/2 \rfloor$	916	4	25.16	687	4	2.79	25.00	0.00	88.91	4.04
Total	34683237	17159013	36543.15	13558833	5069328	15051.35	60.91	70.46	58.81	1.42
Total	34691398	17161240	36591.04	13565162	5070783	15078.78	60.90	70.45	58.79	3.03

Tabela 4.7: Impacto dos planos de corte por densidade do grafo

5. Calcule todos os conjuntos  $H(C, n)_h$  e  $H(C, n, k)_h$ . Este passo executa em tempo  $O(N^3)$ .
6. Obtenha  $\bar{\Gamma}(C)$ ,  $\bar{\Gamma}_H(C)$ ,  $\bar{\Gamma}_A(C)$  e  $\bar{\Gamma}_B(C)$ , que é executado em  $O(N^2 \log N)$ .
7. Obtenha os fatores de *pseudo-lifting*  $\beta_{n,k}$  para todos  $(n, k) \in \bar{\Gamma}_H(C) \cup \bar{\Gamma}_A(C)$ , que executa em tempo  $O(N^3)$ .
8. Em no máximo  $\varepsilon_2 = N$  passos, sugira uma partição  $\{\bar{\Theta}_r\}_{r=0}^R$  de  $\bar{\Gamma}_B(C)$  para induzir uma desigualdade de *pseudo-lifting* que corte a solução:
  - (a) Seja  $U$  o conjunto de folhas de  $G$  que não está em  $N(C)$ .
  - (b) Para cada  $n \in U$ , defina o conjunto  $\bar{\Theta}_i = \Omega(\{n\})$  com  $n_i = n$ .
  - (c) Aleatoriamente, separe  $V \setminus (U \cup N(C))$  em dois conjuntos disjuntos  $A$  e  $B$ .
  - (d) Para cada  $n \in A$ , defina o conjunto  $\bar{\Theta}_i = \Omega(\{n\})$  com  $n_i = n$ .
  - (e) Seja  $\bar{\Theta}_0 = \{(n, k) \in \Omega : n \in B, k = 3, \dots, \kappa(n)\}$ .
  - (f) Para cada  $n \in B$ , adicione  $(n, 2)$  a um conjunto aleatório  $\bar{\Theta}_r$ ,  $r > 0$ , de maneira que  $n \prec n_r$ .
  - (g) Compute os fatores de *pseudo-lifting*  $(n, k) \in \bar{\Gamma}_B(C)$  de acordo com  $\{\bar{\Theta}_r\}_{r=0}^R$ .
  - (h) Se a desigualdade de *pseudo-lifting* corta a solução corrente, então pare a busca e retorne o plano de corte.

Os passos 8-(a) à 8-(h) executam em tempo  $O(N^3)$ , portanto, este passo executa em  $O(N^4)$ .

9. Se o passo 8 não encontrou um plano de corte, notifique a falha da heurística.

Dos passos acima concluímos que a heurística de separação é executada em tempo  $O(N^4)$ . Esta análise considera o pior caso, sendo que uma análise mais apurada pode levar a resultados melhores. Uma análise mais otimista pode assumir  $\hat{\kappa} \in O(1)$ .

Da Tabela 4.6 podemos inferir que a aplicação de planos de corte promove uma diminuição substancial do esforço computacional para resolução do problema em termos do número de iterações e nós de *Branch and Bound*. Reduções maiores são observadas em circunstâncias em que a disponibilidade de gás de injeção é limitada.

A Tabela 4.7 apresenta a influência da densidade do grafo e seus ganhos com a aplicação de planos de corte. Os resultados indicam que o problema se torna mais difícil à medida que a densidade do grafo diminui. Ainda podemos observar que os ganhos absolutos da aplicação de planos de corte diminuem com o aumento da densidade do grafo, enquanto os ganhos relativos aumentam nos extremos (devido às restrições de precedência e cenários altamente restritivos).

Os resultados computacionais mostram que a geração de planos de corte pode participar do processo de resolução do problema. Entretanto, para um aproveitamento completo da teoria, a geração de planos de cortes deveria ocorrer em todos os nós da árvore de soluções, resultando em um algoritmo de *Branch and Cut*; e não apenas no poliedro completo, como foi implementado. Conforme a busca avança nos ramos da árvore de soluções, as variáveis discretas são definidas como 0 ou 1, tornando o grafo de precedência esparso e permitindo que os cortes locais reduzam a distância primal-dual.

## 4.6 Sumário

Neste capítulo vimos uma formulação do problema de alocação ótima de gás de injeção sob restrições de capacidade máxima e de precedência utilizando linearização por partes e programação matemática inteira-mista. Observamos que esta formulação é reduzida a um problema da mochila sob restrições de precedência, assim, pudemos expandir a teoria de coberturas para buscar desigualdades válidas fortes para a formulação. Projetamos as desigualdades para todas as variáveis de decisão, com os fatores de *lifting*, e propusemos uma maneira alternativa, computacionalmente mais leve, para o cômputo de valores aproximados dos fatores de *lifting*, que chamamos de *pseudo-lifting*. Por fim, apresentamos resultados computacionais comparativos que comprovam a eficácia da inserção de planos de corte no problema.

## Capítulo 5

# Interface de otimização

Como parte deste trabalho implementamos uma interface de usuário para resolução de problemas de alocação ótima de gás de injeção. Desenvolvemos uma ferramenta de auxílio à tomada de decisão para operação de campos de petróleo operados por *gas-lift*.

Os algoritmos de otimização são de natureza não similar e com interface algumas vezes complexa. Buscamos criar um ambiente que abstraia os modelos de otimização, para que o operador se preocupe apenas com as informações referentes ao campo de petróleo, de maneira que a visualização dos dados seja o mais próximo da realidade. Assim, fica a cargo do sistema a interpretação dos dados e a montagem e resolução das instâncias, retirando do usuário a necessidade de compreender o modelo. Esta propriedade do sistema permite que para um mesmo cenário possam ser aplicados diferentes algoritmos de otimização.

Utilizamos uma estrutura cliente/servidor para resolução dos algoritmos. Procuramos desenvolver uma interface geral, que não seja aplicável apenas ao problema de interesse. A interface do cliente permite especificar problemas de alocação ótima de gás de injeção e possui flexibilidade para que sejam incluídas restrições diferentes das tratadas em nosso trabalho como, por exemplo, restrições da tubulação, topologia da tubulação, perda de cargas, acoplamento de poços e dinâmica do processo. O servidor para algoritmos de otimização por sua vez, permite que qualquer algoritmo de otimização seja utilizado, não apenas os que tratam de problema de determinação de pontos de operação para campos de petróleo.

A estrutura cliente/servidor foi empregada para permitir que seja utilizada uma máquina diferente para a caracterização do problema da máquina em que a otimização será executada. A resolução de problemas de otimização, em geral, necessita de muito esforço e recursos computacionais para instâncias médias e grandes, exigindo um computador potente. Assim, destinamos a uma máquina adequada, o servidor, a resolução destes problemas; enquanto as máquinas em que as instâncias do problema são geradas podem ter recursos restritos. Outro fator considerado é que algumas das ferramentas utilizadas para a resolução dos problemas de otimização são proprietárias [31], então esta estrutura requer apenas uma licença de uso das ferramentas no servidor.

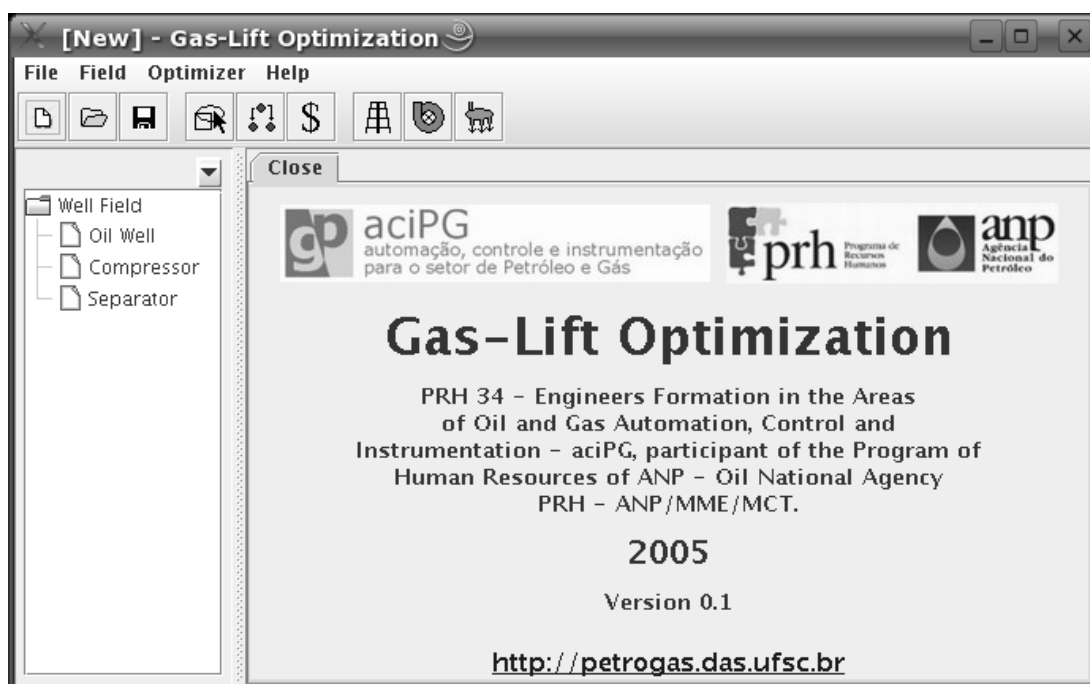


Figura 5.1: Ambiente de especificação do problema

## 5.1 Ambiente de especificação do problema

A interface de otimização possui um módulo de especificação do problema em que o usuário descreve cada um dos elementos que compõem um campo de petróleo. Este módulo foi estruturado de maneira *bottom-up*, para permitir que o sistema seja facilmente expandido, possibilitando que novas características sejam acrescentadas ao sistema de acordo com o surgimento das necessidades.

