

UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

DANIELA BRANDÃO NASCIMENTO

OTIMIZAÇÃO DA PROGRAMAÇÃO DE ORDENS DE CORTE EM INDÚSTRIAS
TÊXTEIS

FLORIANÓPOLIS
2006

DANIELA BRANDÃO NASCIMENTO

OTIMIZAÇÃO DA PROGRAMAÇÃO DE ORDENS DE CORTE EM INDÚSTRIAS
TÊXTEIS

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Doutor em Engenharia de Produção.

Orientador: Prof. João Neiva De Figueiredo, Ph.D.

FLORIANÓPOLIS
2006

DANIELA BRANDÃO NASCIMENTO

OTIMIZAÇÃO DA PROGRAMAÇÃO DE ORDENS DE CORTE EM INDÚSTRIAS
TÊXTEIS

Esta Tese foi julgada e aprovada para a obtenção do grau de **Doutor em Engenharia de Produção** no **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina.

Florianópolis, ____ de _____ de 2006.

Prof. Dr. Edson Pacheco Paladini
Coordenador do Programa

BANCA EXAMINADORA

Prof. Antônio Sérgio Coelho, Dr.
Universidade Federal de Santa Catarina
Presidente

Prof. Dálvio Ferrari Tubino, Dr.
Universidade Federal de Santa Catarina

Profa. Mirian Buss Gonçalves, Dra.
Universidade Federal de Santa Catarina

Prof. Sérgio Fernando Mayerle, Dr.
Universidade Federal de Santa Catarina

Prof. Rafael Carlos Vélez Benito, Dr.
Universidade do Vale do Itajaí

Prof. Ricardo Vilarroel Davalos, Dr.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Em primeiro lugar, a Deus, que me proporcionou todas as condições para que eu pudesse realizar esta pesquisa.

Ao meu marido, Norberto, pelo carinho, incentivo e compreensão, sem os quais não teria sido possível chegar até aqui.

Aos meus pais, Eliana e Paulo, e irmãos, Alessandra e Paulinho, pelo carinho e apoio que sempre souberam dar.

Ao meu orientador, professor João Neiva de Figueiredo, pela competente orientação prestada ao longo da pesquisa, bem como pela amizade e apoio com os quais sempre pude contar durante esses três anos e meio de convívio.

Ao meu pai, professor Paulo R. Nascimento, pelo grande auxílio prestado durante a realização do trabalho e por ter me servido de inspiração e exemplo, como ser humano e como profissional, ao longo de toda a minha vida.

Ao professor Sérgio Fernando Mayerle, por suas valiosas contribuições ao longo do trabalho e pelo grande auxílio prestado durante o curso, bem como por ter despertado em mim o interesse por Pesquisa Operacional, ainda à época do Mestrado.

Ao meu colega de Doutorado e amigo, Rafael Machado Casali, que sempre esteve disponível para me ajudar em tudo que fosse preciso e para discutir as principais questões acerca da elaboração da pesquisa. Sem ele, a realização deste trabalho teria se tornado muito mais difícil.

Ao professor Dálvio Ferrari Tubino, por ter me apresentado o tema da pesquisa e disponibilizado informações fundamentais para a realização da mesma.

Ao professor Clóvis C. Gonzaga, por sua valiosa contribuição na fase final da pesquisa.

Ao Claudinei Vieira e ao Adriano Grabowski, por terem me apresentado o processo de programação de ordens de corte na indústria têxtil, e por terem se mostrado sempre disponíveis para sanar as várias dúvidas que surgiram ao longo do trabalho.

À UNISUL, pela oportunidade de crescimento como professora e pesquisadora, bem como pelo auxílio financeiro ao longo da realização do trabalho. Especialmente à coordenadora do curso de Engenharia de Produção daquela Instituição, prof^a Ana Regina de Aguiar Dutra, por todo incentivo e amizade demonstrados ao longo dos três últimos anos.

Aos demais amigos e colegas que, de alguma forma, auxiliaram na concretização deste trabalho.

RESUMO

NASCIMENTO, Daniela Brandão. Otimização da programação de ordens de corte em indústrias têxteis. 2006. 216 f. **Tese...** (Doutorado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina. Florianópolis: UFSC, 2006.

Orientador: Prof. Dr. João Neiva De Figueiredo
Defesa: 07/07/06

Esta tese tem como objetivo geral conceber um método consistente para resolução do PPOC – Problema de Programação de Ordens de Corte, de forma a considerar todos os custos e restrições envolvidos neste processo decisório. O PPOC é um problema inerente às indústrias têxteis do segmento de confecção e consiste na definição do número de peças – para cada tamanho e cor associados a uma dada referência – que deverão ser cortadas visando atender a uma demanda prevista. Para alcançar o objetivo supracitado, formulou-se o PPOC como um problema de programação não-linear inteira, que foi resolvido através da teoria de grafos. Conceberam-se 17 diferentes algoritmos que envolvem procedimentos clássicos de busca, além de heurísticas inovadoras que visam reduzir o tempo de processamento para obtenção de resultados. Para avaliação da performance de cada um desses algoritmos, foram realizados testes com dados reais acerca de quatro referências de uma indústria do segmento de confecção, além de testes com 30 cenários de demanda criados para esta finalidade. Os resultados obtidos demonstraram que os métodos BVLC/BA – Busca Vertical com Limitação de Custo/ Busca Ávida e BLNALC/BA – Busca com Lista de Nós Abertos e Limitação de Custo/ Busca Ávida possibilitaram a obtenção de melhores resultados para o PPOC, comparativamente aos resultados que podiam ser obtidos anteriormente por aquela indústria. A obtenção de bons resultados para este problema pode propiciar redução nos custos de manutenção de estoques e armazenagem de peças, bem como redução na quantidade de peças que têm de ser vendidas a preços promocionais por excederem a demanda da indústria de confecção.

PALAVRAS-CHAVE: Indústria têxtil. Programação de ordens de corte. Busca em Grafos.

ABSTRACT

This dissertation presents a model to optimize the problem of spreading and cutting in garment manufacturing (PPOC) including all relevant costs and restrictions of this decision-making process. This problem is inherent to the garment manufacturing industry and consists of the definition of the number of garments of every size and color to be produced (i.e., cut) subject to a given demand level. The PPOC was formulated as a non-linear integer programming problem that was solved using graph theory. The graph's nodes and arcs are defined as well as a node successor operator. 17 different algorithms involving different permutations of classic search procedures were developed as well as innovative heuristic procedures which reduce processing time. To evaluate the performance of each of these algorithms tests on four actual industry references were performed with actual expected demand besides 30 others with hypothetical demand scenarios. The results show that the models BVLC/BA – Vertical Search with Cost Limit and Greedy Search and BLNALC/BA – Open Node List Search with Cost Limit and Greedy Search provided the best results, which compared very favorably to results currently obtained in the field. The results allow for a reduction in overall inventory maintenance costs as well as a reduction in the number of SKUs that must be sold by the apparel company at promotional prices.

KEYWORDS: Textile industry. Garment manufacturing, Spreading and cutting. Non-linear integer programming. Search in Graphs.

LISTA DE FIGURAS

FIGURA 01: Exportações têxteis e confecções	15
FIGURA 02: Segmentos do complexo têxtil	17
FIGURA 03: A cadeia produtiva têxtil.	18
FIGURA 04: Exemplo de uma faca de corte de peças	21
FIGURA 05: Exemplo de uma máquina de enfiar de tecido	22
FIGURA 06: Ilustração do problema de corte industrial bidimensional	35
FIGURA 07: Estrutura geral dos Problemas de Corte e Empacotamento	36
FIGURA 08: Estrutura de um algoritmo genético.....	60
FIGURA 09: Exemplo de um grafo.....	64
FIGURA 10: Processo de busca em largura.....	67
FIGURA 11: Processo de busca em profundidade	69
FIGURA 12: Fluxograma do algoritmo heurístico.....	81
FIGURA 13: Fluxograma do processo de expansão de nós.....	89
FIGURA 14: Fluxograma do algoritmo de busca vertical com limitação de custo	97
FIGURA 15: Representação do processo de busca vertical com limitação de custo	98
FIGURA 16: Fluxograma do algoritmo de busca ávida	101
FIGURA 17: Representação do processo de busca ávida.....	102
FIGURA 18: Fluxograma do algoritmo de busca vertical com limitação de custo/ busca ávida	105
FIGURA 19: Fluxograma do algoritmo com lista de nós abertos e limitação de custo	110
FIGURA 20: Representação do processo de busca com lista de nós abertos e limitação de custo	111
FIGURA 21: Fluxograma do algoritmo com lista de nós abertos e limitação de custo/ busca ávida.....	114

LISTA DE TABELAS

TABELA 01: Representação de uma solução gerada a partir da heurística concebida.....	76
TABELA 02: Informações associadas a cada uma das referências disponibilizadas por uma indústria do segmento de confecção	119
TABELA 03: Demanda atual (em peças) associada à Referência 1	119
TABELA 04: Demanda atual (em peças) associada à Referência 2	120
TABELA 05: Demanda atual (em peças) associada à Referência 3	120
TABELA 06: Demanda atual (em peças) associada à Referência 4	120
TABELA 07: Critérios utilizados na construção dos demais cenários de demanda	121

LISTA DE GRÁFICOS

GRÁFICO 01: Avaliação da performance da heurística da indústria, da heurística proposta, do MINLP e do método BVLC	133
GRÁFICO 02: Avaliação da performance dos métodos de Busca Ávida	134
GRÁFICO 03: Avaliação da performance dos métodos BVLC/BA e dos métodos de Busca com Lista de Nós Abertos e Limitação de Custo.	135
GRÁFICO 04: Avaliação da performance dos métodos de Busca com Lista de Nós Abertos e Limitação de Custo/ Busca Ávida	136

LISTA DE QUADROS

QUADRO 01: Caracterização do problema de encaixe na indústria do segmento de confecção segundo os critérios de Dyckhoof (1990)	39
QUADRO 02: Algoritmos testados na resolução do PPOC	117
QUADRO 03: Resultados obtidos para o cenário Dados_R1_geral	124
QUADRO 04: Resultados obtidos para a Referência 1	126
QUADRO 05: Resultados obtidos para a Referência 2	127
QUADRO 06: Resultados obtidos para a Referência 3	128
QUADRO 07: Resultados obtidos para a Referência 4	129
QUADRO 08: Método recomendado × Tempo disponibilizado pela indústria	140

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 CONSIDERAÇÕES INICIAIS	14
1.2 DEFINIÇÃO DO PROBLEMA	16
1.2.1 Descrição da cadeia produtiva têxtil	16
1.2.2 O processo produtivo vinculado ao segmento de confecção	19
1.2.2.1 Pré-montagem.....	19
1.2.2.1.1 Criação.....	19
1.2.2.1.2 Modelagem	19
1.2.2.1.3 Corte	20
1.2.2.2 Montagem.....	20
1.2.2.2.1 Costura.....	20
1.2.2.3 Acabamento	20
1.2.2.3.1 Lavanderia.....	20
1.2.2.3.2 Limpeza e passadoria das peças	21
1.2.3. O Problema de Programação de Ordens de Corte (PPOC)	21
1.3 PRESSUPOSTOS DA PESQUISA	25
1.4 O PROBLEMA ABORDADO NO CONTEXTO LOGÍSTICO.....	26
1.4.1 Da logística ao <i>Supply Chain Management</i> : aspectos conceituais	26
1.4.2 A logística como estratégia de competitividade no setor têxtil	27
1.4.3 O PPOC no contexto logístico	29
1.5 IMPORTÂNCIA DO PROBLEMA	30
1.6 OBJETIVO	31
1.6.1 Objetivo Geral.....	31
1.6.2 Objetivos Específicos.....	31
1.7 LIMITAÇÕES DO TRABALHO	31
1.8 ESTRUTURA DO TRABALHO	32
2 O PPOC E PROBLEMAS CORRELATOS	34
2.1 ASPECTOS INTRODUTÓRIOS	34
2.2. PROBLEMAS CORRELATOS DISPONÍVEIS NA LITERATURA	35
2.2.1 O problema de corte industrial.....	35
2.2.1.1 Problemas de corte em geral.....	35
2.2.1.2 O problema de corte na indústria têxtil	39

2.2.2 O PPOC na fabricação de peças sob encomenda.....	40
2.3 PUBLICAÇÕES SOBRE O PPOC DISPONÍVEIS NA LITERATURA	41
3 FORMULAÇÃO DO PPOC.....	43
3.1 VISÃO GERAL DO PROBLEMA.....	43
3.1.1 Considerações iniciais	43
3.1.2 Custos associados ao PPOC.....	43
3.1.2.1 Custo do corte	43
3.1.2.2 Custo do enfiesto.....	43
3.1.2.3 Custo da perda nas dobras	44
3.1.2.4 Custo da perda de aproveitamento do tecido	44
3.1.2.5 Custo do estoque de peças	45
3.1.2.6 Custo da sobra de peças.....	45
3.1.2.7 Custo de <i>setup</i> na estamparia.....	45
3.1.3 Limitações associadas ao PPOC	46
3.1.3.1 Limitações físicas de produção	46
3.1.3.2 Limitações geradas pelo mercado	47
3.2 VARIÁVEIS DE DECISÃO DO PPOC	47
3.3 FUNÇÃO-OBJETIVO DO PPOC	48
3.3.1 Custo do corte	48
3.3.2 Custo do enfiesto.....	49
3.3.3 Custo da perda nas dobras	49
3.3.4 Custo da perda de aproveitamento de tecido	50
3.3.5 Custo do estoque	50
3.3.6 Custo da sobra.....	51
3.4 RESTRIÇÕES DO PPOC	51
3.4.1 Restrições quanto ao comprimento máximo de enfiesto.....	51
3.4.2 Restrições quanto à espessura da faca de corte.....	52
3.4.3 Restrições quanto à necessidade de atendimento à demanda	52
3.4.4 Restrições de não-negatividade e integridade	53
3.5 FORMULAÇÃO COMPLETA DO PPOC.....	53
4 TÉCNICAS PROMISSORAS DE RESOLUÇÃO DO PPOC.....	55
4.1 ASPECTOS INTRODUTÓRIOS	55
4.2 MÉTODOS EXATOS.....	55
4.2.1 Algoritmos de Planos de Corte	56

4.2.2 Algoritmo Branch-and-Bound	57
4.2.3 Algoritmo de Balas	57
4.3 MÉTODOS HEURÍSTICOS	58
4.3.1 Algoritmo Genético.....	58
4.3.2 Relaxação Lagrangeana.....	61
4.4 MÉTODO DE BUSCA EM GRAFOS	64
4.4.1 Conceitos Básicos	64
4.4.2 Processos de busca em grafos.....	65
4.4.2.1 Aspectos introdutórios.....	65
4.4.2.2. Busca em largura ou horizontal pura.....	66
4.4.2.3 Busca em profundidade ou vertical pura	67
4.4.2.4 Outros critérios de ordenação de nós no processo de busca.....	70
4.4.3 Algoritmo de Dijkstra.....	71
4.4.4 Algoritmo A*	72
4.5 TESTES REALIZADOS PARA RESOLUÇÃO DO PPOC	74
4.5.1 Resolução do PPOC a partir de um <i>solver</i> disponível comercialmente.....	74
4.5.2 Concepção de um método heurístico	75
4.5.3 Resolução do PPOC a partir de um algoritmo genético.....	82
4.5.4 Resolução do PPOC a partir de relaxação lagrangeana	83
5 RESOLUÇÃO DO PPOC ATRAVÉS DE PROCESSOS DE BUSCA EM GRAFOS. 85	
5.1 ASPECTOS INTRODUTÓRIOS	85
5.2 MODELAGEM DO PROBLEMA UTILIZANDO GRAFOS	85
5.2.1 Definição de Nó	85
5.2.2 Definição de custo.....	87
5.2.3 Definição de Arco	87
5.2.4 Processo de expansão de nós	87
5.3 ADAPTAÇÃO DE ALGORITMOS DE BUSCA EM GRAFOS.....	91
5.3.1 Aspectos Introdutórios	91
5.3.2 Definição do <i>nó inicial</i>	92
5.3.3 Algoritmo de busca vertical com limitação de custo	93
5.3.3.1 Aspectos Introdutórios.....	93
5.3.3.2 Ordenação de nós para expansão	94
5.3.3.3 Exclusão de nós do processo de busca	96
5.3.3.4 Algoritmo proposto para resolução do PPOC.....	96

5.3.4 Algoritmo de busca ávida.....	99
5.3.4.1 Aspectos Introdutórios.....	99
5.3.4.2 Critérios de escolha do <i>nó gerador</i>	99
5.3.4.3 Algoritmo proposto para resolução do PPOC.....	100
5.3.5 Algoritmo de busca vertical com limitação de custo/ busca ávida	102
5.3.5.1 Aspectos Introdutórios.....	102
5.3.5.2. Algoritmo proposto para resolução do PPOC.....	103
5.3.6 Algoritmo com lista de nós abertos e limitação de custo	106
5.3.6.1 Aspectos Introdutórios.....	106
5.3.6.2 Critérios considerados na escolha do <i>nó gerador</i>	107
5.3.6.3 Exclusão de nós do processo de busca	107
5.3.6.4 Algoritmo proposto para resolução do PPOC.....	108
5.3.7 Algoritmo de busca com lista de nós abertos e limitação de custo/ busca ávida.....	112
5.3.7.1 Aspectos Introdutórios.....	112
5.3.7.2 Algoritmo proposto para resolução do PPOC.....	113
6 RESULTADOS OBTIDOS.....	118
6.1 REFERÊNCIAS TESTADAS	118
6.2 ALGORITMO ATUALMENTE UTILIZADO PELA INDÚSTRIA TÊXTIL.....	122
6.3 RESULTADOS OBTIDOS PARA REFERÊNCIAS COM CENÁRIOS DE DEMANDA CONSTRUÍDOS A PARTIR DE INFORMAÇÕES REAIS	124
6.4 RESULTADOS OBTIDOS PARA OS CENÁRIOS DE DEMANDA FORNECIDOS PELA INDÚSTRIA	125
6.5 ANÁLISE DOS RESULTADOS	130
6.5.1. Avaliação da performance dos algoritmos em termos de custo e tempo de processamento	130
6.5.2 Avaliação da performance dos algoritmos com base na metodologia proposta por Dolan e Moré (2000).....	131
7 CONCLUSÕES E RECOMENDAÇÕES	137
7.1 CONCLUSÕES SOB O ENFOQUE ACADÊMICO.....	137
7.2 CONCLUSÕES SOB O ENFOQUE DA INDÚSTRIA TÊXTIL.....	139
7.3 RECOMENDAÇÕES	141
8 REFERÊNCIAS.....	143
ANEXOS	151

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

O setor têxtil é um dos segmentos de maior tradição na indústria, contando com uma posição de destaque na economia mundial e sendo o carro-chefe de muitos dos chamados países emergentes, que devem à sua indústria têxtil o papel de destaque que exercem, hoje, no comércio mundial de manufaturas (TEXTILIANET¹, 2005).

O processo de industrialização iniciou no Brasil com o setor têxtil que, ao longo dos anos, passou a contribuir de maneira cada vez mais expressiva no PIB (Produto Interno Bruto), tanto nacional como catarinense. Segundo informações publicadas no Anuário Expressão Maiores do Sul (2005), mesmo diante de todos os percalços enfrentados nos últimos anos pelo setor têxtil, o mesmo continuou tendo relevância para a economia catarinense, representando 15% do Valor de Transformação Industrial, 8% das exportações e empregando 115 mil trabalhadores.

De acordo com estudo realizado pelo Departamento de Economia da Associação Brasileira de Indústrias Têxteis (ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA TÊXTIL, 2005), o setor têxtil emprega cerca de 1,5 milhão de pessoas em todo o país. Este estudo indica ainda a constante expansão da demanda por trabalho no setor, apontando o segmento de confecção como aquele da cadeia produtiva têxtil que mais gerou empregos nos últimos anos.

Especificamente no estado de Santa Catarina, o setor têxtil apresentou uma expansão de oferta de emprego formal de 10,22% e um incremento de 13,2% na produção industrial ao longo do ano de 2004 (BRDE, 2005). A figura a seguir ilustra o incremento das exportações, em US\$ milhões, verificado entre os anos 2004 e 2005, tanto no Brasil como em Santa Catarina (ANUÁRIO EXPRESSÃO MAIORES DO SUL, 2005).

¹ Site mantido pelo IEMI – Instituto de Estudos e Marketing Industrial, contendo amplas informações sobre a cadeia têxtil nacional.

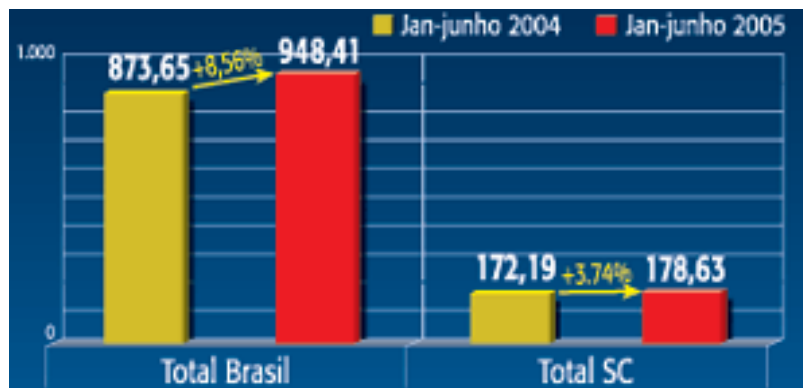


FIGURA 01: Exportações têxteis e confecções

Fonte: Anuário Expressão Maiores do Sul (2005)

Como oportunidade de crescimento associada ao setor, destaca-se a informação publicada pelo Banco Regional de Desenvolvimento do Extremo Sul (2005), que aponta o fato de que o comércio internacional de têxteis e confecções passará, ao longo dos próximos anos, por um processo de abolição do sistema de cotas pelo qual os países ricos importadores restringiram por 40 anos a entrada de têxteis e confecções oriundos de países em desenvolvimento. Esta abolição do sistema de cotas deve ser encarada como uma oportunidade de expandir o mercado de artigos têxteis produzidos no Brasil, mais especificamente aqueles produzidos no estado de Santa Catarina, cujo segmento de vestuário ainda se recupera da crise nas vendas ocorrida na última década.

Saliente-se que a nova ordem mundial, apontada para cenários de globalização da economia, está exigindo novos padrões de competitividade. Isto acaba por impor às indústrias, em geral, readequações estratégicas e otimização de seus processos produtivos, visando o aumento de produtividade e a redução de custos.

Algumas indústrias têxteis já se aperceberam desta realidade, sendo que aquelas com maior poder aquisitivo passaram a investir, nos últimos anos, um significativo montante de recursos financeiros na aquisição de ferramentas que propiciassem uma melhoria de sua performance. Cita-se como exemplo a Marisol que, de acordo com informações divulgadas por seu diretor de Marketing, tem investido intensivamente na aquisição de maquinário, na admissão de mão-de-obra e na expansão de suas unidades industriais. O resultado disso foi um aumento de 14,22% no faturamento, comparando-se os resultados obtidos no primeiro semestre dos anos de 2004 e 2005, respectivamente (ANUÁRIO EXPRESSÃO MAIORES DO SUL, 2005).

No entanto, caso o estado de Santa Catarina queira aumentar ou mesmo manter sua posição de destaque no mercado mundial de têxteis, terá de investir em novos estudos e ferramentas que visem aumentar a competitividade de suas indústrias, já que são poucas as que se encontram atualizadas tecnológica e organizacionalmente (GORINI, 2000).

1.2 DEFINIÇÃO DO PROBLEMA

1.2.1 Descrição da cadeia produtiva têxtil

A cadeia têxtil abrange uma rede de segmentos industriais heterogênea, no que se refere ao tamanho e número de indústrias envolvidas em cada um deles. De acordo com Goularti Filho e Genoveva Neto (1997, p. 63), quatro segmentos principais participam da cadeia produtiva têxtil:

- O de produção de fibras;
- O de fiação, tecelagem e acabamento;
- O de confecção e vestuário;
- O de máquinas e equipamentos.

Conforme trabalho publicado pelo Instituto Euvaldo Lodi intitulado como Análise da eficiência econômica e da competitividade da cadeia têxtil brasileira (2000, p. 21), os principais segmentos do complexo têxtil podem ser representados através da seguinte figura:

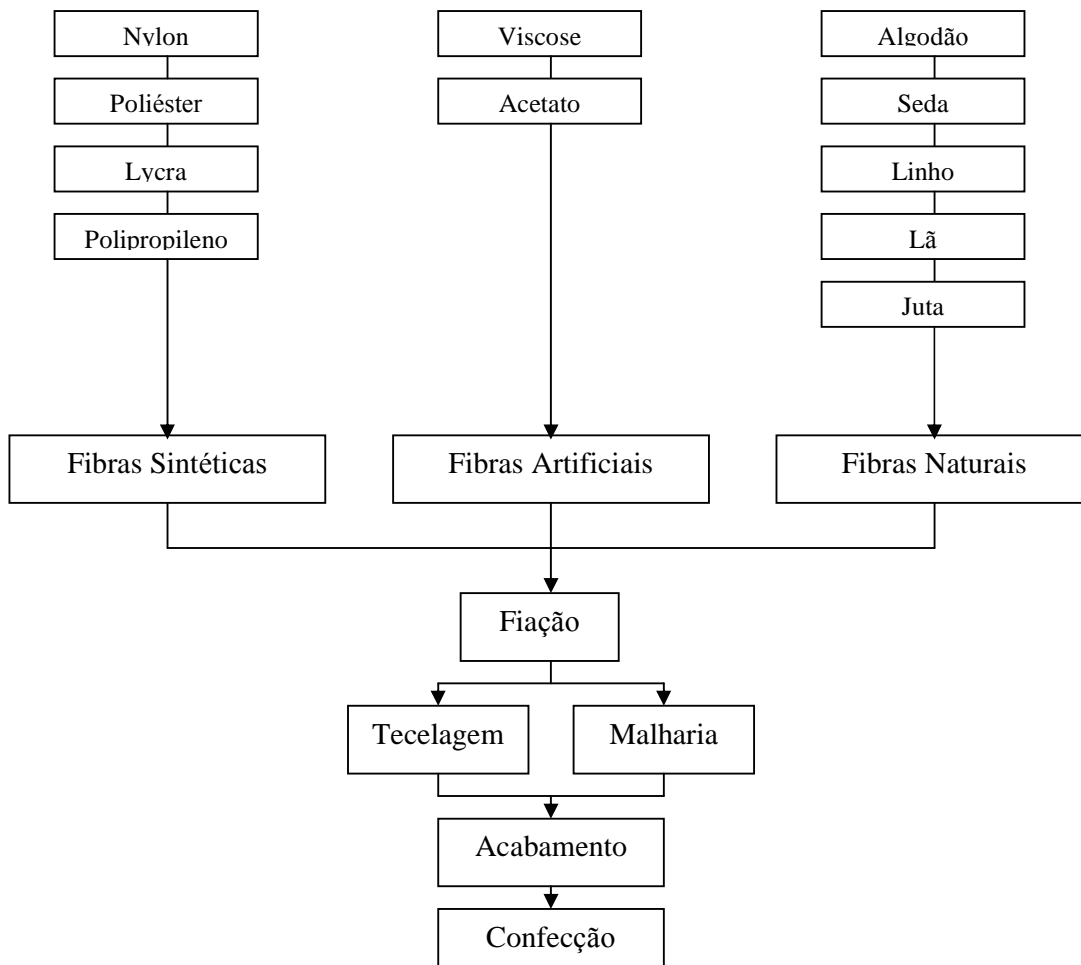


FIGURA 02: Segmentos do complexo têxtil

Fonte: Trabalho publicado pelo Instituto Euvaldo Lodi intitulado como Análise da eficiência econômica e da competitividade da cadeia têxtil brasileira (2000, p. 21).

Uma outra ilustração da cadeia produtiva têxtil, proposta pelo TextiliaNet (2005), procura representar, através de um esquema em forma de cone, a proporção de indústrias em nível nacional que atuam em cada um dos segmentos têxteis.



FIGURA 03: A cadeia produtiva têxtil.

Fonte: TextiliaNet (2005)

(1) Segmento de fornecedores - insumos químicos, fibras e filamentos, máquinas e equipamentos

De acordo com o SEBRAE-SP&IPT (2001), embora os segmentos ou etapas do processo estejam interligados por aspectos técnicos dos produtos, tais etapas não precisam ser internalizadas por uma mesma empresa.

Algumas indústrias optaram por integrar verticalmente seu processo produtivo, fabricando desde os fios para fabricação dos tecidos até o produto acabado. Há também aquelas que produzem parcialmente os tecidos para confeccionar seus produtos, ou seja, dispõem de tecnologia e maquinário para produzir alguns tipos de tecidos, enquanto outros têm de ser comprados.

Segundo Machado (2004), encontram-se neste padrão de integração empresas como a Cia. Hering, a Karsten S/A, a Marisol, a Têxtil Canatiba, a Vicunha Têxtil, a Santista Alpargatas Têxtil, o grupo Coteminas, entre outras.

Finalmente, grande parte das indústrias do setor têxtil, caracterizadas em sua maioria como sendo de pequeno porte, optaram por dedicar-se apenas às atividades de confecção e acabamento, adquirindo todos os tecidos, acessórios e aviamentos necessários. Segundo

Gorini (2000), são essas pequenas indústrias que normalmente se encontram mais distantes das modernas práticas de gestão.

1.2.2 O processo produtivo vinculado ao segmento de confecção

O segmento de confecção é a ponta final da cadeia produtiva têxtil e inclui não somente a produção de roupas, mas também artigos de cama, mesa e banho e acessórios. Segundo dados do Instituto Euvaldo Lodi (2000), o ramo de confecção de vestuário é o mais representativo em nível nacional, uma vez que as indústrias atuantes no mesmo correspondem a cerca de 83% do total de indústrias do segmento de confecção.

Quanto ao processo produtivo, as atividades de confecção de roupas estão associadas às fases de pré-montagem (que inclui criação, modelagem e corte), montagem (ou costura) e acabamento.

Estas atividades são brevemente descritas a seguir, na ordem em que são executadas. As descrições que se seguem foram adaptadas do trabalho de Cunha (2002).

1.2.2.1 Pré-montagem

1.2.2.1.1 Criação

Consiste na concepção da roupa. Nessa etapa são desenhados os modelos e escolhidos os tecidos (textura e padrão). O sucesso da fase de criação depende do nível de informação e da criatividade do estilista.

1.2.2.1.2 Modelagem

Consiste na concretização das idéias do estilista através de protótipos, que servirão de base para a geração de moldes. Esta etapa é realizada pelo modelista, que também executa a gradação (ou gradeamento) - adaptação dos moldes aos diferentes tamanhos. Na modelagem também se inclui a atividade de encaixe, que é o posicionamento dos moldes no tecido de forma a se obter o melhor aproveitamento, minimizando perdas. Existem procedimentos automatizados que permitem a simulação de diferentes combinações de encaixes de peças no tecido, determinando também os tamanhos de cada uma delas, com base nas regras de gradeamento inseridas nos programas operacionais.

1.2.2.1.3 Corte

O corte feito manualmente exige grande habilidade por parte do cortador e é feito com o auxílio de uma serra fita ou circular, ou ainda de uma guilhotina. Atualmente, os tecidos podem ser cortados através de máquinas automatizadas – equipamentos com dispositivo de controle numérico e ligado diretamente ao sistema CAD (*Computer Aided Design*) utilizado no gradeamento e encaixe. Neste caso, utilizam-se lâminas de corte com capacidade para até 300 camadas de tecido (o número máximo de camadas depende também da espessura do tecido a ser cortado).

Com relação à atividade de corte, vale salientar que a mesma é precedida da programação das ordens de corte, que consiste no tema desta pesquisa e será explicitado posteriormente.

1.2.2.2 Montagem

1.2.2.2.1 Costura

É a parte mais complexa e intensiva de trabalho humano no processo de produção. Segundo Gorini (2000), são identificadas nesta fase tarefas que retardam avanços tecnológicos no campo da automação industrial. Estas restrições estariam ligadas, segundo a referida autora, às características do tecido e às suas diferentes texturas.

Nas fases anteriores à da costura, os avanços tecnológicos obtidos até o momento se deram em função da utilização dos sistemas CAD e CAM (*Computer-Aided Manufacturing*), sendo que os benefícios no processo produtivo estão associados à redução no tempo de produção e no desperdício de tecido, além de flexibilidade para alteração dos modelos.

1.2.2.3 Acabamento

1.2.2.3.1 Lavanderia

A lavagem das peças já prontas não é realizada em todos os artigos, mas apenas naqueles cujo tecido é jeans ou outro tecido plano, que possa adquirir maciez e aparência de lavado. Artigos de malha também podem passar por esta etapa.

1.2.2.3.2 Limpeza e passadoria das peças

Nesta fase, são executadas tarefas como corte de linhas, de sobras de pano e passadoria da roupa pronta, a fim de deixá-la pronta para embalagem e transporte.

Neste item, procurou-se apresentar uma visão geral do processo de confecção de peças na indústria têxtil. O próximo item descreve o problema de programação de ordens de corte, problema objeto deste estudo e que se encontra vinculado à etapa de pré-montagem, como já citado anteriormente.

1.2.3. O Problema de Programação de Ordens de Corte (PPOC)

O PPOC é um problema inerente às indústrias que atuam no segmento de confecção têxtil. Para entendê-lo, é importante conceituar duas operações realizadas no processo de confecção: o corte e o enfiado.

Entende-se por corte a ação de recortar no tecido, de forma manual ou automatizada, os componentes das peças que serão confeccionadas. A figura a seguir ilustra o exemplo de uma faca de corte utilizada em indústrias do segmento de confecção.



FIGURA 04: Exemplo de uma faca de corte de peças

Denomina-se enfesto, na indústria têxtil, a ação de estender uma ou mais camadas de tecido sobre a mesa de corte. A máquina de enfesto deverá esticar, em cada uma das mesas de corte, um comprimento compatível com aquele previsto na programação realizada previamente. A figura a seguir ilustra a imagem de uma máquina de enfesto utilizada em indústrias do segmento de confecção, acoplada a uma mesa de corte.



FIGURA 05: Exemplo de uma máquina de enfesto de tecido

Para cada coleção de peças lançada ao longo do ano, a indústria têxtil de confecção costuma trabalhar com diferentes referências. Uma referência consiste em um modelo de peça, tendo como características, além do tipo da peça (calça, blusa, vestido ou outro), o tecido a ser utilizado, os aviamentos (botões, linhas de costura, etc.), a base principal da peça e os respectivos contrastes (mangas, bolsos, aplicações de outros tecidos), dentre outras. Para cada uma das referências da coleção, a indústria define previamente a variedade de tamanhos e de cores para as quais serão confeccionadas as peças.

A programação do corte das peças de vestuário é normalmente realizada por meio de Ordens de Programação (OP's) semanais. Através das OP's, são definidas as referências de peças que serão cortadas para atendimento a uma demanda prevista em nível semanal.

Para cada referência envolvida em uma OP, há que se programar a respectiva Ordem de Corte (OC), que consiste na definição do número de mesas de corte a ser utilizado e, para cada uma delas, do número de peças, para cada tamanho e cada cor, a serem cortadas visando atender à demanda prevista. Este número de peças, para cada mesa de corte, é calculado como o produto do número de marcações (comprimento de tecido, no sentido longitudinal da mesa, necessário para o corte de uma peça) de cada tamanho pelo número de camadas previstas para cada cor de tecido. O número total de peças de uma OC consiste no somatório das peças que serão cortadas em cada uma das mesas alocadas para uma mesma referência. Procura-se definir OC's que garantam o atendimento à demanda prevista ao menor custo possível.

Encontra-se apresentada, no Apêndice 01, uma figura que busca representar a vista frontal de uma mesa de corte sobre a qual estão inseridas diferentes marcações e camadas de tecido para o posterior corte das peças.

A programação das OC's é feita para a base principal² da referência, seguindo-se, para os contrastes, as mesmas definições quanto à quantidade de marcações por tamanho e de camadas por cor de tecido. O PPOC é resolvido partindo-se de uma demanda, prevista inicialmente com base na demanda histórica de referências similares (ou na experiência em vendas do analista) e, após algumas semanas de vendas, com base na demanda histórica verificada.

A partir da programação das OC's, analisa-se a capacidade de corte (comparação entre a necessidade e a disponibilidade de mesas e horas de corte), avalia-se o estoque dos materiais necessários para confecção das peças (incluindo-se aí os aviamentos, acessórios e os tecidos propriamente ditos), confirma-se a data prevista para execução da OP e solicita-se ao departamento de suprimentos a aquisição das necessidades de materiais que não estejam disponíveis em estoque.

O período de tempo compreendido entre a programação das ordens de corte e o corte propriamente dito das peças é relativamente grande para as indústrias que não dispõem de supermercado³ de tecidos, já que somente a partir desta programação é que se dá início ao processo de fabricação de determinados tecidos de cores ainda indisponíveis.

Com relação a este aspecto, vale ressaltar a necessidade de que as indústrias deste segmento avaliem o *trade-off* (troca compensatória) associado à decisão de se implantar supermercados de tecido: se por um lado os custos de estoque tendem a aumentar, por outro se torna possível reduzir o intervalo de tempo entre a programação das ordens de corte e o corte efetivo das peças, redução esta que pode culminar em menores erros de previsão de demanda.

Os profissionais das indústrias, responsáveis pela programação das OC's, normalmente utilizam métodos (concebidos na própria indústria ou disponíveis em soluções comerciais) que lhes auxiliam nesta tarefa.

Como forma de melhor entender este processo buscou-se avaliar o PPOC em uma indústria de médio porte, pertencente ao segmento de confecção, que dispõe de um método

² Componente principal da peça, sendo definido como aquele que envolve maior área de tecido. Os demais componentes de uma peça (normalmente bolsos, mangas e demais detalhes) são chamados de contrastes.

³ Termo originalmente empregado na descrição da filosofia de Manufatura Enxuta ou Sistema Toyota de Produção. Refere-se à manutenção de estoques em pontos estratégicos da cadeia produtiva, de maneira a garantir a continuidade do fluxo de materiais (SHINGO, 1996).

para lhe auxiliar na resolução do referido problema. Este método, baseado em heurísticas concebidas dentro da própria indústria, proporcionaliza o número de marcações alocadas a uma mesa, em função da demanda prevista para cada tamanho. Como o referido método não é capaz de considerar todos os custos e dificilmente agrega, em uma mesma mesa, camadas de tecido com cores diferentes, compete ao programador de OC's avaliar os resultados e, com base em sua experiência, verificar a possibilidade de garantir um melhor aproveitamento das mesas de corte.

O método em questão, segundo a opinião de seu usuário, ainda é bastante rudimentar, podendo ser aprimorado no que diz respeito à possibilidade de considerar diferentes alternativas de mistura de cores em uma mesma mesa, bem como à possibilidade de considerar outros custos, inerentes ao problema, que até o presente momento não estão sendo considerados.

Como outra dificuldade associada à obtenção de solução para o PPOC, destaca-se o fato de uma OP, que pode envolver até 100 referências para indústrias de médio porte, ter de ser definida em poucas horas de trabalho para não atrasar a solicitação de tecidos, acessórios e aviamentos. Considerando os poucos recursos disponíveis no método ao qual se teve acesso, grande parte do sucesso na concepção de uma OP advém do talento e experiência do respectivo programador, que acaba gastando um tempo considerável no ajuste final das OC's para cada uma das referências de peças. Em determinadas ocasiões, os resultados obtidos são considerados satisfatórios. Em outras ocasiões, os resultados obtidos, medidos em termos de sobras de peças, geram ordens de corte com um grande excedente de peças comparativamente à demanda prevista, culminando em custos excessivos.

Encontram-se disponíveis no mercado algumas soluções que se propõem a solucionar o PPOC, destacando-se entre elas o OPTIPLAN (LECTRA, 2005) e o AUDACES VESTUÁRIO (AUDACES, 2005). Estas soluções, no entanto, estão mais focadas na resolução do problema de encaixe (denominação do problema de corte nas indústrias têxteis), acabando por desconsiderar alguns custos inerentes ao PPOC e, com isto, prejudicando a qualidade dos resultados obtidos.

Com base nas informações apresentadas, a definição do problema pode ser resumida através da seguinte pergunta: é possível conceber um modelo de obtenção da solução para o PPOC que inclua as principais parcelas de custo identificadas no problema?

1.3 PRESSUPOSTOS DA PESQUISA

No início da pesquisa acerca do problema definido anteriormente, foram especificados alguns pressupostos, sem os quais tornar-se-ia inviável buscar conceber um método para resolução do PPOC. Estes pressupostos encontram-se apresentados a seguir:

a) **É possível quantificar os principais custos inerentes ao PPOC.**

Sem a possibilidade de quantificar os custos supracitados, não há como solucionar o problema visando alcançar o mínimo custo total.

b) **É possível modelar os principais custos inerentes ao PPOC.**

Sem a modelagem dos principais custos inerentes ao PPOC, não seria possível conceber um modelo que buscasse a solução de mínimo custo total, ou seja, uma solução melhor do que aquelas obtidas até o presente momento.

c) **O método a ser concebido para resolução do PPOC pode ser processado em um tempo compatível com aquele disponível pela indústria têxtil.**

Para as indústrias que não dispõem de supermercado de malhas, torna-se necessário resolver o PPOC para as OP's semanais em um intervalo de tempo relativamente curto, de forma a não atrasar a colocação dos pedidos de tecidos e acessórios. A concepção de um método que implique em um tempo de processamento maior do que o disponível pode inviabilizar a utilização do mesmo por parte destas indústrias. Portanto, existe uma limitação de tempo que tem de ser considerada na concepção do método de resolução do PPOC.

d) **É possível gerar um método que encontre a solução do PPOC em qualquer época do ano, mesmo com a previsão de demanda futura tendendo a zero.**

A explicação deste pressuposto deve se precedida da definição dos seguintes conceitos: demanda atual e demanda futura. Ambos os conceitos estão relacionados a previsões de venda em nível semanal.

Entende-se por demanda atual de peças a demanda prevista para um horizonte de curto prazo (normalmente a semana seguinte à semana do corte das peças). A demanda futura diz respeito a uma previsão de demanda para um horizonte maior (normalmente duas semanas após a realização do corte das peças). Conforme será explicitado adiante, o modelo para resolução do PPOC deve considerar a possibilidade de gerar peças adicionais à demanda atual, já que as mesmas podem ser estocadas para atender a demanda futura.

A partir de testes realizados com as heurísticas adotadas atualmente por uma indústria de confecção já mencionada anteriormente, verificou-se que a qualidade dos resultados piora consideravelmente à medida que os valores previstos de demanda futura tendem a zero. Esta

dificuldade em solucionar o PPOC para demandas futuras baixas foi confirmada ao se utilizar um *solver* matemático (experiência a ser descrita no capítulo quatro), através do qual não se conseguia obter a solução do problema para valores de demanda futura próximos de zero. As indústrias do segmento de confecção costumam lançar cerca de quatro a seis coleções por ano e, portanto, defrontam-se freqüentemente com baixas demandas futuras, em termos de número de peças. Portanto, torna-se imprescindível dispor de um método que gere boas soluções, independentemente dos valores previstos quanto à demanda futura.

1.4 O PROBLEMA ABORDADO NO CONTEXTO LOGÍSTICO

1.4.1 Da logística ao *Supply Chain Management*: aspectos conceituais

Segundo Fleury, Wanke e Figueiredo (2000), o período entre 1980 e 2000 foi marcado por grandes transformações no mundo empresarial, sendo que o movimento da qualidade total e o conceito de produção enxuta trouxeram consigo um conjunto de técnicas como o JIT (*just-in-time*), CEP (Controle Estatístico do Processo), QFD (*Quality Function Deployment* – Desdobramento da Função Qualidade), *kanban* e Engenharia Simultânea. Na trilha de todas essas mudanças, os autores inserem dois outros conceitos:

- a. O primeiro deles seria a logística integrada, conseqüência principalmente dos movimentos da produção enxuta e do JIT. Os autores citam que para a logística ser gerenciada de forma integrada, a cadeia produtiva deve ser tratada como um sistema, ou seja, um conjunto de componentes interligados que trabalham de forma coordenada com o objetivo de atender o cliente de maneira satisfatória ao menor custo possível;
- b. O segundo dos conceitos, o *Supply Chain Management* (SCM) – Gerenciamento da Cadeia de Suprimentos, começou a se desenvolver no início dos anos 90. Segundo Novaes (2001), nessa nova abordagem a integração entre os processos ao longo da cadeia de suprimentos continua a ser feita em termos de fluxo de materiais, de informação e de dinheiro. No entanto, a partir de então, os agentes participantes atuam em uníssono e de forma estratégica, buscando os melhores resultados possíveis em termos de redução de custos, de desperdícios e de agregação de valor para o consumidor final.

Com relação ao conceito supracitado, é importante salientar que existe, ainda nos dias atuais, uma certa polêmica acerca do significado de SCM e de sua relação com a logística. Alguns autores, tais como Fleury, Wanke e Figueiredo (2000), acreditam que o conceito de

SCM não é simplesmente uma evolução do conceito de logística integrada como proposto por Novaes (2001). Neste contexto e segundo aquele autor, a logística seria um “subconjunto” do gerenciamento da cadeia de suprimentos.

No entanto, estes e outros autores concordam que a implementação do SCM traz vantagens à cadeia produtiva como um todo, dentre as quais a redução de estoques, o menor *lead-time* (tempo de ciclo do pedido), a maior flexibilidade e a maior produtividade (HARRISON; VAN HOEK, 2003).

1.4.2 A logística como estratégia de competitividade no setor têxtil

De acordo com estudo elaborado pelo Departamento Intersindical de Estatísticas e estudo Socioeconômicos (2002), o setor têxtil brasileiro e catarinense passou por transformações muito importantes na década de 90, transformações estas que alteraram significativamente os processos de produção, da gestão da força de trabalho e de organização das empresas. Segundo este estudo, houve renovação do parque de máquinas e a introdução de tecnologias de organização, como programas de qualidade total, *just-in-time*, sistemas de melhoria contínua, caixas de sugestões etc.

Estudos como os de Gorini (2000) salientam a importância de se gerenciar de maneira eficiente os fluxos de produtos e informações ao longo da cadeia produtiva têxtil como forma de manter-se competitivo num mercado cada vez mais dinâmico e incerto.

Cunha (1996) salienta que a infra-estrutura precária identificada em grande parte das indústrias têxteis nacionais ainda é um entrave à adoção de filosofias como o *just-in-time* e o *quick-response*.

A filosofia *just-in-time*, concebida na década de 50 e utilizada pela primeira vez no sistema de produção da Toyota, consiste em se estabelecer ações que permitam atingir a produção sem estoque (ou estoque zero). O autor da filosofia salienta que os insumos/ produtos devem ser disponibilizados apenas no momento em que se tornarem necessários, já que estoques representam custos significativos para a indústria (SHINGO, 1996). Vale lembrar que o gerenciamento de estoques é uma das atividades primárias da logística (BALLOU, 1993).

O *quick-response*, que em linhas gerais consiste em promover ações que propiciem a entrega dos produtos solicitados às indústrias em pequenos volumes e frequências maiores, é considerado por Ângelo (2001) o *pipeline* da logística.

Os estudos supracitados, vinculados ao setor têxtil, indicam que o caminho para o aumento da competitividade em nível global passa pelo aprimoramento do gerenciamento logístico por parte das indústrias ou, de maneira mais abrangente, pelo aprimoramento do *Supply Chain Management* - SCM.

Tanto o *just-in-time* como o *quick-response* estão intimamente associadas à logística e, num contexto mais amplo, ao SCM. Em outras palavras, para que se obtenha êxito nos objetivos de redução de estoques e resposta rápida ao cliente, torna-se necessário gerenciar de maneira eficiente os fluxos de materiais e informações desde o fornecedor de matéria-prima até o cliente final (CHRISTOPHER, 2002).

Especificamente para o caso da indústria têxtil, Goularti Filho e Genoveva Neto (1997, p. 94) avaliam que:

Os fatores de competitividade internos à cadeia produtiva têxtil-vestuário concentram-se basicamente em dois elementos: os diferenciais no custo total de produção entre os países e a capacidade de “resposta rápida” (*quick response*) às variações da demanda. Enquanto o primeiro é fortemente influenciado pela dispersão internacional dos salários pagos nessas indústrias, o segundo enfatiza a necessidade de flexibilidade no processo de produção e na gerência das empresas.

Segundo Rubio (2002), o processo de organização que busca satisfazer o cliente ao menor custo possível envolve as funções de compras, Planejamento e Controle da Produção – PCP e distribuição. Ainda segundo este autor, exceto as estratégias de desenvolvimento de marcas, *design* e de posicionamento no mercado, relacionadas às atividades de marketing, todas as outras fazem parte de um gerenciamento logístico.

Ballou (2001) sugere que uma estratégia logística tem três objetivos: (I) redução de custos, dirigida basicamente para reduzir custos variáveis associados à movimentação e à estocagem de produtos; (II) redução do capital, visando reduzir investimentos no sistema logístico e; (III) melhorias no serviço, reconhecendo que o aumento de receita depende do nível de serviço oferecido ao cliente.

Neste contexto, é importante esclarecer que a utilização da estratégia logística diz respeito à adoção de ferramentas tecnológicas e de gestão que propiciem um maior compartilhamento de informações e a otimização do fluxo logístico de produtos ao longo da cadeia produtiva.

1.4.3 O PPOC no contexto logístico

A otimização do fluxo logístico de produtos envolvido no segmento de confecção passa por diversas questões, dentre as quais aquelas ligadas à automação de algumas atividades que vêm sendo executadas manualmente ou com o auxílio de ferramentas rudimentares por grande parte das indústrias do ramo. Entende-se que pesquisas que tenham como produto final a disponibilização de ferramentas para automatizar as atividades envolvidas no processo de confecção, incluindo-se aí a programação de ordens de corte, é de fundamental importância quando se pensa em aprimorar o gerenciamento dos fluxos logísticos e, conseqüentemente, tornar o setor têxtil mais competitivo.

A resolução do PPOC de maneira adequada culmina na redução dos custos de manutenção de estoques e de armazenagem. Segundo Ballou (1993), a atividade de manutenção de estoques, bem como a atividade de transportes, são consideradas atividades primárias porque são essenciais para a garantia da continuidade do fluxo de materiais, além de serem as mais onerosas dentre todas as atividades logísticas. Pelo fato do PPOC estar intimamente ligado à quantidade estocada de peças semi-acabadas e acabadas, respectivamente, entende-se que ele está inserido no escopo do gerenciamento logístico nas indústrias de confecção. Além disso, ações que promovam a redução da quantidade estocada de peças semi-acabadas e acabadas vão ao encontro do que preconiza a filosofia *just-in-time* e, por conseqüência, ao encontro do que os especialistas julgam necessário para aumentar a competitividade do setor têxtil.

Saliente-se também que o PPOC envolve diferentes parcelas de custos que, quando consideradas de maneira individual, podem apontar para soluções divergentes em termos de disposição das camadas de tecido e número de marcações de cada tamanho a ser cortado (esta característica será mais bem detalhada posteriormente). Apenas um modelo que considere todas as parcelas de custo e que seja capaz de avaliar o *trade-off* existente entre os mesmos, poderá apontar uma boa solução em nível global para este problema.

Resultados melhores do que os obtidos atualmente para o PPOC culminam na redução dos custos de produção, dos custos de manutenção de estoques e de armazenagem, dos custos associados à sobra de peças não absorvidas pelo mercado e, conseqüentemente, na redução de custos totais. A redução de custos, bem como a garantia do nível de serviço esperado pelo cliente, são os dois grandes objetivos inseridos no conceito de *Supply Chain Management*, e essenciais para uma empresa que busque se tornar mais competitiva em seu mercado de atuação.

1.5 IMPORTÂNCIA DO PROBLEMA

A definição das ordens de corte em uma indústria do segmento de confecção interfere diretamente na quantidade estocada de peças semi-acabadas e acabadas, respectivamente. As indústrias têxteis deste segmento são caracterizadas por um elevado risco de obsolescência de seus produtos. Sendo assim, grandes estoques em uma coleção de determinada estação podem representar prejuízos expressivos, devido ao aumento de peças que terão de ser comercializadas a preços promocionais, por não terem sido vendidas ao preço pelo qual foram inicialmente ofertadas. Por outro lado, conforme já citado, alguns custos envolvidos na programação de ordens de corte parecem apontar para um maior aproveitamento do comprimento das mesas, podendo indicar o corte de um maior número de peças como forma de reduzir custos totais. Sem a análise deste *trade-off* de custos, dificilmente serão obtidas boas soluções para o PPOC.

As soluções disponíveis no mercado (OPTIPLAN, 2005; AUDACES VESTUÁRIO 2005) não consideram todas as parcelas de custo inerentes ao PPOC e, portanto, não geram resultados que visem reduzir custos totais. Essas soluções desconsideram, por exemplo, a possibilidade de estocar peças durante determinado tempo, no caso de haver demanda prevista para um horizonte futuro.

A concepção de um modelo que leve em conta as várias parcelas de custo envolvidas no problema é de extrema relevância, como forma de subsidiar as indústrias na decisão de quanto produzir e, conseqüentemente, quanto estocar. Sendo assim, esta pesquisa constitui-se em uma importante contribuição ao setor, que não dispõe até o presente momento de um modelo tão abrangente (em termos de custos avaliados) como o ora proposto, caracterizando desta forma o seu ineditismo.

Finalmente, e conforme será apresentado no capítulo três, o PPOC consiste em um problema de otimização combinatória, pode incluir um número considerável de variáveis (cerca de 200 para cada OC, dependendo do número de tamanhos, de cores e de mesas a ela alocadas) e possui outras características que o tornam extremamente complexo do ponto de vista matemático. Por este motivo, a obtenção de uma solução para o problema envolveu a avaliação de diferentes métodos e a realização de uma série de testes, conforme será explicitado adiante.

1.6 OBJETIVO

1.6.1 Objetivo Geral

Esta pesquisa tem por objetivo conceber um modelo e propor uma solução consistente para resolução do PPOC, de forma a considerar os principais custos envolvidos neste processo decisório, bem como as limitações de recursos disponíveis e de atendimento à demanda.

1.6.2 Objetivos Específicos

O objetivo geral apresentado anteriormente pode ser desmembrado nos seguintes objetivos específicos:

- I. Identificar as parcelas de custos associadas ao PPOC;
- II. Identificar as restrições que terão de ser respeitadas na resolução do problema;
- III. Modelar o PPOC como um problema de programação matemática de minimização de custos;
- IV. Selecionar possíveis técnicas de resolução do problema de minimização modelado na etapa anterior, buscando definir uma técnica adequada sob os enfoques “qualidade da solução obtida” e “custo computacional”, medido em termos de tempo de processamento;
- V. Implementar e testar a técnica selecionada para resolução do PPOC;
- VI. Avaliar a consistência da solução proposta, comparando seus respectivos resultados (com relação a diferentes referências de peças) àqueles que seriam obtidos pela utilização do método adotado atualmente pela indústria pesquisada.

1.7 LIMITAÇÕES DO TRABALHO

Devido à grande complexidade do PPOC do ponto de vista matemático, não se tem a pretensão de chegar a soluções “exatas”, ou seja, atingir o “mínimo custo global” referente às OC’s não é objetivo deste trabalho. O que se busca é a concepção de um método com consistência suficiente para obter soluções melhores do que aquelas obtidas atualmente pela indústria têxtil.

Além disso, conforme já explicitado, a solução do problema tem de ser obtida em um tempo de processamento compatível com aquele disponível pelas indústrias. Esta limitação

foi considerada na concepção do método, mesmo diminuindo, em alguns momentos, a precisão dos resultados obtidos.

Conforme será apresentado no capítulo três, dentre os custos identificados no PPOC, existe o custo de *setup* de estampa. Nas referências para as quais definiu-se a aplicação de estampa na peça, existe um custo de *setup* (associado basicamente à preparação das máquinas e troca de tintas) a cada mudança de cor do tecido. O modelo apresentado desconsidera esta parcela de custo.

Pelo que se pôde obter de informações quanto às indústrias do segmento de confecção, não é prática ainda muito comum a junção de tecidos de diferentes cores em uma mesma OC. A mistura de diferentes cores em uma ordem de corte implica na disponibilização, no momento do corte, de uma quantidade suficiente de tecido de todas as cores envolvidas na OC previamente programada. Para isto, indústrias do setor vêm avaliando a viabilidade de implementação de um supermercado de tecidos. Esta questão, no entanto, não faz parte do escopo da presente pesquisa.

Um outro limitante do método é que o mesmo resolve o PPOC apenas para a base principal das peças de uma dada referência. Considerar-se-á, por simplificação, que as OC's dos contrastes seguem aquelas definidas para a base principal.

1.8 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em sete capítulos. O conteúdo apresentado em cada um deles é resumido a seguir.

No primeiro capítulo, procura-se contextualizar o problema, caracterizá-lo como um problema associado à logística, definir os pressupostos da pesquisa, salientar sua importância, seus objetivos (geral e específicos) e limitações.

No segundo capítulo, faz-se uma revisão acerca dos trabalhos publicados sobre o PPOC e problemas correlatos, tais como os problemas de corte.

No terceiro capítulo, o PPOC é formulado como um problema de programação não linear inteira, descrevendo-se as variáveis de decisão, as parcelas da função-objetivo e as restrições do problema.

No quarto capítulo, são apresentadas técnicas promissoras de resolução do PPOC, buscando-se conceituá-las e citar possíveis aplicações das mesmas. Ao final deste capítulo, relata-se a experiência obtida na tentativa de utilização de algumas dessas técnicas para resolução do PPOC.

