



Programa Interdisciplinar de Pós-Graduação em  
**Computação Aplicada**  
Mestrado Acadêmico

Felipe Rodrigues Ferrary

Apoio à Tomada de Decisão e Minimização da Perda de  
Matéria Prima em Processos de Manufatura

São Leopoldo, 2015

Felipe Rodrigues Ferrary

**APOIO À TOMADA DE DECISÃO E MINIMIZAÇÃO DA PERDA DE  
MATÉRIA PRIMA EM PROCESSOS DE MANUFATURA**

Dissertação apresentada como requisito parcial para a  
obtenção do título de Mestre, pelo Programa  
Interdisciplinar de Pós-Graduação em Computação  
Aplicada da Universidade do Vale do Rio dos Sinos  
- UNISINOS

Orientador: Dr. José Vicente Canto dos Santos

São Leopoldo

2015



F378a

Ferrary, Felipe Rodrigues.

Apoio à tomada de decisão e minimização da perda de matéria prima em processos de manufatura / Felipe Rodrigues Ferrary. – 2015.

143 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, 2015.

"Orientador: Dr. José Vicente Canto dos Santos."

1. Manufatura. 2. Chapas. 3. Nesting. 4. Otimização de peças. 5. Minimização de sucata I. Título.

CDU 004.9

Felipe Rodrigues Ferrary

Apoio à tomada de decisão e minimização da perda de matéria prima em processos de  
manufatura

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Prof. Dr. João Carlos Furtado – UNISC

---

Prof. Dr. Leonardo Dagnino Chiwiacowsky – UNISINOS

---

Prof. Dr. José Vicente Canto dos Santos – UNISINOS

Prof. Dr. José Vicente Canto dos Santos (Orientador)

Visto e permitida a impressão  
São Leopoldo,

Prof. Dr. Cristiano André da Costa  
Coordenador PPG em Computação Aplicada



## RESUMO

O estudo a seguir tem por objetivo otimizar e tornar automatizado um sistema de manufatura que utiliza chapas como matéria prima. A preparação do processo de manufatura através do sistema CAM analisado atualmente possui diversas etapas que não possuem comunicação entre si, além disso, exige um alto nível de intervenção do usuário em suas tomadas de decisão. Esse processo deve ser unificado de forma a buscar um resultado aprimorado, com base nos critérios de otimização propostos. As etapas do processo de manufatura envolvendo chapas são analisadas ao longo do trabalho, passando por etapas iniciais como a definição das peças que devem ser produzidas, etapas intermediárias como o processo de otimização de peças (*nesting*) e sua respectiva parametrização, e a etapa final, ou seja, a obtenção do código NC para a produção das peças. Para otimizar tais etapas, é proposto um sistema de apoio à tomada de decisão, com características híbridas, formado por um sistema especialista e por técnicas de otimização, tais como metaheurísticas. O método proposto aprimora os resultados através da parametrização automatizada utilizando o sistema de apoio à tomada de decisão, definindo a melhor parametrização com base nos produtos a serem manufaturados, reduzindo assim a necessidade de decisões manuais e, por conseguinte, a interferência do usuário no processo e, ainda, eliminando a necessidade do mesmo ser um especialista. Essa automação deve analisar as possíveis chapas em estoque assim como parâmetros oferecidos pelo processo de *nesting* e buscar a melhor configuração para o processo, analisando possíveis permutações. Além disso, é proposta a adição de um novo componente nas etapas de manufatura, responsável por analisar as sobras aproveitáveis do processo e organizar a sucata gerada pelo mesmo, tornando-a disponível para ser reutilizada futuramente. Analisando o SAD implementado, foi possível observar que os resultados obtidos foram satisfatórios e, em muitos casos, superiores aos obtidos em outros testes realizados na literatura. Além disso, a aceitação do sistema pelos usuários que realizaram os testes de performance e viabilidade de uso foi considerada excelente. Conforme apontado pelos usuários, o número de parâmetros a ser selecionado reduziu drasticamente, tornando assim o sistema mais simples de ser utilizado.

**Palavras-chave:** Manufatura, Chapas, Minimização de sucata, Nesting, Otimização de peças.





## ABSTRACT

The following work aims to optimize and make automated a manufacturing system that uses metal plates as raw material. The analysed process currently has several steps with no communication between each other, moreover requires a high level of user's intervention in their decision making. This process must be unified in order to achieve the optimal result based on the optimization criteria proposed. The analysed processes throughout this work address all stages of the manufacturing process involving plates, going through the initial stages as the definition of the parts to be produced, intermediate steps as the process of optimizing parts (nesting) and its parameters and final step, i.e., obtaining the CN code for the production of parts. To optimize these steps, a method of hybrid solution using a system to support the decision making aided by an expert system and known optimization (such metaheuristics) is proposed. The proposed method will improve the results through an automated parameterization using the system to support the decisions making, defining the best parameter based on the products to be manufactured, thus reducing the need for manual decisions and therefore the user input in the process and eliminating the need for an expert to be the same. This automation should analyse the possible plates in stock as well as parameters offered by the nesting process and seek the best configuration for the process, analysing possible permutations. Furthermore, we propose the addition of a new component in the steps of manufacturing, responsible for analysing the process remains usable and organize the scrap generated by it, making it available for reuse in the future. Through the proposed system, good and relevant results were obtained. In several cases the obtained results are better than results in the literature. In addition, the acceptance of the system by users who performed performance system tests was considered excellent. As pointed by these users, the number of parameters to be selected has drastically reduced, thus making the system simpler to use.

