PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO E SISTEMAS

ALANA PAULATTI PERI

OTIMIZAÇÃO NA UTILIZAÇÃO DE ÁREAS INTERNAS EM UMA INDÚSTRIA AUTOMOTIVA: UM ESTUDO DE CASO.

CURITIBA
2017

ALANA PAULATTI PERI

OTIMIZAÇÃO NA UTILIZAÇÃO DE ÁREAS INTERNAS EM UMA INDÚSTRIA AUTOMOTIVA: UM ESTUDO DE CASO.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas, área de concentração: Gerência de Produção e Logística, da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção

Orientadora: Prof.ª Dr.ª Maria Teresinha Arns Steiner

CURITIBA

2017

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

P441o 2017

Peri, Alana Paulatti
Otimização na utilização de áreas internas em uma indústria automotiva : um estudo de caso / Alana Paulatti Peri ; orientador, Maria Teresinha Arns Steiner. -- 2017
78f.: il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2017.
Bibliografia: f. 75-77

1. Administração da produção. 2. Indústria automobilística. 3. Peças de máquinas – Indústria. 4. Algoritmo. I. Steiner, Maria Teresinha Arns Steiner. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. III. Título.

CDD 20. ed. – 658.51

ALANA PAULATTI PERI

OTIMIZAÇÃO NA UTILIZAÇÃO DE ÁREAS INTERNAS EM UMA INDÚSTRIA AUTOMOTIVA: UM ESTUDO DE CASO.

Este trabalho, que é requisito para a qualificação no Programa de Pós-Graduação em Engenharia de Produção e Sistemas, área de concentração: Gerência de Produção e Logística, da Pontifícia Universidade Católica do Paraná, foi avaliado pela seguinte banca examinadora:

COMISSÃO EXAMINADORA

Orientadora
Prof. Dr.ª Maria Teresinha Arns Steiner

Prof. Dr. Celso Carnieri
Universidade Federal do Paraná

Prof. Dr. Anderson Roges Teixeira Góes
Universidade Federal do Paraná

Curitiba, 09 de junho de 2017.

A todos que comigo compartilharam o percurso até aqui,

minha eterna gratidão.

AGRADECIMENTOS

Agradeço, primeiramente e principalmente, aos meus pais, Mara e Dilmar, por não terem poupado esforços para que eu tivesse a melhor formação possível. Por sempre acreditarem em mim, por todo amor e dedicação ao longo de todos estes anos, a vocês o meu amor incondicional.

À minha orientadora, Prof. Dr. ª Maria Teresinha, exímio exemplo de profissional, por toda paciência, apoio e conhecimentos compartilhados durante este projeto. A você toda minha admiração.

Ao meu coorientador, Ricardo de Almeida, por todas as orientações e suporte dados durante o trabalho, meu muito obrigada.

Aos meus amigos, Wilton Neves e Matheus Bofinger, por sempre estarem comigo mesmo que a distância tenha sido um obstáculo nestes últimos anos. A vocês toda minha amizade e saudade.

À minha amiga, Luiza Pivatto, por todas as revisões, sugestões, companheirismo e paciência exigidos no decorrer deste trabalho. A você todo meu carinho.

Aos meus amigos, Caroline Ohpis e Eros Milla, meu muito obrigada por terem sido o melhor acontecimento que o edital me trouxe.

Agradeço, por fim, aos membros da banca, Prof. Dr. Celso Carnieri e Prof. Dr. Anderson Roges Teixeira Góes, pelas importantes e valiosas sugestões que já agregaram e agregarão muito neste trabalho.

RESUMO

A crise econômica brasileira que atinge a indústria automotiva reduziu, no primeiro semestre de 2016, 20,25% da produção de veículos, se comparado ao mesmo período do ano anterior. Já as vendas de automóveis foram impactadas em 20,19%, no mesmo período analisado. Diante disso, inúmeras indústrias têm buscado aprimorar seus processos e meios de produção. No caso da empresa analisada, objetivou-se otimizar as áreas internas da linha de montagem, conhecidas como áreas de *kitting*, através de técnicas de otimização aplicadas aos problemas do tipo *Bin Packing Problem* (BPP) unidimensional. Foram abordadas 20 áreas de *kitting*, com o intuito de criar uma ferramenta que permita a definição da quantidade ótima de *flowracks*, meio logístico tubular que apresenta três prateleiras, necessária para a alocação das embalagens das peças. Foram aplicados, para fins de comparação, o algoritmo de aproximação *First-Fit Decreasing* (FFD), a heurística *Minimum Bin Slack* (MBS) e uma de suas derivações *Minimum Bin Slack'* (MBS'), além do método exato (modelo matemático). Ao se aplicar o FFD nas 20 áreas de *kitting* existentes, obteve-se a redução total de 44 unidades de *flowracks*, representando uma diminuição de 21,36% nas quantidades necessárias estimadas pela empresa, e ganho de 167,06m² de área livre total nestas áreas. Já com a utilização do MBS e MBS', a quantidade de *flowracks* foi reduzida em 21,43%, 151 unidades. Os métodos apresentaram diferença apenas na quantidade total de prateleiras, em que o MBS apresentou resultado inferior em apenas uma unidade. O método exato apresentou deficiência quando aplicado à grandes instâncias, gerando resultados factíveis em apenas 17 áreas de *kitting*; foram obtidos resultados semelhantes ao MBS' diferenciando-se em uma unidade a menos de prateleiras. Para fins de adequação às características presentes na empresa, foi proposta a adaptação do algoritmo FFD para que o modelo tendesse a agrupar referências de peças de uma mesma família. Assim, o modelo se mostrou satisfatório, com redução em quantidade de *flowracks* idêntica ao método MBS'.

Palavras-chave: Problemas de corte e empacotamento. *Bin Packing Problem*. indústria automotiva.

ABSTRACT

Brazilian economic crisis, that affects the automotive industry, reduced vehicles production in 20,25% in the first semester of 2016 when compared with the previous year. Sales were reduced 20,19% in the same period. Several industries have sought to improve processes and production methods. For the company analyzed in this work, the aim was to optimize the internal areas, known as kitting areas, through optimization techniques applied to one-dimensional Bin Packing Problem (BPP). Twenty kitting areas were analyzed, with the aim of creating a tool for definition of the optimum amount of flowracks, a tubular transport device that has three shelves, required for the allocation of pieces packages. First-Fit Decreasing algorithm (FFD), Minimum Bin Slack heuristic (MBS) and its derivation, Minimum Bin Slack' (MBS'), and the exact method were used for comparison purposes. Preliminarily, FFD was applied in the 20 kitting areas, achieving a total reduction of 44 flowracks units, representing a decrease of 21,36% in the quantities estimated by the company, and representing a gain of 167,06 m² of free area. For MBS and MBS', the quantity of flowracks was reduced by 21,43%, 151 units. MBS and MBS' presented difference only in the total amount of shelves, in which MBS presented inferior result, with 1 unit less. The exact method presented deficiency when applied to large instances, generating feasible results in only 17 areas of kitting; results obtained ere similar to MBS', differentiating by one shelf unit less. For purposes of adaptation to the characteristics present in the company, it was proposed to adapt the FFD algorithm so that the model tends to group references of parts of the same family. Thus, the model was satisfactory, with a reduction in flowrack quantity identical to the MBS' method.

Keywords: Cutting and Packing Problems, Bin Packing Problem, Automotive industry.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Diferença entre item e objeto e exemplos de padrões de corte 17
Figura 2.2 - Equivalência entre corte e empacotamento 18
Figura 2.3 – Padrão de corte guilhotinado.....20
Figura 2.4 - Padrão de corte não guilhotinado 20
Figura 2.5 - Visão geral dos tipos de problemas relacionados com PCEs 22
Figura 2.6 - Tipos básicos de PCEs. 23
Figura 2.7 - Exemplificação do problema da mochila.....24
Figura 2.8 - Exemplificação do problema de empacotamento.....25
Figura 2.9 - Árvore binária para o FF. 33
Figura 2.10 - Procedimento MBS 34
Figura 3.1- Áreas do fluxo produtivo da unidade fabril. 41
Figura 3.2 - Localização da área de montagem na unidade fabril. 42
Figura 3.3 - *Layout* da área de montagem 43
Figura 3.4 - As áreas de *kitting* e seus respectivos *tronçons. *44
Figura.3.5 - Layout de uma área de *kitting.....*45
Figura 3.6 - Ilustração parcial da área de *kitting.....*45
Figura 3.7 - PE do tipo ECM-1087. 46
Figura 3.8 - Visão lateral de um *flowrack. *47
Figura 3.9 - *Flowrack* na área de *kitting *47
Figura 3.10 - GE do tipo SLI-0760.....48
Figura 3.11 – Cenário atual das áreas de *kitting *49
Figura 3.12 - Cenário futuro das áreas de *kitting *49
Figura 5.1 - Mensagem de esgotamento de memória.....64
Figura 5.2 - Passo 1 do algoritmo adaptado.....67
Figura 5.3 – Passos 2 e 3 do algoritmo adaptado 68
Figura 5.4 - Passo 4 do algoritmo adaptado.....69
Figura 5.5 – Gráfico comparativo entre quantidades de flowracks e prateleiras. 71

LISTA DE TABELAS

Tabela 2.1 - Tipologia de Dyckhoff (1990).....19
Tabela 2.2 – Razão de pior caso para os algoritmos de aproximação 31
Tabela 3.1 - Impacto da entrada de projetos nas quantidades de PE 49
Tabela 4.1 - Quantidades de flowracks e referências de cada área de kitting.....51
Tabela 5.1 – Tempos computacionais para o algoritmo FFD. 53
Tabela 5.2 - Resultados comparativos após a implementação do método FFD.....54
Tabela 5.3 - Redução entre os valores estimados e o resultado da otimização do FFD. 55
Tabela 5.4 -Ganhos em área com a aplicação do método FFD. 56
Tabela 5.5 – Tempos computacionais para o algoritmo MBS 56
Tabela 5.6 - Resultados comparativos após a implementação do método MBS.....57
Tabela 5.7 - Redução entre os valores estimados e o resultado da otimização do MBS. 58
Tabela 5.8 - Resultados comparativos com a exclusão do valor não-ótimo.....58
Tabela 5.9 -Ganhos em área com a aplicação do método MBS 59
Tabela 5.10 – Tempos computacionais para o algoritmo MBS'. 60
Tabela 5.11 - Resultados comparativos após a implementação do método MBS'.... 60
Tabela 5.12 - Redução entre os valores estimados e o resultado da otimização do MBS'. 61
Tabela 5.13 -Ganhos em área com a aplicação do método MBS'. 62
Tabela 5.14 – Tempos computacionais para o algoritmo exato 63
Tabela 5.15 - Resultados comparativos após a implementação do método exato....63
Tabela 5.16 - Redução entre os valores estimados e o resultado da otimização do método exato.....64
Tabela 5.17 -Ganhos em área com a aplicação do método exato 65
Tabela 5.18 – Tempos computacionais para FFD adaptado.....69
Tabela 5.19 - Resultados comparativos após a implementação do método FFD adaptado 70
Tabela 5.20 - Redução entre os valores estimados e o resultado da otimização do FFD adaptado 70
Tabela 5.21 -Ganhos em área com a aplicação do FFD adaptado 71
Tabela A.1 – Compilado dos resultados por métodos e AK 78

LISTA DE QUADROS

Quadro 2.1 - Panorama dos tipos de problemas intermediários: minimização de entrada 26
Quadro 2.2 - Panorama dos tipos de problemas intermediários: maximização de saída. 26
Quadro 5.1 – Exemplo de famílias de peças da AK18 66

LISTA DE SIGLAS

ANFAVEA – Associação Nacional dos Fabricantes de Veículos Automotores

BPP – *Bin Packing Problem*

BF – Best-Fit

BFD - *Best-Fit Decreasing*

CSP – *Cutting Stock Problem*

FENABRAVE – Federação Nacional de Distribuição de Veículos Automotores

FF – *First-Fit*

FFD – *First-Fit Decreasing*

GE – Grandes Embalagens

IIPP - *Identical Item Packing Problem*

IPT – *Intermediate Problem Types*

KP - *Knapsack Problem*

MBS - Minimum Bin Slack

MBS' - *Minimum Bin Slack'*

MBSBPP – *Multiple Bin-Size Bin Packing Problem*

MIKP – *Multiple Identical Knapsack Problem*

MHKP – *Multiple Heterogeneous Knapsack Problem*

MILOPP – *Multiple Identical Large Object Placement Problem*

MSSCSP – *Multiple Stock-Size Cutting Stock Problem*

ODP – *Open dimension Problem*

NF – *Next-Fit*

NFD - *Next-Fit Decreasing*

PCE – Problemas de Corte e Empacotamento

PE – Pequenas Embalagens

PL – Programação Linear

PLI – Programação Linear Inteira

PLIB – Programação Linear Inteira Binária

Programação Linear Inteira Mista (PLIM)

PP – *Placement* Problem

RBPP – *Residual Bin Packing Problem*

RCSP – *Residual Cutting Stock Problem*

SBSBPP – *Single Bin-Size Bin Packing Problem*

SLOPP – *Single Large Object Placement Problem*

SKP – *Single Knapsack Problem*

SSSCSP – *Single Stock-Size Cutting Stock Problem*

WF – *Worst-Fit*

WFD - *Worst-Fit Decreasing*

SUMÁRIO

1 INTRODUÇÃO 13
1.1 OBJETIVOS 14
1.1.1 Objetivos específicos 14
1.2 MOTIVAÇÃO 15
1.3 MÉTODOS 16
1.4 ESTRUTURA DO TRABALHO 16
2 O PROBLEMA DE CORTE E EMPACOTAMENTO 17
2.1 CONSIDERAÇÕES PRELIMINARES 17
2.2 DEFINIÇÃO DO PROBLEMA DE BIN PACKING UNIDIMENSIONAL 27
2.3 MÉTODOS DE RESOLUÇÃO DO BPP 28
2.3.1 Método exato para o BPP 28
2.3.2 Algoritmos de aproximação 29
2.3.2.1 *First-Fit Decreasing* 31
2.3.3 *Minimum Bin Slack* (MBS) e *Minimum Bin Slack'* (MBS') 33
2.4 TRABALHOS CORRELATOS: PROBLEMA DE *BIN PACKING* 36
3 DESCRIÇÃO DO PROBLEMA 41
3.1 A EMPRESA 41
3.1.1 Área de montagem 42
3.1.2 Área de *kitting* 43
3.1.2.1 Armazenamento de Pequenas Embalagens (PE) 46
3.1.2.2 Armazenamento de Grandes Embalagens (GE) 47
3.2 O PROBLEMA DE OTIMIZAÇÃO DE *FLOWRACKS* 48
4 METODOLOGIA 51
5 RESULTADOS 53
5.1 ALGORITMO DE APROXIMAÇÃO – *FIRST-FIT DECREASING* 53
5.2 MÉTODOS BASEADOS EM MBS 56
3.2.1 *Minimum Bin Slack* (MBS) 56
3.2.2 *Minimum Bin Slack'* (MBS') 59
5.3 MÉTODO EXATO 62
5.4 PROPOSTA DE ADAPTAÇÃO DE ALGORITMO 66
6 CONCLUSÕES E DIRECIONAMENTOS 73
REFERÊNCIAS 74
APÊNDICE A 78

1 INTRODUÇÃO

Segundo o relatório anual da Associação Nacional dos Fabricantes de Veículos Automotores (ANFAVEA), a crise econômica brasileira impactou em 20,25% a produção de veículos, quando comparado o primeiro semestre de 2016 em relação ao mesmo período de 2015. Ainda, a Federação Nacional da Distribuição de Veículos Automotores (FENABRAVE) aponta que as vendas de automóveis caíram 20,19%, na análise dos mesmos dois períodos.

Diante disso, inúmeras indústrias têm buscado otimizar seus processos e meios de produção, com o intuito de aumentar a produtividade e reduzir o número de atividades que não geram valor agregado ao produto final. Um exemplo, no caso específico da indústria abordada, é a existência de áreas intermediárias (áreas de *kitting)*, que são dedicadas ao armazenamento de peças, formação de conjuntos de peças e do envio destes à linha de montagem principal.

Nas áreas de *kitting*, as embalagens que apresentam dimensões superiores a 1.000mm são armazenadas no solo, e as demais devem ser armazenadas em móveis tubulares (*flowracks*). Dessa forma, a concepção desses espaços é diretamente proporcional ao número de referências distintas de peças existentes, e a otimização se torna uma alternativa viável na consideração de ocupação de superfícies produtivas.

A otimização mencionada acima encontra na área de Pesquisa Operacional boas alternativas, já que esta reúne diversos métodos quantitativos que permitem a resolução de problemas reais (PIZZOLATO E GANDOLPHO, 2013). Dentre eles, pode-se citar o Problema de Corte e Empacotamento (PCE), presente sob diversas formas dentro do contexto industrial: carregamento de contêineres, paletes e caminhões com restrições de peso; e corte de chapas e tubos metálicos.

O Problema de *Bin Packing* (BPP ou Problema de Empacotamento) é um caso particular do PCE. O BPP pode ser definido como a necessidade de se alocar itens (por exemplo, pequenas embalagens) em grandes objetos (por exemplo, prateleiras de *flowracks)*. Objetiva-se, com isso, obter uma melhor disposição dos itens nos objetos, sem que ocorra a violação da capacidade destes, ao passo que também se obtenha a menor quantidade possível de objetos.

Por serem problemas de otimização combinatória do tipo NP-*hard*, torna-se difícil a obtenção de soluções ótimas em tempos computacionais razoáveis. Assim, é

possível encontrar na literatura diversos métodos heurísticos e meta-heurísticos que abordam a resolução destes problemas (ALOISE, 1992).

De encontro a isto, o objetivo deste trabalho surge do estímulo de se atender às necessidades de uma montadora automotiva, no âmbito de seus aspectos logísticos relativos às áreas de *kitting*, por meio de técnicas da Pesquisa Operacional.

1.1 OBJETIVOS

Os objetivos deste trabalho podem ser divididos em objetivo geral e objetivos específicos.

1.1.1 Objetivo geral

O objetivo geral do presente trabalho é otimizar a utilização de áreas internas em uma montadora automotiva, por meio de técnicas de otimização aplicadas aos problemas do tipo BPP unidimensional. Mais especificamente, objetiva-se a criação de ferramentas matemáticas para a definição da quantidade ótima de *flowracks* necessária para a alocação de embalagens de peças nos mesmos, assim como a otimização no uso de suas áreas.

1.1.2 Objetivos específicos

Os objetivos específicos desse trabalho são:

a) Revisar a bibliografia referente aos temas que envolvem o Problema de Corte e Empacotamento, mais especificamente, o *Bin Packing Problem*, como algoritmos de aproximação e heurística *Minimum Bin Slack* e *Minimum Bin Slack',* além do modelo exato proposto por Martello e Toth (1990);
b) Realizar a coleta de dados referente às áreas de *kitting* da empresa estudada: quantidade de referências e disposição atual nos *flowracks*;
c) Implementar o algoritmo *First-Fit Decreasing* e realizar testes de otimização dos *flowracks*;
d) Implementar os algoritmos dos métodos baseados em *Minimum Bin Slack* e *Minimum Bin Slack'*, realizando testes com os mesmos dados coletados;

e) Gerar o modelo exato e implementá-lo;
f) Comparar os resultados obtidos pela utilização dos quatro métodos com o cenário real encontrado na empresa;
g) Propor uma modificação em que as referências das peças sejam agrupadas por familiaridade antes de serem alocadas de forma otimizada nos *flowracks*;
h) Apresentar os resultados de otimização para a empresa analisada, a fim de propor melhorias nas áreas estudadas.

1.2 MOTIVAÇÃO

Sempre que é lançado um novo projeto automotivo na empresa estudada, ocorrem duas situações problemáticas: a falta de ferramentas padronizadas para a determinação da real necessidade de aquisição de novos *flowracks*; e, o impacto da não utilização de ferramentas para otimizar o espaço consumido por áreas de *kitting*.

Todas as atividades relacionadas a esses dois fatores são realizadas de acordo com o conhecimento prévio dos analistas. A título de exemplo, pode-se citar o seguinte: para cada inserção de embalagem de peça em um *flowrack,* é necessário o encaixe por meio da experiência dos funcionários, a “olho nu”. Assim, cada modificação implica em retrabalho grandioso, já que é preciso realocar todas as demais referências de peças já existentes. Atualmente, 1.356 referências distintas de peças já estão alocadas em 134 *flowracks*. Com a entrada dos novos veículos essa demanda subirá para 2.457 referências, somente em 2017.