A linguagem escolhida para implementação deste sistema foi a Java [34], devido principalmente a sua portabilidade entre sistemas operacionais e facilidade de implementação de ambientes gráficos. Na Figura 5.1 podemos visualizar o ambiente desenvolvido e seus componentes. Este ambiente está dividido em painéis, cada um responsável pela edição de um elemento ou característica do campo de petróleo:

**Editor gráfico:** este painel permite que o usuário formule um diagrama esquemático dos *componentes* do campo (poços, compressores, separadores e manifolds<sup>1</sup>) e suas conexões através das tubulações, facilitando a visualização e navegação entre os elementos. Na Figura 5.2 temos um exemplo de um diagrama representativo de um campo de petróleo, seus componentes e tubulações.

**Preços:** nesta guia o usuário informa ao sistema os preços de mercado do óleo e gás produzidos, assim como o custo de processamento da água extraída. Estes valores são necessários pois, nossos algoritmos de otimização utilizam um critério econômico na função objetivo, buscando o maior lucro da operação do campo.

<sup>1</sup>Elementos de conexão de tubulação.



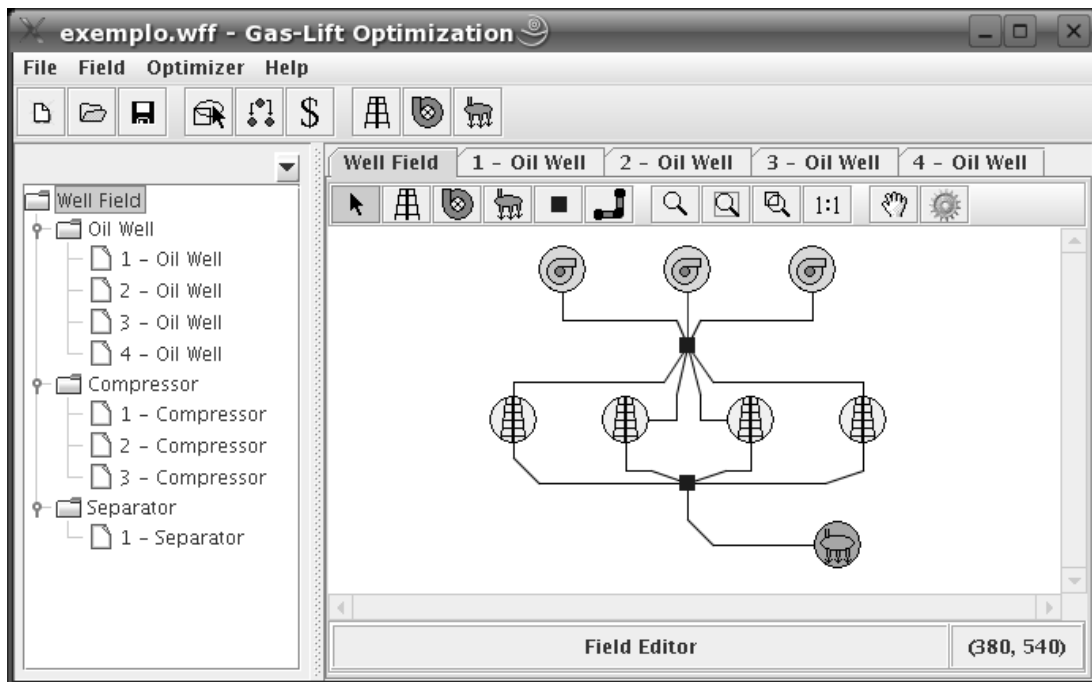


Figura 5.2: Editor gráfico

**Grafo de precedência:** este painel permite que o usuário informe ao sistema as arestas que compõem o grafo de restrições de precedência de ativação. Os vértices do grafo são os próprios poços. Uma aresta significa que um poço só será ativado se seu predecessor estiver operando, pelo menos, em seu nível mínimo de produção. O grafo de precedência não possui relação com o grafo criado no editor gráfico, então um poço pode preceder outro mesmo que eles não estejam conectados.

**Compressor:** um campo de petróleo operado por *gas-lift* possui pelo menos um compressor para produzir o gás de injeção, podendo ter várias unidades formando um banco de compressores. Cada compressor possui uma capacidade máxima de compressão, que é somada com a capacidade total do banco. Possui também um custo de compressão associado. Uma característica comum dos compressores, separadores e poços é um *nome* que identifica o componente no campo, e a possibilidade de desabilitá-lo. Um compressor desabilitado pode representar, por exemplo, que ele está desligado para manutenção, que gera a necessidade do recômputo das taxas de injeção de cada poço, pois a capacidade total de injeção é uma restrição significativa.

**Separador:** este painel permite que o usuário informe ao sistema o limite de separação do petróleo produzido em óleo, gás e água, assim como as capacidades máximas de tratamento de cada um dos sub-produtos da extração. Estes limites são tratados como restrições por Camponogara e Nakashima [35], e foram incluídos para tornar o programa compatível com esta formulação.

**Poço:** nesta guia o usuário pode descrever cada um dos poços do campo operados por *gas-lift*. Para cada poço o usuário pode inserir várias CPPs, utilizando uma descrição linear por partes, polinomial, polinomial-logarítmica (polilog) ou exponencial (veja Seção 2.3). São permitidas múltiplas curvas, pois podemos utilizar tipos diferentes de expressões para representar a mesma CPP. Também é possível que o operador não tenha certeza da curva que melhor descreve o poço,

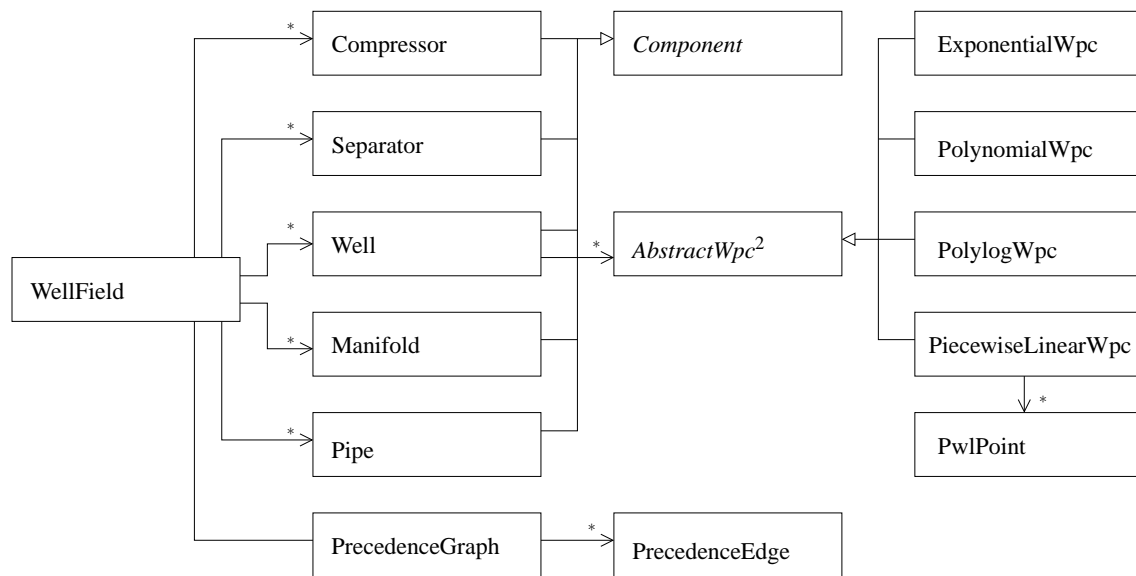


Figura 5.3: Diagrama de classes da estrutura de dados da interface

e deixa para o algoritmo a escolha da curva baseada em algum critério como, por exemplo, o pior caso ou caso médio [9]. Associado a cada CPP também temos as frações médias de óleo, gás e água presentes no petróleo produzido pelo poço.

Outra capacidade do sistema é a de transformar curvas de um tipo para outro. Curvas descritas como lineares por partes podem ser convertidas em polinomiais ou polilogs em um processo de ajuste por mínimos quadrados. O ajuste de curva pode ser irrestrito ou com garantia que a curva seja côncava em seu intervalo de interesse. A garantia de concavidade é uma exigência de alguns algoritmos de otimização que utilizam programação dinâmica [9].

### 5.1.1 Estrutura de dados

O sistema foi estruturado internamente em um conjunto de classes que descrevem o campo de petróleo. A Figura 5.3 mostra um diagrama de classes UML<sup>3</sup> [36, 37] que representa as principais classes desta estrutura. Nele apresentamos apenas os nomes das classes, omitindo seus atributos e métodos. A classe base `WellField` representa o campo como um todo, e é passada para todos os objetos que a utilizam ou manipulam. A classe `PrecedenceGraph` representa o grafo de restrições de precedência. Ela possui uma lista de arestas (`PrecedenceEdge`) e seus vértices são os próprios poços do campo.

A classe abstrata `Component` representa todos os elementos que compõem fisicamente o campo. Dela herdam os compressores (`Compressor`), separadores (`Separator`), poços de petróleo (`Well`), tubos (`Pipes`) e conexões de tubos (`Manifold`). A ampliação dos tipos de elementos do campo pode ser feita herdando novas classes de `Component`. A estrutura do campo possui uma lista para cada classe filha de `Component`. Elas estão ordenadas com base em um atributo identificador de

<sup>2</sup>WPC - *Well Performance Curve*, Curva de Performance do Poço.

<sup>3</sup>UML - *Unified Modeling Language*, linguagem de modelagem que utiliza diagramas padronizados.

Component, assim a busca de um elemento na lista, método mais utilizado na implementação, pode ser feita como uma busca binária, que executa em tempo  $\Theta(\log_2 n)$ [38]. Esta política de manter a lista ordenada, em contrapartida, penaliza outros métodos, como a inserção de novos elementos que é executada em  $\Theta(n)$ .

Cada poço pode ter várias CPPs. A base para estas curvas é a classe abstrata `AbstractWpc`. Desta classe herdam as especializações da curva, cada uma para sua expressão matemática equivalente: exponencial (`ExponentialWpc`), polinomial (`PolynomialWpc`), polinomial-logarítmica (`PolylogWpc`) e linear por partes (`PiecewiseLinearWpc`), que por sua vez possui um conjunto de pontos `PwlPoint`. Caso se deseje que o sistema trabalhe com alguma outra formulação para a CPP basta herdar uma nova classe de `AbstractWpc` e desenvolver a interface com o usuário para a curva.

