

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática
Curso de Engenharia de Produção

**Um Método Baseado em Branch-And-Bound para a Solução
do Problema de Cometimento de Unidades**

Luís André de Oliveira Jayme

TCC-EP-37-2007

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática
Curso de Engenharia de Produção

**Um Método Baseado em Branch-And-Bound para Solução
do Problema de Cometimento de Unidades**

Luís André de Oliveira Jayme

TCC-EP-37-2007

Relatório Técnico 1 apresentado como requisito de
avaliação no curso de graduação em Engenharia de
Produção na Universidade Estadual de Maringá – UEM.
Orientadora: Prof^ª.Dr^ª. Márcia Marcondes Altimari Samed

**Maringá - Paraná
2007**

Luís André de Oliveira Jayme

**Um Método Baseado em Branch-And-Bound para a Solução do
Problema de Cometimento de Unidades**

Este exemplar corresponde à redação final do Trabalho de Conclusão de Curso aprovado como requisito parcial para obtenção do grau de Bacharel em Engenharia de Produção da Universidade Estadual de Maringá, pela comissão formada pelos professores:

Orientador(a): Prof^(a). Dr^(a). Márcia M. Altimari Samed
Departamento de Informática, CTC

Maringá, outubro de 2007

RESUMO

Um sistema de geração de energia elétrica de uma região deve ser capaz de atender as necessidades da população e planejar a produção de acordo com um perfil de demanda de curto ou longo prazo. Foi desenvolvida uma metodologia e um algoritmo capaz de solucionar o problema Cometimento de Unidades para um valor de demanda fixo. O método proposto parte de uma solução já existente para o problema implícito de Despacho Econômico e formula uma abordagem Branch-and-Bound para encontrar a solução do problema de Cometimento de Unidades. Diversas heurísticas são desenvolvidas e testadas no intuito de melhorar a precisão e a eficiência do método, gerando resultados encorajadores.

Palavras-chave: *Unit Commitment, Economic Dispatch, Branch-and-Bound, Integer Programming, Operational Research, Electric Engineering.*

SUMÁRIO

RESUMO	iv
SUMÁRIO	v
LISTA DE ILUSTRAÇÕES	vii
LISTA DE TABELAS	viii
LISTA DE QUADROS	ix
LISTA DE ABREVIATURAS E SIGLAS	x
1 INTRODUÇÃO	1
2 REVISÃO DE LITERATURA	3
2.1 Discussão de Metodologias	5
2.2 Programação Inteira	10
2.2.1 Algoritmos Gulosos	11
2.2.2 Métodos Heurísticos	12
2.2.3 Enumeração Exaustiva.....	13
2.2.3.1 Algoritmo de Enumeração Exaustiva	15
2.2.4 Branch-and-Bound.....	17
2.2.4.1 Idéias Gerais	18
2.2.4.2 Sub-rotinas do BB.....	19
2.2.4.3 Eliminando nós por limites	20
2.2.4.4 Eliminando nós por infactibilidade.....	23
2.2.4.5 Determinando uma solução factível	23
2.2.4.6 Ajustando variáveis.....	24
2.2.4.7 Encontrando uma variável separável.....	24
2.2.4.8 Algoritmo genérico de BB para PPI 0-1	25
2.2.5 Cutting Plane	29
3 METODOLOGIA	30
3.1 Escolha do método.....	30
3.2 Modelagem do problema	31
3.3 Método BB para o PCU	32
3.3.1 Problema relaxado para o PCU.....	34
3.3.2 Branch e redução do PCU inicial.....	35
3.3.3 Índice de Utilização Livre	36
3.3.4 Procedimento aproximativo utilizando Lista de Prioridades	37
3.3.4.1 Lista de prioridades.....	39
3.3.4.2 Aprimoramento da lista de prioridades.....	40
3.3.4.3 Aprimoramento do parâmetro β	42
3.3.5 Escolha da unidade para o branch	43
3.3.6 Sub-rotina Restringir	43
3.3.7 Demais sub-rotinas do método BB	44
4 RESULTADOS E DISCUSSÃO	46
4.1 Calibração de parâmetros	46
4.1.1 Parâmetro γ	49
4.1.2 Parâmetro β	52
4.1.3 Parâmetro α	54
4.2 Aprimoramento da lista de prioridades.....	55
4.3 Critério de desempate para os IUL	58

4.4	Resultados para a versão final do método proposto	58
5	CONCLUSÃO.....	62
	REFERÊNCIAS	63
	APÊNDICE	66
	ANEXO	76

LISTA DE ILUSTRAÇÕES

FIGURA 1: ÁRVORE DE BUSCA PARA A ENUMERAÇÃO EXAUSTIVA.....	14
--	----

LISTA DE TABELAS

TABELA 1: RESULTADOS DA BUSCA EXAUSTIVA PARA UM CASO DE 6 GERADORES.....	48
TABELA 2: RESULTADOS DA BUSCA EXAUSTIVA PARA UM CASO DE 13 GERADORES.....	48
TABELA 3: RESULTADOS DO PARÂMETRO Γ PARA UM CASO DE 6 GERADORES.....	49
TABELA 4: RESULTADOS DO PARÂMETRO Γ PARA UM CASO DE 13 GERADORES.....	50
TABELA 5: RESULTADOS DO PARÂMETRO B PARA UM CASO DE 6 GERADORES.....	52
TABELA 6: RESULTADOS DO PARÂMETRO B PARA UM CASO DE 13 GERADORES.....	52
TABELA 7: RESULTADOS DO PARÂMETRO A PARA UM CASO DE 6 GERADORES.....	54
TABELA 8: RESULTADOS DO PARÂMETRO A PARA UM CASO DE 13 GERADORES.....	54
TABELA 9: RESULTADOS DOS PARÂMETROS PARA UM CASO DE 13 GERADORES (COM APRIMORAMENTO DA LISTA DE PRIORIDADES)	56
TABELA 10: RESULTADOS FINAIS DO PCU PARA UM CASO DE 13 GERADORES.....	58
TABELA 11: RESULTADOS DO PDE PARA UM CASO DE 13 GERADORES.....	59
TABELA 12: RESULTADOS FINAIS DO PCU PARA UM CASO DE 6 GERADORES.....	59
TABELA 13: RESULTADOS DO PDE PARA UM CASO DE 6 GERADORES.....	60
TABELA 14: RESULTADOS DO PDE E PCU PARA UM CASO DE 12 GERADORES.....	60

LISTA DE QUADROS

QUADRO 1: VALORES DAS VARIÁVEIS PARA O CASO DE 6 GERADORES.....	46
QUADRO 2: VALORES DAS VARIÁVEIS PARA O CASO DE 13 GERADORES.....	47
QUADRO 3: ANÁLISE DE SENSIBILIDADE PARA Γ	51
QUADRO 4: ANÁLISE DE SENSIBILIDADE PARA B.....	53
QUADRO 5: ANÁLISE DE SENSIBILIDADE PARA A.....	55
QUADRO 6: ANÁLISE DE SENSIBILIDADE PARA E.....	57

LISTA DE ABREVIATURAS E SIGLAS

BB	<i>Brand-and-Bound</i>
CP	<i>Cutting Plane</i>
PCU	Problema de Cometimento de Unidades
PDE	Problema de Despacho Econômico
PI	Programação Inteira
PIML	Programação Inteira Mista com Linear
PL	Programação Linear
PPI	Problema de Programação Inteira
PPIM	Problema de Programação Inteira Misto
PPIML	Problema de Programação Inteira Mista com Linear
PPL	Problema de Programação Linear

1 INTRODUÇÃO

A demanda de energia elétrica apresenta uma grande variação durante as horas do dia e entre os dias de semana. No horário comercial o consumo é maior do que de madrugada, e aos sábados e domingos, o consumo é menor do que nos dias de semana (DAMOUSIS e BAKIRTZIS, 2004; PADHY, 2004). Quando ocorrem feriados, o perfil de demanda da semana também é diferenciado, já que o horário comercial é o maior causador da variação de demanda. Caso a geração permaneça constante, a energia sobrarão durante os horários de pouco consumo e poderá faltar durante os horários de maior consumo.

Para que não ocorra desperdício e tampouco falta de energia, é necessário adequar a produção à demanda. Quanto melhor for a solução obtida para este problema, menor serão os custos e maior será a confiabilidade do sistema de geração de energia.

O Problema de Comprometimento de Unidades (PCU), ou em inglês, *Unit Commitment*, consiste em decidir quais unidades geradoras devem ser acionadas em cada período a fim de satisfazer uma previsão de demanda de energia (PADHY, 2004; SAKSORNCHAI *et al*, 2005). Para solucionar este problema, também é necessário solucionar o Problema de Despacho Econômico (PDE), ou em inglês, *Economic Dispatch*, o qual, de posse do número de geradores, calcula o quanto cada um deverá produzir. É importante observar que o PDE encontra-se dentro do PCU e, para resolver o último, alguns dos métodos existentes exigem que o primeiro problema seja resolvido repetidas vezes (LI *et al*, 1997).

O PCU pode ser tratado de maneira mais abrangente, considerando uma região com várias usinas, ou mais específica, no caso de uma usina com vários geradores. Quando se tem um conjunto de várias usinas, é necessário apontar quais delas funcionarão durante cada período e qual quantidade de energia cada uma irá produzir. Traduzindo para o ponto de vista de uma única usina, é preciso decidir quais geradores serão ativados e com que potência os mesmos deverão funcionar.

A segunda abordagem, na qual existe uma usina e vários geradores, será adotada para este trabalho. Esta opção se justifica principalmente pela intenção de contribuir com o método proposto por Rodrigues (2007), para o PDE, a fim de construir, valendo-se deste, um método capaz de solucionar o PCU, que é o problema real existente em sistemas de geração. Esta decisão, no entanto, não afetou a modelagem do problema, de modo que o método proposto pode ser utilizado para ambos os fins.

O problema de geração de energia é um tema que não possui uma solução definitiva, o que abre espaço para novas propostas. Além disso, é um assunto de permanente importância frente ao desenvolvimento mundial e o crescimento da população.

O objetivo do trabalho é desenvolver um método para solucionar um PCU que planeje a geração de uma usina termoeletrica de acordo com um perfil demanda horária. Será desenvolvido um método capaz de utilizar a solução do problema interno de despacho econômico sugerida por Rodrigues (2007).

Depois de elaborado o método, pretende-se implementá-lo computacionalmente e validá-lo. Os resultados obtidos serão comparados com resultados de outros trabalhos, de modo que o desempenho do método concebido possa ser avaliado.

Este trabalho está organizado em 5 capítulos. O Capítulo 2 explora a literatura em busca de definições e formulações do problema, discute os métodos existentes e descreve em detalhe um método *Branch-and-Bound* genérico. O Capítulo 3 discorre sobre a modelagem definida para o problema e sobre o método desenvolvido para solucioná-lo. O Capítulo 4 apresenta e discute os resultados obtidos. Finalmente, no Capítulo 5, são feitas as considerações finais sobre o trabalho.

2 REVISÃO DE LITERATURA

O PCU é um problema de Pesquisa Operacional da área de Programação Não-Linear que consiste em determinar o agendamento das unidades geradoras dentro de um sistema (SHEBLE' e FAHD, 1994; PADHY, 2004). Para uma melhor contextualização, este capítulo será iniciado com a apresentação de uma formulação geral para o problema, composta pelas propostas de Padhy (2004), Sheble' e Fahd (1994) e Tseng *et al* (1997).

O PCU possui a seguinte formulação genérica:

$$\text{Minimize } CO = \sum_{i=1}^N \sum_{t=1}^T CC_{it}(P_{it}) + CM_{it}(P_{it}) + CI_{it} + CD_{it} \quad (1)$$

A Equação 1, que é a função objetivo, consiste na minimização do Custo Operacional (CO), que é o somatório duplo para N unidades geradoras e T períodos de tempo (convencionalmente expressado em horas) dos Custos de Combustível (CC), Custos de Manutenção (CM), Custos de Inicialização (CI) e Custo de Desligamento (CD). Cada um destes custos possui uma formulação, dentre os quais convém definir, para fins deste trabalho, apenas o Custo de Combustível:

$$CC_{it}(P_{it}) = a_{it}P_{it}^2 + b_{it}P_{it} + c_{it} \quad (2)$$

Como se pode observar na Equação 2, o Custo de Combustível é uma função quadrática da potência utilizada da unidade geradora *i* em questão.

O PCU está sujeito a restrições operacionais e de equipamento, tais como:

- a) Demanda de potência a ser suprida;
- b) Número de geradores no sistema;
- c) Tempo mínimo que o gerador precisa permanecer ligado, depois de ligado;
- d) Tempo mínimo que o gerador precisa permanecer desligado, depois de desligado;
- e) Limite da taxa de partida do gerador;
- f) Limites de potência inferior e superior do gerador;

- g) Limites de funcionários na planta (que pode influenciar na quantidade que os geradores podem ser ligados e desligados simultaneamente);
- h) Geradores que não podem ser ativados no momento ou estão em manutenção;
- i) Geradores que devem ser ativados (por motivos de economia ou confiabilidade);
- j) Reserva *Spinning* (ou limites de reserva);
- k) Alimentação do gerador (pode mudar o limite superior da capacidade).

A Reserva *Spinning* é uma restrição particularmente importante para o PCU, que consiste na quantidade total de energia disponível pelas unidades do sistema subtraída da demanda atual e das perdas de transmissão a serem supridas. Esta reserva deve ser definida de modo que a perda ocasional de uma unidade possa ser compensada pela reserva existente nas outras unidades, em um intervalo específico de tempo.

A reserva *Spinning* pode constituir em unidades ligadas que tenham capacidade de fornecer mais energia quando necessário e em unidades que se encontram desligadas, mas que podem iniciar a produção de energia em tempo hábil. Geralmente estas últimas são unidades de inicialização rápida, como turbinas a gás ou a diesel. A reserva *Spinning* deve obedecer a algumas regras, sendo que algumas podem ser definidas por regulamentações governamentais. As regras variam desde definir a reserva como uma porcentagem do pico da demanda até calcular a mesma como uma função da probabilidade de não se ter geração suficiente para suprir a demanda. Em sistemas reais, deve-se considerar o limite de transmissão da linha, pois não faz sentido atribuir parte da reserva em uma unidade cuja comunicação com o sistema possui limitações que impediria o uso desta reserva, quando fosse necessária.

Os objetivos do PCU variam entre a maximização da receita, minimização do consumo de combustível, minimização de emissão de poluentes até a minimização dos custos, ou uma combinação entre eles, sendo que a minimização do custo operacional é considerada mais usual (SHEBLE' e FAHD, 1994). Por outro lado, Padhy (2004) considera a maximização do lucro mais adequada, enfatizando que a minimização dos custos não, necessariamente, causa o aumento dos lucros, visto que se ocorrer um aumento da receita maior do que um aumento dos custos, o lucro obtido será maior.

Sabe-se que o sequenciamento das operações em uma usina tem um grande impacto econômico. O combustível é, para a maioria das delas, o item de maior custo operacional e reduzir os custos de combustíveis em proporções pequenas, como 0,5%, pode resultar em uma economia de milhões de dólares por ano em grandes usinas (LI *et al*, 1997; BALDICK, 1995 apud PADHY, 2004; SAKSORNCHAI *et al*, 2005).

As características do problema, neste sentido, variam muito com o tipo de usina. Em um extremo, uma planta nuclear pode fornecer energia com um pequeno incremento de custo para cada MW/h adicional, no entanto possui um alto custo de inicialização, cujo processo também é demorado. Por outro lado, uma turbina de uma planta de geração a gás leva alguns minutos para ser iniciada, mas o custo adicional por MW/h é muito maior (LINDO, 2003; WOOD e WOLLENBERG, 1984).

A fim de minimizar os custos, a estratégia óbvia seria, com o aumento da demanda, acionar preferencialmente os geradores eficientes, que custam mais para iniciar, e em último caso, acionar os menos eficientes, com menor custo de inicialização. Com a diminuição da demanda, as unidades seriam desligadas em ordem reversa, de modo que as unidades menos eficientes sejam menos utilizadas (LINDO, 2003).

Mas esta estratégia apresenta um problema nos casos em que há um pico de demanda com uma curta duração. Nesta situação, seria mais econômico utilizar um gerador menos eficiente, porém com menor custo de inicialização (LINDO, 2003).

A programação das operações no sistema de geração pode ser em longo ou curto prazo. A programação a longo prazo é uma tarefa importante, pois considera restrições técnicas e econômicas ao longo do horizonte de planejamento de um ano. Todavia, a solução exata de um planejamento de longo prazo não é possível, devido ao tempo de processamento exorbitante e à extrapolação causada ao planejamento de curto prazo, que se torna inadequado ao passo que negligencia uma série de restrições, como tempos de manutenção, mudança de preços, entre outras (HUSSAIN, 1991; VEMURI e LEMONIDIS, 1991 apud PADHY, 2004; GUAN *et al*, 1992 apud PADHY, 2004).

2.1 Discussão de Metodologias

O problema de cometimento de unidades já foi resolvido por vários métodos. A maneira mais simplista de resolvê-lo é chamada Enumeração Exaustiva, que consiste em testar todas as

combinações possíveis de unidades geradoras e diferentes porcentagens de utilização para cada uma delas (SHEBLE' e FAHD, 1994; PADHY, 2004). Será apresentado, posteriormente, que esta solução é aplicável apenas quando se tem um número pequeno de geradores.

A Lista de Prioridades é um método eficiente que classifica as unidades geradoras baseando-se em características de custos operacionais. As unidades geradoras são ligadas em ordem de prioridade até que a demanda seja satisfeita. Vários autores aprimoraram o método de Lista de Prioridades, levando em conta outros fatores que acabaram por melhorar o resultado final (SHEBLE' e FAHD, 1994; PADHY, 2004).

A princípio, as unidades são ordenadas de acordo com o custo de carga total (VOORSPOOLS e D'HAESELEER, 2002). Como nem sempre todas são utilizadas completamente, Lee (1988) apud Voorspools e D'haeseleer (2002) introduziu um fator de comprometimento para cada uma das unidades, enquanto Voorspools e D'haeseleer (2000) as dividiu em partes constantes de uso de energia e ordena estas partes, ao invés das unidades como um todo.

Voorspools e D'haeseleer (2002) incrementaram o método e o definiram como Lista de Prioridade Avançada, no qual as unidades são classificadas em vários grupos. Estes grupos, em ordem de prioridade, são: grupo sempre utilizado, grupo de teste, grupo de pico, grupo de modulação e grupo inativo. É definida uma matriz de utilização para determinar qual unidade deve ser ativada de acordo com a demanda horária. O autor comparou o desempenho da Lista de Prioridades Avançada com o método de *Unit Decommittment* proposto por Tseng *et al* (2000) e identificou um ganho no tempo de processamento, que foi de 5 a 10 vezes menor e uma perda de precisão pequena, da ordem de 0,03 a 0,6%.

Senjyu *et al* (2003) desenvolveu uma Lista de Prioridade Extendida, que se mostrou extremamente mais eficiente e, em menor grau, mais preciso, quando comparado aos métodos de Algoritmo Genético proposto por Kazarlis *et al* (1996) e de Programação Evolucionária de Juste *et al* (1999).

O *Unit Decommittment* é um método que utiliza uma lógica inversa para solucionar o PCU, iniciando com todas as unidades ativadas e, em seguida, desativando aquelas que são menos convenientes, até encontrar uma configuração ótima (VOORSPOOLS e D'HAESELEER, 2002). A abordagem proposta por Tseng *et al* (1997) mostrou-se vantajosa em relação ao

tradicional método de Relaxação Lagrangeana, apresentando um erro aproximado de 0,2% e um tempo de processamento de 50 a 85% menor.

A Relaxação Lagrangeana é um método de maior complexidade matemática que funciona melhor para grandes quantidades de unidades geradoras, sendo, portanto, o caso em que é mais utilizada (PADHY, 2004; DIEU e ONGSAKUL, 2005; LI *et al*, 1997). A Relaxação Lagrangeana consiste em uma técnica de otimização que decompõe o problema em outros menores e mais fáceis, que são resolvidos iterativamente até que uma solução próxima da ótima é obtida. Os subproblemas são resolvidos independentemente e cada um determina o cometimento de uma única unidade. Os problemas são conectados através do multiplicador de Lagrange, que é adicionado ao problema inicial (SHEBLE' e FAHD, 1994). Para solucionar um problema utilizando este método, três questões devem ser respondidas: como obter um multiplicador de Lagrange, como resolver o problema relaxado e o quão próxima do ótimo a solução é (PADHY, 2004).

A Relaxação Lagrangeana é dita flexível às características do modelo e possibilita, com uma relativa facilidade, a inclusão de mais restrições. Sua maior dificuldade consiste na sua dependência de uma subestrutura ótima, como a Programação Dinâmica (PADHY, 2004; LI *et al*, 1997).

A Programação Dinâmica foi um dos primeiros métodos de otimização aplicados ao problema e ainda é utilizada. A lógica da Programação Dinâmica é subdividir o problema em outros menores, e tentar resolvê-los de forma ótima. Para que isto seja possível, o problema a ser resolvido deve possuir uma subestrutura ótima, de modo que a solução dos subproblemas possa compor a solução do problema inicial (PADHY, 2004). De acordo com Li *et al* (1997), a Programação Dinâmica geralmente é utilizada com o auxílio de uma Lista de Prioridade.