As áreas de *kitting* representam 21% do espaço destinado ao processo de montagem. Até julho de 2017, se nenhuma medida de otimização for adotada, esse valor subirá para 26%.

Frente a essas informações, este trabalho propõe a aplicação de algoritmos de forma a otimizar o processo de alocação de embalagens nos *flowracks*, contribuindo também para a criação de uma ferramenta que permita o melhor dimensionamento da quantidade de *flowracks* necessária para a compra. Assim, o impacto dos novos projetos no espaço consumido pelas áreas de *kitting* também poderá ser reduzido.

1.3 MÉTODOS

Foram aplicados dois métodos nesta pesquisa: o estudo de caso, definido por Gil (2002) como “estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento”; e, pesquisa-ação, conceituada por Berto e Nakano (1998) como “pesquisa de base empírica, concebida e realizada em estreita associação com uma ação e/ou solução de um problema no qual o pesquisador esteve envolvido – participando ou colaborando diretamente”.

Portanto, no primeiro momento serão coletados dados referentes às áreas de *kitting* e de *flowracks*, de forma a proporcionar elementos de entrada para os algoritmos que serão aqui utilizados. A partir dos resultados obtidos, em um segundo momento, sugestões de melhor alocação de itens nos *flowracks* serão fornecidas, de forma a otimizar as áreas em questão.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma: no capítulo 2 é apresentada a fundamentação teórica acerca dos tópicos de PCE, BPP, algoritmos relacionados e também alguns trabalhos correlatos a estes assuntos. No capítulo 3 tem-se a descrição da empresa bem como o detalhamento do problema abordado.

Já no capítulo 4, detalha-se a metodologia a ser aplicada. No capítulo 5, os resultados são apresentados. Por fim, no capítulo 6, são apresentados as conclusões e os direcionamentos para trabalhos futuros.

2 O PROBLEMA DE CORTE E EMPACOTAMENTO

São aqui realizadas algumas considerações preliminares a respeito do PCE; a descrição dos métodos que serão utilizados para a resolução dos problemas de otimização junto à montadora automotiva e, também, alguns trabalhos correlatos ao tema aqui abordado.

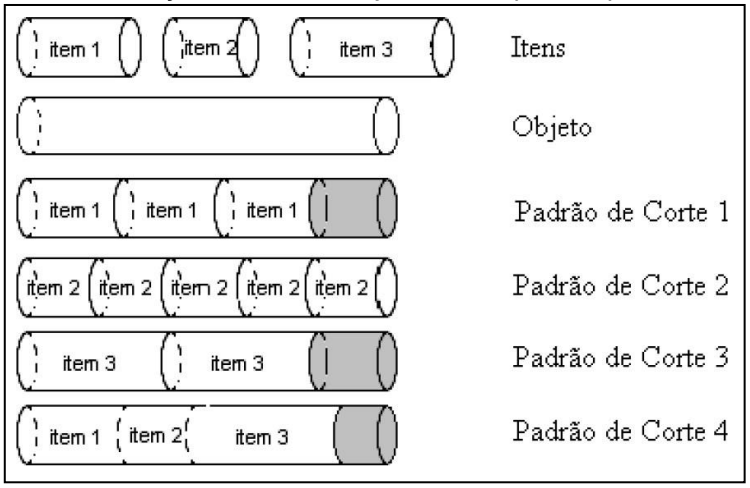
2.1 CONSIDERAÇÕES PRELIMINARES

Em muitas situações relacionadas aos ambientes industriais, tais como produção de chapas e tubos, surgem circunstâncias nas quais é necessário cortar peças ou itens em partes menores, caracterizando um problema de corte (*Cutting problem*). Quando esse caso ocorre, procura-se minimizar os possíveis efeitos negativos, entre eles o desperdício de matéria-prima. Segundo Mosquera (2007), as diversas maneiras em que determinados itens podem ser cortados a partir do objeto são denominadas “padrões de corte”.

Em outros cenários, como o carregamento de contêineres, preenchimento de paletes, dentre outros, faz-se necessário alocar itens menores em objetos maiores, explicitando um problema de empacotamento (*Packing problem*). Neste caso, objetiva-se maximizar a utilização do espaço disponível.

A Figura 2.1 consolida a diferença entre “item” e “objeto” e exemplifica alguns padrões de corte.

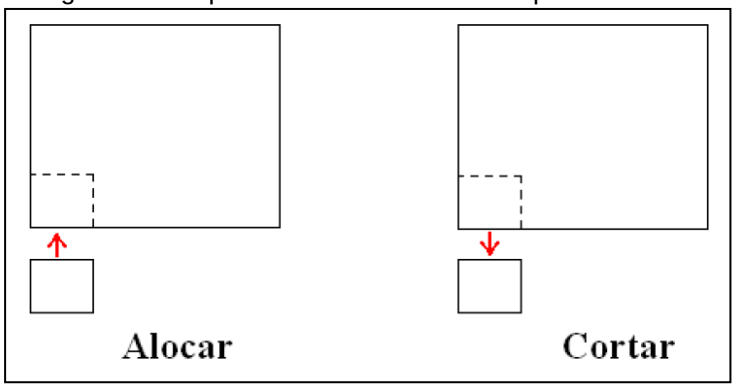
Figura 2.1 - Diferença entre item e objeto e exemplos de padrões de corte.



Fonte: Mosquera (2007).

Apesar da diferença semântica, os PCEs são equivalentes quanto à estrutura, já que se referem a mesma atividade no espaço, isto é, tanto o ato de empacotar quanto o de cortar buscam otimizar um espaço em partes menores, seja afim de permitir a alocação dos itens que devem ser empacotados ou afim de serem melhores localizados para o corte. A Figura 2.2 demonstra essa equivalência.

Figura 2.2 - Equivalência entre corte e empacotamento.



Fonte: Mosquera (2007).

Dyckhoff (1990) afirma que os PCEs apresentam diversas nomenclaturas na literatura: *cutting stock and trim loss problem*, corte de estoque, *bin packing*, *bin packing* dual, *strip packing*, problema da mochila, carregamento de veículo, carregamento de paletes, carregamento de contêineres, alocação de memória, dentre outros. Além disso, podem ser encontrados nos mais diversos campos de pesquisa, tais como na da Ciência da Computação, Engenharia Industrial e/ou de Produção, mais especificamente, envolvendo temas de estudo relacionados a Pesquisa Operacional, Otimização Combinatória, Manufatura, Matemática e outros.

Sweeney e Paternoster (1992) relatam que apesar de alguns artigos sobre PCEs terem surgido durante os anos de 1950, foram os trabalhos de Gilmore e Gomory (1961;1963) sobre o problema de corte de estoque unidimensional, os primeiros a apresentar técnicas que poderiam ser aplicadas a problemas reais.

Diante da grande diversidade de PCEs encontrados na literatura, Dyckhoff (1990) publicou, na década de 1990, uma tipologia que apresentava quatro quesitos a serem considerados na classificação de um problema. São eles: dimensionalidade; tipo de atribuição; classificação dos objetos; e classificação dos itens. Os quesitos possuem suas respectivas subclasses, a saber:

(α) **Dimensionalidade**:

(1) Problemas unidimensional;

(2) Problemas bidimensional;

(3) Problemas tridimensional;

(N) Problemas *N*-dimensionais (N>3).

(β) **Tipos de atribuição**:

(B) Todos os objetos e uma seleção dos itens pretendidos;

(V) Uma seleção dos objetos e todos os itens pretendidos.

(γ) **Classificação dos objetos:**

(O) Um objeto;

(I) Objetos idênticos;

(D) Objetos distintos.

(δ) **Classificação dos itens:**

(F) Poucos itens de formas diferentes;

(M) Vários itens de várias formas diferentes;

(R) Vários itens com pouca diversidade de formas;

(C) Itens com formas congruentes.

Dyckhoff (1990) reitera que é possível gerar 96 combinações a partir das características e suas subclasses mencionadas e que cada problema pode ser representado através da quádrupla α/β/γ/δ. Souza (2012) cita como exemplo a configuração 3/B/O/F, a qual caracteriza um problema tridimensional (3), em que um objeto grande (O) deve ser empacotado através da seleção (B) de poucos itens de tamanhos diferentes (F). Algumas tipologias definidas por Dyckhoff (1990) são demonstradas na tabela 2.1.

Tabela 2.1 - Tipologia de Dyckhoff (1990).

Problemas	Tipologia
Problema de corte de estoque (geral)	1/ / /, 2/ / /, ou 3/ / /
Problema de corte de estoque (clássico)	1/V/I/R
Problema da mochila (clássico)	1/B/O/
Problema de carregamento de Bin's (clássico)	1/V/I/M
Problema de carregamento de Bin's (bidimensional)	2/V/I/M
Problema de carregamento de contêiner	3/V/I/ , ou 3/B/O/

Fonte: Souza (2012).

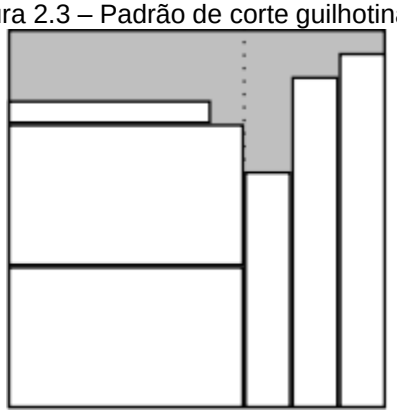
Ainda na década de 1990, Lodi *et al.* (1999) observaram que a tipologia proposta por Dyckhoff não se adequava ao contexto do trabalho intitulado "*Heuristic*

and metaheuristic approaches for a class of two-dimensional bin packing problems", já que os quatro problemas analisados eram todos classificados com a mesma forma 2/V/I/M. Assim, de acordo com o tipo de variação do PCEs, propuseram algumas restrições adicionais:

a) **Orientação:** Os itens podem possuir orientação fixa ou podem ser rotacionados em 90º;

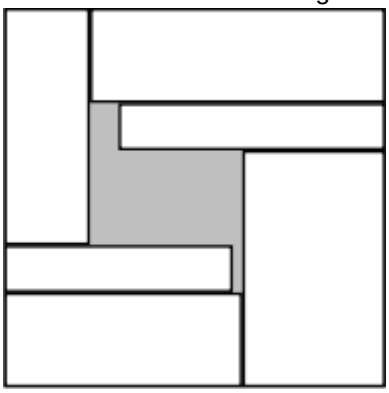
b) **Padrão de corte:** Os itens podem ser obtidos através de cortes em linha reta de um lado até o lado oposto do objeto (padrão guilhotinado), ou não (padrão não guilhotinado). As Figuras 2.3 e 2.4 mostram a diferença entre os tipos de padrão de corte.

Figura 2.3 – Padrão de corte guilhotinado.



Fonte: Torres (2004).

Figura 2.4 - Padrão de corte não guilhotinado.



Fonte: Torres (2004).

Entretanto, Wäscher *et al.* (2007) apontam algumas deficiências e inconsistências desse modelo. Relatam, também, que a tipologia de Dyckhoff não é amplamente aceita no âmbito internacional e que isso, possivelmente, ocorra em função da falta de auto explicação no esquema de codificação. Dessa forma, os

autores propuseram uma melhoria na tipologia já existente e que possui os seguintes critérios:

a) Dimensionalidade:

- Problemas unidimensional;
- Problemas bidimensional;
- Problemas tridimensional;
- Problemas N-dimensionais (N>3).

b) Tipo de atribuição:

- Maximização de saída: objetiva-se maximizar a utilização dos objetos grandes através da seleção de itens pequenos. Como a quantidade de objetos fornecidos é limitada e não suficiente para comportar todos os itens, o valor destes deve ser maximizado e, no fim, todos os objetos serão utilizados;

- Minimização de entrada: objetiva-se minimizar o uso do objeto grande através da alocação dos itens pequenos. Desta maneira, a disponibilidade de objetos é suficientemente grande para acomodar todos os itens e, portanto, deve-se minimizar o valor dos objetos utilizados.

c) Sortimento de itens pequenos:

- Itens pequenos idênticos;
- Itens com sortimento fracamente heterogêneo;
- Itens com sortimento fortemente heterogêneo.

d) Sortimento de objetos grandes:

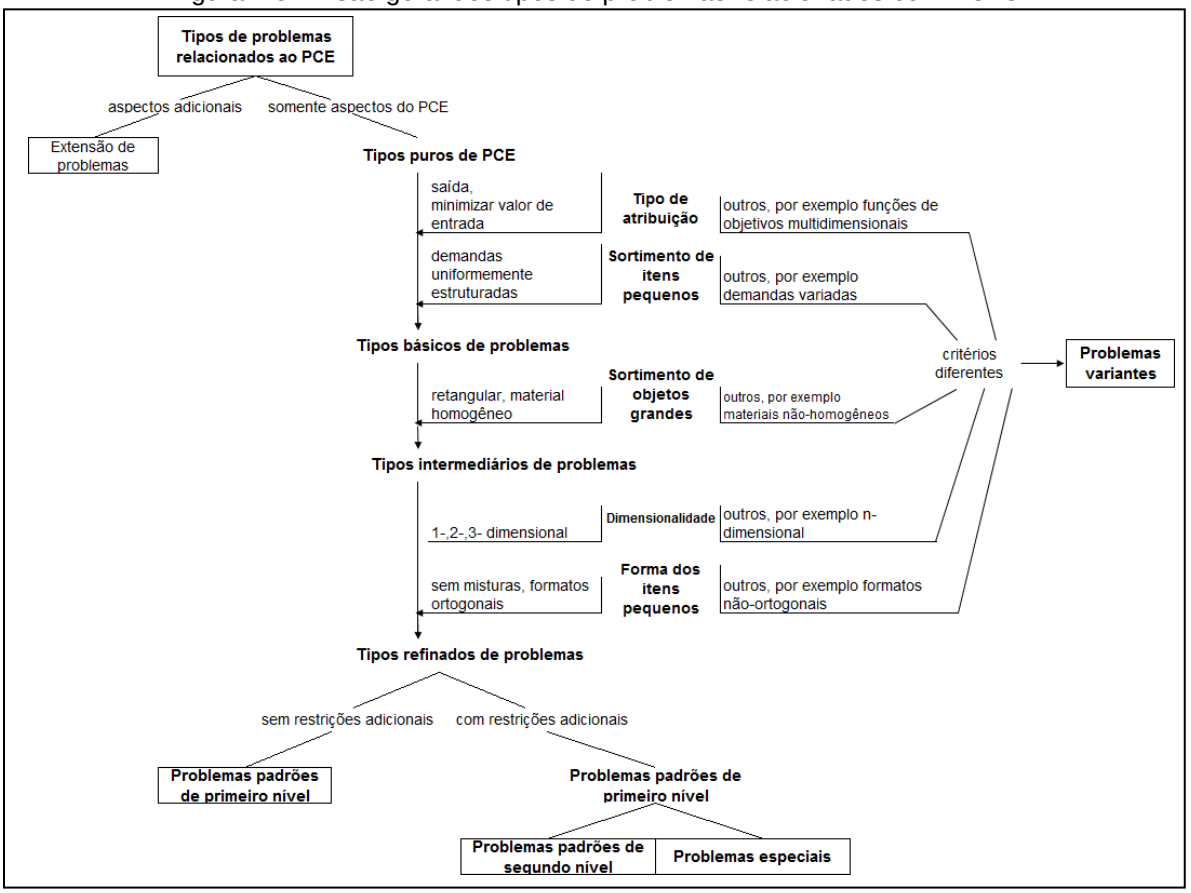
- Um único objeto grande;
  - Com todas as dimensões fixas;
  - Com uma ou mais dimensões variáveis.
- Vários objetos grandes com todas as dimensões fixas:
  - Idênticos;
  - Com sortimento fracamente heterogêneo;
  - Com sortimento fortemente heterogêneo.

e) Forma dos itens pequenos:

- Regulares;
- Irregulares.

A partir dos cinco critérios definidos, Wäscher *et al.* (2007) propuseram uma ilustração (Figura 2.5) para diferenciar os problemas de corte e empacotamento “puros”, das extensões do problema, cujas características adicionais não fazem parte dos aspectos gerais inerentes ao PCE. É possível verificar que os tipos de problemas considerados “puros” podem ser divididos em: tipos básicos; tipos intermediários; e, tipos de problemas refinados.

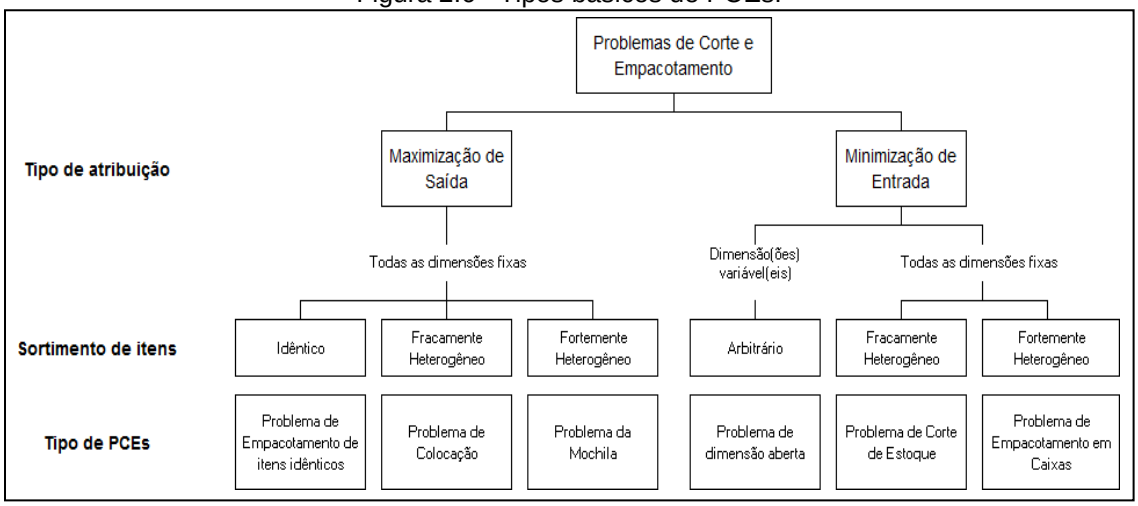
Figura 2.5 - Visão geral dos tipos de problemas relacionados com PCEs



Fonte: adaptado de Wäscher *et al.* (2007).

Os tipos básicos de PCEs são desenvolvidos a partir da combinação dos critérios “tipos de atribuição” e “sortimento de itens pequenos”. A Figura 2.6, a seguir, apresenta algumas combinações relevantes e seus tipos de problemas básicos correspondentes.

Figura 2.6 - Tipos básicos de PCEs.



Fonte: Adaptado de Wäscher *et al.* (2007).

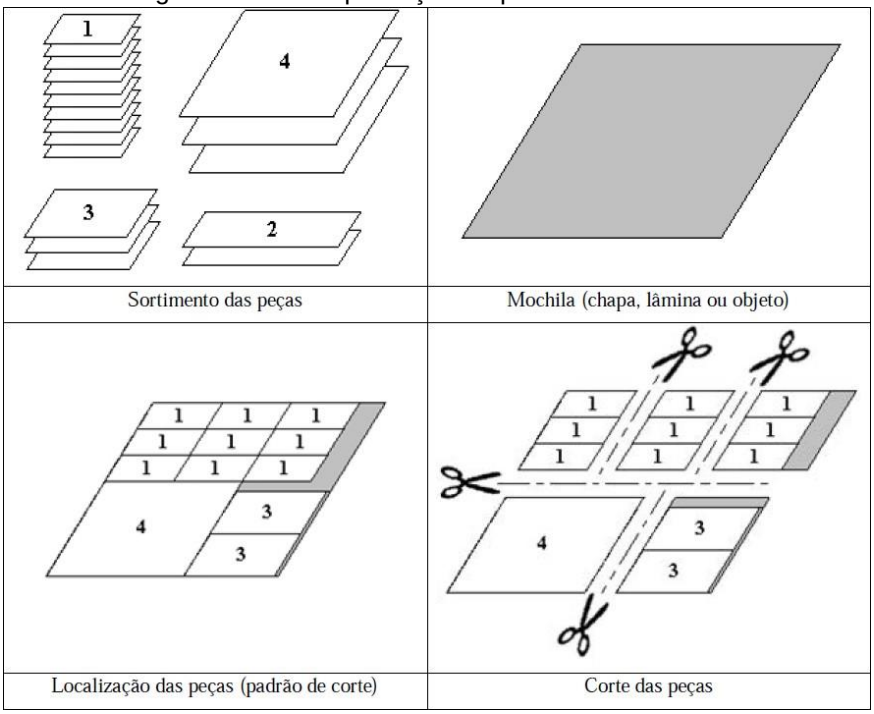
Com o objetivo de definir problemas mais homogêneos, os tipos básicos de PCEs foram estruturados em problemas intermediários, levando em consideração o critério de “sortimento de objetos grandes” como um aspecto de diferenciação adicional.