### 5.1.2 Ajuste de curvas

O sistema possui a capacidade de transformar curvas entre as formulações aceitas no sistema. Especificamente, foram implementadas transformações da curva linear por partes para as curvas polinomiais de terceira ordem e polilog. Estas transformações são feitas por um método de ajuste de curvas por mínimos quadrados, que tenta aproximar a curva ao máximo dos pontos que formam a curva linear por partes. Algoritmos como o de programação dinâmica apresentado em [9] exigem que as curvas de entrada no sistema sejam côncavas, por isso o sistema possui a capacidade de garantir a concavidade da curva no ajuste.

Outra transformação possível é uma discretização de qualquer uma das outras curvas em linear por partes. Entretanto, este processo considera apenas que a distribuição dos pontos seja feita em intervalos constantes, não levando em conta a correlação da curva proposta.

A seguir apresentaremos a base matemática utilizada para os ajustes para curvas polinomias e polilog.

#### Polinomial

Considerando a equação polinomial de terceira ordem

$$p_3(q_{in}) = q_{out} = c_1 + c_2 q_{in} + c_3 q_{in}^2 + c_4 q_{in}^3 \quad (5.1)$$

e um conjunto de  $k$  pontos  $Q = \{(q_{in}^1, q_{out}^1), \dots, (q_{in}^k, q_{out}^k)\}$ . Se substituirmos cada ponto de  $Q$  em (5.1), chegamos a um sistema linear com 4 incógnitas e  $k$  equações, que pode ser escrito matricialmente

$$\begin{bmatrix} 1 & (q_{in}^1) & (q_{in}^1)^2 & (q_{in}^1)^3 \\ 1 & (q_{in}^2) & (q_{in}^2)^2 & (q_{in}^2)^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (q_{in}^k) & (q_{in}^k)^2 & (q_{in}^k)^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} q_{out}^1 \\ q_{out}^2 \\ \vdots \\ q_{out}^k \end{bmatrix}.$$

Como  $k \gg 4$  neste sistema linear  $Ax = b$ , minimizamos o resíduo  $\|Ax - b\|$  para obter uma solução que melhor se ajuste aos pontos, ou seja:

$$\text{Minimize } \frac{1}{2} \|Ax - b\|^2 \quad (5.2)$$

Para resolver (5.2) desenvolvemos a função objetivo

$$\begin{aligned} \frac{1}{2} \|Ax - b\|^2 &= \frac{1}{2} (Ax - b)^T (Ax - b) \\ &= \frac{1}{2} (x^T A^T - b^T) (Ax - b) \\ &= \frac{1}{2} (x^T A^T Ax - x^T A^T b - b^T Ax + b^T b) \\ &= \frac{1}{2} x^T A^T Ax - b^T Ax + \frac{1}{2} b^T b. \end{aligned} \quad (5.3)$$

Como  $\frac{1}{2} b^T b$  é constante, pode ser removido da função objetivo sem que seja alterado o valor das variáveis de decisão na solução do problema. Minimizar a expressão (5.3) consiste em um problema de Programação Quadrática. Note que  $x^T A^T Ax = \|Ax\|^2 \geq 0, \forall x$ , ou seja, a matriz  $A^T A$  é positiva semi-definida. Se o posto de  $A$  é completo,  $\text{rank}(A) = 4$ , então  $A^T A$  é inversível e a solução para (5.2) pode ser obtida com  $x = (A^T A)^{-1} A^T b$ .

Desejamos poder impor que o polinômio  $p_3(q_{in})$  seja côncavo em todo o intervalo  $[q_{in}^1, q_{in}^k]$ . Para isto sua segunda derivada deve ser negativa em todo o intervalo contínuo,

$$p_3''(q_{in}) = 6c_4 q_{in} + 2c_3 \leq 0, \forall q_{in} \in [q_{in}^1, q_{in}^k].$$

Isto implica em um número infinito de restrições para cobrir todo o intervalo. Este número de restrições pode ser reduzido segundo a seguinte proposição:

**Proposição 5.1** *Se  $p_3(q_{in})$  for côncava em  $q_{in}^1$  e  $q_{in}^k$ , então  $p_3(q_{in})$  é côncava para todo  $q_{in} \in [q_{in}^1, q_{in}^k]$ .*

**Prova:** Como  $p_3''(q_{in})$  é um polinômio de primeira ordem ( $c_4 \neq 0$ ), ele possui no máximo uma raiz no intervalo  $[q_{in}^1, q_{in}^k]$ . Se  $p_3''(q_{in}) \leq 0, q_{in} \in \{q_{in}^1, q_{in}^k\}$ , segundo o Teorema de Bolzano [39], existe um número par de raízes para  $p_3''(q_{in})$  em  $[q_{in}^1, q_{in}^k]$ . Então  $p_3''(q_{in})$  não possui raízes em  $[q_{in}^1, q_{in}^k]$ , ou seja, não ocorre alteração na concavidade de  $p_3(q_{in})$  no intervalo em questão. Como  $p_3(q_{in})$  é côncava em  $q_{in}^1$  e  $q_{in}^k$ , então  $p_3(q_{in})$  é côncava  $\forall q_{in} \in [q_{in}^1, q_{in}^k]$ . ■

Esta proposição restringe para dois o número de restrições, que são lineares. Isto faz com que seja possível utilizar um algoritmo de Programação Quadrática com Restrições para ajustar a curva polinomial aos pontos com garantia de concavidade.

### Polinomial-logarítmica

Considere agora a equação polinomial-logarítmica

$$p_l(q_{in}) = q_{out} = c_1 + c_2 q_{in} + c_3 q_{in}^2 + c_4 \ln(1 + q_{in}) \quad (5.4)$$

e o conjunto de  $k$  pontos  $Q = \{(q_{in}^1, q_{out}^1), \dots, (q_{in}^k, q_{out}^k)\}$ . De maneira análoga à seção anterior substituímos cada ponto de  $Q$  em (5.4) e obtemos o sistema linear representado matricialmente

$$\begin{bmatrix} 1 & (q_{in}^1) & (q_{in}^1)^2 & \ln(1 + q_{in}^1) \\ 1 & (q_{in}^2) & (q_{in}^2)^2 & \ln(1 + q_{in}^2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (q_{in}^k) & (q_{in}^k)^2 & \ln(1 + q_{in}^k) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} q_{out}^1 \\ q_{out}^2 \\ \vdots \\ q_{out}^k \end{bmatrix}.$$

Com o mesmo raciocínio que foi visto anteriormente, podemos obter a curva que melhor se ajuste a  $Q$  sem garantia de que a curva seja côncava.

Para que  $p_l(q_{in})$  seja côncava em  $[q_{in}^1, q_{in}^k]$  sua derivada segunda deve ser negativa no intervalo,

$$p_l''(q_{in}) = 2c_3 - c_4 \frac{1}{(1 + q_{in})^2} < 0, \forall q_{in} \in [q_{in}^1, q_{in}^k]$$

**Proposição 5.2** Se  $p_l(q_{in})$  for côncava em  $q_{in}^1$  e  $q_{in}^k$ , então  $p_l(q_{in})$  é côncava para todo  $q_{in} \in [q_{in}^1, q_{in}^k]$ .

**Prova:** Se  $p_l''(q_{in}^k) < 0$  e  $c_4 > 0$ , conforme  $q_{in}$  decresce a partir de  $q_{in}^k$ ,  $p_l''(q_{in})$  se torna mais negativa, garantindo que  $p_l(q_{in})$  seja côncava para todo  $q_{in} \leq q_{in}^k$ . Caso  $c_4 < 0$ ,  $p_l''(q_{in})$  decresce conforme  $q_{in}$  cresce, então é necessário que  $p_l''(q_{in}^1) < 0$ . Como  $c_4$  pode assumir qualquer sinal, é necessário que nos dois casos a concavidade seja verificada. ■

Assim, é necessário apenas impor garantia de concavidade nos limites do conjunto para que a função seja côncava em todo o intervalo.

Na figura 5.4 apresentamos a guia utilizada para caracterizar um poço de petróleo, na aba de CPP (WPC). No lado esquerdo o usuário informa separadamente as características de cada uma das curvas. No caso, está sendo editada a curva número 1, que é uma curva linear por partes com pontos de acordo com os da tabela. No topo à direita estão as frações de óleo, gás e água produzidas pelo poço. Abaixo está um gráfico que mostra as curvas (algumas ou todas) do poço. A curva descontínua desenhada representa a linear por partes que está sendo editada. A curva de melhor correlação (mais próxima) à descontínua foi gerada a partir do ajuste da primeira em uma expressão polinomial-logarítmica com garantia de concavidade, resultando na expressão:

$$q_{out} = -5,5069 \cdot 10^3 - 9,3484 \cdot 10^{-1} \cdot q_{in} + 3,7372 \cdot 10^{-5} \cdot q_{in}^2 + 1,2951 \cdot 10^3 \cdot \ln(1 + q_{in}).$$

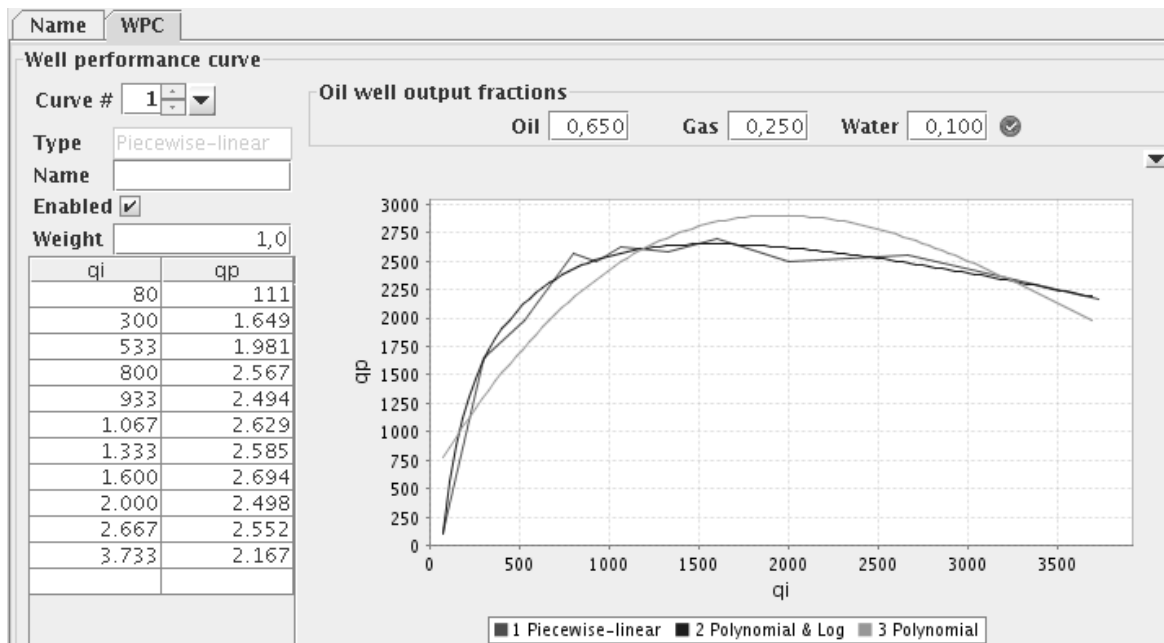


Figura 5.4: Guia de caracterização de poço com exemplo de ajuste de curva

A terceira curva é uma polinomial de terceira ordem, que também foi gerada com garantia de concavidade:

$$q_{out} = 5,5866 \cdot 10^2 + 2,7233 \cdot q_{in} - 9,4509 \cdot 10^{-4} \cdot q_{in}^2 + 8,439 \cdot 10^{-8} \cdot q_{in}^3.$$