No quinto capítulo apresenta-se a proposta de um método para resolução do PPOC, método este que consiste basicamente na utilização de processos de busca em grafos. Conforme será detalhado adiante, os algoritmos propostos são uma adaptação, com inovações em determinados aspectos, de algoritmos clássicos, de forma que os mesmos possam ser utilizados na resolução do problema.

No sexto capítulo são apresentados os resultados obtidos para 34 cenários de demanda, construídos a partir de informações acerca de quatro referências cujos dados foram disponibilizados por uma indústria do segmento de confecção. A partir destes resultados, identificam-se os algoritmos que atingiram a melhor performance em termos de custo da solução obtida, dentro da limitação de tempo para resolução do PPOC.

Finalmente, apresenta-se no sétimo capítulo uma avaliação dos objetivos alcançados ao longo da pesquisa, salientando-se os aspectos inéditos contemplados na mesma, bem como seus possíveis benefícios para a indústria de confecção. Ao final deste capítulo, sugerem-se futuras pesquisas vinculadas ao PPOC.

2 O PPOC E PROBLEMAS CORRELATOS

2.1 ASPECTOS INTRODUTÓRIOS

O PPOC, tal como é definido na presente pesquisa, é ainda pouco abordado na literatura. Por outro lado, encontram-se disponíveis várias publicações acerca dos problemas de encaixe de peças, que consistem no problema de corte vinculado à indústria têxtil. Mesmo julgando não se tratar do mesmo problema, realizou-se ampla pesquisa bibliográfica procurando identificar a possibilidade de associar o PPOC a alguma das classes de problemas de corte e empacotamento já estudadas.

No que se refere à programação de ordens de corte, há alguns trabalhos relacionados à experiência asiática na produção de peças exclusivas sob encomenda. Conforme será detalhado nos itens subseqüentes, nesta situação o objetivo consiste em minimizar o tempo ocioso das máquinas de corte e/ou o tempo de produção das peças, o que difere da abordagem que se está considerando para o PPOC na presente pesquisa.

Especificamente para o caso das indústrias que atuam no segmento de confecção e que fabricam peças por lotes, há um pequeno número de publicações sobre o PPOC. Destas publicações, grande parte apresenta o problema e sugere técnicas para sua resolução sem, contudo, testá-las. A única publicação encontrada que traz uma abordagem semelhante àquela que se está considerando para o PPOC é a de Jackobs-Blecha (1993), que sugere heurísticas e apresenta alguns resultados sem, no entanto, explicitar o algoritmo utilizado para obtenção dos mesmos.

Conforme já citado, encontram-se disponíveis no mercado as soluções OPTIPLAN (2005) e AUDACES VESTUÁRIO (2005), que se propõem a resolver os problemas de encaixe e o PPOC. No entanto, estas soluções aparentemente não consideram todos os custos inerentes ao problema que vem sendo tratado. Nos próximos itens, apresentam-se as publicações identificadas, bem como as principais características das soluções comerciais supracitadas.

2.2. PROBLEMAS CORRELATOS DISPONÍVEIS NA LITERATURA

2.2.1 O problema de corte industrial

2.2.1.1 Problemas de corte em geral

O estudo do Problema de Corte Industrial, conhecido na literatura inglesa como *Cutting Stock Problem*, chama de arranjo geométrico eficiente àquele que, ao final do processo de corte, gera a menor quantidade de sobras de matéria-prima, ou maior rendimento de uso da matéria-prima original, que pode ser o papel, tecido, couro, plástico, laminado, dentre outras. Neste sentido, Gilmore e Gomory (1961, 1963 e 1965) prestaram importantes contribuições na década de 60, quando estudaram a resolução de problemas de corte com uma, duas, três ou mais dimensões, respectivamente, formulando-os como problemas de programação linear inteira. A figura a seguir ilustra um exemplo de problema de corte industrial bi-dimensional, que pode ser identificado em indústrias de fabricação de papel, vidro, madeira, vestuário, dentre outras. As partes da figura desprovidas de cor representam a matéria-prima não aproveitada no corte.



FIGURA 06: Ilustração do problema de corte industrial bidimensional

Fonte: Optimal Spektrum (2005)

Nos últimos anos, vem-se dando prioridade à resolução dos problemas de corte a partir da utilização de algoritmos heurísticos e metaheurísticos, podendo-se destacar a utilização dos algoritmos genéticos. Neste sentido, destacam-se os trabalhos de Hopper e Turton (2001), Valenzuela e Wang (2001) e Burke, Kendall e Whitwell (2002).

Encontram-se apresentadas a seguir as principais classes de Problemas de Corte e Empacotamento. Julgou-se conveniente apresentá-las neste trabalho como forma de verificar se o PPOC poderia, em função de suas características, ser enquadrado em alguma delas.

Um aspecto peculiar destes problemas é o fato de que alguns problemas sem relação aparente com o contexto podem ser modelados como problemas dessa natureza (OPTIMAL SPEKTRUM, 2005). A figura a seguir ilustra a estrutura geral dos Problemas de Corte e Empacotamento.

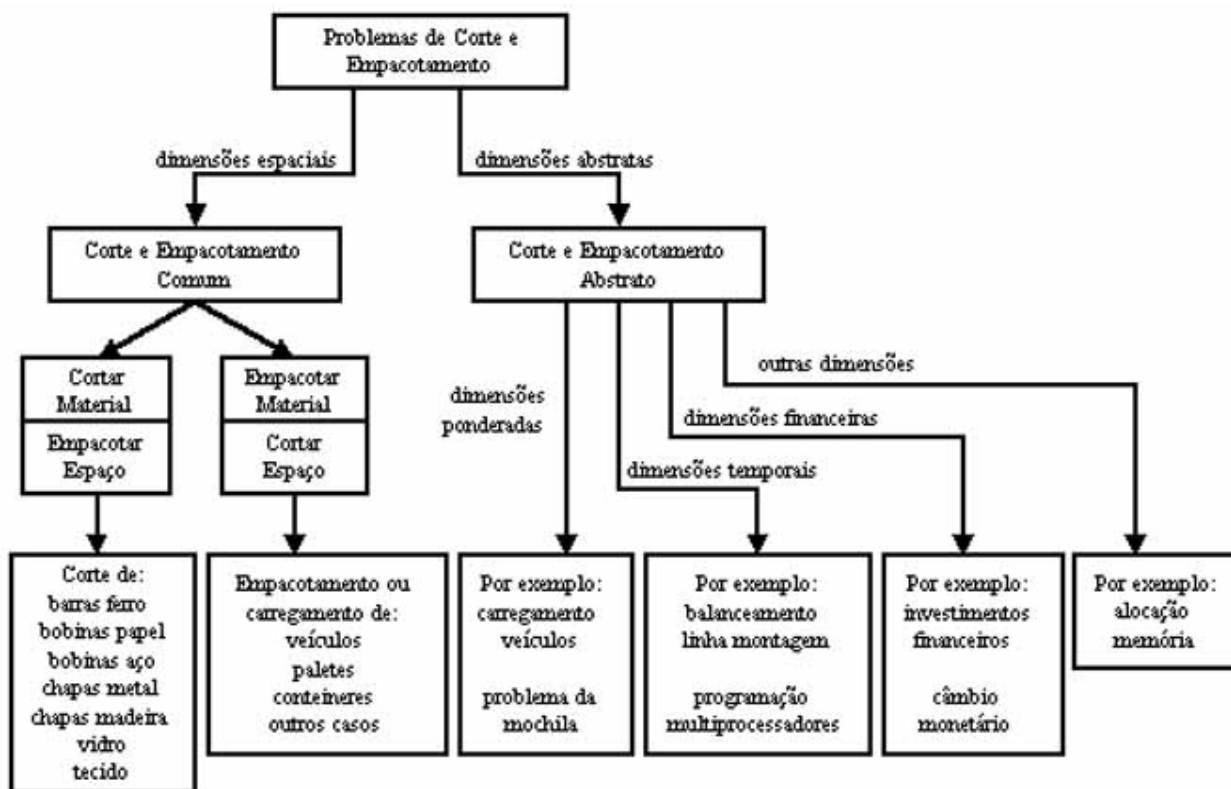


FIGURA 07: Estrutura geral dos Problemas de Corte e Empacotamento

Fonte: OPTIMAL SPEKTRUM (2005)

Dyckhoff (1990) propôs em seu trabalho uma classificação dos problemas de corte e empacotamento com base em uma série de critérios, destacando-se entre eles a dimensionalidade (problemas que envolvem o arranjo de figuras uni, bi, tri ou multidimensionais), o sistema de medidas (medição discreta ou contínua), a forma geométrica dos itens (peças geradas no processo de corte) e objetos (matéria-prima no processo de corte), o sortimento de itens e as restrições envolvidas no problema.

A seguir, apresenta-se uma definição sucinta dos problemas de corte, definição esta adaptada do trabalho de Cunha (1998).

a) ***Trim Loss Problem*** (Problema de Perdas por Corte)

Este problema tem a mesma natureza do problema de corte industrial, já definido anteriormente. O problema de perdas por corte consiste na otimização do processo de corte de peças maiores para a produção de peças menores, visando reduzir o desperdício de matéria-prima.

O problema de perdas por corte pode ser classificado como uni-dimensional (quando apenas uma dimensão é relevante no processo de corte), bi-dimensional (quando duas dimensões são relevantes em tal processo), tri-dimensional (quando três dimensões são relevantes) ou multi-dimensional. Evidentemente, quanto maior o número de dimensões envolvidas no problema, maior seu grau de complexidade (OPTIMAL SPEKTRUM, 2005).

b) ***Bin Packing Problem*** (Empacotamento de caixas, tiras ou faixas)

Neste tipo de problema, o objetivo é reduzir os espaços vazios dentro dos objetos, empilhando os itens da lista de pedidos (itens que têm de ser arranjados em um objeto). Segundo Dyckhoff (1990), há um estreito relacionamento entre problemas de corte e problemas de empacotamento, já que existe uma relação dual entre material – espaço em ambas as situações.

c) ***Knapsack Problem*** (Problema da Mochila)

O problema da mochila consiste na escolha de um subconjunto de itens, cada qual com uma correspondente utilidade e um valor (em geral denominado “peso”) que define o quanto este item utilizará da capacidade da mochila. A escolha dos itens a serem inseridos na mochila deve repercutir no maior benefício possível, medido de acordo com a natureza do problema em estudo.

Este problema tem sido alvo de intensos estudos de vários pesquisadores em função de suas aplicações estratégicas, bem como por interesse teórico (OPTIMAL SPEKTRUM, 2005).

d) ***Vehicle Loading*** (Carregamento de veículos)

O problema de carregamento de veículos consiste em determinar a maneira de acondicionar uma variedade de itens dentro de um veículo, visando minimizar os desperdícios de espaço e, com isso, minimizar o número de viagens necessárias (WHITWELL, 2004).

e) ***Container and Pallet Loading*** (Carregamento de Paletes e Contêineres)

O problema de carregamento de paletes e contêineres é análogo ao problema descrito anteriormente, e consiste em definir um acondicionamento de itens dentro de um equipamento de manuseio (palete ou contêiner) de forma a reduzir espaços vazios.

f) ***Layout Problems*** (Problemas de Leiaute)

Os problemas de leiaute envolvem a alocação de componentes em um espaço disponível, de forma que um conjunto de objetivos pré-especificados possam ser atingidos, respeitando-se restrições de espaço e performance (WHITWELL, 2004).

g) ***Nesting and Partitioning Problems*** (Problemas de Encaixe e Particionamento)

Segundo Whitwell (2004), o termo “encaixe” é utilizado para representar o empacotamento de itens com duas dimensões e formato irregular, problema este muito comum na indústria do segmento metal-mecânico. O arranjo proposto de itens é referenciado como um “particionamento” por aquele segmento industrial.

h) ***Budgeting Capital Problem*** (Problema de Cálculo de Capital)

Em sua forma elementar, o problema de cálculo de capital diz respeito a um montante fixo de capital, de valor C , que tem de ser alocado entre um conjunto n de investimentos possíveis. Cada investimento tem um nível mínimo, denominado L , e um nível máximo, denominado U . O problema consiste em definir a alocação de investimentos que maximize o retorno auferido, respeitadas as restrições de montante máximo de recursos disponíveis, bem como nível mínimo e máximo associado a cada investimento possível (MATHEMATICAL PROGRAM GLOSSARY, 2005).

i) ***Assembly Line Balancing*** (Balanceamento de Linhas de Montagem)

O problema consiste na divisão de trabalho necessário para montar uma variedade de produtos ao longo de diferentes estações e linhas de montagem, com o objetivo de otimizar a utilização de recursos disponíveis em uma indústria (BOCKMAYR; PISARUK, 2001).

j) ***Multiprocessor Scheduling Problems*** (Problemas de Ordenamento de Tarefas com Múltiplos Processadores)

Este problema consiste na programação de uma série de tarefas ao longo de um período definido, de forma que restrições de tempo, mão-de-obra e seqüenciamento sejam respeitadas (CORTES E EMPACOTAMENTO ASSISTIDOS POR COMPUTADOR⁴, 1997).

⁴ CEAC – Grupo de estudos criado e mantido pelo LAC – INPE.

2.2.1.2 O problema de corte na indústria têxtil

O problema de corte na indústria têxtil, denominado no segmento de confecções como problema de encaixe, consiste em definir a disposição das peças a serem cortadas em uma mesa de corte, visando minimizar o desperdício de tecido (TEXTILIANET, 2005).

O quadro apresentado a seguir resume as características do problema de encaixe do segmento de confecção, segundo os principais critérios de classificação e tipologia dos problemas de corte propostos por Dyckhoof (1990).

CRITÉRIO	CARACTERÍSTICAS DO PROBLEMA DE ENCAIXE
Dimensionalidade	Bidimensional
Sistema de medidas	Sistema de medida discreta ou inteira
Forma geométrica	Problema com formas irregulares e assimétricas
Sortimento	Itens com figuras diferentes quanto à forma e tamanho
Disponibilidade	Disponibilidade finita de itens e objetos

QUADRO 01: Caracterização do problema de encaixe na indústria do segmento de confecção segundo os critérios de Dyckhoof (1990)

Fonte: Autora

Com relação às restrições do problema, Dyckhoof (1990) destaca que, para um problema de corte bidimensional (classe em que se encontra o problema de encaixe), o somatório da área ocupada por cada item mais a área “desperdiçada” não pode exceder a área total do objeto dentro do qual serão executados os encaixes. Além disso, o autor define como função-objetivo do problema a minimização da área desperdiçada do objeto, dada pela diferença entre a área total do mesmo e o somatório das áreas de cada peça encaixada.

Conforme exposto por Katsurayama (2005), o problema de encaixe pode ser formulado como um problema de programação linear inteira.

De todos os problemas de corte listados anteriormente, pode-se verificar que nenhum deles possui as características inerentes ao PPOC. Desta forma, constatou-se ser necessário conceber um modelo específico para obtenção de uma solução para este problema.

2.2.2 O PPOC na fabricação de peças sob encomenda

As publicações de Wong, Chan e Ip (2000) e de Wong (2003) apresentam algoritmos que visam resolver o problema de seqüenciamento das ordens de corte e de dimensionamento do número de máquinas de enfiar/ corte na indústria têxtil, visando diminuir o tempo de confecção das peças e o tempo ocioso das máquinas.

O trabalho publicado por Wong, Chan e Ip (2000) diz respeito à utilização de um algoritmo genético para resolver o problema dos sistemas de corte computadorizados utilizados nas indústrias de vestuário que fabricam artigos sob encomenda. Para isto, os autores conceberam um método que visa minimizar o tempo de fabricação dos artigos encomendados, bem como o tempo ocioso das máquinas.

Seguindo esta mesma linha de pesquisa, Wong, Chan e Ip (2001) conceberam um método em dois níveis para resolução do problema de dimensionamento do número de máquinas de enfiar/ corte na indústria têxtil. Com esta finalidade, os autores propuseram a utilização de um modelo de programação não-linear inteira para determinar o número ótimo de máquinas (modelo este que leva em conta a taxa de chegada de pedidos e a taxa de serviço das máquinas) e, posteriormente, um método baseado em teoria das filas para minimizar o tempo ocioso de cada máquina.

Em 2003, Wong (2003) concebeu um método difuso de alocação de capacidade para resolver o problema de balanceamento das linhas de produção na indústria têxtil. Segundo o autor, as operações de enfiar/ corte deveriam ser alocadas entre as máquinas disponíveis em pontos “discretos” do intervalo de tempo considerado, devendo-se monitorar as operações de enfiar/ corte continuamente, com o intuito de replanejá-las quando necessário. O objetivo do método concebido consistia em reduzir a quantidade de estoques intermediários, bem como o tempo ocioso das máquinas de enfiar/ corte.

Com relação aos trabalhos de Wong, Chan e Ip (2000 e 2001) e de Wong (2003), saliente-se que todos eles estão vinculados às indústrias que fabricam artigos sob encomenda, visando minimizar o tempo ocioso das máquinas e/ou de produção das peças, o que diverge completamente do foco abordado na presente pesquisa.

2.3 PUBLICAÇÕES SOBRE O PPOC DISPONÍVEIS NA LITERATURA

Com base na pesquisa bibliográfica realizada, a abordagem que mais se assemelha àquela apresentada neste trabalho foi proposta por Jackobs-Blecha (1993), que afirma ter concebido alguns algoritmos para resolver o PPOC sem, no entanto, detalhá-los. Este problema, na visão da autora, envolve aspectos associados a custos de produção, custos de *setup*, necessidades eventuais de hora-extra, a necessidade de atendimento à demanda atual para diferentes tamanhos, cores e referências. A autora propõe em seu trabalho a utilização de heurísticas de “construção” das ordens de corte, a partir das quais são alocadas numa mesma mesa as peças de um mesmo tamanho (a partir da pesquisa realizada, a autora comenta que é menos custoso para a indústria cortar peças de um mesmo tamanho em uma mesma mesa de corte). Propõe-se também neste trabalho uma heurística de melhoria, que busca efetuar trocas de tamanhos entre mesas de corte diferentes, além de tentar agregar as peças programadas para duas mesas em uma única, visando a diminuição de custos totais.

Destaca-se também a publicação de Mazziotti e Horn (1997), que propuseram a utilização de ferramentas baseadas em técnicas de simulação para a programação das atividades vinculadas ao segmento de confecção têxtil. Os autores desenvolvem uma metodologia baseada no conceito de programação finita, que consiste em programar a produção considerando as limitações de recursos dentro de um sistema. Apesar de apresentarem um exemplo de utilização das ferramentas que envolvem simulação no planejamento da produção de longo prazo, os autores citam a possibilidade de se conceber ferramentas que subsidiem as decisões de curto prazo, tais como as operações de corte na indústria têxtil. Para isto, eles sugerem que se busque gerar, através da técnica de simulação, ordens de corte que minimizem os estoques intermediários ao longo da confecção das peças.

Encontram-se disponíveis para aquisição no comércio algumas soluções comerciais que visam subsidiar a resolução do PPOC em indústrias têxteis. Pode-se citar entre elas o OPTIPLAN (LECTRA, 2005) e o AUDACES VESTUÁRIO (AUDACES, 2005), que geram uma solução para o PPOC que complementa a solução obtida para o problema de encaixe. Entretanto, as empresas responsáveis pela comercialização de tais soluções não divulgam os algoritmos implementados nas mesmas. Pelo que se constatou a partir da verificação de alguns resultados propiciados a partir da utilização destas soluções, elas são capazes de gerar resultados viáveis, mas não há nenhuma evidência de que os mesmos se aproximem da solução ótima.

A partir das publicações encontradas na literatura e apresentadas anteriormente, observa-se que possivelmente não há divulgação, até o presente momento, de estudos acerca do PPOC tal como ele é abordado nesta pesquisa.

3 FORMULAÇÃO DO PPOC

3.1 VISÃO GERAL DO PROBLEMA

3.1.1 Considerações iniciais

Conforme já mencionado no primeiro capítulo deste trabalho, a resolução do PPOC consiste na definição, para cada uma das referências contidas em uma OP (Ordem de Programação semanal), do número de peças a serem cortadas nas mesas de corte, de forma a satisfazer a demanda prevista para cada par *tamanho × cor* da referência. O número necessário de mesas de corte para satisfazer a demanda de uma dada referência também deriva da solução obtida para o PPOC.

O objetivo do problema consiste em definir os parâmetros supracitados de forma que a demanda de peças, prevista em nível semanal, possa ser atendida ao menor custo possível. Na contabilização desses custos, consideram-se as parcelas apresentadas a seguir.

3.1.2 Custos associados ao PPOC

3.1.2.1 Custo do corte

O custo do corte refere-se ao custo da operação de corte das peças de vestuário, sendo calculado em função de uma estimativa do tempo de operação da faca de corte e do custo de operação desta faca.

A estimativa do tempo de corte é realizada pelas indústrias em função do número de camadas de tecido a serem cortadas, multiplicado pelo perímetro de corte (estimado pelas indústrias em função do formato da peça e de seu respectivo tamanho). Com relação a este parâmetro, a indústria procura estimar um perímetro padrão para cada referência.

O tempo de corte estimado pelas indústrias (função do número de camadas de tecido e da soma do perímetro de corte de todas as peças que serão cortadas) é multiplicado pelo custo de operação da faca (medido em \$/tempo).

3.1.2.2 Custo do enfiesto

O custo do enfiesto está associado à operação de esticar o tecido sobre a mesa de corte.

O enfiado é custeado a partir do tempo estimado para sua realização, tempo este multiplicado pelo custo (medido em \$/tempo) de operação da máquina que realiza a operação de enfiado.

Os dados disponibilizados pelas indústrias indicam que o tempo de enfiado é estimado em função do número de camadas de tecido, bem como do comprimento de tecido a ser enfiado em uma mesa.

3.1.2.3 Custo da perda nas dobras

Quando se promove o dobramento do tecido para formar a próxima camada, há uma pequena parcela da mesma que não poderá ser aproveitada no corte das peças, representando dessa forma uma perda a ser contabilizada.

O custo da dobra é calculado em função do custo do tecido (medido em \$/kg), de sua gramatura (medida em kg/m²), do comprimento de dobra não aproveitado (em geral quatro centímetros), da largura do tecido e do número de camadas a serem enfiadas. Portanto, este custo varia linearmente em função do número de camadas definidas para a mesa de corte.

3.1.2.4 Custo da perda de aproveitamento do tecido

O custo da perda de aproveitamento do tecido refere-se ao desperdício entre as peças que são cortadas. A estimativa deste custo é feita normalmente em função do comprimento de tecido previsto para a mesa de corte. Quanto maior o comprimento, menor a perda de aproveitamento de tecido entre as peças, dado que as possibilidades de promover um melhor encaixe (disposição das peças ao longo da mesa de corte) são maiores.

Para quantificação deste custo, as indústrias têxteis podem construir uma tabela que associe o comprimento de enfiado ao aproveitamento de tecido estimado, em termos percentuais. Desta forma, esta parcela de custo é calculada em função da diferença entre o máximo aproveitamento possível (no entorno de 87%, segundo informações de especialistas neste segmento) e o que se estima de aproveitamento para o comprimento de tecido previsto.

A perda de aproveitamento estimada é multiplicada pelo custo do tecido (medido em \$/kg), gramatura (medida em kg/m²), largura (medida em m) e pelo comprimento de enfiado previsto para a mesa de corte (também medido em m), bem como pelo número de camadas a serem enfiadas.

Com relação a este custo, é importante salientar que, a rigor, o aproveitamento de tecido deveria depender também do formato da peça em questão (quanto mais irregular o formato, maiores as perdas no encaixe). No entanto, as indústrias têxteis em geral não consideram esta particularidade na estimativa do aproveitamento de tecido.

3.1.2.5 Custo do estoque de peças

Este custo refere-se ao valor desembolsado para se estocar uma quantidade de peças que não será comercializada imediatamente após sua confecção, mas que poderá atender uma demanda futura. A quantificação do custo de estoque é feita em função de um tempo médio de estoque, do custo da peça estocada e de uma taxa que agrega custos de oportunidade do capital, de seguros contra roubos e de manutenção de estoques.

3.1.2.6 Custo da sobra de peças

O custo da sobra refere-se ao prejuízo financeiro decorrente das peças que não foram vendidas durante a estação, e que desta forma têm de ser comercializadas a preços promocionais. Consiste numa perda de receita significativa para as indústrias de confecção, já que a margem de lucro embutida no preço das peças é menor que a taxa de desconto concedida nas promoções de fim de estação.

Esta parcela é calculada em função de uma estimativa do número de peças que sobraram (a diferença entre o número de peças produzidas e as demandas atual e futura), do preço das referidas peças e da taxa de desconto concedido nos preços promocionais.

Com base nos resultados obtidos para as referências analisadas, constatou-se que esta é a parcela mais significativa de todas as parcelas de custo e, desta forma, uma ordem de corte bem programada deve implicar em sobra reduzida.

3.1.2.7 Custo de *setup* na estamparia

Este custo é calculado em função do tempo para realização dos ajustes nas máquinas, necessários à execução da estamparia, quando da troca de cor de tecido em uma ordem de corte, além do tempo para execução do *setup* das máquinas de estamparia em função da mudança de referência. Estes tempos são multiplicados pelo custo de *setup* das máquinas, estimado em \$/minuto. Há ainda que se considerar o custo da tinta para realização de acertos neste *setup*.

A solução obtida para o PPOC pode ser influenciada pelo custo necessário para promover os ajustes supracitados, no processo de estamparia, associados à troca de cor do tecido.

No entanto, algumas referências possuem estampa (referências para as quais deve-se considerar este custo adicional), sendo que outras não a possuem. Nesta pesquisa, concebeu-se um modelo de resolução do PPOC para aquelas referências que não possuem estampa. A formulação do problema, apresentada a seguir, desconsidera o custo de *setup* de estamparia.

Além das parcelas de custo supracitadas, há que se considerar algumas limitações na obtenção de uma solução para o PPOC. Estas limitações encontram-se apresentadas a seguir.

3.1.3 Limitações associadas ao PPOC

As limitações associadas ao problema de programação de ordens de corte referem-se aos recursos disponíveis para efetuação dos cortes (recursos medidos em termos de comprimento máximo do enfiesto e comprimento da faca de corte), bem como à necessidade de se atender à demanda prevista para um intervalo de previsão de curto prazo. Desta forma, estas limitações foram divididas em duas categorias: limitações físicas de produção e limitações geradas pelo mercado.

3.1.3.1 Limitações físicas de produção

a) **Limitação quanto ao comprimento máximo de enfiesto**

Entende-se por comprimento de enfiesto o comprimento de tecido esticado na mesa de corte. Com relação ao comprimento máximo do enfiesto, a indústria define para este parâmetro um valor para cada referência da coleção, em função do comprimento da mesa que será utilizada.

Para cada mesa de corte cuja programação é realizada, há que se considerar esta limitação, quando da definição do número de marcações de cada tamanho a serem cortadas. Para isto, a indústria fornece também uma relação do comprimento da marcação para cada tamanho.

Desta forma, para cada mesa de corte, o somatório do número de marcações multiplicadas por seus respectivos comprimentos não pode ultrapassar o comprimento máximo de enfiesto pré-fixado.

b) Limitação quanto ao comprimento da faca de corte

Com relação a esta limitação, a indústria têxtil costuma fornecer um número máximo de camadas de tecido, para cada referência, compatível com o comprimento da faca de corte. Vale salientar que, apesar da faca de corte ter um comprimento pré-determinado, a espessura de tecido pode variar, motivo pelo qual o valor deste parâmetro é específico para cada referência.

Sendo assim, para cada mesa de corte o somatório do número de camadas associadas a cada cor não pode ultrapassar o número máximo de camadas previsto para a referência em questão.

3.1.3.2 Limitações geradas pelo mercado

Esta limitação prevê que as ordens de corte programadas contêm um número de peças suficiente para atendimento da demanda líquida (demanda prevista menos as peças em estoque) atual, tanto por tamanho como por cor da referência avaliada.

Com relação a este fato, vale lembrar a dificuldade deste segmento de indústria em prever a demanda de peças, principalmente na fase de lançamento das coleções. Como forma de imprimir maior flexibilidade ao modelo proposto, definiu-se um parâmetro que indica a proporção da demanda prevista que deverá ser atendida pela indústria, parâmetro este cujo valor pode ser definido pelo usuário.

A partir da identificação das parcelas de custo e das limitações inerentes ao PPOC, verificou-se que o mesmo poderia ser formulado como um problema de otimização combinatória. Encontra-se apresentada nos itens subseqüentes a formulação proposta para o PPOC.

3.2 VARIÁVEIS DE DECISÃO DO PPOC

Para que se possa definir o número de peças associadas a cada par *tamanho x cor*, há que se determinar o número de marcações de cada tamanho e de camadas de cada cor de tecido, em cada uma das mesas de corte previstas para atendimento à demanda de uma dada referência. Desta forma, as variáveis de decisão do problema são definidas como segue:

- NMA_{kt} = número de marcações do tamanho t a serem cortadas na mesa k ;
- NCA_{kc} = número de camadas referentes à cor c a serem enfiadas na mesa k .

As variáveis supracitadas são discretas e, desta forma, têm de estar associadas a valores inteiros e não-negativos.

O número necessário de mesas de corte para uma referência é consequência da resolução do problema, ou seja, as mesas k cujas respectivas variáveis NMA_{kt} e NCA_{kc} apresentarem valor maior que zero para ao menos um tamanho t e ao menos uma cor c estarão sendo consideradas na programação de ordens de corte da referência. Contudo, torna-se necessário definir com antecedência um limite máximo de número de mesas de corte.

3.3 FUNÇÃO-OBJETIVO DO PPOC

Conforme já explicitado anteriormente, o objetivo consiste em minimizar o somatório das parcelas de custo associadas ao PPOC, definidas no item 3.1.2. deste trabalho. Encontra-se apresentada a seguir a formulação matemática de cada uma das parcelas de custo.

3.3.1 Custo do corte

$$CCO = \sum_k TPC_k \times CPM \quad (3.3 \text{ a})$$

Onde:

CCO = Custo do corte, em \$;

TPC_k = Tempo padrão de corte para a mesa k , em minutos;

CPM = Custo padrão por minuto do corte/enfesto;

k = índice que especifica a mesa de corte.

Para contabilização deste custo, as indústrias podem construir uma tabela que relacione os tempos de corte ao perímetro total das peças cortadas e ao número de camadas de tecido. A partir da observação dos dados contidos em uma tabela fornecida por uma indústria deste segmento, constatou-se que a variação do tempo de corte era linear, tanto com relação ao perímetro total como com relação ao número de camadas. Desta forma, ajustou-se uma equação de regressão linear múltipla para estimativa do tempo de corte, obtendo-se com relação à mesma um R^2 (coeficiente de determinação) de 0,92.

3.3.2 Custo do enfiesto

$$CEN = \sum_k TPE_k \times CPM \quad (3.3 \text{ b})$$

Onde:

CEN = Custo do enfiesto, em \$;

TPE_k = Tempo padrão do enfiesto para a mesa k , em minutos;

CPM = Custo padrão por minuto do corte/enfiesto;

k = índice que especifica a mesa de corte.

Da mesma forma que para o parâmetro TPC definido no item anterior, pode-se construir uma tabela que relacione os tempos de enfiesto ao comprimento do mesmo e ao número de camadas de tecido. A partir da observação dos dados contidos em uma tabela fornecida por uma indústria do segmento, chegou-se à conclusão de que a variação dos tempos de enfiesto, tanto em função do comprimento como em função do número de camadas, era aproximadamente linear. A partir desta constatação, ajustou-se uma equação de regressão linear múltipla para correlacionar estes parâmetros, obtendo-se com relação à mesma um R^2 (coeficiente de determinação) de 0,88.

3.3.3 Custo da perda nas dobras

$$CDE = \sum_k \sum_c NCA_{kc} \times LARG \times CKM \times CDO \times GRAMAT \quad (3.3 \text{ c})$$

Onde:

CDE = Custo da perda nas dobras, em \$;

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

$LARG$ = Largura do tecido, em m;

CKM = Custo por kg do tecido, em \$/kg;

CDO = Comprimento da dobra, em m;

$GRAMAT$ = gramatura do tecido, em kg/m²;

k = índice que especifica a mesa de corte;

c = índice associado à cor do tecido.

3.3.4 Custo da perda de aproveitamento de tecido

$$CPAT = \sum_k \left[(PEP - PE_k) \times LARG \times \sum_t (NMA_{kt} \times COP_t) \times \sum_c NCA_{kc} \times GRAMAT \times CKM \right]$$

(3.3 d)

Onde:

$CPAT$ = Custo da perda de aproveitamento de tecido, em \$;

PEP = Percentual de aproveitamento para o encaixe padrão (aproveitamento máximo);

PE_k = Percentual de aproveitamento para a mesa de corte k ;

$LARG$ = Largura do tecido;

NMA_{kt} = Variável de decisão que define o número de marcações do tamanho t na mesa k ;

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

COP_t = Comprimento associado à marcação do tamanho t ;

$GRAMAT$ = Gramatura do tecido, em kg/m^2 ;

CKM = Custo por kg do tecido;

k = índice que especifica a mesa de corte;

c = índice associado à cor do tecido;

t = índice associado ao tamanho da peça.

3.3.5 Custo do estoque

$$CES = \sum_c \sum_t \left[\sum_k (NMA_{kt} \times NCA_{kc}) - DEM_{tc} \right] \times TMA \times PPE \times CP \quad (3.3 e)$$

Onde:

CES = Custo do estoque de peças, em \$;

NMA_{kt} = Variável de decisão que define o número de marcações do tamanho t na mesa k ;

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

DEM_{tc} = Demanda atual do tamanho t e da cor c , deduzido o número de peças que já se encontram em estoque;

TMA = Taxa mensal de armazenagem (incluindo-se aí o custo de oportunidade do capital, de seguro contra roubos e de manutenção de estoques), em %;

PPE = Período médio de permanência das peças em estoque, em meses;

CP = Custo da peça, em \$/unidade;

k = índice que especifica a mesa de corte;
 c = índice associado à cor do tecido;
 t = índice associado ao tamanho da peça.

3.3.6 Custo da sobra

$$CSP = \sum_t \sum_c \left[\sum_k (NMA_{kt} \times NCA_{kc}) - DEM_{tc} - DEMFUT_{tc} \right] \times PED \times PVP \quad (3.3 f)$$

Onde:

CSP = Custo da sobra de peças, em \$;

NMA_{kt} = Variável de decisão que define o número de marcações do tamanho t na mesa k ;

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

DEM_{tc} = Demanda atual do tamanho t e da cor c , deduzido o número de peças que já se encontram em estoque;

$DEMFUT_{tc}$ = Demanda futura do tamanho t e da cor c ;

PED = Percentual de desconto no preço das peças excedentes;

PVP = Preço de venda da peça, em \$/unidade;

k = índice que especifica a mesa de corte;

c = índice associado à cor do tecido;

t = índice associado ao tamanho da peça.

3.4 RESTRIÇÕES DO PPOC

Conforme já citado no item 3.1.3. deste trabalho, as limitações ou restrições do problema são de duas naturezas distintas: restrições físicas de produção (comprimento máximo de enfesto e espessura da faca de corte) e restrição de atendimento à demanda.

A formulação de cada um destes conjuntos de restrições encontra-se apresentada a seguir.

3.4.1 Restrições quanto ao comprimento máximo de enfesto

$$\sum_t NMA_{kt} \times COP_t \leq COE_{max} \quad \forall k = 1, \dots, K \quad (3.4 a)$$

Onde:

NMA_{kt} = Variável de decisão que define o número de marcações do tamanho t na mesa k ;

COP_t = Comprimento associado à marcação do tamanho t ;

COE_{max} = Comprimento máximo de enfiesto;

t = índice associado ao tamanho da peça;

k = índice que especifica a mesa de corte;

K = número máximo de mesas de corte, definido pelo usuário do modelo.

3.4.2 Restrições quanto à espessura da faca de corte

$$\sum_c NCA_{kc} \leq NCA_{max} \quad \forall k = 1, \dots, K \quad (3.4 b)$$

Onde:

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

NCA_{max} = Número máximo de camadas;

c = índice associado à cor do tecido;

k = índice que especifica a mesa de corte;

K = número máximo de mesas de corte, definido pelo usuário.

3.4.3 Restrições quanto à necessidade de atendimento à demanda

$$\sum_k NCA_{kc} \times NMA_{kt} \geq DEM_{tc} \times PERC \min \quad \forall t = 1, \dots, T \text{ e } \forall c = 1, \dots, C \quad (3.4 c)$$

Onde:

NCA_{kc} = Variável de decisão que define o número de camadas por cor c na mesa k ;

NMA_{kt} = Variável de decisão que define o número de marcações do tamanho t na mesa k ;

DEM_{tc} = Demanda atual do tamanho t e da cor c ;

$PERC \min$ = Percentual mínimo de atendimento à demanda (parâmetro opcional para indicar um nível mínimo de atendimento à demanda);

k = índice que especifica a mesa de corte;

c = índice associado à cor do tecido;

t = índice associado ao tamanho da peça;

C = número total de cores associado à referência;

T = número total de tamanhos associado à referência.

3.4.4 Restrições de não–negatividade e integridade

Além dos conjuntos de restrições apresentados anteriormente, é importante salientar que os valores associados às variáveis de decisão do problema não podem ser menores que zero e têm de ser inteiros. Desta forma, há que se considerar também as seguintes restrições:

$$NMA_{kt} \geq 0 \text{ e inteiro} \quad \forall k = 1, \dots, K \text{ e } \forall t = 1, \dots, T \quad (3.4 \text{ d})$$

$$NCA_{kc} \geq 0 \text{ e inteiro} \quad \forall k = 1, \dots, K \text{ e } \forall c = 1, \dots, C \quad (3.4 \text{ e})$$

3.5 FORMULAÇÃO COMPLETA DO PPOC

Com base nas formulações dos custos e restrições especificados nos itens 3.3. e 3.4., respectivamente, a formulação matemática completa do problema encontra-se apresentada a seguir:

$$\text{Minimizar } Z = \text{CCO} + \text{CEN} + \text{CDE} + \text{CPAT} + \text{CES} + \text{CSP}$$

s.a.:

$$\sum_t NMA_{kt} \times COP_t \leq \text{COE}_{\text{max}} \quad \forall k = 1, \dots, K$$

$$\sum_c NCA_{kc} \leq \text{NCA}_{\text{max}} \quad \forall k = 1, \dots, K$$

$$\sum_k NCA_{kc} \times NMA_{kt} \geq \text{DEM}_{tc} \times \text{PERC}_{\text{min}} \quad \forall t = 1, \dots, T \text{ e } \forall c = 1, \dots, C$$

$$\text{CCO} = \sum_k TPC_k \times \text{CPM} \quad (3.3 \text{ a})$$

$$\text{CEN} = \sum_k TPE_k \times \text{CPM} \quad (3.3 \text{ b})$$

$$\text{CDE} = \sum_k \sum_c NCA_{kc} \times \text{LARG} \times \text{CKM} \times \text{CDO} \times \text{GRAMAT} \quad (3.3 \text{ c})$$

$$\text{CPAT} = \sum_k \left[(\text{PEP} - \text{PE}_k) \times \text{LARG} \times \sum_t (NMA_{kt} \times \text{COP}_t) \times \sum_c NCA_{kc} \times \text{GRAMAT} \times \text{CKM} \right]$$

(3.3 d)

$$\text{CES} = \sum_c \sum_t \left[\sum_k (NMA_{kt} \times NCA_{kc}) - \text{DEM}_{tc} \right] \times \text{TMA} \times \text{PPE} \times \text{CP} \quad (3.3 \text{ e})$$

$$\text{CSP} = \sum_t \sum_c \left[\sum_k (NMA_{kt} \times NCA_{kc}) - \text{DEM}_{tc} - \text{DEM}_{\text{FUT}_{tc}} \right] \times \text{PED} \times \text{PVP} \quad (3.3 \text{ f})$$

$NMA_{kt} \geq 0$ e inteiro

$\forall k = 1, \dots, K$ e $\forall t = 1, \dots, T$

$NCA_{kc} \geq 0$ e inteiro

$\forall k = 1, \dots, K$ e $\forall c = 1, \dots, C$

Onde:

Z = valor da função-objetivo do problema;

CCO = Custo do corte;

CEN = Custo do enfiesto;

CDE = Custo da dobra de tecido;

CPAT = Custo da perda de aproveitamento de tecido;

CES = Custo da estocagem de peças;

CSP = Custo da sobra de peças.

Com relação à formulação apresentada, vale salientar que as parcelas de custos formuladas em (3.3 e) e (3.3 f), bem como o conjunto de restrições formulado em (3.4 c) são não-lineares. Além disso, a soma das parcelas de custos pode gerar uma função-objetivo não convexa. Encontra-se apresentado no Apêndice 06 um gráfico que busca ilustrar, através de um exemplo, a não-convexidade da função-objetivo do problema.

Com base no exposto acima, verifica-se que o PPOC é bastante complexo do ponto de vista matemático e, para sua resolução, torna-se necessário definir um método que alie qualidade da solução (medida em termos de valor da função-objetivo) ao custo computacional (medido em termos de tempo de processamento).

Com base nas características apresentadas e na alta complexidade do problema, buscou-se identificar os métodos aparentemente promissores para sua resolução. Estes métodos encontram-se apresentados no próximo capítulo.

4 TÉCNICAS PROMISSORAS DE RESOLUÇÃO DO PPOC

4.1 ASPECTOS INTRODUTÓRIOS

Com base na definição das variáveis de decisão do problema, apresentada no capítulo anterior, verifica-se que o PPOC faz parte da classe dos problemas de programação combinatória não linear inteira. De acordo com Pardalos (2002), problemas de programação inteira são especialmente difíceis de serem resolvidos, dispondo-se, atualmente, de métodos exatos e heurísticos para sua resolução. Segundo o autor, a escolha do método mais apropriado depende das particularidades do problema em questão, tais como número de variáveis envolvidas, características da função-objetivo e das restrições do problema.

McCarl e Spreen (1997) citam diferentes métodos que podem ser utilizados na resolução de problemas de programação inteira, destacando-se entre eles os Algoritmos de Cortes (ex.: Cortes de Gomory), Métodos de Enumeração (ex.: *Branch-and-Bound* e Balas), Algoritmos de Particionamento (ex.: Decomposição de Benders), Relaxação Lagrangeana e Processos de Busca em Grafos.

Staggemeier e Clark (2001) citam que as publicações atuais acerca dos problemas de programação inteira e inteira mista vêm dando ênfase especial aos métodos metaheurísticos⁵, tais como busca tabu, *simulated annealing* e algoritmos genéticos.

Encontram-se apresentadas a seguir as características básicas dos principais métodos disponíveis na literatura para resolução de problemas de programação inteira.

4.2 MÉTODOS EXATOS

Os algoritmos apresentados a seguir são apropriados para obtenção da solução exata para problemas cuja função-objetivo é convexa. Para problemas associados a funções não-convexas, ainda não se tem conhecimento de métodos que garantam a obtenção de uma solução exata.

Saliente-se também que alguns dos métodos apresentados a seguir (especificamente os algoritmos de planos de corte e o algoritmo de Balas) são mais apropriados para resolução de problemas cujas restrições sejam lineares, o que não se aplica ao PPOC. No entanto, julgou-se oportuno citá-los neste trabalho porque os mesmos poderiam ser utilizados em conjunção com

⁵ Estratégias gerais para concepção de procedimentos heurísticos com alta performance na resolução de problemas de otimização (GARCIA-LÓPEZ et al., 2003).

um algoritmo de relaxação, que adicionasse as restrições não-lineares à função-objetivo, suprimindo-as do conjunto de restrições do problema (esta abordagem será mais bem discutida nos itens subseqüentes).

4.2.1 Algoritmos de Planos de Corte

Segundo Zionts (1974), os algoritmos associados à idéia de planos de corte foram os primeiros desenvolvidos formalmente para resolução do problema de programação inteira. O autor chama de “corte” uma restrição adicionada ao problema com a função de “cortar” parte do conjunto de soluções viáveis, sem excluir qualquer solução inteira.

Seguindo esta filosofia, Dantzig (1963) sugeriu um procedimento que implica na adição, a cada iteração realizada, de restrições que façam com que a soma de variáveis não-básicas seja maior ou igual a um, no intuito de fazer com que os valores associados às variáveis básicas passem a ser inteiros. Tal procedimento, segundo o autor, seria repetido até a obtenção de uma solução inteira para o problema.

Zionts (1974, p. 343) salienta que “embora o procedimento proposto por Dantzig convirja para o ótimo em um número finito de iterações em vários exemplos, foi comprovado que o mesmo em geral não trabalha bem”. Com relação a este assunto, Salkin (1975) especifica as condições necessárias para que o algoritmo de Dantzig possa funcionar.

Há que se destacar as contribuições de Gomory no estudo de problemas de programação inteira, podendo-se citar o algoritmo inteiro de Gomory (GOMORY, 1963) e o algoritmo inteiro misto de Gomory (GOMORY, 1960).

Assim como no caso do algoritmo proposto por Dantzig (1963), os algoritmos propostos por Gomory podem demorar muito para atingir a solução ótima, dependendo das características do problema com o qual se está trabalhando.

Beale (1977) ressaltou que a performance dos algoritmos que envolvem planos de corte, medida em termos de custo computacional, fez com que os mesmos passassem a ser menos populares do que outros métodos para resolução do problema de programação inteira.

4.2.2 Algoritmo Branch-and-Bound

Segundo Maculan (1978), o algoritmo *Branch-and-Bound* foi o primeiro método de enumeração implícita⁶ a ser desenvolvido para resolução de problemas de programação inteira.

Originalmente introduzido por Land e Doig (1960), o algoritmo inicia com a obtenção de uma solução para o problema, desconsiderando as restrições de integridade. A partir daí, escolhe-se uma variável cujo valor obtido na solução seja fracional e consideram-se duas possibilidades:

- I. Adicionar uma restrição que impeça o valor desta variável de exceder o menor inteiro mais próximo ao valor da mesma na solução anterior;
- II. Adicionar uma restrição que impeça o valor desta variável de ser menor que o maior inteiro mais próximo ao valor da mesma na solução anterior.

O algoritmo repete esses passos até que uma solução inteira viável seja encontrada, sendo a mesma armazenada em uma lista. Por considerar todos os “ramos” criados a partir das possibilidades descritas em (I) e (II), a cada estágio do algoritmo são resolvidos mais dois problemas que envolvem iterações duais. A quantidade de ramos gerada a partir do algoritmo normalmente implica em alto custo computacional. De acordo com Zionts (1974), este método trabalha bem quando se consegue obter cedo uma boa solução inteira. Saliente-se que, desde a sua concepção na década de 60, já foram propostas muitas melhorias ao algoritmo com o intuito de reduzir a desvantagem de alto custo computacional.

4.2.3 Algoritmo de Balas

O algoritmo de Balas (1965) é classificado como um método de enumeração implícita, adequado para resolução de problemas de programação inteira zero-um. Considerou-se oportuno apresentá-lo neste trabalho (mesmo atentando para o fato de que o PPOC não é originalmente um problema de programação inteira zero-um), devido à constatação de que qualquer problema de programação inteira pode ser transformado em um problema de programação inteira binária (SIMONNARD, 1973).

⁶ Método de resolução de problemas de programação inteira que enumera todas as possíveis soluções, examinando apenas aquelas potencialmente melhores do que uma solução já encontrada até o momento e descartando as demais.

Segundo Zionts (1974), o algoritmo de Balas permite explorar de modo eficaz o conjunto de soluções viáveis do problema, através do exame de cada uma delas no máximo uma vez, obtendo a solução ótima (se ela existir) num número finito de iterações. Vale lembrar, no entanto, que o número de soluções possíveis é de 2^n , sendo n o número de variáveis de decisão do problema.

A exploração de soluções, a partir de uma solução inicial S_0 ⁷, utiliza:

- I. Regras de escolha entre as diversas possibilidades de se fazer as variáveis do problema iguais a zero ou um, respectivamente;
- II. Regras de parada sobre uma solução já avaliada anteriormente.

Como desvantagem do método, pode-se citar o fato de que quando o mesmo é utilizado para resolução de problemas cujos coeficientes associados às variáveis na função-objetivo têm valores muito próximos, pode ser necessário um número muito grande de iterações para obtenção da solução exata.

Segundo McCarl e Spreen (1997), muitos problemas de programação inteira são de difícil resolução a partir dos métodos exatos supracitados. Tais dificuldades, que implicam em alto custo computacional, podem ser reduzidas a partir da utilização de diferentes heurísticas, que não garantem a obtenção da solução ótima, mas podem auxiliar na obtenção de uma boa solução com um tempo de processamento significativamente menor. Além disso, o fato da função-objetivo associada ao PPOC não ser convexa induz à utilização de métodos heurísticos para sua resolução.

Encontram-se apresentados no próximo item dois métodos que utilizam procedimentos heurísticos e que foram amplamente divulgados na bibliografia sobre o assunto.

4.3 MÉTODOS HEURÍSTICOS

4.3.1 Algoritmo Genético

O algoritmo genético, inicialmente concebido por Holland (1975), consiste num método metaheurístico de resolução de problemas de otimização combinatória, tendo sido criado a partir da observação dos processos de evolução na natureza (KOZA; RICE, 1992). Para implementá-lo, cria-se inicialmente uma população de indivíduos, cada um deles

⁷ A solução inicial considera todas as variáveis iguais a zero, pelo fato da formulação padrão consistir num problema de minimização. Para utilização do algoritmo de Balas, todos os coeficientes da função-objetivo têm de ser maiores ou iguais a zero (SIMONNARD, 1973).

representando uma possível alternativa de solução para o problema em estudo. Cada um destes indivíduos é codificado em um *string* ou cromossomo.

A partir desta população inicial, a cada iteração são selecionados dois indivíduos (denominados pais) que, através de cruzamentos, gerarão outros indivíduos denominados filhos. Um *crossover* (cruzamento) consiste na união de partes de ambos os pais para formação de novos indivíduos, sendo que as regras de fragmentação dos indivíduos pais, bem como a forma de união dos fragmentos, são definidas especificamente para cada problema a ser resolvido. Além da operação de *crossover* no algoritmo genético, realizam-se também operações denominadas mutações, que têm como objetivo promover melhorias em algumas características dos indivíduos gerados a partir dos cruzamentos, com o intuito de melhorar a função *fitness*⁸ associada aos mesmos. Além disso, segundo Miranda (1997), as mutações podem evitar que o algoritmo convirja muito cedo para um mínimo local.

Desta forma, as operações básicas de um Algoritmo Genético são as seguintes:

- I. INICIALIZAÇÃO – Uma população de n indivíduos é gerada aleatoriamente;
- II. CÁLCULO DO *FITNESS* – Geralmente o *fitness* (aptidão) do indivíduo é determinado através do cálculo da função-objetivo, que depende das especificações do problema;
- III. SELEÇÃO – Nesta fase os indivíduos mais aptos da geração atual são selecionados. Estes indivíduos geram “filhos” a partir das operações de *crossover*. Segundo Miranda (1997), cada indivíduo tem uma probabilidade de ser selecionado em função de seu respectivo *fitness*;
- IV. *CROSSOVER* (Cruzamento) – Os indivíduos selecionados na etapa anterior geram outros indivíduos a partir de regras pré-especificadas;
- V. MUTAÇÃO – A operação de mutação é realizada para aprimorar características dos indivíduos recém gerados e evitar que o algoritmo convirja muito cedo para um mínimo local.

A estrutura básica de funcionamento do Algoritmo Genético pode ser ilustrada a partir da figura a seguir:

⁸ Função que exprime, quantitativamente, a capacidade de um indivíduo competir em seu ambiente.

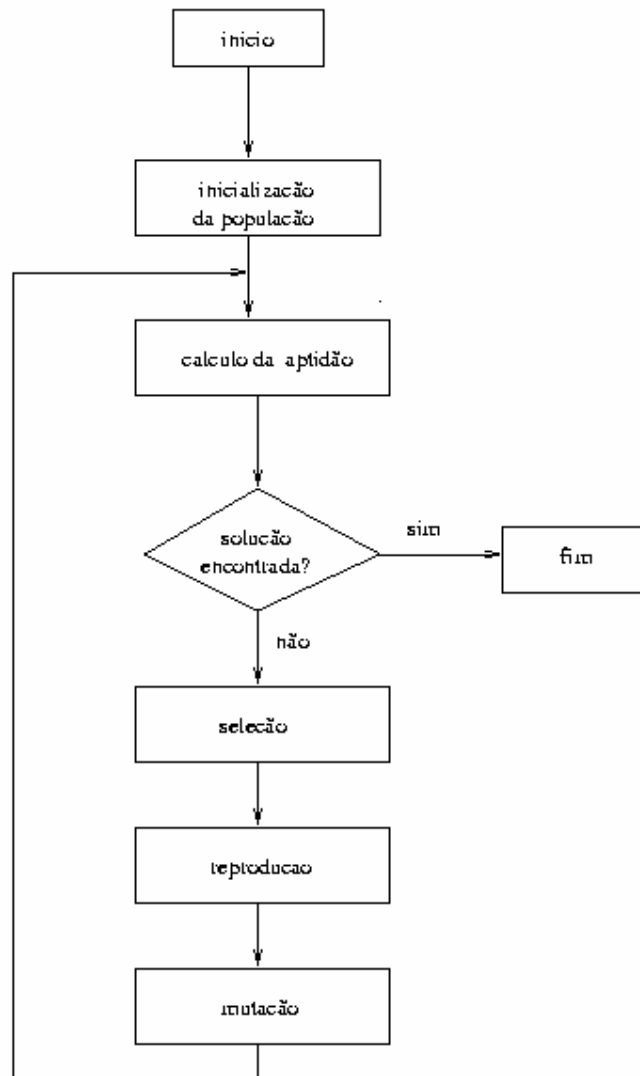


FIGURA 08: Estrutura de um algoritmo genético

Fonte: Miranda (1997)

De acordo com a bibliografia disponível sobre o assunto, a utilização do algoritmo genético vem se mostrando adequada à resolução de alguns problemas logísticos clássicos, tais como o problema de roteirização de veículos (McCARL; SPREEN, 1997). Obtiveram-se bons resultados com a utilização do algoritmo genético na otimização do leiaute de facilidades interligadas por corredores (IVANQUI, 1997). Oliveira (1999) também avaliou a utilização do algoritmo genético na resolução do problema de cobertura de conjuntos.

4.3.2 Relaxação Lagrangeana

O método de relaxação lagrangeana, consagrado por Geoffrion (1974), consiste em um procedimento no qual alguma(s) das restrições (é) são inserida(s) na função-objetivo, utilizando uma abordagem inspirada nos multiplicadores de Lagrange (McCARL; SPREEN, 1997).

Segundo os autores supracitados, o método de relaxação lagrangeana tem sido utilizado basicamente com duas finalidades:

- I. Melhorar a performance, medida em termos de tempo de processamento, das soluções obtidas;
- II. Desenvolver modelos através dos quais se possa obter diretamente, ou através de heurísticas, soluções viáveis para o problema (FISHER, 1981).

Segundo McCarl e Spreen (1997), o segredo do sucesso da implementação do método de relaxação lagrangeana consiste na escolha correta das restrições que devem ser relaxadas, bem como na adoção de um bom algoritmo para incrementar o valor dos multiplicadores de Lagrange a cada iteração.

Para entender a filosofia do método, imagine-se um problema (P) formulado como segue:

$$(P) = \min_x \{cx\}$$

$$s.a. Ax \leq b$$

$$Dx \leq e$$

$$x \geq 0 \text{ e inteiro.}$$

De acordo com Espejo e Galvão (2002), uma relaxação lagrangeana de (P) é obtida multiplicando o lado esquerdo de um conjunto de restrições do tipo $Ax - b \leq 0$ por um vetor u de multiplicadores de Lagrange de sinal apropriado, adicionando-se este produto à função-objetivo do problema. A relaxação lagrangeana (P_u) do problema (P) é dada por:

$$(P_u)$$

$$v(P_u) = \min_x \{cx + u(Ax - b)\}$$

$$s.a. Dx \leq e$$

$$x \geq 0 \text{ e inteiro.}$$

Os autores supracitados observam que o problema lagrangeano é um problema em x , resolvido para um dado vetor fixo u . Para a formulação apresentada anteriormente, deve-se escolher $u \geq 0$ para garantir que $v(P_u)$ seja um limite inferior para (P) .

O primeiro trabalho acerca da utilização da relaxação lagrangeana foi proposto por Held e Karp (1971), que apresentaram uma abordagem para resolução do problema do caixeiro viajante a partir de um método lagrangeano. No entanto, foi Geoffrion (1974) quem atribuiu ao método o nome de relaxação lagrangeana, em seu trabalho sobre a utilização desta abordagem na resolução de problemas de programação inteira.

Naquela ocasião, Geoffrion (1974) definiu o problema de relaxação lagrangeana como um problema de minimização, que teria de atender aos seguintes requisitos:

- O conjunto de soluções viáveis, associado ao problema relaxado (P_u) deveria conter o conjunto de soluções viáveis do problema original (P) ;
- Sobre o conjunto de soluções viáveis do problema original (P) , o valor da função-objetivo para o problema relaxado (P_u) domina (é melhor que) o valor correspondente àquela do problema original.

Em outras palavras e segundo aquele autor, o valor da função-objetivo para o problema relaxado é uma versão “otimista” do valor para o problema original. Geoffrion (1974) ainda cita que o papel da relaxação lagrangeana consiste em promover limites para os valores ótimos de problemas difíceis, já que as soluções inviáveis para o problema original podem freqüentemente ser utilizadas como pontos iniciais (guias) para heurísticas concebidas especificamente para o problema em estudo.

Para um problema original de minimização, a resolução do problema lagrangeano possibilita a estimativa de um limite inferior para o valor da solução ótima, limite este associado a uma solução inviável, mas que a partir de iterações sucessivas tende a se aproximar da viabilidade.