**Keywords:** Manufacturing, Raw material sheet, scrap minimization, Nesting, Parts Optimization.



## LISTA DE FIGURAS

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Figura 1 - Passos para realização de um projeto de usinagem utilizando chapas ..... | 20 |
| Figura 2 - Desenho criado utilizando uma ferramenta CAD.....                        | 21 |
| Figura 3 - Exemplo de linguagem APT.....                                            | 22 |
| Figura 4 - Caminho de ferramenta e simulação do processo de usinagem.....           | 23 |
| Figura 5 - <i>Nesting</i> aplicado.....                                             | 24 |
| Figura 6 - <i>Nesting</i> - Exemplo de peça a ser otimizada .....                   | 25 |
| Figura 7 - <i>Nesting</i> - Resultado obtido através da otimização .....            | 25 |
| Figura 8 - Retorno ao processo de parametrização.....                               | 29 |
| Figura 9 - Código G utilizando os comandos G1,G2 e G3.....                          | 30 |
| Figura 10 - Exemplo de chaveamento em um sistema híbrido.....                       | 32 |
| Figura 11 - Ligação entre componentes de um SAD.....                                | 33 |
| Figura 12 - Elementos de um sistema especialista .....                              | 38 |
| Figura 13 - Entrada sendo transformada em fatos .....                               | 39 |
| Figura 14 - Fluxograma demonstrando a escolha com base nas regras .....             | 42 |
| Figura 15 - Fases de um estudo de Pesquisa Operacional.....                         | 43 |
| Figura 16 - Pseudocódigo da Busca Tabu .....                                        | 46 |
| Figura 17 - Inserção de duas componentes para a otimização do sistema .....         | 51 |
| Figura 18 - Componentes do sistema híbrido proposto.....                            | 52 |
| Figura 19 - Arquitetura do SAD proposto .....                                       | 53 |
| Figura 20 - Obtenção das características da peça .....                              | 55 |
| Figura 21 - Exemplos de extração de sobras aproveitáveis.....                       | 62 |
| Figura 22 - Interface do sistema de tomada à decisão.....                           | 65 |
| Figura 23 - Exemplo de “Lista de <i>Nesting</i> ” .....                             | 65 |
| Figura 24 - Opções extras do diálogo .....                                          | 66 |
| Figura 25 - Interface do banco de dados .....                                       | 66 |
| Figura 26 - Diferentes tipos de madeira .....                                       | 69 |
| Figura 27 - Peça que necessita de rotação para ser encaixada.....                   | 70 |
| Figura 28 - Exemplo de corte excede a espessura do material.....                    | 71 |
| Figura 29 - Pseudocódigo do algoritmo (solução inicial).....                        | 73 |
| Figura 30 - Pseudocódigo do algoritmo (intensificação da busca).....                | 74 |
| Figura 31 - Arquitetura da Busca Tabu proposta .....                                | 75 |
| Figura 32 - Pseudocódigo do algoritmo (avaliação da função objetivo).....           | 77 |

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| Figura 33 - Sintaxe das regras (Drools.NET) .....                                     | 78  |
| Figura 34 - Amostra de material com e sem marcas .....                                | 79  |
| Figura 35 - Diferentes tipos de seleção de peça .....                                 | 81  |
| Figura 36 - Diálogo de inserção de novas regras.....                                  | 87  |
| Figura 37 - Arquitetura do banco de dados .....                                       | 88  |
| Figura 38 - Extração das sobras aproveitáveis através do método Retangular .....      | 89  |
| Figura 39 - Extração das sobras aproveitáveis através do método de Varredura .....    | 90  |
| Figura 40 - Simplificação de peça .....                                               | 90  |
| Figura 41 - Busca fina sendo realizada.....                                           | 91  |
| Figura 42 - Perfil de sucata extraído pelo SAD.....                                   | 92  |
| Figura 43 - Sobra rejeitada e sobra aceita pela avaliação.....                        | 93  |
| Figura 44 - Parametrização do SAD.....                                                | 96  |
| Figura 45 - Resultados das chapas obtidas .....                                       | 97  |
| Figura 46 - Novos testes com peças não retangulares .....                             | 98  |
| Figura 47 - Exemplo de instância de teste utilizada na segunda métrica de testes..... | 100 |



## LISTA DE TABELAS

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Tabela 1 - Níveis de automação propostos por Black (1998).....                      | 18  |
| Tabela 2 - Parâmetros principais do <i>nesting</i> .....                            | 27  |
| Tabela 3 - Parametrização dos processos de <i>nesting</i> .....                     | 28  |
| Tabela 4 - Estrutura utilizada para armazenar regras .....                          | 41  |
| Tabela 5 - Informações obtidas das peças .....                                      | 67  |
| Tabela 6 - Características armazenadas no banco de dados .....                      | 68  |
| Tabela 7 - Regras utilizadas para escolha do método de <i>nesting</i> .....         | 79  |
| Tabela 8 - Regras aplicadas aos parâmetros da peça.....                             | 80  |
| Tabela 9 - Parâmetros que necessitam caminhos de ferramenta .....                   | 81  |
| Tabela 10 - Regras com restrições quanto ao número de peças .....                   | 82  |
| Tabela 11 - Identificação de parâmetros.....                                        | 82  |
| Tabela 12 - Regras utilizadas para selecionar parâmetros especiais .....            | 83  |
| Tabela 13 - Parâmetros de peça não automatizados .....                              | 85  |
| Tabela 14 - Parâmetros especiais escolhidos pelo usuário .....                      | 86  |
| Tabela 15 - Lista de instâncias para testes.....                                    | 94  |
| Tabela 16 - Resultados dos testes realizados com as instâncias 1 à 10 .....         | 96  |
| Tabela 17 - Comparativo de resultados entre literatura e estudo .....               | 97  |
| Tabela 18 - Listas de instâncias para testes definidas neste estudo .....           | 99  |
| Tabela 19 - Resultados dos testes realizados utilizando as instâncias 11 à 15 ..... | 99  |
| Tabela 20 - Dados das chapas disponíveis para segunda métrica de testes.....        | 101 |
| Tabela 21 - Resultados pelo método de teste manual (usuário 1) .....                | 102 |
| Tabela 22 - Resultados obtidos através do SAD (usuário 1).....                      | 103 |
| Tabela 23 - Resultados pelo método de teste manual (usuário 2) .....                | 105 |
| Tabela 24 - Resultados obtidos através do SAD (usuário 2).....                      | 105 |