### 5.1.3 Padrão dos arquivos

É desejável no sistema que o usuário possa gravar e recuperar os dados referentes ao campo de petróleo que está sendo estudado, para que o processo de caracterização do campo e invocação dos algoritmos de otimização possa ser interrompido e recuperado a qualquer momento. Como um requisito do nosso sistema é a capacidade de expansão, buscamos uma estrutura aberta e facilmente manipulável. Para isto, utilizamos a linguagem XML<sup>4</sup> [40], que procura padronizar a representação de dados. Este padrão é bastante abrangente, sendo que foi utilizado apenas uma pequena parte de suas funcionalidades.

Na Figura 5.5 apresentamos um exemplo da estrutura do arquivo para um campo de petróleo. Nela podemos observar os preços do óleo e gás, e o custo do tratamento da água extraída. Observamos também que o campo possui um poço (*Well*), um compressor (*Compressor*) e um tubo (*Pipe*). O poço, por sua vez, é descrito por duas CPPs: uma linear por partes, com quatro pontos, e outra polinomial-logarítmica, ajustada a partir da primeira. O compressor possui sua capacidade máxima de injeção de gás e seu custo de operação. Tanto no poço como no compressor, existe uma informação de posição, que é utilizada no editor esquemático do campo. O tubo liga o poço “1” ao compressor “1”, e possui uma quina. Para os atributos que não estão descritos no arquivo são assumidos valores padrão, como por exemplo, o “Habilitado”, que assume que o poço e o compressor estão em operação.

<sup>4</sup>XML - *eXtensible Markup Language*, linguagem de marcação extensível. É uma recomendação para gerar linguagens de marcação cujo propósito principal é facilitar o compartilhamento de informações através da Internet.

```
<?xml version="1.0"encoding="UTF-8"?>
<WellField>
  <OilPrice>20.0</OilPrice>
  <GasPrice>2.0</GasPrice>
  <WaterCost>1.0</WaterCost>
  <Well>
    <Number>1</Number>
    <Position X="60.0"Y="200.0"/>
    <Function Type="PieceWise">
      <Oil>0.731</Oil>
      <Gas>0.203</Gas>
      <Water>0.066</Water>
      <Point QI="0.08"QP="1774.0"/>
      <Point QI="0.3"QP="2053.0"/>
      <Point QI="0.6"QP="2215.0"/>
      <Point QI="0.9"QP="2266.0"/>
    </Function>
    <Function Type="Polylog">
      <Oil>0.731</Oil>
      <Gas>0.203</Gas>
      <Water>0.066</Water>
      <C1>1616.5813240907655</C1>
      <C2>-6422.52084981815</C2>
      <C3>1099.2515464911717</C3>
      <C4>8630.147464122258</C4>
      <LowerBound>0.08</LowerBound>
      <UpperBound>0.9</UpperBound>
    </Function>
  </Well>
  <Compressor>
    <Number>1</Number>
    <Capacity>0.1</Capacity>
    <CompCost>5.0</CompCost>
    <Position X="170.0"Y="250.0"/>
  </Compressor>
  <Pipe>
    <Extremity1 Number="1"Type="Well"/>
    <Extremity2 Number="1"Type="Compressor"/>
    <PipeCorner X="60.0"Y="250.0"/>
  </Pipe>
</WellField>
```

Figura 5.5: Exemplo do formato de gravação dos dados

## 5.2 Servidor de otimização

O servidor é a implementação que permite que algoritmos de otimização possam ser executados remotamente. A máquina que hospeda o servidor pode possuir programas específicos de otimização, assim não é necessário que estes programas sejam instalados na máquina do usuário.

Em nossa aplicação utilizamos ferramentas proprietárias para resolução de nossos algoritmos. Fica a cargo do servidor disponibilizar o uso destas ferramentas para todos os usuários do sistema. Outro aspecto é o sistema operacional necessário para a execução das ferramentas. Como desejávamos um ambiente de especificação de problema livre desta restrição, precisamos criar uma interface genérica para os algoritmos. Desejávamos também que esta funcionalidade não estivesse disponível apenas para algoritmos de otimização de taxas de injeção de gás, mas que pudesse resolver outros problemas de otimização.

Para deixar a interface do servidor o mais genérica possível, optamos por utilizar uma estrutura baseada em arquivos. O usuário envia um, ou vários, arquivos para o servidor contendo a instância do problema e mais algum arquivo que possa necessitar, como por exemplo, o modelo caso não esteja junto com a instância. Envia também um arquivo com um ou mais comandos, que indica ao servidor quais devem ser as ações tomadas com os arquivos de instância. O servidor, por sua vez, grava todas as mensagens de resposta recebidas da execução das instâncias do problema em um arquivo de saída e outro de erros, que é enviado ao usuário. Caso o usuário deseje direcionar passos intermediários para arquivos auxiliares, também é possível, podendo consultá-los posteriormente. Uma desvantagem desta estrutura é que não permite que o usuário monitore a resolução de seu problema enquanto ela é executada; apenas permite que veja a resposta após a resolução ter sido concluída.

Um grande desafio enfrentado foi que o computador que possui os recursos para executar os algoritmos de otimização estava localizado dentro de uma rede segura. Desejávamos que este servidor pudesse ser acessado a partir de qualquer computador, ou seja, que também pudesse ser acessado a partir de computadores localizados fora da rede segura. A conexão direta entre estas duas máquinas não é permitida, pois abre um caminho para falhas na segurança da rede.

Para resolver este problema foram desenvolvidos dois programas. O primeiro é o gerenciador das instâncias, responsável por receber, gerenciar e executar as instâncias, e enviar e permitir consultas das soluções obtidas. Este programa está instalado e executa no computador com recursos para resolver os problemas de otimização. O segundo programa é uma interface do servidor, e fica alojado na máquina que conecta a rede segura ao meio externo. Ele é responsável por receber as instâncias, verificar se elas são seguras (não são arquivos *maliciosos*, com intenção de “atacar” a rede interna), e encaminhar ao servidor; assim como pesquisar as respostas e apresentar para o usuário.

A seguir, apresentamos detalhadamente cada um dos programas que formam o sistema.



### 5.2.1 Gerenciador de instâncias

O gerenciador de instâncias é o núcleo do servidor de otimização. Desenvolvido em linguagem C++ [32, 41] para o sistema operacional Linux, o sistema foi estruturado utilizando programação concorrente [42], que permite que várias ações sejam executadas simultaneamente (na realidade, ocorre um *pseudo-paralelismo* entre os processos [43]). No computador em que está sendo executado, o gerenciador roda em segundo plano (*background*), ficando totalmente transparente para algum usuário que eventualmente esteja utilizando aquela máquina.

Os processos que compõem o gerenciador são:

**Gerenciador de conexões:** Este processo é responsável por receber e despachar novas conexões de rede executadas no servidor. Quando um usuário deseja conectar-se com o servidor, ele realiza uma solicitação em uma *porta TCP-IP* [44] específica do servidor. Este processo, por sua vez, fica travado, aguardando esta solicitação. Quando ela ocorre, o processo avalia se a conexão pode ser estabelecida (limitando o número máximo de conexões), e caso positivo dispara um novo processo que faz a comunicação com o usuário.

**Conexões:** Cada conexão estabelecida com o servidor gera um novo processo desta classe. Este processo recebe comandos do usuário e responde de maneira apropriada, por exemplo, a carga de uma nova instância ou obtenção de uma resposta.

**Monitor de processador:** Quando existe uma nova instância para ser executada, o sistema necessita saber se a máquina está disponível para executá-la. Algoritmos de otimização são *CPU bounded*, isto é, necessitam do processador intensamente. Adotamos então como métrica para saber se é possível executar o algoritmo um histórico (de aproximadamente um minuto) de ocupação do processador. Se o processador estiver sendo pouco utilizado (no caso menos de 50% em todos os pontos do histórico), o sistema está apto para disparar processos de otimização. Este processo tem como função verificar ciclicamente (em intervalos de aproximadamente 20 segundos) a ocupação do processador e montar um histórico de uso, assim como permitir ou bloquear que novas instâncias sejam iniciadas.

**Gerenciador de execução:** Quando o processo de conexões recebe uma nova instância e manda que seja executada, ela é colocada em uma fila. Este processo é responsável por verificar se existem elementos nesta fila e, quando for permitido, executar os comandos da instância. Enquanto se busca a solução de uma instância, este processo também é responsável por verificar quanto tempo de processador o problema utilizou, o limitando de acordo com os direitos do usuário, evitando assim que problemas muito grandes sejam resolvidos por usuários com pouco direito de acesso. Finalizada a execução da instância, este gerenciador é responsável por enviar a resposta ao usuário.

**Gerenciador de instâncias antigas:** Este processo é executado em intervalos de tempo bastante longos. Sua função é monitorar há quanto tempo cada instância está armazenada no servidor, e descartá-la se este período for maior que um certo limite. Isto faz com que instâncias

muito antigas, e potencialmente não mais necessárias ao usuário, sejam eliminadas do servidor, liberando espaço de armazenamento.