Para Wäscher *et al.* (2007), o **problema de empacotamento de itens idênticos** (IIPP - *Indetical Item Packing Problem*) consiste na atribuição do maior número possível de itens idênticos em um dado conjunto limitado de objetos. Por se tratarem de itens idênticos, não existe de fato um problema em selecioná-los e combiná-los, nem um problema real de atribuição de itens a objetos. Assim, o problema é reduzido a colocar os itens nos objetos de maneira a respeitar as restrições geométricas.

Já o **problema de colocação** (PP – *Placement* Problem) é definido como uma situação em que itens de sortimento fracamente heterogêneo devem ser atribuídos a um dado conjunto limitado de objetos grandes. Assim, o valor total da acomodação dos itens deve ser maximizado, ou as perdas correspondentes minimizadas.

O **problema da mochila** (KP - *Knapsack Problem*) é caracterizado pela presença de itens com sortimento fortemente heterogêneo, os quais devem ser alocados em um certo conjunto de objetos. Entretanto, a disponibilidade dos objetos é limitada de tal forma que nem todos os itens poderão ser atribuídos, e esse valor dos itens alocados deve ser maximizado. Martínez (2014) ilustra este problema através da Figura 2.7.

Figura 2.7 - Exemplificação do problema da mochila



Fonte: Martínez (2014).

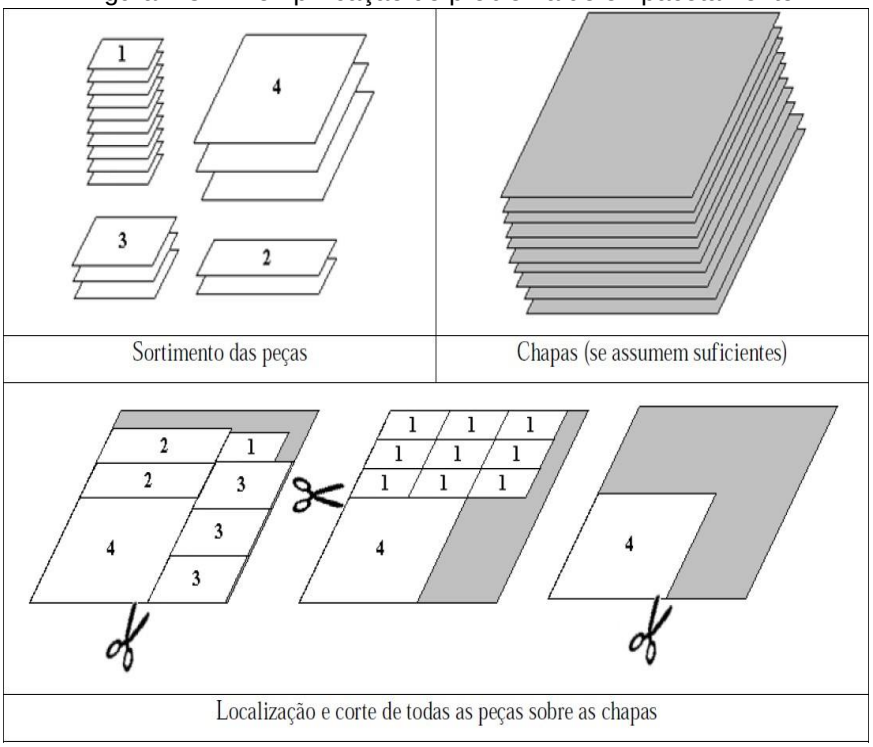
Adicionalmente, o **problema de dimensão aberta** (ODP – *Open dimension Problem*) é definido pela necessidade de se alocar completamente um conjunto de itens em um ou mais objetos, porém, a extensão de pelo menos uma das dimensões do objeto pode ser considerada variável (indefinida ou ilimitada). Desta maneira, o problema implica na decisão de fixar a(s) extensão(ões) da(s) dimensão(ões) variável(eis) dos objetos. Por outro lado, somente as partes dos objetos necessários para alocar completamente os itens representam as “entradas” na estrutura geral dos PCEs. Objetiva-se minimizar o valor de entrada (ou uma medida auxiliar como extensão, tamanho ou volume).

O **problema de corte de estoque** (CSP – *Cutting Stock Problem*)**,** por sua vez**,** é caracterizado pela completa alocação de itens de sortimento fracamente heterogêneo em uma seleção de objetos de valor mínimo, número ou tamanho total. Ao contrário do problema mencionado anteriormente, a extensão dos objetos é fixa em todas as suas dimensões geométricas.

Por fim, os autores abordam o **problema de empacotamento** (BPP – *Bin Packing Problem*) que, diferentemente do problema de corte de estoque, é caracterizado pela presença de itens de sortimento fortemente heterogêneo que devem ser atribuídos a um conjunto de objetos que podem ser idênticos, fracamente ou fortemente heterogêneos. De acordo com Martínez (2014), é possível tratar o

problema de empacotamento como uma generalização do problema da mochila. Isso pode ser verificado na Figura 2.8, a seguir.

Figura 2.8 - Exemplificação do problema de empacotamento.



Fonte: Martínez (2014).

Carvalho (2002) alega que o CSP e o BPP são bastante similares. Entretanto, o que provavelmente motivou a utilização de uma classificação distinta seja o fato de que, tradicionalmente, tem-se utilizado métodos diferentes para solucioná-los.

Wäscher *et al.* (2007) formalizaram um panorama das configurações dos tipos de problemas intermediários. O quadro 2.1 revela os problemas intermediários relacionados com a minimização de entrada. O quadro 2.2, por sua vez, descreve os tipos relacionados com a maximização da saída.

Quadro 2.1 - Panorama dos tipos de problemas intermediários: minimização de entrada

Característica dos objetos \ Sortimento de itens		Fracamente heterogêneo	Fortemente heterogêneo
Todas as dimensões fixas	Objetos Idênticos	**Problema de corte de estoque em objetos de tamanhos idênticos** SSSCSP - *Single Stock-Size Cutting Stock Problem*	**Problema de empacotamento em objetos idênticos** SBSBPP - *Single Bin-Size Bin Packing Problem*
	Objetos fracamente heterogêneos	**Problema de corte de estoque de múltiplos objetos** MSSCSP - *Multiple Stock-Size Cutting Stock Problem*	**Problema de empacotamento em múltiplos objetos** MBSBPP - *Multiple Bin-Size Bin Packing Problem*
	Objetos fortemente heterogêneos	**Problema de corte de estoque residual** RCSP - *Residual Cutting Stock Problem*	**Problema de empacotamento residual** RBPP - *Residual Bin Packing Problem*
Um objeto grande com dimensão(ões) variável(eis)		**Problema de dimensão aberta** ODP - *Open Dimension Problem*	

Fonte: Adaptado de Wäscher *et al.* (2007).

Quadro 2.2 - Panorama dos tipos de problemas intermediários: maximização de saída.

Característica dos objetos \ Sortimento de itens		Idêntico	Fracamente heterogêneo	Fortemente heterogêneo
Todas as dimensões fixas	Um Objeto grande	**Problema de empacotamento de itens idênticos** IIPP - *Identical Item Packing Problem*	**Problema de colocação em um objeto** SLOPP - *Single Large Object Placement Problem*	**Problema de uma mochila** SKP - *Single Knapsack Problem*
	Objetos Idênticos		**Problema de colocação em múltiplos objetos idênticos** MILOPP - *Multiple Identical Large Object Placement Problem*	**Problema de múltiplas mochilas idênticas** MIKP - *Multiple Identical Knapsack Problem*
	Objetos Heterogêneos		**Problema de colocação em múltiplos objetos heterogêneos** MHLOPP - *Multiple Heterogeneous Large Object Placement Problem*	**Problema de múltiplas mochilas heterogêneas** MHKP - *Multiple Heterogeneous Knapsack Problem*

Fonte: Adaptado de Wäscher *et al.* (2007).

Por conseguinte, os tipos de problemas refinados são obtidos através da aplicação dos critérios “dimensionalidade” e, para problemas bi e tridimensionais, o critério “forma dos itens pequenos”. As subcategorias resultantes são caracterizadas

por adjetivos adicionados aos nomes dos problemas intermediários (IPT – *Intermediate Problem Types*), de acordo com a seguinte estrutura: {1,2,3}-dimensional {retangular, circular,...,irregular} {IPT}.

2.2 DEFINIÇÃO DO PROBLEMA DE BIN PACKING UNIDIMENSIONAL

Aloise (1992) aponta que o termo *bin packing* foi introduzido por Johnson, em 1973, em sua tese de doutorado intitulada "*Near optimal bin packing algorithm"*, a fim de substituir o termo *loading* quando se aborda problemas de designação de arquivos de dados de determinados comprimentos às trilhas de um disco rígido. Assim, iniciou-se a denominação de *bin packing*, variantes ou generalizações, a assuntos relacionados a área de *scheduling*, *assembly line balancing,* alocação de memória e CSP.

Shwerin e Wäscher (1997) definem o BPP como a necessidade de se alocar um conjunto de itens em um número mínimo de grandes objetos (*bins*), de capacidades idênticas. Assim, é dado um número inteiro positivo C e um conjunto $N = \{1, \dots, n\}$ de itens, cada qual com dimensão positiva $w_i (i = 1, \dots, n)$ que satisfaça $w_i \leq C$. O BPP consiste em obter o menor m inteiro tal que exista uma partição $B_1, B_2, \dots, B_m$ do conjunto N em que o somatório $w(B_j) = \sum_{w_i \in B_j}^{j=1,\dots,m}$ dos pesos dos itens em cada subconjunto $B_j (j = 1, \dots, m)$ não exceda C.

De acordo com a tipologia de Dyckhoff (1990), o BPP é um problema do tipo 1/V/I/M (problema unidimensional a partir de uma seleção de objetos idênticos e com todos os itens, de formas variadas, pretendidos), conhecido por se tratar de um problema *NP-hard*. Junqueira (2007) e Aloise (1992) argumentam que problemas desta classe são intratáveis, de forma que se torna improvável a existência de um algoritmo de otimização exata que seja capaz de abranger toda e qualquer instância em tempo polinomial ou pseudopolinomial, a menos que $P = NP$.

O BPP possui diversas aplicações práticas e cotidianas. Alocações de arquivos, ordenação de atividades independentes, carregamento de caminhões com restrições de peso, e inserção de anúncios em intervalos comerciais na programação televisiva são exemplos dessas aplicações. (ALOISE, 1992; COFFMAN; GAREY; JOHNSON, 1997).

No trabalho de Kantorovich (1939) é possível encontrar, também, outros exemplos de aplicação relacionadas a organização do trabalho na indústria: organização da produção de tal forma a garantir a máxima realização do plano sob condições de um determinado *mix* de produtos; distribuição de ordens de produção a máquinas com índices de produtividade diferentes; maximização da utilização de maquinário; minimização de refugos produtivos oriundos de processos de corte; melhor utilização de combustíveis, dentre outros. Almeida (2014) reitera que, deste trabalho de Kantorovich, surgiu um dos principais modelos matemáticos para os problemas de BPP e CSP.

2.3 MÉTODOS DE RESOLUÇÃO DO BPP

Alvin (2003) esclarece que o BPP pode ser solucionado através de algoritmos exatos, algoritmos de aproximação e heurísticas. Neste trabalho foram abordados o algoritmo de aproximação, conhecido como *First-Fit Decreasing* e as heurísticas *Minimum Bin Slack* e *Minimum Bin Slack'.*

2.1.1 Método exato para o BPP

Martello e Toth (1990) fornecem uma possível formulação matemática para o problema, apresentada de (2.1) a (2.7), a seguir, em que:

w_j = peso do item j,

c = capacidade cada *bin,*

atribui-se cada item a um *bin*, tal que o peso total dos itens não exceda a capacidade c e que a quantidade de *bins* seja mínima. Dessa forma:

$$min\ z = \sum_{i=1}^{n} y_i \tag{2.1}$$

$$s.a \quad \sum_{j=1}^{n} w_j x_{ij} \leq c y_i\,, \qquad i \in N = \{1, \ldots., n\}, \tag{2.2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j \in N, \tag{2.3}$$

$$y_i = 0 \text{ ou } 1, \qquad i \in N, \tag{2.4}$$

$$x_{ij} = 0 \text{ ou } 1, \qquad i \in N, j \in N, \qquad (2.5)$$

onde, a função objetivo (2.1) busca obter a quantidade mínima de *bins* utilizados. A restrição (2.2) assegura que a capacidade de cada *bin* seja respeitada. A restrição (2.3) garante que o item j seja atribuído uma única vez ao bin. Em (2.4) e (2.5), as variáveis y_i assumem o valor "1", se o *bin* i for usado, e "0" caso contrário; e para x_{ij} se atribui o valor "1", se o item j for designado ao *bin* i, e "0" caso contrário. Em qualquer solução viável, por questão de simplicidade, assume-se que os *bins* de menor índice são utilizados, isto é, $y_i \geq y_{i+1}$ para $i = 1, \ldots, n-1$.

Dessa forma, o modelo apresentado acima se trata de um problema de Programação Linear Inteira Binária (PLIB), caracterizado por objetivar a otimização, maximização ou minimização, de uma função-objetivo (eq. 2.1), sujeita às restrições modeladas por equações ou inequações lineares (2.2, 2.3, 2.4 e 2.5).

Os problemas de Programação Linear (PL) são conhecidos como métodos exatos, com capacidade de garantir soluções ótimas. A PLIB é utilizada para tratar CSP, em que os itens possuem sortimento fracamente heterogêneo, já que soluções de boa qualidade podem ser obtidas por relaxação da PLI através de heurísticas. Já no caso do BPP, em que os itens são fortemente heterogêneos, as variáveis de decisão na solução ótima na relaxação da PLI geralmente possuem valores que são iguais a fração de unidade, o que dificulta a localização de soluções inteiras (CARVALHO, 2002).

Frente ao apresentado, neste trabalho aplica-se o *software* IBM® ILOG® CPLEX® *Interactive Optimizer* para o problema modelo de PLIB modelado anteriormente. Este *software* faz uso do método *branch & cut* aliado a utilização de heurísticas para encontrar soluções inteiras de forma otimizada.

.

2.1.2 Algoritmos de aproximação

A classe dos algoritmos de aproximação pode ser dividida em *on-line* e *off-line*. Um algoritmo *on-line* é aquele em que os itens chegam em alguma ordem pré-definida e o seu empacotamento deve ocorrer tão logo quanto possível, sem o conhecimento dos demais itens posteriores. Os algoritmos *First-Fit* (FF), *Best-Fit* (BF), *Next-fit* (NF) e *Worst-Fit* (WF) são classificados como tal, e Alvin (2003) os define, a saber:

a) *First-Fit*: ao se atribuir um item a um *bin*, considera-se todos os *bins* parcialmente cheios. Aloca-se o item no *bin* de menor índice que o caiba. Um novo *bin* só é aberto quando não há nenhum objeto que possa ser incluído nos *bins* abertos. De acordo com Johnson (1973), este algoritmo pode ser implementado com complexidade $O(n \log n)$;

b) *Best-Fit*: aloca-se o item no bin mais cheio em que é possível colocá-lo, dando preferência aos *bins* de menor índice. Abre-se um novo *bin* caso não exista *bin* capaz de recebê-lo. Esse algoritmo possui a mesma complexidade do FF;

c) *Next-Fit*: diferentemente do FF e do BF, atribui-se o primeiro item ao *bin* de índice "1". Os outros itens são alocados no *bin* em que o item anterior foi alocado sucessivamente, se houver espaço; caso contrário, o item é empacotado em um novo *bin*. A complexidade deste algoritmo é $O(n)$;

d) *Worst-Fit*: de forma contrária ao BF, atribui-se o item no *bin* menos cheio e de menor índice, onde o item couber. Caso não exista um *bin* capaz de empacotá-lo, abre-se um novo. Sua complexidade é igual a do FF e do BF.

Alvin (2003) afirma que, de maneira geral, o pior caso para um empacotamento ocorre quando se atribui muitos itens de pequenos pesos aos primeiros *bins*. Dessa forma, quando se relaxa a restrição do processamento *on-line*, os itens são rearranjados em ordem decrescente e subsequentemente é aplicada uma regra de empacotamento. Quando a regra de empacotamento é a de FF, tem-se o algoritmo *off-line* conhecido como *First-Fit Decreasing* (FFD). Analogamente, têm-se os algoritmos *Best-Fit Decreasing* (BFD), *Next-Fit Decreasing* (NFD) e *Worst-Fit Decreasing* (WFD).

Almeida (2014) assegura que os algoritmos de aproximação são simples e fornecem soluções de forma eficiente, entretanto, podem gerar soluções insatisfatórias. Porém, isso pode ser compensado pela facilidade de implementação e a rapidez com que é possível obter os resultados.

Segundo Coffman *et al.* (1997), para cada algoritmo de aproximação há um desempenho assintótico de pior caso relativo. Alvin (2003) descreve que: seja P um problema de minimização e A uma heurística de resolução para P. Para cada instância I do problema P, têm-se $z(I)$ como valor da solução ótima e $A(I)$ como solução obtida pela heurística. Assim, a razão de desempenho de pior caso absoluto da heurística *A* pode ser definida conforme (2.8).

$$r(A) = sup\{\frac{A(I)}{z(I)} : para\ todas\ as\ instâncias\ I\} \quad (2.8)$$

Para o BPP, especificamente, o pior caso absoluto se trata de uma informação muito ruim sobre o comportamento de pior caso do algoritmo. Desta maneira, a medida padrão, para estes casos, é a razão do desempenho do pior caso assintótico, ou seja, o limite do desempenho de pior caso quando o tamanho da instância sofre crescimento. Logo, a razão de desempenho de pior caso assintótico pode ser conhecido por meio de (2.9) (ALVIN, 2003).

$$r^{\infty}(A) = \lim_{z(I)\to\infty} sup\{\frac{A(I)}{z(I)} : para\ todas\ as\ instâncias\ I\} \quad (2.9)$$

Portanto, quanto mais próximo de "1" for o $r^{\infty}(A)$, melhor será o algoritmo *A*. A Tabela 2.2 traz os valores de razão de pior caso para os algoritmos mencionados.

Tabela 2.2 – Razão de pior caso para os algoritmos de aproximação.

Algoritmo	r^{∞}
First-Fit	1,7
Best-Fit	1,7
Next-Fit	2
Worst-Fit	1,7
First-Fit Decreasing	1,222...
Best-Fit Decreasing	1,222...
Next-Fit Decreasing	1,69103...

Fonte: adaptado de Alvin (2003).

Para Scholl *et al*. (1997), o FFD e BFD apresentam o melhor comportamento de pior caso, com uma razão de pior caso igual a 1,222. Dessa forma, o valor da função objetivo obtida pelo algoritmo se desvia no máximo em 22,2% do valor ótimo, desde que o número de *bins* seja suficientemente grande.

2.3.1.1 First-Fit Decreasing

Como mencionado anteriormente, o FFD funciona com uma regra de pré-processamento de reordenação decrescente dos itens. Junqueira (2007) coloca que após esse processo, o FFD funciona exatamente como o FF. Sejam:

1. P um empacotamento de uma lista L;
2. $j = MIN\{j' | BIN_{j'F}$ está usado em $P\ e\ \alpha_{j'F} + w_i \leq 1\}$; e

3. w_i o próximo item de L a ser atribuído.

considera-se α_{jF} como a capacidade do *bin* já utilizada e w_i como o peso do item a ser atribuído, neste caso a capacidade do *bin* é igual a "1". Junqueira (2007), então, define a regra de empacotamento do FF, da seguinte forma:

- Se $\exists j$, como mencionado em 2, então atribua o w_i ao BIN_j;
- Senão, abra um novo *bin* e atribua w_i a este *bin*.