## LISTA DE ABREVIATURAS

|         |                                                                  |
|---------|------------------------------------------------------------------|
| 2D      | Espaço Bidimensional, usualmente XY                              |
| 3D      | Espaço Tridimensional, usualmente XYZ                            |
| API     | <i>Application Programming Interface</i>                         |
| APT     | <i>Automatically Programmed Tool</i>                             |
| .NET    | <i>.NET Framework</i>                                            |
| C# .NET | Linguagem de Programação C <i>Sharp</i> utilizando recursos .NET |
| CAD     | Desenho Assistido por Computador                                 |
| CAE     | Engenharia Auxiliada por Computador                              |
| CAM     | Manufatura Assistida por Computador                              |
| DBMS    | Sistema de gerenciamento de Banco de Dados                       |
| DDM     | Diagrama Diálogo, Dados e Modelos                                |
| ERP     | Sistema Integrado de Gestão Empresarial                          |
| GUI     | <i>Graphical User Interface</i>                                  |
| HDF     | <i>High-Density Fiberboard</i>                                   |
| LHS     | <i>Left Hand Side</i>                                            |
| MDF     | <i>Medium-Density Fibreboard</i>                                 |
| NC      | Comando Numérico                                                 |
| PO      | Pesquisa Operacional                                             |
| RHS     | <i>Right Hand Side</i>                                           |
| SAD     | Sistema de Apoio à Decisão                                       |
| VBA     | Linguagem de Programação <i>Visual Basic for Applications</i>    |





## SUMÁRIO

|                                                                                                   |           |
|---------------------------------------------------------------------------------------------------|-----------|
| <b>1 INTRODUÇÃO .....</b>                                                                         | <b>15</b> |
| <b>1.1 Justificativa .....</b>                                                                    | <b>16</b> |
| <b>1.2 Objetivos .....</b>                                                                        | <b>17</b> |
| <b>1.3 Estrutura do Trabalho .....</b>                                                            | <b>18</b> |
| <b>2 FLUXO DE MANUFATURA E REVISÃO DE SUAS ETAPAS .....</b>                                       | <b>20</b> |
| <b>2.1 Desenho Assistido por Computador ou CAD (<i>Computer Aided Design</i>).....</b>            | <b>21</b> |
| <b>2.2 Manufatura Auxiliada por Computador ou CAM (<i>Computer-Aided Manufacturing</i>) .....</b> | <b>22</b> |
| <b>2.3 Otimização de Peças - <i>Nesting</i> .....</b>                                             | <b>24</b> |
| <b>2.4 Parametrização <i>Nesting</i> .....</b>                                                    | <b>26</b> |
| <b>2.5 Resultados .....</b>                                                                       | <b>29</b> |
| <b>2.6 Código NC.....</b>                                                                         | <b>29</b> |
| <b>3 REVISÃO BIBLIOGRÁFICA: SISTEMAS INTELIGENTES E OTIMIZAÇÃO .....</b>                          | <b>31</b> |
| <b>3.1 Sistemas Híbridos .....</b>                                                                | <b>32</b> |
| <b>3.2 Subsistemas de um SAD.....</b>                                                             | <b>33</b> |
| 3.2.1 Banco de Dados.....                                                                         | 33        |
| 3.2.2 Banco de Modelos .....                                                                      | 35        |
| 3.2.3 Interface.....                                                                              | 36        |
| <b>3.3 Sistemas Especialistas .....</b>                                                           | <b>36</b> |
| 3.3.1 Arquitetura de um Sistema Especialista .....                                                | 38        |
| 3.3.1.1 Base de Conhecimento .....                                                                | 39        |
| 3.3.1.2 Mecanismo de Inferência.....                                                              | 40        |
| 3.3.1.3 Interface de Aquisição de Conhecimento .....                                              | 40        |
| 3.3.1.4 Interface com o Usuário .....                                                             | 41        |
| <b>3.4 Pesquisa Operacional.....</b>                                                              | <b>42</b> |
| 3.4.1 Otimização.....                                                                             | 44        |
| 3.4.2 Metaheurísticas.....                                                                        | 44        |
| 3.4.2.1 Busca Tabu.....                                                                           | 45        |
| <b>3.5 Análise das Contribuições Encontradas na Literatura .....</b>                              | <b>47</b> |
| 3.5.1 Sistema de Apoio à Tomada de Decisão .....                                                  | 47        |
| 3.5.2 Sistema Especialista e Sistemas Baseados em Regras.....                                     | 48        |
| 3.5.3 Sobras Aproveitáveis e seu Reuso .....                                                      | 48        |
| 3.5.4 Problema de Corte ( <i>nesting</i> ).....                                                   | 49        |
| <b>4 METODOLOGIA E MODELO CONCEITUAL.....</b>                                                     | <b>51</b> |
| <b>4.1 Visão Geral do Sistema Proposto .....</b>                                                  | <b>51</b> |
| <b>4.2 Visão Geral do SAD Híbrido Proposto.....</b>                                               | <b>51</b> |
| <b>4.3 Reconhecimento das Características das Peças .....</b>                                     | <b>54</b> |
| <b>4.4 Seleção Automatizada da Matéria Prima .....</b>                                            | <b>56</b> |
| <b>4.5 Parametrização <i>Nesting</i> .....</b>                                                    | <b>57</b> |
| 4.5.1 Sistema Especialista .....                                                                  | 57        |
| 4.5.2 Critérios de Otimização .....                                                               | 59        |
| 4.5.2.1 Redução do uso de Matéria Prima .....                                                     | 59        |
| 4.5.2.2 Minimização do Custo Total do Processo .....                                              | 60        |
| 4.5.2.3 Minimização do Tempo de Processo ( <i>Makespan</i> ).....                                 | 61        |
| <b>4.6 Extração das Sobras Aproveitáveis.....</b>                                                 | <b>61</b> |
| <b>5 IMPLEMENTAÇÃO DO MODELO PROPOSTO .....</b>                                                   | <b>64</b> |
| <b>5.1 Interface do <i>Software</i>.....</b>                                                      | <b>64</b> |
| <b>5.2 Reconhecimento das Características.....</b>                                                | <b>66</b> |
| <b>5.3 Seleção Automatizada da Matéria Prima .....</b>                                            | <b>68</b> |
| 5.3.1 Avaliação das Matérias Primas Disponíveis no Banco de Dados.....                            | 68        |
| 5.3.2 Busca Tabu.....                                                                             | 71        |
| 5.3.2.1 Solução Inicial.....                                                                      | 72        |
| 5.3.2.2 Intensificação da Busca .....                                                             | 74        |
| 5.3.2.3 Diversificação da Busca .....                                                             | 74        |
| 5.3.2.4 Avaliação da Função Objetivo .....                                                        | 76        |
| <b>5.4 Sistema Especialista.....</b>                                                              | <b>77</b> |