O protocolo de conexão com o servidor foi feito inspirado nos protocolos SMTP [45] e POP [46]. A troca de dados é feita utilizando caracteres ASCII<sup>5</sup>, que torna a comunicação bastante simples. Comandos são enviados ao servidor em um formato texto, interpretados e as respostas geradas são enviadas. Podemos dividi-los em três grupos: identificação, que permite que o usuário se apresente ao servidor através de uma identificação (comando ID) e uma senha (comando PSW). Caso o usuário não se identifique, o servidor vai admitir que a conexão é *anônima*, e seus direitos (como por exemplo o tempo de processador para rodar sua instância) serão restritos.

Um segundo grupo de comandos permite que o usuário informe ao servidor o problema a ser resolvido. Os comandos FILE e CMD permitem que o usuário envie arquivos contendo a instância e os comandos que serão executados, respectivamente. Para solicitar que o servidor execute a instância enviada utiliza-se os comandos RUN e RUNMAIL, sendo que o último envia a resposta por e-mail para o usuário.

O último grupo de comandos é utilizado para verificar a resposta da execução de um problema. Eles são apresentados no exemplo da Figura 5.6, em que é estabelecida uma conexão com o servidor e solicitada esta resposta. As sentenças grifadas são os comandos enviados pelo usuário, e as sem grifo são as respostas obtidas. Na primeira linha é solicitada uma conexão com o servidor, que está hospedado no mesmo computador em que o teste foi realizado, na porta 1209. A conexão é estabelecida e na linha 2 o servidor envia uma mensagem de boas vindas. Em seguida o usuário informa que deseja trabalhar com a instância número 18, e solicita que o sistema informe o seu estado. O servidor responde (na linha 6) que já executou a instância. Outras respostas poderiam ser que está na fila aguardando (*Queue*) ou que está sendo executada naquele momento (*Running*). Por fim, o usuário solicita a resposta gerada, que está no arquivo out.txt.

## 5.2.2 Interface

Para abstrair os comandos do gerenciador de conexões, e permitir que este seja acessado de fora da rede segura, foi desenvolvida uma interface amigável para o servidor. Esta interface é acessada como uma página de Internet<sup>6</sup> e fica hospedada no computador que faz a conexão da rede segura com o meio externo. Esta parte do código do servidor foi escrita em linguagem PHP [47], que é uma linguagem que permite gerar *páginas dinâmicas*. Quando a página é solicitada ao servidor de páginas, o código escrito em PHP é executado e uma página é gerada especificamente para aquela conexão. Isto permite que quando o usuário solicitar uma página de respostas, por exemplo, o servidor de páginas se conecta com o gerenciador de instâncias, consulta as informações necessárias e gera a página de acordo com a resposta do gerenciador.

<sup>5</sup>ASCII - *American Standard Code for Information Interchange*, código padronizado americano para troca de informações. Conjunto de códigos criado em 1961 para o computador representar números, letras, pontuação e outros caracteres.

<sup>6</sup>No momento da escrita desta dissertação esta interface pode ser acessada pelo endereço <http://www.das.ufsc.br/otimizacao>.

```
1  deconto@deconto:~> telnet localhost 1209
2  100 hello.  welcome to das-ufsc optimization server
3  SETINST 18
4  100 working with instance 18
5  GETSTATUS
6  101 6 FINISHED
7  GETFILE out.txt
8  100 sending file
9      ANP/PRH34 Gas-lift Optimization
10     MILP answer
11     -----
12     Optimum profit:  30787.4
13     Oil Well Injection Rate
14     1 - Oil Well 80
15     2 - Oil Well 154
16     3 - Oil Well 0
17     4 - Oil Well 0
18     .
19  100 end of file
20  QUIT
21  100 bye
```

Figura 5.6: Exemplo de conexão com o gerenciador de instâncias

A interface possui dois grupos de código bastante semelhantes. O primeiro possui um ambiente amigável para ser acessado pelo usuário, e permite que este escolha quais arquivos serão carregados, e verifique o estado de execução e a resposta de sua instância. No segundo as funcionalidades são as mesmas do primeiro, mas o ambiente não permite interação com o usuário (como botões e *links*). Foi gerado para que outros programas utilizem os recursos do servidor (no caso, o Ambiente de Especificação de Problema), uma vez que a comunicação do programa com o servidor é feita de maneira mais direta, com chamadas http [48, 49].

Na Figura 5.7 apresentamos o ambiente para utilizar o servidor de otimização a partir de uma página de Internet. No topo podemos observar as três funcionalidades disponíveis: *Submit data* permite que o usuário submeta uma nova instância para ser executada no servidor; *Get Status* é utilizado para verificar o estado de uma instância submetida, podendo ser *Aguardando*, *Executando* ou *Concluído*; a opção *Get Answer* retorna para o usuário a resposta obtida da execução de uma dada instância. Abaixo estão os campos para submeter uma nova instância: *E-mail* é o endereço eletrônico do usuário, caso ele deseje que a resposta lhe seja enviada por este meio; *Commands* é o arquivo que contém os comandos que o servidor deve utilizar; e *Data File n* são os campos destinados aos arquivos de instância que o usuário deseja submeter.

### 5.3 Estudo de caso

Nesta seção apresentaremos como utilizar a interface de otimização para obter as taxas de injeção ótimas em uma instância pequena. Para problemas em escala maior o procedimento é exatamente o mesmo.

Considere um campo de petróleo com quatro poços operando por *gas-lift*. Este campo possui um banco com três compressores e toda a produção é enviada para um separador. Na interface é possível

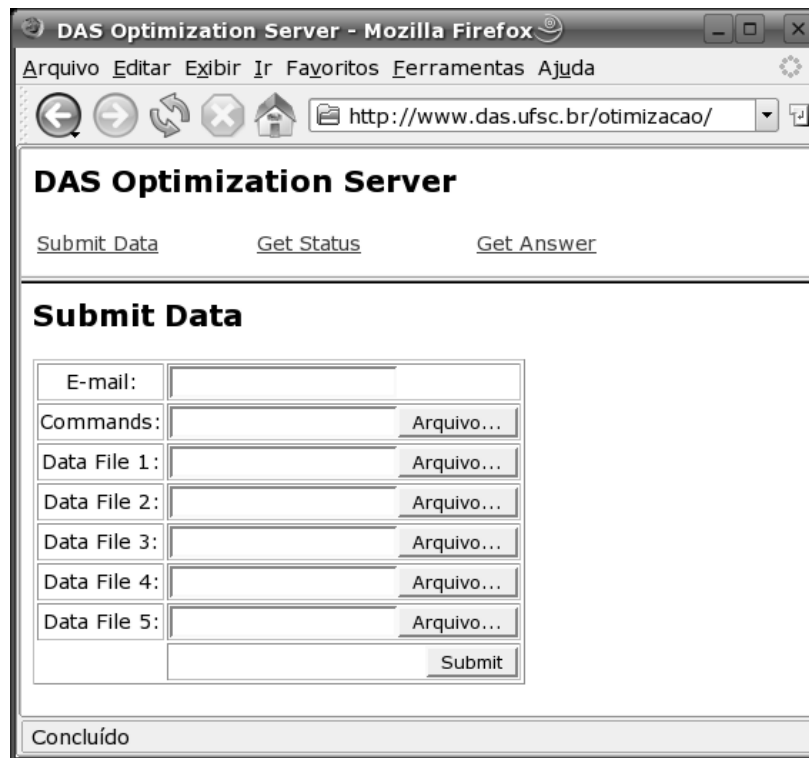


Figura 5.7: Interface para o usuário do servidor de otimização

$n$	$(q_{in}^{n,1}, q_{out}^{n,1})$	$(q_{in}^{n,2}, q_{out}^{n,2})$	$(q_{in}^{n,3}, q_{out}^{n,3})$	$\gamma_o^n$	$\gamma_g^n$	$\gamma_w^n$
1	( 80, 960)	(200, 1044)	(267, 1060)	0,70	0,20	0,10
2	( 80, 998)	(133, 1140)	(200, 1412)	0,75	0,17	0,08
3	( 80, 1108)	(133, 1132)	(267, 1652)	0,65	0,25	0,10
4	( 80, 1090)	(133, 1200)	(267, 1500)	0,66	0,24	0,10

Tabela 5.1: Descrição dos campos de petróleo para o estudo de caso

abrir um editor esquemático do campo, e montar um diagrama como o apresentado na Figura 5.2 arrastando os componentes para dentro do painel de edição. Este diagrama não é necessário para que os problemas sejam instanciados, mas em casos de tamanho médio e grande ele pode ajudar o usuário organizando as informações do campo.

Em seguida devemos caracterizar o campo. Como nossos algoritmos são baseados em um critério econômico, é necessário que sejam informados os preços do óleo e gás produzido, assim como o custo de tratamento da água extraída. Em nosso exemplo assumimos \$ 20,00 e \$ 2,00 para os preços do óleo e gás produzidos, e \$ 1,00 para o custo de tratamento da água. Descrevemos cada poço  $n$  com uma CPP contendo três pontos cada. A Tabela 5.1 apresenta os pontos da curva de cada poço, assim como suas proporções de óleo, gás e água ( $\gamma_o^n$ ,  $\gamma_g^n$  e  $\gamma_w^n$ ) produzidas.

No banco de compressores consideramos que a capacidade de cada compressor é 60, 60 e 80 unidades de gás comprimido, resultando em 200 unidades de capacidade total de injeção, a um custo comum de \$ 5,00. Como o algoritmo de otimização que utilizaremos não considera restrições no separador, ele não foi caracterizado. Admitimos também que não existe restrição de precedência em nossa instância, mas um grafo de precedência pode ser facilmente introduzido.

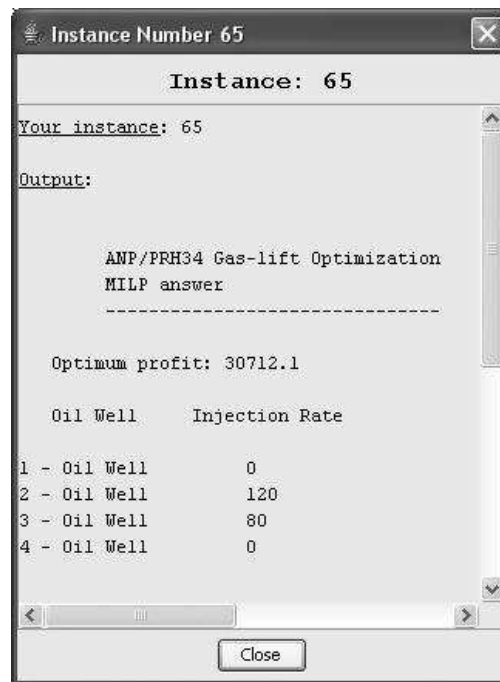


Figura 5.8: Apresentação da solução de um problema

A caracterização do campo não precisa ser feita nesta sequência. O sistema dá total liberdade para o usuário, de maneira que ele insira as informações conforme elas lhe forem disponibilizadas. Também é possível fazer a qualquer momento alterações na estrutura do campo, como adicionar novos poços ou os excluir.