A estratégia de Programação Dinâmica aplicada ao PCU baseia-se na seguinte proposição: para o custo total de carregar x em N geradores ser mínimo, a carga y carregada pela unidade n deve ser tal que a carga restante de $x-y$ é carregada pelos restantes $N-1$ geradores também a custo mínimo (PADHY, 2004).

A Programação Dinâmica, contudo, possui alguns problemas, entre eles o aumento de esforço computacional exigido com o aumento do número de geradores (a chamada “maldição da dimensionalidade”) e a dificuldade de inclusão de certas restrições na modelagem do problema (PADHY, 2004; MANTAWY *et al*, 1998).

A Programação Linear é um recurso aplicável em conjunto com a programação dinâmica ou após ser decomposto pelo princípio de Dantzig-Wolfe. Também existem abordagens em conjunto com a Programação Inteira (PADHY, 2004).

A Programação Inteira pode ser utilizada com base na abordagem de *Benders*, que divide o PCU em um PDE não linear e um problema inteiro e não-linear de cometimento de unidades (SHEBLE' e FAHD, 1994; PADHY, 2004). Um trabalho parecido foi desenvolvido por Carrión e Arroyo (2006). Este, no entanto, aproximou cada restrição não-linear por uma função linear definida por partes, possibilitando o uso de Programação Linear para solucionar o restante do problema. O problema de Programação Inteira foi tratado com *Branch-and-Bound*, embora o autor não entre em detalhes sobre como elaborou sua abordagem para método.

O *Branch-and-Bound* é um método capaz de solucionar problemas de programação inteira e problemas de otimização combinatorial. Consiste na execução repetida de alguns passos que procuram a solução ótima do problema em subdivisões de sua região factível, que é um conjunto de possíveis soluções delimitado pelas restrições do modelo (PADHY, 2004; JENSEN e BARD, 2003; WINSTON, 1997). Lauer *et al* (1982) apud Padhy (2004) e Cohen e Yoshimura (1983) apud Padhy (2004) apresentaram soluções baseadas em *Branch-and-Bound* que incorporam todas as variáveis independentes do tempo e não requerem uma ordenação de prioridade de unidades. Huang *et al* (1998) apud Padhy (2004) propôs uma programação lógica de restrições junto com uma técnica de *Branch-and-Bound* para alcançar um meio eficiente e flexível de solucionar o PCU.

A Busca Tabu tem se mostrado um procedimento de otimização poderoso para problemas de combinatoriais de minimização. Possui a habilidade de evitar aprisionamento em um mínimo local, através de um sistema de memória flexível e de grandes proporções (PADHY, 2004). Rajan e Mohan (2004) desenvolveram uma metodologia híbrida de Algoritmos Evolutivos e Busca Tabu que superou em precisão e eficiência as diversas abordagens de Programação Evolucionária, Programação Dinâmica, Relaxação Lagrangeana e Busca Tabu referenciadas em seu trabalho.

O *Simulated Annealing* é um método inspirado no fenômeno físico de resfriamento de metais e divide o problema em dois subproblemas: um de otimização combinatorial e outro de otimização não linear (PADHY, 2004). O método é capaz de resolver o problema, no entanto,

demanda um extremo esforço computacional (MANTAWY *et al*, 1998). Curiosamente, Sheble' e Fahd (1994) afirmam que o *Simulated Annealing* gera soluções com rapidez, no entanto considera as soluções geradas pelo método aproximadamente ótimas.

A Lógica *Fuzzy*, ou Lógica Nebulosa, é uma teoria matemática que abrange o conceito de “incerteza” ao definir um conceito ou significado. As variáveis não possuem valores exatos, mas sim, conceitos como “baixo”, “médio”, “alto”. Este tipo de lógica apresenta vantagens em situações incertas. Para o PCU, a Lógica Nebulosa pode representar variáveis como: capacidade utilizada em cada gerador, custo incremental, custo de partida (incluindo manutenção e mão-de-obra) e custo de produção (MANTAWY *et al*, 1998). Segundo Padhy (2004) e Mantawy *et al* (1998), a Lógica *Fuzzy* oferece vantagem em situações de incerteza de demanda.

As Redes Neurais tentam modelar o comportamento de uma rede neural biológica e requerem um treinamento, no qual diversos problemas e suas respostas são introduzidos na rede de modo a torná-la capaz de resolver problemas do mesmo tipo. Já foi explorada a possibilidade de resolver um problema de otimização combinatorial, como o PCU, utilizando Redes Neurais de *Hopfield*. Esta rede, a princípio, não suportava algumas restrições, mas tem sido aprimorada por outros autores neste sentido. Dada a sua estrutura, Redes Neurais são vantajosas quando executadas em sistemas computacionais de processamento paralelo (PADHY, 2004).

Os Algoritmos Evolutivos, como o Algoritmo Genético e o Algoritmo Cultural, baseiam-se na evolução biológica dos seres vivos, gerando uma diversidade de indivíduos (possíveis soluções) e selecionando os mais aptos (as soluções mais próximas do ótimo). O processo é repetido até atingir um critério de parada. Os Algoritmos Evolutivos têm se mostrado uma técnica robusta para resolver o problema e várias soluções já foram propostas por diversos autores (PADHY, 2004). De acordo com Damousis e Bakirtzis (2004), as desvantagens da aplicação deste tipo de algoritmo ao PCU são o longo tempo de execução e o fato de que, devido à heurística utilizada pelo método, a solução ótima não é garantida. Como o próprio autor analisa, a cada execução do algoritmo, diferentes soluções são encontradas para o mesmo problema. Entretanto, neste mesmo trabalho foi aprimorada a técnica de codificação dos genes utilizada em abordagens tradicionais de Algoritmos Genéticos (transformando-a de binária para inteira) e se obteve resultados positivos.

O Algoritmo de Colônia de Formigas, também conhecido como *Ant System*, baseia-se no comportamento das formigas, que são capazes de encontrar o menor caminho até o alimento através dos rastros de ferormônios de outras formigas. O caminho mais curto é o caminho pelo qual mais formigas passam, deixando um rastro mais forte de ferormônios (PADHY, 2004). Este tipo de algoritmo, no entanto, parece ter sido menos utilizado até o momento.

A partir da literatura, pode-se observar que existem métodos que oferecem soluções precisas, porém, são mais complexos. Alternativamente, existem métodos mais simples, com maior ou menor grau de eficiência, e que apresentam soluções aproximadas. Devido às falhas e qualidades de cada método, muitos deles têm sido combinados na tentativa de criar um melhor.

A maioria dos métodos discutidos neste capítulo foi projetada para tratar integralmente o PCU, ou seja, também resolvem o problema implícito de Despacho Econômico. Para este trabalho, em particular, já se possui uma solução para o último problema, de modo que o método a ser escolhido será avaliado em função da parte restante do PCU que deve ser solucionada.

2.2 Programação Inteira

Problemas de Programação Inteira (PPI) são aqueles em que as variáveis devem assumir apenas valores numéricos inteiros. Os PPI nos quais algumas variáveis devem ser inteiras mas outras não necessariamente precisam ser são chamados Problemas de Programação Inteira Mistos (PPIM) (JENSEN e BARD, 2003).

Um PPI pode ser caracterizado como muito fácil ou muito difícil. Isto é verdade mesmo quando todas as funções são lineares e todos os dados são conhecidos. Os problemas fáceis incluem aqueles que exibem total unimodularidade e, portanto, podem ser resolvidos diretamente como Problemas de Programação Linear (PPL), sem recorrer a técnicas mais avançadas. A maioria dos modelos que apresenta esta propriedade se enquadra na definição de *Pure Network* (Aplicações Puramente para Redes). Sua estrutura especial levou ao desenvolvimento de técnicas específicas que se mostraram muito mais rápidas do que o próprio método simplex (JENSEN e BARD, 2003).

Uma segunda classe de PPI fáceis são aqueles que podem ser resolvidos com algoritmos gulosos. Encaixam-se nesta categoria o problema de espaço mínimo entre árvores, o problema

da mochila com pesos iguais e diversos problemas de sequenciamento para uma máquina. O esforço computacional necessário para solucionar cada um deles cresce em proporções lineares à medida que o problema cresce (JENSEN e BARD, 2003).

Os PPI difíceis formam uma classe própria e estão entre os mais difíceis problemas de otimização. O número de iterações necessário para encontrar a solução ótima, no pior caso, cresce exponencialmente à medida que o problema cresce. Isto implica que embora resolver um problema de menores proporções seja possível, resolver problemas maiores pode ser impraticável (JENSEN e BARD, 2003).

O PCU é um problema de otimização combinatorial (SENJYU *et al*, 2003). Na definição de Winston (1997), este é um problema de otimização que possui um número finito de soluções factíveis. Winston (1997) enquadra este tipo de problema na classificação de PPI 0-1, que Jensen e Bard (2003) definem como um caso especial de PPI difícil, no qual as variáveis de decisão devem ser iguais a zero ou um, ao invés de números inteiros arbitrários. Será discutido adiante que, para o PCU, um valor 1 indicará um gerador ligado e o valor 0, um gerador desligado.

2.2.1 Algoritmos Gulosos

Na definição de Jensen e Bard (2003), um algoritmo guloso para um problema de otimização é aquele que iterativamente constrói uma solução que a cada passo é melhorada com a tentativa de realizar o máximo ganho imediato, sem considerar os passos seguintes. Estes algoritmos possuem duas características principais. A primeira, é que a seleção a cada passo não é alterada em passos subsequentes e a segunda, é que o número de iterações é uma função polinomial do tamanho do problema.

Os algoritmos gulosos não olham à frente durante a solução do problema. Em consequência disso, geralmente incorrem em altos custos de oportunidade para fazer uma boa decisão nos estágios iniciais que, nos estágios finais, acabará por limitar as escolhas às melhores opções dentro de um conjunto de opções ruins. O algoritmo do “vizinho mais próximo” para a solução do problema do caixeiro viajante é um exemplo. Neste algoritmo, o nó mais próximo ao de partida é selecionado como o primeiro a ser visitado. No segundo passo, o nó mais próximo do anterior é selecionado, e assim por diante, até que todos os nós tenham sido sequenciadas, sendo o último ligado ao nó inicial, completando o circuito. O que ocorre na prática, é que a solução é ótima até alcançar os passos finais. A este ponto, sequenciar o

restante dos nós faz o comprimento do circuito aumentar drasticamente, pois as únicas opções disponíveis são distantes entre si. Ao final, a solução alcançada estará longe da solução ótima. Em um grafo orientado incompleto, nem mesmo uma solução poderá ser encontrada (JENSEN e BARD, 2003).

2.2.2 *Métodos Heurísticos*

Para solucionar PPI difíceis, é comum aplicar métodos que executam com rapidez e oferecem uma solução boa, mas que não é, necessariamente, a solução ótima. Se um procedimento computacional é capaz de encontrar, em alguns segundos, uma solução confiável e próxima da solução ótima, este procedimento pode ser imensamente preferível a um procedimento exato que leva horas para encontrar a solução ótima. Tais procedimentos são chamados métodos heurísticos. Por terem sido aplicados com sucesso a problemas cuja solução com algoritmos exatos não é possível, os métodos heurísticos foram bastante explorados e desenvolvidos (JENSEN e BARD, 2003).

Muitos métodos heurísticos que geram soluções boas, mas não necessariamente ótimas, utilizam critérios gulosos no momento em que selecionam o valor das variáveis de decisão. Considera-se, por exemplo, o problema da mochila, que representa o dilema de selecionar um conjunto de itens que propicia o máximo benefício sem exceder a limitação de peso da mochila. Uma heurística para encontrar a solução utiliza a taxa de proporção entre peso e benefício para cada um dos itens. O item com maior taxa, do ponto de vista guloso, consiste na melhor escolha e é escolhido primeiro. Os itens subsequentes são escolhidos em ordem decrescente da taxa. Se o item não fizer com que o peso da mochila se exceda, ele é selecionado, em caso contrário, é excluído. Este método nem sempre obtém a solução ótima, no entanto é facilmente aplicável e obtém uma boa solução. A programação dinâmica é capaz de garantir uma solução ótima para este problema, mas a custo de um esforço computacional substancialmente maior (JENSEN e BARD, 2003).

Embora seja bastante simples, o problema da mochila, que é um problema clássico de Pesquisa Operacional, apresenta algumas semelhanças com o PCU, sobretudo em relação à parte de Programação Inteira do problema, que é o foco deste trabalho. A escolha do item a ser colocado na mochila é, de certa forma, equivalente à escolha da unidade a ser cometida no PCU. Além disso, em ambos os problemas, o elemento de escolha em questão possui uma relação entre custo e benefício.

2.2.3 Enumeração Exaustiva

Enumerar todas as possíveis soluções de um PPI, o que é chamado de enumeração exaustiva, não é, em geral, uma abordagem praticável para problemas que não possuam um número bastante reduzido de variáveis. O número de soluções possíveis é igual a 2^n - onde n é o número de variáveis de decisão – e cada uma delas precisa ser testada para satisfazer as restrições (JENSEN e BARD, 2003).

A enumeração exaustiva é descrita aqui com o único intuito de apresentar os conceitos que serão necessários para a compreensão do método *Branch-and-Bound*.

Seguindo Jensen e Bard (2003), será tomado como exemplo o problema da mochila, que é um PPI 0-1. O problema possui a seguinte formulação genérica:

$$\text{Maximize } z = \sum_{j=1}^n c_j x_j \quad (3)$$

$$\text{Sujeito a: } \sum_{j=1}^n a_j x_j \leq b \quad (4)$$

$$x_j = 0 \text{ ou } 1, j = 1, \dots, n$$

O caso específico deste problema é definido:

$$\text{Maximize } z = 4x_1 + 9x_2 + 6x_3 \quad (5)$$

$$\text{Sujeito a: } 5x_1 + 8x_2 + 6x_3 \leq 12 \quad (6)$$

$$x_j = 0 \text{ ou } 1, j = 1, 2, 3$$

As Equações 3 e 5 definem as funções objetivo, enquanto as Equações 4 e 6 são as restrições do problema.

A despeito de sua simplicidade, a enumeração exaustiva requer um meio formal para gerar todas as possíveis soluções. Isto é feito como uma árvore de busca, ou árvore de enumeração. A Figura 1 mostra uma árvore com as três variáveis binárias do problema. Os círculos são chamados nós e as linhas, galhos. Conforme se desce a partir do topo, decisões são feitas como indicadas pelos números nos galhos. Um número negativo, $-j$, implica que a variável x_j foi igualada a 0 e, conseqüentemente, um número positivo, $+j$, implica que x_j foi igualado a 1 (JENSEN e BARD, 2003).

Os nós são numerados conforme as variáveis são determinadas como 0 ou 1, em uma seqüência que pode variar de acordo com o esquema de enumeração. Cada nó k herda todas as restrições definidas pelos galhos no caminho que o liga à raiz, caminho este denominado P_k . Por exemplo, o nó 1 possui a decisão +1 que é indicada pelo galho que conecta o nó 0 ao 1, o que significa que a variável x_1 foi igualada a 1. O nó 5 possui a decisão -2 que é indicada pelo galho conectando 1 e 5, o que implica em uma restrição adicional $x_2 = 0$. Cada nó folha, situado no último nível da árvore, indica que todas as variáveis tiveram seus valores definidos. Cada um dos oito nós folha na Figura 1 representa uma solução completa que pode ser identificada traçando-se o caminho a partir da folha até a raiz e coletando as decisões associadas com os galhos ao longo do caminho. Assim, o nó 6, por exemplo, representa a solução $x = (1,0,1)$ (JENSEN e BARD, 2003).

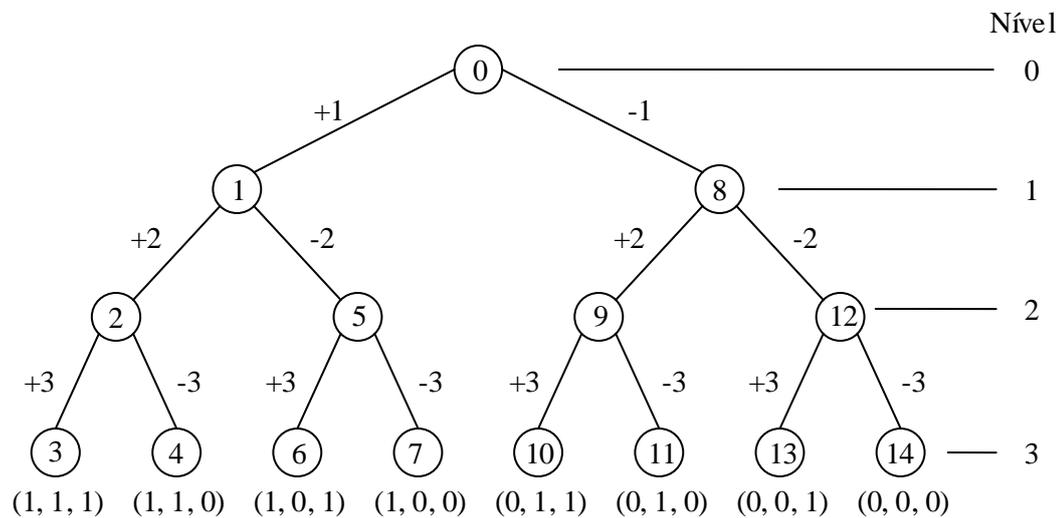


Figura 1: Árvore de busca para a enumeração exaustiva

Cada nó da árvore reside em um nível particular que indica o número de decisões tomadas para alcançar este mesmo nó. O nó 0 está no nível 0, indicando que nenhuma decisão foi tomada, ao passo que o nó 14 está no nível 3 e, portanto, três decisões foram tomadas. Uma árvore de busca completa possui $2^{n+1} - 1$ nós (JENSEN e BARD, 2003).

Para cada nó k não seja folha, existe um caminho parcial P_k partindo do nó até a raiz, que corresponde a uma atribuição de valores binários a um subconjunto de variáveis, que é chamado de solução parcial (JENSEN e BARD, 2003). As variáveis de separação são aquelas cujos dois valores possíveis são atribuídos no próximo nível. Com isto, o conjunto de solução do nó atual é dividido em dois subconjuntos mutuamente exclusivos. A ação de escolher uma

variável de separação e mover ao próximo nível é chamada de *branch* (JENSEN e BARD, 2003).

Segundo Jensen e Bard (2003), no processo de enumeração, a opção mais comum é executar uma busca em profundidade. Isto significa que primeiro se cria um caminho direto da raiz até um nó folha qualquer e então se retorna explorando outros caminhos divergindo deste caminho. Este retorno pelo caminho anteriormente cursado é referido como *backtrack* e acontece sempre que um nó folha é atingido na busca. O processo continua desta maneira, guiado pelas seguintes regras, definidas por Jensen e Bard (2003):

- Se ainda restar uma variável no nó k , escolha uma variável de separação e execute o *branch* até o próximo nível criando o nó $k+1$.
- Se nenhuma variável restar, calcule o valor da solução e execute o *backtrack* até encontrar um nível cujo algum galho não tenha sido explorado e execute o *branch*.

Desta forma, pode-se constatar que a árvore da Figura 1 representa uma busca em profundidade, para o problema de exemplo, na qual o *branch* foi executado na ordem da esquerda para a direita (JENSEN e BARD, 2003).

2.2.3.1 Algoritmo de Enumeração Exaustiva

Segundo Jensen e Bard (2003), para implementar o processo de *branch*, é necessária uma estrutura de dados que oferece o estado da árvore a qualquer ponto. O vetor P_k é usado para este propósito. Para o nó k no nível l da árvore, P_k possui as seguintes características:

- O comprimento do vetor é l e é definido por $P_k = (j_1, j_2, \dots, j_l)$.
- A magnitude absoluta de j_i é a variável de separação no nível i .
- O sinal de j_i indica o valor da variável de separação no caminho atual. Um sinal negativo indica que o valor da variável é 0 e um sinal positivo (ou nenhum sinal) indica que o valor é 1.
- O componente j_i pode ser sublinhado ou não. Se for sublinhado, o nó alternativo no nível i já foi explorado. Se não for sublinhado, o nó ainda deve ser explorado.

- As variáveis não mencionadas em P_k são variáveis livres, pois neste vetor constam apenas variáveis de decisão.

As definições de sublinhado e sinal para j , implicam que:

- \underline{j} ocorre quando $x_j = 1$ e $x_j = 0$ ainda não foi considerado no nível i
- \underline{j} ocorre quando $x_j = 1$ e $x_j = 0$ já foi considerado no nível i
- $-j$ ocorre quando $x_j = 0$ e $x_j = 1$ ainda não foi considerado no nível i
- $-j$ ocorre quando $x_j = 0$ e $x_j = 1$ já foi considerado no nível i

Para enumerar todas as soluções, costuma-se escolher as variáveis sequencialmente e definir uma ordem para qual lado (esquerdo ou direito) o *branch* sempre será executado primeiro. É definido um algoritmo para a enumeração exaustiva no qual os nós são verificados da esquerda para a direita e o galho com o valor 1 atribuído à variável é escolhido primeiro. A definição de uma ordem para a busca é, no entanto, indiferente, pois tanto a busca em profundidade quanto o próprio algoritmo de enumeração exaustiva são capazes de executar em ordem arbitrária, pois cada nó da árvore é acessível através de um caminho único e todos os nós visitados são lembrados. Caso haja uma hipótese de que a solução ótima ocorra com um valor de x_j conhecido, o algoritmo pode ser modificado de forma que prefira sempre seguir o caminho associado com tal valor no momento do *branch* para a variável, o que na árvore, significa escolher o caminho da esquerda (igual a 1) ou da direita (igual a 0). Nomeia-se Z_B o melhor valor da função objetivo encontrado até o momento, x_b um vetor de dimensão igual ao número de variáveis separáveis e que contém o valor destas para compor a solução Z_B , k o índice do nó, l o índice do nível, j_i o índice atual disponível no nível l (JENSEN e BARD, 2003).