O método consiste na realização de iterações sucessivas onde, de um lado, resolve-se o problema dual e obtém-se um limite inferior para a solução ótima do problema original. De outro lado e normalmente através de métodos heurísticos, viabiliza-se a solução encontrada para o problema dual e obtém-se um limite superior para a solução ótima do problema original. A cada iteração, definem-se, com base em algum método de otimização, novos valores para o vetor u de multiplicadores de Lagrange. Atingido algum dos critérios de parada especificados previamente, considera-se como solução para o problema original a solução viável com mais baixo custo encontrada até o momento (GEOFFRION, 1974).

Na ocasião em que publicou o artigo, Geoffrion (1974) buscou associar a relaxação lagrangeana ao algoritmo *branch-and-bound* e a outros algoritmos de enumeração implícita, com o intuito de acelerar a convergência dos mesmos. No entanto, o autor já chamou atenção para a possibilidade de utilização do método em outras aplicações.

Destacam-se também os trabalhos de Fisher (1981) e Parker e Radin (1988), que conseguiram demonstrar que a função-objetivo do problema relaxado (Pu) é uma função linear por partes, contínua e côncava, mas geralmente não diferenciável no ponto ótimo. Esta propriedade da relaxação lagrangeana faz com que ela seja uma proposta atraente para obtenção de limites da solução do problema original (ESPEJO; GALVÃO, 2002).

Em seu trabalho, Fisher (1981) também colocou algumas questões que, em sua visão, deveriam ser mais bem estudadas para propiciar uma melhor utilização deste método. Merecem destaque as seguintes questões colocadas pelo autor:

- Como selecionar valores apropriados para os multiplicadores de Lagrange u (associados às restrições relaxadas), de modo a garantir a convergência do método?
- Como podem os limites inferiores e superiores propiciados pelo método de relaxação lagrangeana serem integrados ao algoritmo *branch-and-bound* com o intuito de acelerar sua convergência?

As questões colocadas por este autor inspiraram vários estudos sobre o método de relaxação lagrangeana ao longo das duas últimas décadas.

Christofides e Beasley (1983) utilizaram a relaxação lagrangeana aliada ao método do subgradiente na obtenção de limites inferiores para o problema de localização capacitada. Saliente-se que a utilização combinada dos métodos de relaxação lagrangeana e do subgradiente foi considerada promissora por Christofides e Beasley (1983), Galvão e Raggi (1989) e Beasley (1993).

Lorena e Narciso (2000) propuseram a utilização de heurísticas de relaxação para resolução do problema de atribuição generalizada (*Generalized Assignment Problem – GAP*), utilizando para incremento dos multiplicadores de Lagrange, a cada iteração, o método do subgradiente generalizado.

A literatura acerca da evolução e das possíveis aplicações do método de relaxação lagrangeana é bastante vasta. No presente trabalho, procurou-se citar algumas das referências mais relevantes, que apresentaram inovações significativas. Não foi encontrada nenhuma publicação que envolvesse a utilização do método para problemas de otimização cuja função-objetivo fosse não-convexa.

Ao final deste capítulo, são discutidos os resultados obtidos a partir das tentativas de se utilizar algoritmos genéticos e relaxação lagrangeana, respectivamente, para a resolução do PPOC. Em função dos resultados obtidos com a utilização desses dois métodos não terem sido satisfatórios, avaliou-se a possibilidade de utilizar o método de busca em grafos, a ser explicitado no próximo item.

4.4 MÉTODO DE BUSCA EM GRAFOS

4.4.1 Conceitos Básicos

Segundo Papadimitriou e Steiglitz (1998, p. 20), “um grafo G é um par $G = (V, E)$, onde V é um conjunto finito de *nós* ou *vértices* e E tem, como elementos, subconjuntos de V de cardinalidade dois denominados *arestas*”.

A figura a seguir ilustra a representação de um grafo contendo seis nós e sete arestas.

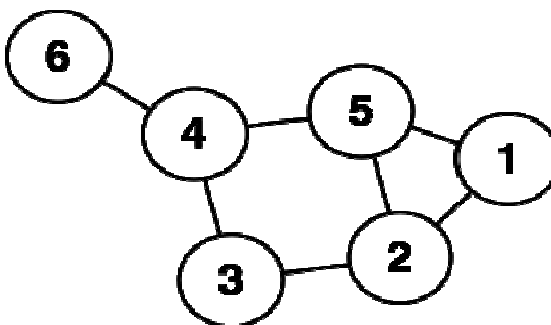


FIGURA 09: Exemplo de um grafo

Quando uma aresta apresenta uma direção definida na conexão entre dois nós, ela é chamada de *arco*.

A cada arco que une dois nós quaisquer n_1 e n_2 , associa-se um custo $c(n_1, n_2) \geq 0$. Um caminho é uma seqüência de nós $n_i = (n_1, n_2, n_3, \dots)$, ligados dois a dois por arcos, sendo seu respectivo custo calculado como a soma dos custos de cada um dos arcos que o compõem (FIGUEIREDO; GONZAGA, 2003). Papadimitriou e Steiglitz (1998) chamam a atenção para uma classe de grafos denominada *árvore*. Uma árvore, segundo os autores, é um grafo conexo⁹ sem ciclos¹⁰.

⁹ Grafo que sempre possui um caminho entre qualquer par de seus respectivos nós.

Um grafo pode representar uma série de problemas reais, tais como um mapa de estradas, redes PERT (*Program Evaluation and Review Technique*) no âmbito de planeamento de projetos, redes de computadores, dentre outros. A partir da modelagem do problema como um grafo, os esforços devem se concentrar na obtenção de um processo matemático que permita caminhar pelos nós e arcos deste grafo.

Em algumas situações, os nós e arcos do grafo que se pretende avaliar não são conhecidos a priori. Recorre-se então à utilização de um operador sucessor Γ , que associa a cada nó n o conjunto de nós $\Gamma(n)$ (sucessores de n), tais que $n_j \in \Gamma(n)$ se e somente se (n, n_j) é um arco. Seguindo esta idéia, um grafo pode ser construído recursivamente, a partir de um nó inicial, através de aplicações sucessivas de Γ (FIGUEIREDO; GONZAGA, 2003).

Encontram-se apresentados a seguir os principais conceitos associados a processos de busca em grafos.

4.4.2 Processos de busca em grafos

4.4.2.1 Aspectos introdutórios

Segundo Nilsson (1971), pode-se obter soluções para problemas reais através da aplicação de operadores sucessores a uma solução inicial, até que se atinja uma solução que seja considerada adequada. “A linguagem dos grafos é extremamente útil na descrição eficiente de estratégias de busca no espaço de estados” (NILSSON, 1971, p. 43).

Um nó inicial está associado a uma proposta inicial de solução, ou à configuração atual do problema que se pretende resolver. A aplicação de um operador sucessor Γ a um determinado nó é chamada de *expansão* deste nó. Este processo continua até que se encontre um nó com as características procuradas.

Quando o objetivo pretendido é a obtenção de um caminho, são utilizados *apontadores*, que têm a função de indicar o nó antecessor de cada um dos nós do grafo.

Além dos conceitos supracitados, a especificação completa de um processo de busca descreve a ordem na qual os nós devem ser expandidos (PRICE, 1971).

Nilsson (1971) define diferentes classes para os processos de busca em grafos, a serem apresentadas nos próximos itens.

¹⁰ Caminho sem nós repetidos, exceto o primeiro e o último.

4.4.2.2. Busca em largura ou horizontal pura

Segundo Nilsson (1971, p. 45), “o método de busca em largura (*breadth-first method*) expande nós na ordem em que os mesmos foram gerados”. O autor define o seguinte algoritmo básico para o processo de busca em largura:

- I. Coloque o nó inicial em uma lista de nós denominada LISTA DE ABERTOS;
- II. Se LISTA DE ABERTOS estiver vazia, pare sem ter obtido uma solução para o problema; senão, continue;
- III. Remova o primeiro nó da LISTA DE ABERTOS e coloque-o na LISTA DE FECHADOS; chame este nó de nó n ;
- IV. Expanda o nó n , gerando todos os seus sucessores. Se não houver sucessores, volte para (II). Coloque os sucessores no final da LISTA DE ABERTOS e providencie apontadores para tais sucessores que apontem para n ;
- V. Se qualquer dos sucessores tiver as características da solução de interesse, pare com sucesso. A partir dos apontadores, é possível recuperar o caminho que conduziu à solução do problema; caso contrário, volte a (II).

Christofides (1975) salienta que no método de busca em largura, os procedimentos de expansão ocorrem de nível em nível. Desta forma, o problema inicial P_0 é particionado em subproblemas P_1, P_2, \dots, P_k no nível 1, sendo cada um deles investigado antes que o processo de busca passe para o nível 2, e assim sucessivamente.

A figura apresentada a seguir ilustra a filosofia do processo de busca em largura apresentado por Christofides (1975, p. 393).

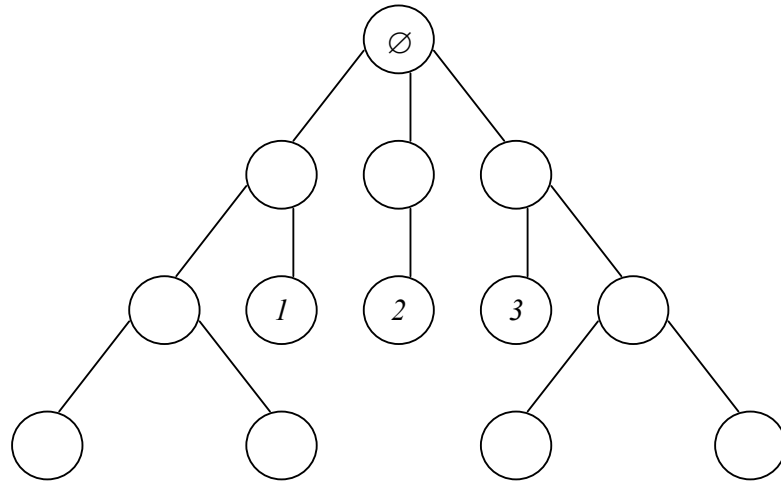


FIGURA 10: Processo de busca em largura

Fonte: Adaptado de Christofides (1975, p. 393)

\circledast : Solução do Problema, onde n é a ordem de obtenção.

Nilsson (1971) chama a atenção para o fato de que podem existir problemas cujo objetivo não seja o de encontrar a solução na menor sequência de passos, mas sim encontrar uma solução de mínimo custo. Esta situação será mais bem detalhada posteriormente.

Cormen et al. (2000) alertam para o fato do método de busca em largura ser impraticável para problemas muito grandes, devido ao grande número de iterações que podem ser necessárias até se obter a solução.

Processos de busca horizontal podem ser utilizados para resolver uma série de problemas, tais como a obtenção de todos os componentes conectados em um grafo, a obtenção do menor caminho entre dois nós u e v , dentre outros (CORMEN et al., 2000).

4.4.2.3 Busca em profundidade ou vertical pura

No método de busca em profundidade (*depth-first method*), procura-se expandir, a cada iteração, o último nó gerado. Segundo Nilsson (1971), a profundidade do nó inicial é zero. A profundidade de qualquer outro nó é calculada como sendo a profundidade do nó que o gerou mais um.

O autor ainda cita que a política de expandir o nó mais profundo a cada iteração pode conduzir à obtenção de um caminho promissor. “Depois que o processo gera um nó que exceda um dado limite de profundidade, expande-se o nó mais profundo que não exceda este limite, e assim sucessivamente” (NILSSON, 1971, p. 49).

O método de busca em profundidade pode ser implementado através do seguinte algoritmo, proposto por Nilsson (1971, p. 50):

- I. Coloque o nó inicial na LISTA DE ABERTOS;
- II. Se LISTA DE ABERTOS estiver vazia, pare (sem ter obtido uma solução); senão, continue;
- III. Remova o primeiro nó da LISTA DE ABERTOS e coloque-o na LISTA DE FECHADOS. Chame este nó de n ;
- IV. Se a profundidade de n for igual ao limite de profundidade pré-definido, volte a (II); senão, continue;
- V. Expanda o nó n , gerando todos os seus sucessores. Coloque-os no início da LISTA DE ABERTOS e defina apontadores para os mesmos que retornem a n ;
- VI. Se qualquer dos sucessores for identificado como uma solução do problema, pare: a solução foi obtida e o caminho pode ser recuperado através dos apontadores; caso contrário, volte para (II).

Christofides (1975) salienta a necessidade de se fazer um retorno (*backtracking*) após a obtenção de uma solução pelo método de busca em profundidade, reiniciando-se o processo de busca a partir do último nó que não exceda o limite de profundidade da solução recém-obtida.

A figura a seguir busca ilustrar o processo de busca em profundidade apresentado por Christofides (1975, p. 393).

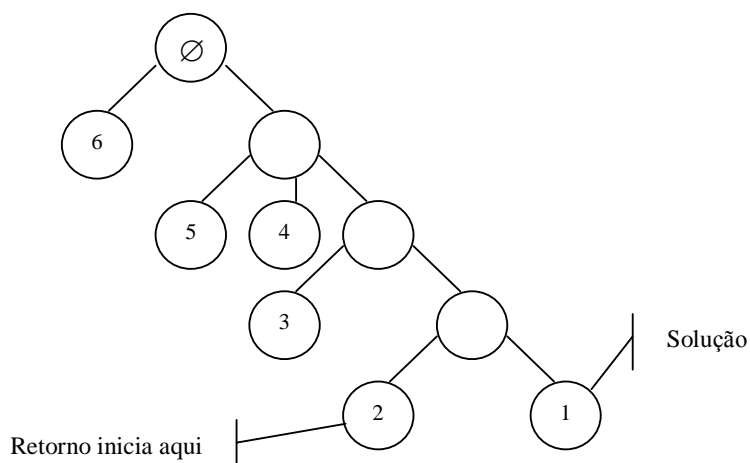


FIGURA 11: Processo de busca em profundidade

Fonte: Adaptado de Christofides (1975, p. 393)

O grafo ilustrado na figura 11 representa um processo de busca em profundidade. Os números associados a cada nó indicam a ordem de prioridade para efeitos de investigação da solução. A partir de cada um dos nós enumerados, reinicia-se o processo de busca até que não haja mais nós na LISTA DE ABERTOS.

Segundo Cormen et al. (2000), as aplicações deste processo assemelham-se àquelas já citadas quanto ao processo de busca em largura.

Nilsson (1971) coloca que ambos os processos, o de busca em largura e o de busca em profundidade, são métodos exaustivos que podem culminar em um tempo computacional muito alto, dependendo do tamanho do problema. Visando solucionar este entrave, pode ser utilizado o método de busca “gulosa” (*greedy algorithm*).

Segundo Helman, Moret e Shapiro (1993), um método de busca gulosa para obtenção do menor caminho é um processo iterativo no qual, a cada iteração, escolhe-se o melhor nó (de acordo com algum critério pré-definido) para ser expandido, procedendo-se assim até a obtenção de uma solução. Evidentemente, quanto melhor o critério de escolha do nó para expansão, maior a precisão da solução obtida.

Os autores supracitados apresentam, como um exemplo de aplicação do método de busca gulosa, a seguinte estratégia na resolução do problema do caixeiro viajante: “a cada estágio, visite a cidade ainda não visitada mais próxima da cidade atual”.

Em função do PPOC poder envolver um grande número de nós e, conseqüentemente, um alto custo computacional, considera-se o método de busca gulosa uma alternativa promissora de resolução do problema. Este assunto será abordado no próximo capítulo.

4.4.2.4 Outros critérios de ordenação de nós no processo de busca

Em grande parte dos problemas reais, o objetivo consiste em obter uma solução de mínimo custo, e não uma solução com o menor número de expansões. Com base nesta idéia, concebeu-se o método de custo uniforme (*uniform-cost method*), discutido por Nilsson (1971). Conforme poderá ser visto mais adiante, o algoritmo de Dijkstra (1959) enquadra-se nesta categoria.

Para utilização do método de custo uniforme, há que se armazenar o custo do menor caminho entre o nó inicial e cada um dos nós de um grafo. Estes custos são armazenados em um vetor denominado pelo autor de $g(n)$.

Nilsson (1971, p. 48) apresenta em sua publicação o seguinte algoritmo para implementação do método de custo uniforme:

- I. Coloque o nó inicial s na LISTA DE ABERTOS. Faça $g(s) = 0$;
- II. Se LISTA DE ABERTOS estiver vazia, pare sem ter obtido a solução; caso contrário, continue;
- III. Remova o nó da LISTA DE ABERTOS associado ao menor valor de g e coloque-o na LISTA DE FECHADOS. Chame este nó de n (no caso de empate entre dois ou mais nós, escolha n de maneira arbitrária, mas sempre visando a obtenção da solução do problema);
- IV. Se n for a solução do problema, pare e recupere o caminho a partir dos apontadores dos nós; caso contrário, continue;
- V. Expanda o nó n , gerando todos os seus sucessores [não havendo sucessores, retorne imediatamente a (II)]. Para cada sucessor n_i , compute $g(n_i) = g(n) + c(n, n_i)$. Insira esses sucessores na LISTA DE ABERTOS, associando aos mesmos os valores g anteriormente computados, e providencie apontadores que apontem para n ;
- VI. Volte para (II).

Nilsson (1971) cita também a possibilidade de que sejam utilizadas funções de avaliação no processo de ordenação de nós para expansão. O autor sugere que seja quantificada uma função que vise estimar o custo do menor caminho entre um nó n e a solução do problema, idéia esta que deu origem ao algoritmo A*, a ser apresentado posteriormente.

Os processos descritos anteriormente dizem respeito aos métodos clássicos de busca em grafos disponíveis na literatura. Encontram-se apresentados a seguir dois dos algoritmos de busca em grafos considerados mais eficientes na obtenção de caminhos mínimos.

4.4.3 Algoritmo de Dijkstra

Segundo Christofides (1975), o algoritmo mais eficiente, dentre os que não utilizam informação heurística para obtenção do menor caminho entre dois nós s e t , foi concebido por Dijkstra (1959).

O algoritmo de Dijkstra é baseado na atribuição de um limite superior de custo (inicialmente definido como temporário) do menor caminho entre o nó inicial s e cada um dos nós n do grafo de interesse. A cada iteração, obtém-se o custo exato do menor caminho entre s e um dos nós n do grafo, sendo este custo assinalado como um custo permanente. Encontram-se apresentados a seguir os passos do algoritmo de Dijkstra aplicado a grafos contendo apenas arcos com custos maiores que zero.

- **Inicialização**

- I. Inicialize um vetor $l(x_i)$ para guardar o limite superior de custo do caminho entre s e cada nó x_i do grafo;
- II. Faça $l(s) = 0$ e assinale este limite de custo como permanente. Faça $l(x_i) = \infty$ para todos os nós $x_i \neq s$ e assinale esses limites de custo como temporários. Faça $p = s$.

- **Atualização dos limites superiores de custo**

- III. Para todos os nós $x_i \in \Gamma(p)$ e que tenham limites de custo temporários, atualize os limites de custo da seguinte maneira:

$$l(x_i) = \min[l(x_i), l(p) + c(p, x_i)]$$

- **Fixação de um limite de custo como permanente**

- IV. De todos os nós cujos limites de custo sejam temporários, encontre x_i^* para o qual $l(x_i^*) = \min[l(x_i)]$;
- V. Assinale o limite de custo de x_i^* como permanente e faça $p = x_i^*$;
- VI. 1. Caso seja desejado somente o menor caminho entre s e t :
 - Se $p = t$, $l(p)$ é o custo do caminho de interesse. Pare.
 - Se $p \neq t$, volte a (III).
2. Caso se queira encontrar o menor caminho entre s e todos os nós do grafo

- Se todos os nós têm seus respectivos limites de custo declarados como permanentes, então estes limites são os custos dos menores caminhos. Pare.
- Se alguns nós têm seus respectivos limites de custo ainda declarados como temporários, volte a (III).

Devido à sua eficiência e relativa facilidade de implementação, Cormen et al. (2000) citam que o algoritmo de Dijkstra é aplicado frequentemente a problemas envolvendo redes de computadores (percurso entre roteadores), em problemas de tráfego urbano e de roteirização de veículos, dentre outros. Este algoritmo também pode ser utilizado de maneira indireta na resolução de problemas de fluxo em redes (PRICE, 1971).

Papadimitriou e Steiglitz (1998) citam que cada iteração no algoritmo de Dijkstra requer um número de passos proporcional ao número de nós cujos respectivos limites de custo do menor caminho ainda sejam definidos como temporários. Considerando que haja n iterações, sendo n o número de nós do grafo, o algoritmo requer no máximo um tempo proporcional a n^2 .

4.4.4 Algoritmo A*

Conforme já mencionado, os métodos de busca horizontal e vertical pura podem implicar em um alto custo computacional para obtenção da solução para grafos contendo um grande número de nós. O mesmo pode ocorrer com o algoritmo de Dijkstra.

Com base neste fato, Nilsson (1971) propôs a utilização de uma função $\hat{f}(n)$, que consiste na soma do custo $g(n)$ – custo do menor caminho entre o nó inicial e o nó n – com uma estimativa (calculada através de um método heurístico) do custo do menor caminho entre n e a solução do problema. Ou seja: $\hat{f}(n) = g(n) + \hat{h}(n)$, onde $\hat{h}(n)$ é uma estimativa do custo entre n e a solução.

Calculado o valor deste parâmetro para cada um dos nós ao longo do processo de busca, escolhe-se para expansão, a cada iteração, o nó associado ao menor valor de $\hat{f}(n)$. A este algoritmo que usa o parâmetro $\hat{f}(n)$ na ordenação de nós para expansão dá-se o nome de A*.

Nilsson (1971, p. 59) salienta que “se $\hat{h}(n)$ é um limite inferior de $h(n)$, então A* é admissível”, ou seja, ao longo do processo de busca sempre haverá ao menos um nó na LISTA DE ABERTOS, pertencente ao caminho ótimo, com um custo $\hat{f}(n)$ menor que o

custo real $f(s)$, sendo s o nó inicial. No entanto, há que se considerar que quanto menor o valor estimado para $\hat{h}(n)$, maior o tempo de processamento até a obtenção da solução, já que nesse caso o processo de busca tende a se aproximar do método de custo uniforme, apresentado em 4.4.2.4..

Com relação à escolha da heurística para estimativa dos valores de $\hat{h}(n)$, Nilsson (1971, p. 68) salienta a necessidade de se ponderar sobre os seguintes aspectos:

- 1) A importância da precisão da solução obtida;
- 2) Eventuais restrições quanto ao número de nós expandidos, até que se obtenha a solução;
- 3) Eventuais restrições quanto ao custo computacional associado à obtenção da solução.

No caso de haver limitações quanto ao tempo de processamento máximo na obtenção de uma solução, pode-se utilizar a estratégia de superestimar valores para $\hat{h}(n)$, o que conduz à diminuição do número de nós expandidos. O algoritmo que utiliza estimativas para $\hat{h}(n)$ maiores que os reais valores de $h(n)$ é referenciado por alguns autores como \hat{A} (A – chapéu).

Christofides (1975, p. 395) cita em sua publicação a possibilidade de utilizar uma “função de expansão” (*branching function*) na escolha do nó a ser expandido. Esta função estaria associada à probabilidade de que um dado nó da LISTA DE ABERTOS conduza à solução do problema.

Segundo Cormen et al. (2000), o algoritmo A^* tem sido citado como um exemplo do processo de busca ordenada pelo melhor nó (*best-first search*).

Na literatura acerca do assunto, encontram-se apresentadas várias aplicações do algoritmo A^* na resolução de problemas reais, destacando-se entre elas a utilização do referido algoritmo na resolução de jogos (NILSSON, 1971), na locação de torres de transmissão (FIGUEIREDO; GONZAGA; MACULAN, 1979), no alinhamento de múltiplas seqüências de proteína (KELLY e LABUTE, 1996) e na montagem da estrutura de bibliotecas (BATTEN, 2004).

Após a concepção do algoritmo A^* , alguns autores publicaram outros algoritmos que traziam embutida a idéia de utilizar uma função de avaliação no processo de ordenação de nós. Cita-se como exemplo o algoritmo B^* , concebido por Berliner (1979, p. 23), que visa encontrar o “melhor arco na raiz de uma sub-árvore”, ao longo do processo de busca, como forma de promover a expansão de nós.

Chakrabarti et al. (1989) também propuseram a utilização do algoritmo MA*, no qual é inserida uma função de avaliação dos nós que considera um *trade-off* entre a estimativa do custo e do tempo de processamento para se obter uma solução. Segundo os autores, este algoritmo é ideal para problemas que têm de ser resolvidos em um tempo de processamento restrito e/ou quando há limitação de memória para armazenagem da LISTA DE ABERTOS.

Zhou e Hansen (2006) avaliam a utilização de uma função $\hat{f}(n)$ em processos de busca vertical, para situações onde haja limitação quanto à memória para armazenagem da LISTA DE ABERTOS e/ou do tempo de processamento para obtenção de uma solução. Os autores citam em sua publicação o algoritmo SMAG (*graph-search extension of memory-bounded A**), concebido por Kaindl e Khorsand (1994), no qual utiliza-se um procedimento heurístico na ordenação de nós para expansão em processos de busca vertical.

Zhou e Hansen (2006, p. 385) sugerem também a adoção de um “processo de busca horizontal aliado a uma estratégia *branch-and-bound*” com regras heurísticas de poda, como forma de obter resultados melhores do que aqueles que poderiam ser obtidos a partir do algoritmo A*.

Mesmo divergindo em sua avaliação acerca do melhor algoritmo de busca em grafos, esses e outros autores admitem a importância de se inserir informações heurísticas no processo de ordenação de nós para expansão, como forma de obter soluções em um tempo de processamento aceitável para grafos de qualquer tamanho.

No presente capítulo, buscou-se fazer uma revisão da literatura acerca de métodos considerados promissores, em uma primeira análise, para resolução do PPOC. No próximo item, apresentar-se-á a experiência adquirida na tentativa de utilizar alguns dos métodos anteriormente explicitados como forma de resolver o problema.

4.5 TESTES REALIZADOS PARA RESOLUÇÃO DO PPOC

4.5.1 Resolução do PPOC a partir de um *solver*¹¹ disponível comercialmente

A primeira tentativa de resolução do problema deu-se a partir da utilização do *solver* MINLP¹², disponível no NEOS¹³. Este *solver* utiliza o algoritmo *branch-and-bound* para geração de uma “árvore” cujos nós correspondam a problemas de otimização com restrições

¹¹ Implementação computacional de um método para resolução de problemas de programação matemática.

¹² *Solver* para resolução de Problemas de Programação não-linear Inteira Mista.

¹³ NEOS Server for Optimization. – *site* que disponibiliza a utilização gratuita de *solvers* para problemas de programação matemática (<http://www-neos.mcs.anl.gov/neos/>)

não lineares contínuas. Posteriormente, os problemas contínuos são resolvidos com o auxílio do filterQSP, um *solver* de Programação Quadrática Sequencial que pode resolver problemas que envolvam restrições não lineares (NEOS, 2005).

Para utilização do *solver* MINLP, o problema foi reescrito em linguagem AMPL¹⁴ e submetido ao NEOS, considerando-se diferentes conjuntos de dados de entrada, cada um deles associado a uma referência de peça específica.

Os resultados obtidos para estes conjuntos de dados mostraram-se consistentes para algumas referências testadas. Para outras, especialmente aquelas associadas a uma grande quantidade de peças demandadas, o tempo de processamento até a obtenção de uma solução ultrapassou o limite considerado admissível neste estudo.

Além disso, não se conseguiu obter resultados, a partir do *solver* supracitado, para as situações em que a demanda futura tendesse a zero (situações estas que buscavam representar o final de uma estação). Em tais situações, o tempo de processamento excedia, antes de chegar a uma solução, o limite de tempo disponibilizado pelo NEOS.

Em vista disso, deu-se continuidade ao processo de busca de um método consistente para resolução do PPOC.

4.5.2 Concepção de um método heurístico

A segunda proposta de resolução do PPOC consistiu na concepção de um método heurístico, formulado com base em algumas das características identificadas nas soluções obtidas com a utilização do *solver* MINLP.

Dentre as características identificadas nas soluções para o PPOC obtidas a partir do *solver* supracitado, destacam-se as seguintes:

- I. Há uma tendência de se priorizar o aproveitamento máximo do enfesto, comparativamente ao aproveitamento do comprimento da faca de corte. Em outras palavras, normalmente as soluções apresentam um comprimento de enfesto que se aproxima do limite da mesa (na medida em que haja demanda para isto) e não mantêm o mesmo critério com relação ao número de camadas. Entende-se que isto deve ser motivado pela menor perda de aproveitamento do tecido à medida que aumenta o comprimento de enfesto, bem como pela menor perda de tecido nas dobras em função de um menor número de camadas ser necessário;

¹⁴ *A Modeling Language For Mathematical Programming* (FOURER; GAY; KERNIGHAN, 1993).

- II. Normalmente, com base nas soluções obtidas a partir do MINLP, aproveita-se toda a extensão de uma mesa de corte antes de se passar para a próxima;
- III. As soluções obtidas indicaram que as marcações de um mesmo tamanho costumam ser alocadas em uma mesma mesa de corte, visando diminuir as sobras de peças.

Com base nestas tendências, concebeu-se o seguinte algoritmo para resolução do PPOC a partir de um método heurístico, desenvolvido em função das constatações listadas anteriormente em (I), (II) e (III). Saliente-se que essas constatações são válidas para problemas cuja proporcionalidade entre os diferentes custos envolvidos é aquela verificada para as referências testadas.

O algoritmo heurístico proposto encontra-se apresentado a seguir:

- **Passo 1: Inicialização de parâmetros**

No início do algoritmo, os parâmetros “demanda não atendida” para cada par *tamanho x cor* da referência, “número da mesa de corte” (cujo valor inicial é 1) e os elementos de uma primeira solução para o problema são inicializados. Com relação à solução do PPOC, a mesma é definida a partir de elementos que representam, respectivamente, o número de marcações e o número de camadas em cada mesa de corte programada para atendimento à demanda de uma dada referência. A tabela a seguir ilustra a representação de uma solução, obtida a partir da heurística concebida, para o PPOC associado a uma referência com dois tamanhos e duas cores.

TABELA 01: Representação de uma solução gerada a partir da heurística concebida

MESA 1			
TAMANHO 1	TAMANHO 2	COR 1	COR 2
10	12	5	5

Fonte: Autora

A solução representada a partir da tabela anterior indica que foi alocada apenas uma mesa de corte, e que foram definidas 10 marcações para o TAMANHO 1 e 12 marcações para o TAMANHO 2, respectivamente. Com relação ao número de camadas, definiram-se cinco camadas tanto para a COR 1 como para a COR 2. Inicializados os parâmetros, vai-se ao Passo 2, apresentado a seguir.

- **Passo 2: Estimativa do comprimento de tecido e do número de camadas necessárias**

Em função da demanda associada a cada par *tamanho x cor* da referência, bem como do comprimento associado à marcação dos vários tamanhos, estima-se o comprimento total de tecido a ser enfiado para posterior corte através da seguinte fórmula:

$$Comp = \sum_{c=1}^C \sum_{t=1}^T DEM_{tc} \times COP_t, \text{ onde:}$$

C = número total de cores da referência;

T = número total de tamanhos da referência;

COP_t = comprimento de tecido necessário para corte de uma marcação do tamanho t ;

DEM_{tc} = Demanda de peça para cada um dos tamanhos t e das cores c da referência;

$Comp$ = comprimento necessário de tecido.

Dado que o comprimento de enfiado tem um limite máximo, estima-se (com base no comprimento total estimado anteriormente) o número total de camadas de tecido.

$$Num_camadas = \left\lceil \frac{Comp}{COE\ max} \right\rceil, \text{ onde:}$$

$Num_camadas$ = estimativa do número de camadas necessárias de tecido;

$COE\ max$ = comprimento máximo de enfiado.

A partir daí, pode ser estimado o número de marcações de cada tamanho para atendimento da demanda. Esta operação é realizada no Passo 3, apresentado a seguir.

- **Passo 3: Arredondamento do número de marcações de cada tamanho**

Em função do número de camadas estimado no passo anterior, calcula-se o número de marcações, associado a cada tamanho, para atendimento da demanda. A solução obtida consiste num vetor de valores, cada um deles representando o número de marcações de cada tamanho da referência. Tais valores normalmente não são inteiros, necessitando de arredondamento. Arbitrou-se como regra de arredondamento a seguinte:

a) Valores cuja parte decimal seja menor ou igual a dois décimos ($\leq 0,2$) são arredondados

para o inteiro inferior: $Marc_t = \left\lfloor \frac{\sum_{c=1}^C DEM_{tc}}{Num_camadas} \right\rfloor$, onde:

- $Marc_t$ = Número de marcações associadas ao tamanho t ;

- C = Número total de cores da referência;

- $Num_camadas$ = Número total de camadas estimado no passo anterior.

b) Valores cuja parte decimal seja maior ou igual a oito décimos ($\geq 0,8$) são arredondados

para o inteiro superior: $Marc_t = \left\lceil \frac{\sum_{c=1}^C DEM_{tc}}{Num_camadas} \right\rceil$, onde:

- $Marc_t$ = Número de marcações associadas ao tamanho t ;

- C = Número total de cores da referência;

- $Num_camadas$ = Número total de camadas estimado no passo anterior.

c) Valores entre os dois limites supracitados (0,2 e 0,8, respectivamente) são arredondados de maneira aleatória. Para isto, geram-se números aleatórios entre 0 e 1 (considerando-se nesta geração uma distribuição uniforme) que indicam se o arredondamento dar-se-á para cima ou para baixo. Arbitra-se inicialmente um valor, entre zero e um, abaixo do qual o número de marcações será arredondado para baixo e acima do qual o número de marcações será arredondado para cima.

O resultado dos arredondamentos é armazenado em NMA_{kt} , onde t representa cada um dos tamanhos e k , o número da mesa de corte. Cada uma das marcações tem um comprimento de tecido COP_t , conforme já mencionado.

Se o produto escalar entre NMA_{kt} e COP_t resultar em um valor menor ou igual ao comprimento máximo do enfesto ($COEmax$), evolui-se para o Passo 4. Caso contrário, retorna-se ao Passo 3.

- **Passo 4: Processo de alocação de camadas**

A partir do vetor de número de marcações definido no passo anterior, calcula-se o número de camadas necessárias para atendimento à demanda de cada cor. Para isto, dividem-se os valores da matriz de demanda não atendida (para cada par *tamanho x cor*) pelo respectivo número de marcações de cada tamanho, visando obter o número de camadas (de cada cor) suficiente para atendimento da demanda.

Os valores obtidos para cada camada normalmente não são inteiros, e busca-se avaliar a relação entre os mesmos e seu inteiro mais próximo. Lembrando que as soluções obtidas com o *solver* MINLP procuram atender a demanda de cada um dos tamanhos em uma mesma mesa de corte e visando diminuir as sobras de peças, busca-se alocar, na mesa de corte ora definida, aquelas marcações do(s) tamanho(s) cuja relação entre os valores de camadas - calculado e

arredondado - é menor ou igual a um (1), considerando todas as cores da referência. O(s) tamanho(s) que atende(m) a essa premissa é(são) “marcado(s)” no algoritmo, pois será(ão) alocado(s) nesta mesa de corte. Tal operação pode ser descrita da seguinte maneira:

Para cada tamanho t pertencente ao conjunto de tamanhos T da referência:

$$\text{Se } \frac{\text{Numero_calculado_camadas}_c}{\text{Numero_arredondado_camadas}_c} \leq 1 \text{ para todo } c = 1, \dots, C,$$

O tamanho t será alocado na mesa k ora programada.

Caso não caibam na mesa todas as camadas necessárias para atendimento à demanda deste(s) tamanho(s), proporcionaliza-se o número de camadas a ser alocado na mesa, em função do número máximo de camadas aceitável, da seguinte maneira:

$$\text{Num_camadas}_c = \left\lceil \frac{\text{Num_camadas}_c}{\sum_{c=1}^C \text{Num_camadas}_c} \times \text{NCA max} \right\rceil, \text{ para todo } c = 1, \dots, C.$$

Vai-se para o Passo 5, apresentado a seguir:

- **Passo 5: Atualização dos parâmetros**

Após definição do número de marcações e camadas a serem alocados na mesa de corte que está sendo definida, atualizam-se os elementos da solução associados à referida mesa (elementos que representam o número de marcações e de camadas da ordem de corte recém programada), a produção de peças e a demanda ainda não atendida para cada par *tamanho x cor* da referência. Caso o valor da demanda não atendida seja zero para todos os pares *tamanho x cor*, vai-se ao Passo 6. Caso contrário, incrementa-se o número da mesa em uma unidade e retorna-se ao Passo 2.

- **Passo 6: Melhoria da solução**

Neste passo, a solução para o PPOC já está estruturada e é viável. No entanto, possivelmente a mesma ainda possa ser “melhorada”, ou seja, pode-se eventualmente suprimir marcações/ camadas de modo a reduzir as sobras de peças e melhorar a solução obtida. As tentativas de melhoria da solução foram classificadas nas seguintes categorias:

- a) **SOBRA EVIDENTE NAS MARCAÇÕES:** Se a sobra para um ou mais tamanhos superar o número de camadas previstas para cada uma das cores da referência em uma

- dada mesa de corte, pode-se suprimir uma marcação deste(s) tamanho(s) nesta mesa e manter a solução viável;
- b) SOBRA EVIDENTE NAS CAMADAS: Se a sobra para uma ou mais cores superar o número de marcações previstas para cada um dos tamanhos da referência em uma dada mesa de corte, pode-se suprimir uma camada desta(s) cor(es) nesta mesa e manter a solução viável;
- c) TENTATIVAS DE MELHORAR A SOLUÇÃO: Em algumas situações, consegue-se diminuir sobras através de trocas entre marcações e camadas. Para isto, identificam-se os pares *tamanho x cor* associados às maiores sobras e busca-se verificar se a maior sobra total encontra-se na “linha” associada ao tamanho ou na “coluna” associada à cor de uma matriz de sobras de dimensões $T \times C$ (sendo T o número total de tamanhos e C o número total de cores), calculada previamente. Caso a maior sobra total se encontre na “linha”, suprime-se uma marcação do tamanho associado à mesma e verifica-se a necessidade, em função desta supressão, de inserir alguma camada de determinada cor para atendimento da demanda. Caso a maior sobra total se encontre na “coluna”, suprime-se uma camada da respectiva cor e verifica-se a necessidade, em função desta supressão, de inserir alguma marcação de determinado tamanho para atendimento da demanda. Em ambas as situações avaliadas, a nova solução é guardada em uma variável provisória, sendo considerada como solução “definitiva” apenas se tiver um custo inferior ao da solução anterior.

O fluxograma apresentado a seguir ilustra a estrutura do algoritmo heurístico proposto e detalhado anteriormente para resolução do PPOC. Encontram-se no Apêndice 04 as listagens dos programas implementados em MATLAB¹⁵ (versão 6.5), referentes ao algoritmo heurístico.

¹⁵ Linguagem de Computação Técnica.

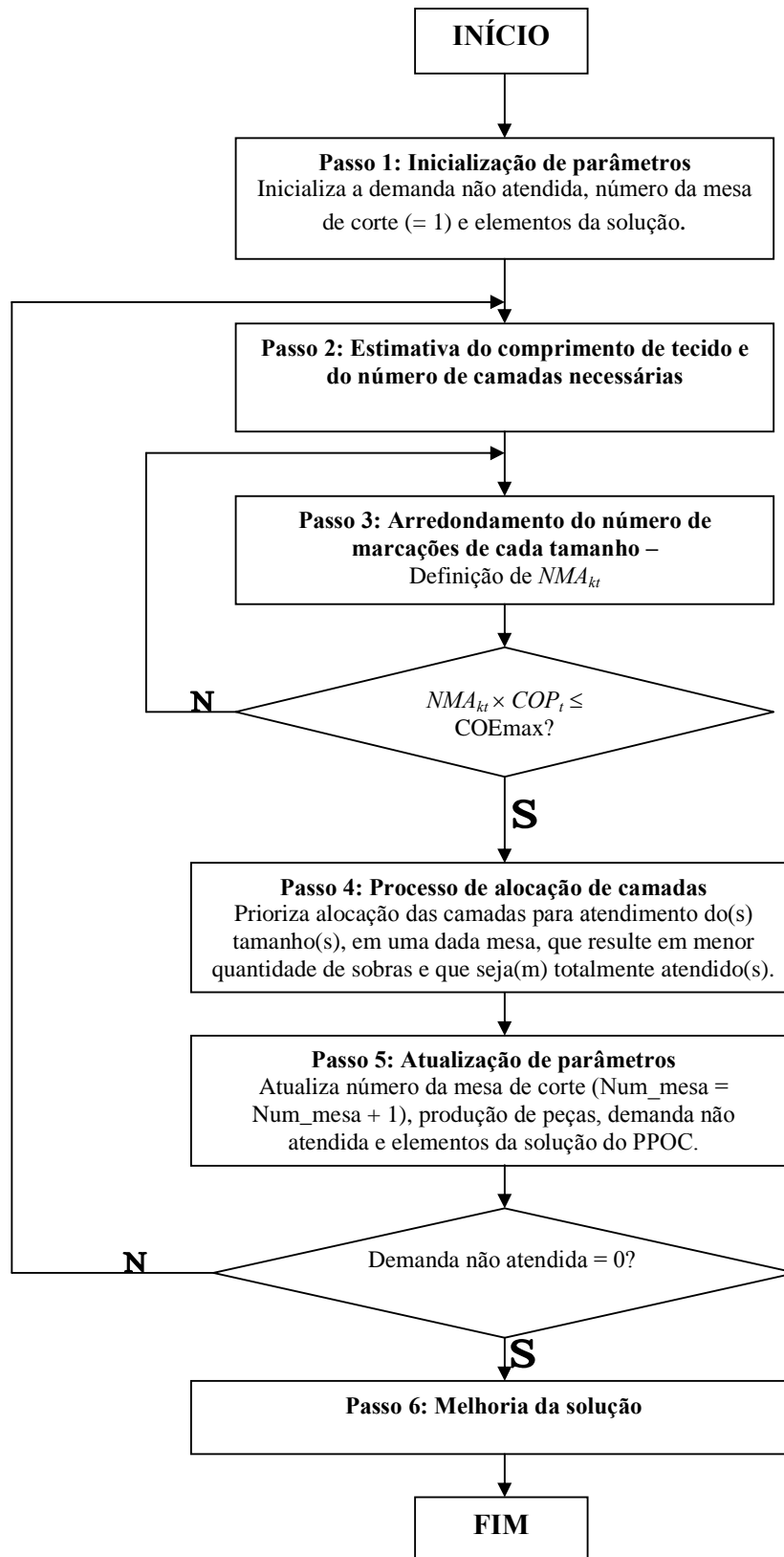


FIGURA 12: Fluxograma do algoritmo heurístico

Os resultados obtidos a partir do algoritmo heurístico mostraram-se consistentes, aproximando-se (em termos de valor da função-objetivo) daqueles obtidos quando da utilização do MINLP, para determinadas referências testadas. Para outras, no entanto, os resultados mostraram-se consideravelmente piores, em termos de custo da solução, do que os obtidos com o MINLP. Como principal vantagem do método heurístico sobre o anteriormente testado, cita-se o tempo de processamento, que sempre se manteve menor que um segundo. Mesmo assim, deu-se continuidade à pesquisa com o intuito de garantir a concepção de um método mais consistente, que pudesse gerar bons resultados para qualquer tipo de referência e qualquer cenário de demanda.

4.5.3 Resolução do PPOC a partir de um algoritmo genético

Visando garantir a obtenção de resultados para o PPOC em um tempo de processamento compatível com aquele disponível pela indústria têxtil, sem contudo prejudicar significativamente a precisão da solução obtida, testou-se a utilização de um algoritmo genético.

Para isto, definiu-se um indivíduo como uma seqüência dos elementos que representariam uma OC. A título de exemplo, se a OC para atendimento a uma dada referência (com dois tamanhos e duas cores) implicasse na utilização de duas mesas, poder-se-ia ter um indivíduo com a seguinte configuração:

MESA 1				MESA 2			
TAM. 1	TAM. 2	COR 1	COR 2	TAM. 1	TAM. 2	COR 1	COR 2
NMA_{11}	NMA_{12}	NCA_{11}	NCA_{12}	NMA_{21}	NMA_{22}	NCA_{21}	NCA_{22}

Onde:

NMA_{11} = Mesa 1: número de marcações do tamanho 1;

NMA_{12} = Mesa 1: número de marcações do tamanho 2;

NCA_{11} = Mesa 1: número de camadas da cor 1;

NCA_{12} = Mesa 1: número de camadas da cor 2;

NMA_{21} = Mesa 2: número de marcações do tamanho 1;

NMA_{22} = Mesa 2: número de marcações do tamanho 2;

NCA_{21} = Mesa 2: número de camadas da cor 1;

NCA_{22} = Mesa 2: número de camadas da cor 2.

A partir de uma população inicial de indivíduos, bem como da concepção de regras de cruzamento e mutação dos mesmos, buscou-se chegar na solução do PPOC. Com base nos resultados obtidos a partir de várias tentativas efetuadas, verificou-se que as características inerentes aos indivíduos da população não pareciam permitir a utilização da propriedade de aditividade, essencial ao sucesso da implementação de um algoritmo genético na resolução de um problema. Entende-se por propriedade de aditividade a possibilidade de fragmentar o indivíduo para posterior cruzamento, mantendo as características boas de cada um dos “pais” no fragmento a ser utilizado para geração dos “filhos”.

Quando esta propriedade se verifica, alguns dos indivíduos filhos mantêm as características boas de ambos os pais, tornando-se melhores que seus antecessores. Em função da não identificação desta propriedade nos testes efetuados, optou-se por interromper as tentativas de utilização do algoritmo genético.

4.5.4 Resolução do PPOC a partir de relaxação lagrangeana

O fato do PPOC abarcar um conjunto de restrições não-lineares torna sua resolução mais complexa, fato que poderia eventualmente ser abrandado se estas restrições fossem retiradas do conjunto de restrições do problema.

Desta forma, reformulou-se o PPOC, relaxando o conjunto de restrições de atendimento à demanda e agregando-as à função-objetivo, sendo cada uma delas multiplicadas por um dos elementos de um vetor de multiplicadores, que teria de ser ajustado a cada iteração como forma de promover a convergência do método. Encontrada uma solução para o problema reformulado, buscava-se adequá-la ao problema original, viabilizando-a a partir de procedimentos heurísticos.

Com base em várias tentativas efetuadas, constatou-se que em uma determinada iteração, chegava-se a uma situação na qual o valor da solução inviável (solução do problema reformulado) era maior do que aquele obtido para a solução viável (solução do problema original) obtida na iteração anterior, o que resultava na não convergência do algoritmo. Atribui-se esta ocorrência à não-convexidade da função-objetivo, optando-se por também interromper a utilização da relaxação lagrangeana na tentativa de resolver o PPOC.

As experiências realizadas com a utilização deste método apontaram para a possibilidade da relaxação lagrangeana não ser adequada para resolução de problemas com

função-objetivo não convexa (recomenda-se uma investigação mais profunda desta possibilidade em trabalhos futuros).

Ao longo das tentativas de resolução do PPOC (especificamente na concepção de heurísticas para resolução do problema primal no método de relaxação lagrangeana), realizaram-se algumas experiências com algoritmos de busca em grafos, experiências essas que geraram resultados promissores. Em função disso, a pesquisa foi direcionada à avaliação de algoritmos de busca em grafos que se mostrassem adequados à resolução do PPOC e que serão detalhados no próximo capítulo.

5 RESOLUÇÃO DO PPOC ATRAVÉS DE PROCESSOS DE BUSCA EM GRAFOS

5.1 ASPECTOS INTRODUTÓRIOS

Conforme já descrito no capítulo anterior, após algumas tentativas de resolução do problema a partir de métodos inicialmente julgados promissores, direcionou-se a pesquisa para a modelagem e posterior utilização de algoritmos de busca em grafos, que deveriam ser adaptados para o caso específico do PPOC.

A modelagem visou conceber o processo de busca de uma solução para o problema, busca esta realizada a partir de uma proposta inicial de configuração da OC, representada através de um nó. Para se chegar à solução, tornou-se necessário definir previamente os conceitos de nó e respectivo custo, arco e processo de expansão do grafo. Estes conceitos, adaptados ao caso específico do PPOC, serão explicitados no próximo item.

Ao longo do processo de concepção dos algoritmos avaliados, levou-se em conta a restrição de tempo de que dispõe a indústria têxtil na resolução do PPOC de suas respectivas referências. Atrasos nesta programação podem gerar atrasos na solicitação de tecidos e, por consequência, no adiamento da execução das ordens de corte. Além disso, o PPOC pode envolver um número considerável de variáveis de decisão, número este resultante do produto do número pré-fixado de mesas de corte pelo número de tamanhos e de cores de uma referência, o que torna a resolução do problema ainda mais complexa.

Em vista disso, buscou-se adaptar diferentes algoritmos de busca em grafos, inserindo nos mesmos procedimentos heurísticos inovadores que permitissem a obtenção da solução em um tempo compatível com o limite de tempo considerado aceitável pela indústria, sem comprometer a precisão dos resultados.

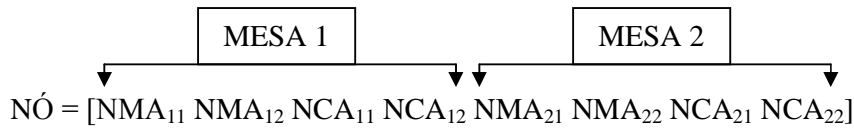
O próximo item descreve a modelagem do problema como um processo de busca em grafos.

5.2 MODELAGEM DO PROBLEMA UTILIZANDO GRAFOS

5.2.1 Definição de Nó

Definiu-se como nó de um grafo a seqüência do número de marcações e de camadas que compõem a OC de uma dada referência. Como exemplo, imagine-se o nó apresentado a

seguir, representando uma OC (composta por duas mesas) associada a uma referência que contemple dois tamanhos e duas cores, respectivamente.



Onde:

NMA₁₁ = Mesa 1: número de marcações do tamanho 1;

NMA₁₂ = Mesa 1: número de marcações do tamanho 2;

NCA₁₁ = Mesa 1: número de camadas da cor 1;

NCA₁₂ = Mesa 1: número de camadas da cor 2;

NMA₂₁ = Mesa 2: número de marcações do tamanho 1;

NMA₂₂ = Mesa 2: número de marcações do tamanho 2;

NCA₂₁ = Mesa 2: número de camadas da cor 1;

NCA₂₂ = Mesa 2: número de camadas da cor 2.

É importante caracterizar os conceitos de viabilidade e otimalidade de um nó.

Um **nó viável** é aquele cuja configuração atende aos três conjuntos de restrições do PPOC, explicitadas no capítulo três deste trabalho (atendimento ao comprimento máximo de enfiado e ao número máximo de camadas por mesa, bem como atendimento à demanda de cada par *tamanho x cor* de peça). Um **nó ótimo** é aquele que atende a todas as restrições do problema ao menor custo.

Ao longo do processo de busca, conforme será explicado posteriormente, os nós gerados sempre atendem, no mínimo, aos dois primeiros conjuntos de restrições.

Cada nó gerado ao longo do processo de busca possui, além de uma configuração específica, um custo, cujo procedimento de cálculo será apresentado posteriormente. Uma vez que a configuração de um nó já fornece todas as informações necessárias para definição da respectiva OC, torna-se desnecessário armazenar o caminho percorrido desde o nó inicial até a obtenção da solução para o problema. Sendo assim, os nós gerados no processo de expansão não requerem apontadores de seus respectivos nós geradores.

5.2.2 Definição de custo

Define-se como custo associado a um nó a soma das parcelas de custo envolvidas na função-objetivo do PPOC, parcelas estas cuja respectiva modelagem foi apresentada no item 3.3. deste trabalho. Lembrando que cada nó do grafo representa uma configuração de OC para a referência ora avaliada, seu respectivo custo é calculado com base nesta configuração, através do somatório do custo do corte das peças, do enfiamento do tecido, da perda de tecido nas dobras, da perda no aproveitamento de tecido em função do comprimento de enfiamento, bem como do estoque e da eventual sobra de peças no atendimento à demanda.

Em função do procedimento adotado para expansão dos nós (a ser explicitado posteriormente), cada nó do grafo sempre terá um custo menor do que o custo de seus sucessores.

5.2.3 Definição de Arco

Na modelagem do problema como um grafo, um arco representa a passagem de uma determinada configuração de OC para outra, passagem esta que sempre ocorre através da inserção de uma marcação de determinado tamanho, de uma camada de determinada cor ou de ambos – marcação e camada (o processo de expansão de nós será detalhado no próximo item). O custo associado a cada arco refere-se ao incremento, no valor da função-objetivo do PPOC, decorrente do procedimento efetuado para geração do novo nó.

Pelo fato do processo de expansão de nós ocorrer através de inserções (de marcações, de camadas ou de ambos), nenhum arco deste grafo tem custo negativo. Além disso, pode-se afirmar que os nós sucessores representam OC's menos deficitárias, em termos de atendimento à demanda de peças, do que as OC's representadas por seus respectivos nós geradores.

5.2.4 Processo de expansão de nós

O processo de geração de sucessores de um dado nó ocorre através do incremento unitário no número de marcações, no número de camadas ou simultaneamente no número de ambos (marcações e camadas). Para isto, há que se definir previamente o tamanho e a cor para os quais será avaliada a possibilidade de efetuar tais inserções.

Para cada nó selecionado para gerar sucessores, gera-se uma *matriz de inviabilidades*, de dimensões $T \times C$, sendo T o número total de tamanhos e C , o número total de cores da

referência. Cada elemento desta matriz indica o número de peças deficitárias do respectivo par *tamanho x cor*, sendo que para os pares não-deficitários (em termos de atendimento à demanda), a matriz é preenchida com um elemento de valor zero. Para escolha do par *tamanho x cor* no qual serão efetuadas inserções ao longo do processo de geração de novos nós, testaram-se as seguintes possibilidades:

- a) Identificar, a cada iteração, o par *tamanho x cor* mais deficitário, em termos de número de peças necessárias para atendimento à demanda;
- b) Identificar, a cada iteração, os dois pares *tamanho x cor* mais deficitários, em termos de número de peças necessárias para atendimento à demanda;
- c) Identificar, a cada iteração, além do par *tamanho x cor* mais deficitário, o *tamanho* e a *cor*, isoladamente, associados ao maior número de peças deficitárias;
- d) Identificar, a cada iteração, o par *tamanho x cor* mais deficitário, em termos percentuais com relação à demanda de peças.

A partir de testes realizados, verificou-se que a alternativa (a) foi a que garantiu o processo mais eficaz de expansão de nós.

O algoritmo de expansão de nós, embutido nos processos de busca da solução para o PPOC, é composto pelos seguintes passos:

- **Passo 1:** Identifique o *nó gerador*, ou seja, o nó a ser expandido, através de procedimento a ser apresentado na descrição dos algoritmos de busca (item 5.3. deste trabalho);
- **Passo 2:** Encontre, para este nó, o par *tamanho x cor* mais deficitário, em termos de número de peças necessárias para atendimento da demanda. Caso seja encontrado mais de um par com o mesmo valor de déficit, selecione o primeiro par encontrado [faça TAMANHO \leftarrow *tamanho(1)* e COR \leftarrow *cor(1)*];
- **Passo 3:** Marque a(s) mesa(s), pertencentes à OC representada pelo *nó gerador*, para a(s) qual(is) não seria possível diminuir o déficit em TAMANHO e COR selecionados no Passo 2 pelo fato de já se ter atingido o limite do comprimento de enfiado e/ou o limite de número de camadas de tecido. Se o número de mesas marcadas for igual ao número de mesas já inseridas no nó, vá ao Passo 4. Senão, vá ao Passo 5;
- **Passo 4:** Insira mais uma mesa no nó, inicializando seus respectivos elementos com o valor zero;
- **Passo 5:** Para cada mesa da OC cuja respectiva capacidade não tiver sido esgotada:

5.a: Se couber uma marcação do TAMANHO e já existir uma camada da COR, gere um novo nó através da inserção de uma marcação em TAMANHO (os demais elementos permanecem iguais àqueles do *nó gerador*);

5.b: Se couber uma camada da COR e já existir uma marcação do TAMANHO, gere um novo nó através da inserção de uma camada em COR (os demais elementos permanecem iguais àqueles do *nó gerador*);

5.c: Se couber tanto uma nova marcação como uma nova camada na mesa ora avaliada, gere um novo nó através da inserção de ambos (marcação e camada) em TAMANHO e COR, respectivamente (os demais elementos permanecem iguais àqueles do *nó gerador*).