Montado o campo é possível fazer chamadas ao algoritmo de otimização para que as melhores taxas de injeção de gás sejam calculadas. A primeira etapa consiste em escolher qual algoritmo será utilizado. Em seguida é aberto um diálogo com o usuário para que este configure detalhes da execução do algoritmo. No caso do modelo apresentado nesta dissertação, é possível configurar se deseja-se ou não que as restrições de precedência sejam consideradas. O passo seguinte é executar um pré-processamento dos dados do campo. Nesta etapa são verificadas inconsistências na instância, como por exemplo, um poço não caracterizado por sua CPP, ou um compressor habilitado com capacidade de compressão nula. Se o sistema gerar mensagens de alerta como as exemplificadas, o usuário ainda pode continuar o processo, mas mensagens de erro como incoerência nas frações de produção, exigem correções do usuário.

Por fim os dados são submetidos ao servidor. É possível escolher se a resposta deve ser enviada para a caixa de correio eletrônico do usuário, ou apenas poderá ser consultada pelo sistema. Então é aberto um diálogo com o usuário informando a solução ótima para sua instância (Figura 5.8).

## 5.4 Sumário

Neste capítulo apresentamos a ferramenta desenvolvida para o cômputo das taxas ótimas de injeção de gás em um campo operando por *gas-lift*. Adotamos uma estrutura cliente-servidor que permite

---

que o usuário caracterize localmente o problema, e que seja resolvido em uma máquina específica para este fim. O ambiente de especificação do problema permite que o usuário desenhe um esquema do campo de petróleo que está tratando, e descreva cada um de seus componentes, permitindo inclusive adicionar características que possam ser utilizadas em outros algoritmos de otimização. Permite também que a partir do próprio ambiente, solicite que o servidor obtenha as taxas ótimas de gás de injeção.

O servidor foi desenvolvido em duas partes: um gerenciador de instâncias, que efetivamente realiza as chamadas dos algoritmos de otimização; e uma interface que permite que o gerenciador seja acessado de maneira segura e amigável. Por fim, apresentamos um estudo de caso, em que mostramos passo a passo como utilizar o sistema para obter uma resposta ótima para um campo de petróleo simulado.

## Capítulo 6

# Conclusões e trabalhos futuros

A atual competição no mercado do petróleo e os altos custos envolvidos tornam qualquer ganho relativo no lucro de extração de hidrocarbonetos significativos. Neste contexto, a obtenção de taxas ótimas de injeção de gás no processo de elevação por *gas-lift*, que permitam maximizar o lucro da extração assume um papel de destaque nestes campos de produção. Apresentamos uma formulação para este problema baseada em programação linear inteira mista com restrições de quantidade máxima de gás de injeção e precedência de ativação de poços. Também foi implementada uma ferramenta para especificação e resolução destes problemas.

A programação linear inteira mista é uma ferramenta poderosa que permite o cômputo de soluções ótimas globais com variáveis de decisão inteiras em problemas que a função objetivo é linear. Para o problema da mochila as coberturas permitem buscar facetos para o poliedro de restrições. Se adicionarmos restrições de precedência, temos as  $K$ -coberturas que permitem gerar desigualdades fortes para o poliedro. Caso a função objetivo não seja linear, é possível linearizá-la por partes adicionando variáveis binárias de decisão. O modelo proposto por Sherali [28] se mostrou muito melhor que os outros devido a sua simplicidade e características poliedrais.

A formulação do problema de alocação ótima de gás de injeção utilizando programação linear inteira mista apresenta uma série de vantagens: generaliza os modelos existentes tratando explicitamente as decisões de ativação e desativação dos poços de petróleo; permite a aplicação intensiva da teoria e ferramentas algorítmicas do domínio da programação inteira, que permitem a obtenção de ótimos globais e uma medida da qualidade de soluções aproximadas; o modelo pode ser facilmente expandido para comportar outras restrições. Esta classe de problemas é NP-Difícil no sentido forte, então é importante a inserção de cortes para agilizar o processo de busca, principalmente em instâncias de grande porte.

Apresentamos uma formulação para obter desigualdades válidas para um subconjunto das variáveis de decisão do problema, e um procedimento de *lifting* para expandir estas desigualdades para todo o espaço das variáveis de decisão. Este procedimento de *lifting* envolve a resolução de uma seqüência de problemas, tão difíceis quanto resolver o próprio problema de alocação de gás. Então apresentamos um procedimento para o cômputo de fatores aproximados de *lifting*. Experimentos

computacionais mostraram que estas desigualdades de coberturas podem reduzir o número de iterações e o tamanho da árvore de *Branch and Bound*.

Criamos uma ferramenta para especificação e resolução de problemas de alocação de gás de injeção. Sua estrutura faz com que o cômputo das taxas de injeção seja realizada em um computador específico, com capacidade de processamento suficiente para resolver o problema. O emprego desta ferramenta permite que o operador do campo utilize a formulação desenvolvida sem necessariamente conhecer detalhes do modelo e do algoritmo de resolução. Em um estudo de caso mostramos que são tratadas apenas características que descrevem a operação do campo, sendo que detalhes do funcionamento do algoritmo são abstraídos do usuário.

Como sugestão para trabalhos futuros propomos que no algoritmo de *Branch and Bound* para resolução do problema sejam gerados cortes em cada nó da árvore de busca, que pode agilizar o processo de resolução, e sejam feitos estudos comparativos. No Apêndice B apresentamos uma formulação para o problema que leva em consideração a concavidade da curva de performance do poço. Propomos que esta formulação seja explorada, inicialmente fazendo um comparativo dos ganhos na diminuição do esforço computacional desta formulação, por reduzir o número de variáveis de decisão, com a formulação  $P_{pl}(G)$ . Em seguida, podem ser expandidos os conceitos de  $K$ -coberturas e *lifting* para esta formulação.

Na interface de otimização, propomos que seja estudada uma forma mais direta de conexão do ambiente de especificação do problema com o servidor, que permita o acompanhamento do cálculo da solução. Também propomos uma ampliação do modelo para outros tipos de restrições, como perdas de cargas na tubulação, interferência entre poços e acompanhamento da dinâmica do reservatório.



## Apêndice A

# Fatores de *lifting* exatos para uma floresta

Nesta seção apresentamos um algoritmo em programação dinâmica para o cômputo dos fatores exatos de *lifting* caso o grafo de restrições de precedência de ativação seja uma floresta. Tomemos como referência a formulação  $\mathcal{X}'$  apresentada em (4.10), da Seção 4.3.

As simplificações sobre (4.10e) podem ser programadas sequencialmente por (4.10e)–(4.10e) para reescrever o problema como:

$$\mathcal{X}'' : \varepsilon_t = \delta_t + \text{Maximize } \sum_{n \in A_t} \sum_{(n,k) \in B_{t,n}} a_{n,k} x_{n,k} \quad (\text{A.1a})$$

$$\text{Sujeito a } \sum_{n \in A_t} \sum_{(n,k) \in B_{t,n}} b_{n,k} x_{n,k} \leq b \quad (\text{A.1b})$$

$$\sum_{(m,i) \in B_{t,m}} x_{m,i} \geq \sum_{(n,j) \in B_{t,n}} x_{n,j}, \forall (m,n) \in E(\tilde{G}_t) \quad (\text{A.1c})$$

$$\sum_{(n,k) \in B_{t,n}} x_{n,k} \leq 1, \forall n \in r(\tilde{G}_t) \quad (\text{A.1d})$$

$$x_{n,k} \in \{0, 1\}, \forall (n,k) \in \cup_{n \in A_t} B_{t,n} \quad (\text{A.1e})$$

para os conjuntos  $A_t \subseteq N_t$ ,  $B_{t,n} \subseteq S_{t,n}$  e  $\tilde{G}_t = G[A_t]$ ; e o escalar  $\delta_t$ . Podemos assumir sem perda de generalidade que  $b$  e  $b_{n,k}$  são inteiros. Existe uma solução recursiva para  $\mathcal{X}''$  se  $\tilde{G}_t$  é uma floresta. Seja  $\mathcal{T} = \{n_1, n_2, \dots, n_T\}$  uma ordenação topológica de  $\tilde{G}_t$ . Para simplificar a notação, tomemos  $B_n = B_{t,n}$

e seja  $R_n$  os filhos de  $n$  em  $\tilde{G}_t$ , isto é,  $R_n = \{m : (n, m) \in E(\tilde{G}_t)\}$ . A recursão é:

$$P^y(n, q) : \\ J^y(n, q) = \text{Maximize } \sum_{(n,k) \in B_n} a_{n,k} x_{n,k} + J^y(R_n, q - \hat{q}) \quad (\text{A.2a})$$

$$\text{Sujeito a } \sum_{(n,k) \in B_n} x_{n,k} = y \quad (\text{A.2b})$$

$$\sum_{(n,k) \in B_n} b_{n,k} x_{n,k} \leq \hat{q} \quad (\text{A.2c})$$

$$\hat{q} \in \{0, \dots, q\} \quad (\text{A.2d})$$

$$x_{n,k} \in \{0, 1\}, \forall (n, k) \in B_n \quad (\text{A.2e})$$

$$P^y(R, q) :$$

$$J^y(R, q) = \text{Maximize } J^{\hat{y}}(n, \hat{q}) + J^y(R \setminus \{n\}, q - \hat{q}) \quad (\text{A.3a})$$

$$\text{Sujeito a } \hat{y} \in \{0\} \cup \{y\} \quad (\text{A.3b})$$

$$\hat{q} \in \{0, 1, \dots, q\} \quad (\text{A.3c})$$

$$n \prec \hat{n}, \forall \hat{n} \in R \setminus \{n\} \text{ de acordo com } \mathcal{T} \quad (\text{A.3d})$$

onde a interpretação dos problemas está a seguir.