|                                                                  |            |
|------------------------------------------------------------------|------------|
| 5.4.1 Regras.....                                                | 77         |
| 5.4.1.1 Parâmetros Principais .....                              | 78         |
| 5.4.1.2 Parâmetros Especiais .....                               | 80         |
| 5.4.1.3 Parâmetros Seleccionados pelo Usuário .....              | 85         |
| 5.4.1.4 Módulo de Aprendizado .....                              | 86         |
| <b>5.5 Banco de Dados.....</b>                                   | <b>87</b>  |
| <b>5.6 Extração das Sobras Aproveitáveis.....</b>                | <b>88</b>  |
| 5.6.1 Extração do Perfil das Sobras .....                        | 89         |
| 5.6.2 Avaliação das Sobras Aproveitáveis .....                   | 92         |
| <b>6 TESTES E RESULTADOS .....</b>                               | <b>94</b>  |
| <b>6.1 Análise das Instâncias Encontradas na Literatura.....</b> | <b>94</b>  |
| <b>6.2 Análise Feita por Usuários.....</b>                       | <b>100</b> |
| 6.2.1 Usuário 1 (Especialista).....                              | 102        |
| 6.2.2 Usuário 2 (Não-especialista) .....                         | 104        |
| <b>7 CONCLUSÕES .....</b>                                        | <b>107</b> |
| <b>REFERÊNCIAS .....</b>                                         | <b>109</b> |
| <b>ANEXO A - TRABALHOS PÚBLICADOS ATÉ JANEIRO/2015 .....</b>     | <b>117</b> |

## 1 INTRODUÇÃO

A busca pela minimização de perdas na manufatura é indispensável pelas empresas que almejam se manter competitivas e presentes na liderança do mercado, onde o aprimoramento econômico dos processos de manufatura é uma das principais metas dos gestores no cenário atual (HAJIREZAIE et al., 2010). Na maioria dos casos, simples soluções como produção em lotes pode trazer grande benefício a todos os tipos de empresas. Porém, devido à atual competitividade de preços, grandes empresas se veem em situações onde simples soluções não garantem a fidelidade do cliente, que muitas vezes decide apenas com base no preço dos produtos. Existindo tal necessidade de reduzir o máximo possível em gastos de matéria prima para repassar tal economia ao cliente, surge o termo minimização.

Ao discutir-se minimização de tempo e processos, é necessário citar a automação dos mesmos que, ao longo das décadas, vem sendo melhorada e cada vez mais possibilitando produção de peças em grandes quantidades. Um exemplo são as empresas do setor mecânico, onde a usinagem de peças passou de um processo artesanal para a produção em massa. Tal melhoria é possível através de tecnologias como aplicativos de desenho assistido por computador (CAD) e manufatura assistida por computador (CAM). Neste caso, quando falamos em melhorias em processos que são relacionadas à manufatura assistida por computador, pode-se afirmar a necessidade de investimento em ferramentas automatizadas para evitar as perdas que grandes empresas enfrentam (CIMDATA, 2005).

Na manufatura de peças que usam chapas de metal ou madeira como matéria prima, o processo de *nesting* é o usualmente adotado para reduzir a quantidade de matéria prima utilizada e, por conseguinte, o custo de produção como um todo. Este processo tem a finalidade de evitar perdas devido a sua eficácia em alocar o maior número ou a quantidade desejada de peças dentro de uma área plana utilizando a menor quantidade de espaço possível.