Este algoritmo pode ser representado através do seguinte fluxograma:

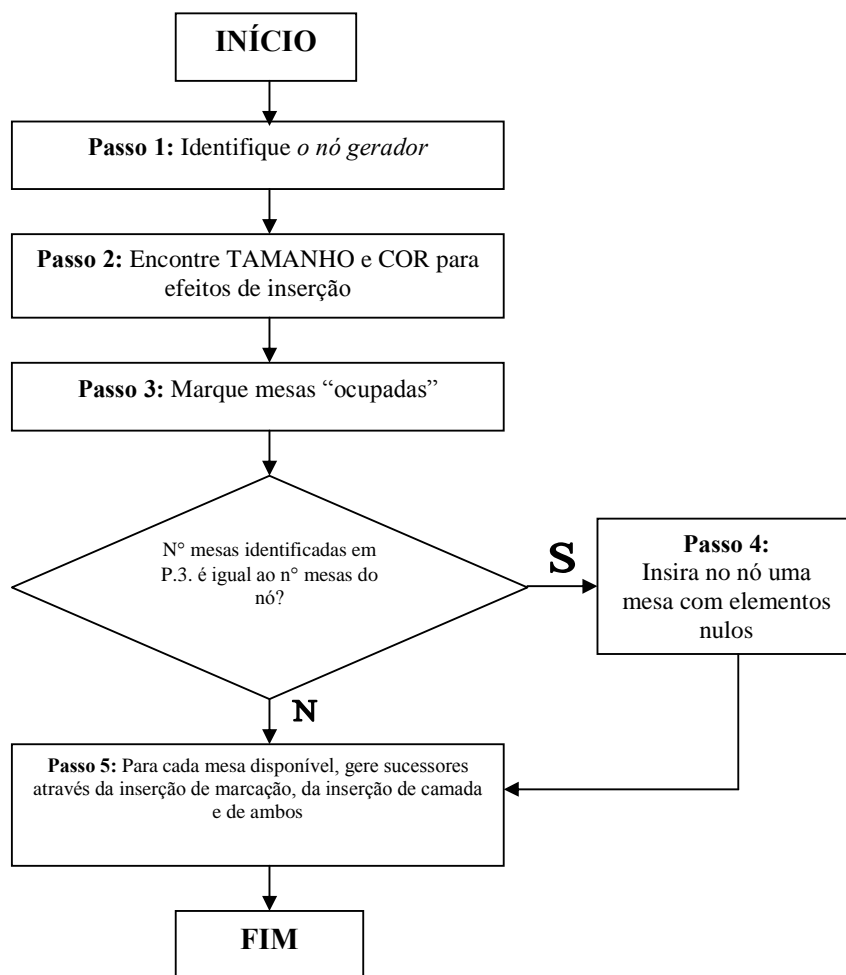


FIGURA 13: Fluxograma do processo de expansão de nós

A seguir, procura-se ilustrar o processo de expansão de nós, considerando o exemplo de um nó inicial que represente uma OC composta por duas mesas e cuja referência associada contemple dois tamanhos e duas cores. Em ambas as mesas, supõe-se que haja espaço para a inserção tanto de novas marcações de tamanho como de novas camadas de tecido. Através da metodologia de identificação do par mais deficitário, imagine-se que o par *tamanho 1 x cor 1* tenha sido selecionado para a realização das inserções.

Nó inicial = [(NMA₁₁) (NMA₁₂) (NCA₁₁) (NCA₁₂) (NMA₂₁) (NMA₂₂) (NCA₂₁) (NCA₂₂)]

Sucessor 1 = [(NMA₁₁+1) (NMA₁₂) (NCA₁₁) (NCA₁₂) (NMA₂₁) (NMA₂₂) (NCA₂₁) (NCA₂₂)]

Sucessor 2 = [(NMA₁₁) (NMA₁₂) (NCA₁₁+1) (NCA₁₂) (NMA₂₁) (NMA₂₂) (NCA₂₁) (NCA₂₂)]

Sucessor 3 = [(NMA₁₁+1) (NMA₁₂) (NCA₁₁+1) (NCA₁₂) (NMA₂₁) (NMA₂₂) (NCA₂₁) (NCA₂₂)]

Sucessor 4 = [(NMA₁₁) (NMA₁₂) (NCA₁₁) (NCA₁₂) (NMA₂₁+1) (NMA₂₂) (NCA₂₁) (NCA₂₂)]

Sucessor 5 = [(NMA₁₁) (NMA₁₂) (NCA₁₁+1) (NCA₁₂) (NMA₂₁) (NMA₂₂) (NCA₂₁+1) (NCA₂₂)]

Sucessor 6 = [(NMA₁₁+1) (NMA₁₂) (NCA₁₁+1) (NCA₁₂) (NMA₂₁+1) (NMA₂₂) (NCA₂₁+1) (NCA₂₂)]

Onde:

NMA₁₁ = Mesa 1: número de marcações do tamanho 1 para o nó inicial;

NMA₁₂ = Mesa 1: número de marcações do tamanho 2 para o nó inicial;

NCA₁₁ = Mesa 1: número de camadas da cor 1 para o nó inicial;

NCA₁₂ = Mesa 1: número de camadas da cor 2 para o nó inicial;

NMA₂₁ = Mesa 2: número de marcações do tamanho 1 para o nó inicial;

NMA₂₂ = Mesa 2: número de marcações do tamanho 2 para o nó inicial;

NCA₂₁ = Mesa 2: número de camadas da cor 1 para o nó inicial;

NCA₂₂ = Mesa 2: número de camadas da cor 2 para o nó inicial.

Para a geração de seis sucessores a partir do nó inicial recém apresentado, parte-se da premissa de que os elementos NMA₁₁, NCA₁₁, NMA₂₁ e NCA₂₁ tenham, originalmente, valores maiores que zero. De acordo com o algoritmo recém apresentado e supondo-se que, por exemplo, o elemento NMA₁₁ tenha valor igual a zero no **nó inicial**, o *Sucessor 2* não pode ser gerado. Encontra-se apresentado no Apêndice 03 um exemplo do processo de expansão de nós.

Com base no procedimento de expansão descrito e posteriormente ilustrado, pode-se chegar às seguintes conclusões acerca do número de sucessores de um nó:

- Para cada uma das mesas contidas em um nó a ser expandido, gera-se um mínimo de zero (no caso da mesa já ter sua capacidade esgotada) e um máximo de três sucessores;
- Para cada mesa recém-criada ao longo do processo de expansão, gera-se um único sucessor através da inserção simultânea de uma marcação e de uma camada no par *tamanho x cor* mais deficitário;
- Somente são inseridas marcações e camadas em uma mesa caso as restrições de comprimento máximo de enfiado e de número máximo de camadas não sejam violadas a partir de tais inserções;
- A inserção de uma marcação em *tamanho* (do par *tamanho x cor* selecionado previamente), para a geração de um novo sucessor, somente ocorre se houver ao menos uma camada em *cor* pois, caso contrário, não se consegue reduzir o déficit de peças no par *tamanho x cor*. Da mesma forma, a inserção de uma camada em *cor*, para a geração de um novo sucessor, somente ocorre se houver ao menos uma marcação em *tamanho*.

No próximo item, são apresentados os algoritmos de busca em grafos adaptados para obtenção de uma solução para o PPOC.

5.3 ADAPTAÇÃO DE ALGORITMOS DE BUSCA EM GRAFOS

5.3.1 Aspectos Introdutórios

Tendo como base alguns dos processos de busca apresentados no capítulo anterior, foram adaptados algoritmos específicos para resolução do PPOC. Essas adaptações consistiram na inserção de procedimentos heurísticos de ordenação de nós no processo de expansão, bem como de exclusão daqueles nós considerados não promissores ao longo da busca.

Entende-se que esta tenha sido uma das contribuições deste trabalho: a inserção de procedimentos heurísticos em algoritmos clássicos de busca em grafos, de modo a adaptá-los para a resolução do PPOC.

A primeira inovação proposta, e que será apresentada em seguida, consiste na definição do nó *inicial*, cujas características são fundamentais para possibilitar a redução do tempo de

processamento envolvido na resolução do problema. Posteriormente, são apresentados os algoritmos testados durante a pesquisa, salientando-se que a eficiência (medida em termos de tempo de processamento) e eficácia (medida em termos de custo da solução gerada) de cada um deles, na resolução do PPOC dependem, dentre outras coisas, das características da referência com a qual se está trabalhando.

5.3.2 Definição do *nó inicial*

Em um primeiro momento, pensou-se em definir como *nó inicial* um nó composto de uma mesa com elementos (marcações e camadas) de valor zero. Como a expansão de nós ocorre através de inserções nos valores de seus respectivos elementos, julgou-se que um *nó inicial* formado por elementos nulos, aliado a um bom processo de busca, seria suficiente para a resolução do problema.

Após vários testes, constatou-se que um *nó inicial* nulo acarreta em um processo de busca muito longo, sendo necessárias muitas iterações até que se chegue ao primeiro nó viável e até que possam ser avaliadas todas as possibilidades de expansão. Desta forma, definiu-se um *nó inicial* que já atendesse (parcialmente) à demanda de peças prevista para uma dada referência.

Visando reduzir o tempo de processamento e com base nos resultados promissores obtidos a partir da utilização do método heurístico descrito em 4.5.2., testou-se a possibilidade de definir, como *nó inicial*, um nó gerado a partir do referido processo, considerando uma demanda de peças menor do que a originalmente prevista para a referência.

A partir de testes realizados, verificou-se que a concepção de um nó (através do processo heurístico) que atendesse a uma demanda de 25% da demanda original, já era suficiente para reduzir o tempo de processamento e, por outro lado, não comprometia em muito a precisão dos resultados. Saliente-se que, para definição da proporção mais adequada de demanda a ser atendida com o *nó inicial*, tornou-se necessário analisar o *trade-off* entre tempo de processamento e precisão da solução: proporções muito pequenas de demanda não seriam suficientes para reduzir o tempo de processamento envolvido no processo de busca, sendo que proporções muito grandes poderiam gerar nós iniciais muito próximos da viabilidade, restando poucas alternativas de expansão a serem analisadas.

Verificou-se também que a inserção, no *nó inicial*, de uma mesa adicional com elementos nulos, além daquelas mesas definidas com o procedimento anteriormente descrito, repercutiria em maior flexibilidade no processo de expansão. Isto porque o processo de

expansão descrito em 5.2.4. somente prevê a adição de mais uma mesa quando as mesas atuais têm suas respectivas capacidades (dimensionadas pelo comprimento máximo de enfiado e pelo número máximo de camadas) alcançadas. A adição de uma mesa com elementos nulos proporciona outras possibilidades de expansão do nó.

Com base no exposto acima, pode-se estruturar o algoritmo de obtenção do *nó inicial* através dos seguintes passos:

- **Passo 1:** Defina *demanda inicial* \leftarrow 25% da demanda atual prevista para a referência;
- **Passo 2:** Utilize o procedimento heurístico apresentado em 4.5.2. para gerar uma solução para o PPOC, considerando a *demanda inicial* definida no Passo 1. A solução assim obtida será definida como *nó inicial* do processo de busca;
- **Passo 3:** Adicione, ao *nó inicial* obtido no Passo 2, uma mesa formada por elementos com valor zero.

A seguir, serão explicitados os algoritmos concebidos com base em diferentes processos de busca em grafos: processo vertical com limitação de custo, processo de busca ávida e processo com lista de nós abertos e limitação de custo, além de métodos híbridos.

5.3.3 Algoritmo de busca vertical com limitação de custo

5.3.3.1 Aspectos Introdutórios

A primeira tentativa de implementar um algoritmo de busca em grafos para resolução do PPOC consistiu na utilização de um processo de busca vertical com limitação de custo.

O processo de busca vertical com limitação de custo segue a idéia, citada em Christofides (1975, p. 394), de restringir o processo de busca a partir da definição de um “limite máximo de custo” (no caso de um problema de minimização), limite este normalmente associado ao custo da melhor solução obtida até o momento. Neste caso, tão logo seja encontrada uma solução para o problema, seu respectivo custo é um parâmetro limitante na continuidade da busca, ou seja, quando um nó (advindo do processo de expansão) apresenta um custo igual ou superior àquele limite, é excluído do processo por não ser mais considerado um nó promissor para o alcance de uma solução satisfatória.

Para implementar este algoritmo foi necessário, inicialmente, definir critérios de ordenação de nós para expansão. Estes critérios serão apresentados no próximo item.

5.3.3.2 Ordenação de nós para expansão

Os critérios avaliados foram os seguintes:

- a) Ordenação de nós com base no seu respectivo custo, contabilizado através do procedimento explicitado em 5.2.2.;
- b) Ordenação de nós com base no seu respectivo custo, somado a uma estimativa de custo para se chegar na solução do PPOC - utilização do conceito de $\hat{f}(n)$ citado por Nilsson (1971). Vale lembrar que o parâmetro $\hat{f}(n)$ é calculado através da fórmula $\hat{f}(n) = g(n) + \hat{h}(n)$, sendo $g(n)$ o custo associado a um nó n e $\hat{h}(n)$, a estimativa do custo para obtenção de uma solução a partir de n .

A utilização do critério de ordenação exposto em (a), se por um lado poderia garantir a obtenção da solução ótima, por outro incorria em um tempo de processamento que inviabilizava a resolução do PPOC, ultrapassando em muito o limite de tempo disponibilizado pela indústria têxtil para esta finalidade quando do planejamento da produção.

Sendo assim, partiu-se para a utilização do critério de ordenação exposto em (b), tendo sido testadas diferentes funções que visassem a estimativa de custo entre um nó n do grafo e a solução do problema - função $\hat{h}(n)$ apresentada em Nilsson (1971) e já mencionada anteriormente.

A função $\hat{h}(n)$ que gerou os melhores resultados, dentre as funções testadas, foi aquela calculada através de uma estimativa de custo para eliminação da inviabilidade identificada em um nó n . Para descrever o procedimento adotado na estimativa dos valores desta função, torna-se necessário definir previamente os seguintes parâmetros:

I. *Matriz de inviabilidades 0-1*: A definição do grau de inviabilidade de um nó do grafo foi realizada a partir da soma dos elementos de uma *matriz de inviabilidades 0-1* de dimensões $T \times C$, sendo T o número de tamanhos e C o número de cores da referência. Esta matriz, gerada com base na *matriz de inviabilidades* já apresentada anteriormente em 5.2.4. (matriz que indica o número de peças deficitárias associadas a cada par *tamanho x cor* da referência), é formada por elementos de valor **um** nas posições cujo déficit de peças tenha sido identificado, e por elementos de valor **zero** nas demais posições. A título de exemplo, imagine-se um nó que representa a OC para uma referência que envolva dois tamanhos (*tamanho 1* e *tamanho 2*) e duas cores (*cor 1* e *cor 2*). Suponha-se que a *matriz de inviabilidades 0-1* identificada para este nó tenha apresentado a seguinte configuração:

$$\text{Matriz de inviabilidades } 0-1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

A matriz apresentada anteriormente estaria indicando déficit de peças (com relação à demanda a ser atendida) no par *tamanho 1 x cor 1* da referência, não se tendo constatado déficit nos demais pares.

O grau de inviabilidade de um nó, para efeitos de estimativa dos valores de $\hat{h}(n)$, foi definido como sendo a soma dos elementos de sua respectiva *matriz de inviabilidades 0-1*. No caso do exemplo supracitado, o grau de inviabilidade do nó em questão seria igual a um, valor este resultante da soma dos elementos da *matriz de inviabilidades 0-1*.

II. *Custo marginal de redução de inviabilidade*: Em função do procedimento de expansão (apresentado em 5.2.4.) adotado neste trabalho, verifica-se que os sucessores de um nó n sempre apresentarão um grau de inviabilidade igual ou menor do que aquele identificado para n . Essa redução no grau de inviabilidade acarreta num custo adicional, decorrente da inserção unitária de marcação, camada ou de ambos. Definiu-se como *custo marginal de redução de inviabilidade* o aumento de custo verificado para reduzir uma unidade do grau de inviabilidade entre um nó n e um sucessor do mesmo.

Este parâmetro, portanto, é calculado através da seguinte fórmula:

$$CM(\text{sucessor}_n) = \frac{\text{Custo}(\text{sucessor}_n) - \text{Custo}(n)}{\text{Grau_inviabilidade}(n) - \text{Grau_inviabilidade}(\text{sucessor}_n) + CTE},$$

Onde:

$CM(\text{sucessor}_n)$ = Custo marginal de redução de inviabilidade para um dado sucessor do nó n ;

$\text{Custo}(\text{sucessor}_n)$ = custo do nó sucessor de n ;

$\text{Custo}(n)$ = custo do nó n ;

$\text{Grau_inviabilidade}(n)$ = Soma da *matriz de inviabilidades 0-1* obtida para o nó n ;

$\text{Grau_inviabilidade}(\text{sucessor}_n)$ = Soma da *matriz de inviabilidades 0-1* obtida para o nó sucessor de n ;

CTE = constante somada ao denominador como forma de eliminar a possibilidade de divisão por zero.

Com base nos dois parâmetros apresentados anteriormente em (I) e (II), pode-se definir a função $\hat{h}(n)$ como sendo o produto do grau de inviabilidade do nó n por seu respectivo *custo marginal de redução de inviabilidade*. A estimativa $\hat{f}(n)$, como já explicitado anteriormente, consiste na soma entre o custo $g(n)$ e a estimativa $\hat{h}(n)$ para o nó n . O processo de ordenação de nós com base neste critério gerou bons resultados para grande parte das referências testadas (conforme será apresentado no próximo capítulo).

5.3.3.3 Exclusão de nós do processo de busca

Visando acelerar o processo de busca, inseriu-se um procedimento de exclusão dos nós considerados não promissores para resolução do PPOC. O critério de exclusão seguiu a filosofia do processo de ordenação de nós: apenas sofrem expansão os nós n cujo respectivo custo estimativo $\hat{f}(n)$ seja menor que o custo da melhor solução obtida até o momento, sendo que os nós que não atendam a esta premissa são excluídos do processo de busca.

5.3.3.4 Algoritmo proposto para resolução do PPOC

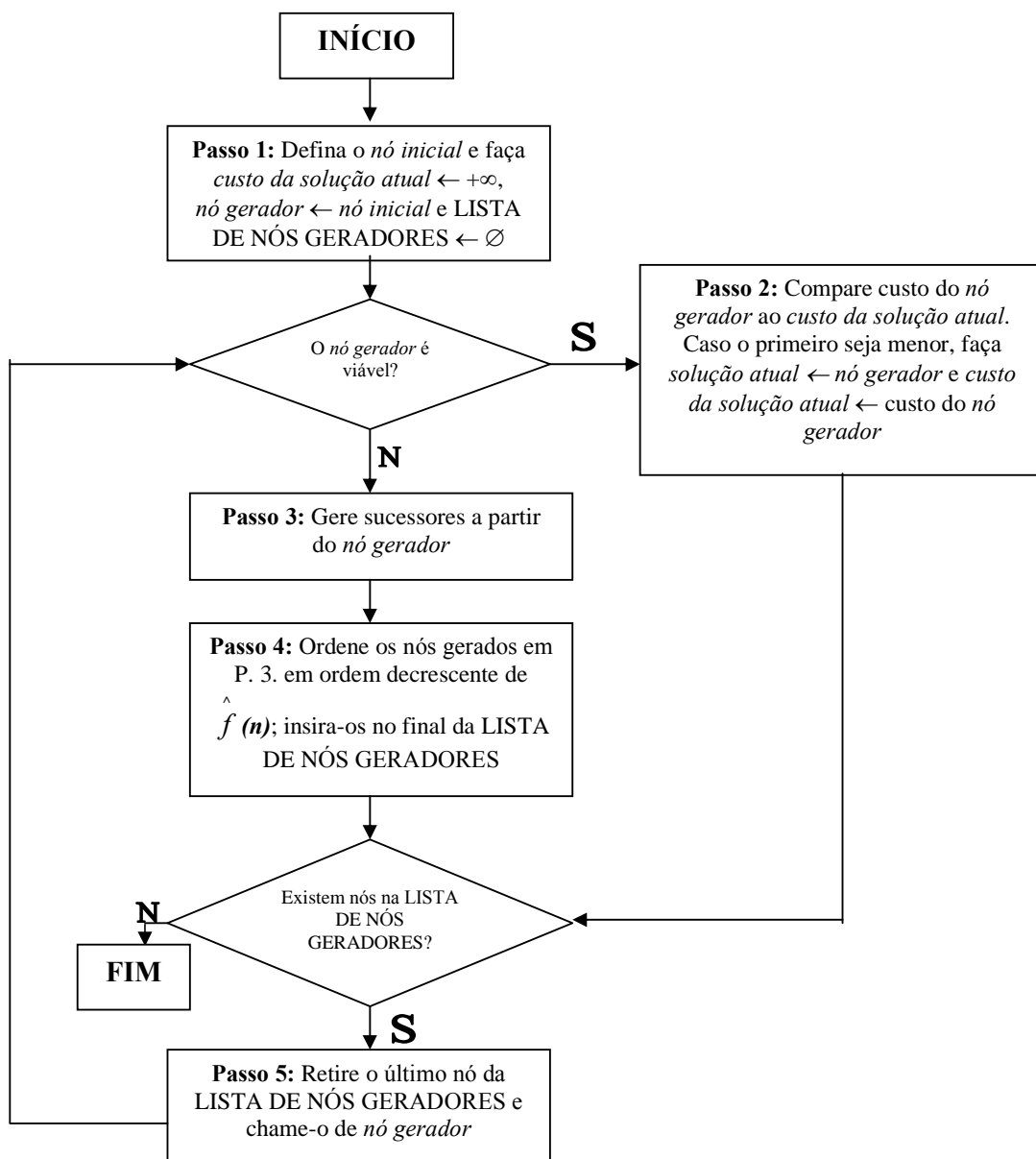
O algoritmo implementado para resolução do PPOC, sob o enfoque da busca vertical com limitação de custo, compreende os seguintes passos:

- **Passo 1:** Defina o *nó inicial* (com base no procedimento apresentado em 5.3.2.) e um custo de $+\infty$ como *custo da solução atual*. Faça *nó gerador* \leftarrow *nó inicial*, crie uma LISTA DE NÓS GERADORES (inicialmente vazia);
- **Passo 2:** Se o *nó gerador* for viável, compare seu respectivo custo ao *custo da solução atual* (se o custo do *nó gerador* for menor, faça *solução atual* \leftarrow *nó gerador* e *custo da solução atual* \leftarrow custo do *nó gerador*). Retire da LISTA DE NÓS GERADORES aqueles nós cujo respectivo custo $\hat{f}(n)$ seja igual ou maior do que o *custo da solução atual*. Se *nó gerador* for viável e diferente de *nó inicial*, vá ao Passo 5; se *nó gerador* for inviável, vá ao Passo 3;

- **Passo 3:** Gere sucessores a partir do *nó gerador*, através do procedimento apresentado em 5.2.4.;
- **Passo 4:** Ordene os nós gerados no Passo 3 em ordem decrescente de seus respectivos custos $\hat{f}(n)$. Insira, ao final da LISTA DE NÓS GERADORES, os nós recém-ordenados que tiverem apresentado um custo $\hat{f}(n)$ menor do que o *custo da solução atual*, excluindo os demais do processo de busca;
- **Passo 5:** Se não houver nós na LISTA DE NÓS GERADORES, **pare**. Se houver, defina o último nó da lista como *nó gerador* e exclua-o em seguida da mesma. Volte ao Passo 2.

Este algoritmo pode ser representado através do seguinte fluxograma:

FIGURA 14: Fluxograma do algoritmo de busca vertical com limitação de custo



O processo de busca ora descrito pode ser definido como um método de enumeração implícita, que finaliza apenas quando todos os nós gerados, que apresentem custo $\hat{f}(n)$ menor que o custo da melhor solução obtida até o momento, tiverem sido pesquisados. Saliente-se também que o procedimento de ordenação embutido neste algoritmo garante que a regra FILO¹⁶ (*first-in last-out*) seja utilizada na expansão dos nós. A figura a seguir procura representar este processo de busca.

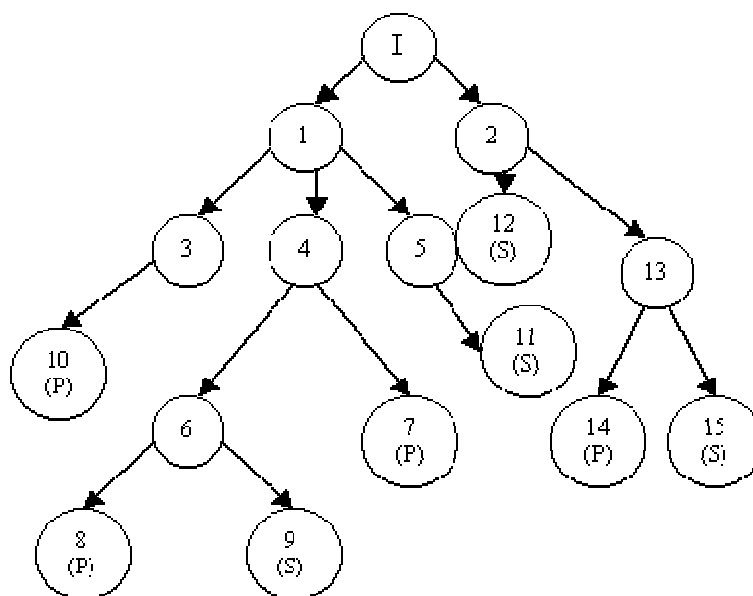


FIGURA 15: Representação do processo de busca vertical com limitação de custo

Na figura 15, o número associado a cada um dos nós representa a ordem na qual os mesmos foram gerados, sendo que o símbolo I representa o nó inicial do processo de busca. O símbolo (P), inserido em alguns dos nós gerados, indica que seus respectivos ramos foram podados no processo de busca, em função de seu custo $\hat{f}(n)$ ser igual ou maior do que o custo da melhor solução obtida até o momento. Finalmente, o símbolo (S) indica os nós que representam soluções para o problema. Ao longo da busca, armazena-se somente a solução associada ao menor custo obtido até o momento.

Os resultados para o PPOC, obtidos a partir da utilização do algoritmo anteriormente apresentado, são explicitados no próximo capítulo.

¹⁶ No caso do processo de busca ora descrito, esta regra garante que o primeiro nó a ter entrado na LISTA DE GERADORES seja o último a ser expandido.

5.3.4 Algoritmo de busca ávida

5.3.4.1 Aspectos Introdutórios

Visando reduzir o tempo de processamento do processo de busca, implementou-se num segundo momento um algoritmo de busca ávida. Este algoritmo escolhe, a cada iteração e dentre os nós recém-gerados, aquele nó considerado o mais promissor segundo algum critério pré-estabelecido. Os nós que não foram escolhidos são simplesmente eliminados do processo de busca e promove-se nova expansão a partir do nó escolhido, e assim sucessivamente até a obtenção de uma solução.

Encontram-se apresentados a seguir os critérios definidos para escolha do *nó gerador*.

5.3.4.2 Critérios de escolha do *nó gerador*

O primeiro passo para concepção de um algoritmo de busca ávida, que pudesse ser utilizado na resolução do PPOC, consistiu na definição de critérios para escolha, a cada iteração, do nó gerador de sucessores.

Testaram-se diferentes critérios, sendo que cada um deles se mostrou mais adequado a determinados tipos de referência e de cenários de demanda, conforme poderá ser observado a partir dos resultados apresentados no próximo capítulo. Encontram-se a seguir apresentados os critérios testados para escolha do *nó gerador*.

- a. Escolha do nó com base em uma medida de inviabilidade - a primeira tentativa de utilização do processo de busca ávida deu-se pela escolha do *nó gerador* associado à menor medida de inviabilidade. Como medida de inviabilidade, adotou-se a norma da *matriz de inviabilidades* - apresentada em 5.2.4.;
- b. Escolha do nó com menor *custo marginal de redução de inviabilidade* - para calcular o valor deste parâmetro, utilizou-se a fórmula já apresentada em (II) do item 5.3.3.2.;
- c. Escolha do nó com o menor valor de $\hat{f}(n)$ – estimativa de custo entre o nó inicial e a solução do problema, passando pelo nó n ;

- d. Escolha do nó com o menor valor de uma função de mérito¹⁷ ajustada previamente - com o intuito de testar outras possibilidades de escolha do *nó gerador*, avaliou-se a utilização de uma função de mérito. A função que propiciou os melhores resultados foi a do tipo $função_mérito = g(n) + B \times inv(n)$,

Onde:

$função_mérito$ = valor da função de mérito;

$g(n)$ = custo do nó, medido em termos de valor da função objetivo;

B = Peso atribuído à medida de inviabilidade, ajustado neste trabalho como um valor que vai diminuindo com o número de iterações efetuadas ao longo do processo de

busca: $\frac{100}{número_iterações}$;

$inv(n)$ = medida de inviabilidade do nó n , definida como a soma dos elementos da *matriz de inviabilidades 0-1*, já apresentada em (I) do item 5.3.3.2.;

- e. Escolha do nó com o menor valor de uma função de mérito adaptada - como último critério para escolha do nó gerador, utilizou-se a mesma função de mérito apresentada anteriormente, mas com uma pequena modificação no cálculo do custo $g(n)$ associado a um nó: como a parcela de custo mais significativa é a da sobra de peças, atribuiu-se um peso de 70% à mesma e um peso de 30% ao somatório das demais parcelas de custo do PPOC. Esta ponderação das parcelas de custo foi obtida de maneira iterativa, testando-se diferentes pares de peso e escolhendo aquele que propiciou as melhores soluções para o problema. Os demais parâmetros são idênticos àqueles já citados anteriormente em 5.3.4.2 (d).

Encontram-se apresentados a seguir os passos do algoritmo de busca ávida testado na resolução do PPOC.

5.3.4.3 Algoritmo proposto para resolução do PPOC

O algoritmo proposto sob o enfoque de um processo de busca ávida envolve os seguintes passos:

¹⁷ Parâmetro, definido para cada nó do grafo, que busca aliar uma medida de custo a uma medida do grau de inviabilidade deste nó.

- **Passo 1:** Defina o *nó inicial* (com base no procedimento apresentado em 5.3.2.) e escolha um critério, dentre aqueles critérios de escolha do *nó gerador* apresentados em 5.3.4.2.. Faça $nó\ gerador \leftarrow nó\ inicial$;
- **Passo 2:** Se o *nó gerador* for viável, **pare:** o *nó gerador* representa a solução do PPOC. Senão, vá ao Passo 3;
- **Passo 3:** Gere sucessores a partir de *nó gerador*, através do procedimento apresentado em 5.2.4.;
- **Passo 4:** Escolha o novo *nó gerador* dentre os nós sucessores gerados no passo anterior, de acordo com o critério especificado no Passo 1. Volte ao Passo 2.

O fluxograma a seguir representa o funcionamento deste algoritmo.

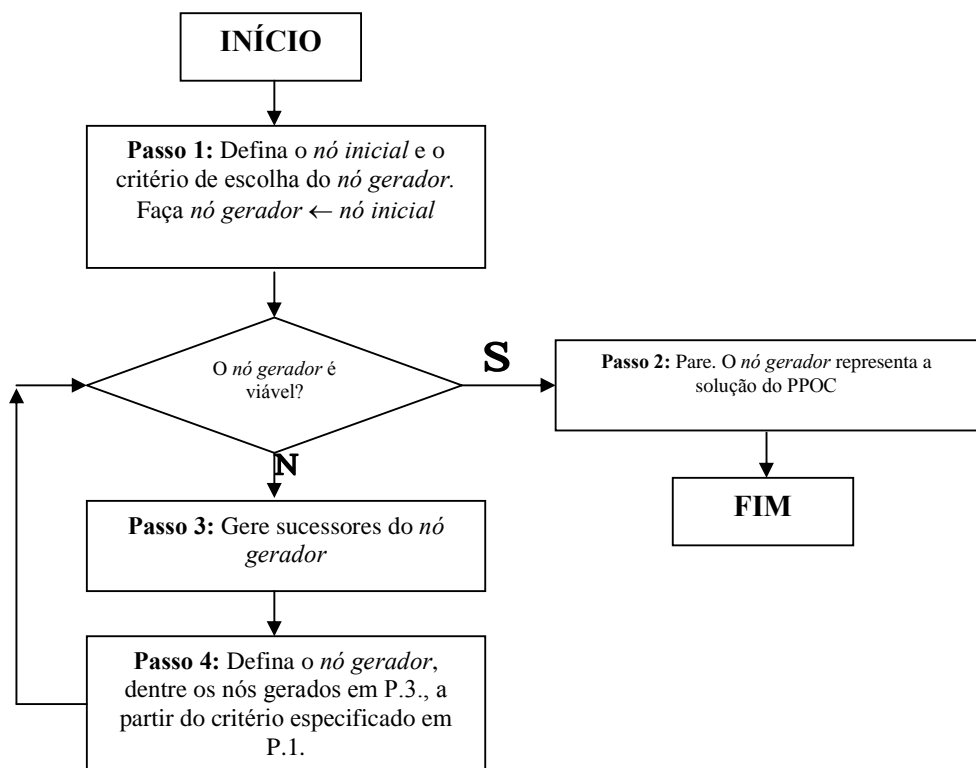


FIGURA 16: Fluxograma do algoritmo de busca ávida

A figura a seguir ilustra um processo de busca ávida.

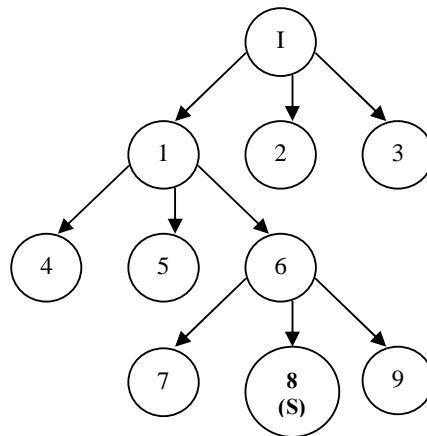


FIGURA 17: Representação do processo de busca ávida

Na figura 17, o número inserido em cada nó indica a ordem na qual o mesmo foi gerado. O símbolo (S) indica que o referido nó consiste na solução do problema. A cada iteração do algoritmo, elege-se o *nó gerador* e eliminam-se os demais do processo de busca. Tão logo seja obtida uma solução para o problema, interrompe-se o processamento do algoritmo.

Conforme poderá ser visualizado no próximo capítulo, os resultados obtidos a partir do algoritmo anteriormente apresentado demonstraram a grande eficiência do método em termos de tempo de processamento, mas baixa eficácia, de maneira geral, em termos de precisão da solução obtida. Em vista disso, testaram-se outros processos de busca a serem apresentados nos próximos itens.

5.3.5 Algoritmo de busca vertical com limitação de custo/ busca ávida

5.3.5.1 Aspectos Introdutórios

Visando aliar a eficácia (medida em termos de custo da solução) do processo de busca vertical com limitação de custo à eficiência (medida em termos de tempo de processamento) do processo de busca ávida, concebeu-se um método híbrido, que iniciasse com aquele primeiro processo e, depois de determinado tempo, promovesse um processo de busca ávida até a obtenção da solução.

O tempo associado à aplicação do processo de busca vertical com limitação de custo pode ser controlado pelo usuário. Neste trabalho, porém, testou-se o algoritmo considerando um tempo de dois minutos vinculado ao primeiro processo.

É importante salientar, com relação aos critérios de ordenação e de exclusão de nós inseridos no processo de busca, que houve uma modificação com relação àqueles especificados para o algoritmo de busca vertical com limitação de custo, citados em 5.3.3.2 e 5.3.3.3, respectivamente: para o algoritmo ora apresentado, tanto o processo de ordenação como o de exclusão de nós, ao longo da busca, deu-se em função de seus respectivos custos $g(n)$. Não se utilizou neste método híbrido o conceito de $\hat{f}(n)$ proposto por Nilsson (1971), por entender que a redução no tempo de processamento seria promovida pela busca ávida na segunda etapa da busca, não sendo necessário inserir um procedimento heurístico adicional que pudesse comprometer ainda mais a precisão da solução obtida para o PPOC.

Quanto ao processo de busca ávida, o critério de escolha do nó gerador deu-se sempre em função do *custo marginal de redução de inviabilidade*. Portanto, neste método híbrido de busca o usuário não tem a opção de escolher entre um dos critérios de seleção do *nó gerador* especificados em 5.3.4.2.

Os resultados obtidos a partir da utilização deste algoritmo encontram-se apresentados no próximo capítulo. A seguir, apresentam-se os passos de implementação do mesmo.

5.3.5.2. Algoritmo proposto para resolução do PPOC

O algoritmo de implementação do método híbrido pode ser descrito a partir dos seguintes passos:

- **Passo 1:** Defina o *nó inicial* (com base no procedimento apresentado em 5.3.2.), *custo da solução atual* $\leftarrow +\infty$ e arbitre um tempo máximo (*TempoMax*) associado ao processo de busca vertical com limitação de custo. Inicialize também um contador de tempo de processamento (*Cont_TP*) e uma LISTA DE NÓS GERADORES. Faça *nó gerador* \leftarrow *nó inicial*;
- **Passo 2:** Se o *nó gerador* for viável, compare seu respectivo custo ao *custo da solução atual* (se o custo do *nó gerador* for menor, faça *solução atual* \leftarrow *nó gerador* e *custo da solução atual* \leftarrow custo do *nó gerador*). Retire da LISTA DE NÓS GERADORES aqueles nós cujo respectivo custo $g(n)$ seja igual ou maior que o *custo da solução*

atual. Se *nó gerador* for viável e diferente de *nó inicial*, vá ao Passo 5; se *nó gerador* for inviável, vá ao Passo 3;

- **Passo 3:** Gere sucessores a partir do *nó gerador*, através do procedimento apresentado em 5.2.4.. Se *Cont_TP* for menor ou igual ao *TempoMax*, vá ao Passo 4; senão, vá ao Passo 6;
- **Passo 4:** Promova a ordenação dos nós gerados no Passo 3 por ordem decrescente de seus respectivos custos $g(n)$. Insira, ao final da LISTA DE NÓS GERADORES, os nós recém-ordenados que tiverem apresentado um custo $g(n)$ menor do que o *custo da solução atual*, excluindo os demais do processo de busca;
- **Passo 5:** Caso não haja nós na LISTA DE NÓS GERADORES, **pare**. Caso haja, defina o último nó desta lista como *nó gerador* e exclua-o da mesma em seguida. Volte ao Passo 2;
- **Passo 6:** Escolha o *nó gerador* dentre os nós gerados no Passo 3, tendo como critério o *custo marginal de redução de inviabilidade*. Caso o *nó gerador* seja viável, vá ao Passo 7. Caso o *nó gerador* seja inviável e apresente um custo $g(n)$ maior ou igual ao *custo da solução atual*, abandone-o e volte ao Passo 5. Caso o *nó gerador* seja inviável e tenha um custo menor que o *custo da solução atual*, volte ao Passo 3;
- **Passo 7:** Compare o custo do *nó gerador* ao *custo da solução atual* (caso o custo do *nó gerador* seja menor, faça *solução* ← *nó gerador* e *custo da solução atual* ← custo do *nó gerador*) e volte ao Passo 5.

O algoritmo descrito anteriormente pode ser ilustrado através do seguinte fluxograma:

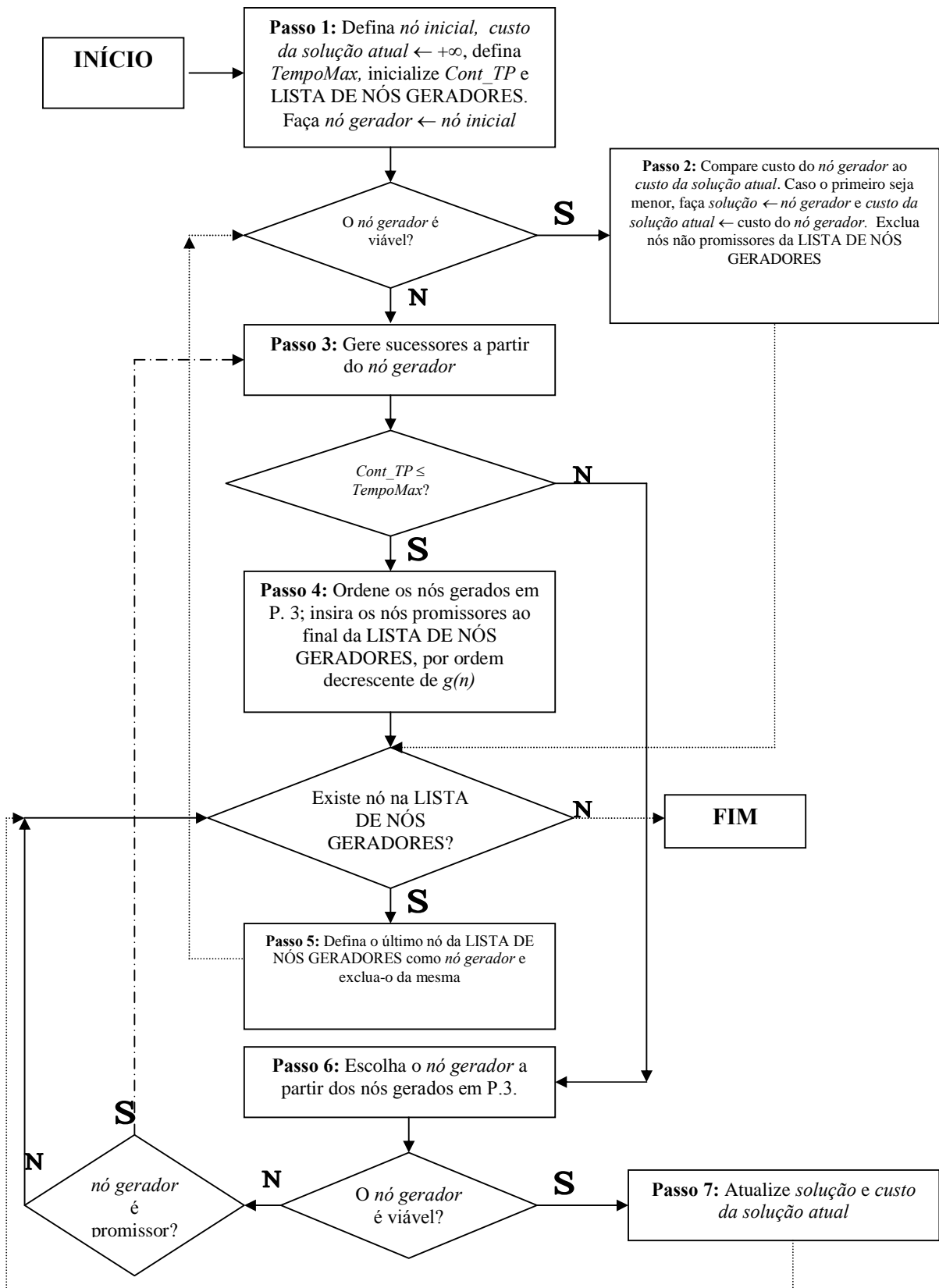


FIGURA 18: Fluxograma do algoritmo de busca vertical com limitação de custo/ busca ávida

Em função de já se ter apresentado uma figura ilustrativa para os processos de busca vertical com limitação de custo e de busca ávida, respectivamente, entende-se não ser necessário ilustrar (através de uma figura) o método híbrido. Resultados obtidos com a utilização do mesmo podem ser encontrados no próximo capítulo.

5.3.6 Algoritmo com lista de nós abertos e limitação de custo

5.3.6.1 Aspectos Introdutórios

A principal característica do tipo de processo implementado no algoritmo ora descrito diz respeito ao armazenamento de uma lista de nós, considerados promissores na obtenção de uma solução para o PPOC (esta lista será referenciada no algoritmo como LISTA DE ABERTOS). A expansão do grafo é realizada de maneira iterativa através da escolha, a cada iteração, de um nó desta lista para gerar sucessores.

Na modelagem concebida, se o critério adotado para escolha do *nó gerador* estiver associado ao custo do mesmo, o grafo tende a sofrer uma expansão em largura, que consiste na escolha (para expansão) de todos os nós de mesma profundidade antes que seus respectivos sucessores sejam selecionados, e assim sucessivamente até a obtenção da solução do problema. Por outro lado, se o critério adotado para escolha do *nó gerador* estiver associado a algum parâmetro que meça o grau de inviabilidade do mesmo, o processo se verticaliza, fazendo com que uma solução para o problema seja encontrada mais rapidamente. Em ambas as situações, nós com diferentes profundidades são comparados, a cada iteração, para escolha do nó a ser expandido.

Um processo de busca mais “horizontalizado”, se por um lado é mais eficaz (em termos de precisão da solução), por outro implica em um maior tempo de processamento, já que são promovidas diversas expansões até que se obtenha a primeira solução para o problema. Por sua vez, um processo mais “verticalizado” tende a ser mais rápido na obtenção da solução, que pode não ser muito precisa. Em função desta constatação, testaram-se diferentes critérios para escolha do nó a ser expandido, conforme será apresentado no próximo item.

5.3.6.2 Critérios considerados na escolha do *nó gerador*

Os critérios de escolha do *nó gerador* testados neste método de busca foram exatamente aqueles adotados no método de busca ávida, já apresentados no item 5.3.4.2. e citados novamente a seguir:

- a) Escolha do nó com a menor medida de inviabilidade, quantificada como a norma da *matriz de inviabilidades*;
- b) Escolha do nó com o menor *custo marginal de redução de inviabilidade*;
- c) Escolha do nó n com o menor valor de $\hat{f}(n)$, calculado como sendo a soma do custo $g(n)$ do nó com a estimativa de custo $\hat{h}(n)$ entre o mesmo e a solução do problema;
- d) Escolha do nó n com o menor valor de uma função de mérito do tipo $\text{função_mérito}(n) = g(n) + B \times \text{inv}(n)$;
- e) Escolha do nó com o menor valor de uma função de mérito adaptada (majoração do peso atribuído ao custo da sobra no cálculo do custo do nó).

A precisão da solução e o tempo de processamento de cada um dos critérios apresentados anteriormente estão associados também às características da referência e do cenário de demanda para os quais se busca a solução do PPOC. Os resultados obtidos a partir da utilização de cada um destes critérios poderão ser avaliados no próximo capítulo.

5.3.6.3 Exclusão de nós do processo de busca

Como forma de agilizar o processo de busca, utilizou-se o critério de comparar a estimativa $\hat{f}(n)$ de cada nó, candidato a entrar na LISTA DE ABERTOS, ao custo da melhor solução para o PPOC obtida até o momento. Portanto, só são acrescentados a esta lista os nós que potencialmente puderem, quando expandidos, propiciar uma solução melhor que a atual.

Conforme já salientado no item 5.3.6.1., dependendo do critério adotado para escolha do *nó gerador*, o processo pode culminar na realização de muitas iterações até se chegar na solução do problema e, conseqüentemente, exceder o tempo disponibilizado pela indústria para resolução do PPOC. Visando reduzir o tempo de processamento a padrões aceitáveis, restringiu-se o número máximo de nós que podem fazer parte da LISTA DE ABERTOS.

O número máximo de nós que pode fazer parte desta lista é especificado no início do algoritmo pelo usuário, conforme será apresentado nos itens subseqüentes. Visando atender a esta limitação, sempre que o número de nós inseridos na LISTA DE ABERTOS ultrapassar o

número máximo de nós definido previamente, há que se utilizar algum procedimento para exclusão de nós considerados menos promissores.

Os critérios para exclusão de nós da referida lista foram exatamente os mesmos já especificados para selecionar o *nó gerador* a cada iteração. Sendo assim, o usuário, ao definir no início do processo o critério para escolha do *nó gerador*, está também definindo o critério de exclusão de nós quando o número daqueles inseridos na lista ultrapassar o limite máximo permitido.

No próximo item, apresentam-se os passos do algoritmo referente ao processo de busca com lista de nós abertos e limitação de custo.

5.3.6.4 Algoritmo proposto para resolução do PPOC

O algoritmo para resolução do PPOC associado ao processo ora descrito envolve os seguintes passos:

- **Passo 1:** Defina o *nó inicial* com base no critério apresentado em 5.3.2.; inicialize a variável *custo da solução atual* com o valor $+\infty$ e defina a função de escolha do *nó gerador* e de exclusão de nós a ser utilizada ao longo do algoritmo, além do número máximo de nós permitido na LISTA DE ABERTOS (N_{max});
- **Passo 2:** Insira o *nó inicial* na LISTA DE ABERTOS;
- **Passo 3:** Se houver nós na LISTA DE ABERTOS, escolha o *nó gerador* dentre os nós inseridos na mesma, de acordo com a função de escolha definida no Passo 1. Se a LISTA DE ABERTOS estiver vazia, **pare**;
- **Passo 4:** Retire o *nó gerador*, escolhido no Passo 3, da LISTA DE ABERTOS;
- **Passo 5:** Gere sucessores a partir do *nó gerador* escolhido no Passo 3. Adicione os sucessores inviáveis, e com estimativa de custo $\hat{f}(n)$ menor que o *custo da solução obtida*, à LISTA DE ABERTOS. Verifique se há algum sucessor viável e com custo menor que o *custo da solução atual*. Caso haja, atualize *solução* e *custo da solução atual* (o nó associado à *solução* não entra na LISTA DE ABERTOS);
- **Passo 6:** Se o número de nós na LISTA DE ABERTOS for maior que N_{Max} , exclua nós desta lista até que restem apenas N_{Max} nós na mesma. O critério de exclusão é aquele associado à função de escolha definida no Passo 1. Retorne ao Passo 3.

Com relação a este algoritmo, é importante ressaltar uma particularidade associada ao processo de geração de sucessores. De maneira geral, este processo seguiu os passos apresentados em 5.2.4.. No entanto, o sucessor gerado a partir da inserção simultânea de

marcação e camada no par *tamanho x cor*, identificado como maior deficitário de peças, foi suprimido. Somente para o caso de uma mesa recém inserida no nó é que se admite a geração de um sucessor através do procedimento de inserção supracitado. Esta modificação foi efetuada como forma de reduzir a possibilidade de haver dois nós idênticos na LISTA DE ABERTOS.

O algoritmo anteriormente apresentado pode ser ilustrado através do fluxograma apresentado a seguir.

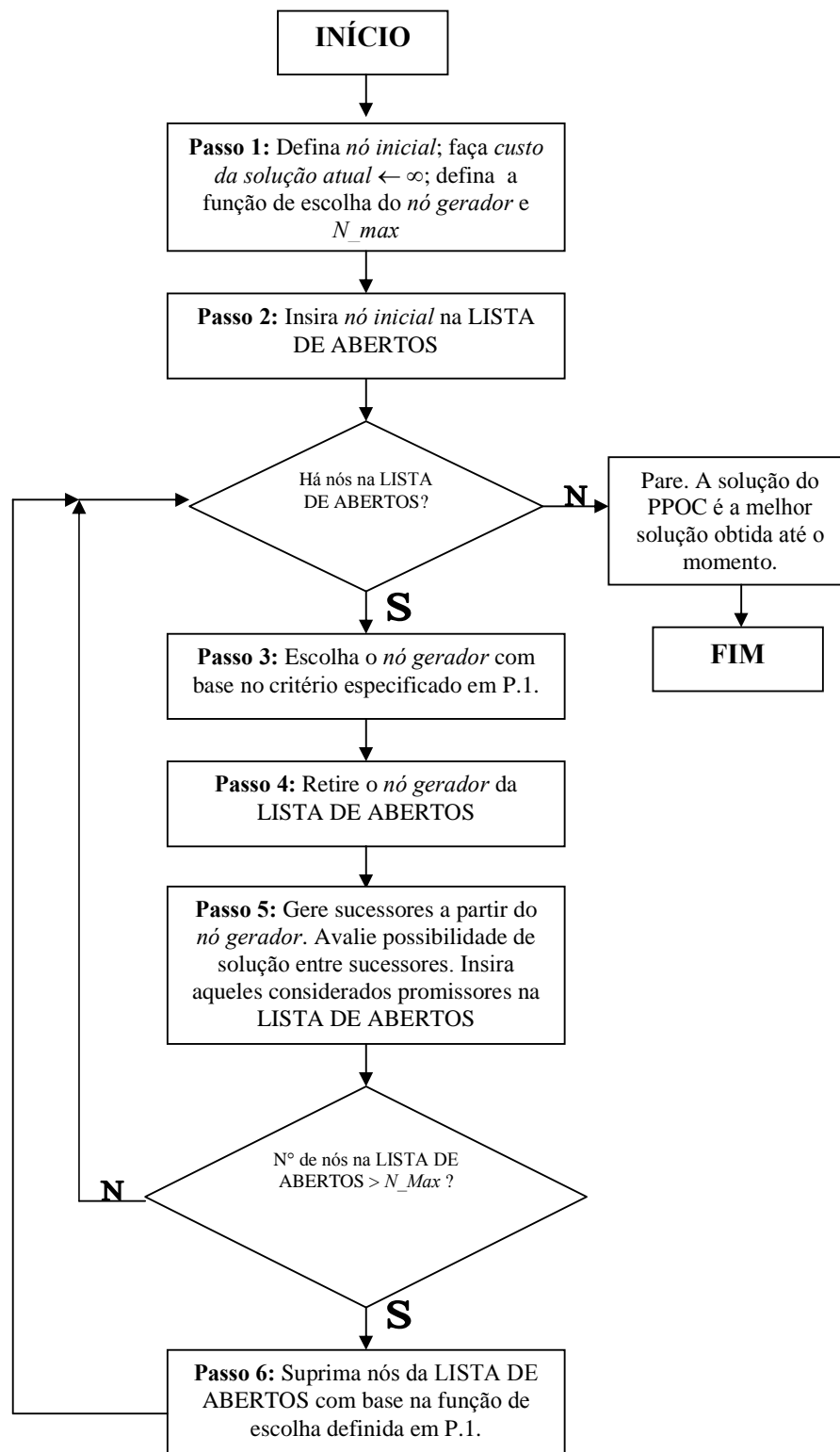
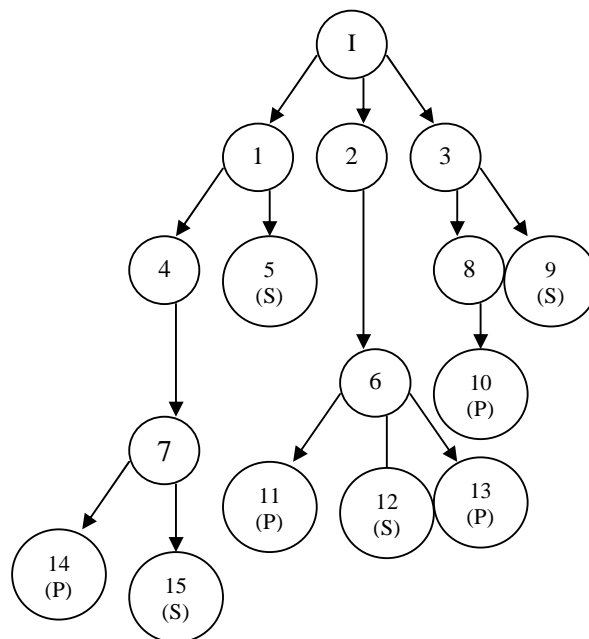


FIGURA 19: Fluxograma do algoritmo com lista de nós abertos e limitação de custo

Encontra-se apresentada a seguir uma figura que ilustra o funcionamento deste algoritmo. Da mesma forma que para as figuras anteriores, o número inserido em cada nó do grafo da figura 20 indica a ordem na qual o mesmo foi gerado. O símbolo (P) indica que o ramo associado ao referido nó foi podado no processo de busca, ou porque apresentou um custo maior do que a melhor solução obtida até o momento, ou porque foi excluído no intuito de manter um número máximo de nós, na LISTA DE ABERTOS, igual a N_{max} . O símbolo (S) indica que o respectivo nó representa uma solução para o problema.

FIGURA 20: Representação do processo de busca com lista de nós abertos e limitação de custo



5.3.7 Algoritmo de busca com lista de nós abertos e limitação de custo/ busca ávida

5.3.7.1 Aspectos Introdutórios

Conforme já salientado anteriormente, dependendo do critério que se utilize para escolha do nó da LISTA DE ABERTOS a gerar sucessores, pode-se incorrer em um tempo de processamento muito maior do que aquele aceitável pela indústria têxtil.

No caso do algoritmo de busca com lista de nós abertos e limitação de custo (apresentado em 5.3.6.), mesmo inserindo informação heurística ao processo de busca através da estimativa do custo $\hat{h}(n)$, verificou-se que o tempo de processamento ainda excedia o padrão aceitável, sendo necessário limitar o número de nós na LISTA DE ABERTOS, conforme já explicitado em 5.3.6.3.

Esta limitação reduziu o tempo de processamento, mas implicou no comprometimento da eficácia do método. Por este motivo, testou-se a possibilidade de limitar o tempo associado àquele processo de busca e, decorrido este tempo, realizar um processo de busca ávida até o esvaziamento completo da LISTA DE ABERTOS. Este processo alternativo de busca possibilitou um aumento do número máximo de nós que poderiam ser inseridos na LISTA DE ABERTOS, aumentando em muitos casos a precisão da solução obtida para as referências/cenários de demanda testados (conforme poderá ser visto no próximo capítulo).

Saliente-se que neste método híbrido, eliminou-se a estimativa do custo $\hat{f}(n)$ ao longo do processo de busca com lista de nós abertos e limitação de custo, sendo aquele parâmetro substituído pelo custo real de cada nó – $g(n)$ – no procedimento de comparação de custos para inserção de nós promissores na LISTA DE ABERTOS. Quanto aos critérios de escolha do *nó gerador*, continuou-se mantendo a possibilidade de escolher um dentre aqueles já citados em 5.3.6.2.

No que se refere ao processo de busca ávida, a partir de um determinado tempo de processamento, cada um dos nós remanescentes da LISTA DE ABERTOS é expandido. Quando um nó gerado a partir deste procedimento ultrapassa o custo da melhor solução obtida até o momento, seu respectivo ramo é eliminado da busca.

A seguir, encontram-se apresentados os passos de implementação do método híbrido ora descrito.