$P^y(n, q)$  é a forma restrita de  $\mathcal{X}_t''$ , para um  $t$  particular, incluindo o poço  $n$  e seus sucessores em  $\tilde{G}_t$ , isto é,  $S_n = \{m \in V(\tilde{G}_t) : n \prec m\}$ , e tendo uma capacidade de compressão de gás de  $q$  unidades. Ainda, o poço  $n$  está ativado se  $y = 1$  e desativado se  $y = 0$ .  $J^y(n, q)$  é o valor da função objetivo de uma solução ótima para  $P^y(n, q)$ . Note que  $P^y(n, q)$  é sempre factível se  $y = 0$ . Se  $y = 1$  e  $q$  unidades são insuficientes para ativar o poço  $n$ , então  $P^y(n, q)$  é infactível e  $J^y(n, q) = -\infty$ . Note também que os sucessores de  $n$ ,  $S_n$ , podem ser ativados apenas se o poço  $n$  estiver ativo.

$P^y(R, q)$  é a forma restrita de  $\mathcal{X}_t''$ , para um  $t$  particular, considerando apenas as árvores com raízes nos elementos de  $R \subseteq V(\tilde{G}_t)$  onde  $(S_m \cup \{m\}) \cap (S_n \cup \{n\}) = \emptyset$  para todos  $m, n \in R$  distintos, e possuindo apenas  $q$  unidades de capacidade de compressão. Ainda, cada poço  $n \in R$  deve estar inativo se  $y = 0$ , mas os poços podem estar ativos ou inativos se  $y = 1$ .  $J^y(R, q)$  é o valor de uma solução ótima para  $P^y(R, q)$ . Note que  $P^y(R, q)$  é sempre factível independente dos valores de  $y$  e  $q$ , garantindo que  $J^y(R, q) \geq 0$ .

As recorrências (A.2a)–(A.2e) e (A.3a)–(A.3d) levam o problema à equivalência  $\mathcal{X}_t'' \equiv P^1(r(\tilde{G}_t), b)$  e, portanto,  $\varepsilon_t = \delta_t + J^1(r(\tilde{G}_t), b)$ . Um algoritmo em programação dinâmica pode ser desenvolvido a partir de uma execução seqüencial destas recorrências. A seguir, apresentamos uma visão geral do algoritmo.

Seja  $\mathcal{J}^y(n) = \{J^y(n, q) : q = 0, \dots, b\}$  um conjunto de soluções para  $P^y(n, q)$  variando  $q$ , em que  $\mathcal{J}(n) = \mathcal{J}^0(n) \cup \mathcal{J}^1(n)$  é o conjunto de todas as soluções. Seja  $\mathcal{J}^y(R) = \{J^y(R, q) : q = 0, \dots, b\}$  o conjunto de soluções para  $P^y(R, q)$  variando  $q$  e  $y \in \{0, 1\}$ , onde  $R \subseteq R_n$  para algum  $n \in V(\tilde{G}_t)$  ou  $R \subseteq r(\tilde{G}_t)$ . Ainda, seja  $\mathcal{J}(R) = \mathcal{J}^0(R) \cup \mathcal{J}^1(R)$  o conjunto de todas as soluções.

O algoritmo consiste em varrer os nós de  $\tilde{G}_t$  em ordem reversa de  $\mathcal{T}$ , isto é, das folhas da floresta até as raízes, computando  $\mathcal{J}(n)$  em cada nó  $n$ , sobre  $\mathcal{J}(R_{n,i})$  para  $i = j, j-1, \dots, 1$ , onde  $R_{n,i} = \{r_i^n, \dots, r_j^n\}$  e  $R_n = \{r_1^n, \dots, r_j^n\}$  é o conjunto de filhos de  $n$  obtidos de acordo com a ordem topológica  $\mathcal{T}$ . Pode ser mostrado que este algoritmo é executado em tempo  $\Theta(4|V|q^2)$  e utiliza  $\Theta(4|V|q)$  unidades de memória para armazenar as tabelas.

## Apêndice B

# Modelo linear por partes considerando concavidade

Propomos aqui um modelo linear por partes para o problema de alocação ótima de gás de injeção que considera a concavidade das curvas. Segundo Sherali [28], se um conjunto de pontos formar uma região côncava, é necessário apenas uma variável de decisão binária por região.

Esta formulação aproveita o fato que a curva de performance do poço (CPP) ou é côncava, ou pode ser dividida em regiões côncavas. Esta formulação não reduz a complexidade computacional da resolução do problema. Entretanto, há uma redução no número de variáveis de decisão inteiras, que pode levar a um número menor de nós no algoritmo de *Branch and Bound* ou de planos de corte, e por conseqüência em uma redução nos custos computacionais para obtenção da resposta.

Para formulação deste modelo utilizamos as definições apresentadas no Capítulo 4.

**Definição B.1** Para um poço  $n$ , um conjunto  $T^n = \{k : q^{n,k} \in Q^n\} = \{t_1^n, \dots, t_{\theta^n}^n\}$  de índices consecutivos de pontos representam uma região côncava se para todos os índices  $a, b$  e  $c \in T^n$ ,  $a < b < c$ , valer:

$$q_{out}^{n,b} \geq \left( \frac{q_{in}^{n,c} - q_{in}^{n,b}}{q_{in}^{n,c} - q_{in}^{n,a}} \right) q_{out}^{n,a} + \left( \frac{q_{in}^{n,b} - q_{in}^{n,a}}{q_{in}^{n,c} - q_{in}^{n,a}} \right) q_{out}^{n,c} \quad (\text{B.1})$$

Uma região côncava é máxima se  $T^n \cup \{t_1^n - 1\}$  e  $T^n \cup \{t_{\theta^n}^n + 1\}$  não representam uma região côncava.

Dividimos  $K^n = \{1, \dots, \kappa(n)\}$ , que é o conjunto dos índices de  $Q^n$ , em  $\mu(n)$  conjuntos  $T_m^n$  que representam regiões côncavas, tal que:

- $\cup_{m=1}^{\mu(n)} T_m^n = K^n$ ;
- $|T_m^n| \geq 2$ ,  $m = 1, \dots, \mu(n)$ ;
- $t_{m-1, \theta^n}^n = t_{m,1}^n$ ,  $m = 2, \dots, \mu(n)$ .

Se os conjuntos  $T_m^n$ ,  $m = 1, \dots, \mu(n)$ , representarem regiões côncavas máximas, a representação é ótima no número de variáveis de decisão.

$$P_{pl}^{con}(G) : \text{Maximize } f = \sum_{n=1}^N \sum_{m=1}^{\mu(n)} \sum_{k \in T_m^n} f^{n,k} \lambda_{m,k}^n \quad (\text{B.2a})$$

$$\text{Sujeito a: } \sum_{n=1}^N q_{in}^n \leq q_{in}^{max} \quad (\text{B.2b})$$

$$\sum_{m=1}^{\mu(i)} x_m^i \leq \sum_{m=1}^{\mu(j)} x_m^j \quad \forall (j, i) \in E(G) \quad (\text{B.2c})$$

$$\sum_{k=1}^{\mu(n)} x_m^n \leq 1 \quad \forall n \in R(G) \quad (\text{B.2d})$$

$$q_{in}^n = \sum_{m=1}^{\mu(n)} \sum_{k \in T_m^n} q_{in}^{n,k} \lambda_{m,k}^n \quad n = 1, \dots, N \quad (\text{B.2e})$$

$$\sum_{k \in T_m^n} \lambda_{m,k}^n = x_m^n \quad \forall (n, m) \in \Xi \quad (\text{B.2f})$$

$$\lambda_{m,k}^n \geq 0 \quad \forall (n, m) \in \Xi \quad \forall k \in T_m^n \quad (\text{B.2g})$$

$$x_m^n \geq 0 \quad \forall (n, m) \in \Xi \quad (\text{B.2h})$$

$$x_m^n \in \mathbb{Z} \quad \forall (n, m) \in \Xi \quad (\text{B.2i})$$

onde  $\Xi = \{(n, m) : n = 1, \dots, N \text{ e } m = 1, \dots, \mu(n)\}$ .

Se cada conjunto de pontos  $T_m^n$  possuir cardinalidade  $|T_m^n| = 2$ , então  $P_{pl}^{con}(G)$  pode ser reduzido a  $P_{pl}(G)$ . Isto é, se cada conjunto  $T_m^n$  possui dois pontos consecutivos que representam uma região de  $P_{pl}(G)$ , então  $P_{pl}^{con}(G)$  é equivalente a  $P_{pl}(G)$ .

## B.1 Exemplo ilustrativo

Considere que para um dado poço  $n$  a CPP possui a configuração apresentada na Figura B.1. Observe que os pontos  $(1, 1)$ ,  $(2, 3)$  e  $(3, 2)$  formam uma região côncava máxima, e que os pontos  $(3, 2)$ ,  $(4, 3)$ ,  $(5, 3)$  e  $(6, 2)$  formam outra região. Assim, assumimos  $\mu(n) = 2$  regiões côncavas, e temos que  $T_1^n = \{1, 2, 3\}$  e  $T_2^n = \{3, 4, 5, 6\}$ . Note que  $T_1^n \cup T_2^n = \{1, 2, 3, 4, 5, 6\} = K^n$ .

Utilizando esta formulação temos 8 variáveis de decisão reais ( $\lambda$  e  $q_{in}^n$ ) e 2 variáveis inteiras para representar este poço  $n$ . Se aplicarmos esta curva no modelo  $P_{pl}(G)$  teremos 11 variáveis de decisão reais e 5 inteiras. Podemos observar uma redução significativa nas variáveis de decisão, principalmente nas inteiras, que levam a uma redução dos custos computacionais para resolução do problema.

Se fizermos  $T_1^n = \{1, 2\}$ ,  $T_2^n = \{2, 3\}$ ,  $T_3^n = \{3, 4\}$ ,  $T_4^n = \{4, 5\}$  e  $T_5^n = \{5, 6\}$ , serão necessárias 11 variáveis reais e 5 inteiras, que corresponde à formulação  $P_{pl}(G)$  original.