O processo de *nesting* pode ser descrito como um problema clássico, onde peças são alocadas em uma chapa plana visando minimizar a perda de matéria prima (AKUNURU; BABU, 2013). Wäscher et al. (2007), reforçam a ideia de que este é um problema conhecido e pode frequentemente ser encontrado nas indústrias do setor metal-mecânico, vidro, construção naval, fabricação de móveis (XIE et al., 2007) ou até mesmo na indústria têxtil (CHRYSSOLOURIS et al., 2000).

Com o avanço dos estudos relacionados aos problemas de corte e empacotamento, com suas diversas propostas de solução, os recursos de otimização de peças passaram a ser de uso comum em grandes processos de manufatura envolvendo chapas. O processo de manufatura como um todo é composto por diversas etapas, muitas delas envolvendo grande interação humana, o que potencializa o aumento de erros. Por esse motivo, o aumento do controle do processo através da parametrização automatizada, análise das possíveis combinações de matéria prima disponíveis em estoque e o aproveitamento da sucata gerada, são propostos como solução a fim de otimizar tais etapas.

Visando alcançar um resultado eficiente, a solução proposta busca acompanhar o diagrama DDM (Diálogo, Dados e Modelos), proposto por Sprague e Carlson (1982), onde se deve buscar o equilíbrio perfeito entre os três elementos. Tais elementos são de grande importância, uma vez que um SAD (Sistema de Apoio à Decisão) tem de consistir em um conjunto de recursos nas áreas de diálogo, dados e construção de modelos. Para tanto, deve-se desenvolver um sistema de fácil utilização onde não seja necessário um vasto conhecimento técnico sobre o assunto.

## 1.1 Justificativa

Na pesquisa anual realizada pela CIMdata, fornecedores de software CAM são convidados a listar três principais tendências tecnológicas conforme que definirão o caminho que seu *software* deverá seguir. Os resultados são publicados no relatório de avaliação de mercado anual da CIMdata, avaliando o estado da indústria. Neste relatório, um dos pontos constantemente levantado é a automação dos sistemas CAM.

Tal automação em *softwares* CAM vem se tornando cada vez mais elaborada em todos os aspectos, tornando-se de fácil aprendizagem e facilitando seu uso no dia a dia; o que torna o *software* mais produtivo para o usuário (CIMDATA, 2005).

Por esse motivo, as companhias que buscam a liderança de mercado, devem possuir sistemas capazes de executarem decisões que possibilitem a redução do custo dos seus processos de manufatura. Tal capacidade de automação auxilia diretamente e indiretamente na questão de redução de custos, evitando perdas desnecessárias ou despercebidas.

Segundo Brinson (1996), perdas e desperdícios são constituídos pelas atividades que não agregam valor e que resultam em gastos de tempo, dinheiro, recursos sem lucro, além de adicionarem custos desnecessários aos produtos. Atividades que não agregam valor são aquelas que devem ser eliminadas sem que haja deterioração no desempenho da empresa. Sendo assim, a economia de recursos acaba por agregar valor aos produtos manufaturados. Através do uso de conhecimento sobre o processo e da automação proposta por esse trabalho, almeja-se agregar valor.

Um dos grandes gastos de recursos humanos de uma empresa está relacionado à necessidade de manter colaboradores que dominam o conhecimento em determinada área no quadro de funcionários, chamados de especialistas, operando o *software* CAM e gerenciando o fluxo de manufatura.

Reduzir a necessidade de interação entre usuário e o *software* CAM é uma forma de automação, possibilitando assim a redução de custos e diminuindo a necessidade do contato direto e extenso do especialista com o sistema. Além disso, a redução da interação do especialista com o *software* e a automação de processos faz com que esses sejam realizados de forma mais eficiente, reduzindo assim a probabilidade de erros.

Quando um *software* CAM faz o uso da ferramenta *nesting*, isso demanda do usuário um conhecimento ainda mais aprofundado na área, uma vez que além, de conhecer o processo é também necessária a total imersão do usuário ao *software*. Isso se deve ao fato de que a grande maioria das soluções comerciais de *nesting* possuem um alto número de parâmetros a serem selecionados e definidos na criação do código NC (Controle Numérico).

Além da automação propriamente dita, atualmente, a informação é de suma importância para gestão da produção. Gestores dependem de sistemas de informação para provê-los com as informações necessárias nas tomadas de decisões em situações críticas. Tais sistemas garantem que gestores tenham acesso a um grande fluxo de informação, minuto a minuto, em seu ambiente de trabalho (BENDOLY et al., 2009). Estima-se que 400 das 500 empresas listadas na lista “*Fortune 500*”, lista onde são apresentadas 500 empresas mais bem sucedidas no ano, adotaram sistemas ERP ou sistemas de informação para obter tal sucesso (METAGROUP, 2004).

A importância deste estudo também se dá devido à melhoria da integração entre as ferramentas CAD e CAM. Embora tal integração seja vastamente estudada, a realidade encontrada nas soluções comerciais para aplicativos CAM que possuem a ferramenta de *nesting* é diferente, existindo a perda de informação quando os projetos são transcritos do CAD. Em geral, essa perda de informação se dá pela necessidade do *software* em suportar dezenas de diferentes formatos de arquivos CAD, não sendo possível traduzir tais formatos com excelência e, assim, perdendo informações durante a transição. Um exemplo comum é encontrar *softwares* CAM, que quando traduzindo arquivos CAD, carregam apenas nomes e cores de camadas (*layers*) ao invés de uma gama de dados sobre o desenho.