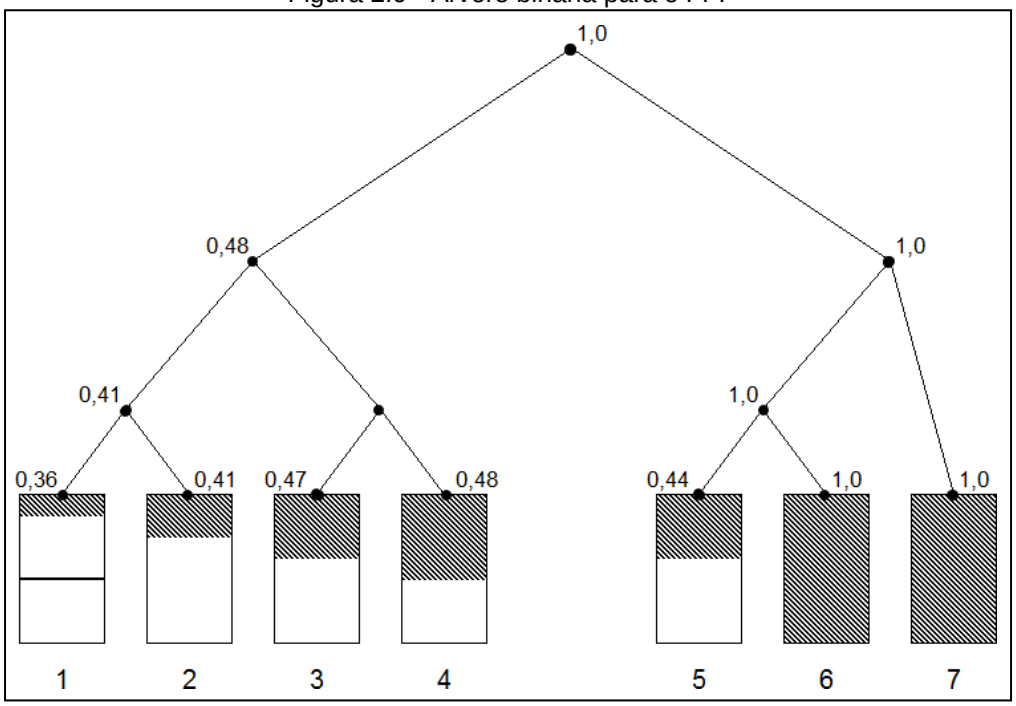
Johson (1973), em sua tese intitulada "*Near-optimal bin packing algorithms*", explica que o FF pode ser implementado através de uma estrutura de árvore binária.

Constrói-se uma árvore binária com profundidade igual a $\log_2 n$, com n folhas correspondentes ao número n de *bins* vazios, no sentido da esquerda para a direita. Os nós internos e raiz da árvore são rotulados com "1". Cada folha da árvore também é rotulada com um valor igual a "1", inicialmente, correspondendo ao máximo espaço vazio possível em um *bin*.

À medida que os itens são atribuídos aos *bins*, os valores dos rótulos das folhas são atualizados de maneira que, para cada item atribuído, o novo valor do rótulo do *bin*, que sofreu a atribuição, corresponderá ao espaço vazio que esse *bin* ainda possui. Da mesma maneira, cada nó interno também sofre atualização e seu novo valor corresponde ao valor máximo dentre os rótulos de seus dois descendentes na árvore, isto é, qualquer nó interno sempre indicará o máximo espaço vazio que ainda existe em qualquer *bin*.

Para alocar um item em um *bin*, inicia-se pela raiz da árvore e atravessa-se até a folha com o galho mais à esquerda que possua um rótulo com valor maior ou igual ao requerido pelo item. A Figura 2.9 mostra o estado da árvore quando um item está prestes a ser alocado. Por exemplo: se um item possui tamanho 0,5, buscar-se-á a folha mais à esquerda possível que possua espaço livre suficiente, neste caso a folha de número 6. Isso irá requerer no máximo $\log_2 n$ comparações de "sim ou não". Depois do item ser atribuído, será necessário atualizar a árvore, requerendo mais $\log_2 n$ comparações. Deve-se iniciar a atualização a partir do rótulo da folha correspondente ao *bin*, em que ocorreu a atribuição, reproduzindo a atualização pelos nós precedentes, até a raiz. Logo, o custo total de implementação de uma lista L será no máximo igual a $2n[\log_2 n]$.

Figura 2.9 - Árvore binária para o FF.



Fonte: adaptado de Johnson (1973).

2.1.3 *Minimum Bin Slack* (MBS) e *Minimum Bin Slack'* (MBS')

Gupta e Ho (1999) propuseram um algoritmo que visa minimizar o tempo ocioso ou a folga em cada *bin*. Ao contrário dos algoritmos FFD e BFD, em que se busca o *bin* a qual um determinado item deve ser atribuído, o algoritmo proposto pelos referidos autores, chamado de *Minimum Bin Slack* (MBS), é focado no objeto (*bin-focus*). Dessa maneira, o algoritmo busca encontrar um conjunto de itens que melhor atenda a capacidade do *bin*.

Neste algoritmo, a lista de itens é ordenada de forma decrescente, e a busca do algoritmo se inicia com os maiores itens. A cada etapa, a lista Z' dos n' itens não atribuídos é mantida. E, a cada empacotamento ocorrido, os itens envolvidos são alocados em um *bin* e removidos da lista Z', preservando a ordem de classificação. O processo acaba quando a lista se torna vazia.

Fleszar e Hindi (2002) explicam que cada empacotamento é determinado através de um procedimento de busca, testando todos os possíveis subconjuntos de itens da lista que atendem a capacidade do *bin*. O subconjunto que gera a menor folga é adotado, e se o algoritmo encontra um subconjunto que atenda completamente o *bin*, a busca é encerrada. A implementação computacional de um empacotamento

para um *bin* pode ser feita de forma recursiva ou iterativa. A implementação recursiva é demostrada pela Figura 2.10.

Figura 2.10 - Procedimento MBS.

```
rotina MBSOnePackingSearch(q)
início
        para r := q até n' faça
        início
                i := Z'_r;
                se t_i ≤ s(A) então
                início
                        A := A ∪ {i};
                        MBSOnePackingSearch(r+1);
                        A := A\{i};
                        se s(A*) = 0 então Saia;
                fim;
        fim;
        se s(A) < s(A*) então A* := A;
fim;
```

Fonte: Adaptado de Fleszar e Hindi (2002).

onde n = quantidade de itens; n' = quantidade de itens não atribuídos; i = índice do item; t_i= tamanho do item i; t^{min}= tamanho do menor item; Z = lista de itens; Z' = lista de itens não atribuídos; r, q = índices da lista de itens Z; Z'_r = r-ésimo item da lista Z; A = conjunto de itens; A^*= conjunto de itens no empacotamento ótimo do *bin*; $s(A)$= folga remanescente em um *bin*.

Gupta e Ho (1999) mostram a superioridade do algoritmo em relação aos algoritmos FFD e BFD, principalmente para instâncias em que é necessário que a solução ótima possua *bins* completos; entretanto, deve-se salientar que os autores aplicaram testes a problemas com no máximo 70 itens.

Fleszar e Hindi (2002) argumentam que é muito fácil criar um problema com pequenas instâncias que fará com que o MBS tenha um gasto computacional enorme. Citam, como exemplo, o caso de um *bin* que tenha capacidade de valor ímpar, e os itens possuam dimensões de valores pares. Isso fará com que o algoritmo procure em todas as combinações possíveis, já que não será possível encontrar um empacotamento completo nesse caso.

Dessa forma, Fleszar e Hindi (2002) propuseram três condições que reduzem o tempo computacional e que não influenciam na solução final fornecida pelo algoritmo, a saber:

a) No *loop* “para”, se $r > q$ e os tamanhos dos itens Z'_{r-1} e Z'_r são os mesmos, então ignora-se o processamento do item Z'_r, já que não é

possível que Z'_r gere empacotamento melhor que os encontrados anteriormente;

b) Ao fazer T_r como a soma dos tamanhos dos itens Z'_r, ... , Z'_n, se no *loop* "para" $s(A) - T_r \geq s(A^*)$, o procedimento é interrompido, já que não é possível construir um empacotamento com os itens restantes melhor do que A^*. Essa condição é capaz de economizar o tempo computacional consideravelmente quando o tamanho do *bin* é muito grande de comparado aos tamanhos dos itens;

c) No começo do procedimento, se $s(A) < t^{min}$, então o *loop* "para" é ignorado, já que não existem itens que possam ser adicionados ao empacotamento A.

Assim, Fleszar e Hindi (2002) desenvolveram uma versão modificada da versão original do MBS, chamada de *Minimum Bin Slack'* (MBS'). A modificação consiste em, antes do procedimento de busca, escolher um item, chamado de "semente" para ser fixado permanentemente no empacotamento. Geralmente, a melhor opção de semente é o item de maior tamanho, ou seja, o primeiro item da lista Z'. Isso fará com que se deixe o menor espaço possível para preencher durante o procedimento de busca e, consequentemente, reduzir o tempo de processamento.

O algoritmo proposto faz uso do mesmo procedimento de busca mencionado anteriormente. O único ponto de diferença se concentra na inicialização, já que o primeiro item é sempre fixo no empacotamento A, então o procedimento é chamado com $q = 2$, e $A = A = \{Z'_1\}$.

O MBS' consegue gerar empacotamentos e soluções diferentes dos sugeridos pelo algoritmo original. Pode levar a resultados superiores - considerando o comportamento do MBS que tende a construir empacotamentos com itens pequenos - e pode gerar *bins* com grandes folgas, quando aloca os itens grandes por último, e não há combinações para preenchimento completo. Também é possível, inclusive, gerar soluções inferiores dependendo das características do problema, como é o caso, por exemplo, se o algoritmo criar empacotamentos não-completos para alguns dos *bins* iniciais, envolvendo itens que poderiam ser alocados em empacotamentos completos se tivesse sido usado o MBS.

Por fim, Fleszar e Hindi (2002) concluem dizendo que o MBS' se comporta melhor que o MBS. Almeida (2014) realizou alguns testes de comparação entre os

dois algoritmos, nos quais é possível verificar que, de acordo com as características do problema, os dois podem gerar resultados superiores. Entretanto, quando se considera o tempo computacional, o MBS' se mostra mais eficiente.

2.4 TRABALHOS CORRELATOS: PROBLEMA DE *BIN PACKING*

Nesta seção são descritos, de forma sucinta, alguns trabalhos relacionados ao desenvolvimento de técnicas de resolução do BPP, assim como a aplicação dessas técnicas. Além disso, também são abordados trabalhos relacionados ao problema do CSP, que possui estreita relação com o BPP.

Eilon e Christofides (1971) propuseram um procedimento exato para resolução do BPP/CSP baseado no algoritmo de enumeração implícita de Balas (1965). Na formulação proposta, a atribuição de pesos em cada *bin* foi realizada de forma que quanto maior o índice do *bin,* maior o peso. Assim, elimina-se a necessidade de fazer uso de outras variáveis para indicar o uso de um *bin,* já que se assegura a utilização do menor número de *bins*.

Ainda sobre BPP/CSP, Hung e Brown (1978) desenvolveram um método de solução em que utilizam uma combinação de enumeração implícita, caracterização de matrizes de alocação e várias regras que buscam evitar redundâncias de atribuições. Utilizaram, também, a Relaxação Lagrangeana para melhorar a eficiência do algoritmo.

Entretanto, ambos os métodos desenvolvidos são capazes de resolver apenas problemas de pequenas instâncias. Dessa forma, Martello e Toth (1990) propuseram o algoritmo MTP, baseado na estratégia de ramificação (*branching*) do FFD. Alvin (2003) aponta a importância do trabalho de Martello e Toth já que, ainda hoje, é referência básica para a resolução deste problema de forma exata.

Schwerin e Wäscher (1997) sugeriram um problema gerador para o BPP para ser utilizado em avaliações futuras de métodos de solução exatos e heurísticos. Realizaram, também, alguns experimentos com o método FFD e o MTP (Martello e Toth, 1990) a fim de identificar problemas difíceis para o BPP. Os autores confirmam a opinião geral de que o MTP representa o estado-da-arte para métodos exatos de solução para o BPP, ainda que o tamanho dos problemas, cuja resolução demande tempo computacional razoável, seja insatisfatório.

Scholl *et al.* (1997) geraram um procedimento híbrido, chamado de BISON, para a solução exata do BPP através da combinação de *bounds* inferiores, procedimentos de redução tradicionais e novos, diversas heurísticas e a apresentação de um novo esquema de *branching* em um procedimento de *branch-and-bound*.

Morabito e Garcia (1998) abordaram um caso particular de CSP em uma indústria brasileira de compensados, já que não é possível aplicar os modelos clássicos de PCE devido às restrições particulares da máquina de corte. A empresa em questão possuía uma perda proveniente do processo de corte de aproximadamente 20 toneladas por dia, que poderia ser minimizada através de um planejamento mais eficiente da produção da máquina de corte. Dessa forma, os autores propuseram uma abordagem baseada em um modelo de duas fases, cada uma como um Problema de Programação Inteira, e apresentaram dois métodos alternativos para resolvê-las: o primeiro, baseado nas técnicas de Programação Dinâmica; e o segundo, sendo uma extensão dos procedimentos de enumeração implícita. Objetivaram determinar o melhor padrão de corte como uma função dos itens demandados, de maneira que os resíduos sejam minimizados. Para ilustrar o desempenho dos dois métodos propostos, os autores resolveram cerca de 600 exemplos aleatórios. Um problema real derivado do estudo de caso foi resolvido para descrever os benefícios dessa abordagem em comparação com a solução adotada pela empresa.

Cunha *et al.* (2008) realizaram a aplicação de três heurísticas, duas baseadas em GRASP (*Greedy Randomized Adaptive Search Procedure*) e uma baseada em Busca em Vizinhança Variável (BVV), para resolver o BPP no contexto logístico de distribuição. Objetivaram minimizar a frota de veículos necessária para alocar as entregas de produtos e mercadorias, respeitando as capacidades. As heurísticas foram avaliadas de acordo com *benchmarks* na literatura para o BPP. Os resultados apresentaram boas soluções em tempos computacionais reduzidos. Os autores destacam a heurística GRASP-2, considerando a qualidade das soluções obtidas para as instâncias de teste.

Cui e Yang (2010) apresentaram uma heurística para o CSP unidimensional com a reutilização de sobras. O algoritmo consiste em dois procedimentos: o primeiro é de PL, que cumpre a maior parte do item demandado; o segundo é heurístico sequencial, que satisfaz a porção restante do item. O algoritmo consegue balancear o custo do objeto consumido, o lucro proveniente das sobras e o lucro da redução de

estoques. Os resultados computacionais mostram que o algoritmo proposto apresenta desempenho melhor do que os apresentados na literatura recente.

Abuabara e Morabito (2009) abordaram o CSP no contexto de corte de tubos metálicos utilizados na fabricação da estrutura de pequenas aeronaves agrícolas. O problema é modelado através da Programação Linear Inteira Mista (PLIM), objetivando minimizar as perdas por corte e considerando a possibilidade de gerar "sobras" com tamanhos suficientes para serem reutilizadas. Com o objetivo de validar os métodos propostos na prática, utilizaram a linguagem GAMS (*General Algebraic Modeling Systems*) com o *software* CPLEX, para realizar experimentos computacionais com dados fornecidos pela empresa fonte do estudo de caso. Os resultados mostram que os modelos são úteis para dar suporte nas decisões oriundas do processo de corte em questão.

Stavrinides e Karatza (2011) avaliaram, por simulação, o desempenho de uma heurística dos algoritmos de escalonamento, para a programação de gráficos de multitarefas com prazos *end-to-end* em um sistema em tempo real de distribuição heterogênea. A fase de seleção de atividades é baseada nas políticas de EDF (*Earliest Deadline First*), HLF (*Highest Level First*) ou LSTF (*Least Space-Time First*). A fase de seleção do processador é implementada sem a exploração de possíveis lacunas no cronograma, ou através da exploração das possíveis lacunas, aplicando as técnicas de *bin packing*, FF, BF e WF. Os resultados da simulação mostraram que as versões alternativas do algoritmo, que exploram as possíveis lacunas de cronograma, resultam em melhores valores. Além disso, o BF, quando usado para a exploração de tempos ociosos, apresenta melhor desempenho que o FF e o WF. Assim, a combinação da política EDF, na fase de seleção de tarefa, com o BF, na fase de seleção do processador, supera todas as outras heurísticas examinadas.

Abidi *et al.* (2013) desenvolveram um Algoritmo Genético para o BPP unidimensional, chamando-o de GA-BPP. Esse algoritmo gera uma série de perturbações, através da aplicação de alguns operadores dependentes do problema, com o objetivo de melhorar a solução atual. Os autores revelam que o algoritmo é eficiente e de fácil implementação. Quando aplicado a 1210 instâncias de *benchmark*, o GA-BPP é considerado o mais conhecido ou o ideal para 930 instâncias, sendo capaz de resolver todas as instâncias do *dataset* Gau-1.

Dokeroglu e Cosar (2014) combinaram as características das melhores heurísticas BPP, Algoritmos Genéticos de Agrupamento (GGA – *Grouping Genetic*

Algorithm) e capacidades da computação paralela, para obter soluções ótimas de uma forma viável sem aumentar a comunicação entre os GGA, ao passo que se alcança maiores benefícios. Os GGAs híbridos e paralelos apresentados foram aplicados de forma satisfatória a 1.318 problemas de *benchmark* de BPP unidimensionais. A taxa de sucesso obtida com esses algoritmos é superior a maioria das abordagens encontradas na literatura, sendo capaz de alcançar a solução ótima em 88,54% das instâncias em tempos computacionais práticos.

Castellanos *et al.* (2015) apresentaram o Algoritmo Genético de Agrupamento com Transmissão Controlada de Genes (GGA-CGT – *Grouping Genetic Algorithm with Controlled Gene Transmission*) para o BPP. O algoritmo inclui estratégias heurísticas que permitem a transmissão dos melhores genes dos cromossomos, permitindo a exploração do espaço de busca. O GGA-CGT controla a seleção dos indivíduos, com o intuito de criar um balanço entre a pressão seletiva e a diversidade populacional, evitando convergências prematuras do algoritmo e obtendo melhores soluções em um número menor de gerações. O algoritmo proposto supera os resultados obtidos pelos algoritmos de última geração no *dataset* Hard28, conhecido como a classe dos casos mais difíceis relatados. Entretanto, um dos problemas relatados com o algoritmo é a necessidade de definição de uma grande quantidade de parâmetros.

Brandão e Pedroso (2016) desenvolveram um método exato, baseado na formulação *arc-flow* com restrições laterais, para resolver o BPP e o CSP. O método inclui um algoritmo de compressão gráfica, que geralmente reduz substancialmente o grafo de base sem enfraquecer o modelo. A formulação proposta é equivalente a fornecida por Gilmore e Gomory (1961), que proporciona uma relaxação linear forte. Entretanto, em vez de utilizar a geração de colunas no processo iterativo, o método constrói um grafo, onde cada “caminho” da origem até o nó de destino representa um padrão de empacotamento viável. Os autores conseguiram resolver a maioria das instâncias de *benchmark,* com uma média de menos de um minuto por instância. A maior limitação encontrada para o método é a combinação entre grandes capacidades e padrões longos, que podem se tornar difíceis de serem representados de uma forma compacta.

Han *et al.* (2016) detalharam processos e estratégias baseados no algoritmo evolutivo DE (*Differential Evolution*) e no Algoritmo Genético, a fim de solucionar o BPP unidimensional. Analisaram as principais diferenças entre os dois algoritmos e concluíram que ambos são capazes de resolver o problema, apesar de apresentarem

soluções com algumas diferenças como: ordens de empacotamento e a quantidade de item por *bin.*

Buljubašić e Vasquez (2016) propuseram um novo algoritmo baseado em busca local para a resolução do BPP unidimensional. O objetivo dessa busca local foi obter uma solução viável a partir de um dado número de *bins*, m, iniciando em $m = UB - 1$, em que UB é o limite superior obtido através de uma variante da heurística FF. A principal característica desse método é prosseguir com configurações parciais do espaço de busca. Os itens ou são atribuídos a $m - 2$ *bins,* enquanto satisfazem a restrição de capacidade, ou não são atribuídos. Assim, o algoritmo busca derivar um conjunto de itens não atribuídos que podem ser empacotados em dois *bins* para obter uma solução viável completa com m *bins.* Essa metodologia faz uso de poucos parâmetros e supera todas as abordagens heurísticas anteriores em termos de número de instâncias resolvidas para otimização. Os autores fazem referência, ainda, à capacidade do algoritmo em obter soluções melhores do que outras heurísticas para os *datasets* hard28 e gau_1.