A execução do algoritmo para o problema de exemplo gera exatamente a árvore da Figura 1. Neste método, muitas soluções são enumeradas, mas apenas um pequeno número delas é factível ou, em outras palavras, respeita as restrições do problema (JENSEN e BARD, 2003).

2.2.4 *Branch-and-Bound*

O grupo de procedimentos do *Branch-and-Bound* (BB) parte da idéia de enumeração, mas a faz de forma mais inteligente, de modo que nem todas as combinações precisam ser verificadas (JENSEN e BARD, 2003).

Embora Winston (1997) também discorra sobre o método, sua abordagem é mais simples e introdutória, enquanto a análise de Jensen e Bard (2003) é mais formal e aprofundada. Portanto, todas as idéias contidas na sessão de BB, que consiste no item 2.2.4 deste trabalho (inclusive os sub-itens), pertencem a Jensen e Bard (2003).

O BB possui duas qualidades interessantes. Uma delas é ser aplicável tanto para PPI quanto para PPIM. Outra é a característica de encontrar uma sucessão de soluções inteiras factíveis e, portanto, se o processamento da solução precisar ser interrompido por restrição de tempo, uma solução aproximada será obtida. Esta característica não é compartilhada por métodos tal como o *Cutting Plane* – que também é aplicável a PI e será discutido adiante – mas é particularmente útil quando se deseja resolver um problema com muitas combinações, pois permite que o algoritmo execute por um período viável e forneça uma solução aceitável.

Os métodos de BB são geralmente personalizados para explorar a estrutura característica de cada problema, possibilitando que o problema seja tratado com maior eficiência e menor uso de memória, em aplicações computacionais. Desta forma, não existe apenas um método, mas uma coleção de métodos que compartilham várias características comuns.

Tanto para os PPI quanto para os PPIM, os procedimentos mais eficientes de BB são baseados no uso do método simplex e iniciam solucionando o problema inicial com PL.

O PPL é obtido substituindo-se a restrição de x_j ser igual a 0 ou 1 por uma restrição dita relaxada, na qual x_j deixa de ser uma variável inteira e pode assumir qualquer valor no intervalo fechado de 0 a 1. O problema resultante é chamado de problema relaxado, o qual se espera que seja mais fácil de solucionar do que o problema original. No caso de um PPIM, o problema relaxado é resolvido por PL.

Enquanto uma solução inteira não é alcançada na árvore de busca, uma das variáveis que assumiu valor fracionário, denotada por x_s , é selecionada e dois descendentes do problema original são criados: um no qual $x_s = 0$ e outro no qual $x_s = 1$. Como dito anteriormente, esta operação é chamada *branch*. Como x_s é uma variável 0-1 e os problemas descendentes são

exatamente iguais ao original, exceto pela atribuição de valores específicos a x_s , a solução para um dos dois descendentes deve, obrigatoriamente, ser solução para o problema original. Desta forma, o problema inicial é substituído por dois problemas que precisam ser resolvidos. A expectativa é de que estes sejam mais facilmente resolvidos do que o problema inicial, pois cada um é mais restrito e possui menos variáveis às quais precisam ser atribuídos valores inteiros.

O processo é repetido, selecionando um dos problemas de PI restantes e resolvendo-os da mesma forma que o original. Cada problema novo irá gerar mais dois problemas, a não ser que seja impossível de ser resolvido. Diversas iterações eventualmente produzem uma solução inteira (caso exista uma) para o problema de PI atual e que se tornará candidata à solução ótima para o problema original. Guardar a melhor destas candidatas (tanto o valor da função objetivo quanto o valor assumido pelas variáveis) possibilita que se eliminem descendentes do problema original e, com isso, várias gerações subsequentes, que não ofereceriam vantagem alguma, caso exploradas, pois não seriam capazes de gerar uma solução que superasse a melhor encontrada até o momento. A melhor candidata atual é chamada de incumbida e o processo de eliminar descendentes de um problema é chamado poda.

Quando uma busca em profundidade é utilizada, o *branch* é representado em uma árvore de busca binária na qual cada nó tem um antecessor e dois sucessores. Um nó que não foi podado e não é folha é chamado de nó vivo. Efetuar o *branch* significa escolher um nó vivo para considerar a próxima operação de poda. Outras estratégias de busca são possíveis, como a busca em largura, na qual mais de uma variável é escolhida em um nó para que se faça o *branch*.

2.2.4.1 Idéias Gerais

Para formalizar os conceitos de BB, denomina-se Z_B o valor da função objetivo da solução incumbida e Z_k o valor da função objetivo do PPL relaxado correspondente a um nó k . Assim, quando um problema de maximização de PI é resolvido como PL, uma das quatro alternativas ocorre:

1. O PPL não possui solução factível e, por conseqüência, tampouco o PPI.

2. O PPL possui uma solução ótima $Z_k \leq Z_B$ e, desta forma, o atual PPI possui $Z_k \leq Z_B$ e não provê melhorias sobre a incumbida.
3. A solução ótima para o PPL é inteira e factível e retorna $Z_k > Z_B$ e, portanto, a solução incumbida Z_B pode ser substituída por Z_k .
4. Nenhuma das anteriores ocorre. Por exemplo, a solução ótima satisfaz o PPL com $Z_k > Z_B$, mas não possui valores inteiros para as variáveis.

Em cada um dos primeiros três casos, o PPI no nó k precisa apenas ser resolvido por PL, ou seja, o PPI é podado. Um problema que é podado como resultado do caso 3 gera uma informação muito útil, pois permite que a solução incumbida seja atualizada. Se o problema não é podado, enquadrando-se no caso 4, deve-se continuar explorando e efetuando o *branch*.

O problema relaxado não precisa, necessariamente, ser um PPL. Uma segunda opção é gerar um PPI que é mais fácil de ser resolvido do que o original. O problema do caixeiro viajante, por exemplo, é relaxado para o problema de atribuição e o problema de espaçamento mínimo entre árvores.

2.2.4.2 Sub-rotinas do BB

Jensen e Bard (2003) apresentam os passos básicos necessários para solucionar PPI 0-1 usando BB. Embora estas sub-rotinas sejam aplicáveis para a maioria dos problemas, vários procedimentos computacionais dependem do problema específico. Embora um objetivo de maximização seja assumido, se o objetivo for a minimização, o problema pode ser resolvido com o mesmo algoritmo após algumas modificações, ou transformando diretamente o problema para maximização.

As cinco sub-rotinas abaixo são utilizadas para guiar a busca e extrair informações úteis para diminuir o tamanho da árvore:

1. Restringir: este procedimento examina o problema relaxado em um nó particular e tenta estabelecer restrições na solução ótima. Possui duas saídas possíveis:
 - Uma indicação de que não há solução factível no conjunto de soluções inteiras representadas pelo nó.

- Um valor Z_{UB} que é um limite superior na a função objetivo para todas as soluções no nó e em seus nós descendentes.
2. Aproximar: este procedimento tenta encontrar uma solução inteira factível a partir da solução do problema relaxado. Caso uma solução seja encontrada, esta possuirá o valor Z_{LB} que é um limite inferior para a solução ótima do problema de maximização.
 3. Ajuste de variável: este procedimento executa um teste lógico na solução encontrada em um nó. A intenção é determinar se alguma das variáveis livres binárias é, necessariamente, 0 ou 1 em uma solução inteira ótima no nó atual ou em seus descendentes, ou ainda, se as mesmas precisam receber o valor 0 ou 1 para garantir a que o problema continue factível conforme o processamento avança.
 4. *Branch*: procedimento cujo objetivo é selecionar uma das variáveis livres para separação. Também decide a primeira direção (0 ou 1) a ser explorada.
 5. *Backtrack*: este procedimento permite que a busca retroceda, determinando qual nó deverá ser explorado em seguida quando o nó atual é eliminado. É projetado para enumerar sistematicamente todos os nós vivos da árvore de BB e assegurar que a solução ótima do problema original não seja desconsiderada.

A implementação destas sub-rotinas em um algoritmo completo possui muitos detalhes, muitos dos quais são característicos do problema. Em seguida, as sub-rotinas serão discutidas mais detalhadamente.

2.2.4.3 Eliminando nós por limites

Uma solução factível X_B com valor da função objetivo Z_B encontrada em um nó na árvore de busca é chamada solução incumbida, pois Z_B denota a melhor solução para o PPI encontrada até o momento. Em outras palavras, Z_B é um limite inferior para um problema de maximização.

Um nó que não seja folha representa um conjunto de soluções que poderiam ser enumeradas e identificadas por meio do teste de todas as combinações de valores possíveis para as variáveis de decisão. As variáveis para as quais o *branch* já foi efetuado, representadas por nós em níveis anteriores, possuem valores fixados, enquanto as demais, representadas por nós em níveis seguintes, permanecem indefinidas e, portanto, constituem o sub-problema a ser

resolvido no nó em questão. Quando se resolve este problema de forma relaxada (ou se resolve o problema de alguma outra forma), é obtido um limite superior Z_{UB} para os valores da função objetivo para todo este conjunto. Se o nó em questão for o de nível zero, tem-se a solução relaxada para o problema inicial, cuja solução pode ser utilizada para definir um limite superior inicial.

O teste de limites é definido por:

$$Z_{UB} \leq Z_B \quad (7)$$

Se Z_{UB} , na Equação 7, que indica a melhor solução possível para o nó atual e seus descendentes, for menor que Z_B , que é a melhor solução para o PPI já encontrada, então nenhuma das soluções no conjunto em questão é capaz de superar Z_B . Assim, o nó e seus descendentes podem ser eliminados e o procedimento pode seguir para o *backtrack* e encontrar o próximo nó. Caso contrário, o nó não pode ser eliminado e deve-se seguir para o *branch*.

A idéia principal do teste de limites é que um limite superior em um problema de maximização pode ser obtido pelo relaxamento de uma ou mais restrições (por exemplo, as restrições que definem que as variáveis devem ser 0 ou 1). Desta forma, é obtida uma região factível maior do que a do problema anterior, pois as soluções fracionárias são consideradas, além das inteiras. Logo, em qualquer nó na árvore e busca o valor ótimo para o problema relaxado é maior ou igual o valor ótimo do PPI neste mesmo nó. Se o problema relaxado não possui uma solução boa, a solução do PPI será ainda pior.

A eficiência do BB depende de alguns critérios:

1. A qualidade do limite superior (para um problema de maximização, quanto menor o limite, melhor);
2. A dificuldade computacional em se obter o limite superior (quanto menor, melhor);
3. A qualidade do limite inferior (para um problema de maximização, quanto maior, melhor);
4. A dificuldade computacional em se obter o limite inferior (quanto menor, melhor);

5. A sensibilidade¹ do valor da função objetivo às variáveis de decisão em x (quanto mais sensível melhor).

Os primeiros dois critérios são, em geral, conflitantes. A qualidade do limite superior é dependente do esforço computacional em se obtê-la. Para muitos problemas, um limite superior ruim pode ser obtido facilmente, enquanto um bom limite só pode ser encontrado com muito esforço. Da mesma forma, os critérios 3 e 4 são contraditórios. Limites inferiores são geralmente obtidos com heurísticas que constroem regiões factíveis a partir de soluções fracionárias encontradas durante o processo enumerativo. Arredondar uma solução fracionária pode levar a um ponto factível, entretanto é mais provável que não esteja próximo da solução ótima. A análise deve balancear a qualidade do limite e o esforço para consegui-lo.

A sensibilidade do valor da função objetivo a mudanças no vetor de solução é característica do problema. O teste de restrição será mais efetivo para problemas com o valor ótimo de Z_{IP} significativamente maior do que valores para outras decisões. Se ocorrerem muitas soluções próximas do ótimo, o teste de limites não será tão efetivo.

O tamanho da árvore de busca também está diretamente relacionado com a efetividade dos testes de eliminação. Com 20 variáveis binárias, a árvore completa tem mais de 2×10^6 nós. Se o teste de limite inferior for 90% efetivo, 2×10^5 nós ainda precisam ser calculados. Isto mostra porque o critério de esforço computacional é tão importante, pois os limites superiores e inferiores são calculados para cada nó.

Procedimentos de enumeração exaustiva têm sido combinados com decomposições avançadas ou técnicas de corte de planos para solucionar muitos problemas. Contudo, a convergência do algoritmo não é certa para os casos com centenas de variáveis ou mais. Para acelerar a convergência, é necessário um comprometimento da qualidade da solução em prol da finitude do algoritmo.

Um modo de alcançar isto é reforçando o teste de eliminação utilizando um parâmetro ρ que especifica uma porcentagem permitida de desvio da solução ótima. Para tal, o teste $Z_{UB} \leq Z_B$ deve ser substituído pelo teste:

¹ A sensibilidade de uma função é uma relação de como os parâmetros do problema afetam a solução ótima. Quanto maior a sensibilidade, maiores são as mudanças percebidas na função objetivo ao se alterar os valores das variáveis (WINSTON, 1997).

$$Z_{UB} \leq Z_B + \rho \left(\frac{Z_B}{100} \right) \quad (8)$$

Assim, um nó k cujo Z_{UB} satisfaz a desigualdade da Equação 8 pode ser eliminado. Embora este teste possa impedir que a solução ótima seja encontrada, é certo que o valor da solução final será sempre $\rho\%$ do valor ótimo. Em muitas situações, este erro pode ser assumido em virtude do tempo economizado de processamento. Muitos algoritmos comerciais utilizam o valor padrão para ρ em torno de 0,5%.

2.2.4.4 Eliminando nós por infactibilidade

O problema relaxado, quando é infactível, indica que o PPI original também é infactível. Isto é verdade porque o problema relaxado possui uma região factível maior, que engloba a região factível do PPI original. Este, portanto, consiste no segundo teste utilizado para eliminar nós na árvore de busca.

2.2.4.5 Determinando uma solução factível

A importância dada ao teste de limites, no qual Z_{UB} é comparado com Z_B na tentativa de eliminar o nó atual, não pode ser exagerada. A efetividade deste teste cresce de acordo com a qualidade de X_B ou, em outras palavras, conforme Z_B se aproxima da solução ótima do PPI. Assim, um componente importante para o BB é uma heurística para se obter boas soluções factíveis. Em geral, isto é feito utilizando a solução do problema relaxado como ponto de partida, mas pode envolver técnicas mais inteligentes como busca tabu, *simulated annealing* e GRASP. Tais heurísticas são chamadas de procedimentos aproximativos.

A enumeração implícita é capaz de funcionar sem um procedimento aproximativo, já que as soluções factíveis são sempre encontradas nos nós folhas. No entanto, os problemas podem ser resolvidos mais rapidamente caso boas soluções factíveis sejam descobertas logo no início, pois os nós são eliminados mais cedo, gerando menos descendentes e, portanto, menos nós a serem explorados. Portanto é recomendado que um procedimento aproximativo seja utilizado pelo menos no nó raiz. De fato, para problemas com mais variáveis, caso os nós não sejam eliminados antes de se encontrarem os nós folhas, é bem provável que a convergência não será atingida em um prazo razoável.

Para um PPI 0-1, o meio mais fácil de se obter uma solução inteira a partir de um resultado fracionário, proveniente da solução do problema relaxado, consiste em arredondar os valores que não sejam iguais a 0 ou 1, sempre respeitando as restrições. Isto levará a uma solução factível que pode, então, definir um limite inferior para os nós.

Quando obtém sucesso, um procedimento aproximativo retorna dois resultados: uma solução factível X_F e o valor correspondente de Z_F . Se a solução encontrada for melhor do que a incumbida Z_B , então a última pode ser atualizada, criando-se um parâmetro melhor para o teste do limites.

2.2.4.6 Ajustando variáveis

Alguns algoritmos incorporam uma variedade de testes lógicos para determinar se alguma das variáveis livres deve ser igualada a 0 ou 1 para assegurar uma solução factível em um nó em particular. Para ilustrar uma situação, considera-se a restrição abaixo:

$$5x_1 + 8x_2 + 6x_3 \leq 12 \quad (9)$$

Analisando a restrição da Equação 9 pode-se chegar a seguinte conclusão: se x_2 for 1, necessariamente x_3 deve ser 0 a fim de garantir a factibilidade. Este teste é opcional, no entanto permite que alguns nós sejam eliminados facilmente da árvore de busca.

Quando o problema relaxado é um PPL, informações da solução podem ser utilizadas para ajustar variáveis. No entanto, não é de interesse o estudo destas técnicas, já que o PCU considerado neste trabalho, como mencionado anteriormente, não é linear.

2.2.4.7 Encontrando uma variável separável

Sempre que um nó é eliminado por limites ou infactibilidade, uma nova variável de separação deve ser escolhida para o *branch*. A direção da exploração (0 ou 1) deve ser especificada. A escolha feita define a seqüência das soluções obtidas pela enumeração e pode desempenhar um importante papel na eficiência do algoritmo. O único requisito para a escolha da variável é que ele deve pertencer ao conjunto de variáveis livres no nó k .

Existem diversas estratégias para tal, incluindo as duas que seguem:

- Selecionar a variável e a direção tais que o nó criado será, mais provavelmente, aquele que contém a solução ótima.

- Selecionar a variável e a direção que possuem mais chances de reduzir a infactibilidade da solução do problema relaxado.

A motivação da primeira regra é que uma vez encontrada a solução ótima, mesmo que seja impossível provar isso imediatamente, terá sido encontrado o maior valor possível para Z_B , o que é extremamente importante para os testes de limite seguintes. Quando os nós são eliminados em níveis baixos (próximos à raiz), mais soluções são eliminadas. Mais precisamente, um nó eliminado a um nível l elimina 2^{n-l} soluções. Além disso, se o processamento precisar ser finalizado antes da convergência, a incumbida é ótima, mesmo que isso não tenha sido verificado.

Para o exemplo do problema da mochila, a primeira regra implica que os nós serão selecionados em uma ordem tal que prioriza a maior taxa de custo/benefício. Consequentemente, a variável x_2 seria escolhida primeiro e igualada a 1.

De forma similar, o PCU pode se valer da lista de prioridades para geradores como estratégia para selecionar variáveis para o *branch*. Para que isto possa ser feito, será necessário um método formal para calcular os índices de prioridade de cada gerador do sistema. Os índices definidos devem, ainda, ser relacionados às possíveis situações do problema, pois, como dito anteriormente, há diversos tipos de geradores e cada um deles é mais apropriado a uma situação específica.

É comum a segunda regra produzir uma solução factível mais rápido do que a primeira. A partir da solução do problema relaxado, a regra prefere explorar as variáveis que surgiram diferente de 0 e 1 e que, portanto, comprometem a factibilidade do problema.

Como não existe uma abordagem teoricamente melhor, a melhor delas depende do problema e pode ser justificada apenas com base em experimentos empíricos, nos quais as duas possibilidades são comparadas. Muitos algoritmos comerciais permitem que o usuário especifique as regras de seleção.

2.2.4.8 Algoritmo genérico de BB para PPI 0-1

Jensen e Bard (2003) definem um algoritmo para BB partindo da lógica de enumeração exaustiva e incorporando as técnicas de eliminação de nós, ajuste de variáveis e heurísticas para encontrar soluções factíveis. No algoritmo que será apresentado adiante k é o contador para identificar os nós, l é o nível na árvore, n é o número de variáveis inteiras, P_k é um vetor

de dimensão l que identifica o caminho da raiz até o nó k . O estado atual das variáveis em cada nível específico é indicado por S_k^+ (igual a 1), S_k^- (igual a 0) e S_k^0 (variável ainda não definida). Os vetores X_k e X_B contêm o valor das variáveis para solução factível no nó k e para a incumbida, respectivamente.

O algoritmo é descrito a seguir em pseudocódigo.

Inicializar:

Crie nó 0 no nível 0. Atribua $k = 0$, $l = 0$ e $P_0 =$ vazio.

Execute o procedimento aproximativo para encontrar uma solução factível.

Se uma solução factível for encontrada:

Atribua a X_B o vetor de variáveis inteiras com o valor da função objetivo Z_B .

Caso contrário:

Atribua $Z_B = -M$ (um número grande, em termos absolutos).

Atualizar:

Se $l = n$, uma solução inteira foi gerada.

Atribua X_k ao vetor solução P_k .

Se X_k é factível:

Compare o valor da função objetivo Z_K com Z_B .

Se $Z_K > Z_B$:

Faça $X_B = X_K$ e $Z_B = Z_K$.

Independente de X_K ser factível ou não, vá para Ajuste de variáveis.

Se $l < n$, continue com Ajuste de variáveis.

Ajuste de variáveis:

Utilize testes lógicos para determinar se uma variável livre x_j deve ser igualada a 0 ou 1, onde $j \in S_k^0$.

Se X_S deve ser igualada a 1:

Faça $l = l + 1$, $k = k + 1$.

Adicione $+s$ ao caminho atual para obter P_k .

Crie o nó k com a decisão $+s$.

Se X_s deve ser igualada a 0:

Faça $l = l + 1$, $k = k + 1$.

Adicione $-s$ ao caminho atual para obter P_k .

Crie o nó k com a decisão $-s$.