A integração das ferramentas de CAD e CAM acontece com maior precisão quando ambas as soluções pertencem à mesma empresa desenvolvedora de software, uma vez que um software é desenvolvido nativamente para se comunicar com o outro, possuindo assim a total integração entre CAD/CAM e, muitas vezes, com outras soluções, como CAE e ERP. Todavia, essa escolha é normalmente a opção mais cara e não garante qualidade para todas as ferramentas escolhidas. Por esse motivo, o sistema proposto através desse estudo traz uma solução genérica, podendo ser utilizada em diversos formatos de arquivos CAD.

## 1.2 Objetivos

O presente trabalho visa desenvolver e implementar um sistema híbrido de apoio à tomada de decisão e aplicá-lo a um sistema de manufatura, onde chapas são utilizadas como matéria prima em conjunto com a ferramenta de otimização de peças, conhecida como *nesting*. Para tanto, uma abordagem utilizando métodos ligados à área de Inteligência Artificial e da Pesquisa Operacional é proposta.

O sistema de apoio à decisão tem como principal objetivo auxiliar o usuário, de forma a realizar o processo de manufatura em um *software* CAM e utilizar os resultados fornecidos pelo sistema, reduzindo assim as interações do usuário com o *software*. Para reduzir tal necessidade de interação, são necessárias sugestões sucintas e eficientes vindas de um sistema de apoio. Por esse motivo, é proposto nesse trabalho um sistema especialista que tem por objetivo auxiliar no processo de tomada de decisão e tornar o sistema automatizado.

No processo atual, cada etapa do ciclo de manufatura funciona de forma independente e, em muitos *softwares* oferecidos comercialmente, não há uma real integração entre as etapas. Por esse motivo, este estudo tem por objetivo unificar tal processo, criando comunicação entre as etapas e utilizando todos os recursos necessários de forma eficiente, aprimorando assim o nível de automação agregado ao *software* e possibilitando a minimização de custos.

Para realizar a automação desejada, os principais objetivos específicos do trabalho são:

- Reconhecimento automatizado de características e informações sobre os itens (peças) que passem pelo fluxo de manufatura;
- Uso de metaheurísticas para a seleção automatizada e otimizada das matérias primas utilizadas no projeto de manufatura.
- Desenvolvimento de um sistema especialista que possibilite a total integração com o *software* e realize de forma automatizada a parametrização dos processos relacionados ao software CAM e à ferramenta de *nesting*;
- Aprimorar os resultados obtidos quando utilizando *nesting*;

- Reconhecimento automático das sobras aproveitáveis deixadas pelo processo de manufatura, possibilitando seu uso futuramente;
- Implementação de um banco de dados que contenha a informação sobre estoque de matéria prima, incluindo chapas novas e criadas a partir de sobras reaproveitáveis.

Embora muitos sistemas busquem um nível de automação determinado como “Automação Real”, definido por Shingo (1996) como um sistema onde a máquina é responsável por todos os processos, reduzindo a zero a intervenção do usuário, o objetivo do presente trabalho é alcançar um nível de automação intermediário.

Através do estudo de Black (1998), é possível quantificar o nível de automação proposto neste estudo. O autor separa em níveis de automação conforme o grau de atributos realizados pela máquina. Esse grau de automação é aprimorado de acordo com a quantidade de atributos humanos, ações até então realizadas por pessoas, que passam a ser realizados pela máquina, ou seja, quanto mais atributos humanos, maior o seu nível de automação. A Tabela 1 demonstra os níveis de automação propostos no estudo.

**Tabela 1 - Níveis de automação propostos por Black (1998).**

| Nível De Automação | Atributo Humano Equivalente                                               |
|--------------------|---------------------------------------------------------------------------|
| 0                  | Nenhum: Alavanca, chave de fenda, roldana                                 |
| 1                  | Energia: trabalho manual                                                  |
| 2                  | Destreza: auto alimentação                                                |
| 3                  | Diligência: sem realimentação                                             |
| 4                  | Julgamento: realimentação posicional                                      |
| 5                  | Avaliação: controle adaptivo, análise dedutiva, realimentação do processo |
| 6                  | Aprendizado: pela experiência                                             |
| 7                  | Raciocínio: apresentam intuição. Relaciona causas e efeitos               |
| 8                  | Criatividade: Realiza Projetos sem auxílio                                |
| 9                  | Dominância: supermáquinas                                                 |

Fonte: adaptado de Black (1998).

Ao longo do trabalho, o nível de automação almejado será 7 (sete) já que o sistema deverá apresentar um determinado nível de intuição sobre quais as estratégias deverão ser tomadas, assim como relacionar as causas e efeitos de tais decisões pelo sistema de apoio.

### 1.3 Estrutura do Trabalho

O trabalho está estruturado de forma a facilitar o entendimento do leitor sobre o problema abordado, iniciando com uma revisão do sistema que será estudado e analisando a bibliografia dos recursos que serão utilizados ao longo do estudo. A estrutura está organizada da seguinte forma:

- **Capítulo 1:** São apresentados os motivos e justificativas para a realização do estudo, assim como um resumo dos objetivos que o mesmo busca alcançar.