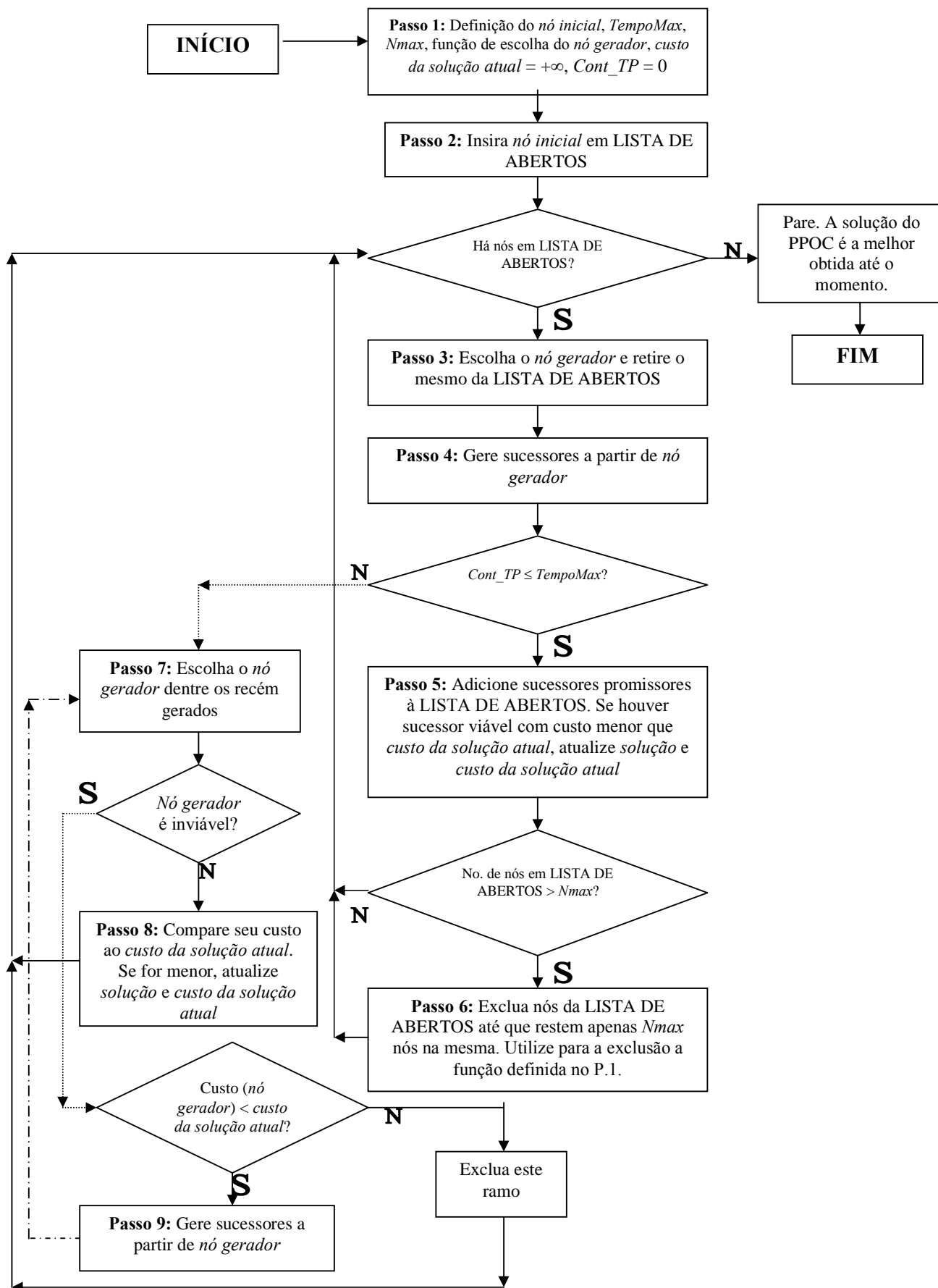
5.3.7.2 Algoritmo proposto para resolução do PPOC

Encontram-se apresentados a seguir os passos do algoritmo de busca com lista de nós abertos e limitação de custo/ busca ávida para resolução do PPOC.

- **Passo 1:** Defina o *nó inicial* com base no critério apresentado em 5.3.2.; inicialize a variável *custo da solução atual* com o valor $+\infty$ e defina a função de escolha do *nó gerador* e de exclusão de nós a ser utilizada ao longo do algoritmo, além do número máximo de nós permitido na LISTA DE ABERTOS (N_{max}). Defina o limite máximo de tempo para o processo de busca com lista de nós abertos e limitação de custo ($TempoMax$) e inicialize um contador de tempo ($Cont_{TP}$) ao algoritmo;
- **Passo 2:** Insira o *nó inicial* na LISTA DE ABERTOS;
- **Passo 3:** Se houver nós em LISTA DE ABERTOS, escolha o *nó gerador* dentre os nós inseridos na LISTA DE ABERTOS, de acordo com a função de escolha definida no Passo 1. Retire-o da LISTA DE ABERTOS e vá ao Passo 4. Se não houver mais nós em LISTA DE ABERTOS, **pare**;
- **Passo 4:** Gere sucessores a partir do *nó gerador*. Se $Cont_{TP} \leq TempoMax$, vá ao Passo 5. Senão, vá ao Passo 7;
- **Passo 5:** Adicione os sucessores inviáveis, e com custo $g(n)$ menor que o *custo da solução atual*, à LISTA DE ABERTOS. Verifique se há algum sucessor viável e com custo menor que o *custo da solução atual*. Caso haja, atualize *solução* e *custo da solução atual* (o *nó* associado à *solução* não entra na LISTA DE ABERTOS);
- **Passo 6:** Se o número de nós na LISTA DE ABERTOS for maior que N_{Max} , exclua nós desta lista até que restem apenas N_{Max} nós na mesma. O critério de exclusão é aquele associado à função de escolha definida no Passo 1. Retorne ao Passo 3;
- **Passo 7:** Escolha, dentre os nós recém-gerados, aquele com o menor custo $g(n)$. Chame-o de *nó gerador*;
- **Passo 8:** Avalie a viabilidade do *nó gerador*. Caso o mesmo seja viável, compare seu respectivo custo ao *custo da solução atual* (se o custo do *nó gerador* for menor, atualize *solução* e *custo da solução atual*) e retorne ao Passo 3. Caso o *nó gerador* seja inviável e apresente custo maior que o *custo da solução atual*, abandone-o e retorne ao Passo 3. Caso o *nó gerador* seja inviável e apresente custo menor que o *custo da solução atual*, vá ao Passo 9;
- **Passo 9:** Gere sucessores a partir do *nó gerador* e retorne ao Passo 7.

O fluxograma apresentado a seguir ilustra a estrutura do algoritmo anteriormente apresentado. Não se julgou necessário inserir figura que representasse seu funcionamento, pelo fato deste algoritmo consistir em um método híbrido entre o de lista de nós abertos com limitação de custo e o de busca ávida, e por já se ter representado (por meio de figura) ambos os processos de maneira isolada. Resultados obtidos a partir da utilização deste algoritmo podem ser avaliados no próximo capítulo.

FIGURA 21: Fluxograma do algoritmo com lista de nós abertos e limitação de custo/ busca ávida



Com base nos métodos expostos anteriormente, conceberam-se e testaram-se 17 diferentes processos de busca em grafos. Para melhor visualizar o processo envolvido na pesquisa ora apresentada, desde a identificação do PPOC até a concepção dos algoritmos de busca em grafos, encontra-se apresentado no Apêndice 02 um fluxograma que ilustra a ordem na qual cada uma das etapas do estudo foi realizada. O quadro a seguir apresenta todos os algoritmos testados (incluindo-se o método heurístico concebido e o *solver* MINLP).

QUADRO 02: Algoritmos testados na resolução do PPOC

ALGORITMO	DENOMINAÇÃO A SER CONSIDERADA NA APRESENTAÇÃO DOS RESULTADOS
Algoritmo heurístico proposto (item 4.5.2.)	Heurística proposta
Resultados obtidos a partir da utilização do <i>solver</i> MINLP (item 4.5.1.)	MINLP
Busca vertical com limitação de custo (item 5.3.3.)	BVLC
Busca ávida – Critério de escolha do <i>nó gerador</i> : menor inviabilidade [item 5.3.4.2(a)]	BA – a
Busca ávida – Critério de escolha do <i>nó gerador</i> : menor <i>custo marginal de redução de inviabilidade</i> [item 5.3.4.2(b)]	BA-b
Busca ávida – Critério de escolha do <i>nó gerador</i> : menor $\hat{f}(n)$ [item 5.3.4.2(c)]	BA – c
Busca ávida – Critério de escolha do <i>nó gerador</i> : menor valor da função de mérito [item 5.3.4.2(d)]	BA – d
Busca ávida – Critério de escolha do <i>nó gerador</i> : menor valor da função de mérito adaptada [item 5.3.4.2(e)]	BA – e
Busca vertical com limitação de custo/ busca ávida (item 5.3.5.)	BVLC/BA
Busca com lista de nós abertos e limitação de custo – critério de escolha do <i>nó gerador</i> : menor inviabilidade [item 5.3.6.2(a)]	BLNALC – a
Busca com lista de nós abertos e limitação de custo – critério de escolha do <i>nó gerador</i> : menor <i>custo marginal de redução de inviabilidade</i> [item 5.3.6.2(b)]	BLNALC – b
Busca com lista de nós abertos e limitação de custo – critério de escolha do <i>nó gerador</i> : menor valor de $\hat{f}(n)$ [item 5.3.6.2(c)]	BLNALC – c
Busca com lista de nós abertos e limitação de custo – critério de escolha do <i>nó gerador</i> : menor valor da função de mérito [item 5.3.6.2(d)]	BLNALC – d
Busca com lista de nós abertos e limitação de custo – critério de escolha do <i>nó gerador</i> : menor valor da função de mérito adaptada [item 5.3.6.2(e)]	BLNALC – e
Busca com lista de nós abertos e limitação de custo/ busca ávida – critério de escolha do <i>nó gerador</i> : menor inviabilidade [item 5.3.6.2(a)]	BLNALC/BA – a
Busca com lista de nós abertos e limitação de custo/ busca ávida – critério de escolha do <i>nó gerador</i> : menor <i>custo marginal de redução de inviabilidade</i> [item 5.3.6.2(b)]	BLNALC/BA – b
Busca com lista de nós abertos e limitação de custo/ busca ávida – critério de escolha do <i>nó gerador</i> : menor valor de $\hat{f}(n)$ [item 5.3.6.2(c)]	BLNALC/BA – c
Busca com lista de nós abertos e limitação de custo/ busca ávida – critério de escolha do <i>nó gerador</i> : menor valor da função de mérito [item 5.3.6.2(d)]	BLNALC/BA – d
Busca com lista de nós abertos e limitação de custo/ busca ávida – critério de escolha do <i>nó gerador</i> : menor valor da função de mérito adaptada [item 5.3.6.2(e)]	BLNALC/BA - e

Fonte: Autora

6 RESULTADOS OBTIDOS

6.1 REFERÊNCIAS TESTADAS

Depois de programados os algoritmos apresentados no capítulo anterior, realizaram-se testes dos mesmos através da resolução do PPOC para quatro referências de peças, considerando diferentes cenários de demanda atual e futura, respectivamente. Conforme já citado no capítulo I, a demanda atual é aquela prevista para o curto prazo (semana seguinte à realização do corte), sendo a demanda futura aquela prevista para o longo prazo (cerca de 15 dias após o corte das peças).

Com relação às referências testadas, saliente-se que os dados associados às mesmas foram obtidos junto a uma indústria de médio porte do segmento de confecção. Com base nestes dados, simularam-se outros cenários de demanda, além daqueles fornecidos pela indústria.

Encontram-se apresentadas, na tabela a seguir, informações acerca das quatro referências cujos dados de entrada para resolução do PPOC foram disponibilizados pela indústria tomada como base para avaliação dos algoritmos.

TABELA 02: Informações associadas a cada uma das referências disponibilizadas por uma indústria do segmento de confecção

PARÂMETROS	REFERÊNCIA 1	REFERÊNCIA 2	REFERÊNCIA 3	REFERÊNCIA 4
Número de cores	4	3	3	3
Número de tamanhos	6	6	6	7
Tem estampa?	Sim	Não	Sim	Sim
Comprimento da dobra (m)	0,04	0,04	0,04	0,04
Custo do tecido (R\$/kg)	12,96	16,46	15,96	18,02
Custo da matéria-prima (R\$/kg)	5,00	5,00	5,00	5,00
Comprimento máximo do enfiado (m)	13,00	13,00	13,00	7,50
Comprimento tam. 1 (m)	0,2625	0,5370	0,2758	0,3275
Comprimento tam. 2 (m)	0,3083	0,5850	0,3027	0,3700
Comprimento tam. 3 (m)	0,3550	0,6483	0,3307	0,4179
Comprimento tam. 4 (m)	0,4100	0,8210	0,3702	0,4795
Comprimento tam. 5 (m)	0,4567	0,9172	0,4011	0,5365
Comprimento tam. 6 (m)	0,5065	1,0470	0,4335	0,5940
Comprimento tam. 7 (m)	-	-	-	0,6545
Gramatura do tecido (kg/m ²)	0,248	0,198	0,216	0,211
Largura do tecido (m)	1,65	1,60	1,89	1,88
Nº máximo de camadas	60	100	80	72
Nº padrão de marcações do risco da referência	16	37	71	25
Nº médio de quadros de referência	7	0	12	6
% de aproveitamento do encaixe padrão	0,8693	0,8384	0,8482	0,7904
Perímetro padrão (m)	78,00	208,95	754,98	85,62
Preço de venda (R\$/peça)	20,00	27,43	18,68	18,23
% de desconto no preço em promoções	50	50	50	50

Fonte: Indústria têxtil

As tabelas a seguir apresentam os valores de demanda atual prevista para cada uma das referências supracitadas.

TABELA 03: Demanda atual (em peças) associada à Referência 1

	TAM 1	TAM 2	TAM 3	TAM 4	TAM 5	TAM 6
COR 1	47	73	119	109	95	71
COR 2	78	117	120	86	78	92
COR 3	68	59	105	104	122	133
COR 4	28	78	24	64	63	64

Fonte: Indústria têxtil

TABELA 04: Demanda atual (em peças) associada à Referência 2

	TAM 1	TAM 2	TAM 3	TAM 4	TAM 5	TAM 6
COR 1	46	91	162	142	150	107
COR 2	141	137	159	162	179	143
COR 3	70	136	157	110	88	117

Fonte: Indústria têxtil

TABELA 05: Demanda atual (em peças) associada à Referência 3

	TAM 1	TAM 2	TAM 3	TAM 4	TAM 5	TAM 6
COR 1	95	108	77	103	80	62
COR 2	64	100	60	61	108	78
COR 3	46	81	132	73	84	85

Fonte: Indústria têxtil

TABELA 06: Demanda atual (em peças) associada à Referência 4

	TAM 1	TAM 2	TAM 3	TAM 4	TAM 5	TAM 6	TAM 7
COR 1	100	93	122	66	30	48	33
COR 2	83	115	97	56	42	39	33
COR 3	145	125	174	93	76	44	27

Fonte: Indústria têxtil

A indústria têxtil cujos dados foram avaliados forneceu como previsão de demanda futura, para cada uma das quatro referências, os mesmos valores já fornecidos com relação à previsão de demanda atual de cada par *tamanho x cor* das peças.

Com o intuito de dispor de um maior número de resultados obtidos a partir dos algoritmos de busca em grafos definidos durante a pesquisa, simularam-se diferentes cenários de demanda, além daqueles disponibilizados pela indústria têxtil.

Para isto, procurou-se resolver o PPOC considerando as seguintes variações quanto às demandas - atual e futura – de cada uma das quatro referências de peças:

- Valores de demanda atual, para cada par *tamanho x cor* de peça, calculados como o produto dos valores originalmente fornecidos pela indústria por um coeficiente que variou de 0,8 a 1,5;
- Valores de demanda futura iguais à demanda atual e à metade da demanda atual, respectivamente, bem como um terceiro cenário que considerou valores de demanda futura iguais a zero (este último cenário buscou representar o final de estação).

A tabela a seguir ilustra os critérios utilizados na construção dos demais cenários de demanda testados.

TABELA 07: Critérios utilizados na construção dos demais cenários de demanda

Cenário	Demanda atual	Demanda futura	Demais parâmetros
Dados_R1_gera1	= 0,8 × demanda da Referência 1	Igual à demanda atual	Iguais aos da Referência 1
Dados_R1_gera2	= 0,8 × demanda da Referência 1	Igual à metade da demanda atual	Iguais aos da Referência 1
Dados_R1_gera3	= 0,8 × demanda da Referência 1	Igual a zero	Iguais aos da Referência 1
Dados_R1_gera4	= 1,2 × demanda da Referência 1	Igual à demanda atual	Iguais aos da Referência 1
Dados_R1_gera5	= 1,2 × demanda da Referência 1	Igual à metade da demanda atual	Iguais aos da Referência 1
Dados_R1_gera6	= 1,2 × demanda da Referência 1	Igual a zero	Iguais aos da Referência 1
Dados_R1_gera7	= 1,5 × demanda da Referência 1	Igual à demanda atual	Iguais aos da Referência 1
Dados_R1_gera8	= 1,5 × demanda da Referência 1	Igual à metade da demanda atual	Iguais aos da Referência 1
Dados_R1_gera9	= 1,5 × demanda da Referência 1	Igual a zero	Iguais aos da Referência 1
Dados_R2_gera1	= 0,8 × demanda da Referência 2	Igual à demanda atual	Iguais aos da Referência 2
Dados_R2_gera2	= 0,8 × demanda da Referência 2	Igual à metade da demanda atual	Iguais aos da Referência 2
Dados_R2_gera3	= 0,8 × demanda da Referência 2	Igual a zero	Iguais aos da Referência 2
Dados_R2_gera4	= 1,2 × demanda da Referência 2	Igual à demanda atual	Iguais aos da Referência 2
Dados_R2_gera5	= 1,2 × demanda da Referência 2	Igual à metade da demanda atual	Iguais aos da Referência 2
Dados_R2_gera6	= 1,2 × demanda da Referência 2	Igual a zero	Iguais aos da Referência 2
Dados_R2_gera7	= 1,5 × demanda da Referência 2	Igual à demanda atual	Iguais aos da Referência 2
Dados_R2_gera8	= 1,5 × demanda da Referência 2	Igual à metade da demanda atual	Iguais aos da Referência 2
Dados_R2_gera9	= 1,5 × demanda da Referência 2	Igual a zero	Iguais aos da Referência 2
Dados_R3_gera1	= 0,8 × demanda da Referência 3	Igual à demanda atual	Iguais aos da Referência 3
Dados_R3_gera2	= 0,8 × demanda da Referência 3	Igual à metade da demanda atual	Iguais aos da Referência 3
Dados_R3_gera3	= 0,8 × demanda da Referência 3	Igual a zero	Iguais aos da Referência 3
Dados_R3_gera4	= 1,2 × demanda da Referência 3	Igual à demanda atual	Iguais aos da Referência 3
Dados_R3_gera5	= 1,2 × demanda da Referência 3	Igual à metade da demanda atual	Iguais aos da Referência 3
Dados_R3_gera6	= 1,2 × demanda da Referência 3	Igual a zero	Iguais aos da Referência 3
Dados_R4_gera1	= 0,8 × demanda da Referência 4	Igual à demanda atual	Iguais aos da Referência 4
Dados_R4_gera2	= 0,8 × demanda da Referência 4	Igual à metade da demanda atual	Iguais aos da Referência 4
Dados_R4_gera3	= 0,8 × demanda da Referência 4	Igual a zero	Iguais aos da Referência 4

Cenário	Demanda atual	Demanda futura	Demais parâmetros
Dados_R4_gera4	= 1,2 × demanda da Referência 4	Igual à demanda atual	Iguais aos da Referência 4
Dados_R4_gera5	= 1,2 × demanda da Referência 4	Igual à metade da demanda atual	Iguais aos da Referência 4
Dados_R4_gera6	= 1,2 × demanda da Referência 4	Igual a zero	Iguais aos da Referência 4

Fonte: Autora

No próximo item, é apresentado o método heurístico atualmente utilizado pela indústria têxtil na resolução do PPOC. Além deste método, foram testados para cada situação avaliada os processos de busca em grafos ora concebidos (apresentados no capítulo anterior), além do método heurístico também criado ao longo desta pesquisa e do MINLP.

6.2 ALGORITMO ATUALMENTE UTILIZADO PELA INDÚSTRIA TÊXTIL

A indústria que forneceu os dados para realização dos testes ora apresentados dispõe, atualmente, de um método heurístico utilizado pelo programador das OC's, método este que subsidia a resolução do problema. De posse dos resultados obtidos a partir deste método, o programador ainda implementa (com base em sua experiência) modificações na configuração de cada mesa, visando diminuir sobras de peças e, assim, diminuir o custo associado às OC's.

O objetivo do método atualmente adotado pela indústria consiste em diminuir sobras de peças. No entanto, considera-se como “sobra”, no referido método, tanto as peças que possam atender a uma demanda futura como aquelas que efetivamente tenham de ser comercializadas a preços promocionais. Os demais custos inerentes ao PPOC não são considerados, por este método, na definição das OC's.

Visando diminuir as sobras, a heurística inserida no método proporcionaliza, em cada mesa, o número de marcações em função da demanda prevista. Desta forma se, por exemplo, para uma determinada referência tiver sido constatada uma demanda de 30% do total para o tamanho 1, 30% para o tamanho 2 e 40% para o tamanho 3, respectivamente, o número de marcações na(s) mesa(s) programada(s) terá de respeitar esta proporção para cada tamanho. Além disso, só são reunidas duas ou mais cores em uma mesma mesa se a relação entre a capacidade desta mesa (medida em quantidade de peças) e o número de cores a serem agregadas for maior ou igual à demanda individual de peças para cada uma dessas cores.

O algoritmo programado para reproduzir a heurística inserida no método pode ser descrito através dos seguintes passos:

- **Passo 1:** Leia os dados da referência;

- **Passo 2:** Faça $Demanda_n\tilde{a}o_atendida \leftarrow$ Demanda da refer\ecencia. Faça $mesa = 1$;
- **Passo 3:** Calcule o n\umero de pe\cas que cabem em uma *mesa*, com base nos par\ametros: comprimento m\edio das pe\cas da refer\ecencia, comprimento m\aximo de enfesto e n\umero m\aximo de camadas;
- **Passo 4:** Se $Demanda_n\tilde{a}o_atendida$ for maior que zero, proporcionalize o n\umero de pe\cas alocadas, por camada para cada tamanho, em fun\c\ao do percentual que representa a demanda daquele tamanho com rela\c\ao \aa $Demanda_n\tilde{a}o_atendida$. Se $Demanda_n\tilde{a}o_atendida$ for igual a zero, **pare**;
- **Passo 5:** Compare o comprimento de enfesto gerado em fun\c\ao da aloca\c\ao de pe\cas (promovida no passo anterior) ao comprimento m\aximo de enfesto permitido para a refer\ecencia. Se o comprimento de enfesto gerado for maior que o comprimento m\aximo, suprima pe\cas (por ordem de tamanho – tamanho 1, 2, ..., n) at\ea que o comprimento de enfesto atinja o limite permitido;
- **Passo 6:** Calcule o n\umero de camadas necess\arias, por cor, para atendimento \aa $Demanda_n\tilde{a}o_atendida$, em fun\c\ao da aloca\c\ao de marca\c\oes promovida nos Passos 4 e 5;
- **Passo 7:** Junte duas ou mais cores da refer\ecencia em uma mesma *mesa*, caso a rela\c\ao entre o n\umero de pe\cas que cabem na *mesa* e o n\umero de cores a serem reunidas seja igual ou maior que a demanda de pe\cas para cada uma dessas cores;
- **Passo 8:** Atualize $Demanda_n\tilde{a}o_atendida$, suprimindo de cada par *tamanho x cor* a demanda atendida na mesa cujo n\umero de marca\c\oes (por tamanho) e de camadas (por cor) acabou de ser definido. Faça $mesa = mesa + 1$. Volte ao Passo 4.

Com rela\c\ao a este algoritmo, vale ressaltar que todos os passos foram constru\idos com base na explica\c\ao, fornecida pelo pr\oprio programador das OC's, acerca do m\etodo adotado pela ind\ustria. As modifica\c\oes posteriores que possam ser feitas por este profissional, na tentativa de diminuir sobras, n\ao seguem regras pr\ea-concebidas e que sejam reprodu\iveis em um algoritmo. Portanto, entende-se que os resultados, obtidos com a utiliza\c\ao do algoritmo apresentado neste item, consistem na melhor aproxima\c\ao que se pode ter acerca das OC's que seriam definidas por esta ind\ustria para as refer\encias/ cen\arios ora avaliados.

Encontram-se apresentados a seguir os resultados obtidos para os cen\arios de demanda testados, considerando os diferentes algoritmos listados no quadro 02. Para realiza\c\ao dos testes, utilizou-se um Pentium III com 700 MHz de *clock* e mem\oria de 256 MB.

6.3 RESULTADOS OBTIDOS PARA REFERÊNCIAS COM CENÁRIOS DE DEMANDA CONSTRUÍDOS A PARTIR DE INFORMAÇÕES REAIS

A título ilustrativo, encontram-se apresentados a seguir os resultados obtidos para o primeiro cenário de demanda construído a partir de informações reais. Os resultados obtidos para os demais cenários simulados encontram-se apresentados no Apêndice 05.

QUADRO 03: Resultados obtidos para o cenário Dados_R1_geral

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 10 12 12 8	412,09	0,0001
Heurística proposta	3 5 6 5 5 6 10 12 11 8	354,99	0,11
MINLP	2 3 3 3 4 4 18 18 16 8 0 2 0 1 0 0 0 0 0 6	139,03	Não Aplicável
BVLC	3 6 7 5 6 6 9 9 9 3 3 3 0 3 2 4 2 3 2 6	149,39	32,19
BA – a	3 7 6 5 5 7 10 12 11 6	416,59	4,28
BA-b	3 6 7 5 6 6 9 9 9 3 3 3 0 3 2 4 2 3 2 6	149,39	5,77
BA – c	3 6 7 5 6 6 9 9 9 3 3 3 0 3 2 4 2 3 2 6	149,39	5,38
BA – d	3 6 7 5 6 6 9 9 9 3 3 3 0 3 2 4 2 3 2 6	149,39	4,73
BA - e	3 6 7 5 6 6 9 9 9 3 3 3 0 3 2 4 2 3 2 6	149,39	4,73
BVLC/BA	4 6 6 5 6 6 9 9 8 4 0 2 0 3 2 3 2 0 5 6	140,13	273,48
BLNALC – a	4 6 6 5 6 6 10 9 9 4 0 3 0 2 1 4 0 0 2 5	133,16	351,46
BLNALC – b	4 5 7 6 5 6 9 10 11 3 1 4 0 2 2 2 0 1 1 7	141,68	130,06
BLNALC – c	4 5 7 6 5 6 9 10 11 3 1 4 0 2 2 2 0 1 1 7	141,68	154,62
BLNALC – d	3 5 6 6 6 6 9 10 10 4 1 2 0 1 1 1 0 7 1 10	145,57	396,89
BLNALC - e	3 6 7 6 5 6 8 9 9 3 3 3 0 2 5 4 1 3 2 6	148,38	31,8
BLNALC/BA – a	4 6 6 6 6 5 9 9 9 4 0 1 0 1 1 3 0 0 6 12	145,51	176,31
BLNALC/BA – b	3 5 7 6 5 6 8 8 9 3 4 6 0 3 5 4 1 3 2 4	144,65	179
BLNALC/BA – c	3 6 7 6 5 6 8 10 8 3 2 3 0 2 4 4 2 3 4 6	150,02	181,04
BLNALC/BA – d	4 6 6 6 5 6 9 9 9 4 0 2 0 1 4 3 0 0 3 6	134,32	192,79
BLNALC/BA - e	4 6 6 6 5 6 9 9 9 4 0 2 0 1 4 3 0 0 3 6	134,32	199,33

Fonte: Autora

O nó obtido com a heurística da indústria, por exemplo, está representando uma OC composta por uma mesa, cujos elementos representam, respectivamente, o número de marcações e de camadas. Assim, foram alocadas 4, 5, 6, 6, 6 e 6 marcações para os tamanhos de 1 a 6. Na seqüência, foram alocadas 10, 12, 12 e 8 camadas para as cores de 1 a 4, respectivamente.

6.4 RESULTADOS OBTIDOS PARA OS CENÁRIOS DE DEMANDA FORNECIDOS PELA INDÚSTRIA

A seguir apresentam-se os resultados obtidos para os cenários de demanda efetivamente fornecidos para a indústria têxtil.

QUADRO 04: Resultados obtidos para a Referência 1

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 20 24 16 0 2 7 3 6 7 7 0 0 6 14	286,67	0,0001
Heurística proposta	4 5 6 6 6 6 18 18 17 7 0 9 0 4 5 9 0 2 2 4	196,64	0,11
MINLP	4 5 6 6 5 4 18 19 16 7 0 4 0 2 4 7 0 3 8 9	187,91	Não aplicável
BVLC	4 6 6 5 6 6 18 16 18 8 1 4 1 3 2 6 3 12 2 6	274,38	107,93
BA – a	3 5 6 5 6 7 16 18 15 11 4 4 2 4 3 3 6 5 7 4	472,94	16,14
BA-b	4 6 6 5 6 6 18 16 18 8 1 4 1 3 2 6 3 12 2 6	274,38	23,9
BA – c	4 6 6 5 6 6 18 16 18 8 1 4 1 3 2 6 3 12 2 6	274,38	23,51
BA – d	4 6 6 5 6 6 18 16 18 8 1 3 1 3 2 4 3 12 3 8	293,67	19,94
BA - e	4 6 6 5 6 6 18 16 18 8 1 3 1 3 2 4 3 12 3 8	293,67	21,14
BVLC/BA	3 6 7 5 6 6 16 18 18 6 3 3 0 4 2 4 5 7 3 12	213,28	3669,7
BLNALC – a	3 6 7 5 6 6 16 18 17 9 4 3 0 5 2 3 4 5 6 6	377,43	59,65
BLNALC – b	3 5 5 5 7 7 16 15 18 9 4 13 6 7 0 2 5 12 2 2	326,40	995,41
BLNALC – c	3 5 5 5 7 7 16 15 18 9 4 13 6 7 0 2 5 12 2 2	326,40	1012
BLNALC – d	3 5 6 5 6 7 17 18 17 8 2 3 1 2 2 1 8 9 6 13	352,26	1310,6
BLNALC - e	3 6 6 7 6 5 17 18 17 8 1 2 2 1 1 3 5 17 16 22	890,74	1913
BLNALC/BA – a	3 6 7 5 6 6 16 21 17 6 2 2 0 2 2 3 11 6 8 19	249,32	234,65
BLNALC/BA – b	4 5 7 6 5 6 16 18 17 6 1 2 0 2 2 2 6 10 16 21	252,16	406,5
BLNALC/BA – c	3 6 7 6 5 6 16 21 17 6 2 2 0 2 4 3 4 6 8 18	231,24	412,98
BLNALC/BA – d	3 5 5 5 6 7 16 16 19 9 5 13 6 7 2 0 5 6 1 2	303,62	416
BLNALC/BA - e	4 6 5 6 6 6 15 18 18 9 0 3 2 1 1 3 17 9 4 6	336,21	437,43

Fonte: Autora

QUADRO 05: Resultados obtidos para a Referência 2

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 71 0 0 1 3 3 3 3 3 0 100 0 2 3 4 3 2 3 0 21 46	5380,9	0,0001
Heurística proposta	2 2 3 3 3 2 27 37 35 2 3 3 2 3 3 25 27 18	446,46	0,11
MINLP	0 0 3 4 6 2 21 22 9 4 4 4 2 1 3 22 32 32	230,84	Não aplicável
BVLC	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 12 28 15	264,74	109,08
BA – a	2 3 4 3 3 2 30 41 29 4 2 0 2 4 4 8 12 7 0 3 4 3 2 3 8 0 8	282,44	21,92
BA-b	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 12 28 15	264,74	28,62
BA – c	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 12 28 15	264,74	27,52
BA – d	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 13 28 27	296,28	31,91
BA - e	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 13 28 15	266,16	27,96
BVLC/BA	2 3 4 3 3 2 33 36 31 2 2 2 3 3 3 12 28 15	264,74	3617,09
BLNALC – a	2 3 4 3 3 2 30 41 29 3 1 0 2 3 6 12 15 10 0 4 5 2 2 0 7 0 7	341,67	72,77
BLNALC – b	2 3 4 3 3 2 30 40 30 2 2 0 3 4 4 10 20 10 1 2 3 1 1 1 9 7 8	303,39	962,79
BLNALC – c	2 3 4 3 3 2 30 40 30 2 2 0 3 4 4 10 20 10 1 2 3 1 1 1 9 7 8	303,39	1117,80
BLNALC – d	2 2 3 3 3 3 34 34 32 2 2 3 2 3 1 16 47 37	875,67	1767,2
BLNALC - e	2 3 3 3 3 2 30 42 28 2 2 3 2 2 2 27 39 34	639,90	2684,6
BLNALC/BA – a	2 3 4 3 3 2 30 41 29 3 1 0 2 3 6 10 15 10 0 2 2 2 1 0 15 0 13	288,90	232,5
BLNALC/BA – b	2 3 4 3 3 2 27 38 35 2 1 2 3 3 3 19 26 18	275,35	548,38
BLNALC/BA – c	2 3 4 3 3 2 27 38 35 2 1 2 3 3 3 19 26 18	275,35	420,91
BLNALC/BA – d	2 3 4 3 3 2 27 38 35 2 1 2 3 3 3 19 26 18	275,35	573,20
BLNALC/BA - e	2 3 4 3 3 2 27 38 35 2 1 2 3 3 3 19 26 18	275,35	590,18

Fonte: Autora

QUADRO 06: Resultados obtidos para a Referência 3

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 24 16 19	406,49	0,0001
Heurística proposta	5 7 6 6 7 5 18 15 20	253,73	0,11
MINLP	4 5 6 4 5 6 24 20 20	364,44	Não aplicável
BVLC	5 6 7 6 7 5 18 15 17	172,77	50,20
BA – a	5 7 8 6 7 4 17 18 19 1 0 0 0 0 1 1 0 1	464,15	9,01
BA-b	5 6 7 6 7 5 18 15 17	172,77	8,08
BA – c	5 6 7 6 7 5 18 15 17	172,77	11,71
BA – d	5 7 6 6 7 5 18 15 18 0 0 1 0 0 0 0 0 1 1	192,17	12,63
BA - e	4 7 7 6 7 5 17 15 17 2 0 0 0 0 0 9 0 0	186,92	12,92
BVLC/BA	5 6 7 5 7 5 19 15 17	170,24	509,38
BLNALC – a	5 6 8 5 7 5 19 15 16	172,61	31,42
BLNALC – b	5 7 6 6 7 5 18 15 17	190,86	477,52
BLNALC – c	5 7 6 6 7 5 18 15 17	190,86	566,94
BLNALC – d	4 6 6 5 6 4 19 18 20 1 0 0 0 0 0 1 1 0 0	181,69	799,55
BLNALC - e	4 6 5 5 5 4 22 20 20 0 0 1 0 0 0 0 0 3 0	207,28	1140,8
BLNALC/BA – a	5 6 8 5 7 5 19 15 16	172,61	170,21
BLNALC/BA – b	5 6 7 6 6 5 18 17 17	172,97	198,06
BLNALC/BA – c	5 6 7 6 6 5 18 17 17	172,97	202,45
BLNALC/BA – d	4 6 6 5 5 4 22 20 20	175,39	261,44
BLNALC/BA - e	4 6 6 5 5 4 22 20 20	175,39	239,36

Fonte: Autora

QUADRO 07: Resultados obtidos para a Referência 4

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 48 24 0 3 3 4 2 2 1 1 0 15 49	253,04	0,0001
Heurística proposta	3 3 4 2 1 1 1 21 18 33 3 4 2 2 2 2 1 13 13 18	207,37	0,11
MINLP	0 7 2 1 0 3 2 15 15 14 6 1 5 3 3 0 0 16 13 26	179,95	Não aplicável
BVLC	4 3 5 2 1 1 1 16 21 35 2 4 2 2 3 2 1 15 11 12	210,08	111,33
BA – a	4 3 5 2 1 1 1 16 23 33 2 6 2 2 4 1 0 15 5 9 0 2 0 0 0 3 3 5 3 0	364,07	11,78
BA-b	4 3 5 2 1 1 1 16 21 35 2 4 2 2 3 2 1 15 11 12	210,08	14,34
BA – c	4 3 5 2 1 1 1 16 21 35 2 4 2 2 3 2 1 15 11 12	210,08	13,73
BA – d	4 3 5 2 1 1 1 16 21 35 2 5 2 2 2 2 1 22 10 17	225,24	17,03
BA - e	4 3 5 2 1 1 1 16 21 35 2 5 2 2 2 2 1 22 10 17	225,24	12,47
BVLC/BA	4 3 5 2 1 1 1 17 23 32 2 4 2 2 3 2 1 14 9 13	196,13	1932,8
BLNALC – a	4 3 5 2 1 1 1 16 23 33 1 5 2 2 4 2 0 14 7 9 3 0 1 0 0 0 4 4 2 0	322,08	36,47
BLNALC – b	4 3 5 2 1 1 1 17 21 34 0 6 1 2 5 2 0 8 7 8 3 0 2 1 0 2 3 10 3 0	214,89	473,35
BLNALC – c	4 3 5 2 1 1 1 17 21 34 0 6 1 2 5 2 0 8 7 8 3 0 2 1 0 2 3 10 3 0	214,89	564,14
BLNALC – d	4 3 5 2 1 1 1 17 20 35 2 4 2 2 3 2 1 14 12 12	200,56	760,39
BLNALC - e	3 4 3 2 1 2 1 18 24 30 3 1 7 2 3 1 0 10 4 14 2 2 0 1 1 0 4 4 2 0	206,04	1252
BLNALC/BA – a	4 3 5 2 1 1 1 17 21 34 2 4 2 2 3 2 1 14 11 12	198,34	210,81
BLNALC/BA – b	4 3 5 2 1 1 1 17 24 31 2 4 2 2 3 2 1 14 8 13	193,99	360,31
BLNALC/BA – c	4 3 5 2 1 1 1 17 24 31 2 4 2 2 3 2 1 14 8 13	193,99	275,34
BLNALC/BA – d	4 4 4 2 1 1 1 19 22 31 1 2 2 2 2 2 1 20 8 21	214,45	330,16
BLNALC/BA - e	4 4 4 2 1 1 1 19 22 31 1 2 2 2 2 2 1 20 8 21	214,45	335,43

Fonte: Autora

6.5 ANÁLISE DOS RESULTADOS

6.5.1. Avaliação da performance dos algoritmos em termos de custo e tempo de processamento

Para avaliar a performance dos algoritmos, procurou-se considerar simultaneamente dois critérios: custo da solução obtida e tempo de processamento.

Conforme já salientado, considera-se como solução para o PPOC aquela que respeite as restrições do problema e que esteja associada ao menor custo possível. No entanto, o tempo disponível para obtenção desta solução costuma ser bastante restrito, especificamente para aquelas indústrias que não dispõem de supermercado de malhas e que dependem de tais resultados para inicializarem a movimentação de sua respectiva cadeia de suprimentos.

Com base nesta realidade e nas informações adquiridas junto ao programador de OC's da indústria em questão, definiu-se um tempo de 600 segundos (dez minutos) como o tempo máximo admissível para obtenção da solução do PPOC para cada uma das referências envolvidas em uma ordem de programação semanal.

Como outra premissa definida para avaliação da performance dos algoritmos, considerou-se a necessidade de que o mesmo fosse capaz de resolver o PPOC independentemente dos valores associados aos parâmetros de entrada. Desta forma, mesmo para demandas futuras tendendo a zero, o algoritmo tem que conseguir gerar uma solução para o problema; caso contrário, não se estaria apto a resolver o PPOC em épocas de final de estação. Segundo este critério, o MINLP (acessado a partir do *site* NEOS) deveria ser descartado num primeiro momento, já que não foi capaz de gerar solução para cenários associados à demanda futura zero.

Com base nos 34 casos testados e desconsiderando os resultados obtidos com a utilização do MINLP, verificou-se que o algoritmo que gerou os melhores resultados em um tempo de processamento dentro do limite considerado aceitável foi o **BLNALC/BA – e**, seguido do **BLNALC/BA – a** e do algoritmo **BVLC/BA**. Caso se desconsidere a limitação de tempo, o algoritmo que gerou os melhores resultados de uma maneira geral foi o **BVLC/BA**. Vale lembrar que, nas presentes análises, desconsideraram-se resultados obtidos com o MINLP em função de não se ter podido obter resultados (através deste *solver*) para todos os cenários avaliados. Saliente-se que a não obtenção de solução, quando da utilização do MINLP para resolver o PPOC com demanda futura zero, possivelmente decorre de uma limitação de tempo inerente ao NEOS, não estando associada especificamente ao *solver*.

Quanto aos resultados obtidos, é importante ressaltar que para três situações dentre todas as analisadas (desconsiderando aquelas para as quais não se conseguiu obter resultados a partir do MINLP), os algoritmos propostos neste trabalho propiciaram resultados melhores (em termos de custo) do que aqueles obtidos a partir do referido *solver*.

6.5.2 Avaliação da performance dos algoritmos com base na metodologia proposta por Dolan e Moré (2000)

Dolan e Moré (2000) propõem, em seu trabalho, a comparação da performance de diferentes métodos de resolução de um problema, comparação esta realizada com base na construção de curvas de probabilidade acumulada. Estas curvas, construídas a partir dos resultados obtidos para vários problemas testados, devem expressar a probabilidade acumulada de que determinado método alcance um certo nível de performance – comparativamente à performance do método tido como mais eficiente.

Com base nesta idéia, Leyffer (2003) comparou diferentes métodos de resolução de problemas de programação não linear a partir da análise de 150 problemas desta natureza. Para cada um dos problemas escolhidos previamente, o autor buscou solucioná-lo através de cada um dos métodos envolvidos na análise. De posse dos resultados o autor construiu, para cada método, uma função de probabilidade acumulada associada à seguinte medida de performance:

$$Eficiência(método_x, p) = \log_2 \left[\frac{TP(método_x, p)}{TP(método_mais_eficiente, p)} \right]$$

Onde:

- $Eficiência(método_x, p)$ = Eficiência do *método x* na resolução do problema *p*;
- \log_2 = Logaritmo na base 2;
- $TP(método_x, p)$ = Tempo de Processamento associado ao *método x* na resolução do problema *p*;
- $TP(método_mais_eficiente, p)$ = Tempo de Processamento associado ao *método mais eficiente* na resolução do problema *p*.

Seguindo esta idéia, buscou-se avaliar a performance dos métodos de resolução do PPOC, performance esta medida em termos de custo da solução obtida. Julgou-se inadequado comparar os métodos em função de seu respectivo tempo de processamento, já que para alguns deles (especificamente os métodos híbridos) o parâmetro tempo foi limitado visando

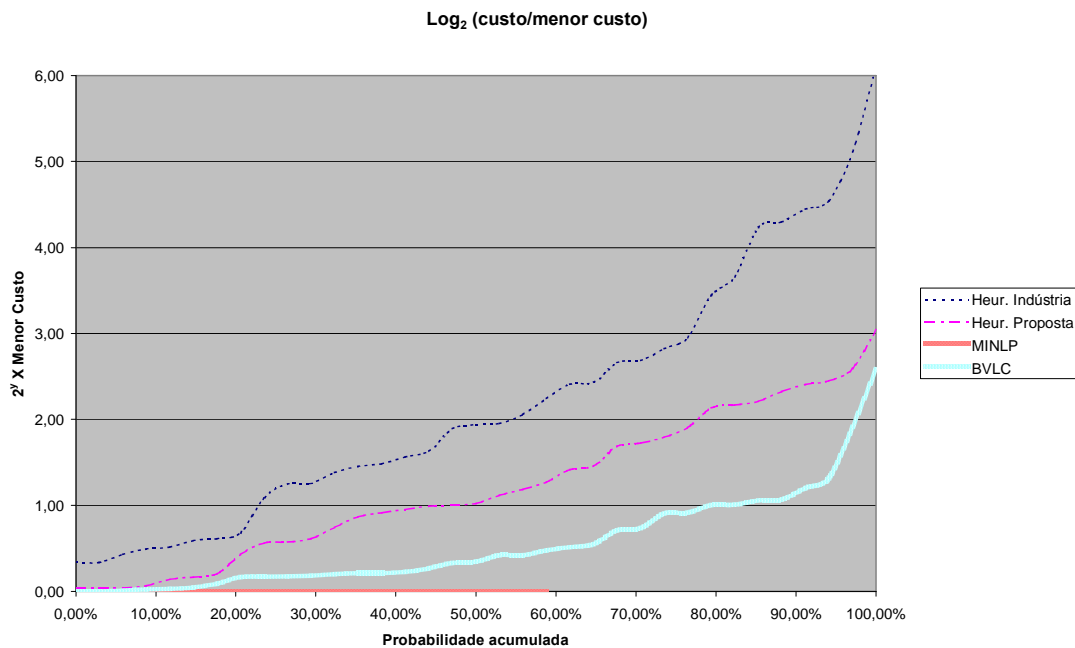
atender a uma exigência da indústria. Além disso, o tempo de processamento depende, evidentemente, das características do processador. Sendo assim, este parâmetro foi utilizado no presente estudo apenas como forma de descartar métodos que tenham incorrido em um tempo de processamento significativamente maior do que aquele disponibilizado pela indústria.

Os valores do parâmetro y (representado na ordenada dos gráficos 1, 2, 3 e 4 – apresentados a seguir) foram calculados, de maneira análoga àquela utilizada por Leyffer (2003), através da seguinte fórmula:

$$y(\text{método}_x, p) = \log_2 \left[\frac{\text{Custo}(\text{método}_x, p)}{\text{Custo}(\text{método}_\text{mais}_\text{eficaz}, p)} \right].$$

A leitura dos referidos gráficos pode ser feita da seguinte maneira: probabilidade acumulada do método, associado à curva ora avaliada, gerar resultado com um custo igual a 2^y vezes o menor custo obtido para um dado problema. Sendo assim, quanto mais uma determinada curva se aproxima do eixo das abscissas, mais eficaz tende a ser o método.

GRÁFICO 01: Avaliação da performance da heurística da indústria, da heurística proposta, do MINLP e do método BVLC

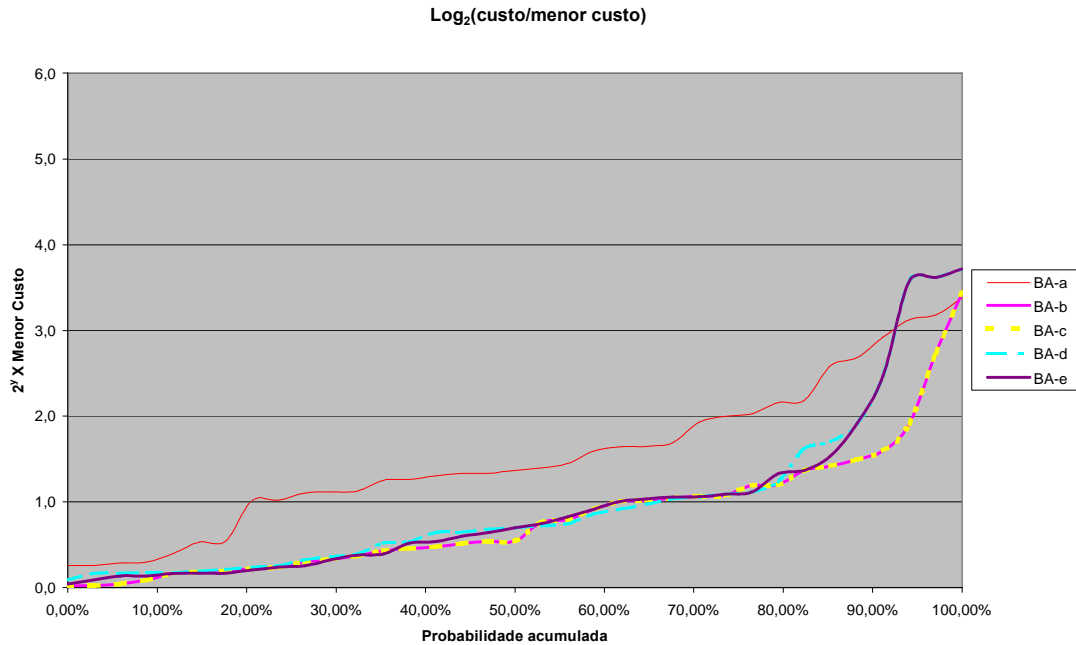


Fonte: Autora

No caso do gráfico 01, a curva que representa a performance do *solver* MINLP foi construída até uma probabilidade acumulada próxima de 60%. A partir desta probabilidade, interrompeu-se a construção da curva por não se ter obtido resultados através deste método.

Para melhor ilustrar o tipo de informação que pode ser extraída a partir do gráfico 01, para a curva associada ao método BVLC, por exemplo, existe uma probabilidade acumulada de 80% do referido método atingir uma performance de até duas vezes a melhor performance, em termos de custo da solução, que pode ser obtida para um dado problema. Para o método heurístico utilizado pela indústria pesquisada, tal medida de performance pode ser de até 2^6 vezes a melhor performance obtida para um dado problema.

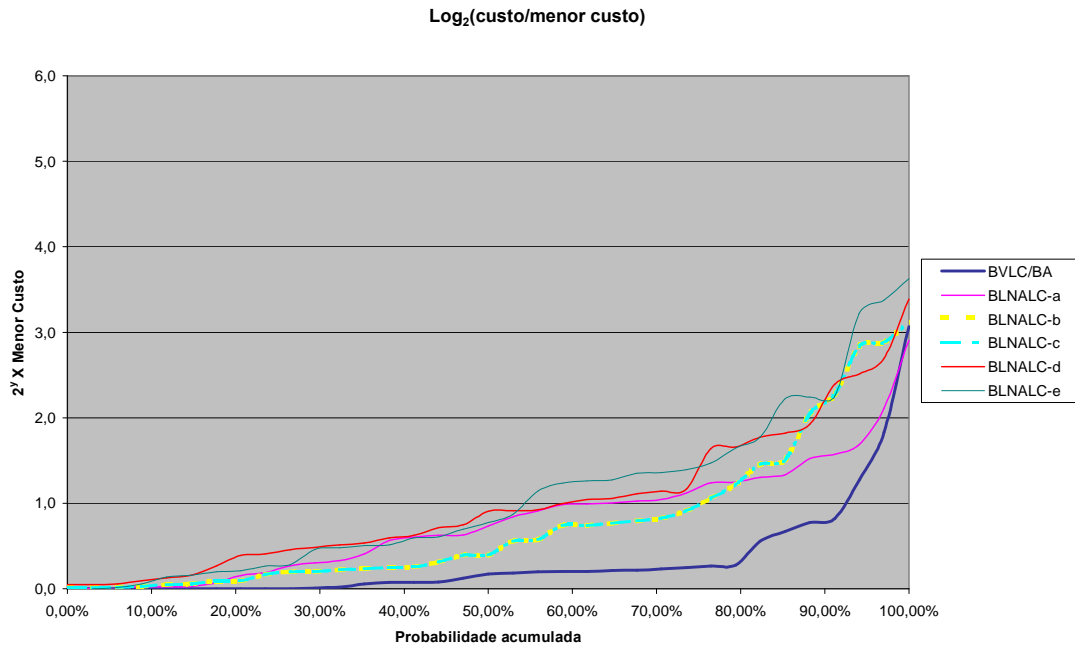
GRÁFICO 02: Avaliação da performance dos métodos de Busca Ávida



Fonte: Autora

Para facilitar a compreensão do gráfico recém apresentado, os métodos de busca ávida podem ter uma performance (em termos de custo da solução) próxima de 2^4 vezes a melhor performance obtida para um dado problema. Isto demonstra que, dentre os processos de busca em grafos avaliados ao longo da pesquisa, o método de busca ávida é o menos eficaz.

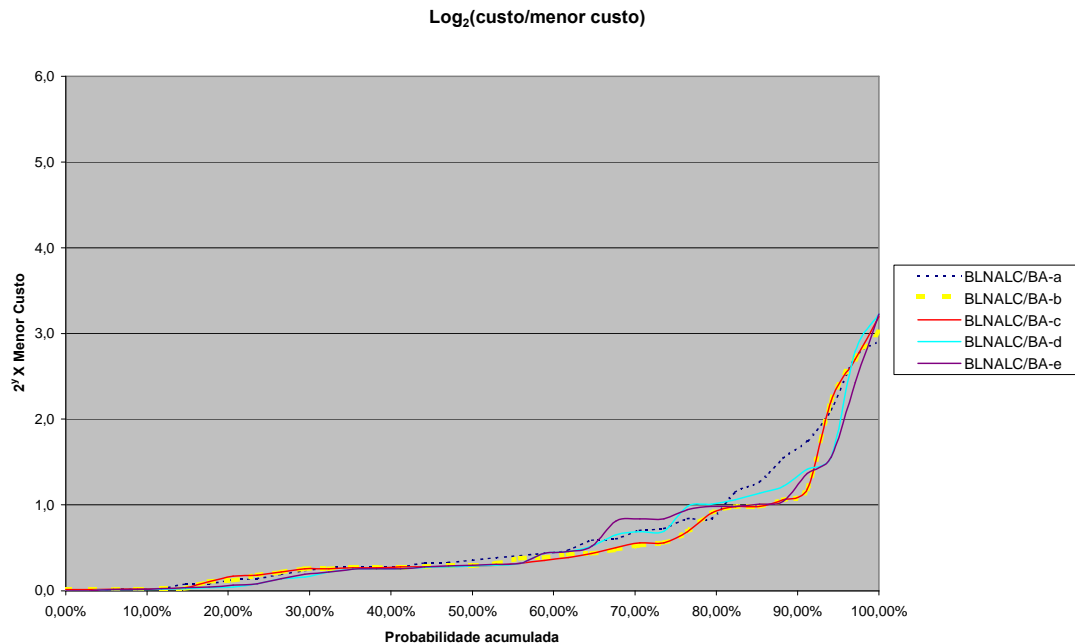
GRÁFICO 03: Avaliação da performance dos métodos BVLC/BA e dos métodos de Busca com Lista de Nós Abertos e Limitação de Custo.



Fonte: Autora

O gráfico 03 indica que o método BVLC/BA, em 80% das situações avaliadas, obteve uma performance muito próxima, em termos de custo da solução, da melhor performance obtida para um dado problema. Dentre os processos de busca em grafos testados ao longo desta pesquisa, o método BVLC/BA foi o que apresentou maior eficácia na resolução do PPOC.

GRÁFICO 04: Avaliação da performance dos métodos de Busca com Lista de Nós Abertos e Limitação de Custo/ Busca Ávida



Fonte: Autora

O gráfico 04 apresenta as curvas de probabilidade acumulada da performance associada ao método que melhor conseguiu agregar eficiência (em termos de tempo de processamento) e eficácia (em termos de custo da solução), dentre os métodos testados ao longo desta pesquisa. As curvas representadas neste gráfico indicam que o processo de busca com lista de nós abertos e limitação de custo/ busca ávida é capaz de proporcionar, em 80% dos casos testados, uma performance de no máximo duas vezes a melhor performance, em termos de custo da solução, que poderia ser obtida para a solução do PPOC.

Considerações adicionais acerca da pesquisa e sugestões de futuros trabalhos ligados ao tema encontram-se apresentadas no próximo capítulo.

7 CONCLUSÕES E RECOMENDAÇÕES

7.1 CONCLUSÕES SOB O ENFOQUE ACADÊMICO

No item 1.6.2. deste trabalho, encontram-se listados os objetivos específicos que, quando plenamente atingidos, marcam o alcance do objetivo geral pretendido com a presente pesquisa.

O objetivo específico (I) consiste na identificação das parcelas de custo associadas ao PPOC. Estas parcelas são discriminadas no item 3.1.2. do capítulo três e formuladas no item 3.3. do mesmo capítulo. A identificação e posterior formulação destas parcelas caracteriza o alcance deste objetivo. Vale ressaltar a importância desta etapa dos estudos como forma de imprimir maior abrangência ao modelo proposto, já que o mesmo leva em conta parcelas de custo que foram desconsideradas em outros modelos concebidos para resolução do problema.

O objetivo específico (II) está associado à identificação das restrições inerentes ao problema em estudo. A identificação e posterior formulação de cada um dos conjuntos de restrições do problema são apresentadas nos itens 3.1.3. e 3.4., respectivamente, marcando o alcance deste objetivo.

De posse da formulação das parcelas de custo e das restrições do problema, modelou-se o mesmo como um problema de programação não linear inteira, conforme apresentado no item 3.5.. Com isso, o objetivo específico (III), que consiste em modelar o PPOC como um problema de programação matemática, foi plenamente atingido.

O objetivo específico (IV) diz respeito à seleção de possíveis técnicas de resolução do problema de minimização, buscando definir uma técnica adequada sob os enfoques “qualidade da solução obtida” e “custo computacional”, medido em termos de tempo de processamento. Com base em ampla revisão bibliográfica apresentada no capítulo quatro, selecionaram-se técnicas que, aparentemente, poderiam contribuir para a resolução do PPOC. A ordenação das técnicas que foram avaliadas ao longo desta pesquisa deu-se em função de sua respectiva facilidade de implementação. Sendo assim, testou-se inicialmente a utilização de um *solver*, num segundo momento a concepção de um método heurístico e, posteriormente, a utilização de um algoritmo genético e da técnica de relaxação lagrangeana, respectivamente. Finalmente, direcionou-se a pesquisa para a utilização de processos de busca em grafos.

A utilização do referido *solver* gerou bons resultados para uma parte dos cenários testados, conforme demonstram os resultados apresentados no capítulo VI. No entanto, não se

conseguiu obter resultado para os cenários nos quais a demanda futura foi igualada a zero. Diante disso, caso a indústria opte por esta alternativa, poderá enfrentar problemas na obtenção de solução para o PPOC em épocas de final de estação.

A concepção de um método heurístico foi fundamental como forma de garantir a obtenção de resultados em um tempo de processamento curto, além de ter servido como base para a concepção do *nó inicial* utilizado nos processos de busca em grafos. Entretanto, não foi suficiente para garantir a obtenção de bons resultados para qualquer tipo de cenário de demanda, por desconsiderar as parcelas de custo que efetivamente incidem no PPOC.