Se alguma mudança ocorreu, repita este passo, senão, vá para *Restringir*.

Restringir:

Resolva o problema relaxado no nó k .

Se não obtiver nenhuma solução factível, elimine o nó e vá para *Backtrack*.

Se o procedimento retornar Z_{UB} , obtenha a solução inteira e compare-a com a incumbida.

Se $Z_{UB} \leq Z_B$, elimine o nó e vá para *Backtrack*.

Caso contrário, vá para *Aproximar*.

Aproximar:

Tente encontrar uma solução factível no conjunto de soluções para o nó k . Se uma solução factível é encontrada, atribua ao vetor de variáveis X_k e ao valor da função objetivo z_k os devidos valores.

Se $Z_k > Z_B$, faça $X_B = X_k$ e $Z_B = Z_k$.

Se $Z_{UB} = Z_B$, elimine o nó e vá para *Backtrack*.

Caso contrário, vá para *Branch*.

Branch:

Escolha uma variável separável x_s , tal que $x \in S_{k0}$, e uma direção de exploração.

Faça $k = k + 1$ e $l = l + 1$.

Crie o nó k no nível l da árvore.

Se X_s deve ser igualado a 1, adicione $+s$ ao caminho atual para obter P_k .

Se X_s deve ser igualado a 0, adicione $-s$ ao caminho atual para obter P_k .

Vá para Atualizar.

Backtrack:

Encontre no vetor P_k um elemento o mais longe possível da direita que não esteja sublinhado.

Se todos os elementos estão sublinhados, finalize - X_B é a solução ótima.

Caso contrário, atribua o elemento a j_i . Faça o *Backtrack*:

Faça nível $l = i$.

Exclua todos os elementos de P_k à direita de j_i .

Faça $j_i = -j_i$ (troque o sinal de j_i).

Sublinhe j_i para obter P_{k+1} .

Faça $k = k + 1$ e crie o nó k .

Vá para Atualizar.

Uma peculiaridade do BB é que a solução ótima, ou soluções quase ótimas, são encontradas em estágios enumerativos iniciais, principalmente quando boas heurísticas são utilizadas, mas não são confirmadas até se atingirem estágios avançados da busca.

Esta é uma implementação simples de BB, mas permite que outros detalhes e melhorias sejam adicionados. Um bom projeto exige um estágio de pré-processamento para eliminar restrições redundantes e ajustar variáveis, uma estrutura modular para acomodar várias relaxações e procedimentos de aproximação, além de regras de *branch* e *backtrack* que dependem do problema com o qual se está lidando.

2.2.5 *Cutting Plane*

O método *Cutting Plane* (CP) consiste em uma segunda abordagem para se resolver PPI, também baseada na idéia de relaxar o requisito de integralidade e resolver o problema resultante (por exemplo, um PPL). Porém, ao invés de impor restrições às variáveis binárias iterativamente, como é feito no BB, é gerada uma série de novas restrições que são adicionadas ao problema original, o qual é resolvido novamente. Estas restrições, chamadas “cortes”, não se referem a apenas uma variável, mas sim ao conjunto de soluções factíveis como um todo (JENSEN e BARD, 2003).

Jensen e Bard (2003) afirmam que se a geração de cortes no espaço de soluções for adequada, de modo a não eliminar a solução ótima, eventualmente, o problema relaxado terá uma solução inteira e ótima. Por isso, a formulação do método requer um estudo da geometria característica do problema no espaço de soluções.

3 METODOLOGIA

Para que o trabalho pudesse ser realizado, foi necessário definir as atividades a serem desenvolvidas, das quais algumas puderam ser executadas paralelamente.

Através da revisão bibliográfica foi possível modelar o problema e compreender todos os principais métodos existentes. Decidiu-se que método a ser desenvolvido seria híbrido, de modo que as vantagens oferecidas por diferentes métodos pudessem ser aproveitadas. Foi selecionado um método principal baseado em BB e outros métodos para auxiliarem no cumprimento de suas etapas, como o *Unit Decommitment* e a Lista de Prioridades.

Após o desenvolvimento do método, o mesmo foi descrito através de um algoritmo utilizando pseudocódigo. Finalmente, a linguagem JAVA foi utilizada para a implementação computacional do algoritmo.

Para a validação do método e do algoritmo, foram executados problemas para os quais a solução já era conhecida, através do método de enumeração exaustiva. A validação é necessária para que o algoritmo torne-se capaz de resolver quaisquer casos do PCU, desde que seja compatível com a modelagem matemática do problema. Determinou-se que uma calibração dos parâmetros criados para o funcionamento do método era necessária a fim otimizar o desempenho do algoritmo. Ao final, foi feita uma análise de resultados e discussão.

3.1 Escolha do método

Devido à intenção de utilizar o método de Rodrigues (2007) para o PDE, é imprescindível que o primeiro passo para a solução do problema de PCU seja a aplicação da abordagem de *Benders*, que divide o problema inicial em um PDE não-linear e um problema inteiro e não-linear de cometimento de unidades. Este PCU resultante é o problema que deverá ser solucionado e consiste num problema de otimização combinatorial que pode ser traduzido para um PPI 0-1, de acordo com Jensen e Bard (2003). Embora a caracterização deste problema como PPI não impeça a aplicação dos diversos métodos descritos na revisão bibliográfica, preferiu-se um método específico de programação inteira, sobretudo devido às vantagens oferecidas pelo método *Branch-and-Bound* para o problema específico.

Dado o interesse pela programação inteira, foram avaliados os métodos *Branch-and-Bound* e *Cutting Plane* e constatou-se que o último possui algumas complicações do ponto de vista deste trabalho.

Uma das dificuldades do CP é que os planos de corte gerados delimitam uma série de variáveis ao mesmo tempo, o que dificulta a compreensão do estado atual do algoritmo e o caminho percorrido até a solução ótima durante a resolução. A busca no BB, por sua vez, é mais facilmente direcionável, pois apenas uma variável por vez é selecionada. Isto facilita a implementação de heurísticas capazes de tirar proveito da natureza específica do problema. Ao passo que o BB tende gradualmente para a solução ótima, o CP não oferece solução alguma até que a aplicação do método tenha chegado ao fim, o que significa que nenhuma solução aproximada pode ser obtida se houver a necessidade de interrupção do método prematuramente.

Além disso, não foi encontrado na literatura, recomendações do método CP em oposição ao BB. Por tais motivos, o método BB foi preferido ao CP para este trabalho.

O BB é capaz de solucionar problemas de otimização combinatorial de forma ótima, de forma que o método desenvolvido terá uma precisão tão boa quanto aquela oferecida pelo método proposto por Rodrigues (2007). No entanto, caso o esforço computacional requerido pelo método para encontrar a solução ótima seja insatisfatória, uma margem de tolerância de erro pode ser assumida, como sugerido por Jensen e Bard (2003), de modo que mais nós da árvore de busca possam ser eliminados. Caso este recurso seja necessário, a margem de erro poderá ser definida pelo usuário.

3.2 Modelagem do problema

O PCU considerado neste trabalho contou com várias simplificações do modelo genérico apresentado, que é o modelo mais completo. Simplificando a Equação 10, obtém-se:

$$\text{Minimize } CO = \sum_{i=1}^N CC_i(P_i) \quad (10)$$

A Equação 10 é a função objetivo, na qual se deseja minimizar o custo operacional, composto apenas pelo Custo de Combustível de cada uma das unidades geradoras em função da utilização de suas respectivas potências. Uma grande mudança do modelo genérico para este é

que não há mais uma somatória de intervalos de tempo para programar o fornecimento de energia durante um período. Ao invés de um perfil de demanda horária, tem-se um único valor de demanda, atribuindo um caráter estático ao problema, que pode ser entendido como o PCU em uma única iteração de tempo.

Desta forma, a definição do Custo de Combustível também muda:

$$CC_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (11)$$

O custo de combustível para a unidade i representado pela Equação 11 é uma equação do segundo grau cujos três coeficientes devem ser fornecidos como dados de entrada.

Para fins deste trabalho, a única restrição do problema, além dos limites de potência mínimo e máximo de cada gerador, será a restrição de demanda:

$$\sum_{i=1}^N P_i = D \quad (12)$$

Embora a restrição definida pela Equação 12 seja, também, a única restrição incluída no método para o PDE definido por Rodrigues (2007), além dos limites de potencia de cada unidade, outras restrições específicas do PCU poderiam ser consideradas, sem que isto impossibilitasse o uso do método para PDE. No entanto, a intenção inicial do trabalho é construir um método que possa solucionar um modelo básico do PCU. A partir deste método, outros aprimoramentos podem ser feitos futuramente.

3.3 Método BB para o PCU

O método de BB desenvolvido partiu da proposta de Jensen e Bard (2003), que consiste em uma abordagem genérica, embora seja mais explorada sob o ponto de vista da Programação Inteira Mista com Linear (PIML), para o qual a solução ótima é garantida.

A abordagem de *Benders*, no entanto, divide o problema inicial em um PDE não-linear e um problema inteiro e não-linear de cometimento de unidades. Logo, uma das dificuldades do desenvolvimento de uma abordagem de BB para o PCU foi identificar qual é a consequência da não-linearidade no problema e definir como isto pode ser tratado.

A parte de PI do problema engloba a escolha de quais unidades geradoras do sistema serão ligadas e quais permanecerão desligadas. Após a escolha, cabe ao PDE definir a potência a ser

utilizada de cada uma das unidades. Desta forma, a fim de melhor representar os dois problemas, pode-se reescrever a função objetivo como:

$$\text{Minimize } CO = \sum_{i=1}^N X_i \times CC_i(P_i) \quad (13)$$

A variável binária X_i introduzida na Equação 13 é a variável de decisão do PPI, que denota o estado da unidade geradora e assume o valor 1 para o estado ligado ou 0 para desligado. Esta variável multiplica o custo em função da potência, de modo que se a unidade estiver desligada, o custo será zero. Seguindo Jensen e Bard (2003) definiu-se o vetor X_k como o vetor de variáveis de decisão representadas em cada nó k da árvore de busca, que representa uma combinação única de valores 0 e 1.

A equação da restrição de potência também pode ser reescrita como:

$$\sum_{i=1}^N X_i \times P_i = D \quad (14)$$

A Equação 14 indica que caso a unidade esteja desligada, não haverá contribuição alguma de potência para atender a demanda.

Se o valor atribuído à variável X_i for 1, as Equações 13 e 14 permanecem inalteradas em relação às Equações 10 e 12, respectivamente. Caso o valor atribuído for 0, algumas mudanças podem ocorrer.

Na Equação 13, o CO é definido pela Equação 11, na qual o parâmetro c_i é independente da potência utilizada. No entanto, a eliminação do gerador pelo PCU, por meio de $X_i = 0$, implica que todos os custos em relação ao mesmo serão eliminados. Além disso, o limite mínimo de potência de cada gerador ligado sempre será utilizado pelo PDE, e uma mudança ocorre no valor da equação sempre que este limite é diferente de zero. Analogamente, o somatório da Equação 14 apresentará valores diferentes da Equação 12 sempre que o limite mínimo do gerador é diferente de zero. Assim, o valor de X_i está relacionado com as contribuições de custo e demanda de cada gerador e, desativando as unidades corretas, o PCU deve ser capaz de diminuir o custo operacional, quando comparado ao PDE no qual todas as N unidades se encontram ligadas.

3.3.1 Problema relaxado para o PCU

De acordo com Jensen e Bard (2003), para solucionar um problema, o BB precisa relaxar as variáveis inteiras 0-1 para variáveis fracionárias que podem assumir todos os valores entre 0 e 1. Cabe ao procedimento aproximativo, em seguida, definir qual valor, 0 ou 1, será atribuído à solução do PPI, respeitando as restrições. Nos exemplos citados pelos mesmos autores, os PPI são relaxados para PPL e o procedimento aproximativo consiste na simples aproximação numérica da solução fracionária encontrada pelo método de Programação Linear.

Para um PCU com N unidades geradoras, se for definido que todas as unidades devem permanecer ligadas, então o PDE calculará, para cada unidade, a potência a ser utilizada para suprir a demanda previamente estabelecida e respeitando as restrições de limite de potência mínimo e máximo de cada unidade.

A solução deste PDE pode ser modificada de forma a representar a porcentagem de uso de cada gerador, o que é feito dividindo-se a potência utilizada pelo limite máximo de potência de cada unidade. Para tal, define-se IU_i como o índice de utilização da unidade i :

$$IU_i = \frac{P_i}{P_{\max_i}} \quad (15)$$

Na Equação 15 P_i e P_{\max_i} representam a potência utilizada (calculada pelo PDE) e o limite máximo de potência para o gerador i , respectivamente. Quanto maior o valor de IU_i , melhor é classificado o gerador i . Desta forma, o valor de IU_i pode ser arredondado para 0 ou 1 e representar as variáveis de decisão X_i do PCU.

O PDE, nesta situação, desempenha a função de problema relaxado para o PCU. Para que um problema relaxado possa ser resolvido em um determinado nó na árvore de busca, deve-se considerar as variáveis X_i indefinidas como sendo provisoriamente iguais a 1, para que o PDE possa definir sua utilização e, posteriormente, o procedimento aproximativo possa decidir se a unidade deverá ser ligada ou não. Se for desejado conhecer o valor da função objetivo, que no caso deste trabalho representa o custo operacional, é necessário que o PDE seja resolvido novamente, desta vez com os novos valores de X_i definidos pelo procedimento aproximativo. Caso o valor encontrado seja melhor do que todos já obtidos anteriormente, esta solução passará a ser a incumbida.

Vale ressaltar que a conclusão de que o PDE pode ser utilizado como problema relaxado para o PCU é decisiva no que diz respeito à possibilidade de aplicação do método genérico de BB definido por Jensen e Bard (2003).

3.3.2 *Branch e redução do PCU inicial*

Durante a execução do BB, as variáveis de decisão são gradativamente fixadas em 0 ou 1, pela sub-rotina *branch*, e cada nó da árvore de busca representa uma combinação de variáveis fixas e variáveis indefinidas, sendo que as últimas, segundo Jensen e Bard (2003), constituem o problema relaxado. Portanto, o problema relaxado – leia-se PDE – a ser resolvido em cada nó é diferente.

Um caso de particular interesse é obtido quando uma ou mais variáveis de decisão são fixadas em 0. Quando isto acontece, a variável X_i , nas Equações 13 e 14, anula a iteração i do somatório, o que implica que o PDE a ser resolvido será menor. Do contrário, o PDE, como problema relaxado, deve também considerar todas as variáveis fixadas em 1, além das variáveis indefinidas. Isto é verdade, pois mesmo quando fixadas em 1, as variáveis X_i apenas indicam que as unidades estão ligadas e somente o PDE pode definir a potência a ser utilizada em cada combinação do PCU. Portanto, do ponto de vista do esforço computacional para a resolução do problema, é mais conveniente fixar as variáveis em 0 do que em 1.

A princípio, o parágrafo anterior parece divergir da idéia de Jensen e Bard (2003) quando afirma que uma variável fixa, especificamente em 1, deve ser considerada na resolução do problema relaxado. Isto se deve à estrutura particular do PCU, que constitui um problema duplo no qual o significado do problema relaxado é um pouco diferente. Para estar em conformidade com Jensen e Bard (2003), $X_i = 1$ deveria representar 100% de utilização da unidade i , enquanto $X_i = 0$ deveria representar uma utilização nula. No entanto, foi definido neste trabalho que X_i representa unicamente o estado ligado/desligado de cada unidade e a potência utilizada é definida pelo PDE e, portanto, $X_i = 1$ representa qualquer potência maior que zero. Como a combinação de unidades muda em cada nó da árvore de busca, a utilização da potência de cada gerador ligado também muda, de modo que o mesmo deve ser incluído na resolução do problema relaxado.

A partir deste conhecimento, o procedimento de *branch* foi projetado para escolher sempre o valor 0 para todo X_i , de modo que o problema relaxado possa ser menor conforme a busca prossegue. Para tal, o problema inicia com todas as unidades ligadas, seguindo a lógica dos

métodos de *Unit Decommitment*, como aquele implementado com sucesso por Tseng *et al* (2000). A inicialização do problema deste modo é vantajosa, pois ainda permite que o problema relaxado no nó zero seja capaz de fornecer um índice de utilização para todas as unidades existentes logo de início. O conjunto destes índices de utilização pode ser utilizado como uma lista de prioridades, dado que os maiores índices indicam unidades com mais chances de oferecer um custo menor, considerando que o PDE tende a aumentar a utilização de unidades com custo menor e, inversamente, diminuir a utilização de unidades com custo maior.

A maioria das propostas descritas nas seções 3.3.1 e 3.3.2 apresenta algumas complicações quando analisadas mais detalhadamente, sobretudo no que diz respeito à eficácia do Índice de Utilização (IU) como parte fundamental do procedimento aproximativo e como o significado atribuído a X_i interfere na capacidade do BB de encontrar a solução ótima. As seções seguintes fazem esta análise e discute as soluções desenvolvidas.

3.3.3 *Índice de Utilização Livre*

Índice de Utilização (IU), definido anteriormente, representa a porcentagem utilizada de cada gerador definido pelo PDE em um nó específico. O intuito deste índice é representar a importância do gerador para contribuir com a solução ótima – o que fará o procedimento aproximativo optar por ligá-lo ou desligá-lo. Embora o PDE atribua utilização maior às unidades melhores e utilização menor às unidades piores, os limites mínimos de potência de cada gerador são sempre utilizados, o que compromete a representatividade deste índice.

Outro problema é a presença do limite de potência máxima no denominador da Equação 14. Considera-se que quanto maior o valor de IU, melhor é o gerador. Todavia, um valor alto de limite de potência máxima – por exemplo, um limite maior do que a média dos outros geradores – causaria uma diminuição no valor de IU, levando a uma conclusão não fundamentada de que o gerador em questão é ruim.

Na tentativa de criar um parâmetro que pudesse medir a qualidade do gerador a partir de sua potência utilizada, definiu-se o Índice de Utilização Livre (IUL):

$$IUL_i = P_i - P \min_i \quad (16)$$

A Equação 16 determina que o IUL não mais é expresso em termos percentuais, como o IU, e também que este novo índice subtraí o limite de potência mínima do gerador, cuja utilização é obrigatória. Logo, o IUL é capaz de quantificar o que o PDE pode distribuir livremente entre as unidades, o que se espera poder retratar melhor a qualidade de cada unidade.

Embora a existência do limite de potência máxima no denominador possa distorcer a informação desejada, sua ausência resulta em outro problema. Uma unidade com limite máximo muito baixo tende a ser prontamente classificada como ruim, já que possuirá sempre um valor baixo para o IUL. Numa realidade onde existam várias unidades com capacidade baixa e, a despeito disso, possuam baixo custo operacional, o método apresentaria problemas em encontrar a solução ótima.

Para corrigir esta deficiência, foi determinado que um teste deve ser aplicado para identificar as unidades cuja utilização definida pelo PDE é próxima de 100% para que estas possam ser automaticamente ligadas. Determinou-se que a porcentagem de utilização a ser alcançada para que esta ação ocorra será representada pelo parâmetro α , que deve ser um dado de entrada do algoritmo.

3.3.4 Procedimento aproximativo utilizando Lista de Prioridades

A função do procedimento aproximativo é gerar uma solução inteira para o problema em um nó da árvore de busca, a partir da solução do problema relaxado neste mesmo nó. Para o PCU, a mera aproximação numérica da porcentagem de utilização de cada unidade para representar a variável X_i é, no mínimo, simplista.

O primeiro problema a surgir seria a possível inexistência de parâmetros superiores a 0,5 que pudessem ser arredondados para 1. Ainda assim, a lista destes parâmetros poderia ser normalizada, a fim de que os elementos fossem encaixados no intervalo de 0 a 1 e os novos valores pudessem ser arredondados. De qualquer modo, a escolha da quantidade de unidades a ser cometida seria aleatória e, em muitos casos, não apropriada. Faz-se necessária uma escolha racional, partindo dos dados de entrada do problema.

Constatou-se que a decisão de quais unidades cometer, mesmo valendo-se da solução do problema relaxado, é a questão mais difícil de se resolver dentro do procedimento aproximativo definido para o PCU. Muitas soluções factíveis, porém não ótimas, existem, de modo que o simples teste de factibilidade não é capaz para encontrar uma boa solução.

A solução gerada pelo procedimento aproximativo em um dado nó tem como principal objetivo servir de parâmetro para testar os descendentes deste mesmo nó. No final do procedimento aproximativo, quando já se possui uma solução inteira, é feita a decisão de continuar a busca com o *branch* – o que significa continuar fixando as variáveis restantes em zero – ou fazer o *backtrack*, caso o problema relaxado não ofereça uma solução melhor do que a incumbida. Isto equivale dizer que se a solução obtida pelo procedimento aproximativo não for suficientemente boa, descendentes com boas chances de oferecer a solução ótima podem ser excluídos da árvore de busca.

Por fim, a solução proposta para o procedimento aproximativo neste trabalho consiste na definição de uma lista de prioridades baseada no IUL a partir da qual as melhores unidades geradoras são cometidas, iterativamente, até que uma porcentagem β da demanda seja atendida. O valor de β deve ser obtido empiricamente, bem como o valor de todos os parâmetros criados neste método de BB. O valor de β deve, também, ser menor do que 100%, pois este valor leva ao cometimento de todas as unidades, dado que o PDE distribui as potências de forma que o total seja igual à demanda (conforme mostra a Equação 14).