- **Capítulo 2:** O funcionamento atual do fluxo de manufatura é revisado, suas seções estão organizadas de forma sequencial, respeitando a ordem que o processo de manufatura ocorre. Além disso, uma breve revisão sobre os conceitos utilizados em tal processo é realizada.
- **Capítulo 3:** Ao longo desse capítulo, a revisão bibliográfica das áreas de conhecimento que serão utilizadas na implementação do sistema proposto é abordada. Também, são analisadas as principais contribuições feitas por outros artigos que auxiliaram na realização do presente estudo.
- **Capítulo 4:** O modelo conceitual do método proposto é apresentado, mostrando como o sistema opera e detalhando as ferramentas que serão utilizadas.
- **Capítulo 5:** Neste capítulo, são apresentados detalhes sobre a implementação do modelo descrito ao longo do capítulo anterior. São apresentados detalhes sobre a forma e as principais problemáticas encontradas durante a implementação.
- **Capítulo 6:** Ao longo desse capítulo serão apresentados, de forma detalhada, os testes realizados no modelo implementado, visando validar o modelo proposto, assim como seus respectivos resultados.
- **Capítulo 7:** Nesse capítulo são apresentadas as principais conclusões observadas durante o desenvolvimento do trabalho. Também são propostas possíveis abordagens para serem realizadas em forma de trabalhos futuros.



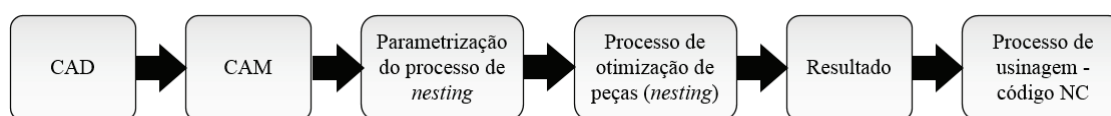
## 2 FLUXO DE MANUFATURA E REVISÃO DE SUAS ETAPAS

Embora o processo de usinagem como um todo seja um procedimento automatizado, muitas etapas deste, ligadas à otimização de peças em chapas como matéria prima, requerem intervenção do usuário; o que pode demandar um longo tempo para finalizar o processo de manufatura.

As etapas de tomada de decisão, em geral, podem exigir tempo e dedicação do usuário para efetuar a seleção dos parâmetros ótimos a serem adotados no processo e a melhor forma, de acordo com as suas exigências, de realizar o projeto. Tais fatores, em um sistema não automatizado, exigem, além da intervenção do usuário, a instrução e treinamento para executar tais tarefas manualmente.

Esse processo como um todo, pode ser dividido em diversas etapas com diferentes graus de intervenção humana (Figura 1). O processo é iniciado com a chegada das peças do setor de projetos; em forma de arquivos digitais criados através de um *software* de desenho assistido por computador (CAD). Na etapa seguinte, os desenhos devem receber caminhos de ferramentas que representam o deslocamento da ferramenta no processo de usinagem, criados através do programa de manufatura assistida por computador (CAM), que posteriormente é responsável pela criação do código NC (comando numérico).

**Figura 1 - Passos para realização de um projeto de usinagem utilizando chapas**



Fonte: elaborado pelo autor (2014).

Embora as etapas citadas acima demandem determinado tempo de interação do usuário, usualmente estas são determinísticas e com padrões já estipulados anteriormente. Todavia, o procedimento de parametrização envolve escolhas múltiplas que alteram diretamente o resultado. Essa parametrização é iniciada desde a escolha de qual será a matéria prima, escolha da chapa onde as peças serão alocadas, rotação das peças (frequentemente utilizada quando a matéria prima é madeira, já que seu veio deve ser respeitado), entre outros parâmetros. Uma vez que não é possível conhecer o resultado no momento da parametrização, muitas vezes se torna inviável analisar todos os possíveis resultados até obter a melhor opção, assim como a combinação de parâmetros que gera esse resultado.

Uma vez que a parametrização seja concluída, algoritmos de otimização que utilizam técnicas de *Packing Problem* serão utilizados para obter o resultado otimizado na alocação das peças. Caso o resultado seja satisfatório, então é possível gerar o código NC e, por conseguinte, o processo de usinagem. Todavia, conforme apresentado em mais detalhes ao longo da seção 2.5, caso o resultado não seja satisfatório, será necessário retornar à etapa de parametrização até que o mesmo se torne satisfatório diante da demanda do projeto.

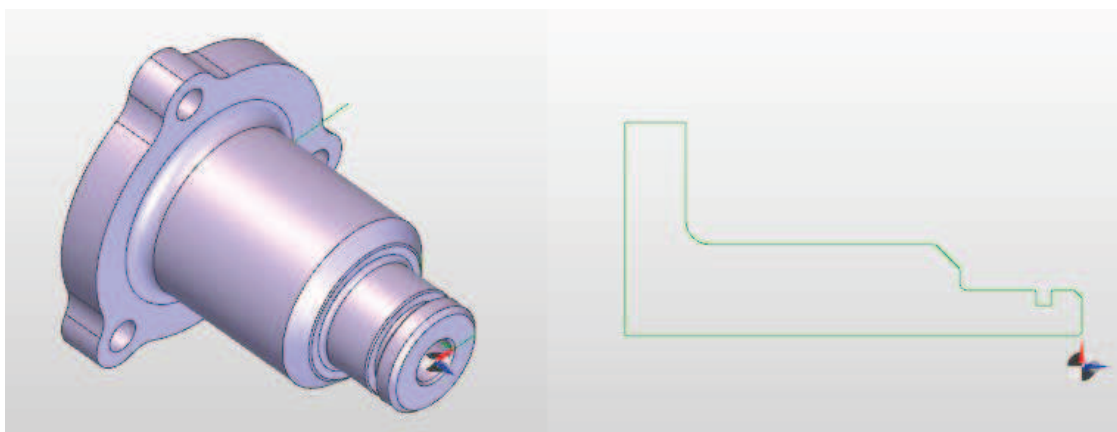
Dentre os passos descritos, duas etapas possuem maior necessidade de otimização e automação. Primeiramente a etapa de parametrização, uma vez que as diversas escolhas podem levar a resultados que não sejam o ótimo. Além disso, a chapa usada como matéria prima deve ser escolhida com tamanho ideal para evitar perdas. Uma vez que a seleção é dependente das características do projeto, a chapa pode acabar não sendo totalmente preenchida, gerando assim o perfil que representa a sucata criada pelo processo; que pode ser aproveitada futuramente em outra ocasião.