Conforme explicitado nos itens 4.5.3. e 4.5.4., respectivamente, tanto a utilização do algoritmo genético como a da relaxação lagrangeana não se mostraram adequadas para a resolução do problema. Por este motivo, deu-se continuidade à pesquisa através da análise de processos de busca em grafos, obtendo-se a partir daí bons resultados em um tempo julgado adequado. A revisão bibliográfica realizada, bem como os testes efetuados quanto à utilização das técnicas selecionadas previamente, foram imprescindíveis para o alcance do objetivo específico (IV).

O objetivo específico (V) diz respeito à implementação da técnica selecionada na etapa anterior da pesquisa. Com base nos conceitos apreendidos ao longo da revisão bibliográfica, bem como no conhecimento adquirido acerca do PPOC, definiu-se uma configuração de grafo que pudesse representar o processo de busca de uma solução para este problema. Além disso, concebeu-se um conjunto de algoritmos que incluíam procedimentos de busca vertical, ávida, com lista de nós abertos e procedimentos híbridos, visando aliar eficácia (em termos de custo da solução) e eficiência (em termos de tempo de processamento) em um único algoritmo. A inserção de procedimentos heurísticos em tais algoritmos culminou na redução do tempo de processamento envolvido na resolução do problema, compatibilizando este tempo de processamento com aquele disponibilizado pela indústria para esta finalidade. A concepção dos algoritmos, bem como a realização de testes envolvendo numerosas alternativas de procedimentos heurísticos, marcaram o alcance do objetivo específico (V).

O objetivo específico (VI) associa-se à avaliação da consistência do método concebido, através da comparação dos resultados gerados pelo mesmo (com relação a diferentes referências de peças) àqueles que seriam obtidos pela utilização do método adotado atualmente pela indústria pesquisada.

O alcance deste objetivo específico implicou na necessidade de se obterem dados para várias referências de peças. Com o intuito de realizar uma análise de diferentes cenários de demanda – atual e futura, bem como de obter um maior número de situações para as quais se

puddesse avaliar a performance dos diferentes algoritmos considerados, gerou-se um conjunto de 30 configurações adicionais de demanda, além daquelas disponibilizadas pela indústria.

A realização desses testes possibilitou a avaliação da performance dos algoritmos, além de ter demonstrado que o processo de busca em grafos (dentre aqueles testados ao longo da pesquisa) foi o único capaz de gerar boas soluções para o PPOC, mesmo para épocas de final de estação. Esta constatação marcou o alcance do objetivo específico (VI).

Com base no exposto acima, pode-se demonstrar o alcance do objetivo geral da presente pesquisa, apresentado no item 1.6.1. e que consiste na concepção de um método de resolução do PPOC que considere todas as parcelas de custo envolvidas, bem como as limitações inerentes ao problema.

Uma outra questão a ser salientada diz respeito ao ineditismo desta pesquisa, tanto na abordagem do PPOC como na solução proposta. No presente trabalho, são identificadas e inseridas parcelas de custo, inerentes ao problema, que até então não haviam sido publicadas na literatura, nem tampouco consideradas em soluções disponíveis comercialmente. A modelagem do problema como um grafo também é inédita e de extrema relevância no processo de obtenção da solução, destacando-se aí a caracterização dos nós e arcos, bem como a definição do procedimento de expansão.

Ainda quanto ao ineditismo, saliente-se os algoritmos de busca em grafos apresentados no capítulo V. Mesmo que vários deles tenham se originado de processos clássicos de busca em grafos, os algoritmos híbridos propostos nesta pesquisa não são referenciados em publicações sobre o assunto. Além disso, todos os procedimentos heurísticos adotados para estimativa do custo $\hat{h}(n)$ foram concebidos e testados ao longo desta pesquisa. A definição de bons procedimentos foi imprescindível para imprimir maior eficiência (medida em termos de tempo de processamento) na obtenção de uma solução.

7.2 CONCLUSÕES SOB O ENFOQUE DA INDÚSTRIA TÊXTIL

Com base no conhecimento adquirido acerca do PPOC e no entendimento do que ele representa para a indústria de confecção têxtil, verifica-se que a obtenção de boas soluções (em termos de custo) não é suficiente, caso não se respeite a restrição de tempo para resolução do problema. Em outras palavras, a concepção de um método que gere boas soluções de nada adianta, caso as mesmas não sejam geradas no tempo disponibilizado, pela indústria, para esta finalidade.

Apenas revisando o que já havia sido explicitado no primeiro capítulo deste trabalho, somente após a definição das ordens de corte (OC's) para cada uma das referências envolvidas em uma OP (ordem de programação semanal) é que são geradas as necessidades de tecido e de aviamentos para a confecção das peças. Sendo assim, atrasos nesta definição culminam em atrasos na confecção das peças.

Ao longo da pesquisa realizada, conceberam-se métodos de resolução do PPOC capazes de gerar boas soluções em um tempo compatível com aquele disponibilizado para esta finalidade. A tabela a seguir indica o método, dentre aqueles concebidos e testados na pesquisa, mais adequado para ser utilizado pela indústria em função de sua disponibilidade de tempo para definir cada OC. Saliente-se que a mesma foi construída em função dos resultados obtidos para cada método avaliado e na análise comparativa entre os mesmos, apresentada no capítulo anterior.

QUADRO 08: Método recomendado × Tempo disponibilizado pela indústria

TEMPO DISPONIBILIZADO PELA INDÚSTRIA PARA DEFINIÇÃO DE CADA OC	MÉTODO RECOMENDADO
MENOS DE UM MINUTO	Método Heurístico
ATÉ 10 MINUTOS	Método de busca com lista de nós abertos e limitação de custo/ busca ávida (BLNALC/BA)
ATÉ UMA HORA	Método de busca vertical com limitação de custo/ busca ávida (BVLC/BA)
TEMPO IRRESTRITO	Método de busca vertical com limitação de custo (BVLC)

Fonte: Autora

Se a disponibilidade de tempo para definição de cada OC se mantiver nos moldes da disponibilidade verificada atualmente, recomenda-se a utilização do método BLNALC/BA, que conseguiu aliar eficácia (medida em termos de custo da solução) e eficiência (medida em termos de tempo de processamento).

Quanto à importância desta pesquisa no gerenciamento logístico, saliente-se que a obtenção de bons resultados para o PPOC poderá culminar na redução da quantidade de peças que terão de ficar estocadas para atender à demanda futura, bem como na redução de peças excedentes à demanda que, na melhor das hipóteses, serão adquiridas pelo mercado a preços promocionais. Isto quando a indústria, por questões de marketing, não optar por descartar essas peças excedentes ao invés de ofertá-las nas promoções de final de estação.

A redução da quantidade de peças estocadas e/ou vendidas a preços promocionais pode gerar uma redução significativa nos custos de produção da indústria. A título de exemplo, para a indústria tomada como base da pesquisa, a margem de lucro auferida costuma ser inferior ao percentual de desconto embutido nos preços promocionais. Atualmente, em épocas de final de estação (quando a demanda futura tende a zero), os custos de sobra decorrentes das peças excedentes geram prejuízos significativos para esta indústria, que acaba repassando-os para o preço das peças e, com isso, perdendo competitividade em seu nicho de atuação.

Um método capaz de reduzir este custo gera uma redução imediata nos custos de produção, redução esta que tende a se propagar ao longo de toda a cadeia produtiva: menores quantidades produzidas induzem à redução de custos de estoque de peças semi-acabadas e acabadas, menores custos de aquisição de matéria-prima para confecção das mesmas, menores custos de movimentação e transporte, bem como menores custos incorridos na destinação das peças não vendidas.

A redução de custos, sem comprometimento do nível de serviço considerado razoável pelos clientes, vai ao encontro do que preconiza a Logística Empresarial e, num contexto mais amplo, do que define o Gerenciamento da Cadeia de Suprimentos.

7.3 RECOMENDAÇÕES

Devido à sua alta complexidade, o PPOC ainda pode ter sua solução aprimorada com base em pesquisas futuras.

Com relação à modelagem do problema, a parcela de custo de *setup* de estamparia incorrido na mudança de cor de tecido, apesar de já ter sido abordada na pesquisa, não foi inserida na função-objetivo do problema. Portanto, uma sugestão de continuidade dos estudos estaria vinculada à inserção desta parcela no método, visando avaliar a repercussão da mesma nos resultados obtidos.

Durante os testes realizados ao longo da pesquisa, constatou-se que o tempo de processamento envolvido na obtenção de uma solução (quando utilizados algoritmos de busca em grafos) depende da quantidade de peças demandadas, bem como da diferença relativa entre a demanda de peças para cada tamanho. Esta constatação abre espaço para a realização de futuros testes com outros cenários de demanda, bem como com referências de peças associadas a outras indústrias de confecção, além daquela considerada na presente pesquisa.

A associação dos diferentes processos de busca ora apresentados a características das referências, também deverá ser de grande valia às indústrias, que poderão escolher um

processo de busca que mais se adequa às características das referências cujas ordens de corte estão sendo programadas em uma dada semana. Dependendo da quantidade de peças demandadas, poder-se-á também rever e definir a melhor proporção de demanda para a geração do nó inicial no processo de busca, nó este concebido através do método heurístico apresentado no quarto capítulo deste trabalho. Para demandas pequenas, poder-se-á eventualmente partir de um nó inicial com elementos iguais a zero, na tentativa de obter ordens de corte de melhor qualidade.

Neste trabalho, considerou-se que todas as camadas de tecido enfiadas em uma mesa deveriam ter um mesmo comprimento de enfiado. A possibilidade de enfiar camadas de tecido com diferentes comprimentos pode culminar em uma redução ainda maior no custo da sobra de peças e, conseqüentemente, na redução do custo das OC's. Esta hipótese teria de ser investigada em futuros trabalhos e, caso a mesma se verifique, haveria necessidade de adequar a proposta de solução ora efetuada. Visando um aproveitamento maior do comprimento de enfiado nas mesas de corte, poder-se-ia avaliar também a viabilidade de alocar diferentes referências (mas que utilizem o mesmo tecido na confecção de suas respectivas bases principais e/ou contrastes) a uma mesma mesa.

Com base no exposto ao longo deste trabalho, torna-se evidente a complexidade do PPOC sob os enfoques matemático e gerencial, havendo diversas possibilidades de explorá-lo a partir de futuras pesquisas. Acredita-se, contudo, estar contribuindo na disponibilização de instrumentos que podem ser de grande utilidade para as indústrias do segmento de confecção, no que se refere à resolução do problema objeto desta pesquisa.

8 REFERÊNCIAS

ÂNGELO, C. F. **Varejo competitivo**. São Paulo: Atlas, 2001.

ANUÁRIO EXPRESSÃO MAIORES DO SUL. 2005. n. 139. Disponível em:
<http://www.expressao.com.br/300/anuarios_eletronicos/anuario2005/conteudos/textil.htm>.
Acessado em: 27 mar. 2006.

ASSOCIAÇÃO Brasileira da Indústria Têxtil. Setor Têxtil. Disponível em:
<<http://www.abit.org.br/content/area/PublicacaoHTML.asp?nCodAreaConteudo=64&nCodPublicacao=128>>. Acesso em: 04 abr. 2005.

AUDACES. Soluções para a indústria têxtil e de confecções. Disponível em: <<http://www.audaces.com.br/vestuario/>>. Acesso em: 05 abr. 2005.

BALAS, E. An additive algorithm for solver linear programming with zero-one variables. **The journal of Operations Research Society of America**. 1965. p. 517 – 546.

BALLOU, R. H. **Gerenciamento da cadeia de abastecimento**: planejamento, organização e logística empresarial. 4. ed. Porto Alegre: Bookman, 2001.

_____. **Logística empresarial**. São Paulo: Atlas, 1993.

BATTEN, C algorithms for optimal assembly. MIT Computer Science and Artificial Intelligence Laboratory. Mar 2004. Disponível em:<<http://www.mit.edu/~cbatten/work/ssbc04/optassembly-ssbc04.pdf>>. Acesso em: 17 abr. 2006

BEALE, E. M. L. **Integer programming**. in **D. Jacobs (ed.)**. The State of the Art in Numerical Analysis. New York :Academic Press, 1977.

BEASLEY, J. E. Lagrangean Heuristics for location problems. **European Journal of Operational Research**, v. 65, Issue 3, 26 mar. 1993. p. 383-399

BERLINER, H. The B* tree-search algorithm: a best-first proof procedure. **Artificial Intelligence**. vol.12. Issue 1. May 1979. p. 23 – 30.

BOCKMAYR, A.; PISARUK, N. Solving assembly line balancing problems by combining IP and CP. **Annals of sixth annual workshop of the ERCIM Working Group on Constraints**, Prague, jun 2001.

BURKE, E. K.; KENDALL, G.; WHITWELL, G. A New Placement Heuristic for the Orthogonal Stock Cutting Problem. **Journal of Operations Research** , 2002.

CEAC: Corte e Empacotamento Assistidos por Computador. 1997Disponível em: <<http://www.lac.inpe.br/po/projects/ceac/ceac.html>>. Acessado em: 03 jun. 2005.

CHAKRABARTI, S.; et. al. Heuristic search in restricted memory. **Artificial Intelligence**. vol.41, Issue 2. dec. 1989. p 197 – 221.

CHRISTOFIDES, N. Graph Teory: Na algorithmic aproach. Academic Press. London: 1975.

CHRISTOFIDES, N; BEASLEY, J. E. Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. **European Journal of Operational Research**, vol 12, Issue 1, Jan. 1983. p. 19-28.

CHRISTOPHER, M. **Logística e gerenciamento da cadeia de suprimentos**. São Paulo: Editora Pioneira, 2002.

CORMEN, T. H. et. Al. Introdution to algorithms. 2 ed. MIT Press and McGraw-Hill, 2000

CUNHA, D. C. da. Aplicação dos resultados da aplicação de postponement em uma grande malharia e confecção de Santa Catarina. 2002. 173 f. **Dissertação...** (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis: UFSC, 2002.

CUNHA, I. J. **A indústria catarinense rumo ao novo milênio**: desafios, evolução e oportunidades. Florianópolis: FIESC/ SEBRAE-SC, 1996.

CUNHA, R. R. M. Um algoritmo de minimização de sobras em corte unidimensional. 1998. 91f. **Dissertação...** (Mestrado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis: UFSC, 1998.

DANTZIG, G. B. **Linear programming and extensions**. Princeton: N. J.: Princeton University Press, 1963.

DEPARTAMENTO Intersindical de Estatística e Estudos Sócio-econômicos. Disponível em: <<http://www.dieese.org.br/anu/2001/anu2001-2.xml>>. Acesso em 10 abr. 2005

DIJKSTRA, E. W. **A note on two problems in connection with graphs**. Numerische Mathematik. 1. 1959. p. 269.

DOLAN, E. D. e MORÉ, J. Benchmarking optimization software with cops. Technical Report MCS-TM-246. **Mathematics and Computer Science Division**, Argonne National Laboratory, Argonne, IL, nov. 2000.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, n. 44, 1990. p. 145-159.

ESPEJO, L. G. A.; GALVÃO, R. D. O uso das relaxações lagrangeana e surrogate em problemas de programação inteira. **Pesqui. Oper.** vol.22, n.3 Rio de Janeiro, july/dec, 2002.

FEOFILLOF, P. Exercícios de teoria dos grafos. Departamento de Ciências da Computação. IME/USP out. 2005. Disponível em: <<http://www.ime.usp.br/~pf/grafos-exercicios/>> Acesso em: 04 abr. 2006

FIESC; Banco Regional de Desenvolvimento do Extremo Sul. Desempenho e Perspectivas da Indústria Catarinense. Florianópolis: FIESC 2005.

FIGUEIREDO, J. NEIVA DE; GONZAGA, C. Aplicação de métodos de busca em grafos com nós parcialmente ordenados à locação de torres de transmissão. **Revista Pesquisa Operacional**. Rio de Janeiro, vol. 23 n.1 jan./apr. 2003.

FIGUEIREDO, J. Neiva de; AGONZAGA, C; MACULAN, N. Localização ótima das torres de transmissão utilizando teoria de grafos. **Revista Brasileira de Pesquisa Operacional**, II (2). 12 – 22. 1979

FISHER, M.L. The lagrangian relaxation method for solving integer programming problems. **Management Science**, n. 27, 1981. p.1-18.

FLEURY, P. F.; WANKE, P.; FIGUEIREDO, K. F. **Logística empresarial: a perspectiva brasileira**. São Paulo: Editora Atlas, 2000.

FOURER, R.; GAY, D. M.; KERNIGHAN, B. W. **AMPL: A modeling language for mathematical programming**. Danvers, Massachusetts: Boyd & fraser publishing company, 1993.

GALVÃO, R.D.; RAGGI, L.A. A method for solving to optimality uncapacitated location problems. 1989 **Annals of Operations Research**, vol. 18, p. 225 – 244.

GARCÍA-LÓPEZ, F. et al. Parallel Computing. 29: 575-589,. Parallelization of the Scatter Search. 2003.

GEOFFRION, A.M.; GRAVES, G; LEE, L., Strategic distribution system planning: a status report. **Studies in Operations Management**, A. Hax (Ed.), North Holland/American Elsevier, 1978. p. 179-204

GILMORE, P. C.; GOMORY, R. E. A linear approach to the cutting-stock problem, **Operations Research**, vol 9, 1961. p. 849-859.

_____ ; _____. A linear approach to the cutting-stock problem: part II, **Operations Research**, vol. 11, 1963. p. 863-888.

_____ ; _____. Multistage cutting stock problems of two and more dimensions, **Operations Research**, vol. 13, 1965. p. 94-120.

GOMORY, R. E. **An algorithm for integer solutions to linear programs**: recent advances in mathematical programming. New York: McGraw-Hill Book Company, 1963.

GOMORY, R. E. **An algorithm for the mixed integer problem**. Rand Corporation, Santa Monica, Cal., 1960.

GORINI, A. P. F. **Panorama do setor têxtil no Brasil e no mundo**: reestruturação e perspectivas. Rio de Janeiro: BNDES Setorial, n. 12, set. 2000. p. 17-50.

GOULARTI FILHO, A.; GENOVEVA NETO, R. **A indústria do vestuário**: economia, estética e tecnologia. Florianópolis: Livraria e Editora Obra Jurídica Ltda, 1997.

HARRISON, A.; VAN HOEK, R. **Estratégia e gerenciamento da logística**. São Paulo: Editora Futura, 2003.

HELD, M.; KARP, R.M. The travelling salesman problem and minimum spanning trees: part II. **Mathematical Programming**, 1, 1971. p. 6-25.

HELMAN, Paul; MORET, Bernard M. E.; SHAPIR, Henry D.. An exact characterization of greedy structures. *SIAM J. Discrete Math*, 1993, p. 274-283.

HOLLAND, J. H. **Adaptation in natural and artificial** systems. Ann Arbor: University of Michigan, 1975.

HOPPER, E.; TURTON, B. C. An empirical investigation of meta-heuristics and heuristics Algorithms for 2D Packing Problem, **European Journal of Operations Research**, vol. 128/1., 2001. p. 34, 57.

INSTITUTO EUVALDO LODI. Análise da eficiência econômica e da competitividade da cadeia têxtil brasileira: IEL, CNA e SEBRAE. Brasília: IEL, 2000. 480p.

IVANQUI, I. L. Um modelo para solução de arranjo físico de instalações interligadas por corredores. 1997. **Tese...** (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis: UFSC 1997.

JACKOBS-BLECHA, C. System Modeling for Apparel Manufacturing: Focus on the Cutting Room. National textile center annual report. Disponível em: <<http://64.233.161.104/search?q=cache:c0Ns4NRDPGIJ:www.ntcresearch.org/pdf-rpts/AnRp93/A93G93N1.pdf+cutting+scheduling+textile&hl=pt-BR>>. Acesso em: 06 maio 2005.

KAINDI, H; KHORSAND. A memory-bounded bidirectional search. In Proc. **Of the Twelfth International Conference on Artificial Intelligence**. 1994. p. 1359-1364

KATSURAYAMA, D. M. Problemas de corte. Disponível em: <<http://www.lac.inpe.br/~massaru/problema.htm>>. Acesso em: 01 jan. 2005.

KELLY, K.; LABUTE, P. The A* search and Applications to Sequence Alignment. Chemical Computing Group Inc. Canada, fev. 1996. Disponível em: <<http://www.chemcomp.com/journal/astar.htm>>.. Acessado em 17 abr. 2006.

KOZA, J. R., RICE, J. P. **Genetic Programming: The Movie**. Cambridge, MA: The MIT Press, 1992.

LAMBERT, D. M.; STOCK, J. R.; ELLRAM, L. M. **Fundamentals of logistics management**. Boston: Irwin McGrawHill, 1998.

LAND, A. H. DOIG, A. G. An automatic method for solving discrete programming Problems. **Econometrica**, vol. 28. 1960. p. 497-520.

LEYFFER, S. Complementarity constrains as nonlinear equations: theory and numerical experience. **Mathematics and Computer Science Division**. Argonne National Laboratory, Argonne, IL 60439, USA. 2003.

LORENA, L. A. N.; NARCISO, M. G. Using local surrogate information in Lagrangean relaxation: An application to symmetric traveling salesman problem. **European Journal of Operational Research**. v. 138, maio 2002. p. 473 – 483.

MACHADO, L.D. O planejamento de recursos de manufatura na cadeia produtiva têxtil. 2004. 177 f. **Dissertação...** (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis: UFSC, 2004.

MACULAN FILHO, Nelson. **Programação linear inteira**. Rio de Janeiro: COPPE/ UFRJ, 1978.

MATHEMATICAL Program Glossary. Disponível em: <<http://carbon.cudenver.edu/~hgreenbe/glossary/index.php?page=C.html>>. Acesso em: 29 abr. 2005.

MAZZIOTTI, B. W., HORN, R. E. Creating a flexible, simulated-based finite scheduling tool. Textile/ Clothing Technology Corp. USA. Disponível em: <<http://64.233.161.104/search?q=cache:ZT1F17C7CfkJ:www.informs-cs.org/wsc97papers/0853.PDF+cutting%2Bscheduling%2Btextile&hl=pt-BR>>. Acesso em: 04 maio, 2005.

MCCARL, B.; SPREEN, T. H. CHAPTER XVI: Integer Programming Formulations. 1997. Disponível em: <<http://agecon2.tamu.edu/people/faculty/mccarl-bruce/mccspr/new16.pdf>>. Acesso em: 20 abr. 2005.

MIRANDA, M. N. Algoritmos genéticos: fundamentos e aplicações. GTA/UFRJ. Disponível em: <<http://www.gta.ufrj.br/~marcio/genetic.html#Introdução>>. Acesso em: 04 maio, 2005.

NILSSON, N. J. Problem-solving methods in ARTIFICIAL INTELLIGENCE. McGraw-Hill. New York: 1971

NOVAES, A.G. **Logística e gerenciamento da cadeia de distribuição**. São Paulo: Editora Campus, 2001.

OLIVEIRA, N. V. Problema de Cobertura de Conjuntos: uma comparação numérica de algoritmos heurísticos. 1999. **Dissertação...** (Mestrado em Engenharia de Produção.) – Universidade Federal de Santa Catarina, Florianópolis: UFSC. 1999.

OPTIMAL SPEKTRUM: O Portal Brasileiro de Otimização. Corte e Empacotamento & Correlatos. Disponível em: <<http://www.mat.uel.br/spektrum/corte.htm>>. Acesso em: 02 maio 2005.

OPTIPLAN. Disponível em: <http://www.lectra.com/en/fashion_apparel/products/otiplan.html>. Acesso em: 11 abr. 2005.

PARADIMITRIOU, C. H.; STEIGLITZ, K. Combinatorial optimization: algorithms and complexity. New York: Dover Publications. 1998

PARDALOS, P. M.; RESENDE, M. G. C. **Handbook of APPLIED OPTIMIZATION**. Oxford University Press, 2002.

PARKER, R.G.; RADIN, R.L. **Discrete optimization**. New York: Academic Press, 1988.

PRICE, W.L. **GRAPHS AND NETWORKS: An Introduction**. Butterworths. London. 1971.

RUBIO, A. L. **A cadeia de suprimento interna**. São Paulo: Editora STS, 2002.

SALKIN, H. **Integer programming**. Addison-Wesley: Reading, Mass, 1975.

SEBRAE-SP&IPT. **O desenho das micro e pequenas empresas no setor têxtil-confecção**. São Paulo: SEBRAE-SP & IPT, 2001.

SHINGO, S. **O Sistema Toyota de produção do ponto de vista da engenharia de produção**. Porto Alegre: Editora Bookman, 1996.

SIMONNARD, M. **Programmation linéaire: technique du calcul économique**. 2. ed. Paris: Editora Dunod, 1973.

STAGEMMEIER, A. T.; CLARCK, A. R. A survey of lot-sizing and scheduling models. **Anais...** Annual Symposium of the Brazilian Operational Research Society. 23. Campos do Jordão, 2001.

TEXTILIANET. Disponível em: <http://www.textilia.net/>. Acesso em: 27 abr. 2005.
VALENZUELA, C. L. WANG, P. Y. **Heuristics for large Strip packing problems with Guillotine Patterns: an empirical study, Proceedings of the 4th Metaheuristics International Conference**, 2001. p. 417-421.

WHITWELL, G. Novel Heuristic and Metaheuristic Approaches to Cutting and Packing. Tese de Doutorado. Universidade de Nottingham. Setembro, 2004. Disponível em: <http://www.cs.nott.ac.uk/~gxk/papers/gxwPhDthesis.pdf+whitwell+2004+cutting+and+packing+vehicle+loading&hl=pt-BR>. Acesso em: 28 abr. 2005.

WONG, W. K. A fuzzy Capacity-Allocation Model for Computerized Fabric-Cut Systems. 2003. **The International Journal of Advanced Manufacturing Technology**. Publisher: Springer London. vol. 21, n. 9. June 2003. p. 699 – 711.

WONG, W. K.; CHAN, C. K.; IP, W. H. Optimization of Spreading and Cutting sequencing model in garment manufacturing. *Computers in Industry*. **Elsevier Science Publishers B. V.** vol. 43 Issue 1 aug. 2000. p. 1 - 10.

WONG, W. K.; CHAN, C. K.; IP, W. H.; HUANG, M. Evaluation of optimum combinations of Spreading and Cutting Machines in a Garment Factory. **The International Journal of Advanced Manufacturing Technology**. Publisher: Springer London.: vol.18, n. 1. July 2001. p 62 - 66

ZHOU, R.; HANSEN, E. A. Breadth-first heuristic search. **Artificial Intelligence**. Vol. 170, Issues 4 – 5. Apr. 2006. p. 385 – 408.

_____. A memory-bounded graph search. **International FLAIRS CONFERENCE**. 15. Pesecola, Florida, may 2002.

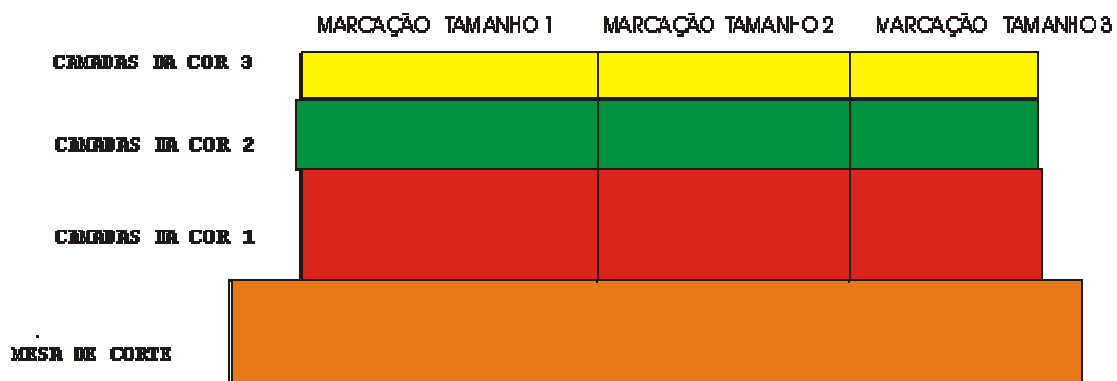
ZIONTS, S. **Linear and integer programming**. New Jersey: Prentice-Hall, 1974.

APÊNDICES

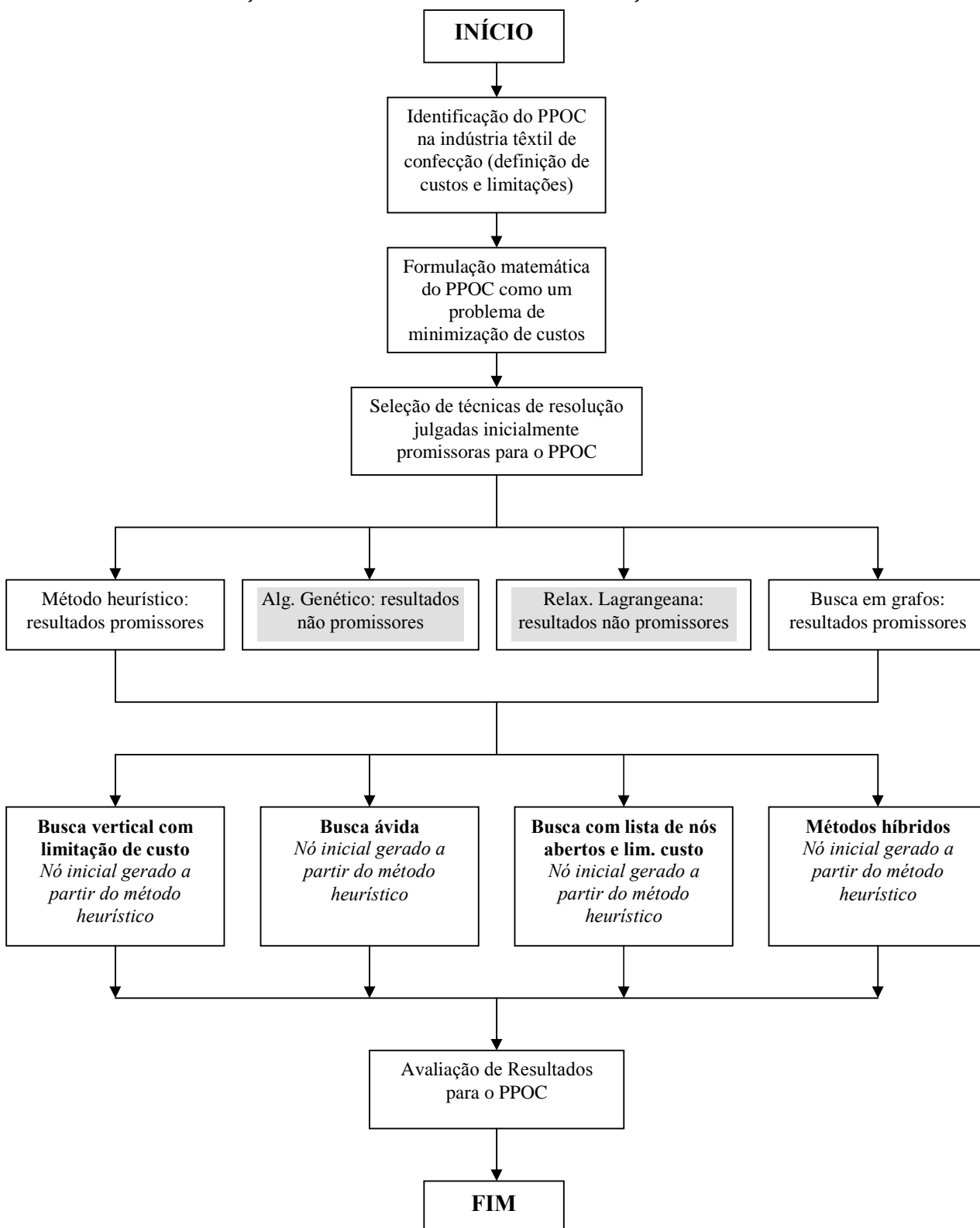
APÊNDICE 01: REPRESENTAÇÃO DE UMA MESA DE CORTE SOBRE A QUAL ESTÃO ENFESTADAS MARCAÇÕES/ CAMADAS DE TECIDO DE DIFERENTES CORES – VISTA FRONTAL

Características da Referência:

Três tamanhos: Tamanhos 1, 2 e 3;
Três cores: vermelha, verde e amarela.



**APÊNDICE 02: FLUXOGRAMA REPRESENTATIVO DAS ETAPAS DE
OBTENÇÃO DE UM MÉTODO DE RESOLUÇÃO PARA O PPOC**



APÊNDICE 03: ILUSTRAÇÃO DO PROCESSO DE EXPANSÃO DE UM NÓ

Dados de entrada:

Referência com dois tamanhos e duas cores;

Demanda atual = Demanda futura:

	COR 1	COR 2
TAM 1	24 peças	39 peças
TAM 2	36 peças	58 peças

* Os demais dados da referência seguem aqueles definidos para a referência 1 (vide Tabela 02).

1^a. iteração: **Nó inicial obtido com o método heurístico:** [17 24 1 1 0 0 0 0]

Nó gerador: [17 24 1 1 0 0 0 0]

Onde:

- Primeiro elemento do nó: número de marcações do tamanho 1 na mesa 1;
- Segundo elemento do nó: número de marcações do tamanho 2 na mesa 1;
- Terceiro elemento do nó: número de camadas da cor 1 na mesa 1;
- Quarto elemento do nó: número de camadas da cor 2 na mesa 1;
- Quinto elemento do nó: número de marcações do tamanho 1 na mesa 2;
- Sexto elemento do nó: número de marcações do tamanho 2 na mesa 2;
- Sétimo elemento do nó: número de camadas da cor 1 na mesa 2;
- Oitavo elemento do nó: número de camadas da cor 2 na mesa 2.

Número de peças geradas a partir da OC representada pelo nó gerador:

	COR 1	COR 2
TAM 1	17 peças	17 peças
TAM 2	24 peças	24 peças

Matriz de inviabilidades para o nó gerador

(considerando PERCmin = 0,9):

	COR 1	COR 2
TAM 1	-4,6 peças	-18,1 peças
TAM 2	-8,4 peças	-28,2 peças

Par escolhido para redução de déficit:

par TAM 2 x COR 2 (par mais deficitário).

Nós sucessores:

Sucessor 1 = [17 **25** 1 1 0 0 0 0]

Sucessor 2 = [17 24 1 **2** 0 0 0 0]

Sucessor 3 = [17 **25** 1 **2** 0 0 0 0]

Sucessor 4 = [17 24 1 1 0 **1 0 1]**

2ª. iteração: **Nó gerador:** [17 25 1 1 0 0 0 0] (*antigo Sucessor 1*)

Número de peças geradas a partir da OC representada pelo nó gerador:

	COR 1	COR 2
TAM 1	17 peças	17 peças
TAM 2	25 peças	25 peças

Matriz de inviabilidades para o nó gerador:

(considerando PERmin = 0,9)

	COR 1	COR 2
TAM 1	-4,6 peças	-18,1 peças
TAM 2	-7,4 peças	-27,2 peças

Par escolhido para redução de déficit:

par TAM 2 x COR 2 (par mais deficitário).

Nós sucessores:

Sucessor 1 = [17 **26** 1 1 0 0 0 0]

Sucessor 2 = [17 25 1 **2** 0 0 0 0]

Sucessor 3 = [17 **26** 1 **2** 0 0 0 0]

Sucessor 4 = [17 25 1 1 0 **1** 0 **1**]

** Este processo continua até a obtenção da solução para o PPOC através de um dos processos de busca concebidos durante a pesquisa.

APÊNDICE 04: LISTAGENS DOS PROGRAMAS EM MATLAB

PROGRAMA DE IMPLEMENTAÇÃO DO MÉTODO HEURÍSTICO PROPOSTO

```

function ind=NewInd_inicial(DadosRef)
% Função que retorna um individuo inicial...eval(DadosRef);
DemNaoAtendida=DEMANDA;
%inicializacao da matriz de demanda nao atendida = demanda total ind=
ones(1,Kmax*(NumCor+NumTam));
OCs=1;
while ((sum(sum(DemNaoAtendida~=0))) )
% enquanto ainda houver demanda para ser atendida
DemTotal=sum(DemNaoAtendida,2);
%DemandaTotal para cada um dos tamanhos da referencia Comprimento=DemTotal'*COP;
% Comprimento total de malha necessario para suprir um tamanho
Camadas=Comprimento/COEmax;
%Numero de camadas aproximado para atender um tamanho
goon=1;
%Calculo do numero medio de pecas que cabem em uma mesa
COPmedio=sum(COP)/NumTam;
NumMedioPecas=(floor(COEmax/COPmedio))*NCAMax;
while goon;
%as marcacoes sao iguais a DemTotal/Camadas e arredondadas pela funcao
ArredAleatorio
if (sum(sum(DemNaoAtendida))>=1.1*NumMedioPecas),
Marc=ArredAleatorio(DemTotal/Camadas,1);
elseif (sum(sum(DemNaoAtendida))>=0.9*NumMedioPecas),
Marc=ArredAleatorio(DemTotal/Camadas,1/5);
else,
Marc=ArredAleatorio(DemTotal/Camadas,1/2);
end; %do if
%se o comprimento total for maior que COEmax, goon = 1 e as marcacoes
% sao recalculadas
goon=(Marc'*COP > COEmax);
end; %do while goon

%inicio do processo de alocao de camadas - Camada eh uma matriz de
%NumTam x NumCor elementos
Camada=zeros(size(DemNaoAtendida));
for i=1:NumCor,
%o numero de camadas para cada marcacao>0 eh igual a demanda nao
%atendida desta marcacao para a cor i dividido pela propria
%marcacao
Camada(Marc>0,i)=DemNaoAtendida(Marc>0,i)./(Marc(Marc>0));
end; %do for i
Atende=2*ones(size(Camada));
%para as camadas diferentes de 0, o param. atende eh a relacao entre o
%numero destas camadas e o seu respectivo arredondamento para o inteiro
%mais proximo

```

```

Atende(Camada~=0)=Camada(Camada~=0)./max(1,round(Camada(Camada~=0)));
%determinacao da linha (tamanho) associada ao valor maximo da soma do
%param. Atende (soma de todas as colunas), cujo valor seja menor ou
%igual a 1
linha=find(sum(Atende<=1,2)==max(sum(Atende<=1,2)));
%se houver mais de uma linha com essas caracteristicas...
if length(linha)>1
    %Camada1 sera igual ao vetor formado pelo numero maximo de camadas,
    %para cada cor, entre os dois ou mais tamanhos seleccionados a partir do vetor
    %linha
    Camada1=ceil(max(Camada(linha,:)));
else
    %Camada1 sera o numero de camadas para cada cor do tamanho 'linha',
    %numero este arredondado para cima
    Camada1=ceil(Camada(linha,:));
end; %do if length

while sum(Camada1)>NCAMax,
    %se o somatorio destas camadas for maior que o NCAMax, faz-se uma
    %ponderacao
    Camada1=round(Camada1/sum(Camada1)*NCAMax);
end; %do while sum

goon=1;
while goon
    %definicao do individuo
    ind((OCs-1)*(NumTam+NumCor)+1:OCs*(NumTam+NumCor))=[Marc' Camada1];
    %verificacao da parcela da demanda que foi atendida
    P=producao(ind,DadosRef);
    DemNaoAtendida=DEMANDA*PERCmin-P;
    goon=(sum(Camada1)<NCAMax) & (max(find(DemNaoAtendida>0)));
    if goon,
        %procura da cor cuja demanda nao estah sendo atendida
        [c,m]=max(sum(DemNaoAtendida>0));
        Camada1(m)=Camada1(m)+1;
    end;
end;
%inserindo mais uma OC
OCs=OCs+1;
%verificando quais os tamanhos/cores jah atendidos
Atendeu=find(DemNaoAtendida<=0);
%zerando o elemento da matriz DemandaNaoAtendida dos tamanhos/cores jah
%atendidos
DemNaoAtendida(Atendeu)=zeros(size(Atendeu));
end %do while DemandaNaoAtendida

%tentativa de melhorar individuos com algumas pequenas modificacoes
%primeiro bloco - sobras evidentes nas marcacoes
[a,b,c,d,sobra]=Ehviavel(ind,DadosRef);
for i=1:NumTam,

```

```

for OC=1:OCs,
    if (sobra(i,:)-ind((OC-1)*(NumTam+NumCor)+NumTam+1:(OC-
1)*(NumTam+NumCor)+NumTam+NumCor)>zeros(1,NumCor)),
        if (ind((OC-1)*(NumTam+NumCor)+i)>0),
            ind((OC-1)*(NumTam+NumCor)+i)=ind((OC-1)*(NumTam+NumCor)+i)-1;
            [a,b,c,d,sobra]=Ehviavel(ind,DadosRef);
        end; %do if
    end; %do if
end; %do for OC
end; %do for i

%segundo bloco - sobras evidentes nas camadas
[a,b,c,d,sobra]=Ehviavel(ind,DadosRef);
for j=1:NumCor,
    for OC=1:OCs,
        if (sobra(:,j)-ind((OC-1)*(NumTam+NumCor)+1:(OC-
1)*(NumTam+NumCor)+NumTam)>zeros(1,NumTam)),
            if (ind((OC-1)*(NumTam+NumCor)+NumTam+j)>0),
                ind((OC-1)*(NumTam+NumCor)+NumTam+j)=ind((OC-
1)*(NumTam+NumCor)+NumTam+j)-1;
                [a,b,c,d,sobra]=Ehviavel(ind,DadosRef);
            end; %do if
        end; %do if
    end; %do for OC
end; %do for j
%heurística que tenta melhorar o individuo utilizando a funcao Melhora_ind
Novo_fitness=Calcula_fitness(ind,DadosRef);
Fitness_base=Novo_fitness+1;
while Novo_fitness<Fitness_base,
    Fitness_base=Novo_fitness;
    [a,b,c,d,e]=Ehviavel(ind,DadosRef);
    %e eh a matriz de sobras de peças
    [ind,Novo_fitness]=Melhora_individuo_posqualif(ind,DadosRef,e,Fitness_base);
end; %do while

```

PROGRAMA DE IMPLEMENTAÇÃO DA HEURÍSTICA DA INDÚSTRIA TÊXTIL

```

function OC_Marisol=Alg_heurMarisol(DadosRef);
%funcao que simula a heuristica da Marisol para obtencao de uma OC para a
%referencia - dados informados pelo Adriano da Marisol
eval(DadosRef);
Num_pecas_porcamada=round(COEmax/(mean(COP)));
Num_maximo_pecas=Num_pecas_porcamada*NCAMax;
Demanda_ao_atendida=DEMANDA;
OC=1;
while sum(sum(Demanda_ao_atendida))~=0,
    DEMANDA_TAMANHO=sum(Demanda_ao_atendida,2);
    %estimativa do numero de marcacoes por tamanho

    Num_pecas=round((DEMANDA_TAMANHO/sum(sum(Demanda_ao_atendida)))*Num_p
    ecas_porcamada);
    TOTAL_CAMADA=Num_pecas*COP;
    Tamanho=1;
    while TOTAL_CAMADA>COEmax,
        if (TOTAL_CAMADA-COP(Tamanho))<=COEmax,
            Num_pecas(Tamanho)=Num_pecas(Tamanho)-1;
            Tamanho=1;
        else,
            Tamanho=min(NumTam,(Tamanho+1));
        end; %do if
        if Tamanho==NumTam,
            Num_pecas(Tamanho)=Num_pecas(Tamanho)-1;
            Tamanho=1;
        end; %do if Tamanho
        TOTAL_CAMADA=Num_pecas*COP;
    end; %do while
    % verificacao de quantas cores podem ser agregadas em uma mesa
    DEMANDA_COR=sum(Demanda_ao_atendida,1);
    Cores_juntas_inicio=0;
    Primeira_cor=1;
    while Cores_juntas_inicio==0,
        if DEMANDA_COR(Primeira_cor)>0,
            Cores_juntas_inicio=Primeira_cor;
        else,
            Primeira_cor=Primeira_cor+1;
        end; %if DEMANDA_COR
    end; %do while
    Cores_juntas_fim=Cores_juntas_inicio;
    Goon=1;
    if Cores_juntas_fim==NumCor,
        Goon=0;
    end; %do if Cores_juntas_fim
    while Goon, %Verificando quantas cores poderao entrar em uma mesma mesa

```



```

SUM_DEM_PROV=DEMANDA_COR(Cores_juntas_inicio:min(NumCor,(Cores_juntas_fi
m+1)));
if max(SUM_DEM_PROV)<=(Num_maximo_pecas/(Cores_juntas_fim+1)),
    Cores_juntas_fim=Cores_juntas_fim+1;
    if Cores_juntas_fim==NumCor,
        Goon=0;
    end; %do if Cores_juntas_fim
else,
    Goon=0;
end; %do if
end; %do while
%estimativa do numero de camadas por cor
Numero_camadas=0;
for cor=Cores_juntas_inicio:Cores_juntas_fim,
    Num_camadas(cor)=0;
    for tam=1:NumTam,
        if ceil(Demanda_nao_atendida(tam,cor)/Num_pecas(tam))>Num_camadas(cor),
            Num_camadas(cor)=ceil(Demanda_nao_atendida(tam,cor)/Num_pecas(tam));
        end; %do if Demanda_nao_atendida
    end; %do for tam
    Num_camadas(cor)=min((NCAmax-Numero_camadas),Num_camadas(cor));
    Numero_camadas=Numero_camadas+Num_camadas(cor);
end; %do for cor
if Cores_juntas_fim<NumCor,
for cor=(Cores_juntas_fim+1):NumCor,
    Num_camadas(cor)=0;
end; %do for cor
end; %do if Cores_juntas_fim
if Cores_juntas_inicio>1,
for cor=1:(Cores_juntas_inicio-1),
    Num_camadas(cor)=0;
end; %do for cor
end; %do if
%criacao de uma mesa
OC_Marisol((OC-1)*(NumTam+NumCor)+1:OC*(NumTam+NumCor))=[Num_pecas'
Num_camadas];
%Atualizacao da demanda nao atendida
for tam=1:NumTam,
    for cor=1:NumCor,
        Demanda_nao_atendida(tam,cor)=max(0,Demanda_nao_atendida(tam,cor)-
Num_camadas(cor)*Num_pecas(tam));
    end; %do for cor
end; %do for tam
OC=OC+1;
end; %do while sum(sum(Demanda_nao_atendida

```

BUSCA VERTICAL – PROGRAMA PRINCIPAL

% Este programa chama os processos de busca vertical com limitação de custo ou o processo de busca ávida, dependendo da opção do usuário

```
function novound=programa_principal_recursivo(ind,DadosRef,Custo_min,nivel);
%esta funcao viabiliza solucoes obtidas a partir
%da relaxacao da restricao de demanda. Baseada em um processo de busca vertical com
exame retroativo,
%onde examina todas as solucoes viaveis possiveis cujo custo seja menor que
%o minimo custo ateh o momento...
eval(DadosRef);
global funcao;
if nargin==2, %na primeira iteracao alguns parametros sao inicializados
    Custo_min=1000000;
    nivel=-1; %nivel eh o numero de vezes em que a funcao viabiliza_ind_new eh chamada
    funcao=upper(input('Entre com o metodo a usar:', 's'));
end;
nivel=nivel+1;
fprintf(['\n Custo Entrada %d - %d : %8.4f'],nivel,Custo_min);
[a,b,c,d,e]=Ehviavel(ind,DadosRef);
disp(a)
disp(ind)
if a==0,
    novound=ind;
else,
    [inds,N_ind]=GERA_SUCESSORES(ind,DadosRef); %gerando populacao inds e trazendo
o numero de individuos nesta populacao
    % Viabilizando a Populacao;
    switch funcao
        case 'A', %BUSCA VERTICAL RETROATIVA
            fitness_base=Calcula_Fitness(ind,DadosRef);
            [a,b,c,d,e]=Ehviavel(ind,DadosRef);
            e=e<0;
            deficit_base=sum(sum(e));
            inds_pais=Define_pais(inds,DadosRef,N_ind,Custo_min,fitness_base,deficit_base);
        case 'B' %BUSCA VERTICAL PURA - CONSIDERANDO COMO CRITERIO DE
SELECAO DO PAI O INDIVIDUO MENOS INVIAVEL
            Deficit_1=1000000;

inds_pais=Seleciona_um_pai_norma_deficit(inds,DadosRef,N_ind,Custo_min,Deficit_1);
    inds_pais={inds_pais};
        case 'C', %BUSCA VERTICAL PURA - CONSIDERANDO COMO CRITERIO DE
SELECAO DO PAI O INDIVIDUO COM MENOR CUSTO MARGINAL PARA
REDUCAO DE INVIABILIDADE
            Fitness_base=Calcula_Fitness(ind,DadosRef);
            [a,b,c,d,e]=Ehviavel(ind,DadosRef);
            Matriz_deficits=e<0;
            Deficit_base=sum(sum(Matriz_deficits));
```

```

inds_pais=Seleciona_um_pai_customarginal(inds,DadosRef,N_ind,Custo_min,Fitness_base,
Deficit_base);
    inds_pais={inds_pais};
    case 'D', %BUSCA VERTICAL PURA - CONSIDERANDO COMO CRITERIO DE
SELECAO DO PAI O INDIVIDUO COM MENOR VALOR DE F CHAPEU - CUSTO
ATUAL + ESTIMATIVA DO CUSTO ATE A VIABILIDADE
        Fitness_base=Calcula_Fitness(ind,DadosRef);
        [a,b,c,d,e]=Ehviavel(ind,DadosRef);
        Matriz_deficits=e<0;
        Deficit_base=sum(sum(Matriz_deficits));

inds_pais=Seleciona_um_pai_menor_f(inds,DadosRef,N_ind,Custo_min,Fitness_base,Deficit
_base);
    inds_pais={inds_pais};
    case 'E', %BUSCA VERTICAL PURA - CONSIDERANDO COMO CRITERIO DE
SELECAO DO PAI UMA FUNCAO DE MERITO CALIBRADA ANTERIORMENTE (DE
MANEIRA ITERATIVA)
        inds_pais=Seleciona_um_pai_funcaomerito(inds,DadosRef,N_ind,Custo_min,nivel);
        inds_pais={inds_pais};
    case 'F', %BUSCA VERTICAL PURA - CONSIDERANDO COMO CRITERIO DE
SELECAO DO PAI UMA FUNCAO DE MERITO COM PRIORIZACAO DO CUSTO DA
SOBRA EM G

inds_pais=Seleciona_um_pai_funcaomerito_adaptada(inds,DadosRef,N_ind,Custo_min,nivel
);
    inds_pais={inds_pais};
end;
novoinde=[];
for j=1:length(inds_pais),
    ind_prov=programa_principal_recurso(inds_pais{j},DadosRef,Custo_min,nivel);
    aux=Calcula_Fitness(ind_prov,DadosRef);
    if ((Custo_min > aux ) | isempty(novoinde) ) & not(isempty(ind_prov))
        Custo_min=aux;
        novoinde=ind_prov;
    end; %do if ((Custo_minimo>aux
end; %do for j
end; %do if a
fprintf(' Custo Saida %d - %d: %8.4f \n',nivel,Custo_min);

```

PROGRAMA DE IMPLEMENTAÇÃO DO PROCEDIMENTO DE BUSCA VERTICAL COM LIMITAÇÃO DE CUSTO/ BUSCA ÁVIDA

```

function
novoind=viabiliza_ind_ExaustivaDesenfreada_fchapeu(ind,DadosRef,Custo_min,nivel,subni
vel,todos);
%esta funcao promove um processo de busca em profundidade ateh 120 seg. e depois passa
para um processo greedy...
eval(DadosRef);
global Fitness_base Tempo_max;
%custo da solucao mais barata - arbitrado inicialmente igual a infinito
if nargin==2,
    Custo_min=1000000;
    nivel=-1; %nivel eh o numero de vezes em que a funcao viabiliza_ind_new eh chamada
    subnivel='0'; %a posicao de um individuo na populacao "inds"
%   Fitness_base=Calcula_fitness(NewInd_inicial(DadosRef),DadosRef);
    Tempo_max=120;
    todos="";
    tic;
end;

nivel=nivel+1;
todos=[todos num2str(nivel) '.' subnivel ' '];

if mod(nivel,10)==0,
fprintf(['%6.3f Custo %8.4f \n %s\n'],toc,Custo_min,todos);
end;

for i=1:Conta_OCs(ind,DadosRef),
    Estado(i)=0;
end; %do for i
[a,b,c,d,Sobra_demanda]=Ehviavel(ind,DadosRef);
if a==0,
    novoind=ind;
else,
    OCs_atual=Conta_OCs(ind,DadosRef);
    [linha,coluna]=find(min(min(Sobra_demanda))==Sobra_demanda);
    linha=linha(1);
    coluna=coluna(1);
    OCs_completas=0; %numero de OCs em que eh impossivel atender o deficit de
linha/coluna
    for i=1:OCs_atual,
        if ind((i-1)*(NumTam+NumCor)+linha)==0,
            if b(i)<COP(linha),
                if OCs_completas<i,
                    OCs_completas=OCs_completas+1;
                end; %do if OCs_completas
                Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
            end; %do if b(i)
        elseif ind((i-1)*(NumTam+NumCor)+NumTam+coluna)==0,

```

```

    if c(i)<1,
        if OCs_completas<i,
            OCs_completas=OCs_completas+1;
        end; %do if OCs_completas
        Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
    end; %do if c(i)
elseif b(i)<COP(linha),
    if c(i)<1,
        if OCs_completas<i,
            OCs_completas=OCs_completas+1;
        end; %do if OCs_completas
        Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
    end; %do if c(i)
else,
    Estado(i)=0; %0 quer dizer que esta OC nao estah completa para este par tamanho/cor
end; %do if b
end; %do for i=1:OCs_atual
if OCs_completas==OCs_atual,

ind((OCs_atual)*(NumTam+NumCor)+1:(OCs_atual)*(NumTam+NumCor)+NumTam+NumCor)=zeros(size(NumTam+NumCor));
[a,b,c,d,Sobra_demanda]=Ehviavel(ind,DadosRef);
OCs_atual=OCs_atual+1;
Estado(OCs_atual)=0;
end; %do if OCs_completas

% Gerando a Populacao de Individuos
inds={ };
N_ind=1;
Fitness_base=Calcula_Fitness(ind,DadosRef);
[a,b,c,d,e]=Ehviavel(ind,DadosRef);
e=e<0;
Deficit_base=sum(sum(e));
%Incremento apenas do tamanho
for i=1:OCs_atual,
    if Estado(i)==0,
        if b(i)>=COP(linha) & not(ind((i-1)*(NumTam+NumCor)+NumTam+coluna)==0),
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+linha)=ind((i-1)*(NumTam+NumCor)+linha)+1;
            inds{N_ind,1}=ind_prov;
            fitness(N_ind)=Calcula_Fitness(ind_prov,DadosRef);
            [a,b,c,d,e]=Ehviavel(inds{N_ind,1},DadosRef);
            e=e<0;
            Deficit(N_ind)=sum(sum(e));
            fchapeu(N_ind)=fitness(N_ind)+((fitness(N_ind)-Fitness_base)/(Deficit_base-Deficit(N_ind)+0.5))*Deficit(N_ind);
            N_ind=N_ind+1;
        end; %do if b(i)
    end; %do if Estado(i)
end; %do if Estado(i)

```

```

end; %do for i
%Incremento apenas da camada
for i=1:OCs_atual,
    if Estado(i)==0,
        if c(i)>=1 & not(ind((i-1)*(NumTam+NumCor)+linha)==0),
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+NumTam+coluna)=ind((i-
1)*(NumTam+NumCor)+NumTam+coluna)+1;
            inds{N_ind,1}=ind_prov;
            fitness(N_ind)=Calcula_Fitness(ind_prov,DadosRef);
            [a,b,c,d,e]=Ehviavel(inds{N_ind,1},DadosRef);
            e=e<0;
            Deficit(N_ind)=sum(sum(e));
            fchapeu(N_ind)=fitness(N_ind)+((fitness(N_ind)-Fitness_base)/(Deficit_base-
Deficit(N_ind)+0.5))*Deficit(N_ind);
            N_ind=N_ind+1;
        end; %do if b(i)
    end; %do if Estado(i)
end; %do for i
%Incremento de marcacao e camada
for i=1:OCs_atual,
    if Estado(i)==0,
        if b(i)>=COP(linha) & c(i)>=1,
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+linha)=ind((i-
1)*(NumTam+NumCor)+linha)+1;
            ind_prov((i-1)*(NumTam+NumCor)+NumTam+coluna)=ind((i-
1)*(NumTam+NumCor)+NumTam+coluna)+1;
            inds{N_ind,1}=ind_prov;
            fitness(N_ind)=Calcula_Fitness(ind_prov,DadosRef);
            [a,b,c,d,e]=Ehviavel(inds{N_ind,1},DadosRef);
            e=e<0;
            Deficit(N_ind)=sum(sum(e));
            fchapeu(N_ind)=fitness(N_ind)+((fitness(N_ind)-Fitness_base)/(Deficit_base-
Deficit(N_ind)+0.5))*Deficit(N_ind);
            N_ind=N_ind+1;
        end; %do if b(i) & c(i)
    end; %do if Estado(i)
end; %do for i

% Viabilizando a Populacao;
novoinde=[]; %o individuo viavel eh inicializado
[best,pos]=sort(fchapeu); % "best" ordena os custos e "pos" aponta a posicao de cada custo
t=toc;
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if Calcula_Fitness(inds{i,1},DadosRef)< Custo_min,
        if Tempo_max>t,
            sub=[num2str(find(pos==i)) '/' num2str(length(pos))];