Para que o procedimento aproximativo gere apenas soluções factíveis, é implementada a seguinte condição:

$$\sum_{i=1}^N X_i \times P_{\max_i} \geq D \quad (17)$$

A condição da Equação 17 garante a factibilidade da solução ao passo que determina que a soma dos limites máximos das unidades cometidas é maior do que a demanda D .

A definição de uma porcentagem da demanda para o parâmetro β em oposição a um valor fixo ou, por exemplo, à porcentagem da soma dos limites máximos, confere uma maior adaptabilidade do algoritmo ao caso específico do PCU que se deseja resolver. Assim, espera-se que os ajustes sobre o parâmetro afetem mais uniformemente os diversos casos demanda para o PCU, facilitando a identificação de um valor ideal. Apesar disso, é possível concluir, antecipadamente, que quanto maior for o coeficiente a da função de custo operacional da Equação 11, maior deverá ser o valor de β .

Para comprovar este fato, deve-se primeiramente lembrar que o problema relaxado é resolvido para todas as unidades fixadas provisoriamente em 1. Quando o procedimento aproximativo

deixa de ligar uma unidade, a potencia antes fornecida pelo gerador em questão deverá ser suprida pelos geradores restantes. Quanto maior for a inclinação da curva de custo (que varia mais ou menos de acordo com valor do parâmetro a) maior tende a ser o custo unitário por unidade de potência. Neste caso, a relação entre custo e benefício será melhor para utilizações pequenas de cada gerador, o que requer um maior número de unidades cometidas para atender a demanda. Se o número de unidades cometidas é maior, também deverá ser maior o valor de β , que funciona como uma condição de parada no processo definido anteriormente, em que as unidades são cometidas iterativamente de acordo com a lista de prioridades.

3.3.4.1 Lista de prioridades

O objetivo da lista de prioridades é classificar as unidades de modo que as melhores e as piores possam ser identificadas. O sentido das palavras “melhor” e “pior”, no entanto, não deve ser relativo, mas sim traduzir o objetivo real do problema. Caso se considere que o objetivo do problema seja a minimização dos custos operacionais, então as melhores unidades serão aquelas que oferecem o menor custo operacional e as piores, aquelas que apresentam custos maiores.

Embora esta pareça ser uma tarefa simples, pode-se apontar duas complicações. A primeira é que o custo é descrito por uma função quadrática, o que equivale dizer que a relação de custo e benefício da unidade varia para cada faixa de potencia utilizada. A segunda é que a potencia dos geradores utilizada em cada nó é diferente. Por fim, conclui-se que não é possível definir uma única lista de prioridades capaz de classificar as unidades durante todos os estágios da resolução do problema. Ao contrário, a lista de prioridades deve ser reformulada cada vez que um problema relaxado precise ser resolvido.

As unidades são classificadas da seguinte forma:

- a) Quanto maior o IUL, maior a prioridade da unidade;
- b) Como critério de desempate, compara-se o custo unitário de cada gerador em função de suas respectivas potências utilizadas. A unidade com menor custo unitário tem maior prioridade;
- c) As unidades cuja utilização ultrapassa a porcentagem α são automaticamente cometidas, independente da prioridade das demais unidades.

O teste utilizando o parâmetro α , em parte, desrespeita a lista de prioridades, porém, fundamenta-se no pressuposto de que o PDE, ao utilizar a unidade quase completamente, já a classificou como boa.

O critério de desempate, que toma como base o custo operacional da unidade em função da potência utilizada, seria pouco efetivo caso aplicado apenas nos casos de estrita igualdade de dois valores de IUL. Estes índices são calculados pelo algoritmo com diversas casas de precisão e, embora o simples arredondamento pudesse facilitar a colisão de dois valores, esta medida possui pouco efeito.

Foi definida uma tolerância de proximidade para os valores dos IUL, de modo que a seleção da melhor ou pior unidade, a partir desta proximidade, é dada em função do custo operacional unitário para a unidade. A escolha do custo operacional unitário em oposição ao custo operacional total justifica-se pela existência de mesmos valores de IUL para quantidades de potência diferente, de modo que uma unidade poderia obter um custo menor apenas porque sua utilização é menor, e não por que é capaz de oferecer unidades de potência a um custo menor.

A proximidade que aciona o critério de desempate é medida pelo parâmetro μ através do teste da Equação 18.

$$|IUL_i - IUL_j| \leq \mu \times D \quad (18)$$

Foi considerado que a proximidade entre os parâmetros deveria ser ajustada de acordo com o valor da demanda, a qual é diretamente proporcional ao valor dos IUL. Assim, o teste da Equação considera uma proximidade menor ou igual a uma porcentagem μ da demanda. Quanto maior a demanda, maior será a dispersão dos IUL e, portanto, uma maior distância é permitida para acionar o critério de desempate.

3.3.4.2 Aprimoramento da lista de prioridades

Após os testes de calibração de parâmetros, a serem expostos no Capítulo 4, foi constatado que a precisão da solução para os casos de demanda baixa não era tão boa quanto a obtida para casos de demanda maior, considerando o mesmo sistema de unidades geradoras. Este resultado revela uma perda de eficácia da lista de prioridades para classificar as unidades

quando a potência requisitada às unidades é muito próxima de seus limites mínimos de potência. Este problema, no entanto, pode ser corrigido.

A perda da eficácia de fato ocorre devido ao comprometimento dos IUL, que indicam a potência utilizada além do limite mínimo de cada gerador. Quando a demanda é muito próxima da soma dos limites mínimos dos geradores do sistema, o PDE possui menos liberdade de atribuir uma maior utilização às unidades com menor custo, devido à necessidade de satisfazer a restrição de limites mínimos. Os valores do IUL tendem a permanecer próximos de zero e com pouca significância. Por outro lado, quando a demanda é alta, a porcentagem de demanda que o PDE deve atribuir para satisfazer os limites mínimos é menor, de forma que resta uma grande quantidade de potência a ser distribuída às unidades.

Para aprimorar a lista de prioridades em sua capacidade de classificação das unidades, nos diferentes casos de demanda, foi implementado um ajuste na demanda utilizada para resolver o problema relaxado, que calcula as utilizações de potência para compor os IUL. Este novo valor de demanda passou a ser chamado de Demanda de Teste (DT) e substitui o valor da demanda quando este se encontra abaixo da porcentagem δ da soma dos limites máximos dos geradores no sistema.

$$DT = D + \left(\delta - \frac{D}{\sum_{i=1}^N L \max_i} \right) \times \sum_{i=1}^N L \max_i \times \varepsilon \quad (19)$$

A Equação 19 parte do valor da demanda D para fazer um incremento na faixa de valores de zero até δ . A dimensão do incremento é inversamente proporcional à demanda. Se D for próximo de zero, o incremento feito será o maior permitido. Por outro lado, se D atingir uma porcentagem δ da soma dos limites máximos, nenhum incremento é feito. Este último caso também ocorre quando D é maior do que a porcentagem δ , já que para estes casos a demanda de teste não se aplica. O parâmetro ε quantifica o incremento feito sobre D , sem alterar a faixa de valores afetada.

A resolução do problema relaxado com uma demanda diferente da real não apresenta problema algum, já que o resultado deste problema é utilizado apenas para que o procedimento aproximativo possa gerar a lista de prioridades. Conforme dito anteriormente,

após a escolha das unidades a serem ligadas, é necessário resolver o PDE novamente (não mais como problema relaxado) para encontrar o valor da função objetivo.

Espera-se que estas ações possam solucionar o problema da lista de prioridades, visto que seu desempenho foi melhor para demandas maiores.

3.3.4.3 Aprimoramento do parâmetro β

Após os testes de calibração de parâmetros também foi constatado que o método fornecia resultados mais precisos se o parâmetro β fosse proporcional à relação entre demanda e soma dos limites máximos. O principal problema observado foi que a precisão do algoritmo diminuía caso a demanda fosse próxima da soma dos limites mínimos e o valor de β fosse grande.

A solução proposta foi definir um auto-ajuste para o parâmetro β em função da fração da demanda e da soma dos limites máximos, conforme a equação:

$$\beta = B - \eta \times \left(\delta - \frac{D}{\sum_{i=1}^N L \max_i} \right) \times \frac{1}{\delta} \quad (20)$$

A Equação 20 introduz um novo parâmetro η , que quantifica o incremento dado ao parâmetro para o valor de β . O restante do termo que multiplica η é responsável por calcular um valor de 0 a 1, em função da demanda e da soma dos limites máximos, que permitirá maior ou menor atuação do parâmetro η . O parâmetro B é o limite superior para β , sendo que o limite inferior é obtido por B- η . Em resumo, a Equação 20 faz com que o parâmetro β varie linearmente entre estes dois limites.

O parâmetro δ na Equação 20, assim como na Equação 19, é responsável pela faixa de atuação do ajuste. Por fim, o valor de β é definido pela Equação 20 se a razão da demanda e da soma dos limites máximos for menor do que δ ou pelo próprio valor de B, caso contrário. Esta ação se justifica pelo interesse em corrigir o valor de β apenas para os casos de demanda baixa, que obtiveram resultados menos precisos.

Foi mencionado anteriormente, que o ajuste do parâmetro β deveria considerar os coeficientes da função de custo, já que este fato influencia na utilização de mais ou menos unidades. No

entanto, dada a complexidade desta análise, no atual trabalho não foi desenvolvido nenhum ajuste desta natureza.

3.3.5 Escolha da unidade para o branch

A partir da definição de que o *branch* optará sempre por $X_i = 0$, resta decidir de qual das unidades indefinidas será escolhida em cada nó. Antes de discutir qual a estratégia foi adotada para esta escolha, é importante ressaltar que o procedimento *branch* implementado, assim como o sugerido por Jensen e Bard (2003), é livre para escolher qualquer variável X_i para ser explorada em cada nível l da árvore de busca, o que não necessariamente acontece nos algoritmos convencionais de busca em profundidade ou largura.

Jensen e Bard (2003) definem duas estratégias para escolha da variável para o *branch*, sendo que se optou por selecionar a variável e a direção – neste caso, zero – cujo nó criado tenha maior chance de conter a solução ótima.

Deste modo, é preciso apontar qual a unidade que, caso desligada, possui a maior chance de contribuir para solução ótima, ou seja, mais contribui para a diminuição do custo operacional total para atender a demanda. Esta unidade é tal que pode ter sua potência utilizada suprida por outras unidades com custo menor.

Após definida, no procedimento aproximativo, uma lista de prioridades capaz de classificar as unidades em cada nó da árvore de busca, é fácil escolher a última colocada e executar o *branch* em zero para a variável que a representa. Como a pior unidade é eliminada, mais cedo soluções boas são encontradas na busca e, com isso, mais nós (com seus respectivos descendentes) podem ser eliminados, acelerando a convergência do algoritmo.

3.3.6 Sub-rotina Restringir

Nesta etapa é verificada a factibilidade do problema para o nó em questão e, em caso afirmativo, um problema relaxado é resolvido a fim de testar a capacidade do nó e seus descendentes de fornecer uma solução melhor do que a incumbida. O problema cuja factibilidade é verificada considera todas as variáveis indefinidas iguais a 1. O teste de factibilidade, por sua vez, consiste na verificação da Equação 16, que garante a satisfação da restrição definida pela Equação 14.

O segundo teste é o teste de limites, que compara a solução do problema relaxado no nó com a solução incumbida. Caso a solução do problema relaxado seja melhor do que a incumbida, o procedimento aproximativo é invocado e, ao final do mesmo, será decidido continuar com o *branch* ou fazer o *backtrack*.

Como sugerido por Jensen e Bard (2003), foi permitida uma porcentagem de erro, definida pelo usuário, a ser tolerada pelo teste de limites. Esta porcentagem de erro, denominada γ , faz com que eficiência do algoritmo possa ser favorecida a custo de uma menor precisão na solução do problema. Assim, o teste de limites, antes definido pela Equação 7, fica da seguinte forma:

$$Z_{UB} \times (1 - \gamma) \leq Z_B \quad (21)$$

Foi determinado que o parâmetro γ na Equação 21 pode ser positivo ou negativo, de acordo com o efeito desejado. Caso seja atribuído um valor negativo, a precisão é comprometida em favor da eficiência, pois Z_{UB} é aumentado percentualmente, dificultando a satisfação da condição. Se a condição não é satisfeita, o nó é eliminado. Isto possibilita que nós com pouca vantagem sobre a solução incumbida sejam descartados. Acredita-se que este parâmetro seja útil para casos do PCU com grande número de geradores.

Por outro lado, se um valor positivo for atribuído a γ , a busca tende a continuar mesmo que Z_{UB} não seja menor do que Z_B , fazendo que mais nós na árvore de busca sejam investigados à procura da solução ótima. Esta estratégia, embora aumente o tempo de processamento, pode ser útil para compensar possíveis imperfeições do modelo de problema relaxado definido para este trabalho, como será discutido no Capítulo 4.

3.3.7 Demais sub-rotinas do método BB

As sub-rotinas “atualizar” e “*backtrack*” não necessitaram de adaptações para cumprir os objetivos deste trabalho e seguiu a proposta de Jensen e Bard (2003). De fato estes dois procedimentos não oferecem grandes margens à personalização, pois sua função é garantir o funcionamento da busca. Quanto ao *backtrack*, se a definição do valor para o *branch* é sempre 0, então automaticamente o *backtrack* atribui o valor 1, dado que as variáveis de decisão são binárias.

Para este trabalho, nenhum procedimento de ajuste de variáveis foi desenvolvido, dada sua complexidade e necessidade de formulação e análise de uma matriz de restrições para o problema.

4 Resultados e Discussão

Para validar o método proposto, o primeiro passo foi executar a calibração dos parâmetros, a fim de encontrar valores com os quais o algoritmo fosse capaz de solucionar o problema com rapidez e precisão equilibradas. Em seguida, foram testados alguns casos do PCU, com a melhor calibração de parâmetros encontrada, para determinar as melhores soluções geradas pelo método. Alguns casos foram comparados com a enumeração exaustiva que, em princípio, é capaz de encontrar a solução ótima para o PCU a custo de um grande esforço computacional.

4.1 Calibração de parâmetros

A calibração dos parâmetros foi executada por meio do teste de várias combinações de valores. Há uma relação entre o espaçamento dos valores selecionados e o número de combinações que devem ser testadas a fim de cobrir um determinado intervalo. Quanto menor o intervalo, mais preciso é o teste, no entanto, mais combinações devem ser testadas para uma determinada faixa de valores. Quanto maior a faixa de valores testados, mais será revelado sobre a influência dos parâmetros sobre a solução final.

A partir de estimativas e alguns testes preliminares, foi possível estabelecer todas as combinações de valores que se acreditou ser capaz possibilitar a análise de sensibilidade de maneira viável.

O primeiro teste foi executado para um caso com 6 geradores. Foram considerados para esta análise os parâmetros α , β e γ , os quais foram combinados com diferentes casos de demanda e número de unidades geradoras.

O Quadro 1, no qual D indica a demanda, mostra os valores das principais variáveis de interesse, enquanto outros valores, como os coeficientes da função de custo operacional e os limites mínimos e máximos de potência para cada unidade encontram-se no Anexo A.

Σ Lim. Mín.	Σ Lim. Máx.	D	α	β	γ
345	1350	550	0,85	0,0	-0,04
		1000	0,90	0,4	0,00
			0,95	0,6	0,04
			1,00	0,9	0,08

Quadro 1: Valores das variáveis para o caso de 6 geradores

Os valores definidos no Quadro 1 resultaram em 128 combinações diferentes para teste, sendo que os valores de α , β e γ geraram 64 combinações, as quais foram aplicadas para dois casos de demanda do PCU – um para 550 e outro para 1000 unidades de potência. A determinação de valores de demanda mais próximos dos limites de potência foi uma tentativa de investigar o desempenho do método e dos parâmetros criados para os casos de grande e pequeno uso da capacidade dos geradores.

O Quadro 2 define os valores dos parâmetros testados para o caso de 13 geradores.

Σ Lim. Min.	Σ Lim. Máx.	D	α	β	γ
550	2960	1000	0,90	0,4	-0,04
		2000	0,95	0,6	0,00
				0,8	

Quadro 2: Valores das variáveis para o caso de 13 geradores

Como se observa pelo Quadro 2, o caso de 13 geradores foi testado para uma quantidade menor de valores, que resultou em 24 combinações. O menor número de combinações deve-se ao fato de que o teste em questão foi realizado após o teste com 6 geradores, o que permitiu a eliminação de alguns valores para os quais não se obtiveram boas soluções.

A saber, uma das principais diferenças entre os casos de 6 geradores e 13 geradores testados é que no último as curvas da função de custo são definidas com um menor valor do coeficiente α , o que equivale dizer que são mais próximas da linearidade. Esperou-se, portanto, uma mudança de comportamento para o parâmetro β , de acordo com o que foi discutido no Capítulo 4.

Até este momento, os parâmetros definidos para o aprimoramento da lista de prioridades e do auto-ajuste de β não existiam. Estes parâmetros – δ , ϵ , B e η – foram aplicados para corrigir as falhas observadas por meio destes primeiros testes.

As tabelas detalhadas contendo as soluções para cada combinação do problema encontram-se no Apêndice A, de modo que apenas as tabelas resumidas serão utilizadas para a discussão. As tabelas resumidas para os parâmetros α , β e γ consideram a média de todas as combinações nas quais o referido valor para o parâmetro foi utilizado. Esta média representa a média das soluções obtidas fixando-se o valor do parâmetro e variando todos os outros. Não foi mensurada numericamente a dispersão dos valores da solução, pois esta análise foi feita

visualmente por meio das tabelas constantes no Apêndice A e pela comparação entre os valores médio e melhor.

Antes de verificar os valores obtidos pelo método BB, serão apresentados os valores obtidos por meio da enumeração exaustiva, para a qual também foi desenvolvido um algoritmo, no intuito de fornecer um parâmetro para comparações do desempenho do método BB.

Tabela 1: Resultados da enumeração exaustiva para um caso de 6 geradores

D	X_B	Z_B	Tempo (s)
500	0,0,0,0,280,269	26486	402
1000	0,0,181,178,325,314	49408	365

A Tabela 1 mostra os valores de Z_B e o tempo de execução para o caso de 6 geradores. O menor valor de tempo para o caso da demanda igual a 1000 deve-se a um menor número necessário de resoluções do PDE, dado que o algoritmo de enumeração exaustiva implementado executa um teste de factibilidade antes solucionar o mesmo. Este teste de factibilidade considera a restrição definida pela Equação 17.

O caso de 13 geradores não pôde ser executado até o fim por motivos de limitação do hardware disponível na ocasião. O aplicativo reportou um erro, em tempo de execução, de falta de memória.

Tabela 2: Resultados da enumeração exaustiva para um caso de 13 geradores

D	X_B	Z_B	Tempo (s)
1000	0,0,360,0,178,0,180,179,0,0,0,101,0	9525	11240

O valor apresentado na Tabela 2 consiste em uma solução aproximada para o método de enumeração exaustiva, para um tempo de processamento de 3 horas e 7 minutos, apenas para o caso da demanda igual a 1000. A explosão do tempo de execução conforme o aumento de unidades era esperado, para a enumeração exaustiva, já que as combinações a serem testadas aumentam exponencialmente. Embora não tenha sido possível executar o algoritmo para a enumeração exaustiva para o caso com 13 geradores, o algoritmo de BB proposto foi capaz de solucioná-lo – após a calibração de parâmetros – em pouco mais de 2 minutos.

4.1.1 Parâmetro γ

O primeiro parâmetro a ser analisado para o método proposto de BB é γ , o qual atua no teste de limites, que compara Z_B e Z_K na tentativa de eliminar o nó e seus descendentes. A Tabela 3 exibe os resultados do teste para o caso de 6 geradores.

Tabela 3: Resultados do parâmetro γ para um caso de 6 geradores

D	γ	Z_B Médio	Melhor Z_B	Tempo Médio (s)	Melhor Tempo (s)
500	-0,04	27367	26487	85	69
500	0,00	27255	26486	82	61
500	0,04	26958	26486	127	70
500	0,08	26486	26486	612	570
1000	-0,04	49410	49408	100	75
1000	0,00	49409	49407	82	69
1000	0,04	49408	49407	1871	1536
1000	0,08	49408	49407	1998	1946

A Tabela 3 permite observar os valores médios para Z_B e para o tempo de execução, bem como os melhores resultados obtidos para cada valor de β . A diferença entre os valores médios e o melhor valor retrata a influência dos outros parâmetros sobre o resultado final e sobre os parâmetros entre si. Embora o melhor resultado de Z_B seja alcançado independente do valor de γ , a média dos resultados é melhor quando γ mais próxima de 0,08. Isto significa que, para cada valor de β , existem uma ou mais combinações de outros parâmetros capazes de se adaptar ao efeito de β e, assim, encontrar a solução ótima. No entanto, se for tomado o valor de β para o qual a média é menor, então este valor tem mais chances de contribuir para a obtenção da solução ótima – inclusive pelo fato de que os outros parâmetros ainda não foram definidos.

Por outro lado, o melhor resultado para o tempo não é alcançado independente de γ , sendo que este, bem como o valor médio, encontra-se quando γ é igual a 0.

Os resultados dos 128 casos que constituem este teste de 6 geradores apresentaram relativamente pouca variação para os valores de Z_B e, por outro lado, maior variação para o tempo de execução. Percebeu-se, no entanto, que o maior responsável pela variação do tempo é o parâmetro γ , cuja relação com o esforço computacional já era conhecida – de fato, o objetivo planejado para este parâmetro é escolher entre o tempo de processamento e a precisão da solução.