3 DESCRIÇÃO DO PROBLEMA

Neste capítulo será descrita a empresa, bem como suas áreas, e o problema em questão.

3.1 A EMPRESA

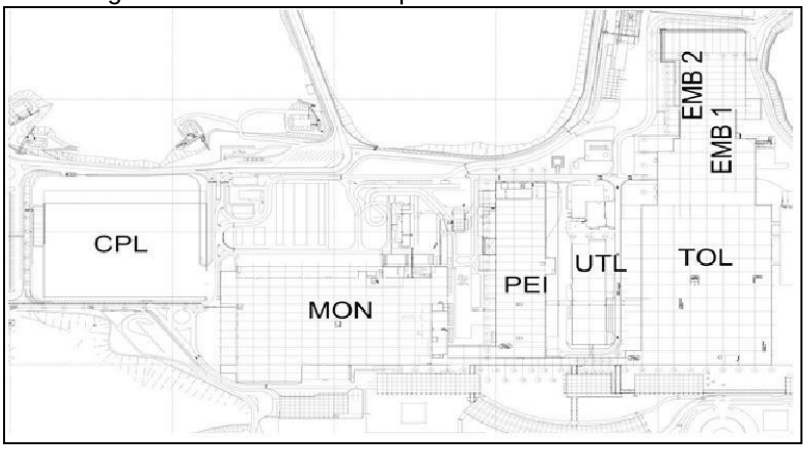
A Empresa ABC estudada está entre as cinco maiores montadoras automotivas, no âmbito mundial. O complexo se localiza em São José dos Pinhais/PR, e é composto por três fábricas, responsáveis pela produção de veículos de passeio, utilitários e motores.

De acordo com o relatório disponibilizado pela FENABRAVE, referente ao ano de 2015, a empresa foi detentora de 7,84% de participação do mercado de automóveis e 4,29% do mercado de utilitários. Com produção total superior a 160 mil automotores produzidos, ocupou a sexta posição do mercado brasileiro.

A montadora possui 32 plantas localizadas na América, Europa, Eurásia, Ásia e Oriente médio, com capacidade de atender 128 países distintos. Emprega 120 mil colaboradores no mundo e, aproximadamente, 6 mil em sua unidade no Brasil.

A unidade fabril escolhida é responsável pela produção de veículos de passeio (Figura 3.1), e possui quatro áreas de processos principais em sua cadeia produtiva: estamparia (EMB 1 + EMB2); carroceria (TOL); pintura (PEI); e, montagem (MON + CPL), sendo essa a fonte dos estudos a serem abordados, e a área secundária de utilidades (UTL) responsável pelos fluidos industriais.

Figura 3.1- Áreas do fluxo produtivo da unidade fabril.



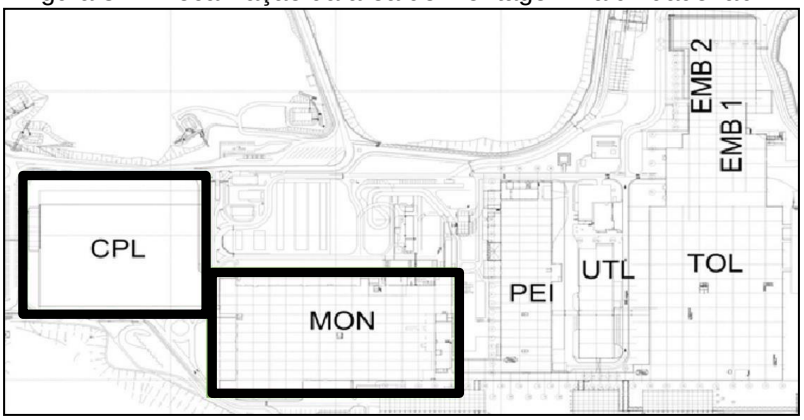
Fonte: Empresa ABC (2016).

Durante o processo da **estamparia**, chapas de aço são conformadas mecanicamente e transformadas em peças para a carroceria do automóvel. Em seguida, na **carroceria**, as peças oriundas do processo anterior são soldadas e dão forma ao carro e, assim, o produto parcial é encaminhado à **pintura**. Nesta fase, é adicionada a cobertura de tinta que fornece durabilidade e embelezamento ao veículo. Por fim, inicia-se a **montagem**, na qual adiciona-se à carroceria previamente pintada todos os acabamentos internos e externos, bancos, vidros, motor, luzes, etc. Assim, ao fim de todas essas etapas que incluem, também, verificações e testes de qualidade, o veículo é considerado concluído.

3.1.1 Área de montagem

A área de montagem compreende toda a superfície destinada ao processo propriamente dito e, também, a superfície utilizada para armazenamento de estoque logístico (CPL) de itens e peças (Figura 3.2).

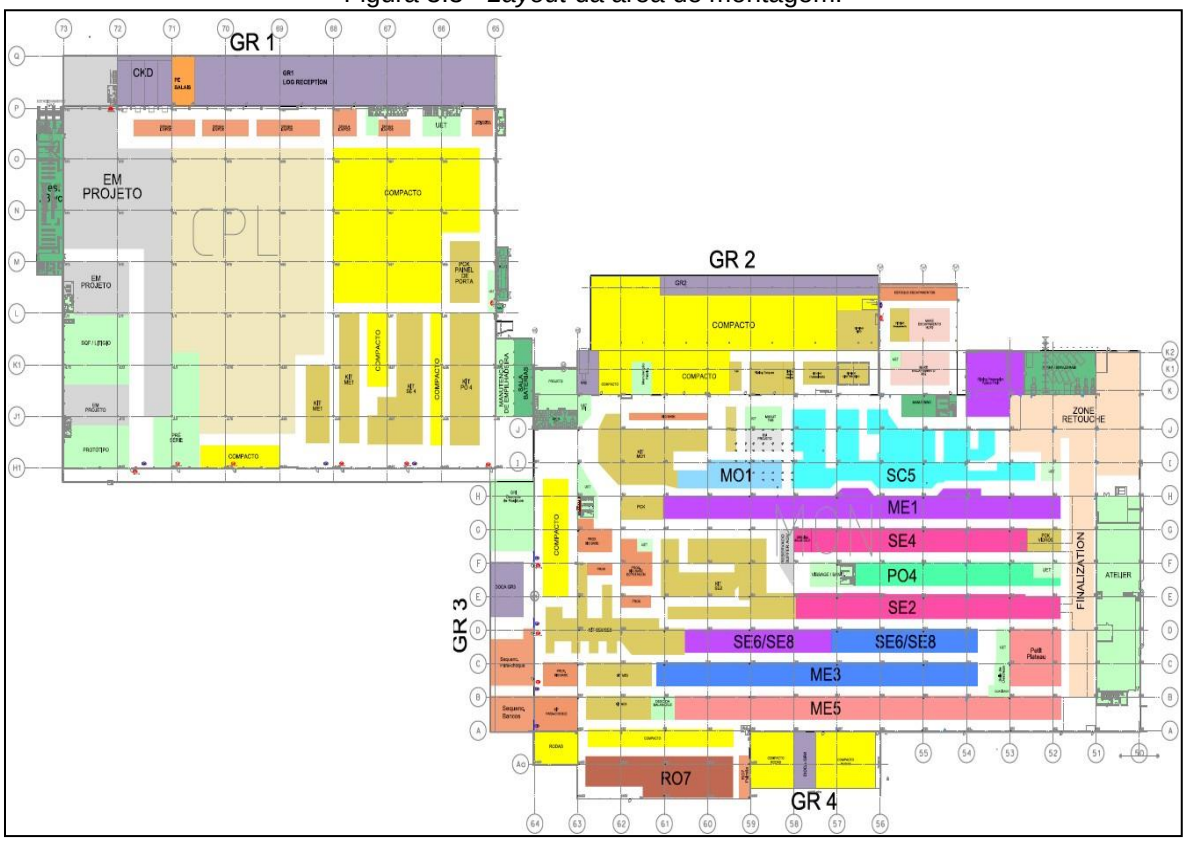
Figura 3.2 - Localização da área de montagem na unidade fabril.



Fonte: Empresa ABC (2016).

Na Figura 3.3 é possível verificar que a montagem inclui a linha principal (SC5, MO1, ME1, SE4, etc.), estoques, áreas de *kitting*, áreas de re-embalagens, áreas de processos complementares, etc.

Figura 3.3 - *Layout* da área de montagem.

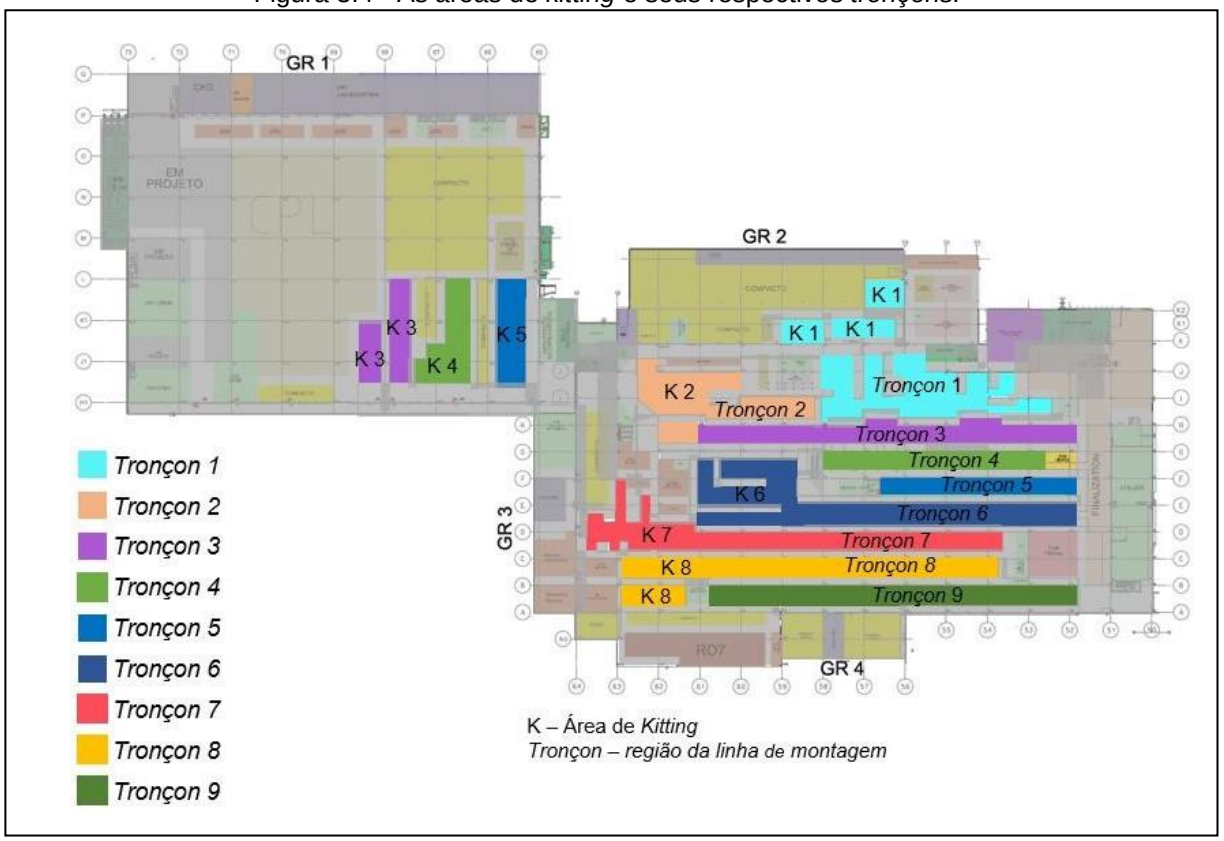


Fonte: Empresa ABC (2016).

3.1.2 Área de *kitting*

As áreas de *kitting* são definidas, dentro da empresa, como áreas destinadas ao agrupamento de itens e peças de uma mesma região (*tronçon*) da linha principal de montagem. Cada *tronçon* é responsável por realizar processos específicos como, por exemplo: *tronçon* responsável pela montagem do conjunto mecânico; *tronçon* responsável pela montagem de portas; *tronçon* responsável pela montagem de bancos e para-choques, etc. Portanto, para cada *tronçon,* exceto o *tronçon* 9, há uma área específica, que podem ser subdivididas em 20 áreas de *kitting*. A Figura 3.4 retrata essa relação.

Figura 3.4 - As áreas de *kitting* e seus respectivos *tronçons.*

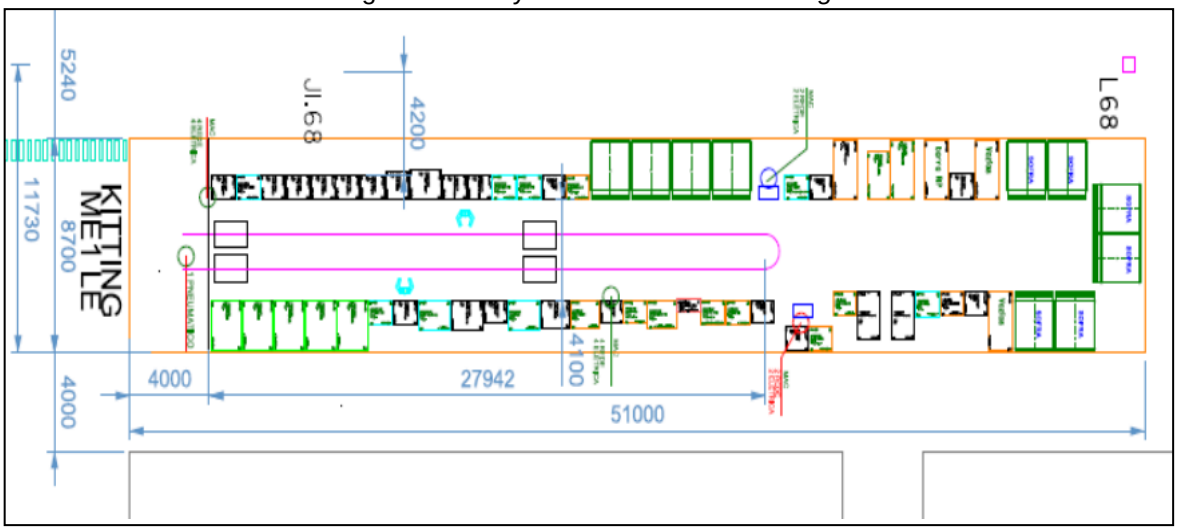


Fonte: Empresa ABC (2016).

De acordo com a “metodologia *Kaizen*” aplicada pela organização, a aplicação de áreas de *kitting* objetiva atender a premissa de que os operadores que geram valor agregado ao produto, ou seja, realizam o processo real de montagem do automóvel, concentrem-se apenas nestas atividades, tanto no quesito movimentação física quanto no quesito cognitivo. Dessa forma, toda e qualquer atividade de valor não agregado nestes postos de trabalho, tais como caminhar para buscar peças, abaixar para alcançar itens em níveis inferiores, ou até mesmo o ato de pensar para escolher o que pegar, devem ser eliminadas ou, ao menos, minimizadas.

Assim, as áreas de *kitting* envolvem a disposição de todas as referências de itens e peças de forma que os *kitteurs* - operadores responsáveis pela montagem dos “carrinhos de *kitting”* - possam acessá-las e acondicioná-las em carrinhos específicos. O percurso realizado pelo carrinho dentro da área acontece através de trilhos. A Figura 3.5 apresenta um exemplo de *layout* de área de *kitting*. Já a Figura 3.6 traz a ilustração de uma parte de área real.

Figura.3.5 - Layout de uma área de *kitting.*



Fonte: Empresa ABC (2016).

Figura 3.6 - Ilustração parcial da área de *kitting.*

Fonte: Empresa ABC (2016).

Cada carrinho é preenchido com uma coleção de peças de um mesmo veículo, que pode ser composta por itens de um ou mais postos de trabalho. A sua concepção é realizada com a intenção de atender toda a demanda de itens e peças, e de modo a permitir a mínima movimentação do operador da linha principal.

Após a montagem, o carrinho é direcionado, através de trilhos RGVs (*Rail Guided Vehicle*) ou com o suporte de AGVs (*Automated Guided Vehicle*), para a entrada na linha, onde acompanhará o veículo até o final da extensão do seu respectivo *tronçon*.

A localização das áreas de *kitting* é realizada de acordo com a disponibilidade de superfície livre e, normalmente, acaba por ser posicionada com certa distância de seu respectivo *tronçon*. A necessidade de superfície demandada para a concepção

de uma área de *kitting* é diretamente proporcional ao número de referências de itens e peças que devem ser ali acondicionadas, além do tipo de embalagem utilizado. Estima-se que 21% da superfície global destinada ao processo de montagem é ocupada por essas áreas.

As embalagens das diversas referências de itens e peças são basicamente divididas em dois tipos: embalagens pequenas, armazenadas em móveis gravitacionais, aqui chamados de *flowracks*; e, em embalagens grandes, armazenadas em demarcações no piso.

3.1.1.1 Armazenamento de Pequenas Embalagens (PE)

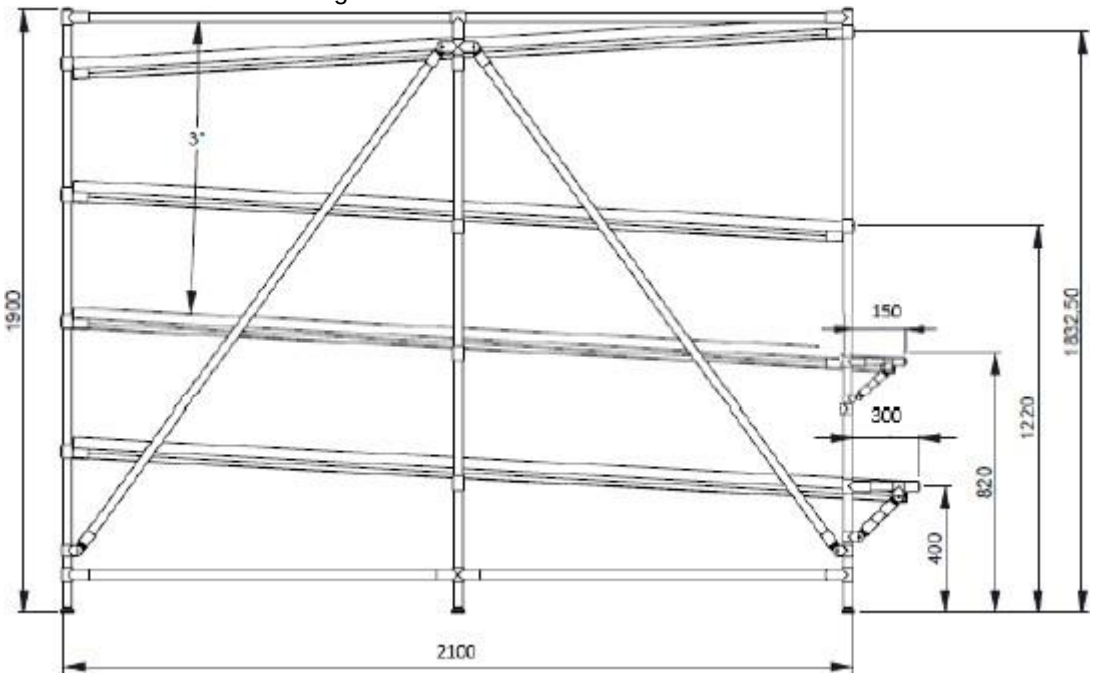
Pequenas Embalagens (PE) são todas as embalagens padronizadas que possuem dimensões menores que 1000 mm, e que podem ser manuseadas pelo operador de abastecimento sem a necessidade de equipamentos de suporte adicionais. A Figura 3.7 traz um exemplo de uma PE conhecida como ECM-1087 que possui dimensões de 1196mm x 328mm x 150mm (Comprimento x Largura x Altura).

Figura 3.7 - PE do tipo ECM-1087.

Fonte: Empresa ABC (2016).