```

```

ind_prov=viabiliza_ind_ExaustivaDesenfreada_fchapeu(inds{i,1},DadosRef,Custo_min,nivel
,sub,todos);
    aux=Calcula_Fitness(ind_prov,DadosRef);
    if ((Custo_min > aux ) | isempty(novoind) ) & not(isempty(ind_prov))
        novoind=ind_prov;
        Custo_min=aux;
    end; %do if ((Custo_min
else,
    ind_prov=viabiliza_ind_new_vertical(inds{i,1},DadosRef,Custo_min,nivel,i);
    aux=Calcula_Fitness(ind_prov,DadosRef);
    if ((Custo_min > aux ) | isempty(novoind) ) & not(isempty(ind_prov))
        novoind=ind_prov;
        Custo_min=aux;
    end; %do if ((Custo_min
end; %do if ((nivel
end; %do if Calcula_Fitness
end; %do for i
end; %do if a
%fprintf(' Custo Saida %d - %d: %8.4f \n',nivel,subnivel,Custo_min);

```

PROGRAMA DE IMPLEMENTAÇÃO DA BUSCA ÁVIDA – critério (a)

```

function
[ind_pai,N_ind_pai,posicao_ind_pai]=Seleciona_um_pai_norma_deficit(inds,DadosRef,N_in
d,Custo_min,Deficit_1);
%esta funcao define o individuo considerado mais promissor para gerar
%filhos
eval(DadosRef);
N_ind_pai=1;
posicao_ind_pai=1;
Perc=1; %Percentual do Custo minimo aceito para que um individuo possa gerar filhos
%calculando o custo de cada solucao
for i=1:length(inds),
    fitness(i)=Calcula_Fitness(inds{i},DadosRef);
end; %do for i
ind_pai=[];
Max_norma=-1000;
% Definindo o melhor gerador de sucessores
[best,pos]=sort(fitness); %"best" ordena os custos e "pos" aponta a posicao de cada custo
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if fitness(i)<Perc*Custo_min,
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        e=e(e<0);
        vetor_e=e(1:size(e,1)*size(e,2)); %transformacao da matriz de inviabilidades em um
vetor
        Deficit_2=norm(vetor_e,2);
        if Max_norma<(Deficit_1-Deficit_2),
            Max_norma=Deficit_1-Deficit_2;
            ind_pai=inds{i};
            posicao_ind_pai=i;
        end; %do if norm(vetor_e)
    end; %do fitness(i)
end; %do for i

```


PROGRAMA DE IMPLEMENTAÇÃO DA BUSCA ÁVIDA – critério (b)

```

function
[ind_pai,N_ind_pai,posicao_ind_pai]=Seleciona_um_pai_customarginal(inds,DadosRef,N_in
d,Custo_min,Fitness_base,Deficit_base);
%esta funcao define o individuo considerado mais promissor para gerar
%filhos
eval(DadosRef);
N_ind_pai=1;
posicao_ind_pai=1;
Perc=1; %Percentual do Custo minimo aceito para que um individuo possa gerar filhos
%calculando o custo de cada solucao
for i=1:N_ind,
    fitness(i)=Calcula_Fitness(inds{i},DadosRef);
end; %do for i
ind_pai=[];
Min_custo=1000000;
% Definindo o melhor gerador de sucessores
[best,pos]=sort(fitness); % "best" ordena os custos e "pos" aponta a posicao de cada custo
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if fitness(i)<=Perc*Custo_min,
        Fitness_pai=fitness(i);
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_deficits=e<0;
        Deficit_pai=sum(sum(Matriz_deficits));
        if ((Fitness_pai-Fitness_base)/(Deficit_base-Deficit_pai+0.1))<Min_custo,
            Min_custo=((Fitness_pai-Fitness_base)/(Deficit_base-Deficit_pai+0.1));
            ind_pai=inds{i};
            posicao_ind_pai=i;
        end; %do if ((Fitness_pai
    end; %do if fitness(i)
end; %do for i

```

PROGRAMA DE IMPLEMENTAÇÃO DA BUSCA ÁVIDA – critério (c)

```

function
[ind_pai,N_ind_pai,posicao_ind_pai]=Seleciona_um_pai_menor_f(inds,DadosRef,N_ind,Cus
to_min,Fitness_base,Deficit_base);
%esta funcao define o individuo considerado mais promissor para gerar
%filhos, com base no conceito de "f chapeu" de Nilsson - custo mais
%estimativa do custo para se chegar a viabilidade
eval(DadosRef);
N_ind_pai=1;
posicao_ind_pai=1;
Perc=1; %Percentual do Custo minimo aceito para que um individuo possa gerar filhos
%calculando o custo de cada solucao
for i=1:N_ind,
    fitness(i)=Calcula_Fitness(inds{i},DadosRef);
end; %do for i
ind_pai=[];
Min_custo=1000000;
% Definindo o melhor gerador de sucessores
[best,pos]=sort(fitness); % "best" ordena os custos e "pos" aponta a posicao de cada custo
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if fitness(i)<=Perc*Custo_min,
        Fitness_pai=fitness(i);
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_deficits=e<0;
        Deficit_pai=sum(sum(Matriz_deficits));
        if (fitness(i)+(((Fitness_pai-Fitness_base)/(Deficit_base-
Deficit_pai+0.1))*Deficit_pai))<Min_custo,
            Min_custo=fitness(i)+(((Fitness_pai-Fitness_base)/(Deficit_base-
Deficit_pai+0.1))*Deficit_pai);
            ind_pai=inds{i};
            posicao_ind_pai=i;
        end; %do if ((Fitness_pai
        end; %do if fitness(i)
    end; %do for i

```

PROGRAMA DE IMPLEMENTAÇÃO DA BUSCA ÁVIDA – critério (d)

```

function
[ind_pai,N_ind_pai,posicao_ind_pai]=Seleciona_um_pai_funcao(inds,DadosRef,N_ind
,Custo_min,nivel);
%esta funcao define o individuo considerado mais promissor para gerar
%filhos, com base em uma funcao de merito do tipo  $f = g + B * h...$ 
eval(DadosRef);
N_ind_pai=1;
posicao_ind_pai=1;
B=100; %peso inicial atribuido a parcela que mede a inviabilidade
Perc=1; %Percentual do Custo minimo aceito para que um individuo possa gerar filhos
%calculando o custo de cada solucao
for i=1:N_ind,
    fitness(i)=Calcula_Fitness(inds{i},DadosRef);
end; %do for i
ind_pai=[];
Min_custo=1000000000;
% Definindo o melhor gerador de sucessores
[best,pos]=sort(fitness); % "best" ordena os custos e "pos" aponta a posicao de cada custo
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if fitness(i)<=Perc*Custo_min,
        Fitness_pai=fitness(i);
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_inviabilidade=(e<0);
        Inviabilidade=sum(sum(Matriz_inviabilidade));
        if (fitness(i)+(B/(nivel+0.1))*Inviabilidade)<Min_custo,
            Min_custo=(fitness(i)+(B/(nivel+0.1))*Inviabilidade);
            ind_pai=inds{i};
            posicao_ind_pai=i;
        end; %do if ((Fitness_pai
    end; %do if fitness(i)
end; %do for i

```

PROGRAMA DE IMPLEMENTAÇÃO DA BUSCA ÁVIDA – critério (e)

```

function
[ind_pai,N_ind_pai,posicao_ind_pai]=Seleciona_um_pai_funcao_merito_adaptada(inds,Dados
Ref,N_ind,Custo_min,nivel);
%esta funcao define o individuo considerado mais promissor para gerar
%filhos, com base em uma funcao de merito do tipo  $f = g + B \cdot h \dots$ 
eval(DadosRef);
N_ind_pai=1;
posicao_ind_pai=1;
B=100; %peso inicial atribuido a parcela que mede a inviabilidade
Perc=1; %Percentual do Custo minimo aceito para que um individuo possa gerar filhos
%calculando o custo de cada solucao
for i=1:N_ind,
    [a,b,c,d,e,f,g]=Calcula_Fitness(inds{i},DadosRef);
    fitness_adaptado(i)=(0.7*e)+(0.3*(b+c+d+f+g));
    fitness(i)=a;
end; %do for i
ind_pai=[];
Min_custo=1000000000;
% Definindo o melhor gerador de sucessores
[best,pos]=sort(fitness); %"best" ordena os custos e "pos" aponta a posicao de cada custo
for i=pos(best<=Custo_min), %para cada posicao cujo custo seja menor ou igual ao
Custo_min
    if fitness(i)<=Perc*Custo_min,
        %Fitness_pai=fitness(i);
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_inviabilidade=(e<0);
        Inviabilidade=sum(sum(Matriz_inviabilidade));
        if (fitness_adaptado(i)+(B/(nivel+0.1))*Inviabilidade)<Min_custo,
            Min_custo=(fitness_adaptado(i)+(B/(nivel+0.1))*Inviabilidade);
            ind_pai=inds{i};
            posicao_ind_pai=i;
        end; %do if ((Fitness_pai
    end; %do if fitness(i)
end; %do for i

```

PROGRAMA PRINCIPAL DE IMPLEMENTAÇÃO DA BUSCA COM LISTA DE NÓS ABERTOS E LIMITAÇÃO DE CUSTO

```

function
novoind=programa_principal_nosabertosefechados_custofchapeu(ind,DadosRef,Custo_min,n
ivel);
%esta funcao consiste em um processo de busca horizontal, na qual cria-se uma lista de nohs
abertos que
%serao geradores de filhos. Para entrar na lista de abertos, o noh tem que ter um custo
estimado (f chapeu)
%menor que o da melhor solucao obtida ateh o momento...
eval(DadosRef);
if nargin==2, %na primeira iteracao alguns parametros sao inicializados
    Custo_min=1000000;
    nivel=-1; %nivel eh o numero de vezes em que a funcao viabiliza_ind_new eh chamada
    funcao=upper(input('Entre com o metodo a usar:', 's'));
    Fitness_base=Calcula_Fitness(ind,DadosRef);
    [a,b,c,d,e]=Ehviavel(ind,DadosRef);
    Matriz_deficits=e<0;
    Deficit_base=sum(sum(Matriz_deficits));
end;
N_max=20;
[a,b,c,d,e]=Ehviavel(ind,DadosRef);
novoind=[];
inds={ind};
    while length(inds)>0
        N_ind=length(inds);
        disp(N_ind);
        nivel=nivel+1;
        fprintf(['\n Custo Entrada %d - %d : %8.4f'],nivel,Custo_min);
        if funcao=='A', %escolha do pai com a menor metrica de inviabilidade - ANALOGO A
FUNCAO B DO PROGRAMA RECURSIVO
            Deficit_1=1000000;

[ind_pai,N,posicao_pai]=Seleciona_um_pai_norma_deficit(inds,DadosRef,N_ind,Custo_min,
Deficit_1);
            elseif funcao=='B', %escolha do pai com o menor custo marginal de reducao de
inviabilidade - ANALOGO A FUNCAO C DO PROGRAMA RECURSIVO

[ind_pai,N,posicao_pai]=Seleciona_um_pai_customarginal(inds,DadosRef,N_ind,Custo_min,
Fitness_base,Deficit_base);
            elseif funcao=='C', %escolha do pai com o menor custo fchapeu - ANALOGO A FUNCAO
D DO PROGRAMA RECURSIVO

[ind_pai,N,posicao_pai]=Seleciona_um_pai_menor_f(inds,DadosRef,N_ind,Custo_min,Fitne
ss_base,Deficit_base);
            elseif funcao=='D', %escolha do pai com o menor valor da funcao de merito previamente
definida - ANALOGO A FUNCAO E DO PROGRAMA RECURSIVO

```

```

[ind_pai,N,posicao_pai]=Seleciona_um_pai_funcao(inds,DadosRef,N_ind,Custo_min,
nivel);
elseif funcao=='E', %escolha do pai com o menor valor da funcao de merito adaptada -
ANALOGO A FUNCAO F DO PROGRAMA RECURSIVO

[ind_pai,N,posicao_pai]=Seleciona_um_pai_funcao(inds,DadosRef,N_ind,C
usto_min,nivel);
end; %DO IF
if Conta_OCs(ind_pai,DadosRef)>0,
% Retira da lista
inds={inds{[1:(posicao_pai-1) (posicao_pai+1):end]}};
N_ind=length(inds);
[a,b,c,d,e]=Ehviavel(ind_pai,DadosRef);
if a==0,
custo_pai=Calcula_Fitness(ind_pai,DadosRef);
if custo_pai<Custo_min,
Custo_min=custo_pai;
novoiind=ind_pai;
end;
else
[sucessores,N_sucessores]=GERA_SUCESSORES_nosabertos(ind_pai,DadosRef);
%inclui sucessores na lista

for inicio=1:N_sucessores,
[a,b,c,d,e]=Ehviavel(sucessores{inicio},DadosRef);
e=e<0;
Deficit(inicio)=sum(sum(e));
hchapeu(inicio)=((Calcula_Fitness(sucessores{inicio},DadosRef)-
Fitness_base)/(Deficit_base-Deficit(inicio)+0.1))*Deficit(inicio);
if (Calcula_Fitness(sucessores{inicio},DadosRef)+
hchapeu(inicio))<(Custo_min*(a>0)),
inds=[inds {sucessores{inicio}}];
%inds{N_ind+1}=sucessores{inicio};
elseif (Calcula_Fitness(sucessores{inicio},DadosRef)<(Custo_min*(a==0)),
Custo_min=Calcula_Fitness(sucessores{inicio},DadosRef);
novoiind=sucessores{inicio};
end; %do if (Calcula_Fitness(sucessores{inicio},DadosRef)
end; %do for inicio
N_ind=length(inds);
if N_ind>N_max, %mantendo fixo o numero de nohs abertos
Vetor_parametro_ordenacao=[];
%ordenando com base em algum criterio pre-definido
funcao_exclusao=funcao;

[Vetor_parametro_ordenacao]=Ordena_para_exclusao(inds,N_ind,DadosRef,funcao_exclusao
,Fitness_base,Deficit_base,nivel);
[best,pos]=sort(Vetor_parametro_ordenacao);
inds={inds{pos(1:N_max)}};
N_ind=length(inds);

```

```
        end; %do if N_ind
    end; %do if a==0
    ind=ind_pai; %O ind eh atualizado por causa do GERA_SUCESSORES
else,
    inds={ };
end; %do if Conta_OCs
disp(ind_pai);
disp(N_ind);
end; %do while
fprintf(' Custo Saida %d - %d: %8.4f \n',nivel,Custo_min);
```

PROGRAMA PRINCIPAL DE IMPLEMENTAÇÃO DA BUSCA COM LISTA DE NÓS ABERTOS E FECHADOS/ BUSCA ÁVIDA

```

function
novoind=programa_principal_nosabertosefechados_exaustivadesenfreada(ind,DadosRef,Custo_min,nivel);
%esta funcao consiste em um processo de busca horizontal, na qual cria-se uma lista de nohs abertos que
%serao geradores de filhos. Para entrar na lista de abertos, o noh tem que
%ter um custo (FUNÇÃO OBJETIVO) menor que o da melhor solucao obtida ateh o momento...
eval(DadosRef);
if nargin==2, %na primeira iteracao alguns parametros sao inicializados
    Custo_min=1000000;
    nivel=-1; %nivel eh o numero de vezes em que a funcao viabiliza_ind_new eh chamada
    funcao=upper(input('Entre com o metodo a usar:', 's'));
    Fitness_base=Calcula_Fitness(ind,DadosRef);
    [a,b,c,d,e]=Ehviavel(ind,DadosRef);
    Matriz_deficits=e<0;
    Deficit_base=sum(sum(Matriz_deficits));
    Tempo_max=120;
end;
N_max=50;
[a,b,c,d,e]=Ehviavel(ind,DadosRef);
novoind=[];
inds={ind};
t1=cputime;
while length(inds)>0
    if (cputime-t1)<Tempo_max, %se ainda estiver dentro do prazo para busca exaustiva...
        N_ind=length(inds);
        disp(N_ind);
        nivel=nivel+1;
        fprintf(['\n Custo Entrada %d - %d : %8.4f'],nivel,Custo_min);
        if funcao=='A', %escolha do pai com a menor metrica de inviabilidade - ANALOGO A
FUNCAO B DO PROGRAMA RECURSIVO
            Deficit_1=1000000;

[ind_pai,N,posicao_pai]=Seleciona_um_pai_norma_deficit(inds,DadosRef,N_ind,Custo_min,
Deficit_1);
            elseif funcao=='B', %escolha do pai com o menor custo marginal de reducao de
inviabilidade - ANALOGO A FUNCAO C DO PROGRAMA RECURSIVO

[ind_pai,N,posicao_pai]=Seleciona_um_pai_customarginal(inds,DadosRef,N_ind,Custo_min,
Fitness_base,Deficit_base);
            elseif funcao=='C', %escolha do pai com o menor custo fchapeu - ANALOGO A FUNCAO
D DO PROGRAMA RECURSIVO

[ind_pai,N,posicao_pai]=Seleciona_um_pai_menor_f(inds,DadosRef,N_ind,Custo_min,Fitne
ss_base,Deficit_base);

```



```
elseif funcao=='D', %escolha do pai com o menor valor da funcao de merito previamente
definida - ANALOGO A FUNCAO E DO PROGRAMA RECURSIVO
```

```
[ind_pai,N,posicao_pai]=Seleciona_um_pai_funcaomerito(inds,DadosRef,N_ind,Custo_min,
nivel);
```

```
elseif funcao=='E', %escolha do pai com o menor valor da funcao de merito previamente
definida - ANALOGO A FUNCAO E DO PROGRAMA RECURSIVO
```

```
[ind_pai,N,posicao_pai]=Seleciona_um_pai_funcaomerito_adaptada(inds,DadosRef,N_ind,C
usto_min,nivel);
```

```
end; %do if
```

```
if Conta_OCs(ind_pai,DadosRef)>0,
```

```
    % Retira da lista
```

```
    inds={inds{[1:(posicao_pai-1) (posicao_pai+1):end]}};
```

```
    N_ind=length(inds);
```

```
    [a,b,c,d,e]=Ehviavel(ind_pai,DadosRef);
```

```
    if a==0,
```

```
        custo_pai=Calcula_Fitness(ind_pai,DadosRef);
```

```
        if custo_pai<Custo_min,
```

```
            Custo_min=custo_pai;
```

```
            novoind=ind_pai;
```

```
        end;
```

```
    else
```

```
        [sucessores,N_sucessores]=GERA_SUCESSORES_nosabertos(ind_pai,DadosRef);
```

```
        %inclui sucessores na lista
```

```
        for inicio=1:N_sucessores,
```

```
            [a,b,c,d,e]=Ehviavel(sucessores{inicio},DadosRef);
```

```
            e=e<0;
```

```
            Deficit(inicio)=sum(sum(e));
```

```
            hchapeu(inicio)=((Calcula_Fitness(sucessores{inicio},DadosRef)-
Fitness_base)/(Deficit_base-Deficit(inicio)+0.1))*Deficit(inicio);
```

```
            if Calcula_Fitness(sucessores{inicio},DadosRef)<(Custo_min*(a>0)),
```

```
                inds=[inds {sucessores{inicio}}];
```

```
                %inds{N_ind+1}=sucessores{inicio};
```

```
            elseif (Calcula_Fitness(sucessores{inicio},DadosRef)<(Custo_min*(a==0)),
```

```
                Custo_min=Calcula_Fitness(sucessores{inicio},DadosRef);
```

```
                novoind=sucessores{inicio};
```

```
                end; %do if (Calcula_Fitness(sucessores{inicio},DadosRef)
```

```
            end; %do for inicio
```

```
N_ind=length(inds);
```

```
if N_ind>N_max, %mantendo fixo o numero de nohs abertos
```

```
    Vetor_parametro_ordenacao=[];
```

```
    %ordenando com base em algum criterio pre-definido
```

```
    funcao_exclusao=funcao;
```

```
[Vetor_parametro_ordenacao]=Ordena_para_exclusao(inds,N_ind,DadosRef,funcao_exclusao
,Fitness_base,Deficit_base,nivel);
```

```
[best,pos]=sort(Vetor_parametro_ordenacao);
```

```
inds={inds{pos(1:N_max)}};
```

```

        N_ind=length(inds);
        end; %do if N_ind
    end; %do if a==0
    ind=ind_pai; %O ind eh atualizado por causa do GERA_SUCESSORES
else,
    inds={ };
end; %do if Conta_OCs
disp(ind_pai);
disp(N_ind);
else, %se tiver acabado o prazo da busca exaustiva...
    Vetor_parametro_ordenacao=[];
    %ordenando com base em algum criterio pre-definido
    funcao_exclusao=funcao;
    N_ind=length(inds);

[Vetor_parametro_ordenacao]=Ordena_para_exclusao(inds,N_ind,DadosRef,funcao_exclusao
,Fitness_base,Deficit_base,nivel);
    [best,pos]=sort(Vetor_parametro_ordenacao);
    for i=1:length(Vetor_parametro_ordenacao),
        [a,b,c,d,e]=Ehviavel(inds{pos(i)},DadosRef);
        while Calcula_Fitness(inds{pos(i)},DadosRef)<Custo_min*(a>0),
            filho=Gera_um_sucessor(inds{pos(i)},DadosRef);
            [a,b,c,d,e]=Ehviavel(filho,DadosRef);
            if Calcula_Fitness(filho,DadosRef)<Custo_min*(a==0),
                novound=filho;
                Custo_min=Calcula_Fitness(filho,DadosRef);
            end; %do if
            inds{pos(i)}=filho;
        end; %do while
        disp(novound);
        disp(Custo_min);
    end; %do for
    inds={ };
end; %do if (cputime-t1)
end; %do while
fprintf(' Custo Saida %d - %d: %8.4f \n',nivel,Custo_min);

```

PROGRAMA DE EXCLUSÃO DE NÓS DA LISTA DE ABERTOS

```

function
[Vetor_parametro_ordenacao]=Ordena_para_exclusao(inds,N_ind,DadosRef,funcao_exclusao
,Fitness_base,Deficit_base,nivel);
%esta funcao calcula um vetor de valores para exclusao de nohs da lista de abertos, em funcao
do criterio selecionado anteriormente para escolha do pai...
eval(DadosRef);
if funcao_exclusao=='A', %ordenacao em funcao da norma do vetor de inviabilidades
    for i=1:N_ind,
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        e=e<0;
        Vetor_inviabilidade=e(1:size(e,1)*size(e,2));
        Vetor_parametro_ordenacao(i)=norm(Vetor_inviabilidade,2);
    end; %do for i
elseif funcao_exclusao=='B', %ordenacao em funcao do menor custo marginal para reducao
de inviabilidade
    for i=1:N_ind,
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        e=e<0;
        Deficit(i)=sum(sum(e));
        Vetor_parametro_ordenacao(i)=((Calcula_Fitness(inds{i},DadosRef)-
Fitness_base)/(Deficit_base-Deficit(i)+0.1));
    end; %do for i
elseif funcao_exclusao=='C', %ordenacao em funcao de um f chapau - g + h chapau
    for i=1:N_ind,
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        e=e<0;
        Deficit(i)=sum(sum(e));

    Vetor_parametro_ordenacao(i)=Calcula_Fitness(inds{i},DadosRef)+(((Calcula_Fitness(inds{
i},DadosRef)-Fitness_base)/(Deficit_base-Deficit(i)+0.1))*Deficit(i));
    end; %do for i
elseif funcao_exclusao=='D', %ordenacao com base na funcao de merito f = g + P*h
    B=100;
    for i=1:N_ind,
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_inviabilidade=(e<0);
        Deficit(i)=sum(sum(Matriz_inviabilidade));

    Vetor_parametro_ordenacao(i)=Calcula_Fitness(inds{i},DadosRef)+((B/(nivel+0.1))*Deficit(
i));
    end; %do for i
elseif funcao_exclusao=='E', %ordenacao com base na funcao de merito f = g_adaptado + P*h
    B=100;
    for i=1:N_ind,
        [A,B,C,D,E,F,G]=Calcula_Fitness(inds{i},DadosRef);
        [a,b,c,d,e]=Ehviavel(inds{i},DadosRef);
        Matriz_inviabilidade=(e<0);
        Deficit(i)=sum(sum(Matriz_inviabilidade));

```

```
Vetor_parametro_ordenacao(i)=(0.7*E)+(0.3*(B+C+D+F+G))+((B/(nivel+0.1))*Deficit(i));  
    end; %do for i  
end; %do if funcao_exclusão
```

PROGRAMA DE EXPANSÃO DE NÓS

```

function [inds,N_ind]=GERA_SUCESSORES(ind,DadosRef);
%esta funcao verifica a necessidade de inserir mais uma mesa, alem de gerar sucessores de
um "pai" selecionado previamente...
eval(DadosRef);
%definindo o par tamanho x cor para o qual o deficit serah reduzido
[linha,coluna]=Encontra_par(ind,DadosRef); %funcao que encontra o par "tamanhoxcor" cujo
deficit deve ser gerado
    linha=linha(1);
    coluna=coluna(1);
%verificando se ainda cabem marcacoes/camadas nas mesas jah existentes
for i=1:Conta_OCs(ind,DadosRef),
    Estado(i)=0;
end; %do for i
[a,b,c,d,Sobra_demanda]=Ehviavel(ind,DadosRef);
OCs_atual=Conta_OCs(ind,DadosRef);
OCs_completas=0; %numero de OCs em que eh impossivel atender o deficit de
linha/coluna
for i=1:OCs_atual,
    if ind((i-1)*(NumTam+NumCor)+linha)==0,
        if b(i)<COP(linha),
            if OCs_completas<i,
                OCs_completas=OCs_completas+1;
            end; %do if OCs_completas
            Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
        end; %do if b(i)
    elseif ind((i-1)*(NumTam+NumCor)+NumTam+coluna)==0,
        if c(i)<1,
            if OCs_completas<i,
                OCs_completas=OCs_completas+1;
            end; %do if OCs_completas
            Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
        end; %do if c(i)
    elseif b(i)<COP(linha),
        if c(i)<1,
            if OCs_completas<i,
                OCs_completas=OCs_completas+1;
            end; %do if OCs_completas
            Estado(i)=1; %1 quer dizer que esta OC estah completa para este par linha/coluna
        end; %do if c(i)
    else,
        Estado(i)=0; %0 quer dizer que esta OC nao estah completa para este par tamanho/cor
    end; %do if b
end; %do for i=1:OCs_atual
if OCs_completas==OCs_atual, %incremento da OC em mais uma mesa

ind((OCs_atual)*(NumTam+NumCor)+1:(OCs_atual)*(NumTam+NumCor)+NumTam+NumCor)=zeros(size(NumTam+NumCor));
    [a,b,c,d,Sobra_demanda]=Ehviavel(ind,DadosRef);

```

```

    OCs_atual=OCs_atual+1;
    Estado(OCs_atual)=0;
end; %do if OCs_completas

% Gerando a Populacao de Individuos
inds={ };
N_ind=0;
%Incremento apenas do tamanho
for i=1:OCs_atual,
    if Estado(i)==0,
        if b(i)>=COP(linha) & not(ind((i-1)*(NumTam+NumCor)+NumTam+coluna)==0),
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+linha)=ind((i-
1)*(NumTam+NumCor)+linha)+1;
            N_ind=N_ind+1;
            inds{N_ind}=ind_prov;
        end; %do if b(i)
    end; %do if Estado(i)
end; %do for i
%Incremento apenas da camada
for i=1:OCs_atual,
    if Estado(i)==0,
        if c(i)>=1 & not(ind((i-1)*(NumTam+NumCor)+linha)==0),
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+NumTam+coluna)=ind((i-
1)*(NumTam+NumCor)+NumTam+coluna)+1;
            N_ind=N_ind+1;
            inds{N_ind}=ind_prov;
        end; %do if b(i)
    end; %do if Estado(i)
end; %do for i
%Incremento de marcacao e camada
for i=1:OCs_atual,
    if Estado(i)==0,
        if b(i)>=COP(linha) & c(i)>=1,
            ind_prov=ind;
            ind_prov((i-1)*(NumTam+NumCor)+linha)=ind((i-
1)*(NumTam+NumCor)+linha)+1;
            ind_prov((i-1)*(NumTam+NumCor)+NumTam+coluna)=ind((i-
1)*(NumTam+NumCor)+NumTam+coluna)+1;
            N_ind=N_ind+1;
            inds{N_ind}=ind_prov;
        end; %do if b(i) & c(i)
    end; %do if Estado(i)
end; %do for i

```

PROGRAMA DE CÁLCULO DO CUSTO DE UM NÓ

```

function [F,Fenf,Fcor,Ftec,Fsobra,Festoque,Fperda]=Calcula_Fitness(individuo,DadosRef);
%funcao que calcula o Fitness e monta um vetor com esta informacao para
%toda a populacao
eval(DadosRef);
NMA=[];
NCA=[];

for k=1:Conta_OC(individuo, DadosRef),
    NMA(end+1:end+size(DEMANDA,1))=individuo(1+(k-
    1)*(size(DEMANDA,1)+size(DEMANDA,2)):size(DEMANDA,1)+(k-
    1)*(size(DEMANDA,1)+size(DEMANDA,2)));
    NCA(end+1:end+size(DEMANDA,2))=individuo((k-
    1)*(size(DEMANDA,1)+size(DEMANDA,2))+ size(DEMANDA,1)+1:(k-
    1)*(size(DEMANDA,1)+size(DEMANDA,2))+size(DEMANDA,1)+size(DEMANDA,2));
end;

DELTA=(NCA>0);
K=Conta_OC(individuo, DadosRef);
C=size(DEMANDA,2);
T=size(DEMANDA,1);
%OS primeiros C sao da primeira ordem de corte.

MatNMA=reshape(NMA,T,K)';

% Cada Linha uma ordem de corte
MatNCA=reshape(NCA,C,K)';

MatDELTA=reshape(DELTA,C,K)';

Fenf=0;
Fcor=0;
Ftec=0;
Fsobra=0;
Festoque=0;
Fperda=0;
% CUSTOENF * (sum{k in K} ( AENF *(sum{c in C} DELTA[k,c]) +
%
%           BENF *(sum{c in C} NCA[k,c]) +
%           CENF *( sum {t in T} (NMA[k,t]*COP[t])) ))

for k=1:K,
    Fenf=Fenf+AENF*sum(MatDELTA(k,:))+BENF*sum(MatNCA(k,:))+ ...
        + CENF*(MatNMA(k,:)*COP);
end;
Fenf=Fenf*CUSTOENF;
% CUSTOCOR * (sum{k in K} ( ACOR + BCOR *( sum{c in C} NCA[k,c]) +
%
%           CCOR * ( sum {t in T} (NMA[k,t]*PERCORTE[t])) )) +

```

```

for k=1:K,
  Fcor=Fcor+ACOR+BCOR*sum(MatNCA(k,:))+ ...
  CCOR*(MatNMA(k,:)*PERCORTE);
end;
Fcor=Fcor*CUSTOCOR;

% CUSTOTEC * (sum{k in K} ((APRMAX-(PARC -
% (PARB/( (0.001+sum{t in T}(NMA[k,t]*COP[t]) ^PARA)))) *
% (sum{t in T,c in C} (GRAMAT * LARG * COP[t] *
% NMA[k,t] * NCA[k,c]))) ) +
% CUSTOTEC * (sum{k in K} (APRMAX - (if(sum{t in T}(NMA[k,t]*COP[t]))<=1 then
0.7535 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=2 then 0.823 else if(sum{t in
T}(NMA[k,t]*COP[t]))<=3 then .8344 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=4 then .8458 else if(sum{t in
T}(NMA[k,t]*COP[t]))<=5 then .8532 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=6 then .8614 else if(sum{t in
T}(NMA[k,t]*COP[t]))<=7 then .8638 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=8 then .8662 else if(sum{t in
T}(NMA[k,t]*COP[t]))<=9 then .8678 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=10 then .8686 else if(sum{t in
T}(NMA[k,t]*COP[t]))<=11 then .869 else
% if(sum{t in T}(NMA[k,t]*COP[t]))<=12 then .869 else if(sum{t in
T}(NMA[k,t]*COP[t]))>12 then .8693))*
% (sum{t in T,c in C} (GRAMAT * LARG * COP[t]*NMA[k,t] * NCA[k,c]))) +
APROV=[0.7535;0.823;0.8344;0.8458;0.8532;0.8614;0.8638;0.8662;0.8678;0.8686;0.869;0.
869;0.8693];
for k=1:K,
  Comprimento=MatNMA(k,:)*COP;
  if (Comprimento ~= 0),
    Ftec=Ftec+(APRMAX-
(APROV(min(13,ceil(Comprimento)))))*sum(MatNCA(k,:))*GRAMAT*LARG*Comprimen
to;
  end;
end;
% for k=1:K,
% Ftec=Ftec+(param{27}-(param{30}-(param{29}/((0.001+ (MatNMA(k,:)*COP)
)^param{28}))))*...
% param{16}*param{17}*(MatNMA(k,:)*COP)*sum(MatNCA(k,:));
% end;
Ftec=Ftec*CUSTOTEC;

% Custo PERDA
% CUSTOTEC * (sum{k in K,c in C} (GRAMAT * LARG * PERDADOB * NCA[k,c])) +
for k=1:K,
  Fperda=Fperda+GRAMAT*LARG*PERDADOB*sum(NCA(1+(k-1)*C:k*C));
end;
Fperda=Fperda*CUSTOTEC;

```



```

% Custo de sobra
% PED * PVE * sum{t in T,c in C} (max(0,(sum{k in K} (NCA[k,c]*NMA[k,t]) -
DEMANDA[t,c]-DEMFUT[t,c])) +
for t=1:T,
    for c=1:C,
        NPecas(t,c)=0;
        for k=1:K,
            NPecas(t,c)=NPecas(t,c)+MatNCA(k,c)*MatNMA(k,t);
        end;
        Fsobra=Fsobra+(max(0,NPecas(t,c)-DEMANDA(t,c)-DEMFUT(t,c)));
    end;
end;
Fsobra=Fsobra*PED*PVE;

%Custo Estoque
% PVE * 0.6 *TMA *PPE * sum{t in T,c in C} (max(0,(sum{k in K}
(NCA[k,c]*NMA[k,t])-DEMANDA [t,c]));
for t=1:T,
    for c=1:C,
        Festoque=Festoque+max(0,NPecas(t,c)-DEMANDA(t,c));
    end;
end;

Festoque=Festoque*PVE*TMA*PPE*0.6;
F= Fenf+Fcor+Ftec+Fsobra+Festoque+Fperda;

```

PROGRAMA DE CÁLCULO DA MATRIZ DE INVIABILIDADES DE UM NÓ

```

function
[viavel,ResCompMax,ResEspFaca,ResDelta,ResDem]=Ehviavel(individuo,DadosRef)
eval(DadosRef);

NMA=[];
NCA=[];
if (nargin<3),
    NOCs=Conta_OCs(individuo,DadosRef);
end;
for k=1:NOCs,
    NMA(end+1:end+size(DEMANDA,1))=individuo(1+(k-
1)*(size(DEMANDA,1)+size(DEMANDA,2)):size(DEMANDA,1)+(k-
1)*(size(DEMANDA,1)+size(DEMANDA,2)));
    NCA(end+1:end+size(DEMANDA,2))=individuo((k-
1)*(size(DEMANDA,1)+size(DEMANDA,2))+size(DEMANDA,1)+1:(k-
1)*(size(DEMANDA,1)+size(DEMANDA,2))+size(DEMANDA,1)+size(DEMANDA,2));
end;

DELTA=(NCA>0);

[ResCompMax,ResEspFaca,ResDelta,ResDem]=Restricoes([NMA';NCA';DELTA'],DadosRe
f);
viavel=0;
for k=1:NOCs,
    if (ResCompMax(k)<0),
        viavel=1;
        break;
    end;
    if (ResEspFaca(k)<0),
        viavel=2;
        break;
    end;
    for c=1:size(DEMANDA,2),
        if (ResDelta(k,c)~=0),
            viavel=0;
            %break;
        end;
    end;
end;

for c=1:size(DEMANDA,2),
    for t=1:size(DEMANDA,1),
        if (ResDem(t,c)<0),
            viavel=8;
            break;
        end;
    end;
end;
end;

```

APÊNCICE 05: RESULTADOS OBTIDOS PARA OS DEMAIS CENÁRIOS

QUADRO 14: Resultados obtidos para o cenário dados_r1_gera2

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 10 12 12 8	1022,1	0,0001
Heurística proposta	4 5 6 5 5 6 10 11 11 8	725,14	0,11
MINLP	4 4 7 6 5 6 9 7 11 1 2 5 1 3 3 3 0 6 0 8	144,64	Não Aplicável
BVLC	4 5 6 6 6 6 9 9 10 3 1 3 0 1 1 1 0 3 1 11	240,88	150,11
BA – a	3 7 6 5 5 7 10 12 11 6	1096,59	4,23
BA-b	4 6 6 5 6 6 9 9 8 3 1 2 0 3 2 3 2 0 5 9	386,43	7,85
BA – c	4 6 6 5 6 6 9 9 8 3 1 2 0 3 2 3 2 0 5 9	386,43	7,36
BA – d	4 6 6 5 6 6 9 9 8 3 1 2 0 2 2 3 3 0 5 17	803,97	8,84
BA - e	4 6 6 5 6 6 9 9 8 3 1 2 0 2 2 3 3 0 5 17	803,97	8,02
BVLC/BA	4 5 6 6 6 6 9 9 9 3 1 3 0 2 2 4 0 3 2 7	212,83	776,37
BLNALC – a	4 5 6 6 6 6 9 9 9 5 0 3 0 0 1 4 0 3 2 4	313,95	87,5
BLNALC – b	4 5 6 6 6 6 9 9 9 3 1 3 0 2 2 4 0 3 2 7	212,83	364,32
BLNALC – c	4 5 6 6 6 6 9 9 9 3 1 3 0 2 2 4 0 3 2 7	212,83	408,81
BLNALC – d	4 5 6 6 6 6 9 9 9 3 1 2 0 1 1 2 0 7 4 13	298,12	479,83
BLNALC - e	3 7 6 6 6 5 7 8 6 6 2 0 2 2 2 3 6 6 12 1	368,66	701,23
BLNALC/BA – a	4 5 6 6 6 6 9 9 9 5 0 2 0 0 1 3 0 5 3 6	320,58	190,81
BLNALC/BA – b	4 5 6 6 6 6 9 9 9 3 1 3 0 2 2 4 0 3 2 7	212,83	177,90
BLNALC/BA – c	4 5 6 6 6 6 9 9 9 3 1 3 0 2 2 4 0 3 2 7	212,83	180,05
BLNALC/BA – d	4 5 7 6 5 6 8 9 8 3 1 5 0 6 7 6 1 2 3 5	318,28	219,65
BLNALC/BA - e	4 4 6 6 6 6 9 9 10 3 1 4 0 2 1 1 0 5 1 11	165,40	207,73

QUADRO 15: Resultados obtidos para o cenário Dados_R1_gera3

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 10 12 12 8	3952	0,0001
Heurística proposta	4 5 6 5 5 6 10 11 11 8	2635	0,05
MINLP	-	-	Ultrapassou limite de tempo permitido pelo NEOS
BVLC	-	1626,02	> 10800
BA – a	3 7 6 5 5 7 10 12 11 6	3146,59	4,83
BA-b	3 5 6 5 6 7 8 7 7 2 2 3 2 3 2 2 4 8 7 10	2059,67	8,57
BA – c	3 5 6 5 6 7 8 7 7 2 2 3 2 3 2 2 4 8 7 10	2059,67	10,05
BA – d	3 5 6 6 6 6 9 6 9 3 6 9 7 3 4 6 0 3 2 4	2095,38	9,61
BA - e	3 5 6 6 6 6 9 6 9 3 6 9 7 3 4 6 0 3 2 4	2095,38	9,06
BVLC/BA	3 5 6 5 6 7 6 6 9 5 2 3 2 2 1 0 10 9 2 5	1442,86	1740
BLNALC – a	4 6 6 5 6 6 7 9 10 6 0 0 3 3 1 1 5 0 1 0	1466,33	2995
BLNALC – b	4 5 6 7 5 6 5 6 7 3 1 2 2 1 2 2 15 12 10 11	2144,74	652,74
BLNALC – c	4 5 6 7 5 6 5 6 7 3 1 2 2 1 2 2 15 12 10 11	2144,74	758,68
BLNALC – d	3 5 6 5 6 7 5 5 7 3 1 3 2 2 1 1 14 21 13 11	2723,32	701,4
BLNALC - e	3 7 6 6 6 5 6 8 6 6 1 0 1 1 1 2 18 12 19 2	1985,22	858,05
BLNALC/BA – a	4 6 6 5 6 6 9 9 10 5 0 2 0 3 0 1 2 0 1 3	1696,3	208,61
BLNALC/BA – b	3 5 6 5 6 7 6 6 8 3 6 9 6 7 4 3 3 3 2 3	1679,5	234,8
BLNALC/BA – c	3 5 6 5 6 7 6 6 8 3 6 9 6 7 4 3 3 3 2 3	1679,5	229,81
BLNALC/BA – d	3 5 6 5 6 7 6 6 8 3 6 9 6 7 4 3 3 3 2 3	1679,5	244,25
BLNALC/BA - e	3 5 6 5 6 7 6 6 8 3 6 9 6 7 4 3 3 3 2 3	1679,5	249,47

QUADRO 16: Resultados obtidos para o cenário Dados_R1_gera4

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 16 19 18 7 0 17 0 6 5 6 0 0 0 2 34 0 0 0 0 0 0 0 0 1	488,63	0,0001
Heurística proposta	4 5 6 6 6 6 15 17 16 2	532,75	0,11
MINLP	4 5 6 6 6 6 15 17 16 6 0 3 0 1 1 1 0 0 0 10	165,36	Não aplicável
BVLC	3 6 7 6 5 6 13 15 14 5 2 2 0 2 4 3 2 6 5 13	191,74	93,65
BA – a	3 5 8 6 5 6 14 18 17 11 1 1 0 0 2 0 0 2 2 1	673,58	12,36
BA-b	3 6 7 6 5 6 13 14 14 5 2 2 0 3 4 3 2 7 5 13	194,82	20,16
BA – c	3 6 7 6 5 6 13 14 14 5 2 2 0 3 4 3 2 7 5 13	194,82	20,6
BA – d	3 6 7 6 5 6 13 14 14 5 2 2 0 3 4 3 2 7 5 13	194,82	21,2
BA - e	3 6 7 6 5 6 13 14 14 5 2 2 0 3 4 3 2 7 5 13	194,82	22,5
BVLC/BA	4 6 6 6 5 6 15 15 14 6 0 2 0 1 4 3 0 1 5 10	173,74	471,37
BLNALC – a	4 5 6 6 6 6 15 17 16 8 0 4 0 0 0 0 0 0 0 4	255,34	45,09
BLNALC – b	4 7 7 6 5 5 13 13 13 5 2 2 0 2 5 6 2 2 6 12	275,92	754,95
BLNALC – c	4 7 7 6 5 5 13 13 13 5 2 2 0 2 5 6 2 2 6 12	275,92	880,45
BLNALC – d	3 6 6 6 7 5 15 17 13 6 1 1 0 1 1 3 0 5 11 20	219,39	1132
BLNALC - e	4 5 6 6 6 6 15 15 15 6 0 1 0 1 1 1 0 2 3 6 26	234,33	1590
BLNALC/BA – a	4 6 6 6 5 5 15 16 14 6 0 2 0 1 4 6 0 0 5 10	174,68	210,15
BLNALC/BA – b	4 5 7 6 5 6 13 13 14 5 1 3 0 2 3 2 5 8 8 14	202,44	249,36
BLNALC/BA – c	4 5 7 6 5 6 13 14 14 5 0 4 0 1 4 2 5 5 6 16	200,59	333,40
BLNALC/BA – d	4 6 6 6 5 5 15 16 14 6 0 2 0 1 4 6 0 0 5 10	174,68	272,05
BLNALC/BA - e	4 6 6 6 5 5 15 16 14 6 0 2 0 1 4 6 0 0 5 10	174,68	272,48

QUADRO 17: Resultados obtidos para o cenário Dados_R1_gera5

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 16 19 18 7 0 17 0 6 5 6 0 0 0 2 3 4 0 0 0 0 0 0 0 1	1408,3	0,0001
Heurística proposta	4 5 6 6 6 6 15 17 16 12	992,74	0,11
MINLP	4 5 10 0 0 0 4 6 2 0 3 3 4 7 6 7 12 10 15 7 1 9 0 0 1 0 0 3 0 4	182,04	Não aplicável
BVLC	4 5 7 6 5 6 13 14 14 4 1 4 0 2 5 4 2 4 4 11	341,03	1897
BA – a	3 5 8 6 5 6 14 18 17 11 2 1 0 0 2 0 0 1 2 1	1093,74	11,48
BA-b	4 5 7 6 5 6 13 14 14 4 1 4 0 3 5 4 3 4 4 11	383,94	28,67
BA – c	4 5 7 6 5 6 13 14 14 4 1 4 0 3 5 4 3 4 4 11	383,94	19,77
BA – d	4 5 7 6 5 6 13 14 13 4 1 4 0 3 5 4 2 4 5 11	330,52	23,34
BA - e	4 5 7 6 5 6 13 14 13 4 1 4 0 3 5 4 3 4 5 11	372,84	17,96
BVLC/BA	4 5 7 6 5 6 13 15 13 4 1 4 0 3 6 5 2 3 4 9	311,10	3597
BLNALC – a	4 5 6 6 6 6 14 14 16 8 0 8 1 0 0 0 2 3 0 2	458,32	2979
BLNALC – b	3 6 7 7 5 5 11 12 11 6 4 4 3 2 8 10 4 5 5 5	879,23	868,81
BLNALC – c	3 6 7 7 5 5 11 12 11 6 4 4 3 2 8 10 4 5 5 5	879,23	934,28
BLNALC – d	3 6 6 6 7 5 14 14 13 4 1 1 0 1 1 3 0 12 11 30	1038,9	885,84
BLNALC - e	3 4 7 6 6 6 13 13 14 4 1 2 0 1 1 2 4 19 7 22	333,18	1207,3
BLNALC/BA – a	4 5 6 6 6 6 15 15 15 9 0 1 0 0 0 2 0 10 3 11	611,31	218,5
BLNALC/BA – b	4 5 7 6 5 6 13 13 13 4 2 4 0 3 6 5 2 5 4 9	340,86	238,27
BLNALC/BA – c	4 5 7 6 5 6 13 13 13 4 2 4 0 3 6 5 2 5 4 9	340,86	232,06
BLNALC/BA – d	4 5 7 6 5 6 13 13 18 5 2 17 1 4 7 4 1 2 0 4	486,63	272,37
BLNALC/BA - e	3 4 7 6 6 6 13 13 13 4 1 2 0 1 1 2 1 18 11 22	320,11	273,69

QUADRO 18: Resultados obtidos para o cenário Dados_R1_gera6

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 16 19 18 7 0 17 0 6 5 6 0 0 0 2 34 0 0 0 0 0 0 0 0 1	5138	0,0001
Heurística proposta	4 5 6 6 6 6 15 17 16 12	4283	0,11
MINLP	-	-	Atingiu o limite de tempo sem se obter solução viável.
BVLC	-	2203,94	> 10800
BA – a	3 6 7 6 5 6 14 19 18 9 0 1 0 0 0 0 0 0 0 2	4361,87	15,6
BA-b	4 6 5 6 6 6 12 13 10 10 3 4 11 3 5 7 4 3 6 0	3660,70	25,9
BA – c	4 6 5 6 6 6 12 13 10 10 3 4 11 3 5 7 4 3 6 0	3660,70	25,5
BA – d	4 6 5 6 6 6 11 12 13 10 2 5 12 4 4 6 4 4 4 0	3620,04	23,2
BA - e	4 6 5 6 6 6 11 12 13 10 2 5 12 4 4 6 4 4 4 0	3620,04	24,1
BVLC/BA	3 6 5 5 6 7 10 11 14 9 2 2 3 2 1 0 15 12 5 1	2158,22	3700
BLNALC – a	-	2399,19	> 10800
BLNALC – b	4 7 5 5 6 6 10 13 10 8 1 0 4 3 2 3 10 6 15 2	2515,76	955,2
BLNALC – c	4 7 5 5 6 6 10 13 10 8 1 0 4 3 2 3 10 6 15 2	2515,76	1017,5
BLNALC – d	3 6 5 5 6 5 10 15 11 10 1 0 2 2 1 2 18 14 23 0	2824,45	1413,2
BLNALC - e	3 6 5 5 5 9 15 10 10 1 0 2 2 2 2 21 11 23 0	2372,66	1520
BLNALC/BA – a	3 5 6 6 6 6 12 13 16 8 1 1 2 1 0 0 10 20 1 16	2812,4	172
BLNALC/BA – b	3 6 5 6 6 4 12 16 10 10 1 0 2 1 1 2 15 9 32 4	3125,13	285,3
BLNALC/BA – c	3 6 5 6 6 4 11 15 9 10 1 0 3 1 2 3 13 11 22 3	3055,98	311,3
BLNALC/BA – d	3 4 6 5 5 5 16 17 16 6 3 12 1 4 5 8 0 2 2 4	2650,0	335,2
BLNALC/BA - e	3 4 6 5 5 5 16 17 16 6 3 12 1 4 5 8 0 2 2 4	2650,0	346,9

QUADRO 19: Resultados obtidos para o cenário Dados_R1_gera7

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 24 28 0 0 3 5 5 6 7 6 0 0 28 0 3 8 7 7 6 6 0 0 0 13	317,03	0,0001
Heurística proposta	3 5 6 6 6 6 19 19 17 5 6 7 4 4 5 7 4 5 6 10	422,62	0,11
MINLP	4 5 8 6 6 4 20 19 16 5 2 6 0 4 4 8 0 6 10 10	210,60	Não aplicável
BVLC	-	525,5	> 10800
BA – a	3 5 6 5 6 7 16 18 16 10 5 7 4 5 5 4 9 7 8 5	456,06	16,31
BA-b	3 7 6 6 5 6 16 17 20 7 3 3 3 3 8 4 18 12 6 12	637,68	28,95
BA – c	3 7 6 6 5 6 16 17 20 7 3 3 3 3 8 4 18 12 6 12	637,68	28,34
BA – d	3 7 6 6 5 6 16 17 20 7 4 3 2 2 8 6 23 17 4 16	2522,19	31,31
BA - e	3 7 6 6 5 6 16 17 20 7 4 3 2 2 8 6 23 17 4 16	2522,19	31,14
BVLC/BA	3 6 7 5 6 6 17 17 18 8 4 4 1 3 3 5 12 11 8 11	335,32	6192
BLNALC – a	4 5 7 5 5 7 16 17 18 9 3 7 2 5 6 3 9 6 7 6	395,27	78,65
BLNALC – b	3 7 6 5 6 6 16 18 19 7 6 6 4 5 4 8 9 6 5 7	576,89	934,94
BLNALC – c	3 7 6 5 6 6 16 18 19 7 6 6 4 5 4 8 9 6 5 7	576,89	1032,84
BLNALC – d	3 6 6 7 5 6 14 17 20 9 4 6 3 2 8 6 15 11 4 6	464,65	1248,1
BLNALC - e	4 6 6 6 6 5 17 17 17 9 1 3 2 2 2 4 15 17 17 11	553,89	1724,9
BLNALC/BA – a	4 5 7 5 5 7 16 17 18 9 3 8 2 5 7 3 9 6 6 5	376,35	195,32
BLNALC/BA – b	4 5 7 5 6 6 16 17 21 6 1 3 1 3 2 3 17 17 6 19	284,59	231,86
BLNALC/BA – c	4 5 7 5 6 6 16 17 21 6 1 3 1 3 2 3 17 17 6 19	284,59	271,09
BLNALC/BA – d	3 5 6 6 5 7 15 17 19 9 4 9 4 4 7 2 11 14 6 5	489,17	304,11
BLNALC/BA - e	3 5 6 5 6 7 17 17 18 8 2 4 2 3 3 2 19 19 10 12	406,10	313,61

QUADRO 20: Resultados obtidos para o cenário Dados_R1_gera8

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 24 8 0 0 3 5 5 6 7 6 0 0 28 0 3 8 3 7 6 6 0 0 0 13	1367	0,0001
Heurística proposta	3 5 6 6 6 6 19 19 17 5 5 7 4 5 5 7 4 6 6 9	932,08	0,11
MINLP	2 0 9 5 4 5 12 9 15 0 4 8 3 6 6 6 10 17 12 12	212,55	Não aplicável
BVLC	-	1272,44	> 10800
BA – a	3 5 6 5 6 7 16 18 16 10 5 7 4 5 5 4 9 7 8 5	1366,05	14,16
BA-b	4 6 6 6 5 6 17 18 17 8 2 3 3 2 5 5 9 10 10 5	1415,99	22,92
BA – c	4 6 6 6 5 6 17 18 17 8 2 3 3 2 5 5 9 10 10 5	1415,99	15,66
BA – d	4 6 6 6 5 6 18 18 17 7 1 3 2 2 5 5 15 16 10 19	2602,25	16,32
BA - e	4 6 6 6 5 6 18 18 17 7 1 3 2 2 5 5 15 16 10 19	2602,25	15,81
BVLC/BA	3 6 7 6 5 6 15 19 19 7 0 13 2 0 11 6 3 1 1 4 4 0 2 6 3 2 9 7 12 5	756,09	3776,7
BLNALC – a	4 5 7 5 5 7 16 17 18 9 3 8 3 7 8 3 6 6 6 5	917,92	219,57
BLNALC – b	3 6 7 5 6 6 19 17 17 7 5 5 0 5 2 14 3 5 10 8 2 2 3 3 6 0 3 5 2 2	1857,54	994,83
BLNALC – c	3 6 7 5 6 6 19 17 17 7 5 5 0 5 2 14 3 5 10 8 2 2 3 3 6 0 3 5 2 2	1857,54	1169,43
BLNALC – d	4 6 6 5 6 6 16 18 19 7 2 7 5 6 4 8 7 7 5 7	1378,38	385,31
BLNALC – e	4 6 5 5 7 6 18 19 15 8 1 2 4 3 2 4 13 10 15 22	1974,93	544,87
BLNALC/BA – a	4 5 7 5 5 7 16 17 18 9 3 8 3 7 7 3 6 6 6 5	915,32	262,33
BLNALC/BA – b	3 6 7 6 5 6 16 19 17 8 2 1 1 3 0 0 13 11 9 8	1380,38	307,33
BLNALC/BA – c	3 6 7 6 5 6 16 19 17 8 2 1 1 3 0 0 13 11 9 8	1380,38	325,35
BLNALC/BA – d	4 5 7 6 5 6 15 17 21 7 2 3 2 2 4 3 20 14 7 19	1472,4	350,78
BLNALC/BA – e	4 5 6 7 6 5 16 17 19 8 2 10 5 0 4 11 5 7 4 4 1 1 1 3 1 1 8 3 5 6	1143,0	365,2

QUADRO 21: Resultados obtidos para o cenário Dados_R1_gera9

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 5 6 6 6 6 24 28 0 0 3 5 5 6 7 6 0 0 28 0 3 8 3 7 6 6 0 0 0 13	6737	0,0001
Heurística proposta	3 5 6 6 6 6 19 19 17 5 5 7 4 5 5 7 4 6 6 9	4382,1	0,11
MINLP	-	-	Ultrapassou o limite máximo de tempo sem chegar na solução.
BVLC	3 6 5 5 6 7 18 18 16 8 3 3 4 3 2 2 10 11 18 15	532,77	6312
BA – a	3 5 6 5 6 7 16 18 16 10 5 7 4 5 4 4 9 7 9 5	4824,92	21
BA-b	3 6 5 5 6 7 18 18 16 8 3 2 4 3 3 2 10 11 16 9	5764,7	23
BA – c	3 6 5 5 6 7 18 18 16 8 3 2 4 3 3 2 10 11 16 9	5764,7	24,5
BA – d	4 6 6 5 5 7 17 16 19 8 1 4 3 5 5 1 9 21 11 14	7004,04	20
BA – e	4 6 6 5 5 7 17 16 19 8 1 4 3 5 5 1 9 21 11 14	7004,04	26
BVLC/BA	4 6 6 5 6 6 17 16 18 9 4 6 6 7 4 6 5 6 6 6	4451,84	2274,9
BLNALC – a	4 5 7 5 5 7 16 17 18 9 3 8 3 7 7 3 6 6 6 5	3995,3	94
BLNALC – b	3 5 6 6 5 7 15 17 19 9 6 9 8 4 9 0 5 6 3 5 0 0 0 3 2 4 3 0 5 2	3919,32	300,2
BLNALC – c	3 5 6 6 5 7 15 17 19 9 6 9 8 4 9 0 5 6 3 5 0 0 0 3 2 4 3 0 5 2	3919,32	353,5
BLNALC – d	3 5 6 5 6 7 16 18 16 10 2 3 3 3 3 2 13 16 16 5	5575,94	912,5
BLNALC – e	3 5 6 5 6 7 17 17 17 9 2 3 3 3 3 2 13 20 13 14	5525,49	413,25
BLNALC/BA – a	4 5 7 5 5 7 16 17 18 9 3 8 3 7 7 3 6 6 6 5	3995,32	478,23
BLNALC/BA – b	4 6 6 5 5 7 14 21 11 14 1 1 4 4 3 2 13 1 34 1	4310,9	340,1
BLNALC/BA – c	4 6 6 5 5 7 12 20 15 11 2 4 6 8 6 6 10 3 10 5	4888,3	346,9
BLNALC/BA – d	4 5 6 6 6 6 16 19 19 6 1 3 3 2 2 3 11 14 12 23	4993,36	494,77
BLNALC/BA - e	4 5 6 6 6 6 16 19 19 6 1 3 3 2 2 3 11 14 12 23	4993,36	494,2