Nota-se que os resultados para o teste com 13 geradores, pela Tabela 4, apresentam um comportamento similar ao teste com 6 geradores.

Tabela 4: Resultados do parâmetro γ para um caso de 13 geradores

D	γ	Z_B Médio	Melhor Z_B	Tempo Médio (s)	Melhor Tempo (s)
1000	-0,04	10120	9616	156	133
1000	0,00	10109	9617	155	129
2000	-0,04	18710	18678	169	138
2000	0,00	18739	18678	164	141

Para o teste de 13 geradores, os valores de γ para os quais o tempo foi excessivamente maior, sem melhorar o valor de Z_B , não foram aplicados. Por este motivo, os valores médios de tempo se apresentam mais próximos dos melhores valores.

O comportamento esperado para este parâmetro era uma relação proporcional entre aumento da precisão, do esforço computacional e do valor de γ , o que não ocorreu em grandes proporções – talvez pela pequena variação dos valores testados – principalmente quando γ assumiu um valor negativo. Neste caso, esperava-se que o tempo de processamento fosse menor do que o obtido para γ igual a 0. Embora este resultado possa ser diferente para outros casos do problema, há uma possível explicação para que isto tenha ocorrido.

Os primeiros estágios da busca são mais determinantes com relação à sua eficiência, pois quanto mais cedo boas soluções forem encontradas, mais nós poderão ser eliminados da árvore de busca, pois os nós eliminados se encontrarão mais próximos do nível zero e, portanto, possuirão mais descendentes. O efeito de valores negativos de γ sobre o teste de limites – aquele que elimina os nós – é fazer com que este desista mais facilmente de investigar seus descendentes, o que pode incorrer na perda de boas soluções. Caso estas soluções sejam perdidas em etapas iniciais da busca, há chances de que o algoritmo possa levar mais tempo para convergir, já que Z_B será pior, tendo assim, menos capacidade de eliminar nós. É possível, também, que valores mais negativos (menores do que -0,04) sejam capazes de suprimir este problema e oferecer resultados em menor tempo. Considerando que estas hipóteses sejam verdadeiras, pode-se estudar a possibilidade de diminuir γ gradualmente durante a execução do algoritmo. Desta forma, valores mais negativos seriam obtidos nas etapas intermediárias e finais, sem prejudicar o desempenho da etapa inicial.

De qualquer modo, no sentido de obter soluções em menor tempo, é importante lembrar que os métodos BB podem ser interrompidos a qualquer momento, de forma que Z_B , no estado em que se encontra, pode ser tomado como solução aproximada. Uma análise pode ser feita a fim de mensurar a convergência de Z_B durante a execução do algoritmo, possibilitando que um valor de tempo que ofereça soluções aproximadamente ótimas seja determinado. Esta análise não foi feita no atual trabalho, podendo ser desempenhada em trabalhos futuros.

Por fim, o Quadro 3 apresenta um resumo para o parâmetro γ , no qual constam a ação executada, os limites do intervalo de atuação, os resultados da análise de sensibilidade, o efeito causado no algoritmo, algumas características e o melhor valor obtido, a ser definido como padrão.

Parâmetro γ	
Ação	Atua no teste de limites, dentro da sub-rotina “restringir”, onde aumenta ou diminui Z_K percentualmente, a fim de dificultar ou facilitar a satisfação do teste que compara Z_K e Z_B .
Limites	Depende da intenção do usuário, mas deve ser próximo de zero. Pode ser negativo ou positivo.
Análise de sensibilidade	
Mais positivo	Mais fácil de Z_K ser melhor que Z_B , desiste mais dificilmente.
0	Não altera Z_K , deixando a escolha simplesmente pela comparação de Z_K e Z_B .
Mais negativo	Mais difícil de Z_K ser melhor que Z_B , desiste mais facilmente.
Considerações	
Efeito	Valores mais positivos aumentam a precisão do algoritmo, pois diminuem as incertezas do processo aproximativo, no entanto aumentam consideravelmente o tempo. Valores mais negativos diminuem a precisão do algoritmo, pois elimina mais nós, mas diminuem o tempo. Pode ser útil para resolver problemas com muitos geradores em menor tempo.
Características	Deve ser atribuído de acordo com a intenção do usuário.
Valor padrão	0 (obtido empiricamente)

Quadro 3: Análise de sensibilidade para γ

A partir das análises, o valor 0 foi escolhido como o mais ideal para constituir o padrão para γ , visto que o aumento do mesmo não causou o esperado aumento de precisão, mas sim um grande aumento do tempo de execução.

Após o questionamento cuidadoso deste fato, concluiu-se que a obtenção dos melhores resultados com $\gamma = 0$ é um bom sinal. Caso o valor de γ fosse maior, aumentando a taxa de investigação dos nós, e a precisão de Z_B fosse melhorada, isso provaria a ineficácia do processo aproximativo de resolver o problema relaxado no nó. Isto é verdade pois a resolução

do problema relaxado em um nó da árvore de busca tem o principal objetivo de qualificar as soluções capazes de serem geradas pelos nós filhos, visto que a investigação do nó – por meio do *branch* – só continua caso a solução para problema seja melhor do que Z_B .

4.1.2 Parâmetro β

O próximo parâmetro a ser analisado é β , que representa uma porcentagem da demanda. A Tabela 5 mostra os resultados para o caso de 6 geradores.

Tabela 5: Resultados do parâmetro β para um caso de 6 geradores

D	β	Z_B Médio	Melhor Z_B	Tempo Médio (s)	Melhor Tempo (s)
500	0,0	27025	26486	211	61
500	0,4	26935	26486	213	73
500	0,6	26845	26486	214	70
500	0,9	27262	26486	269	80
1000	0,0	49409	49407	1028	75
1000	0,4	49409	49407	1001	69
1000	0,6	49408	49407	1020	81
1000	0,9	49409	49408	1002	74

Pela Tabela 5, nota-se que, de forma similar ao parâmetro γ , o melhor resultado para Z_B é alcançado independente do valor de β , mas a média dos resultados é melhor quando β assume um valor específico, neste caso, 0,6. O melhor tempo de execução, por sua vez, é dependente do valor de β .

Para o caso de 13 geradores, no qual o parâmetro a das funções de custo são menores, observou-se uma melhor precisão do algoritmo quando o valor de β foi menor. Os resultados constam na Tabela 6.

Tabela 6: Resultados do parâmetro β para um caso de 13 geradores

D	β	Z_B Médio	Melhor Z_B	Tempo Médio (s)	Melhor Tempo (s)
1000	0,4	9617	9616	136	129
1000	0,6	10044	10043	150	131
1000	0,8	10682	10556	180	171
2000	0,4	18679	18678	163	141
2000	0,6	18679	18678	160	138
2000	0,8	18816	18859	177	155

É possível observar, na Tabela 6, que aumento da precisão com a diminuição de β , sobretudo para os casos de menor valor de demanda. Foi observado, principalmente, que para estes

casos a perda da precisão não ocorreu apenas quando comparado aos casos de maior valor para β , mas também com relação à enumeração exaustiva, de acordo com a Tabela 1.

Neste momento, foram feitas as análises expostas no Capítulo 4 sobre o aprimoramento da lista de prioridades e do ajuste do parâmetro β , a fim de superar a atual deficiência do método.

Depois de implementada as melhorias, foi executado um novo teste, que será demonstrado após a análise dos parâmetros δ , ε , B e η .

O Quadro 4 apresenta um resumo para o parâmetro β , no qual constam a ação executada, os limites do intervalo de atuação, os resultados da análise de sensibilidade, o efeito causado no algoritmo, algumas características e o melhor valor obtido, a ser definido como padrão.

Parâmetro β	
Ação	Permite o cometimento das unidades em ordem crescente de IUL, até que uma porcentagem β da demanda seja alcançada.
Limites	O limite superior é 1 (representa 100%) e o limite inferior é 0 (representa 0%)
Análise de sensibilidade	
Igual a 1	Seleciona todas as unidades, pois a soma da potência atribuída a cada uma resulta em 100% da demanda, após a resolução do PDE.
Quanto maior	Seleciona mais unidades para serem cometidas, resultando em uma utilização menor de cada uma.
Quanto menor	Seleciona menos unidades para serem cometidas, resultando em uma utilização maior de cada uma.
Igual a 0	Seleciona as unidades apenas com a restrição da soma da capacidade máxima ser suficiente para atingir a demanda.
Considerações	
Efeito	Permitiu implementar o procedimento aproximativo através de uma lista de prioridades.
Características	Este parâmetro deve ser calibrado de acordo com os seguintes dados: demanda, soma dos limites máximos, coeficientes das funções de custo.
Melhor valor	Ajuste automático entre 0,3 e 0,6 (obtido empiricamente)

Quadro 4: Análise de sensibilidade para β

Os resultados obtidos para os diferentes casos de β são condizentes com as expectativas descritas no Capítulo 4. Pelas tabelas detalhadas, constantes no Apêndice A, foi possível constatar que, de fato, quanto maior o valor de β , mais unidades são selecionadas e um percentual menor é utilizado de cada uma. Por outro lado, quanto menor o valor de β , menos unidades são selecionadas e um percentual maior de cada uma é utilizado.

4.1.3 Parâmetro α

Este parâmetro é utilizado no teste que comete automaticamente uma unidade, dada sua porcentagem α utilização. Os resultados apresentados apresentam relativamente quase nenhuma variação para os melhores valores de tempo e Z_B , conforme mostra a Tabela 7.

Tabela 7: Resultados do parâmetro α para um caso de 6 geradores

D	α	Z_B Médio	Melhor Z_B	Tempo Médio(s)	Melhor Tempo (s)
500	0,85	27061	26486	220	61
500	0,90	26995	26486	225	70
500	0,95	26972	26486	229	70
500	1,00	27039	26486	233	74
1000	0,85	49409	49407	965	69
1000	0,90	49409	49407	1030	74
1000	0,95	49408	49408	1025	74
1000	1,00	49409	49407	1029	73

A maior influência do parâmetro α é observada sobre os valores médios de tempo, os quais são bastante diferentes dos melhores valores. No entanto, a existência de valores altos para o tempo médio foi causada pelo parâmetro γ , conforme discutido na seção 4.1.1, e não pela influência de α – a notar pela baixa correlação de α com os valores de tempo médio.

A Tabela 8 mostra os resultados para o caso de 13 geradores, para o qual os valores de γ causadores do longo tempo de execução não foram incluídos no teste.

Tabela 8: Resultados do parâmetro α para um caso de 13 geradores

D	α	Z_B Médio	Melhor Z_B	Tempo Médio(s)	Melhor Tempo (s)
1000	0,9	10133	9616	150	129
1000	0,95	10096	9616	160	133
2000	0,9	18709	18678	165	138
2000	0,95	18740	18678	168	145

Pela Tabela 8, não é possível estabelecer uma relação clara entre α e a média das soluções encontradas, no entanto, as soluções são idênticas no melhor caso de Z_B . O tempo de processamento é ligeiramente diminuído para o valor de α igual a 0,9.

De forma similar ao que foi feito com os outros parâmetros, organizou-se um resumo de informações para o parâmetro α , que pode ser visto no Quadro 5, no qual constam a ação executada, os limites do intervalo de atuação, os resultados da análise de sensibilidade, o

efeito causado no algoritmo, algumas características e o melhor valor obtido, a ser definido como padrão.

Parâmetro α	
Ação	É testado no intuito de cometer automaticamente uma unidade caso esta possua uma utilização de no mínimo $\alpha\%$ de sua capacidade máxima.
Limites	O limite superior é 1 (representa 100%) e o limite inferior é 0 (representa 0%)
Análise de sensibilidade	
Igual a 1	Comete unidade automaticamente apenas se 100% de sua capacidade é utilizada.
Quanto maior	Comete unidades automaticamente mais raramente.
Quanto menor	Comete unidades automaticamente com mais frequência.
Igual a 0	Comete todas as unidades.
Considerações	
Efeito	Suprime uma deficiência na eficácia do IUL. Como o limite máximo não é considerado no cálculo do IUL, os geradores de baixa capacidade serão menos escolhidas, pois apenas uma quantidade pequena (menor que o limite superior) é fornecida pelas mesmas, acusando, erradamente, que a unidade é ruim.
Características	A existência do teste para este parâmetro é mais importante do que sua calibração, portanto, pode ser fixo em valores próximos a 1 para quaisquer casos do problema.
Melhor valor	0,9 (obtido empiricamente)

Quadro 5: Análise de sensibilidade para α

A conclusão a respeito este parâmetro é que os valores testados, os quais foram próximos a 1, contribuem de forma parecida para o desempenho do algoritmo. Desta forma, a existência do teste para cometer as unidades próximas de sua utilização total é, de fato, o fator relevante. No entanto, valores muito abaixo de 1 prejudicariam o desempenho do algoritmo, de modo que não seriam respeitadas as escolhas feitas pelo procedimento aproximativo por meio da lista de prioridades.

4.2 Aprimoramento da lista de prioridades

Após os testes de calibração de parâmetros, discutidos nas seções anteriores, foi constatada uma perda de precisão para os casos nos quais a demanda era significativamente menor do que a soma dos limites máximos das unidades.

Neste contexto, o fato que merece maior atenção é a divergência das unidades cometidas pela solução obtida pela enumeração exaustiva e o algoritmo BB, o que pode ser constatado através da comparação dos resultados da Tabela 2 e da tabelas detalhada para o mesmo caso

do problema resolvido com o método BB, que se encontra no Apêndice A. Este fato indica que, para este caso do problema, o algoritmo BB ficou preso em um ótimo local, devido a presente falha do procedimento aproximativo, mais precisamente, com relação à lista de prioridades. A causa da ineficácia da lista de prioridades, neste caso, já foi discutida no Capítulo 4, que também apresentou uma proposta de solução para o problema.

Esta questão foi solucionada com a definição da Demanda de Teste, cujo valor é ajustado pelos parâmetros δ , que indica os limites de atuação do incremento feito à demanda inicial, e ε , que quantifica este incremento.

Os resultados da Tabela 9 constituem o resultado para a versão aprimorada do método BB desenvolvido neste trabalho.

Tabela 9: Resultados dos parâmetros para um caso de 13 geradores (com aprimoramento da lista de prioridades)

D	β	α	γ	δ	ε	Z_B	Tempo
1000	0,4	0,9	-0,04	0,7	0,5	9146	142
1000	0,4	0,95	-0,04	0,7	0,5	9144	177
1000	0,4	0,9	0	0,7	0,5	9146	165
1000	0,4	0,95	0	0,7	0,5	9144	183
1000	0,6	0,9	-0,04	0,7	0,5	9144	168
1000	0,6	0,95	-0,04	0,7	0,5	9144	171
1000	0,6	0,9	0	0,7	0,5	9146	177
1000	0,6	0,95	0	0,7	0,5	9145	201
1000	0,8	0,9	-0,04	0,7	0,5	9406	212
1000	0,8	0,95	-0,04	0,7	0,5	9406	165
1000	0,8	0,9	0	0,7	0,5	9407	212
1000	0,8	0,95	0	0,7	0,5	9146	154

Para a obtenção dos valores da Tabela 9, apenas os valores de β foram ajustados manualmente, dada a intenção de verificar a sua influência sobre os resultados. Neste sentido, é visível que, após o aprimoramento da lista de prioridades, o resultado obtido é melhor mesmo quando valores considerados inadequados de β são testados – a exemplo do valor 0,8.

O valor de δ foi obtido por meio da análise dos resultados da calibração de parâmetros, chegando-se ao valor de 0,7, que indica que toda demanda abaixo de 70% da soma dos limites máximos deve ser incrementada a fim de oferecer uma melhor estimativa para os IUL.

O valor de ε foi obtido por estimativas e testes, sendo que o valor 0,5 se mostrou eficaz para os casos testados do problema. Os testes para este parâmetro não foram exaustivos,

considerando que um valor ideal não poderia ser definido para todos os casos. Mais testes poderiam ser elaborados no sentido de formular um ajuste automático para o parâmetro.

O Quadro 6 apresenta um resumo das informações obtidas para o parâmetro ϵ .

Parâmetro ϵ	
Ação	Define um incremento na demanda para calcular os IUL de acordo com a Equação 19. Se a demanda é pequena, o incremento é maior, caso contrário, o incremento é menor. Atua apenas sobre os valores abaixo de $\delta\%$ da soma dos limites máximos.
Limites	O limite superior é 1 (representa 100%) e o limite inferior é 0 (representa 0%)
Análise de sensibilidade	
Igual a 1	Um valor D abaixo de $\delta\%$ da soma dos limites máximos resultará em DT igual a $\delta\%$ da soma dos limites máximos.
Quanto maior	Valores de DT mais próximos de $\delta\%$ da soma dos limites máximos.
Quanto menor	Valores de DT mais próximos de D.
Igual a 0	Não ocorre ajuste, logo $DT = D$.
Considerações	
Efeito	Melhorou a qualidade da solução para casos de demanda pequena, com um aumento desprezível de tempo de processamento, comprovando a hipótese de que este era o motivo da perda de precisão
Características	A Demanda de Teste, variável na qual o parâmetro ϵ atua, possibilitou uma melhor classificação das unidades na lista de prioridades, ao passo que as testa com uma demanda maior.
Valor padrão	0,5 (obtido por estimativa)

Quadro 6: Análise de sensibilidade para ϵ

Quanto à sensibilidade de ϵ , descrita no Quadro 6, se um valor muito baixo for definido para o parâmetro, o ajuste pode ser insuficiente. Caso um valor muito alto for estabelecido, um provável efeito seria a definição dos IUL para uma faixa de potência maior do que a que viria a ser determinada pelo PDE, após a escolha – pelo procedimento aproximativo – das unidades a serem cometidas. No entanto, isto só consistiria em um problema grave se o caso particular do problema causasse interferência na ordem de classificação dos IUL, o que pode ocorrer para funções de custo muito diferentes entre as unidades. Todavia, o valor definido possibilitou que a solução encontrada fosse significativamente melhor do que a anterior, e melhor do que a solução parcial obtida para 3 horas de processamento da enumeração exaustiva.

4.3 Critério de desempate para os IUL

O critério de desempate definido no Capítulo 4 depende de um parâmetro μ para definir uma proximidade entre os IUL que fará com que a lista de prioridades classifique as unidades empatadas em função do custo operacional unitário.

Foram realizados testes para medir a quantidade de desempates gerados para diversos valores de μ . O valor definido para o parâmetro foi 0,0025, o que equivale dizer que o intervalo de proximidade é igual ao valor da demanda dividido por 400.

O intervalo definido é relativamente pequeno, pois não se deseja prejudicar a classificação obtida pelos IUL que, indiretamente, também consideram os custos operacionais através do PDE. No entanto, foi possível observar que em algumas situações de empate, a diferença do custo unitário de fato ocorreu, o que justifica o uso deste critério de classificação para as unidades com IUL próximos.

4.4 Resultados para a versão final do método proposto

Os resultados apresentados nesta seção consideram, além do aprimoramento da lista de prioridades, o ajuste automático do parâmetro β , definindo a versão final do método proposto neste trabalho. Desta forma, os únicos parâmetro que podem, opcionalmente, ser definidos pelo usuário são γ e o tempo máximo de processamento permitido. Caso o último parâmetro seja omitido, o algoritmo pára quando encontra a solução considerada ótima.

A Tabela 10 mostra os resultados finais obtidos para um caso de 13 geradores do PCU.

Tabela 10: Resultados finais do PCU para um caso de 13 geradores

D	X_B	Z_B	Tempo
560	560,0,0,0,0,0,0,0,0,0,0,0,0	5173,8386	124
1000	665,0,334,0,0,0,0,0,0,0,0,0,0	9143,7762	152
2000	680,355,360,155,0,146,136,0,166,0,0,0,0	18679,0479	190

Nota-se que conforme a demanda diminui, menos unidades são selecionadas pelo método. Entretanto, este comportamento não é compartilhado pelo método de PDE, que apenas distribui as necessidades de potência para unidades constantes no sistema.

Para ilustrar este fato, pode-se tomar os resultados para o mesmo sistema, desta vez solucionado pelo PDE, conforme a Tabela 11.

Tabela 11: Resultados do PDE para um caso de 13 geradores

D	X _B	Z _B	Tempo
560	2,4,1,60,60,60,60,60,60,40,40,55,55	7708,0419	20
1000	173,124,127,71,69,63,60,60,60,40,40,55,55	11299,7939	44
2000	600,311,254,99,111,97,98,110,125,40,40,55,55	19616,8725	43

O tempo de resolução do PDE, constante na Tabela 11, é menor do que o necessário ao PCU, pois este último deve resolver algumas combinações do PDE antes de chegar à solução ótima. Para os casos da Tabela 10, a média foi de 5 resoluções do PDE, dentre elas uma constituindo a resolução do problema para fins do procedimento aproximativo. No entanto, os PDE que devem ser resolvidos durante a execução do PCU possuem dimensão menor do que o PDE inicial, que considera todas as unidades ativadas. Isto justifica o fato do tempo de execução do PCU não ser um múltiplo exato do número de combinações do PDE, embora o tempo gasto na resolução dos PDE seja o mais determinante para o tempo total do PCU.

A Tabela 12 mostra os resultados finais para um caso do PCU com 6 geradores. Neste exemplo, foi considerado dois valores de demanda abaixo da soma dos limites mínimos – definidos como 345, de acordo com a Tabela 12.