Para o armazenamento de PEs, nas áreas de *kitting*, são utilizados móveis tubulares (*flowracks*), com prateleiras confeccionadas com pequenos rolos, não motorizados, que permitem a movimentação das embalagens por gravidade. Os *flowracks* são confeccionados com as dimensões necessárias para atender as mais variadas aplicações dentro da fábrica. A Figura 3.8 representa a visão lateral de um *flowrack* de dimensões 1.900mmx1.850mmx2.100mm (altura x largura x profundidade), já a Figura 3.9 ilustra um *flowrack* sendo utilizado na área de *kitting.*

Figura 3.8 - Visão lateral de um *flowrack.*



Fonte: Empresa (2016).

Figura 3.9 - *Flowrack* na área de *kitting*

Fonte: Empresa ABC (2016).

3.1.1.2 Armazenamento de Grandes Embalagens (GE)

As Grandes Embalagens (GE) são todas as embalagens que possuem dimensões maiores que 1000mm e que alocam peças não comportadas em PEs. As áreas de *kitting* ocupam espaços previamente demarcados no piso. A Figura 3.10

ilustra uma GE do tipo SLI – 0760, com dimensões iguais a 1000mm x 1200mm x 930mm.

Figura 3.10 - GE do tipo SLI-0760

Fonte: Empresa ABC (2016).

3.2 O PROBLEMA DE OTIMIZAÇÃO DE *FLOWRACKS*

Há duas situações a serem consideradas na otimização de *flowracks* na área de montagem. A primeira é relativa a falta de padrão na determinação de quantidades de *flowracks* a serem compradas durante a fase de projetos. A segunda é referente à necessidade de otimizar a alocação das PEs nas prateleiras do *flowrack*, a fim de reduzir o número de unidades presentes nas áreas de *kitting.*

Até o fim do primeiro semestre de 2017, a Empresa ABC trabalhará com uma gama de três veículos, que incluem diversas versões e modelos. Cada modelo de carro produzido requer a presença de referências de peças exclusivas ou compartilhadas nas áreas de *kitting*. A cada inserção de uma nova gama, é imprescindível a aquisição de novos *flowracks,* que comportarão as peças do futuro projeto a serem adicionadas à quantidade já existente. Atualmente, a mensuração da quantidade necessária a ser adquirida é feita de forma empírica, baseada na experiência dos analistas responsáveis pelas atividades.

O impacto da inserção das quantidades de peças referentes aos novos projetos pode ser visto na Tabela 3.1, a seguir. Nota-se que, somente em relação a PEs, há um aumento de 81,19% nas quantidades a serem alocadas em *flowracks* devido à inclusão dos projetos. A quantidade de *flowracks* da Empresa ABC, por sua vez, será acrescida de 53,76%.

Tabela 3.1 - Impacto da entrada de projetos nas quantidades de PE.

	Situação Atual	**Entrada dos projetos**
Quantidade de peças	1356	2457
Quantidade de *Flowracks*	134	206

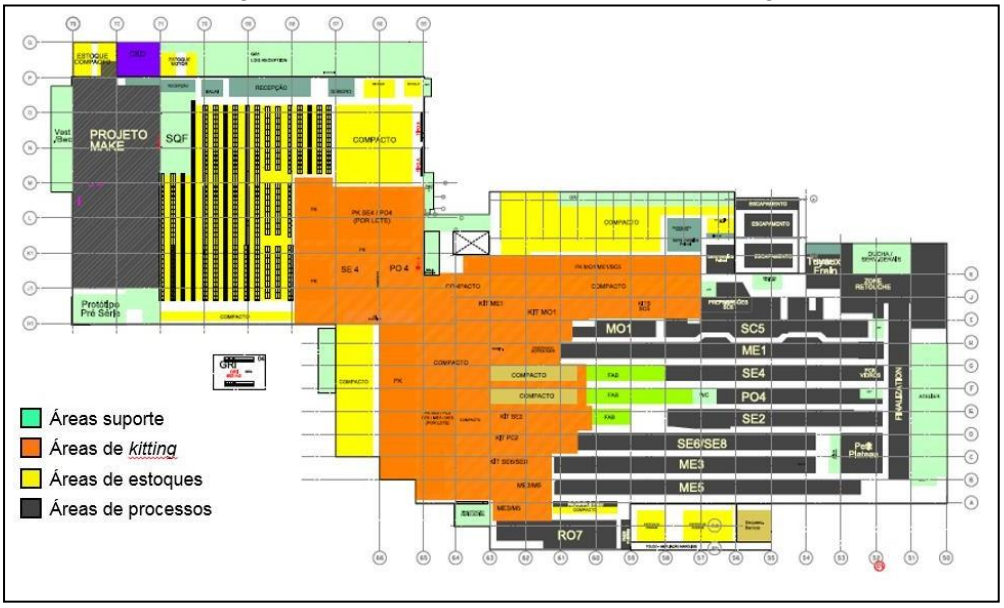
Fonte: Empresa ABC (2016).

A Figura 3.11 apresenta, na cor laranja, o atual comprometimento da área total de montagem com as áreas de *kitting*. Já a Figura 3.12 ilustra o cenário com a entrada de dois novos projetos na gama atual, caso não seja realizada qualquer tipo de otimização nas áreas de *kitting*.

Figura 3.11 – Cenário atual das áreas de *kitting*.

Fonte: Empresa ABC (2016).

Figura 3.12 - Cenário futuro das áreas de *kitting*.



Fonte: Empresa ABC (2016).

A partir da análise das Figura 3.11 e Figura 3.12, verifica-se que a demanda por área de *kitting* é maior do que a área destinada para tal, passando de 21% (18.450m²) para 26%(22.600m²) de superfície consumida. Dessa forma, seria necessário realizar construções prediais para ampliação da área de montagem, demandando investimentos de R$ 3.000,00/m², opção já descartada pela Empresa ABC. Assim, a aplicação de algoritmos de otimização vem de encontro ao interesse da organização de receber os projetos com o menor impacto possível.

4 METODOLOGIA

Conforme comentado no capítulo 2, pretende-se fazer uso de um método exato e das heurísticas FFD, MBS e MBS' para a otimização das áreas de *kitting* do problema apresentado no capítulo 3.

Todos os métodos foram executados em *notebook* provido de processador Intel® Core™ I7-5500U de 2,4GHz, 16GB de memória RAM, e Windows 10 Home como sistema operacional. Para os algoritmos FFD, MBS e MBS' foi utilizado o *software* Microsoft Visual Studio 2010® para implementação; já para o método exato utilizou-se o *software* IBM® ILOG® CPLEX® Interactive Optimizer 12.4.0.0, ferramenta para resolução de problemas de PLIB, mencionado na seção 2.1.1.

As dimensões utilizadas para um *flowrack* foram de 1900mmx1850mmx2100mm subdividida em três prateleiras. Logo, para cada prateleira, tida como *bin*, foi empregada a dimensão de 1850mm. Já para as referências de peças, consideradas como itens, foi adotada a menor dimensão da embalagem entre largura e comprimento.

As 20 áreas de *kitting* onde os testes de otimização foram realizados possuem nomes específicos dentro da Empresa ABC. Neste trabalho, foram denominadas de "AK" e receberam um número de identificação. A Tabela 4.1 apresenta as áreas de *kitting* com as respectivas quantidades de referências de peças, já com as peças atuais e as peças de projetos mescladas, bem como as quantidades estimadas de *flowracks*.

Tabela 4.1 - Quantidades de *flowracks* e referências de cada área de *kitting*.

ÁREA DE *KITTING*	**QUANTIDADE DE REFERÊNCIAS**	**QUANTIDADES ESTIMADAS**	
		FLOWRACKS	**PRATELEIRAS**
AK01	63	4	12
AK02	36	2	6
AK03	4	1	3
AK04	42	5	15
AK05	93	7	21
AK06	105	12	36
AK07	202	20	60
AK08	318	19	57
AK09	119	14	42
AK10	25	5	15
AK11	300	21	63
AK12	151	14	42
AK13	27	4	12

AK14	9	1	3
AK15	364	24	72
AK16	183	21	63
AK17	362	27	81
AK18	22	2	6
AK19	7	2	6
AK20	25	1	3
TOTAL	**2457**	**206**	**618**

Fonte: Empresa ABC (2016).

Para fins de comparação, os quatro métodos de otimização foram aplicados em cada área de *kitting*. Assim, foi possível verificar qual método se adequa melhor ao problema abordado. Avaliou-se a quantidade de *bins* gerada e a quantidade resultante de *flowracks*¸ respeitando a regra de 3 *bins* (prateleiras) por unidade.

Entretanto, os algoritmos abordados são incapazes de manterem as peças em famílias, isto é, respeitar as características de funcionalidade das peças. Frente a isso, e diante da fácil implementação e compreensão do método FFD, foi realizada a adaptação do algoritmo para que as peças de funcionalidade semelhantes permanecessem agrupadas e fossem destinadas a um mesmo *flowrack*. Através dessa adequação, foi possível atender aos requisitos cognitivos exigidos pelo setor de ergonomia da Empresa ABC, mencionadas na seção 3.1.2.

5 RESULTADOS

Neste capítulo os resultados obtidos através da aplicação dos métodos estudados serão mostrados e analisados. Primeiramente, o algoritmo de aproximação FFD é apresentado, seguido pelos métodos baseados em MBS (MBS e MBS') e pelo método exato. Por fim, o algoritmo adaptado proposto é exibido.

5.1 ALGORITMO DE APROXIMAÇÃO – *FIRST-FIT DECREASING*

Como mencionado por Almeida (2014) algumas das principais características do FFD são a facilidade de implementação e o baixo tempo computacional. Dessa forma, o algoritmo foi aplicado a todas as áreas de *kitting*. A Tabela 5.1 apresenta os tempos computacionais utilizados para execução do algoritmo, enquanto que a Tabela 5.2 traz os resultados obtidos após a implementação do método.

Tabela 5.1 – Tempos computacionais para o algoritmo FFD.

ÁREAS DE *KITTING*	**TEMPO COMPUTACIONAL (s)**
AK01	0,16
AK02	0,10
AK03	0,03
AK04	0,16
AK05	0,27
AK06	0,29
AK07	0,58
AK08	0,89
AK09	0,35
AK10	0,09
AK11	1,91
AK12	0,50
AK13	0,09
AK14	0,04
AK15	1,19
AK16	0,63
AK17	1,25
AK18	0,16
AK19	0,05
AK20	0,11
TOTAL	**8,83**

Fonte: autora (2017).

Tabela 5.2 - Resultados comparativos após a implementação do método FFD.

ÁREA DE *KITTING*	QUANTIDADE DE REFERÊNCIAS	QUANTIDADES ESTIMADAS		QUANTIDADES FFD	
		FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	63	4	12	4	10
AK02	36	2	6	2	5
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	38
AK08	318	19	57	19	56
AK09	119	14	42	10	30
AK10	25	5	15	2	5
AK11	300	21	63	21	62
AK12	151	14	42	12	34
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	23	67
AK16	183	21	63	11	33
AK17	362	27	81	23	67
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**163**	**464**

Fonte: autora (2017).

Na coluna nomeada “Quantidades estimadas” constam as quantidades estimadas de *flowracks* e prateleiras para comportar todas as peças - já existentes e de novos projetos - nas áreas de *kitting.* Já na coluna “Quantidade FFD”, na subdivisão “Prateleiras”, são apresentados os resultados obtidos a partir da execução do método, uma vez que cada unidade de prateleira é considerada como *bin* no problema. Para a obtenção dos valores da subdivisão “*flowracks"*, dividiu-se o número obtido de prateleiras pela capacidade de três prateleiras por unidade de *flowrack,* como já mencionado no capítulo 4.

Apresenta-se, na Tabela 5.3, os valores obtidos a partir da comparação entre as condições estimadas de cada área de *kitting* e os resultados da otimização. Na área AK04, por exemplo, houve a redução de 2 *flowracks* (40%) ou 7 prateleiras (46,67%).

Tabela 5.3 - Redução entre os valores estimados e o resultado da otimização do FFD.

ÁREA DE *KITTING*	REDUÇÃO (QUANTIDADE)		REDUÇÃO (%)	
	FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	0	2	0,00%	16,67%
AK02	0	1	0,00%	16,67%
AK03	0	2	0,00%	66,67%
AK04	2	7	40,00%	46,67%
AK05	1	5	14,29%	23,81%
AK06	6	20	50,00%	55,56%
AK07	7	22	35,00%	36,67%
AK08	0	1	0,00%	1,75%
AK09	4	12	28,57%	28,57%
AK10	3	10	60,00%	66,67%
AK11	0	1	0,00%	1,59%
AK12	2	8	14,29%	19,05%
AK13	2	7	50,00%	58,33%
AK14	0	1	0,00%	33,33%
AK15	1	5	4,17%	6,94%
AK16	10	30	47,62%	47,62%
AK17	4	14	14,81%	17,28%
AK18	0	2	0,00%	33,33%
AK19	1	4	50,00%	66,67%
AK20	0	0	0,00%	0,00%
TOTAL	**43**	**154**	**20,87%**	**24,92%**

Fonte: autora (2017).

Os valores obtidos para todas as áreas de *kitting* foram inferiores aos estimados, representando uma redução de 20,87% (43 unidades) na quantidade total de *flowracks* e 24,92% (154 unidades) no número de prateleiras. Dessa forma, esse caso contraria o argumento utilizado por Scholl *et al*. (1997), que prevê 22,2% de desvio do valor ótimo.

Por fim, na Tabela 5.4, apresenta-se o ganho, em área, a partir da redução de 43 *flowracks* nas 20 AK analisadas. É possível gerar 167,06m² de área livre total na área de *kitting*. Como mencionado na seção 3.2, o custo para a construção de 1 m² em área fabril da Empresa ABC é de R$ 3 mil, dessa forma é possível gerar uma economia teórica de aproximadamente R$ 500 mil. A economia é dita como “teórica” pois a geração de área livre não se dá de forma linear em uma única área concentrada. Dessa forma não é possível construir uma nova AK nestes 167,06m² gerados, entretanto é possível que as AK existentes sejam ampliadas com impactos menores.

Tabela 5.4 -Ganhos em área com a aplicação do método FFD.

ÁREA DE *KITTING*	GANHO DE ÁREA (m²)
AK01	0,00
AK02	0,00
AK03	0,00
AK04	7,77
AK05	3,89
AK06	23,31
AK07	27,20
AK08	0,00
AK09	15,54
AK10	11,66
AK11	0,00
AK12	7,77
AK13	7,77
AK14	0,00
AK15	3,89
AK16	38,85
AK17	15,54
AK18	0,00
AK19	3,89
AK20	0,00
TOTAL	**167,06**

Fonte: autora (2017).

5.2 MÉTODOS BASEADOS EM MBS

A seguir são analisados os resultados obtidos a partir da implementação do algoritmo MBS, proposto por Gupta e Ho (1999), e do MBS', proposto por Fleszar e Hindi (2002). Analogamente ao realizado com o FFD, para todas as áreas de *kitting* foram aplicados os dois algoritmos propostos.

5.1.1 *Minimum Bin Slack* (MBS)

A Tabela 5.5 apresenta os tempos computacionais obtidos com a implementação do algoritmo MBS. Apesar de tempos na casa de segundos, o valor total é 2,8 vezes maior que o valor total obtido pelo FFD. As quantidades obtidas para *flowracks* e prateleiras são mostradas na Tabela 5.6.

Tabela 5.5 – Tempos computacionais para o algoritmo MBS.

ÁREAS DE *KITTING*	TEMPO COMPUTACIONAL (s)
AK01	0,58
AK02	0,32
AK03	0,05

AK04	0,45
AK05	0,87
AK06	0,96
AK07	2,01
AK08	3,23
AK09	1,29
AK10	0,24
AK11	3,02
AK12	1,52
AK13	0,26
AK14	0,10
AK15	3,70
AK16	1,78
AK17	3,58
AK18	0,22
AK19	0,09
AK20	0,22
TOTAL	**24,48**

Fonte: autora (2017).

Tabela 5.6 - Resultados comparativos após a implementação do método MBS.

ÁREA DE *KITTING*	**QUANTIDADE DE REFERÊNCIAS**	**QUANTIDADES ESTIMADAS**		**QUANTIDADES MBS**	
		FLOWRACKS	**PRATELEIRAS**	***FLOWRACKS***	**PRATELEIRAS**
AK01	63	4	12	4	10
AK02	36	2	6	2	5
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	38
AK08	318	19	57	20	58
AK09	119	14	42	10	30
AK10	25	5	15	2	5
AK11	300	21	63	21	63
AK12	151	14	42	11	33
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	23	68
AK16	183	21	63	11	33
AK17	362	27	81	23	68
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**163**	**468**

Fonte: autora (2017).

Diferentemente do que aconteceu com o FFD, no qual todos os valores foram inferiores aos valores estimados, no MBS a AK08 apresentou um valor 5,26% superior

(Tabela 5.7). Dessa forma, é possível substituí-lo pelo valor estimado pela Empresa ABC (Tabela 5.8).

Tabela 5.7 - Redução entre os valores estimados e o resultado da otimização do MBS.

ÁREA DE *KITTING*	REDUÇÃO (QUANTIDADE)		REDUÇÃO (%)	
	FLOWRACKS	**PRATELEIRAS**	***FLOWRACKS***	**PRATELEIRAS**
AK01	0	2	0,00%	16,67%
AK02	0	1	0,00%	16,67%
AK03	0	2	0,00%	66,67%
AK04	2	7	40,00%	46,67%
AK05	1	5	14,29%	23,81%
AK06	6	20	50,00%	55,56%
AK07	7	22	35,00%	36,67%
AK08	**-1**	**-1**	**-5,26%**	**-1,75%**
AK09	4	12	28,57%	28,57%
AK10	3	10	60,00%	66,67%
AK11	0	0	0,00%	0,00%
AK12	3	9	21,43%	21,43%
AK13	2	7	50,00%	58,33%
AK14	0	1	0,00%	33,33%
AK15	1	4	4,17%	5,56%
AK16	10	30	47,62%	47,62%
AK17	4	13	14,81%	16,05%
AK18	0	2	0,00%	33,33%
AK19	1	4	50,00%	66,67%
AK20	0	0	0,00%	0,00%
TOTAL	**43**	**150**	**20,87%**	**24,27%**

Fonte: autora (2017).

Tabela 5.8 - Resultados comparativos com a exclusão do valor não-ótimo.

ÁREA DE *KITTING*	QUANTIDADE DE REFERÊNCIAS	QUANTIDADES ESTIMADAS		QUANTIDADES MBS	
		FLOWRACKS	**PRATELEIRAS**	***FLOWRACKS***	**PRATELEIRAS**
AK01	63	4	12	4	10
AK02	36	2	6	2	5
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	38
AK08	318	19	57	19	57
AK09	119	14	42	10	30
AK10	25	5	15	2	5
AK11	300	21	63	21	63
AK12	151	14	42	11	33
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	23	68

AK16	183	21	63	11	33
AK17	362	27	81	23	68
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**162**	**467**

Fonte: autora (2017).

Assim, verifica-se que o resultado do algoritmo MBS, com a devida substituição do valor não-ótimo, gera uma unidade a menos de *flowrack,* quando comparado ao FFD. A Tabela 5.9 apresenta os ganhos em área com a aplicação do método, resultando em economia teórica de aproximadamente R$ 510mil.

Tabela 5.9 -Ganhos em área com a aplicação do método MBS.

ÁREA DE *KITTING*	**GANHO DE ÁREA (m²)**
AK01	0,00
AK02	0,00
AK03	0,00
AK04	7,77
AK05	3,88
AK06	23,31
AK07	27,19
AK08	0,00
AK09	15,54
AK10	11,65
AK11	0,00
AK12	11,65
AK13	7,77
AK14	0,00
AK15	3,88
AK16	38,85
AK17	15,54
AK18	0,00
AK19	3,88
AK20	0,00
TOTAL	**170,94**

Fonte: autora (2017).

5.1.2 *Minimum Bin Slack'* (MBS')

Ao realizar a mesma correlação com os tempos computacionais executados pelo FFD, o MBS' se comporta analogamente ao método MBS, com valor 2,8 vezes superior ao FFD (Tabela 5.10).

Tabela 5.10 – Tempos computacionais para o algoritmo MBS'.