QUADRO 22: Resultados obtidos para o cenário Dados_R2_gera1

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 36 41 24	838,19	0,0001
Heurística proposta	2 3 3 3 3 2 26 33 27	233,03	0,11
MINLP	2 3 3 3 3 2 25 35 27	208,11	Não aplicável
BVLC	2 2 3 3 3 3 23 32 29 0 2 2 0 0 0 2 0 2	167,97	64,32
BA – a	2 3 4 3 3 2 24 33 27 0 0 0 0 0 1 0 1 0 0	194,21	8,73
BA-b	2 2 3 3 3 3 23 32 29 0 2 2 0 0 0 2 0 2	167,97	15,27
BA – c	2 2 3 3 3 3 23 32 29 0 2 2 0 0 0 2 0 2	167,97	14,45
BA – d	2 2 3 3 3 3 23 32 29 0 2 2 0 0 0 2 0 2	167,97	14,83
BA – e	2 2 3 3 3 3 23 32 29 0 2 2 0 0 0 2 0 2	167,97	14,39
BVLC/BA	2 2 3 3 3 3 23 32 19 0 2 2 0 0 0 2 0 1 2	157,89	1270,7
BLNALC – a	2 3 4 3 3 2 23 32 27 0 0 0 0 0 1 3 1 0	165,60	223,17
BLNALC – b	2 3 4 3 3 2 23 32 21 0 0 0 0 0 2 2 1 6	159,54	188,56
BLNALC – c	2 3 4 3 3 2 23 32 21 0 0 0 0 0 2 2 1 6	159,54	199,26
BLNALC – d	2 2 3 3 3 3 23 32 26 0 1 1 0 0 0 25 8 19	220,87	1102,1
BLNALC – e	1 2 3 2 2 2 32 39 29 3 1 0 0 1 0 4 12 4	189,69	1176,5
BLNALC/BA – a	2 3 3 3 3 2 23 32 26 0 0 1 0 0 1 4 1 2	159,27	216,63
BLNALC/BA – b	2 3 3 3 3 2 23 32 26 0 0 1 0 0 1 4 1 2	159,27	229,54
BLNALC/BA – c	2 3 3 3 3 2 23 32 26 0 0 1 0 0 1 4 1 2	159,27	244,92
BLNALC/BA – d	2 3 3 3 3 2 23 32 26 0 0 1 0 0 1 4 1 2	159,27	331,97
BLNALC/BA – e	2 3 3 3 3 2 23 32 26 0 0 1 0 0 1 4 1 2	159,27	345,26

QUADRO 23: Resultados obtidos para o cenário Dados_R2_gera2

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 36 41 24	3087,5	0,0001
Heurística proposta	2 3 3 3 3 2 26 33 27	726,77	0,11
MINLP	8 2 0 1 2 0 3 8 4 0 3 4 3 3 3 21 22 19	157,22	Não aplicável
BVLC	-	315,09	> 10800
BA – a	2 3 4 3 3 2 24 32 27 0 0 0 0 0 1 1 1 0	714,03	9,12
BA-b	2 2 3 3 3 3 23 32 22 0 2 1 0 0 0 6 0 9	315,09	14,78
BA – c	2 2 3 3 3 3 23 32 22 0 2 1 0 0 0 6 0 9	315,09	17,52
BA – d	2 2 3 3 3 3 23 32 22 0 1 1 0 0 0 6 0 18	321,65	12,96
BA – e	2 2 3 3 3 3 23 32 22 0 2 1 0 0 0 6 0 9	315,09	16,59
BVLC/BA	2 3 3 3 3 2 18 32 18 0 1 2 1 1 2 14 1 10	186,55	1656,4
BLNALC – a	2 3 4 3 3 2 21 31 22 0 0 0 1 2 5 3 1 2	373,22	2900
BLNALC – b	2 3 4 3 3 2 18 32 21 0 0 1 1 1 3 14 1 4	327,08	472,03
BLNALC – c	2 3 4 3 3 2 18 32 21 0 0 1 1 1 3 14 1 4	327,08	480,2
BLNALC – d	2 3 3 3 3 2 23 32 22 0 0 2 0 0 2 11 1 8	327,11	928,03
BLNALC – e	1 2 3 2 2 2 32 39 29 3 2 0 0 4 0 1 12 2	1946,88	1564,6
BLNALC/BA – a	2 3 3 3 3 2 17 32 20 0 1 2 1 1 2 17 1 7	189,31	234,2
BLNALC/BA – b	2 3 3 3 3 2 17 32 20 0 1 2 1 1 2 17 1 7	189,31	287,48
BLNALC/BA – c	2 3 3 3 3 2 17 32 20 0 1 2 1 1 2 17 1 7	189,31	310,2
BLNALC/BA – d	1 2 3 2 3 2 29 32 22 3 2 1 3 0 1 2 12 10	189,7	345,1
BLNALC/BA – e	1 2 3 2 2 2 32 39 29 3 2 0 0 4 0 1 12 2	194,69	335,37

QUADRO 24: Resultados obtidos para o cenário Dados_R2_gera3

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 3 6 4 1 24	8038,6	0,0001
Heurística proposta	2 3 3 3 3 2 26 33 27	3497,2	0,11
MINLP	-	-	Excedeu o limite de tempo do NEOS sem obter solução.
BVLC	-	2078,83	> 10800
BA – a	2 3 4 3 3 2 24 32 27 0 0 0 0 0 1 1 1 0	4101,6	9,1
BA-b	1 3 4 3 3 2 20 27 17 4 2 1 1 2 3 4 10 7	2577,78	20,01
BA – c	1 3 4 3 3 2 20 27 17 4 2 1 1 2 3 4 10 7	2577,78	12,09
BA – d	2 3 4 3 3 2 19 21 21 1 0 0 1 2 3 9 22 4	2403,29	14,52
BA – e	2 3 4 3 3 2 19 21 21 1 0 0 1 2 3 9 22 4	2403,29	11,07
BVLC/BA	1 2 3 3 3 2 24 28 18 4 3 2 0 0 2 1 9 9	1272,52	1576,18
BLNALC – a	2 3 4 3 3 2 23 28 23 2 0 0 0 0 4 1 4 2	2589,45	45,03
BLNALC – b	1 2 3 3 3 2 23 28 17 3 2 2 0 0 2 2 12 14	1457,45	802,5
BLNALC – c	1 2 3 3 3 2 23 28 17 3 2 2 0 0 2 2 12 14	1457,45	973,56
BLNALC – d	1 2 3 2 2 2 25 33 31 1 0 0 1 2 0 15 31 1	2185,4	1354,68
BLNALC – e	1 2 2 2 2 2 25 36 31 1 0 1 1 1 0 25 28 10	2176,79	1902,33
BLNALC/BA – a	2 3 4 3 3 2 19 20 19 3 1 0 2 3 5 4 10 5	1912,66	191,25
BLNALC/BA – b	1 2 3 3 3 2 25 27 24 2 1 0 0 0 1 0 19 14	1289,5	230,25
BLNALC/BA – c	1 2 3 3 3 2 25 27 24 2 1 0 0 0 1 0 19 14	1289,5	243,1
BLNALC/BA – d	1 2 3 3 3 2 25 27 24 2 1 0 0 0 1 0 19 14	1289,5	299,5
BLNALC/BA - e	1 2 3 3 3 2 25 27 19 5 4 2 0 0 2 0 8 8	1328,89	311,06

QUADRO 25: Resultados obtidos para o cenário Dados_R2_gera4

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 2 4 3 3 3 3 6 0 0 2 2 2 2 3 2 0 5 9 0 2 3 3 2 2 2 0 0 4 3	269,33	0,0001
Heurística proposta	2 3 4 3 3 2 2 9 3 9 3 2 2 1 0 1 3 6 1 6 5 3 1 5 0 1 0 0 3 1 1 1	706,51	0,11
MINLP	2 2 3 3 4 2 3 0 4 9 1 7 1 4 4 2 0 3 7 4 1 7	213,54	Não aplicável
BVLC	2 2 3 3 3 3 3 4 3 3 3 3 2 3 5 1 2 1 3 1 8 1 3	240,37	130,5
BA – a	2 3 4 3 3 2 2 9 4 0 3 1 4 0 0 2 4 5 8 6 4 0 2 1 0 0 1 1 0 3	458,70	11,4
BA-b	2 2 3 3 3 3 3 4 3 3 3 3 2 3 4 1 2 1 4 1 8 1 3	263,41	28,31
BA – c	2 2 3 3 3 3 3 4 3 3 3 3 2 3 4 1 2 1 4 1 8 1 3	263,41	31,4
BA – d	2 2 3 3 3 3 3 4 3 3 3 3 2 2 5 1 2 1 3 1 9 1 7	247,52	25,31
BA – e	2 2 3 3 3 3 3 4 3 3 3 3 2 3 5 1 2 1 3 1 8 1 3	240,37	23,30
BVLC/BA	2 3 4 3 3 2 2 9 4 0 3 1 2 1 1 2 3 3 8 1 1 8	225,41	4435,9
BLNALC – a	2 3 4 3 3 2 2 9 4 0 3 1 3 0 0 3 4 5 8 5 0 0 2 1 1 1 0 1 0 3	243,19	120,31
BLNALC – b	2 3 4 3 3 2 3 1 3 9 3 0 3 1 0 3 4 4 4 8 9	226,63	480,06
BLNALC – c	2 3 4 3 3 2 3 1 3 9 3 0 3 1 0 3 4 4 4 8 9	226,63	514,1
BLNALC – d	2 2 2 3 3 2 3 5 3 7 2 8 1 3 6 1 1 2 9 2 8 1 7	329,10	1408,5
BLNALC – e	2 2 3 3 3 3 3 0 4 0 3 0 1 3 2 1 2 0 1 7 2 2 1 7	297,44	1257,0
BLNALC/BA – a	2 3 4 3 3 2 2 9 4 0 3 1 2 1 1 2 3 3 8 1 1 8	225,41	206,03
BLNALC/BA – b	2 3 3 3 3 2 3 1 3 9 3 0 1 1 2 1 2 2 1 2 2 4 1 3	242,39	292,10
BLNALC/BA – c	2 3 3 3 3 2 3 1 3 9 3 0 1 1 2 1 2 2 1 2 2 4 1 3	242,39	328,57
BLNALC/BA – d	2 2 3 3 3 3 3 2 3 8 3 0 2 3 5 2 3 0 5 1 3 1 3	235,48	347,34
BLNALC/BA – e	2 2 3 3 3 3 3 2 3 8 3 0 2 3 5 2 3 0 5 1 3 1 3	235,48	353,47

QUADRO 26: Resultados obtidos para o cenário Dados_R2_gera5

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 2 4 3 3 3 36 0 0 2 2 2 2 3 2 0 59 0 2 3 3 2 2 2 0 0 43	529,92	0,0001
Heurística proposta	2 3 4 3 3 2 29 39 32 1 0 0 1 3 6 16 5 4 19 3 1 0 0 0 1 1 1	1227	0,11
MINLP	0 3 4 3 3 3 30 31 29 9 2 0 3 3 1 6 12 6	206,20	Não aplicável
BVLC	-	423,31	> 10800
BA – a	2 3 4 3 3 2 29 40 31 4 0 0 2 4 5 8 6 4 0 2 1 0 0 1 1 0 3	774,15	12,47
BA-b	2 3 4 3 3 2 27 42 31 1 1 1 2 3 4 11 18 6	464,40	29,55
BA – c	2 3 4 3 3 2 27 42 31 1 1 1 2 3 4 11 18 6	464,40	28,78
BA – d	2 3 4 3 3 2 27 42 31 1 1 1 2 2 4 14 18 6	423,31	27,9
BA – e	2 3 4 3 3 2 27 42 31 1 1 1 2 2 4 14 18 6	423,31	20,65
BVLC/BA	2 3 4 3 3 2 27 44 29 1 2 1 3 3 5 9 14 6	358,60	3855,3
BLNALC – a	2 3 4 3 3 2 29 40 31 2 2 0 2 3 6 8 11 4 0 0 1 0 0 0 1 0 0	501,41	2898,5
BLNALC – b	2 3 4 3 3 2 27 44 29 1 1 0 3 4 5 6 14 4 0 3 3 1 1 3 4 0 3	360,20	973,39
BLNALC – c	2 3 4 3 3 2 27 44 29 1 1 0 3 4 5 6 14 4 0 3 3 1 1 3 4 0 3	360,20	1137,3
BLNALC – d	2 3 3 3 3 2 27 41 32 1 1 2 1 1 3 27 20 9	695,09	1740,9
BLNALC – e	2 3 3 3 3 2 27 41 32 1 1 2 1 1 3 27 20 9	695,09	2015,2
BLNALC/BA – a	2 2 3 4 2 2 27 42 31 1 1 2 0 2 1 31 23 37	1341,79	2361,4
BLNALC/BA – b	2 3 4 3 3 2 28 41 31 1 1 1 2 2 4 12 20 6	423,31	247,17
BLNALC/BA – c	2 3 4 3 3 2 28 41 31 1 1 1 2 2 4 12 20 6	423,31	485,26
BLNALC/BA – d	2 3 4 3 3 2 28 41 31 1 1 1 2 2 4 12 20 6	423,31	513,56
BLNALC/BA – e	2 3 3 3 3 2 27 42 31 1 1 4 2 2 4 14 18 6	524,86	567,55

QUADRO 27: Resultados obtidos para o cenário Dados_R2_gera6

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 2 4 3 3 3 3 6 0 0 2 2 2 2 3 2 0 5 9 0 2 3 3 2 2 2 0 0 4 3	2989,0	0,0001
Heurística proposta	2 3 4 3 3 2 2 9 3 9 3 2 1 0 0 1 3 6 1 6 5 4 1 9 3 1 0 0 0 1 1 1	9408,3	0,11
MINLP	-	-	Excedeu o limite de tempo do Neos sem se obter resultado.
BVLC	-	4204,01	> 10800
BA – a	2 3 4 3 3 2 2 9 4 0 3 1 4 0 0 2 4 5 8 6 4 0 2 1 0 0 1 1 0 3	3023,4	11,07
BA-b	1 2 4 3 4 2 2 9 4 3 2 8 4 3 1 2 0 3 8 1 5 1 5	4314,5	20,91
BA – c	1 2 4 3 4 2 2 9 4 3 2 8 4 3 1 2 0 3 8 1 5 1 5	4314,5	18,84
BA – d	2 3 3 3 3 2 3 1 4 2 2 7 1 1 2 1 1 2 1 5 1 8 1 8	3354,1	22,2
BA – e	2 3 4 3 3 2 3 0 3 3 3 3 1 0 0 1 2 4 1 5 3 6 5	4353,2	17,52
BVLC/BA	2 2 3 3 3 2 3 4 4 3 2 3 2 4 3 1 1 3 6 8 1 5	2092,93	3040,4
BLNALC – a	2 3 4 3 3 2 2 9 4 0 3 1 4 1 0 2 3 5 8 6 5 0 1 1 0 0 0 1 0 1	3222,99	153,3
BLNALC – b	1 2 3 3 3 3 2 9 4 3 2 8 4 3 2 1 2 1 1 6 1 5 1 8	4863,58	1748,84
BLNALC – c	1 2 3 3 3 3 2 9 4 3 2 8 4 3 2 1 2 1 1 6 1 5 1 8	4863,58	2043
BLNALC – d	1 2 3 2 3 2 3 2 4 2 2 6 3 2 2 3 1 2 1 3 2 2 2 4	3969,2	1594,23
BLNALC – e	1 2 3 3 2 2 3 1 4 2 2 7 3 2 2 1 4 2 1 3 2 0 2 8	5066,06	1852,11
BLNALC/BA – a	2 3 4 3 3 2 2 9 4 0 3 1 4 0 0 2 4 5 6 6 4 0 2 1 2 0 1 2 0 3	2902,73	304,44
BLNALC/BA – b	1 2 3 3 3 2 3 3 4 2 2 5 2 1 1 1 1 1 1 8 3 0 4 9	4135,1	516,4
BLNALC/BA – c	1 2 3 3 3 2 3 3 4 2 2 5 2 1 1 1 1 1 1 8 3 0 4 9	4135,1	521,9
BLNALC/BA – d	1 2 3 3 3 2 3 3 4 2 2 5 2 1 1 1 1 1 1 8 3 0 4 9	4135,1	490,74
BLNALC/BA – e	1 2 3 3 3 2 3 3 4 2 2 5 2 1 1 1 1 1 1 8 3 0 4 9	4135,1	483,96

QUADRO 28: Resultados obtidos para o cenário Dados_R2_gera7

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 86 0 0 1 3 3 3 3 0 100 0 2 3 4 3 2 3 0 35 55	5003,7	0,0001
Heurística proposta	2 2 3 3 3 2 28 37 35 1 4 2 2 2 3 26 58 16 3 2 5 2 3 1 9 7 9	407,06	0,11
MINLP	3 2 2 3 3 3 13 46 18 1 3 4 3 3 2 41 20 37	266,31	Não aplicável
BVLC	-	370,53	> 10800
BA – a	2 3 4 3 3 2 30 40 30 3 1 0 2 4 5 12 17 14 3 5 5 3 3 0 13 8 10	671,81	19
BA-b	2 3 4 3 3 2 31 40 29 2 3 2 3 3 3 26 37 27	370,53	21,42
BA – c	2 3 4 3 3 2 31 40 29 2 3 2 3 3 3 26 37 27	370,53	31,75
BA – d	2 3 4 3 3 2 31 40 29 2 2 3 3 3 3 24 37 30	317,44	22,95
BA – e	2 3 4 3 3 2 31 40 29 2 2 3 3 3 3 24 37 30	317,44	22,96
BVLC/BA	2 3 3 3 4 2 33 37 30 2 2 4 3 2 3 19 40 29	312,12	3800
BLNALC – a	-	526,48	>10800
BLNALC – b	2 3 4 3 3 2 28 43 29 2 2 3 3 3 3 26 34 30	315,79	1052,8
BLNALC – c	2 3 4 3 3 2 28 43 29 2 2 3 3 3 3 26 34 30	315,79	1235,7
BLNALC – d	3 4 2 2 3 3 30 41 29 1 2 5 4 2 2 37 37 26	575,04	2302,8
BLNALC – e	2 3 3 4 3 2 31 38 31 3 2 4 2 3 3 24 37 39	833,88	2267
BLNALC/BA – a	2 3 4 3 3 2 30 41 29 2 0 0 2 4 6 12 26 12 2 4 5 3 2 0 13 10 15	339,96	315,77
BLNALC/BA – b	2 3 4 3 3 2 32 39 29 2 2 3 3 3 3 23 38 32	322,71	608,32
BLNALC/BA – c	2 3 4 3 3 2 32 39 29 2 2 3 3 3 3 23 38 32	322,71	683,11
BLNALC/BA – d	2 3 4 3 3 2 32 39 29 2 2 3 3 3 3 23 38 32	322,71	663,34
BLNALC/BA - e	2 3 4 3 3 2 32 39 29 2 2 3 3 3 3 23 38 32	322,71	679,54

QUADRO 29: Resultados obtidos para o cenário Dados_R2_gera8

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 86 0 0 1 3 3 3 3 3 0 100 0 2 3 4 3 2 3 0 35 55	13329,0	0,0001
Heurística proposta	2 2 3 3 3 2 28 37 35 1 4 2 2 2 3 26 58 16 3 2 5 2 3 1 9 7 9	2039,1	0,11
MINLP	1 3 0 1 1 1 0 39 54 2 1 3 3 4 2 32 57 11 0 2 6 2 1 2 34 1 25	385,39	Não aplicável
BVLC	-	1448,8	> 10800
BA – a	2 3 4 3 3 2 30 40 30 3 1 0 2 4 5 13 17 14 3 5 5 3 3 0 13 8 11	1539,1	14,37
BA-b	2 3 4 3 3 2 32 39 29 2 2 2 3 4 3 24 38 31	1448,8	20,01
BA – c	2 3 4 3 3 2 32 39 29 2 2 2 3 4 3 24 38 31	1448,8	18,24
BA – d	2 3 4 3 3 2 32 39 29 2 2 2 3 4 3 24 38 31	1448,8	13,53
BA – e	2 3 4 3 3 2 32 39 29 2 2 2 3 4 3 24 38 31	1448,8	23,4
BVLC/BA	2 3 4 3 3 2 32 39 29 2 3 2 3 3 3 24 38 28	932,48	1908,88
BLNALC – a	2 3 4 3 3 2 30 41 29 2 0 0 2 4 6 18 19 8 3 5 6 3 0 2 10 11 13	911,12	70,11
BLNALC – b	2 3 4 3 3 2 27 38 35 2 1 3 3 4 3 24 37 39 1 2 0 1 0 0 1 3 2	2743,8	1125,24
BLNALC – c	2 3 4 3 3 2 27 38 35 2 1 3 3 4 3 24 37 39 1 2 0 1 0 0 1 3 2	2743,8	1340,4
BLNALC – d	2 3 3 3 4 2 32 38 30 2 2 5 3 2 3 19 50 31 0 0 0 1 0 0 1 0 0	2025,5	1784,1
BLNALC – e	2 3 3 3 4 2 30 41 29 2 3 4 3 2 3 23 52 25	1837,8	1778,82
BLNALC/BA – a	2 3 4 3 3 2 30 41 29 2 0 0 2 4 6 18 29 12 1 4 9 5 0 0 7 13 16	636,05	296,34
BLNALC/BA – b	2 3 4 3 3 2 29 39 32 2 2 3 3 3 3 25 38 26	874,36	456,18
BLNALC/BA – c	2 3 4 3 3 2 29 39 32 2 2 3 3 3 3 25 38 26	874,36	452,9
BLNALC/BA – d	2 3 3 3 4 2 30 41 29 2 2 4 3 2 3 24 43 33	1136,75	474,24
BLNALC/BA - e	2 3 3 3 4 2 30 41 29 2 2 4 3 2 3 24 43 33	1136,75	492,04

QUADRO 30: Resultados obtidos para o cenário Dados_R2_gera9

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	2 3 4 2 3 3 86 0 0 1 3 3 3 3 3 0 100 0 2 3 4 3 2 3 0 35 55	25864	0,0001
Heurística proposta	2 2 3 3 3 2 28 37 35 1 4 2 2 2 3 26 58 16 3 2 5 2 3 1 9 7 9	9115,7	0,11
MINLP	0 3 2 1 0 2 0 34 39 3 1 2 2 3 1 0 78 22 1 2 4 3 3 3 54 3 16	3780,3	Não aplicável
BVLC	-	3806,79	> 10800
BA – a	2 3 4 3 3 2 30 40 30 3 1 0 2 4 5 13 17 14 3 5 5 3 3 0 13 8 11	4529,02	15,51
BA-b	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 33 5 1 4 3 2 0 3 13 9 19	5185,7	20,07
BA – c	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 33 5 1 4 3 2 0 3 13 9 19	5185,7	27,18
BA – d	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 38 5 0 2 2 1 0 2 22 16 25	5416,7	23,01
BA – e	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 38 5 0 2 2 1 0 2 22 16 25	5416,7	24,24
BVLC/BA	2 3 4 3 3 2 28 39 33 3 0 2 3 6 2 13 25 7 0 3 2 2 0 3 20 11 17	3827,92	3229,4
BLNALC – a	-	4669,03	> 10800
BLNALC – b	2 3 4 3 3 2 28 39 33 2 3 2 2 3 4 35 38 20	7113,67	1152,84
BLNALC – c	2 3 4 3 3 2 28 39 33 2 3 2 2 3 4 35 38 20	7113,67	1340,61
BLNALC – d	3 3 2 4 3 2 28 42 30 1 3 5 1 2 3 43 34 23	7533,52	1682,67
BLNALC – e	2 3 3 4 3 2 28 41 31 2 3 4 1 2 4 4 4 36 20	8890,83	1710,93
BLNALC/BA – a	2 3 4 3 3 2 30 41 29 2 0 0 2 4 6 18 22 8 3 5 6 3 0 2 10 9 13	4559,23	312,3
BLNALC/BA – b	2 3 4 3 3 2 28 38 34 2 0 0 2 10 1 8 19 0 2 3 4 4 0 4 16 20 16	4599,23	476,52
BLNALC/BA – c	2 3 4 3 3 2 28 38 34 2 0 0 2 10 1 8 19 0 2 3 4 4 0 4 16 20 16	4599,23	469,58
BLNALC/BA – d	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 38 5 0 2 2 1 0 2 22 16 25	5416,7	511,28
BLNALC/BA - e	2 3 4 3 3 2 28 39 33 2 0 2 3 5 3 16 38 5 0 2 2 1 0 2 22 16 25	5416,7	510,56

QUADRO 31: Resultados obtidos para o cenário Dados_R3_gera1

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 19 13 16	313,30	0,0001
Heurística proposta	5 6 7 6 6 5 14 13 14	150,89	0,11
MINLP	4 5 6 4 6 4 19 15 16	147,14	Não aplicável
BVLC	4 7 7 6 7 5 18 12 14	186,93	36,97
BA – a	5 7 8 6 7 4 14 14 16	379,27	7,86
BA-b	5 7 8 6 7 4 14 14 16	379,27	8,13
BA – c	5 7 8 6 7 4 14 14 16	379,27	7,63
BA – d	5 7 7 6 7 4 14 14 16	242,42	6,7
BA – e	5 7 8 6 7 4 14 14 16	379,27	6,65
BVLC/BA	5 6 8 5 7 5 15 12 13	152,74	171,81
BLNALC – a	5 6 7 5 6 5 15 13 14	149,71	19,17
BLNALC – b	5 6 7 5 6 5 15 13 14	149,71	77,28
BLNALC – c	5 6 7 5 6 5 15 13 14	149,71	85,63
BLNALC – d	5 6 7 6 7 5 14 12 14	153,07	238,48
BLNALC – e	5 6 7 6 7 5 14 12 14	153,07	38,73
BLNALC/BA – a	5 6 7 5 6 5 15 13 14	149,71	141,21
BLNALC/BA – b	5 6 7 5 6 5 15 13 14	149,71	148,91
BLNALC/BA – c	5 6 7 5 6 5 15 13 14	149,71	147,8
BLNALC/BA – d	5 6 7 5 6 5 15 13 14	149,71	208,55
BLNALC/BA - e	5 6 7 5 6 5 15 13 14	149,71	195,86

QUADRO 32: Resultados obtidos para o cenário Dados_R3_gera2

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 19 13 16	1761,3	0,0001
Heurística proposta	5 6 7 6 6 5 14 13 14	561,85	0,11
MINLP	5 6 4 5 6 5 15 13 11 0 0 5 0 0 1 0 0 1 1	161,89	Não aplicável
BVLC	4 7 7 6 7 5 13 10 14 3 1 0 0 1 1 6 8 0	216,75	108,16
BA – a	5 7 8 6 7 4 14 14 16	1411,37	9,32
BA-b	4 7 7 6 7 5 13 11 14 2 0 0 0 1 1 9 1 0	280,10	15,65
BA – c	4 7 7 6 7 5 13 11 14 2 0 0 0 1 1 9 1 0	280,10	14,75
BA – d	4 7 7 6 7 5 13 12 14 2 0 0 0 0 1 1 6 2 0	380,87	19,59
BA – e	4 7 7 6 7 5 13 10 14 2 1 0 0 1 1 9 8 0	245,75	19,34
BVLC/BA	5 7 6 6 7 5 14 12 10 0 0 2 0 0 1 0 0 1 8	194,40	1161,5
BLNALC – a	-	321,83	> 10800
BLNALC – b	5 6 6 6 7 5 14 12 11 0 0 3 0 0 1 0 0 1 0	186,11	498,10
BLNALC – c	5 6 6 6 7 5 14 12 11 0 0 3 0 0 1 0 0 1 0	186,11	576,31
BLNALC – d	4 6 6 6 8 5 11 11 13 1 2 2 1 0 1 2 5 3 9	571,09	805,37
BLNALC – e	5 6 6 5 6 4 16 12 13 0 0 1 0 1 2 0 6 2 4	387,54	894,96
BLNALC/BA – a	5 7 8 5 7 5 11 9 12 1 1 0 2 1 1 1 4 1 5 2	289,62	210,11
BLNALC/BA – b	4 7 7 5 7 5 13 10 14 4 1 0 2 1 1 5 8 0	211,47	218,10
BLNALC/BA – c	4 7 7 5 7 5 13 10 14 4 1 0 2 1 1 5 8 0	211,47	243,45
BLNALC/BA – d	5 6 5 5 6 4 16 13 11 0 0 2 0 0 2 0 3 2 1	180,67	284,80
BLNALC/BA – e	5 7 7 6 7 4 14 10 11 0 1 2 0 2 5 0 4 1 1	319,74	302,11

QUADRO 33: Resultados obtidos para o cenário Dados_R3_gera3

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 19 13 16	5142,1	0,0001
Heurística proposta	5 6 7 6 6 5 14 13 14	2616,7	0,11
MINLP	-	-	Excedeu o limite de tempo do NEOS sem obter solução.
BVLC	-	1983,02	> 10800
BA – a	5 7 8 6 7 4 14 14 16	4353,4	5,23
BA-b	4 7 7 6 7 5 10 12 9 2 1 2 1 0 1 17 1 17	2022,1	7,15
BA – c	4 7 7 6 7 5 10 12 9 2 1 2 2 1 0 1 17 1 17	2022,1	12,2
BA – d	6 7 6 6 6 4 13 11 10 0 0 1 0 1 1 10 16 36	1821,37	9,30
BA – e	6 7 6 6 6 4 13 11 10 0 0 1 0 1 1 10 16 36	1821,37	12,3
BVLC/BA	6 7 6 6 7 5 13 11 9 0 0 2 0 1 1 0 1 21	1593,76	1712,32
BLNALC – a	5 6 7 5 6 4 15 13 11 0 0 2 0 0 2 0 2 10	1840,75	1040
BLNALC – b	4 7 7 6 7 5 9 9 9 3 2 3 2 2 2 11 8 11	2357,56	405,42
BLNALC – c	4 7 7 6 7 5 9 9 9 3 2 3 2 2 2 11 8 11	2357,56	484,41
BLNALC – d	7 6 6 7 6 5 11 8 9 0 1 2 0 1 1 12 30 21	2011,12	553,66
BLNALC - e	6 7 5 6 8 4 13 11 9 0 0 1 0 0 1 0 12 51	1603,03	751,87
BLNALC/BA – a	6 8 6 7 6 4 12 9 9 0 0 1 0 1 1 10 24 42	1585,86	350,76
BLNALC/BA – b	6 7 6 6 7 5 13 12 9 0 0 2 0 0 1 0 1 26	1671,34	390,6
BLNALC/BA – c	6 7 6 6 7 5 13 12 9 0 0 2 0 0 1 0 1 26	1671,34	386,67
BLNALC/BA – d	6 7 5 6 6 4 13 11 9 0 0 2 0 1 1 10 16 26	1388,25	410,06
BLNALC/BA - e	6 7 5 6 6 4 13 11 9 0 0 2 0 1 1 10 16 26	1388,25	422,68

QUADRO 34: Resultados obtidos para o cenário Dados_R3_gera4

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 29 30 23	507,63	0,0001
Heurística proposta	5 6 7 6 7 5 21 18 22	186,8	0,11
MINLP	4 5 6 5 6 4 26 22 24	181,73	Não aplicável
BVLC	5 7 7 5 7 5 23 17 21	187	27,8
BA – a	5 6 9 6 7 4 21 22 23	811,49	5,06
BA-b	5 7 7 5 7 5 23 17 21	187	7,01
BA – c	5 7 7 5 7 5 23 17 21	187	3,03
BA – d	5 7 6 6 7 5 21 17 24	284,71	7,48
BA – e	5 7 7 5 7 5 23 17 21	187	5,23
BVLC/BA	5 6 7 6 6 5 21 20 21	182,77	222,9
BLNALC – a	5 6 8 5 7 5 23 18 19	183,17	171,34
BLNALC – b	5 6 8 5 7 5 23 18 19	183,17	168,39
BLNALC – c	5 6 8 5 7 5 23 18 19	183,17	182,38
BLNALC – d	4 6 6 5 5 4 26 24 24	187,99	424,56
BLNALC – e	5 6 8 5 7 5 23 18 19	183,17	256,59
BLNALC/BA – a	5 6 8 5 7 5 23 18 19	183,17	267,90
BLNALC/BA – b	5 6 7 6 6 5 21 20 21	182,77	278,86
BLNALC/BA – c	5 6 8 5 7 5 23 18 20	186,22	282,56
BLNALC/BA – d	5 6 8 5 7 5 23 18 20	186,22	301,24
BLNALC/BA - e	5 6 8 5 7 5 23 18 20	186,22	310,2

QUADRO 35: Resultados obtidos para o cenário Dados_R3_gera5

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 29 30 23	4348,7	0,0001
Heurística proposta	5 6 7 6 7 5 21 18 22	747,20	0,11
MINLP	0 0 7 0 1 3 0 3 12 5 6 4 5 6 4 23 19 16	199,55	Não aplicável
BVLC	4 6 7 6 7 5 19 15 21 3 2 0 0 1 1 9 12 0	231,0	246,12
BA – a	5 6 9 6 7 4 21 21 23 0 0 0 0 0 1 0 1 0	2077,21	12,46
BA-b	4 6 7 6 8 5 20 15 21 3 3 0 0 0 1 9 10 0	562,97	19,72
BA – c	4 6 7 6 8 5 20 15 21 3 3 0 0 0 1 9 10 0	562,97	19,44
BA – d	4 6 6 6 7 5 20 17 20 1 2 2 0 0 0 23 6 12	650,74	25,26
BA – e	4 7 6 5 7 5 20 18 20 1 0 2 1 0 0 23 0 12	578,42	23,61
BVLC/BA	4 6 7 6 6 5 19 15 21 3 2 0 0 3 1 10 10 0	229,39	590,04
BLNALC – a	5 7 8 6 6 5 17 17 19 6 0 0 2 4 0 5 4 0	593,37	171,87
BLNALC – b	5 6 7 6 7 5 21 14 16 0 2 1 0 1 1 1 0 19 31	567,14	1120,4
BLNALC – c	5 6 7 6 7 5 21 14 16 0 2 1 0 1 1 1 0 19 31	567,14	1213,4
BLNALC – d	5 6 5 5 6 5 23 21 17 0 0 2 0 0 1 0 0 5 1	301,17	1364,9
BLNALC – e	5 6 5 5 6 5 23 21 17 0 0 2 0 0 1 0 0 5 1	301,17	1513,9
BLNALC/BA – a	5 7 8 6 6 5 17 13 18 1 1 0 1 2 1 18 20 2	324,43	239,59
BLNALC/BA – b	5 6 6 6 7 5 21 18 16 0 0 2 0 0 1 0 0 35	248,67	322,14
BLNALC/BA – c	5 6 6 6 7 5 21 18 16 0 0 2 0 0 1 0 0 35	248,67	351,2
BLNALC/BA – d	5 6 6 6 7 5 21 18 16 0 0 2 0 0 1 0 0 35	248,67	480,2
BLNALC/BA - e	5 6 6 6 7 5 21 18 16 0 0 2 0 0 1 0 0 35	248,67	510,1

QUADRO 36: Resultados obtidos para o cenário Dados_R3_gera6

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	4 7 7 6 7 5 29 30 23	11055	0,0001
Heurística proposta	5 6 7 6 7 5 21 18 22	4259,23	0,11
MINLP	-	-	Excedeu o limite de tempo do NEOS sem obter solução.
BVLC	-	2468,14	> 10800
BA – a	4 6 8 6 7 5 26 18 19	5047,93	6,04
BA-b	6 8 6 6 6 5 19 15 14 0 0 2 0 1 1 0 27 34	2650,84	10
BA – c	6 8 6 6 6 5 19 15 14 0 0 2 0 1 1 0 27 34	2650,84	13,2
BA – d	6 8 6 6 6 5 19 15 16 0 0 1 0 1 1 0 30 50	3319,12	15,2
BA – e	6 8 6 6 6 5 19 15 16 0 0 1 0 1 1 0 30 50	3319,12	14,82
BVLC/BA	6 7 6 6 7 5 19 17 14 0 0 2 0 0 1 0 0 30	2048,36	3717,2
BLNALC – a	-	3409,64	> 10800
BLNALC – b	4 6 8 5 7 6 13 10 18 4 4 0 4 4 2 13 13 0	2113,65	726,58
BLNALC – c	4 6 8 5 7 6 13 10 18 4 4 0 4 4 2 13 13 0	2113,65	850,13
BLNALC – d	5 6 5 5 6 5 23 20 19 0 0 1 0 0 0 0 0 51	2811,01	874,1
BLNALC – e	6 8 6 6 6 5 19 15 16 0 0 1 0 1 1 0 30 50	3319,12	1102,3
BLNALC/BA – a	5 6 8 5 6 6 16 15 18 1 1 0 1 1 0 32 33 0	2765,3	283,01
BLNALC/BA – b	4 7 7 6 7 5 17 16 13 2 0 1 1 1 1 18 9 52	4044,55	326,45
BLNALC/BA – c	4 7 7 6 7 5 17 16 13 2 0 1 1 1 1 18 9 52	4044,55	352,35
BLNALC/BA – d	5 7 5 5 6 5 21 16 13 0 0 2 1 1 1 9 24 39	2810,85	438,31
BLNALC/BA – e	5 7 5 5 6 5 21 16 13 0 0 2 1 1 1 9 24 39	2810,85	441,52

QUADRO 37: Resultados obtidos para o cenário Dados_R4_gera1

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 38 34 0 3 3 4 2 2 1 1 0 0 39	218,69	0,0001
Heurística proposta	3 3 4 2 1 1 1 21 17 31 2 4 1 1 4 2 1 7 8 6	179,13	0,11
MINLP	-	-	O NEOS não gerou nenhuma solução para este cenário.
BVLC	3 5 4 2 1 1 1 19 15 38 3 1 2 2 3 3 1 7 9 6	184,59	110,62
BA – a	3 4 5 2 1 1 1 18 17 37 3 3 0 2 4 5 2 1 9 7 5	236,14	14,61
BA-b	3 5 4 2 1 1 1 19 15 38 3 1 2 2 3 3 1 7 9 6	184,59	15,98
BA – c	3 5 4 2 1 1 1 19 15 38 3 1 2 2 3 3 1 7 9 6	184,59	16,59
BA – d	3 5 4 2 1 1 1 19 15 38 3 1 2 2 3 2 1 8 9 6	185,52	15,54
BA – e	3 5 4 2 1 1 1 19 15 38 3 1 2 2 4 2 1 8 9 5	211,47	13,24
BVLC/BA	4 3 4 2 2 1 1 17 21 34 1 4 3 2 0 3 1 7 5 1	162,50	1804,6
BLNALC – a	5 4 5 1 1 1 1 20 14 38 0 3 0 6 3 2 1 8 10 6 0 0 1 0 0 0 0 0 1 0	204,33	38,23
BLNALC – b	4 4 5 1 1 1 1 19 21 29 0 0 0 6 3 3 1 6 4 10	168,77	498,73
BLNALC – c	4 4 5 1 1 1 1 19 21 29 0 0 0 6 3 3 1 6 4 10	168,77	605,55
BLNALC – d	3 3 5 2 2 1 1 16 24 32 2 2 1 1 0 2 1 16 6 15	231,69	1408,2
BLNALC – e	3 3 5 2 2 1 1 16 24 32 2 2 1 1 0 2 1 16 6 15	231,69	1615,3
BLNALC/BA – a	4 4 5 1 1 1 1 20 14 38 1 3 0 5 3 2 1 8 10 6 0 0 1 0 0 0 0 0 1 0	196,84	275,9
BLNALC/BA – b	4 3 4 2 2 1 1 18 18 36 0 2 2 1 0 1 1 17 18 0	193,95	389,91
BLNALC/BA – c	4 3 4 2 2 1 1 18 18 36 0 2 2 1 0 1 1 17 18 0	193,95	434,52
BLNALC/BA – d	4 3 4 2 2 1 1 18 18 36 0 2 2 1 0 1 1 17 18 0	193,95	429,73
BLNALC/BA - e	4 3 4 2 2 1 1 18 18 36 0 2 2 1 0 1 1 17 18 0	193,95	440,37

QUADRO 38: Resultados obtidos para o cenário Dados_R4_gera2

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 38 34 0 3 3 4 2 2 1 1 0 0 39	574,18	0,0001
Heurística proposta	3 3 4 2 1 1 1 21 17 31 2 4 1 1 4 2 1 7 8 6	324,97	0,11
MINLP	0 7 3 1 0 3 2 12 12 11 6 1 5 3 3 0 0 12 11 19	149,99	Não aplicável
BVLC	4 4 4 2 1 1 1 20 18 34 0 2 2 2 3 3 1 6 6 7	282,92	94,64
BA – a	3 4 5 2 1 1 1 20 17 35 2 3 0 2 4 3 1 6 7 5	357,49	11,48
BA-b	4 4 4 2 1 1 1 20 18 34 0 2 2 2 3 3 1 6 6 7	282,92	15,16
BA – c	4 4 4 2 1 1 1 20 18 34 0 2 2 2 3 3 1 6 6 7	282,92	15,05
BA – d	3 4 4 2 2 1 1 18 21 33 6 0 3 2 0 4 1 6 3 1	169,01	11,98
BA – e	3 4 4 2 2 1 1 18 21 33 6 0 3 2 0 4 1 6 3 1	169,01	13,18
BVLC/BA	3 4 4 2 2 1 1 18 21 33 6 0 3 2 0 4 1 6 3 1	169,01	2507,9
BLNALC – a	4 4 5 1 1 1 1 20 14 38 0 3 0 7 3 2 0 6 8 6 1 1 0 0 2 3 2 3 2 4	483,13	89,16
BLNALC – b	4 4 5 2 1 1 1 21 20 31 0 2 0 7 3 2 0 5 5 10 0 0 0 0 0 2 2 2 2 0	196,90	808,28
BLNALC – c	4 4 5 2 1 1 1 21 20 31 0 2 0 7 3 2 0 5 5 10 0 0 0 0 0 2 2 2 2 0	196,90	977,23
BLNALC – d	3 3 5 2 2 1 1 19 17 36 1 2 0 1 0 1 1 16 34 4	473,31	1763,8
BLNALC – e	3 3 4 2 2 1 1 16 24 32 2 3 2 1 0 1 1 19 4 15	331,21	1404,31
BLNALC/BA – a	4 4 5 1 1 1 1 20 14 38 0 3 0 7 3 2 0 5 7 6 1 1 1 0 0 2 2 3 6 0	437,45	272,18
BLNALC/BA – b	4 3 4 2 2 1 1 18 19 33 0 2 2 1 0 1 1 17 15 1	194,90	385,3
BLNALC/BA – c	3 3 4 2 2 1 1 18 20 34 4 3 2 1 0 2 1 13 8 1	191,41	395,34
BLNALC/BA – d	3 3 4 2 2 1 1 18 21 33 4 3 2 1 0 2 1 16 9 2	202,67	410,46
BLNALC/BA – e	3 3 4 2 2 1 1 18 21 33 4 3 2 1 0 2 1 16 9 2	202,67	420,52

QUADRO 39: Resultados obtidos para o cenário Dados_R4_gera3

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 38 34 0 3 3 4 2 2 1 1 0 0 39	3791,8	0,0001
Heurística proposta	3 3 4 2 1 1 1 21 17 31 2 4 1 1 4 2 1 7 8 6	1099,7	0,11
MINLP	-	-	Excedeu o limite de tempo do NEOS sem se chegar em uma solução.
BVLC	-	1313,35	> 10800
BA – a	4 2 7 1 1 1 1 16 24 32 0 6 0 5 3 2 0 7 6 8 2 0 0 0 0 1 2 5 0 0	2951,33	8,36
BA-b	4 4 4 2 1 1 1 20 21 23 1 0 2 1 2 2 1 8 5 21	1320,67	13,2
BA – c	4 4 4 2 1 1 1 20 21 23 1 0 2 1 2 2 1 8 5 21	1320,67	13,2
BA – d	3 4 4 2 1 1 1 24 24 23 2 0 2 1 1 1 0 11 8 32	2083,50	15,34
BA – e	3 4 4 2 1 1 1 24 24 23 2 0 2 1 1 1 0 11 8 32	2083,50	15,32
BVLC/BA	3 4 4 2 2 1 1 14 19 22 3 2 3 2 1 2 1 11 5 13	979,74	2771,6
BLNALC – a	4 3 6 1 1 1 1 18 18 36 0 4 0 6 3 2 0 8 7 6 0 1 0 0 1 1 6 1 1 1	2815,49	43,2
BLNALC – b	3 2 8 1 1 1 1 27 21 24 5 9 0 3 2 0 0 5 5 12	4047,3	848
BLNALC – c	3 2 8 1 1 1 1 27 21 24 5 9 0 3 2 0 0 5 5 12	4047,3	965,3
BLNALC – d	2 4 6 2 1 1 1 17 17 38 2 1 0 1 0 0 0 0 2 1 2 1 2 0 1 7 3 0 8 5 3 0 2 0 2 0 0 3 3 3 0	3059,28	1134,06
BLNALC – e	4 3 3 2 1 1 1 25 18 29 0 1 2 1 1 1 1 11 29 26	2514,06	1298,94
BLNALC/BA – a	4 3 6 1 1 1 1 20 16 32 0 4 0 5 3 2 1 8 9 8	2359,69	290,52
BLNALC/BA – b	2 3 8 1 1 1 1 18 17 37 2 0 2 0 0 0 0 2 2 2 0 2 0 4 2 1 1 17 14 9	4544,47	469,68
BLNALC/BA – c	2 3 8 1 1 1 1 18 17 37 2 0 2 0 0 0 0 2 2 2 0 2 0 4 2 1 1 17 14 9	4544,47	469,68
BLNALC/BA – d	2 3 4 2 2 2 1 15 21 36 1 6 2 3 0 0 0 0 4 2 3 0 7 3 3 0 1 2 6 3 0	1972,82	448,62
BLNALC/BA – e	2 3 4 2 2 2 1 15 21 36 1 6 2 3 0 0 0 0 4 2 3 0 7 3 3 0 1 2 6 3 0	1972,82	463,18

QUADRO 40: Resultados obtidos para o cenário Dados_R4_gera4

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 58 0 0 3 3 4 2 2 1 1 0 47 0 3 3 4 2 2 1 1 0 0 58	319,19	0,0001
Heurística proposta	2 3 3 2 1 1 1 20 18 32 4 3 4 2 2 1 1 22 24 25 0 0 0 0 0 1 1 0 0 1 1	268,28	0,11
MINLP	1 7 2 1 0 3 2 19 18 16 5 1 5 3 3 0 0 19 16 32	199,45	Não aplicável
BVLC	3 5 4 2 1 1 1 18 16 38 2 2 3 2 2 2 1 27 23 22	251,05	249,03
BA – a	5 4 5 1 1 1 1 18 18 36 0 5 2 6 2 1 0 18 8 15 3 2 1 0 3 3 3 6 7 6	262,32	19,11
BA-b	3 5 4 2 1 1 1 18 16 38 2 2 3 2 2 2 1 27 23 22	251,05	35,32
BA – c	3 5 4 2 1 1 1 18 16 38 2 2 3 2 2 2 1 27 23 22	251,05	25,43
BA – d	3 5 4 2 1 1 1 18 16 38 3 3 2 2 3 2 1 31 21 20	613,13	24,33
BA – e	3 5 4 2 1 1 1 18 16 38 3 2 3 2 3 2 1 27 25 20	501,13	32,52
BVLC/BA	4 4 4 2 1 1 1 18 16 38 2 3 4 2 2 2 1 18 21 22	231,17	4482,4
BLNALC – a	4 3 5 2 1 1 1 18 24 30 2 6 3 2 3 1 0 18 9 20 0 0 0 1 0 4 4 5 3 1	240,72	73,78
BLNALC – b	3 4 5 2 1 1 1 18 19 35 4 3 3 2 3 2 0 14 12 16 0 3 0 2 0 2 5 4 5 0	234,59	997,72
BLNALC – c	3 4 5 2 1 1 1 18 19 35 4 3 3 2 3 2 0 14 12 16 0 3 0 2 0 2 5 4 5 0	234,59	1020,98
BLNALC – d	3 4 5 2 1 1 1 18 24 30 3 3 2 2 2 2 1 21 12 32	239,59	1711,9
BLNALC - e	3 4 5 1 2 1 1 18 22 32 13 0 2 5 0 0 0 14 5 12 0 2 1 1 1 2 1 18 19 18	303,44	2448,3
BLNALC/BA – a	4 3 5 2 1 1 1 18 24 30 2 6 3 2 3 1 0 18 9 20 0 0 0 1 0 4 4 5 3 1	240,72	197,57
BLNALC/BA – b	4 3 5 2 1 1 1 18 23 31 2 4 3 2 2 2 1 27 14 26	243,63	475,65
BLNALC/BA – c	4 3 5 2 1 1 1 18 23 31 2 4 3 2 2 2 1 27 14 26	243,63	423,46
BLNALC/BA – d	4 3 5 2 1 1 1 18 23 31 2 4 3 2 2 2 1 27 14 26	243,63	475,82
BLNALC/BA - e	4 3 5 2 1 1 1 18 23 31 2 4 3 2 2 2 1 27 14 26	243,63	530,91

QUADRO 41: Resultados obtidos para o cenário Dados_R4_gera5

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 58 0 0 3 3 4 2 2 1 1 0 47 0 3 3 4 2 2 1 1 0 0 58	920,78	0,0001
Heurística proposta	2 3 3 2 1 1 1 20 18 32 4 3 4 2 2 1 1 22 24 25 0 0 0 0 0 1 1 0 0 1 1	441,46	0,11
MINLP	0 7 3 1 0 3 2 18 18 17 6 1 5 3 3 0 0 18 15 28	194,18	Não aplicável
BVLC	-	446,81	> 10800
BA – a	4 3 5 2 1 1 1 20 24 28 2 6 3 2 3 1 0 17 7 23 0 3 0 0 0 3 4 6 4 1	510,85	19,95
BA-b	3 5 4 2 1 1 1 20 18 34 3 2 3 2 3 2 1 18 18 19	446,81	21,29
BA – c	3 5 4 2 1 1 1 20 18 34 3 2 3 2 3 2 1 18 18 19	446,81	22,37
BA – d	3 5 4 2 1 1 1 20 18 34 3 2 4 2 2 2 1 17 26 24	380,85	25,67
BA – e	3 5 4 2 1 1 1 20 18 34 3 2 4 2 2 2 1 17 18 24	344,26	24,51
BVLC/BA	4 3 5 2 1 1 1 20 21 31 3 5 2 2 3 2 0 11 12 17 0 1 2 3 0 3 4 5 4 2	235,29	3666,8
BLNALC – a	4 3 5 2 1 1 1 20 24 28 2 6 3 2 3 1 0 17 9 23 0 0 0 0 0 4 4 5 3 1	400,36	79,06
BLNALC – b	4 3 5 2 1 1 1 20 23 29 2 6 2 2 3 2 0 11 10 18 2 0 3 2 0 2 4 6 4 4	240,77	505,74
BLNALC – c	4 3 5 2 1 1 1 20 23 29 2 6 2 2 3 2 0 11 10 18 2 0 3 2 0 2 4 6 4 4	240,77	598,03
BLNALC – d	3 4 5 2 1 1 1 20 21 31 7 4 1 1 3 2 0 17 11 23 0 0 2 2 0 0 2 8 8 8	435,25	1272,3
BLNALC – e	4 2 4 3 2 1 1 20 24 28 2 4 5 4 1 1 2 1 13 28 31 1 0 0 0 0 2 1 4 0 0	905,33	1935,4
BLNALC/BA – a	4 3 5 2 1 1 1 20 24 28 2 5 2 2 3 2 0 11 11 24 1 0 1 1 0 2 4 10 3 1	243,89	316,90
BLNALC/BA – b	4 3 5 2 1 1 1 20 23 29 2 4 2 2 4 2 0 9 12 21 1 2 2 2 0 2 3 13 6 1	270,80	366,95
BLNALC/BA – c	4 3 5 2 1 1 1 20 24 28 1 1 1 2 1 3 0 0 5 5 13 2 0 2 2 1 3 1 18 12 17	285,78	380,32
BLNALC/BA – d	4 3 5 2 1 1 1 20 24 28 1 7 2 1 6 0 0 5 5 18 3 2 2 3 0 4 1 16 12 9	304,95	418,50
BLNALC/BA – e	4 4 4 2 1 1 1 20 23 29 2 2 4 2 2 2 1 17 20 27	347,19	412,34

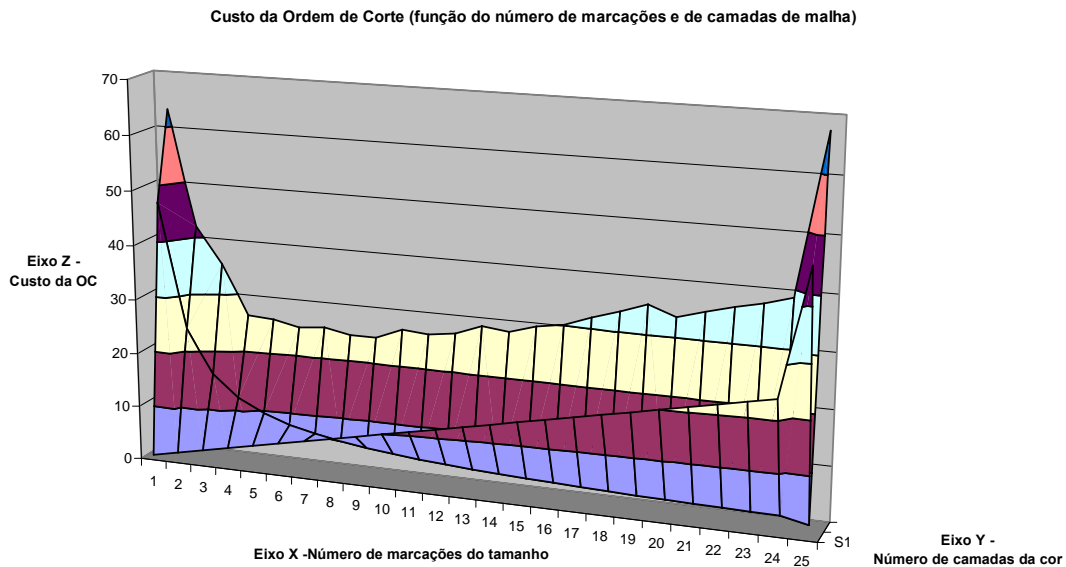
QUADRO 42: Resultados obtidos para o cenário Dados_R4_gera6

Algoritmo	Nó que representa a solução	Custo (em R\$)	Tempo de Processamento (em segundos)
Heurística Indústria	3 3 4 2 1 1 1 58 0 0 3 3 4 2 2 1 1 0 47 0 3 3 4 2 2 1 1 0 0 58	5605,9	0,0001
Heurística proposta	2 3 3 2 1 1 1 20 18 32 4 3 4 2 2 1 1 22 24 25 0 0 0 0 0 1 1 0 0 1 1	2255,4	0,11
MINLP	-	-	Excedeu o limite de tempo do NEOS sem se chegar numa solução.
BVLC	-	2822,3	> 10800
BA – a	5 4 5 1 1 1 1 18 18 36 0 5 2 6 2 1 0 18 8 15 3 2 1 0 3 3 3 6 7 6	3224,70	20,37
BA-b	4 4 4 2 1 1 1 18 22 32 2 3 3 2 2 2 1 20 14 32	2822,3	33,48
BA – c	4 4 4 2 1 1 1 18 22 32 2 3 3 2 2 2 1 20 14 32	2822,3	25,71
BA – d	4 4 4 2 1 1 1 18 22 32 2 2 3 2 2 2 1 21 19 32	2919,55	24,31
BA – e	4 4 4 2 1 1 1 18 22 32 2 2 3 2 2 2 1 21 19 32	2919,55	27,42
BVLC/BA	3 4 5 2 1 1 1 19 19 34 4 3 3 2 3 2 0 12 12 17 1 3 1 2 0 2 4 5 5 0	1566,17	6749,8
BLNALC – a	4 3 5 2 1 1 1 18 24 30 2 6 3 2 3 1 0 18 9 20 0 0 0 1 0 4 4 5 3 1	2045,50	120,09
BLNALC – b	3 4 5 2 1 1 1 20 20 32 3 2 2 2 2 2 1 17 23 25	2248,59	884,30
BLNALC – c	3 4 5 2 1 1 1 20 20 32 3 2 2 2 2 2 1 17 23 25	2248,59	1047,79
BLNALC – d	2 5 4 2 2 1 1 18 24 30 4 1 3 2 1 2 1 27 12 28	2204,91	911,30
BLNALC – e	3 4 5 1 2 1 1 18 22 32 13 0 2 5 0 0 0 5 2 7 0 4 2 4 0 4 1 10 7 6 0 3 4 2 3 0 3 3 3 6	1345,96	818,30
BLNALC/BA – a	4 3 5 2 1 1 1 18 24 30 2 6 3 2 3 1 0 18 9 20 0 0 0 1 0 4 4 5 3 1	2045,50	286,45
BLNALC/BA – b	4 3 5 2 1 1 1 18 23 31 0 14 1 0 3 0 0 4 4 5 3 0 3 3 2 2 1 18 13 18	2171,55	434,68
BLNALC/BA – c	4 3 5 2 1 1 1 18 23 31 0 14 1 0 3 0 0 4 4 5 3 0 3 3 2 2 1 18 13 18	2171,55	432,68
BLNALC/BA – d	4 3 5 2 1 1 1 18 23 31 0 14 1 0 3 0 0 4 4 5 3 0 3 3 2 2 1 18 13 18	2171,55	435,80
BLNALC/BA – e	3 4 4 2 2 1 1 18 24 30 3 2 3 2 1 2 1 24 15 33	2772,14	462,80

APÊNDICE 06: NÃO-CONVEXIDADE DA FUNÇÃO-OBJETIVO – GRÁFICO ILUSTRATIVO

Características da referência avaliada:

- Um tamanho e uma cor;
- Demanda para o par *tamanho 1 x cor 1*: 47 peças;
- Demais características idênticas àquelas definidas para a Referência 1 (vide Tabela 02).



Obs.: Os valores relativos ao custo da OC (indicados no eixo Z) demonstram a não-convexidade da função-objetivo, característica esta que se torna mais evidente à medida que aumenta o número de variáveis de decisão do problema.