Tabela 12: Resultados finais do PCU para um caso de 6 geradores

D	X _B	Z _B	Tempo
100	0,0,0,0,130,0	6737,9452	58
200	0,0,0,0,200,0	9768,5855	66
350	0,0,0,0,183,166	17263,0659	75
500	0,0,0,0,256,243	24108,2366	95
1000	0,0,186,174,324,314	49408,5119	57

Observa-se que, para o valor de demanda igual a 100, o melhor resultado obtido foi para o cometimento de uma unidade apenas, sendo que o limite inferior desta unidade (conforme o Anexo A) é 130. Isto aconteceu pois os casos para a demanda abaixo de 345 foram testados com uma mudança na condição da Equação 17 de “igual” para “maior ou igual”. Isto proporcionou uma região factível maior, de modo que minimização do custo pôde ser efetuada com maior flexibilidade. Isto possibilitou que uma solução cuja potência fornecida é maior do que a demanda e, mesmo assim, possui um custo operacional menor.

A Tabela 13 mostra os valores obtidos para o mesmo caso do problema solucionado pelo PDE.

Tabela 13: Resultados do PDE para um caso de 6 geradores

D	X _B	Z _B	Tempo
350	10,10,35,40,130,125	20572,1259	12
500	16,10,64,78,169,160	27007,0059	14
1000	35,17,167,152,311,314	50366,5344	20

A Tabela 13 não apresenta as soluções para os casos de demanda iguais a 100 e 200, pois o PDE não é capaz de definir uma utilização de potência menor do que a soma dos limites mínimos de todas as unidades do sistema. Isto acontece porque, ao contrário do PCU, o PDE não pode desativar as unidades existentes para encontrar a melhor solução. Em um problema real, onde a demanda fosse menor que a soma dos limites mínimos, um desperdício potência e capital ocorreria, pois a mínima potência possível de ser produzida no PDE é igual à soma dos limites mínimos.

É possível identificar os casos para os quais o PCU melhora as soluções em maior proporção. À medida que o coeficiente c , da curva de custo, é maior – o que indica um maior custo fixo para manter a unidade ligada – maior tende a ser a diminuição de custo do PCU em relação a o PDE. Esta relação se repete quando os limites mínimos dos geradores são aumentados, quando a demanda é próxima dos limites mínimos ou quando o número de unidades é aumentado.

Nos casos considerados até agora, nesta seção, o aumento das unidades causou maior diferença entre os resultados dos dois problemas, pois as condições definidas para os dois casos – tais como limites mínimos, limites máximos e função de custo - são diferentes.

Para demonstrar que o aumento das unidades pode comprometer o desempenho do PDE, mas não o do PCU, basta considerar um caso de 12 unidades constituído pelo duplicamento do caso de 6 unidades utilizado anteriormente, o que garante que as condições do problema sejam mantidas. Os resultados para o PDE e para o PCU são mostrados na Tabela 14.

Tabela 14: Resultados do PDE e PCU para um caso de 12 geradores

Problema	D	X _B	Z _B	Tempo
PDE	1000	20,10,62,72,186,153,19,10,63,73,173,154	54012,7401	51
PCU	1000	0,0,0,0,261,236,0,0,0,0,261,240	48219,5358	214

De início, pode-se constatar que o PDE não é capaz de resolver os casos de demanda igual a 350 e 500, pois a nova soma dos limites mínimos é igual a 690. Logo, foi considerado apenas o caso para demanda igual a 1000.

Analisando a Tabela 14, percebe-se que enquanto o PDE foi prejudicado pela nova restrição da soma limites mínimos, o PCU foi capaz de melhorar a solução a partir das novas possibilidades de combinação oferecidas.

Embora estas comparações sejam feitas, não é esperado que o resultado do PDE seja melhor do que o obtido para o PCU, já que este último deve, em princípio, possuir a capacidade de selecionar a melhor combinação de unidades a ser utilizada. Em suma, o resultado do PCU é o resultado de um PDE para um grupo seletivo de unidades e, portanto, depende da capacidade deste último de oferecer uma boa solução.

5 Conclusão

O PCU é um problema complexo e o método proposto foi capaz de solucionar uma versão que contou com algumas simplificações do modelo original. Por meio de uma revisão bibliográfica diversificada, foi possível apontar o método BB como o mais ideal para atender ao objetivo de utilizar uma solução já existente para o PDE, de modo que a atenção pudesse ser voltada às questões específicas do PCU. A integração dos dois métodos não apresentou nenhum problema de compatibilidade.

O método BB proposto contou com várias heurísticas que levaram a ganhos de eficiência e precisão. O modelo genérico de BB utilizado como base foi enriquecido com a aplicação de uma lista de prioridades e com a adoção da lógica do *unit decommitment*.

Durante o desenvolvimento do método fez-se necessário a criação de diversos parâmetros, para os quais se executou uma calibração a fim de que os melhores valores pudessem ser encontrados. Observou-se que os parâmetros executavam consideráveis mudanças sobre os resultados finais.

Ao final da calibração dos parâmetros, o algoritmo foi capaz de resolver vários casos do problema com rapidez e precisão, quando comparados à busca exaustiva e ao PDE considerando todas as unidades no sistema.

O algoritmo de BB também demonstrou bons resultados quanto à utilização de recursos computacionais, a exemplo do tempo de processamento e memória.

Por fim, os objetivos deste trabalho foram atingidos. Embora o método proposto esteja longe de atender à formulação mais completa do PCU, o mesmo constituiu uma primeira etapa neste sentido.

REFERÊNCIAS

- BALDICK, R. **The generalized unit commitment problem.** IEEE Trans. Power Syst., vol. 10, pp. 465–475, Feb. 1995.
- CARRIÓN, M.; ARROYO, J. M. **A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem.** IEEE Transactions On Power Systems, Vol. 21, No. 3, August 2006.
- COHEN, A. I.; YOSHIMURA, M. **A branch- and -bound algorithm for unit commitment.** IEEE Trans. Power App. Syst., vol. PAS-102, pp. 444–451, Feb. 1983.
- DAMOUSIS, Ioannis G.; BAKIRTZIS, Anastasios G. **A solution to the unit-commitment problem using integer-coded genetic algorithm.** IEEE Transactions on Power Systems, Vol. 19, No. 2, May 2004.
- DIEU, Vo Ngoc; ONGSAKUL, Weerakorn. **Enhanced merit order and augmented lagrange hopfield network for unit commitment.** In: Power Systems Computation Conference, 15., 2005, Liege, Bélgica. Disponível em: <<http://www.montefiore.ulg.ac.be/services/stochastic/pscc05/papers/fp414.pdf>>. Acesso em: 2 mar. 2007.
- GUAN, X.; LUH, P. B.; YAN, H. **An optimization based method for unit commitment.** Int. J. Elect. Power Energy Syst., vol. 14, no. 1, pp. 9–17, Feb. 1992.
- HUANG, K. Y.; YANG, H. T.; HUANG, C. L. **A new thermal unit commitment approach using constraint logic programming.** IEEE Trans. Power Syst., vol. 13, pp. 936–945, Aug. 1998.
- HUSSAIN, K. **Solution method for unit commitment limitations and utility constraints.** IEEE Comput. Appl. Power, vol. 4, pp. 16–20, Jan. 1991.
- JENSEN, A. Paul; BARD, F. Jonathan. **Operations research: models and methods.** John Wiley & Sons, Inc. 2003.
- JUSTE, K. A.; KITA, H.; TANAKA E.; HASEGAWA, J. **An evolutionary programming solution to the unit commitment problem.** IEEE Trans. Power Syst., vol. 14, pp. 1452–1459, Nov. 1999.

KAZARLIS, S. A.; BAKIRTZIS, A. G.; PETRIDIS, V. **A genetic algorithm solution to the unit commitment problem**. IEEE Trans. Power Syst., vol. 11, pp. 83–92, Feb. 1996.

LAUER, G. S.; SANDELL JR., N. R.; BERTSEKAS, N. R.; POSBERGH, T. A. **Solution Of large scale optimal unit commitment problems**. IEEE Trans. Power App. Syst., vol. PAS-101, pp. 79–96, Jan. 1982.

LEE, F. N. **Short-term unit commitment - A new method**. IEEE Transactions on Power Systems, Vol.3, No.2, pp.421-428, 1988.

LI, C.; JOHNSON, R. B.; SVABODA A. F. **A new unit commitment method**. IEEE Trans. Power Syst., vol. 12, pp. 113–119, Feb. 1997.

LINDO Systems. **Electrical generation unit commitment planning**. Application Survey Paper, Jun, 2003. Disponível em: <www.lindo.com/library/unitcomt1.pdf>. Acesso em: 2 mar. 2007.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. **A simulated annealing algorithm for unit commitment**. IEEE Trans. Power Syst., vol. 13, pp. 197–204, Feb. 1998.

PADHY, Narayana Prasad. **Unit commitment - a bibliographical survey**. IEEE transactions on power systems, vol. 19, no. 2, May 2004

RAJAN, C. C. A.; MOHAN, M. R. **An evolutionary programming based tabu search method for solving the unit commitment problem in utility system**. IEEE Transactions on Power Systems, Vol.19, No.1, pp.577-585, 2004.

RODRIGUES, Nátalli M. **Um Algoritmo Cultural para Problemas de Despacho**. 2007. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, UEM, Maringá.

SAMED, Márcia M. A. **Algoritmo Genético Híbrido Co-Evolutivo para Resolver Problemas de Despacho**. 2004. Tese (Doutorado em Engenharia Química) - Universidade Estadual de Maringá, UEM, Maringá.

SAKSORNCHAI, Titti; LEE, Wei-Jen; METHAPRAYOON, Kittipong; LIAO, James R.; ROSS, Richard J. **Improve the Unit Commitment Scheduling by Using the Neural-**

Network-Based Short-Term Load Forecasting. IEEE Transactions on Industry Applications, Vol. 41, No. 1, Jan/Feb, 2005.

SENJYU, Tomonobu; SHIMABUKURO, Kai; UEZATO, Katsumi; FUNABASHI, Toshihisa. **A fast technique for unit commitment problem by extended priority list.** IEEE Transactions on Power Systems, Vol. 18, No. 2, May 2003.

SHEBLE', Gerald B.; FAHD, George N. **Unit commitment literature synopsis.** IEEE Transactions on Power Systems, Vol. 9, No. 1, February 1994.

TSENG, Chung-Li; LIZ, Chao-an; ORENX Shmuel S. **Electronic publishing at PSERC: Solving unit commitment by a unit decommitment method.** Power Systems Engineering Research Center, 1997. Disponível em: < <http://www.pserc.org/ecow/get/publicatio/1997public/OREN2.pdf> >. Acesso em: 2 mar. 2007.

VEMURI, S.; LEMONIDIS, L. **Fuel constrained unit commitment.** IEEE Trans. Power Syst., vol. 7, pp. 410–415, Feb. 1991.

VOORSPOOLS, K.; D'HAESELEER, W. **Long term unit commitment optimisation for large power systems; unit decommitment versus advanced priority listing.** ENERGEX'2002 Conference, Krakow, Poland, May 19-24, 2002.

VOORSPOOLS, K.; D'HAESELEER, W. **The influence of the instantaneous fuel mix for electricity generation on the corresponding emissions.** Energy, No.25, pp.1119-1138, 2000.

WINSTON, L. Wayne. **Operations research: applications and algorithms.** Wadsworth Publishing Company. 1997.

WOOD, J. Allen; WOLLENBERG, F. Bruce. **Power generation: operation & control.** John Wiley & Sons. 1984.

APÊNDICE A – Tabelas de resultados da calibração de parâmetros

Tabela: Resultados para o teste de calibração de parâmetros com 6 geradores
(ordenados tempo)

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox.	Tempo
550	0	0,85	0	0,0,0,109,227,213	27203	4	1	61
550	0	0,85	-0,04	0,0,0,106,225,218	27204	4	1	69
550	0,6	0,9	0	0,0,0,111,221,216	27205	5	1	70
550	0	0,95	0,04	0,0,0,102,231,216	27205	5	1	70
550	0,4	0,85	0,04	0,0,0,114,231,204	27206	5	1	73
550	0,4	1	0	0,0,0,0,277,272	26487	5	1	74
550	0,6	0,9	-0,04	0,0,0,0,279,270	26487	5	1	75
550	0,4	0,9	-0,04	0,0,0,107,224,217	27204	5	1	75
550	0,6	0,9	0,04	0,0,0,111,229,209	27204	5	1	75
550	0	1	0,04	0,0,136,135,278,0	27561	5	1	76
550	0	1	0	0,0,0,0,278,271	26486	5	1	79
550	0,6	0,95	-0,04	0,0,0,0,282,267	26487	5	1	79
550	0	0,9	-0,04	0,0,0,104,229,215	27204	5	1	79
550	0,6	0,85	0	0,0,0,0,275,274	26487	5	1	80
550	0,9	0,85	0	0,0,76,90,199,183	28037	5	1	80
550	0,4	0,9	0,04	0,0,0,0,275,274	26487	5	1	81
550	0,6	0,95	0	0,0,0,104,226,219	27204	5	1	81
550	0,9	0,9	0	0,0,82,84,198,183	28038	5	1	81
550	0,4	0,85	0	0,0,137,137,275,0	27561	5	1	82
550	0,6	0,85	-0,04	0,0,0,108,227,213	27203	5	1	84
550	0	0,95	-0,04	0,0,0,104,229,216	27204	5	1	84
550	0,4	0,85	-0,04	0,0,0,104,228,216	27205	5	1	84
550	0,9	1	0	0,0,81,91,192,183	28038	5	1	84
550	0,9	0,95	0	0,0,74,96,196,182	28038	5	1	84
550	0,6	1	0,04	0,0,0,110,225,214	27204	5	1	85
550	0,4	0,95	0,04	0,0,0,106,223,219	27205	5	1	85
550	0	0,9	0,04	0,0,0,101,235,213	27206	5	1	85
550	0	1	-0,04	0,0,134,136,279,0	27561	5	1	87
550	0,6	0,85	0,04	0,0,0,109,231,209	27204	5	1	88
550	0,9	0,9	-0,04	0,0,76,93,197,182	28036	5	1	88
550	0,6	0,95	0,04	0,0,0,0,271,278	26488	5	1	89
550	0,6	1	0	0,0,0,109,228,212	27203	5	1	89
550	0,6	1	-0,04	0,0,0,110,226,212	27204	5	1	89
550	0,4	0,9	0	0,0,0,108,229,212	27204	5	1	89
550	0,9	0,95	-0,04	0,0,72,89,206,181	28038	5	1	89
550	0	0,9	0	0,0,0,106,230,213	27204	5	1	90
550	0,4	0,95	-0,04	0,0,135,139,275,0	27561	5	1	90
550	0,4	0,95	0	0,0,0,0,281,268	26487	5	1	93
550	0,9	1	-0,04	0,0,80,85,201,183	28038	5	1	93
550	0,9	0,85	-0,04	0,0,78,90,202,178	28036	5	1	95
550	0	0,95	0	0,0,0,105,231,212	27204	5	1	96
550	0	0,85	0,04	0,0,0,107,227,214	27203	5	1	100
550	0,4	1	-0,04	0,0,0,110,228,210	27204	5	1	103
550	0,4	1	0,04	0,0,0,106,229,213	27204	5	1	109
550	0,9	0,95	0,04	0,0,0,0,276,273	26486	18	6	240
550	0,9	0,85	0,04	0,0,0,0,279,270	26487	18	6	249
550	0,9	0,9	0,04	0,0,0,0,280,269	26487	18	6	260

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox.	Tempo
550	0,9	1	0,04	0,0,0,0,281,268	26487	18	6	271
550	0,4	0,85	0,08	0,0,0,0,276,273	26486	45	15	570
550	0,6	0,85	0,08	0,0,0,0,277,272	26486	47	16	576
550	0	0,95	0,08	0,0,0,0,277,272	26486	47	16	582
550	0,4	0,9	0,08	0,0,0,0,278,271	26486	47	16	593
550	0,4	0,95	0,08	0,0,0,0,277,272	26486	47	16	598
550	0	1	0,08	0,0,0,0,277,272	26487	47	16	602
550	0	0,9	0,08	0,0,0,0,277,272	26486	47	16	603
550	0,4	1	0,08	0,0,0,0,277,272	26486	47	16	607
550	0,6	0,9	0,08	0,0,0,0,277,272	26486	47	16	607
550	0	0,85	0,08	0,0,0,0,278,271	26486	46	16	608
550	0,9	0,85	0,08	0,0,0,0,280,269	26487	47	16	623
550	0,6	1	0,08	0,0,0,0,276,273	26487	47	16	629
550	0,6	0,95	0,08	0,0,0,0,277,272	26486	47	16	634
550	0,9	1	0,08	0,0,0,0,278,271	26486	47	16	644
550	0,9	0,9	0,08	0,0,0,0,276,273	26487	47	16	652
550	0,9	0,95	0,08	0,0,0,0,278,271	26486	47	16	671
1000	0,4	0,85	0	0,0,185,174,324,315	49408	5	1	69
1000	0,4	1	0	0,0,184,176,324,313	49409	5	1	73
1000	0,9	0,95	0	0,0,181,180,322,314	49411	5	1	74
1000	0,9	0,9	0	0,0,191,170,323,314	49412	5	1	74
1000	0,4	0,95	-0,04	0,0,186,173,325,315	49409	5	1	75
1000	0	1	-0,04	0,0,189,170,324,314	49410	5	1	75
1000	0,9	0,85	0	0,0,180,179,325,314	49409	5	1	77
1000	0,9	1	-0,04	0,0,186,173,324,314	49408	5	1	79
1000	0,4	0,95	0	0,0,186,173,324,314	49408	5	1	80
1000	0,9	1	0	0,0,182,178,323,315	49409	5	1	80
1000	0,6	1	-0,04	0,0,182,177,324,315	49408	5	1	81
1000	0,6	0,9	0	0,0,179,180,324,314	49409	5	1	81
1000	0,6	0,95	-0,04	0,0,188,173,323,315	49410	5	1	81
1000	0	1	0	0,0,186,175,325,313	49410	5	1	82
1000	0,6	0,95	0	0,0,186,174,324,314	49409	5	1	83
1000	0,6	0,9	-0,04	0,0,184,177,324,313	49409	5	1	84
1000	0,4	1	-0,04	0,0,176,183,324,314	49413	5	1	84
1000	0	0,95	-0,04	0,0,189,171,324,315	49409	5	1	87
1000	0	0,9	-0,04	0,0,190,169,324,315	49410	5	1	87
1000	0	0,9	0	0,0,182,177,325,315	49408	5	1	88
1000	0	0,95	0	0,0,184,178,322,315	49410	5	1	89
1000	0,6	0,85	0	0,0,184,175,324,315	49407	5	1	90
1000	0,9	0,9	-0,04	0,0,186,174,324,314	49409	5	1	90
1000	0,6	1	0	0,0,183,178,323,314	49409	5	1	90
1000	0,4	0,9	0	0,0,180,180,324,315	49410	5	1	90
1000	0	0,85	0	0,0,185,174,324,314	49408	5	1	91
1000	0,4	0,9	-0,04	0,0,185,174,324,314	49408	5	1	96
1000	0	0,85	-0,04	0,0,192,169,323,314	49412	5	1	96
1000	0,9	0,95	-0,04	0,0,182,177,325,315	49408	5	1	100
1000	0,4	0,85	-0,04	0,0,178,181,325,315	49412	9	1	153
1000	0,6	0,85	-0,04	0,0,185,174,324,314	49408	9	1	157
1000	0,9	0,85	-0,04	0,0,188,171,325,314	49410	9	1	169
1000	0,4	0,85	0,04	0,0,183,176,325,315	49408	100	41	1536

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox.	Tempo
1000	0,9	0,85	0,04	0,0,182,177,324,315	49408	100	41	1616
1000	0,6	0,85	0,04	0,0,186,173,325,315	49408	102	42	1623
1000	0	0,85	0,04	0,0,184,175,324,314	49408	102	42	1655
1000	0,9	0,9	0,04	0,0,185,174,325,315	49408	121	45	1876
1000	0,6	0,9	0,04	0,0,185,174,325,314	49407	121	45	1915
1000	0,4	0,95	0,04	0,0,183,176,325,315	49408	121	45	1922
1000	0,9	1	0,04	0,0,186,173,325,315	49408	121	45	1925
1000	0,6	0,95	0,04	0,0,184,175,324,315	49408	121	45	1940
1000	0,9	0,95	0,04	0,0,183,176,324,315	49408	123	46	1940
1000	0,4	0,9	0,08	0,0,183,176,324,315	49408	123	46	1946
1000	0,6	1	0,08	0,0,183,176,324,315	49408	123	46	1951
1000	0,4	0,95	0,08	0,0,182,177,325,314	49408	123	46	1952
1000	0	1	0,04	0,0,186,173,324,315	49408	121	45	1957
1000	0,9	0,95	0,08	0,0,185,174,324,314	49408	123	46	1962
1000	0,4	1	0,04	0,0,184,175,324,314	49407	121	45	1966
1000	0,9	0,9	0,08	0,0,185,174,325,314	49408	123	46	1973
1000	0,6	0,95	0,08	0,0,184,175,325,315	49408	123	46	1978
1000	0,4	0,9	0,04	0,0,185,174,325,315	49407	121	45	1980
1000	0,4	0,85	0,08	0,0,182,177,324,315	49408	123	46	1985
1000	0	0,95	0,08	0,0,183,176,324,315	49408	123	46	1986
1000	0,9	1	0,08	0,0,184,175,324,314	49408	123	46	1997
1000	0,9	0,85	0,08	0,0,184,175,325,315	49408	123	46	1998
1000	0,4	1	0,08	0,0,185,174,325,314	49408	123	46	2005
1000	0,6	1	0,04	0,0,184,175,324,314	49407	121	45	2012
1000	0	1	0,08	0,0,185,174,325,315	49407	123	46	2014
1000	0	0,9	0,04	0,0,184,175,325,315	49408	121	45	2026
1000	0	0,9	0,08	0,0,185,174,324,314	49408	123	46	2026
1000	0	0,85	0,08	0,0,183,176,325,314	49408	123	46	2037
1000	0	0,95	0,04	0,0,184,175,325,315	49408	121	45	2048
1000	0,6	0,9	0,08	0,0,183,176,324,315	49407	123	46	2055
1000	0,6	0,85	0,08	0,0,185,174,325,315	49408	123	46	2095