ÁREAS DE *KITTING*	TEMPO COMPUTACIONAL (s)
AK01	0,62
AK02	0,31
AK03	0,07
AK04	0,41
AK05	0,90
AK06	0,99
AK07	1,97
AK08	3,08
AK09	1,33
AK10	0,25
AK11	3,32
AK12	1,53
AK13	0,26
AK14	0,10
AK15	3,42
AK16	1,71
AK17	3,51
AK18	0,23
AK19	0,14
AK20	0,23
TOTAL	**24,36**

Fonte: autora (2017).

Dentre os três métodos analisados, o MBS' foi o que obteve o melhor resultado em quantidade de prateleiras (*bins*) e *flowracks*. Os valores obtidos pelo método proposto por Fleszar e Hindi (2002) podem ser visualizados na Tabela 5.11. Já a Tabela 5.13 apresenta as reduções obtidas ao se comparar às quantidades estimadas.

Tabela 5.11 - Resultados comparativos após a implementação do método MBS'.

ÁREA DE *KITTING*	QUANTIDADE DE REFERÊNCIAS	QUANTIDADES ESTIMADAS		QUANTIDADES MBS'	
		FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	63	4	12	4	10
AK02	36	2	6	2	5
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	38
AK08	318	19	57	19	57
AK09	119	14	42	10	30
AK10	25	5	15	2	5
AK11	300	21	63	21	62

AK12	151	14	42	11	32
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	23	67
AK16	183	21	63	11	32
AK17	362	27	81	23	67
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**162**	**462**

Fonte: autora (2017).

Tabela 5.12 - Redução entre os valores estimados e o resultado da otimização do MBS'.

ÁREA DE *KITTING*	**REDUÇÃO (QUANTIDADE)**		**REDUÇÃO (%)**	
	FLOWRACKS	**PRATELEIRAS**	***FLOWRACKS***	**PRATELEIRAS**
AK01	0	2	0,00%	16,67%
AK02	0	1	0,00%	16,67%
AK03	0	2	0,00%	66,67%
AK04	2	7	40,00%	46,67%
AK05	1	5	14,29%	23,81%
AK06	6	20	50,00%	55,56%
AK07	7	22	35,00%	36,67%
AK08	0	0	0,00%	0,00%
AK09	4	12	28,57%	28,57%
AK10	3	10	60,00%	66,67%
AK11	0	1	0,00%	1,59%
AK12	3	10	21,43%	23,81%
AK13	2	7	50,00%	58,33%
AK14	0	1	0,00%	33,33%
AK15	1	5	4,17%	6,94%
AK16	10	31	47,62%	49,21%
AK17	4	14	14,81%	17,28%
AK18	0	2	0,00%	33,33%
AK19	1	4	50,00%	66,67%
AK20	0	0	0,00%	0,00%
TOTAL	**44**	**156**	**21,36%**	**25,24%**

Fonte: autora (2017).

O método MBS' se comportou 21,36% (44 unidades) melhor, se comparado à quantidade de *flowracks* estimada pela Empresa ABC, e 25,24% (156 unidades) se comparado ao número de prateleiras. Quando confrontado com o método FFD apresentou superioridade de 0,61% (1 unidade) na quantidade de *flowracks* e 0,43% (2 unidades) na quantidade de prateleiras. Já na comparação com o MBS, foi 1,07% (5 unidades) inferior na quantidade de prateleiras, comprovando a conclusão gerada

por Fleszar e Hindi (2002) que defendiam que o MBS' possuía um comportamento melhor que o MBS.

De forma semelhante ao realizado no tópico 4, a Tabela 5.13 apresenta os ganhos em área livre com a otimização das quantidades de *flowracks*. Com a redução de 44 unidades, é possível gerar uma economia teórica de aproximadamente R$ 510 mil.

Tabela 5.13 -Ganhos em área com a aplicação do método MBS'.

ÁREA DE *KITTING*	**GANHO DE ÁREA (m²)**
AK01	0,00
AK02	0,00
AK03	0,00
AK04	7,77
AK05	3,89
AK06	23,31
AK07	27,20
AK08	0,00
AK09	15,54
AK10	11,66
AK11	0,00
AK12	11,66
AK13	7,77
AK14	0,00
AK15	3,89
AK16	38,85
AK17	15,54
AK18	0,00
AK19	3,89
AK20	0,00
TOTAL	**170,94**

Fonte: autora (2017).

5.3 MÉTODO EXATO

Para a otimização através deste método foi utilizada a formulação matemática proposta por Martello e Toth (1990). Carvalho (2002) menciona as dificuldades na aplicação de algoritmos exatos para resolução de problemas de grandes instâncias. A fim de reduzir o tempo computacional, utilizou-se como solução inicial os resultados obtidos com o método MBS' (Tabela 5.11). A Tabela 5.14 apresenta os tempos computacionais obtidos após a resolução do método para todas as áreas de *kitting.*

Tabela 5.14 – Tempos computacionais para o algoritmo exato.

ÁREAS DE *KITTING*	TEMPO COMPUTACIONAL (s)
AK01	0,30
AK02	0,03
AK03	0,00
AK04	0,80
AK05	1,01
AK06	0,05
AK07	67.898,26
AK08	21.766,94
AK09	0,14
AK10	0,02
AK11	39.224,13
AK12	10.229,83
AK13	0,02
AK14	0,02
AK15	37.316,67
AK16	38.100,91
AK17	51.603,61
AK18	0,03
AK19	0,00
AK20	0,03
TOTAL	**266.142,80**

Fonte: autora (2017).

Foram necessárias 73,93h para que o modelo fosse rodado por completo. O valor é 10.926 vezes maior que o tempo necessário para o MBS', método de melhor desempenho até então. Em todos os casos em que o algoritmo excedeu 10h de execução, o resultado foi obtido após o esgotamento da memória RAM do computador. Entretanto, é possível verificar na Tabela 5.15 que as áreas de *kitting* AK15, AK16 e AK17 não obtiveram soluções inteiras até o esgotamento da memória.

Tabela 5.15 - Resultados comparativos após a implementação do método exato.

ÁREA DE *KITTING*	QUANTIDADE DE REFERÊNCIAS	QUANTIDADES ESTIMADAS		QUANTIDADES MÉTODO EXATO	
		FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	63	4	12	4	10
AK02	36	2	6	2	5
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	38
AK08	318	19	57	19	56
AK09	119	14	42	10	30

AK10	25	5	15	2	5
AK11	300	21	63	21	62
AK12	151	14	42	11	32
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	-	s/solução
AK16	183	21	63	-	s/solução
AK17	362	27	81	-	s/solução
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**162**	**295**

Fonte: autora (2017).

A Figura 5.1 mostra o momento em que a AK15 teve o esgotamento de memória, não resultando em solução inteira viável.

Figura 5.1 - Mensagem de esgotamento de memória.

```
Nodefile size = 156593.55 MB (30208.20 MB after compression)
 11126593 11115075       65.2627    83                    64.8243 1.29e+008
 11139765 11128243       65.7562    84                    64.8243 1.30e+008
 11153262 11141740       65.4276    78                    64.8243 1.30e+008
 11166572 11155050       65.1822    88                    64.8243 1.30e+008
 11179356 11167821       65.8011    87                    64.8243 1.30e+008
 11192745 11181197       65.5886    74                    64.8243 1.30e+008

There may be further error information in the clone logs.

Root node processing (before b&c):
  Real time             =    4.83
Parallel b&c, 4 threads:
  Real time             = 37309.84
  Sync time (average)   =    1.14
  Wait time (average)   =    0.51
                          -------
Total (root+branch&cut) = 37314.67 sec.
Warning: MIP starts not constructed because of out-of-memory status.

Consider using CPLEX node files to reduce memory usage.
CPLEX Error  1001: Out of memory.


MIP - Error termination, no integer solution.
Current MIP best bound = 6.4824324324e+001 (gap is infinite)
Solution time = 37316.67 sec.  Iterations = 130310334  Nodes = 11204552 (11191152)
Deterministic time = 13448324.96 ticks  (360.38 ticks/sec)
```

Fonte: autora (2017).

A fim de permitir a comparação de resultados, as três áreas foram excluídas. A Tabela 5.16 mostra a comparação entre os resultados estimados pela empresa e os obtidos pelo método exato, para as 17 áreas de *kitting*.

Tabela 5.16 - Redução entre os valores estimados e o resultado da otimização do método exato.

ÁREA DE *KITTING*	**REDUÇÃO (QUANTIDADE)**		**REDUÇÃO (%)**	
	FLOWRACKS	**PRATELEIRAS**	***FLOWRACKS***	**PRATELEIRAS**
AK01	0	2	0,00%	16,67%
AK02	0	1	0,00%	16,67%
AK03	0	2	0,00%	66,67%

AK04	2	7	40,00%	46,67%
AK05	1	5	14,29%	23,81%
AK06	6	20	50,00%	55,56%
AK07	7	22	35,00%	36,67%
AK08	0	1	0,00%	1,75%
AK09	4	12	28,57%	28,57%
AK10	3	10	60,00%	66,67%
AK11	0	1	0,00%	1,59%
AK12	3	10	21,43%	23,81%
AK13	2	7	50,00%	58,33%
AK14	0	1	0,00%	33,33%
AK18	0	2	0,00%	33,33%
AK19	1	4	50,00%	66,67%
AK20	0	0	0,00%	0,00%
TOTAL	**29**	**107**	**21,64%**	**26,62%**

Fonte: autora (2017).

Para as 17 áreas de *kitting* analisadas houve redução de 21,64% (29 unidades) e 26,62% (107 unidades) nas quantidades de *flowracks* e prateleiras, respectivamente. Se comparado ao MBS', o método exato gera a mesma quantidade de *flowracks* e 1 unidade a menos de prateleiras. Dessa forma, apenas para a quantidade de AKs analisadas, é possível gerar 112,67m² de área livre e uma economia teórica de aproximadamente R$330 mil (Tabela 5.17).

Tabela 5.17 -Ganhos em área com a aplicação do método exato.

ÁREA DE *KITTING*	**GANHO DE ÁREA (m²)**
AK01	0,00
AK02	0,00
AK03	0,00
AK04	7,77
AK05	3,89
AK06	23,31
AK07	27,20
AK08	0,00
AK09	15,54
AK10	11,66
AK11	0,00
AK12	11,66
AK13	7,77
AK14	0,00
AK18	0,00
AK19	3,89
AK20	0,00
TOTAL	**112,17**

Fonte: autora (2017).

5.4 PROPOSTA DE ADAPTAÇÃO DE ALGORITMO

A partir dos resultados e conclusões obtidos anteriormente e da necessidade de se agrupar as peças por famílias, foi proposta uma adaptação do algoritmo FFD. A escolha deste algoritmo como ponto de partida se deu pela facilidade de compreensão do seu funcionamento e também pelo baixo tempo computacional apresentado. O Quadro 5.1, abaixo, exemplifica as referências das peças e as famílias da AK18.

Quadro 5.1 – Exemplo de famílias de peças da AK18.

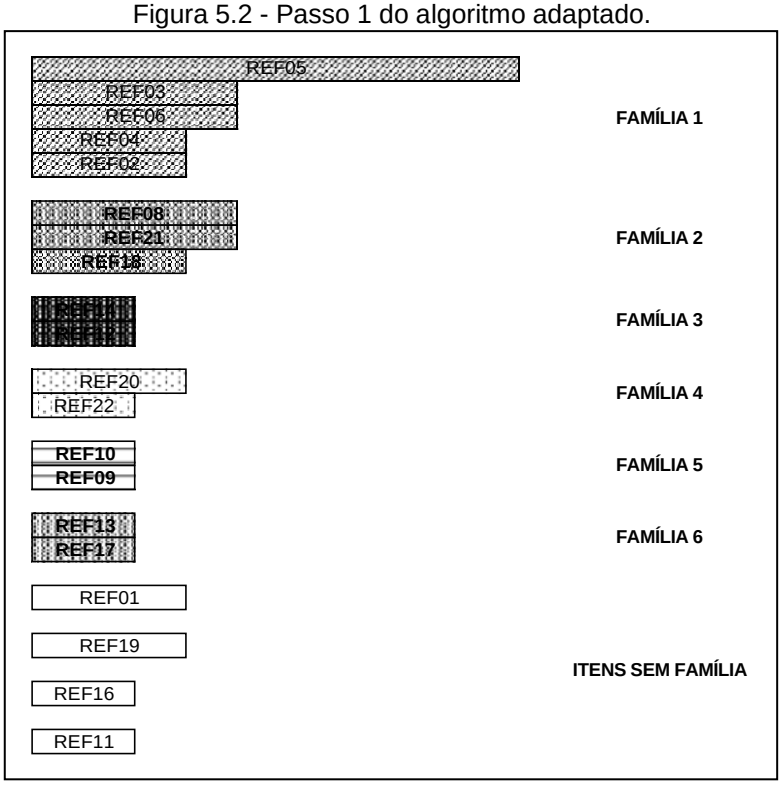
REFERÊNCIA	FAMÍLIA	DESCRIÇÃO	DIMENSÃO (mm)
REF02	F10480---	9B-SENSOR VELOCIDADE	300
REF03	F10480---	AN-SENSOR VELOCIDADE	400
REF04	F10480---	XC-SENSOR VELOCIDADE	300
REF05	F10480---	UG-CAPTOR VELOC ROD	950
REF06	F10480---	PD-SENSOR VELOCIDADE	400
REF01	F10711---	99-FLEXIVEL FREIO	300
REF08	F15655---	PX-APOIO INFER MOLA	400
REF18	F15655---	6H-APOIO INFER MOLA	300
REF21	F15655---	YJ-APOIO INFER MOLA	400
REF12	F15660---	JJ-PORCA SEM CROMO	200
REF14	F15660---	KB-PORCA SUPORTE	200
REF20	F15661---	ZL-TAMPA TAMBOR RODA	300
REF22	F15661---	F9-TAMPA TAMBOR RODA	200
REF11	L003584--	UC-PARAFUSO	200
REF16	L005423--	5H-GRAMPO SUPORTE	200
REF19	L006180--	47-GRAMPO ARAME SCR6	300
REF09	L007761--	FN-PARAFUSO	200
REF10	L007761--	H1-PARAFUSO M10	200
REF13	L007765--	KA-PORCA SEM CROMO	200
REF17	L007765--	75-PARAFUSO M12X	200
REF07	L007770--	G6-PARAFUSO	200
REF15	L020372--	8P-GRAMPO SUPORTE 2	200

Fonte: autora (2017).

Como mencionado, o método proposto possui como base o algoritmo FFD, dessa forma a designação dos itens segue o processo descrito por Junqueira (2007) na seção 2.3.1.1. O passo-a-passo do funcionamento do algoritmo é descrito a seguir:

1. Dentro da própria família os itens são ordenados pela dimensão de forma decrescente (Figura 5.2);

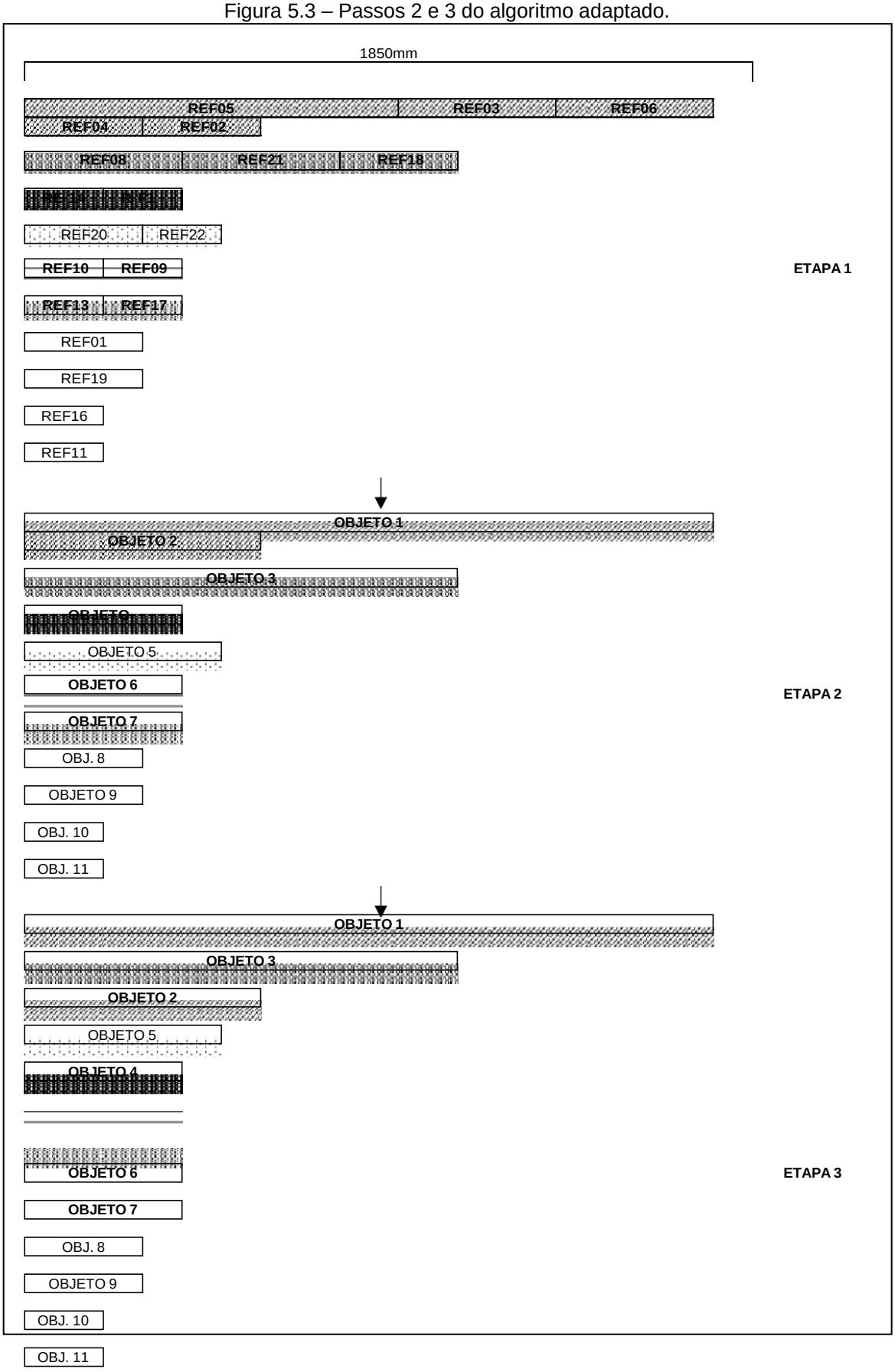
Figura 5.2 - Passo 1 do algoritmo adaptado.



Fonte: autora (2017).

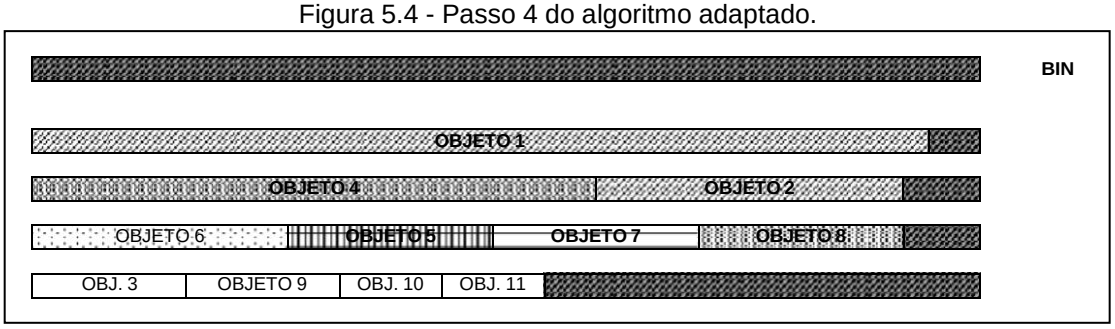
2. A partir desta ordenação, os itens são designados a *bins* exclusivos de suas famílias, respeitando a dimensão do *bin* de 1.850mm (Etapa 1 - Figura 5.3). Após a designação, este *bin* recebe o nome de “objeto” (Etapa 2 - Figura 5.3);
3. Os objetos são novamente reordenados de forma decrescente (Etapa 3 – Figura 25);
4. Por fim, os objetos são designados aos *bins* (Figura 5.4).

Figura 5.3 – Passos 2 e 3 do algoritmo adaptado.



Fonte: autora (2017).

Figura 5.4 - Passo 4 do algoritmo adaptado.



Fonte: autora (2017).