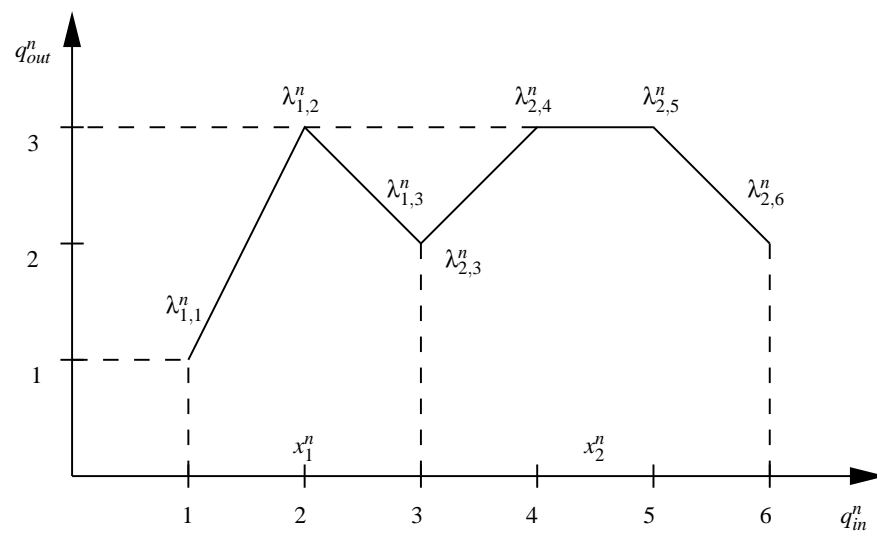


Figura B.1: Exemplo de CPP considerando concavidade nas variáveis de decisão.

# Referências Bibliográficas

- [1] MCKIE, C. et al. Economic benefits from automated optimization of high pressure gas usage in an oil production system. In: *SPE Production and Operation Symposium*. Oklahoma City, Oklahoma: [s.n.], 2001. Paper SPE 67187.
- [2] THOMAS, J. *Fundamentos de Engenharia de Petróleo*. [S.l.]: Editora Interciência Ltda, 2001.
- [3] JAHN, F.; COOK, M.; GRAHAM, M. *Hydrocarbon exploration and production*. [S.l.]: Elsevier Science, 2003.
- [4] BUITRAGO, S.; RODRÍGUEZ, E.; ESPIN, S. D. Global optimization techniques in gas allocation for continuous flow gas lift systems. In: *Proc. SPE Gas Technology Conference*. Calgary, Canada: [s.n.], 1996. p. 375–379. Paper SPE 15616.
- [5] ECONOMIDES, M.; HILL, A.; EHLIG-ECONOMIDES, C. *Petroleum Production Systems*. [S.l.]: Prentice Hall, 1993.
- [6] KANU, E. P.; MACH, J. M.; BROWN, K. E. Economic approach to oil production and gas allocation in continuous gas lift. *Journal of Petroleum Technology*, p. 1887–1892, Outubro 1981.
- [7] MACH, J.; PROANO, E.; BROWN, K. E. A nodal approach for applying systems analysis to the flowing and artificial lift of oil or gas well. In: *Society of Petroleum Engineers*. [S.l.: s.n.], 1979. Paper SPE 8025.
- [8] ALARCÓN, G.; TORRES, C.; GÓMEZ, L. Global optimization of gas allocation to a group of wells in artificial lift using nonlinear constrained programming. *ASME Journal of Energy Resources Technology*, v. 124, p. 262–268, Dezembro 2002.
- [9] NAKASHIMA, P.; CAMPONOGARA, E. Solving a gas-lift optimization problem by dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics—Part A*, v. 36, n. 2, p. 407–414, 2006.
- [10] GAREY, M.; JOHNSON, P. *A Guide to Theory or NP-Completeness*. New York, NY: W.H. Freeman, 1979.
- [11] MAYHILL, T. D. Simplified method for gas-lift well problem identification and diagnosis. In: *Proc. SPE Annual Fall Meeting*. Houston, Texas: [s.n.], 1974. Paper SPE 5151.

- [12] REDDEN, J. D.; SHERMAN, T. A. G.; BLANN, J. R. Optimizing gas-lift systems. In: *Proc. of the 49th Annual Fall Meeting of the Society of Petroleum Engineers of AIME*. Houston, Texas: [s.n.], 1974. Paper SPE 5150.
- [13] NISHIKIORI, N. et al. An improved method for gas lift allocation optimization. In: *Proc. SPE Annual Technical Conference and Exhibition*. San Antonio, TX: [s.n.], 1989. Paper SPE 19711.
- [14] NISHIKIORI, N. et al. An improved method for gas lift allocation optimization. *ASME Journal of Energy Resources Technology*, v. 117, p. 87–92, 1995.
- [15] FANG, W. Y.; LO, K. K. A generalized well-management scheme for reservoir simulation. *SPE Reservoir Engineering*, v. 11, n. 2, p. 116–120, May 1996.
- [16] WANG, P.; LITVAK, M.; AZIZ, K. Optimization of production operations in petroleum fields. In: *Proc. 17th World Petroleum Congress*. Rio de Janeiro, Brazil: [s.n.], 2002.
- [17] CAMPONOGARA, E.; NAKASHIMA, P. H. R. Applying dynamic programming to a gas-lift optimization problem. In: *Proc. 2nd Brazilian Congress on Research and Development in Oil and Gas*. Rio de Janeiro, Brazil: [s.n.], 2003.
- [18] WILLIAMS, H. *Model Building in Mathematical Programming*. 4. ed. New York, NY: John Wiley & Sons, 1999.
- [19] NAKASHIMA, P. *Otimização de Processos de Produção de Petróleo via Injeção Contínua de Gás*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, Maio 2004.
- [20] VANDERBEI, R. J. *Linear Programming: Foundations and Extensions*. Boston: Kluwer Academic Publishers, 1996. Disponível em: <[citeseer.ist.psu.edu/vanderbei96linear.html](http://citeseer.ist.psu.edu/vanderbei96linear.html)>.
- [21] WOLSEY, L. *Integer Programming*. New York, NY: Wiley-Interscience, 1998. (Wiley-Interscience series in discrete mathematics and optimization).
- [22] NEMHAUSER, G. L.; WOLSEY, L. A. *Integer and Combinatorial Optimization*. [S.l.]: John Wiley & Sons, 1988.
- [23] WRIGHT, S. *Primal-Dual Interior-Point Methods*. Philadelphia, PA: Soc for Industrial & Applied Math, 1997.
- [24] 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.
- [25] KAPARIS, K.; LETCHFORD, A. A cut-and-branch algorithm for the multidimensional knapsack problem. Lancaster University. nov 2005.
- [26] BOYD, E. Polyhedral results for the precedence-constrained knapsack problem. *Discrete Applied Mathematics*, v. 41, p. 185–201, 1993.
- [27] PADBERG, M. Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters*, v. 27, p. 1–5, 2000.



- [28] SHERALI, H. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Operations Research Letters*, v. 28, p. 155–160, 2001.
- [29] CONTO, A. de; CAMPONOGARA, E. Alocação Ótima de lift-gas: Modelos de linearização por partes. In: *3º Congresso Brasileiro de P&D em Petróleo e Gás*. Salvador, BA: [s.n.], 2005.
- [30] CAMPONOGARA, E.; CONTO, A. de. Gas-lift allocation under precedence constraints: Piecewise-linear formulation and  $k$ -covers. In: *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*. Sevilha, Espanha: [s.n.], 2005. p. 4422–4427.
- [31] ILOG INCORPORATED. *ILOG CPLEX 9.0: Getting Started*. Mountain View, CA, 2003.
- [32] STROUSTRUP, B. *The C++ Programming Language*. 3. ed. Reading: Addison-Wesley, 1997.
- [33] MEHLHORN, K.; NAHER, S. *LEDA: A Platform for Combinatorial and Geometric Computing*. [S.l.]: Cambridge University Press, 1999.
- [34] DEITEL, H.; DEITEL, P. *Java How to Program*. 6. ed. Upper Saddle River, NJ: Prentice-Hall, 2004.
- [35] CAMPONOGARA, E.; NAKASHIMA, P. Optimal allocation of lift-gas rates under multiple facility constraints: A mixed-integer linear programming approach. *A ser publicado no Transactions of ASME*, 2007.
- [36] MARTIN, R. *UML Tutorial: Part 1 - Class Diagram*. [S.l.]. Disponível em: <[www.objectmentor.com/publications/umlClassDiagrams.pdf](http://www.objectmentor.com/publications/umlClassDiagrams.pdf)>.
- [37] FOWLER, M.; KOBRYN, C.; BOOCH, G. *UML Essencial*. 3. ed. Porto Alegre: Bookman, 2005.
- [38] CORMEN, T. et al. *Introduction to Algorithms*. 2. ed. [S.l.]: MIT Press, 2001.
- [39] LEITHOLD, L. *O Cálculo com Geometria Analítica*. 3. ed. São Paulo: Harbra, 1993.
- [40] DEITEL, A.; DEITEL, P.; NIETO, R. *XML: Como Programar*. 1. ed. Porto Alegre: Bookman, 2003.
- [41] MITCHELL, M.; OLDHAM, J.; SAMUEL, A. *Advanced Linux Programming*. 1. ed. Indianapolis, Indiana: New Riders, 2001.
- [42] OLIVEIRA, R.; CARISSIMI, A.; TOSCANI, S. *Sistemas Operacionais*. 3. ed. Porto Alegre: Editora Sagra-Luzzato, 2004.
- [43] TANENBAUM, A. *Sistemas Operacionais Modernos*. 2. ed. Upper Saddle River, NJ: Prentice-Hall, 2003.
- [44] CASAD, J.; WILLSEY, B. *Aprenda TCP/IP em 24 horas*. 5. ed. São Paulo: Campus, 1999.
- [45] POSTEL, J. *RFC 821 - Simple Mail Transfer Protocol*. University of Southern California, 1982. Disponível em: <<http://www.ietf.org/rfc/rfc0821.txt>>.

- 
- [46] MYERS, J.; ROSE, M. *RFC 1725 - Post Office Protocol - Version 3*. Carnegie-Mellon University, 1994. Disponível em: <<http://www.ietf.org/rfc/rfc1725.txt>>.
- [47] SOARES, W. *Programando em PHP: Conceitos e Aplicações*. São Paulo: Érica, 2000.
- [48] HAROLD, E. *Java Network Programming*. 3. ed. Cambridge, Massachusetts: O'Reilly Media, 2004.
- [49] FIELDING, R. et al. *RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1*. The Internet Society, 1999. Disponível em: <<http://www.ietf.org/rfc/rfc2616.txt>>.