**Tabela: Resultados para o teste de calibração de parâmetros com 13 geradores
(ordenados por Z_B)**

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox	Tempo
550	0,4	1	0,08	0,0,0,0,277,272	26486	47	16	607
550	0	0,85	0,08	0,0,0,0,278,271	26486	46	16	608
550	0	0,9	0,08	0,0,0,0,277,272	26486	47	16	603
550	0,4	0,95	0,08	0,0,0,0,277,272	26486	47	16	598
550	0	1	0	0,0,0,0,278,271	26486	5	1	79
550	0	0,95	0,08	0,0,0,0,277,272	26486	47	16	582
550	0,4	0,9	0,08	0,0,0,0,278,271	26486	47	16	593
550	0,4	0,85	0,08	0,0,0,0,276,273	26486	45	15	570
550	0,9	1	0,08	0,0,0,0,278,271	26486	47	16	644
550	0,6	0,9	0,08	0,0,0,0,277,272	26486	47	16	607
550	0,9	0,95	0,04	0,0,0,0,276,273	26486	18	6	240
550	0,9	0,95	0,08	0,0,0,0,278,271	26486	47	16	671
550	0,6	0,95	0,08	0,0,0,0,277,272	26486	47	16	634
550	0,6	0,85	0,08	0,0,0,0,277,272	26486	47	16	576
550	0	1	0,08	0,0,0,0,277,272	26487	47	16	602
550	0,6	1	0,08	0,0,0,0,276,273	26487	47	16	629
550	0,9	0,85	0,04	0,0,0,0,279,270	26487	18	6	249
550	0,9	0,9	0,04	0,0,0,0,280,269	26487	18	6	260
550	0,4	1	0	0,0,0,0,277,272	26487	5	1	74
550	0,9	0,85	0,08	0,0,0,0,280,269	26487	47	16	623
550	0,4	0,9	0,04	0,0,0,0,275,274	26487	5	1	81
550	0,9	1	0,04	0,0,0,0,281,268	26487	18	6	271
550	0,9	0,9	0,08	0,0,0,0,276,273	26487	47	16	652
550	0,6	0,9	-0,04	0,0,0,0,279,270	26487	5	1	75
550	0,6	0,85	0	0,0,0,0,275,274	26487	5	1	80
550	0,4	0,95	0	0,0,0,0,281,268	26487	5	1	93
550	0,6	0,95	-0,04	0,0,0,0,282,267	26487	5	1	79
550	0,6	0,95	0,04	0,0,0,0,271,278	26488	5	1	89
550	0,6	0,85	-0,04	0,0,0,108,227,213	27203	5	1	84
550	0	0,85	0	0,0,0,109,227,213	27203	4	1	61
550	0	0,85	0,04	0,0,0,107,227,214	27203	5	1	100
550	0,6	1	0	0,0,0,109,228,212	27203	5	1	89
550	0,6	1	-0,04	0,0,0,110,226,212	27204	5	1	89
550	0	0,9	-0,04	0,0,0,104,229,215	27204	5	1	79
550	0,4	1	0,04	0,0,0,106,229,213	27204	5	1	109
550	0,4	0,9	0	0,0,0,108,229,212	27204	5	1	89
550	0,4	1	-0,04	0,0,0,110,228,210	27204	5	1	103
550	0,4	0,9	-0,04	0,0,0,107,224,217	27204	5	1	75
550	0	0,95	0	0,0,0,105,231,212	27204	5	1	96
550	0,6	1	0,04	0,0,0,110,225,214	27204	5	1	85
550	0	0,9	0	0,0,0,106,230,213	27204	5	1	90
550	0,6	0,85	0,04	0,0,0,109,231,209	27204	5	1	88
550	0	0,95	-0,04	0,0,0,104,229,216	27204	5	1	84
550	0,6	0,95	0	0,0,0,104,226,219	27204	5	1	81
550	0,6	0,9	0,04	0,0,0,111,229,209	27204	5	1	75
550	0	0,85	-0,04	0,0,0,106,225,218	27204	4	1	69
550	0,4	0,95	0,04	0,0,0,106,223,219	27205	5	1	85

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox	Tempo
550	0,6	0,9	0	0,0,0,111,221,216	27205	5	1	70
550	0,4	0,85	-0,04	0,0,0,104,228,216	27205	5	1	84
550	0	0,95	0,04	0,0,0,102,231,216	27205	5	1	70
550	0	0,9	0,04	0,0,0,101,235,213	27206	5	1	85
550	0,4	0,85	0,04	0,0,0,114,231,204	27206	5	1	73
550	0	1	-0,04	0,0,134,136,279,0	27561	5	1	87
550	0,4	0,85	0	0,0,137,137,275,0	27561	5	1	82
550	0	1	0,04	0,0,136,135,278,0	27561	5	1	76
550	0,4	0,95	-0,04	0,0,135,139,275,0	27561	5	1	90
550	0,9	0,85	-0,04	0,0,78,90,202,178	28036	5	1	95
550	0,9	0,9	-0,04	0,0,76,93,197,182	28036	5	1	88
550	0,9	0,85	0	0,0,76,90,199,183	28037	5	1	80
550	0,9	1	0	0,0,81,91,192,183	28038	5	1	84
550	0,9	0,95	-0,04	0,0,72,89,206,181	28038	5	1	89
550	0,9	1	-0,04	0,0,80,85,201,183	28038	5	1	93
550	0,9	0,95	0	0,0,74,96,196,182	28038	5	1	84
550	0,9	0,9	0	0,0,82,84,198,183	28038	5	1	81
1000	0,4	0,9	0,04	0,0,185,174,325,315	49407	121	45	1980
1000	0,6	1	0,04	0,0,184,175,324,314	49407	121	45	2012
1000	0	1	0,08	0,0,185,174,325,315	49407	123	46	2014
1000	0,6	0,9	0,08	0,0,183,176,324,315	49407	123	46	2055
1000	0,6	0,85	0	0,0,184,175,324,315	49407	5	1	90
1000	0,4	1	0,04	0,0,184,175,324,314	49407	121	45	1966
1000	0,6	0,9	0,04	0,0,185,174,325,314	49407	121	45	1915
1000	0,9	0,9	0,08	0,0,185,174,325,314	49408	123	46	1973
1000	0	0,9	0,04	0,0,184,175,325,315	49408	121	45	2026
1000	0	0,85	0,04	0,0,184,175,324,314	49408	102	42	1655
1000	0,9	0,9	0,04	0,0,185,174,325,315	49408	121	45	1876
1000	0,9	0,85	0,08	0,0,184,175,325,315	49408	123	46	1998
1000	0,6	1	0,08	0,0,183,176,324,315	49408	123	46	1951
1000	0	0,95	0,08	0,0,183,176,324,315	49408	123	46	1986
1000	0	0,9	0,08	0,0,185,174,324,314	49408	123	46	2026
1000	0,6	0,95	0,08	0,0,184,175,325,315	49408	123	46	1978
1000	0,6	0,95	0,04	0,0,184,175,324,315	49408	121	45	1940
1000	0,9	1	0,04	0,0,186,173,325,315	49408	121	45	1925
1000	0,9	0,95	0,04	0,0,183,176,324,315	49408	123	46	1940
1000	0,9	0,95	0,08	0,0,185,174,324,314	49408	123	46	1962
1000	0	1	0,04	0,0,186,173,324,315	49408	121	45	1957
1000	0,4	0,95	0,04	0,0,183,176,325,315	49408	121	45	1922
1000	0,6	0,85	0,04	0,0,186,173,325,315	49408	102	42	1623
1000	0,6	0,85	0,08	0,0,185,174,325,315	49408	123	46	2095
1000	0,4	0,85	0,04	0,0,183,176,325,315	49408	100	41	1536
1000	0	0,95	0,04	0,0,184,175,325,315	49408	121	45	2048
1000	0,4	0,95	0,08	0,0,182,177,325,314	49408	123	46	1952
1000	0,4	0,85	0	0,0,185,174,324,315	49408	5	1	69
1000	0	0,85	0,08	0,0,183,176,325,314	49408	123	46	2037
1000	0,9	0,85	0,04	0,0,182,177,324,315	49408	100	41	1616
1000	0,4	0,9	0,08	0,0,183,176,324,315	49408	123	46	1946
1000	0,9	1	0,08	0,0,184,175,324,314	49408	123	46	1997
1000	0,9	0,95	-0,04	0,0,182,177,325,315	49408	5	1	100

D	β	α	γ	X_B	Z_B	Nº total de resoluções do PDE	Nº de resoluções do PDE no proced. aprox	Tempo
1000	0,4	0,85	0,08	0,0,182,177,324,315	49408	123	46	1985
1000	0,4	1	0,08	0,0,185,174,325,314	49408	123	46	2005
1000	0	0,9	0	0,0,182,177,325,315	49408	5	1	88
1000	0,6	1	-0,04	0,0,182,177,324,315	49408	5	1	81
1000	0,6	0,85	-0,04	0,0,185,174,324,314	49408	9	1	157
1000	0	0,85	0	0,0,185,174,324,314	49408	5	1	91
1000	0,4	0,95	0	0,0,186,173,324,314	49408	5	1	80
1000	0,9	1	-0,04	0,0,186,173,324,314	49408	5	1	79
1000	0,4	0,9	-0,04	0,0,185,174,324,314	49408	5	1	96
1000	0,6	0,95	0	0,0,186,174,324,314	49409	5	1	83
1000	0,4	0,95	-0,04	0,0,186,173,325,315	49409	5	1	75
1000	0,9	0,9	-0,04	0,0,186,174,324,314	49409	5	1	90
1000	0,9	1	0	0,0,182,178,323,315	49409	5	1	80
1000	0,6	1	0	0,0,183,178,323,314	49409	5	1	90
1000	0,4	1	0	0,0,184,176,324,313	49409	5	1	73
1000	0	0,95	-0,04	0,0,189,171,324,315	49409	5	1	87
1000	0,6	0,9	-0,04	0,0,184,177,324,313	49409	5	1	84
1000	0,9	0,85	0	0,0,180,179,325,314	49409	5	1	77
1000	0,6	0,9	0	0,0,179,180,324,314	49409	5	1	81
1000	0	1	0	0,0,186,175,325,313	49410	5	1	82
1000	0,9	0,85	-0,04	0,0,188,171,325,314	49410	9	1	169
1000	0	1	-0,04	0,0,189,170,324,314	49410	5	1	75
1000	0,4	0,9	0	0,0,180,180,324,315	49410	5	1	90
1000	0	0,95	0	0,0,184,178,322,315	49410	5	1	89
1000	0	0,9	-0,04	0,0,190,169,324,315	49410	5	1	87
1000	0,6	0,95	-0,04	0,0,188,173,323,315	49410	5	1	81
1000	0,9	0,95	0	0,0,181,180,322,314	49411	5	1	74
1000	0,9	0,9	0	0,0,191,170,323,314	49412	5	1	74
1000	0,4	0,85	-0,04	0,0,178,181,325,315	49412	9	1	153
1000	0	0,85	-0,04	0,0,192,169,323,314	49412	5	1	96
1000	0,4	1	-0,04	0,0,176,183,324,314	49413	5	1	84

**Tabela: Resultados para o teste de calibração de parâmetros com 13 geradores
(ordenados por Z_B)**

D	β	α	γ	X_B	Z_B	PDE Total	PDE Rel.	Tempo
1000	0,4	0,9	-0,04	450,247,203,0,97,0,0,0,0,0,0,0,0,0	9616	4	1	141
1000	0,4	0,95	-0,04	463,240,204,91,0,0,0,0,0,0,0,0,0	9616	5	1	133
1000	0,4	0,95	0	481,205,211,0,0,0,101,0,0,0,0,0,0	9617	5	1	140
1000	0,4	0,9	0	402,267,233,0,0,0,96,0,0,0,0,0,0	9618	5	1	129
1000	0,6	0,95	0	381,173,168,0,92,0,91,93,0,0,0,0,0	10043	5	1	176
1000	0,6	0,95	-0,04	360,166,184,84,94,0,108,0,0,0,0,0,0	10044	5	1	136
1000	0,6	0,9	0	347,185,167,0,0,0,100,94,104,0,0,0,0	10044	5	1	131
1000	0,6	0,9	-0,04	328,191,176,0,0,103,92,0,108,0,0,0,0	10045	5	1	157
1000	0,8	0,95	0	255,135,181,0,83,79,85,0,84,40,0,55,0	10556	5	1	179
1000	0,8	0,95	-0,04	226,126,130,78,97,71,88,90,90,0,0,0,0	10698	5	1	197
1000	0,8	0,9	-0,04	235,101,141,76,79,90,92,89,93,0,0,0,0	10699	5	1	171
1000	0,8	0,9	0	236,136,143,89,0,75,71,74,76,40,0,0,55	10775	5	1	173
2000	0,4	0,9	-0,04	680,359,360,160,0,138,0,145,155,0,0,0,0	18678	5	1	161
2000	0,6	0,9	0	678,360,359,132,0,0,155,154,160,0,0,0,0	18678	5	1	185
2000	0,4	0,95	-0,04	675,359,360,0,142,148,0,151,161,0,0,0,0	18679	5	1	204
2000	0,4	0,9	0	679,358,358,140,0,0,167,132,164,0,0,0,0	18680	5	1	141
2000	0,4	0,95	0	680,349,358,154,0,0,149,143,164,0,0,0,0	18680	5	1	145
2000	0,6	0,9	-0,04	669,360,359,0,142,156,158,0,153,0,0,0,0	18680	5	1	138
2000	0,6	0,95	-0,04	674,358,357,0,139,0,152,167,149,0,0,0,0	18680	5	1	147
2000	0,6	0,95	0	678,352,360,142,163,0,0,145,157,0,0,0,0	18680	5	1	168
2000	0,8	0,9	-0,04	676,353,359,0,138,159,152,0,160,0,0,0,0	18680	5	1	209
2000	0,8	0,9	0	679,348,357,121,111,127,0,123,128,0,0,0,0	18859	5	1	155
2000	0,8	0,95	0	679,352,357,134,114,121,122,0,117,0,0,0,0	18859	5	1	188
2000	0,8	0,95	-0,04	644,347,349,148,116,128,134,130,0,0,0,0,0	18864	5	1	157

**Tabela: Resultados para o teste de calibração de parâmetros com 13 geradores
(ordenados pelo tempo)**

D	β	α	γ	X_B	Z_B	PDE Total	PDE Rel.	Tempo
1000	0,4	0,9	0	402,267,233,0,0,0,0,96,0,0,0,0,0,0	9618	5	1	129
1000	0,6	0,9	0	347,185,167,0,0,0,100,94,104,0,0,0,0	10044	5	1	131
1000	0,4	0,95	-0,04	463,240,204,91,0,0,0,0,0,0,0,0,0	9616	5	1	133
1000	0,6	0,95	-0,04	360,166,184,84,94,0,108,0,0,0,0,0,0	10044	5	1	136
1000	0,4	0,95	0	481,205,211,0,0,0,101,0,0,0,0,0,0	9617	5	1	140
1000	0,4	0,9	-0,04	450,247,203,0,97,0,0,0,0,0,0,0,0	9616	4	1	141
1000	0,6	0,9	-0,04	328,191,176,0,0,103,92,0,108,0,0,0,0	10045	5	1	157
1000	0,8	0,9	-0,04	235,101,141,76,79,90,92,89,93,0,0,0,0	10699	5	1	171
1000	0,8	0,9	0	236,136,143,89,0,75,71,74,76,40,0,0,55	10775	5	1	173
1000	0,6	0,95	0	381,173,168,0,92,0,91,93,0,0,0,0,0	10043	5	1	176
1000	0,8	0,95	0	255,135,181,0,83,79,85,0,84,40,0,55,0	10556	5	1	179
1000	0,8	0,95	-0,04	226,126,130,78,97,71,88,90,90,0,0,0,0	10698	5	1	197
2000	0,6	0,9	-0,04	669,360,359,0,142,156,158,0,153,0,0,0,0	18680	5	1	138
2000	0,4	0,9	0	679,358,358,140,0,0,167,132,164,0,0,0,0	18680	5	1	141
2000	0,4	0,95	0	680,349,358,154,0,0,149,143,164,0,0,0,0	18680	5	1	145
2000	0,6	0,95	-0,04	674,358,357,0,139,0,152,167,149,0,0,0,0	18680	5	1	147
2000	0,8	0,9	0	679,348,357,121,111,127,0,123,128,0,0,0,0	18859	5	1	155
2000	0,8	0,95	-0,04	644,347,349,148,116,128,134,130,0,0,0,0,0	18864	5	1	157
2000	0,4	0,9	-0,04	680,359,360,160,0,138,0,145,155,0,0,0,0	18678	5	1	161
2000	0,6	0,95	0	678,352,360,142,163,0,0,145,157,0,0,0,0	18680	5	1	168
2000	0,6	0,9	0	678,360,359,132,0,0,155,154,160,0,0,0,0	18678	5	1	185
2000	0,8	0,95	0	679,352,357,134,114,121,122,0,117,0,0,0,0	18859	5	1	188
2000	0,4	0,95	-0,04	675,359,360,0,142,148,0,151,161,0,0,0,0	18679	5	1	204
2000	0,8	0,9	-0,04	676,353,359,0,138,159,152,0,160,0,0,0,0	18680	5	1	209

Tabela: Resultados para o teste de calibração de parâmetros com 13 geradores (com aprimoramento da lista de prioridades)

D	β	α	γ	X_B	Z_B	PDE Total	PDE Rel.	Tempo
1000	0,4	0,9	-0,04	678,321,0,0,0,0,0,0,0,0,0,0,0,0	9146	5	1	142
1000	0,4	0,95	-0,04	673,0,326,0,0,0,0,0,0,0,0,0,0,0	9144	6	1	177
1000	0,4	0,9	0	672,327,0,0,0,0,0,0,0,0,0,0,0,0	9146	6	1	165
1000	0,4	0,95	0	679,0,320,0,0,0,0,0,0,0,0,0,0,0	9144	6	1	183
1000	0,6	0,9	-0,04	654,0,345,0,0,0,0,0,0,0,0,0,0,0	9144	6	1	168
1000	0,6	0,95	-0,04	666,0,333,0,0,0,0,0,0,0,0,0,0,0	9144	6	1	171
1000	0,6	0,9	0	667,332,0,0,0,0,0,0,0,0,0,0,0,0	9146	6	1	177
1000	0,6	0,95	0	647,0,352,0,0,0,0,0,0,0,0,0,0,0	9145	6	1	201
1000	0,8	0,9	-0,04	490,250,259,0,0,0,0,0,0,0,0,0,0,0	9406	6	1	212
1000	0,8	0,95	-0,04	492,256,250,0,0,0,0,0,0,0,0,0,0,0	9406	6	1	165
1000	0,8	0,9	0	517,208,273,0,0,0,0,0,0,0,0,0,0,0	9407	6	1	212
1000	0,8	0,95	0	680,320,0,0,0,0,0,0,0,0,0,0,0,0	9146	6	1	154

**ANEXO A – DADOS DOS ARQUIVOS DE ENTRADA PARA OS CASOS
DE PDE E PCU RESOLVIDOS**

Tabela: Dados do arquivo de entrada para o caso de 6 geradores

<i>a</i>	<i>b</i>	<i>c</i>	lim mín	lim máx
10	125	0,15247	38,53973	756,7989
10	150	0,10587	46,15916	451,3251
35	225	0,02803	40,39655	1049,998
35	210	0,03546	38,30553	1243,531
130	325	0,02111	36,32782	1658,57
125	315	0,01799	38,27041	1356,659

Tabela: Dados do arquivo de entrada para o caso de 13 geradores

<i>a</i>	<i>b</i>	<i>c</i>	lim mín	lim máx
0	680	0,00028	8,1	550
0	360	0,00056	8,1	309
0	360	0,00056	8,1	307
60	180	0,00324	7,74	240
60	180	0,00324	7,74	240
60	180	0,00324	7,74	240
60	180	0,00324	7,74	240
60	180	0,00324	7,74	240
60	180	0,00324	7,74	240
40	120	0,00284	8,6	126
40	120	0,00284	8,6	126
55	120	0,00284	8,6	126
55	120	0,00284	8,6	126

**Universidade Estadual de Maringá
Departamento de Informática
Curso de Engenharia de Produção
Av. Colombo 5790, Maringá-PR
CEP 87020-900**

Tel: (044) 3261-4324 / 4219 Fax: (044) 3261-5874