Este método foi aplicado nas 20 áreas de *kitting* e para fins de diferenciação foi denominado “FFD adaptado”. Possui tempo computacional 54,51% superior que seu método base, FFD, e 44,01% inferior se comparado aos métodos MBS e MBS’. A Tabela 5.18 traz os tempos para cada AK e a Tabela 5.19 mostra os valores comparativos entre o existente na Empresa ABC e os obtidos após a implementação do algoritmo adaptado.

Tabela 5.18 – Tempos computacionais para FFD adaptado.

ÁREAS DE *KITTING*	**TEMPO COMPUTACIONAL (s)**
AK01	0,33
AK02	0,18
AK03	0,04
AK04	0,18
AK05	0,34
AK06	0,36
AK07	0,59
AK08	1,32
AK09	0,37
AK10	0,17
AK11	1,49
AK12	0,66
AK13	0,15
AK14	0,08
AK15	2,16
AK16	0,83
AK17	4,11
AK18	0,13
AK19	0,06
AK20	0,09
TOTAL	**13,64**

Fonte: autora (2017).

Tabela 5.19 - Resultados comparativos após a implementação do método FFD adaptado.

ÁREA DE *KITTING*	QUANTIDADE DE REFERÊNCIAS	QUANTIDADES ESTIMADAS		QUANTIDADES FFD ADAPTADO	
		FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	63	4	12	4	10
AK02	36	2	6	2	6
AK03	4	1	3	1	1
AK04	42	5	15	3	8
AK05	93	7	21	6	16
AK06	105	12	36	6	16
AK07	202	20	60	13	37
AK08	318	19	57	19	56
AK09	119	14	42	10	30
AK10	25	5	15	2	5
AK11	300	21	63	20	60
AK12	151	14	42	12	34
AK13	27	4	12	2	5
AK14	9	1	3	1	2
AK15	364	24	72	23	67
AK16	183	21	63	11	33
AK17	362	27	81	23	68
AK18	22	2	6	2	4
AK19	7	2	6	1	2
AK20	25	1	3	1	3
TOTAL	**2457**	**206**	**618**	**162**	**463**

Fonte: autora (2017).

Este método apresenta redução de 21% e 25% na quantidade de *flowracks* (44 unidades) e de prateleiras (155 unidades), respectivamente (Tabela 5.20), se comparado ao valor estimado pela empresa. Ao se comparar ao FFD, a versão adaptada retorna uma unidade a menos de *flowrack* e prateleira. Já se comparado com o método de melhor desempenho, MBS', o algoritmo proposto apresenta a mesma quantidade de *flowracks* e uma unidade a mais de prateleiras (Figura 5.5).

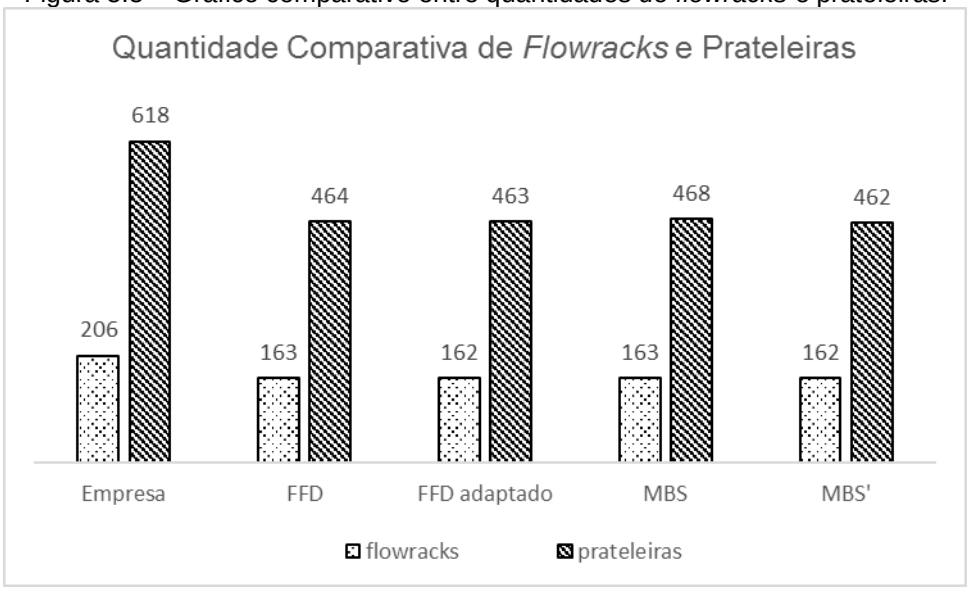
Tabela 5.20 - Redução entre os valores estimados e o resultado da otimização do FFD adaptado.

ÁREA DE *KITTING*	REDUÇÃO (QUANTIDADE)		REDUÇÃO (%)	
	FLOWRACKS	PRATELEIRAS	*FLOWRACKS*	PRATELEIRAS
AK01	0	2	0,00%	16,67%
AK02	0	0	0,00%	0,00%
AK03	0	2	0,00%	66,67%
AK04	2	7	40,00%	46,67%
AK05	1	5	14,29%	23,81%
AK06	6	20	50,00%	55,56%
AK07	7	23	35,00%	38,33%
AK08	0	1	0,00%	1,75%
AK09	4	12	28,57%	28,57%

AK10	3	10	60,00%	66,67%
AK11	1	3	4,76%	4,76%
AK12	2	8	14,29%	19,05%
AK13	2	7	50,00%	58,33%
AK14	0	1	0,00%	33,33%
AK15	1	5	4,17%	6,94%
AK16	10	30	47,62%	47,62%
AK17	4	13	14,81%	16,05%
AK18	0	2	0,00%	33,33%
AK19	1	4	50,00%	66,67%
AK20	0	0	0,00%	0,00%
TOTAL	**44**	**155**	**21,36%**	**25,08%**

Fonte: autora (2017).

Figura 5.5 – Gráfico comparativo entre quantidades *de flowracks* e prateleiras.



Fonte: autora (2017).

Por fim, a Tabela 5.21 apresenta os ganhos em área com a aplicação do método proposto, na qual verifica-se que a redução em 44 unidades de *flowracks* gera ganho de 170,94m², resultando em economia teórica de aproximadamente R$ 510 mil.

Tabela 5.21 -Ganhos em área com a aplicação do FFD adaptado.

ÁREA DE *KITTING*	GANHO DE ÁREA (m²)
AK01	0,00
AK02	0,00
AK03	0,00
AK04	7,77
AK05	3,89
AK06	23,31
AK07	27,20
AK08	0,00
AK09	15,54

AK10	11,66
AK11	3,89
AK12	7,77
AK13	7,77
AK14	0,00
AK15	3,89
AK16	38,85
AK17	15,54
AK18	0,00
AK19	3,89
AK20	0,00
TOTAL	**170,94**

Fonte: autora (2017).

O Apêndice A apresenta o compilado de todos os resultados obtidos.

6 CONCLUSÕES E DIRECIONAMENTOS

A crise econômica brasileira vivenciada nos últimos tempos estimulou as empresas a buscar novas alternativas para solução problemas sem a geração de novos custos. A Empresa ABC, estudada neste trabalho, vivenciava um momento de contradição: a necessidade de aumentar seu *mix* de automóveis sem liberação de verba para ampliação predial da fábrica.

Este trabalho surgiu a partir da identificação da oportunidade de gerar um método de padronização para a determinação de quantidades de *flowracks* a serem instalados nas áreas de *kitting*, que são diretamente impactadas pela inserção de novos projetos; e, da necessidade de otimizar a disposição das PEs nas prateleiras dos *flowracks*, a fim de reduzir o número de unidades.

Primeiramente, foi proposta a utilização de um método de fácil e rápida implementação. O FFD reduziu em 20,87% a quantidade de *flowracks* (43 unidades), e em 24,92% o número de prateleiras necessárias para alocar todas as PEs (154 unidades), se comparado ao número estimado pela empresa. Com relação a área consumida, a redução gera 167,06 m² de área livre, resultando em economia teórica de aproximadamente R$500 mil. Diante do baixíssimo tempo computacional, na casa de segundos, e a possibilidade de compreensão total do funcionamento do método por parte dos analistas, considera-se um método plausível para o fim ao qual é destinado.

Na sequência, foram aplicados os métodos MBS e MBS', ambos com tempos computacionais superiores ao FFD, porém ainda na casa dos segundos. Diferentemente do ocorrido com o FFD, o MBS apresentou valores superiores aos estimados pela empresa, portanto estes foram mantidos. Ambos os métodos geraram redução de 21,36% na quantidade de *flowracks* (44 unidades), Em relação ao número de prateleiras – quando comparado com o valor estimado - o MBS gerou 24,43% de redução (151 unidades a menos) e o MBS' apresentou queda de 24,24% (156 unidades a menos). Assim, obteve-se uma economia teórica de aproximadamente R$510 mil.

Para fins de comparação entre os métodos, foi realizada a implementação da formulação matemática do BPP. Dentre as 20 áreas de *kitting* analisadas, foi possível obter resultados para 17. Para as demais, não foi obtida uma solução inteira viável, devido ao esgotamento da memória. Para as 17 áreas, houve redução de 21,64% (29

unidades) e 26,62% (107 unidades) nas quantidades de *flowracks* e prateleiras, respectivamente. Se comparado ao método de melhor resultado, o MBS', para a mesma quantidade de AKs, diferencia-se apenas em uma unidade de prateleiras, em que o método exato se mostra superior.

Diante das conclusões mencionadas anteriormente, adicionou-se ao problema mais um agente de decisão: o fato de que as peças deveriam ser designadas conforme suas famílias, de forma a tender a ficarem agrupadas. Desta maneira, foi proposta uma adaptação do algoritmo FFD, que proporcionaria o agrupamento das referências por família. Após a implementação desta adaptação, obteve-se resultados bastantes similares aos gerados pelo MBS', diferenciando-se apenas por uma unidade a mais de prateleiras, aliado ao ponto positivo de se ter tempos computacionais excelentes.

Frente ao exposto e diante das necessidades apresentadas pelo problema-alvo deste trabalho, conclui-se que o método que mais se adequa às características e fornece resultados condizentes à realidade fabril é o método denominado FFD adaptado, que possibilita uma economia teórica de aproximadamente R$510 mil ao se reduzir 44 unidades de *flowracks*, ao mesmo tempo que não prejudica o quesito cognitivo da operação.

Frente as conclusões, propõe-se alguns direcionamentos futuros:

- Aprofundamento de outros métodos para a resolução do BPP unidimensional encontrado na literatura, a partir da pressuposição de adaptação do método ao agrupamento de referências por famílias;
- Análise da aplicação destes métodos, e do método proposto, em outros problemas reais.

REFERÊNCIAS

ABIDI, S.; KRICHEN, S.; ALBA, E.; MOLINA, J. M. Improvement heuristic for solving the one-dimensional bin-packing problem, **5th International Conference on Modeling, Simulation and Applied Optimization**, Hammamet, 2013.

ABUABARA, A.; MORABITO, R. Cutting optimization of structural tubes to build agricultural lights aircrafts, **Annals of Operation Research**, v. 169, p. 149-165, 2009.

ALMEIDA, R. **Proposta de um método meta-heurístico híbrido para resolução de problemas do tipo bin packing**, 114 p., Dissertação (Mestrado em Engenharia de Produção) – Pontifícia Universidade Católica do Paraná, PR, 2014.

ALOISE, D. J. **Contribuições à solução do problema bin-packing: formulações, relaxações e novos algoritmos aproximativos.**165 p., Tese (Doutorado em Ciências em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, RJ, 1992.

ALVIN, A. C. F **Uma heurística híbrida de melhoria para o problema de bin packing pae sua aplicação ao problema de escalonamento de tarefas**, 136 p., Tese (Doutorado em Informática) – Pontifícia Universidade Católica do Rio de Janeiro, RJ, 2003.

ANFAVEA. **Anuário da Indústria Automobilística Brasileira.** Disponível em: http://www.anfavea.com.br/anuario.html

BALAS, E. An additive algorithm for solving linear programs with zero-one variables, **Operations Research**, v. 13, p. 517546, 1965.

BERTO, R. M. V. S.; NAKANO, D. N. Metodologia da pesquisa e a Engenharia de Produção, **in: Encontro Nacional de Engenharia de Produção**, Niterói, 1998.

BRANDÃO, F.; PEDROSO, J. P. Bin packing and related problems: general arc-flow formulation with graph compression, **Computers Operations Research**, v. 69, p. 56-67, 2016.

BULJUBAŠIĆ, M.; VASQUEZ, M. Consistent neighborhood search for one-dimensional bin packing and two-dimensional vector packing, **Computers & Operations Research**, v. 76, p. 12-21, 2016.

CARVALHO, J. M. V. LP models for bin packing and cutting stock problems, **European Journal of Operational Research**, v. 141, p. 253-273, 2002.

CASTELLANOS, M. Q.; REYES, L.C; JIMENEZ, J. T.; GÓMEZ, C.S.; HUACUJA, H. J. F.; ALVIM, A. C. F. A grouping genetic algorithm with controlled gene transmission for the bin packing problem, **Computers & Operations Research**, v. 55, p. 52-64, 2015.

COFFMAN JR., E.G.; GAREY, M.R.; JOHNSON, D.S. **Approximation algorithms for bin-packing: a survey**. In: Hochbaum, D., editor, APPROXIMATION ALGORITHMS FOR NP-HARD PROBLEMS, p. 46-93. PWS Publishing, 1997.

CUNHA, C. B.; MAINIERI, G.; YOSHIZAKI, H. T. Y.; MALUTA, L.; HENRIQUES, L R. S. Heurísticas para o problema de bin packing no contexto da distribuição física de produtos, **in: XL Simpósio Brasileiro de Pesquisa Operacional**, João Pessoa, 2008.

CUI, Y.; YANG, Y. A heuristic for the one-dimensional cutting stock problem with usable leftover, **European Journal of Operational Research**, v. 204, p. 245-250, 2010.

DOKEROGLU, T.; COSAR, A. Optimization of one-dimensional bin packing problem with island. Paralllel grouping genetic algorithms, **Computers & Industrial Engineering**, v. 75, p. 176-186, 2014.

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

EILON, S.; CHRISTOFIDES, N. The loading problem, **Management Science**, v.17, p. 259-268, 1971.

FENABRAVE. Índices e números de emplacamento de Janeiro/16. Disponível em: http://www3.fenabrave.org.br:8082/plus/modulos/listas/index.php?tac=indices-e-numeros&idtipo =1&layout=indices-e-numeros

FLESZAR, K.; HINDI, K. S. New heuristics for one-dimensional bin-packing, **Computers & Operations Research**, v. 29, p. 821-839, 2002.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4.ed. São Paulo: Atlas, 2002.

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

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem-Part II, **Operations Research**, v. 11, p. 863-888, 1963.

GUPTA J. N. D.; HO, J. C. A new heuristic algorithm for the one-dimensional bin-packing problem, **Production Planning & Control**, v. 10, p. 598-603, 1999.

HAN, S.Y.; WAN, X.Y.; WANG, L.; ZHOU, J. Comparison between genetic algorithm and differential evolution algorithm applied too ne dimensional bin-packing problem, **in: 3rd International Conference on Informative and Cybernetics for Computational Social Systems**, Jinzhou, 2016.

HUNG. M.; BROWN, J. Na algorithm for a class of loading problems, **Naval Research Logistics Quarterly**, v.17, p. 289-297, 1978.

JUNQUEIRA, N. M. P. **Algoritmos aproximados para solucionar o problema de bin packing unidimensional**, 137 p., Dissertação (Mestrado em Ciências) – Instituto Tecnológico de Aeronáutica, SP, 2007.

JOHNSON, D. S. Near-optimal bin packing algorithms, 401 p., Tese (Doutorado em Matemática) – Massachusetts Institute of Technology, EUA, 1973.

KANTOROVICH, L. V. Mathematical methods of organizing and planning production, **Management Science**, v.6, p. 366-422, 1960.

LODI, A.; MARTELLO, S.; VIGO, D. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems, **Journal on Computing**, v. 11, p. 345-357, 1999.

MARTELLO, S.; TOTH, P. **Knapsack problems: algorithms and computer implementations.** John Wiley & Sons, Cichester, 1990.

MARTÍNEZ, D. A. **Estudo dos problemas de corte e empacotamento**, 163 p., Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual Paulista, SP, 2014.

MORABITO, R.; GARCIA, V. The cutting stock problem in a hardboard industry: a case study, **Computers Operations Research**, v. 25, p. 469-485, 1998.

MOSQUERA, G. P. **Contribuições para o Problema de Corte de Estoque Bidimensional na Indústria Moveleira**, 146 p., Dissertação (Mestrado em Matemática Aplicada) – Universidade Estadual Paulista, SP, 2007.

PIZZOLATO, N. D.; GANDOLPHO, A. A. **Técnicas de otimização**. 1ed. Rio de Janeiro: LTC, 2013.

STAVRINIDES, G. L.; KARATZA, H. D. Scheduling multiple task in heterogeneous distributed real-time systems by exploiting schedule holes with bin packing techiniques, **Simulation Modelling Practice and Theory**, v. 19, p. 540-552, 2011.

SCHOLL, A.; KLEIN, R.; JÜRGENS, C. BISON: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem, **Computers Operations Research**, v. 24, p. 627-645, 1997.

SOUZA, J. S. **Uma aplicação simulated annealing em problemas de corte de estoque**, 72 p., Dissertação (Mestrado em Matemática Aplicada) – Universidade Estadual de Campinas, SP, 2012.

SHWERIN, P.; WÄSCHER, G. The bin-packing problem: a problem generator and some numerical experiments with FFD packing and MTP , **International Transactions in Operational Research**, v. 4, p. 377-389, 1997.

SWEENEY, P. E.; PATERNOSTER, E. R. Cutting and packing problems: a categorized, application-orientated research bibliography, **The Journal of the Operational Research Society**, v. 43, p. 691-706, 1992.

TORRES, F. P. **Algoritmos heurísticos y exactos para problemas de corte no guillotina em dos dimensiones**, 214 p., Tese (Doutorado em Ciências Matemáticas) – Universidade de Valência, Espanha, 2004.

WÄSCHER, G.; HAUβNER, H.; SCHUMANN, H. An improved typology of cutting and packingproblems, **European Journal of Operational Research**, v. 183, p. 1109-1130, 2007.

APÊNDICE A – COMPILADO DOS RESULTADOS

Tabela A.1 – Compilado dos resultados por métodos e AK.

ÁREA DE *KITTING*	QTDE. DE REFERÊNCIAS	QTDE. ESTIMADA		QTDE. FFD		QTDE. FFD ADAPTADO		QTDE. MBS		QTDE. MBS'		QTDE. MÉTODO EXATO	
		FR	PRAT.	FR	PRAT.	FR	PRAT.	FR	PRAT.	FR	PRAT.	FR	PRAT.
AK01	63	4	12	4	10	4	10	4	10	4	10	4	10
AK02	36	2	6	2	5	2	6	2	5	2	5	2	5
AK03	4	1	3	1	1	1	1	1	1	1	1	1	1
AK04	42	5	15	3	8	3	8	3	8	3	8	3	8
AK05	93	7	21	6	16	6	16	6	16	6	16	6	16
AK06	105	12	36	6	16	6	16	6	16	6	16	6	16
AK07	202	20	60	13	38	13	37	13	38	13	38	13	38
AK08	318	19	57	19	56	19	56	20	58	19	57	19	56
AK09	119	14	42	10	30	10	30	10	30	10	30	10	30
AK10	25	5	15	2	5	2	5	2	5	2	5	2	5
AK11	300	21	63	21	62	20	60	21	63	21	62	21	62
AK12	151	14	42	12	34	12	34	11	33	11	32	11	32
AK13	27	4	12	2	5	2	5	2	5	2	5	2	5
AK14	9	1	3	1	2	1	2	1	2	1	2	1	2
AK15	364	24	72	23	67	23	67	23	68	23	67	-	s/solução
AK16	183	21	63	11	33	11	33	11	33	11	32	-	s/solução
AK17	362	27	81	23	67	23	68	23	68	23	67	-	s/solução
AK18	22	2	6	2	4	2	4	2	4	2	4	2	4
AK19	7	2	6	1	2	1	2	1	2	1	2	1	2
AK20	25	1	3	1	3	1	3	1	3	1	3	1	3
TOTAL	**2457**	**206**	**618**	**163**	**464**	**162**	**463**	**163**	**468**	**162**	**462**	**105**	**295**

Fonte: autora (2016). Legenda: FR = *Flowrack;* PRAT.= Prateleira.