## 2.1 Desenho Assistido por Computador ou CAD (*Computer Aided Design*)

CAD pode ser definido como o uso de ferramentas computacionais para desenvolver, analisar, modificar ou otimizar um desenho ou projeto de engenharia. Em sua grande maioria, é responsável por construir imagens digitais, através de elementos tais como geometrias básicas, linhas, arcos, pontos, entre outros elementos. Através dele, também é possível aumentar, diminuir, mover, rotacionar, entre diversas outras transformações, em modelos já existentes (SARCAR et al., 2008).

Quando comparado ao desenho técnico tradicional criado manualmente, a ferramenta CAD é indubitavelmente superior em critérios básicos como: tempo para realização de projetos, qualidade, precisão, facilidade na transição de dados. Gonçalves (1995) afirma que a velocidade de compreensão gráfica pode alcançar uma proporção 50.000 vezes maior que a de leitura. Por esse motivo, o método conhecido como desenho manual vem sendo substituído por outro chamado Projeto Auxiliado por Computador. A Figura 2 apresenta um desenho criado através de uma ferramenta CAD.

**Figura 2 - Desenho criado utilizando uma ferramenta CAD**



Fonte: elaborado pelo autor (2014).

Quando o objetivo de produzir um desenho é a manufatura do mesmo, e no caso de ser criado através da ferramenta CAD; o desenho será responsável por levar todas as informações da área de *design* até a área de produção, onde serão gerados códigos NC para a produção final do projeto.

Basicamente, isso ocorre devido ao fato do código NC ser criado com base nas linhas e arcos contidos no desenho através do *software*. Esses comandos são enviados para a máquina CNC, seguidos das coordenadas onde se encontram pontos relevantes das geometrias, para que uma ferramenta de corte siga tais coordenadas e crie o produto final.

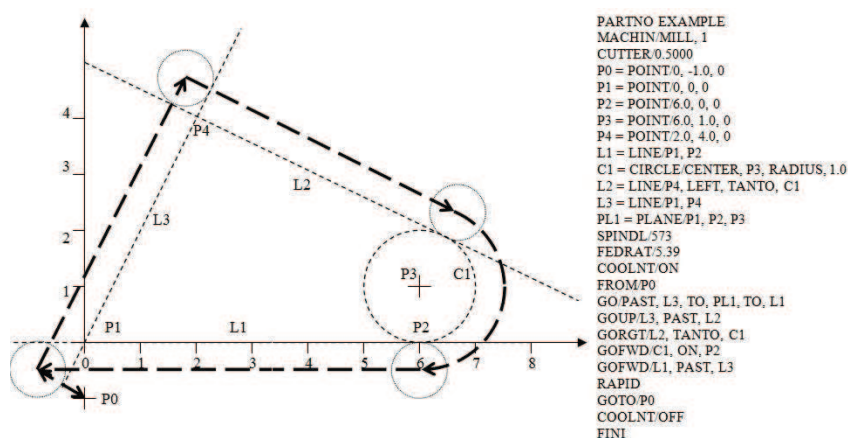
Uma vez incorporado ao fluxo de manufatura, o desenho criado através da ferramenta CAD pode ser considerado a entrada do sistema. Diferentes das demais etapas, onde é possível variar parâmetros para alcançar o resultado ótimo, os desenhos utilizados no sistema são estáticos e representam as peças a serem produzidas. No sistema atual, não são identificados padrões ou características do desenho tão bem como as etapas que deverão ocorrer após as modificações propostas.

## 2.2 Manufatura Auxiliada por Computador ou CAM (*Computer-Aided Manufacturing*)

A raiz da tecnologia CAM surgiu da demanda pela automação das linhas de produção onde a tecnologia de Controle Numérico Computadorizado (CNC) era utilizada. O processo de automação de máquinas utilizando linguagem simples teve como pioneiro John Parsons e a Associação Frank Stulen - Parsons Corporation; em Traverse City, Michigan (ELANCHEZHIAN; SUNDAR, 2007). Os estudos começaram em 1949 após um contrato entre a Força Aérea dos Estados Unidos da América (*U.S. Air Force*) e Parsons Corporation. Através dessa união, foram realizados os primeiros e importantes passos para o começo da criação do código NC, e consecutivamente, a primeira máquina controlada via controle numérico foi criada (SMID, 2000). Com o desenvolvimento da tecnologia NC, foi possível criar as primeiras máquinas automatizadas onde o comando numérico era utilizado, tendo como base a linguagem APT (*Automatically Programmed Tool*); linguagem considerada como sendo a matriarca da tecnologia CAM e dos atuais códigos G.

O propósito almejado pela pesquisa da linguagem APT era desenvolver um modo de automação onde o programador da peça pudesse informar as instruções de usinagem a uma máquina usando uma linguagem similar ao Inglês. A Figura 3 ilustra um exemplo da linguagem APT.

Figura 3 - Exemplo de linguagem APT



Fonte: APT Programming Notes (2014).

Embora a linguagem APT tenha sido muitas vezes criticada por seu tamanho demasiadamente grande para muitos dos computadores da época, início da década de 1960, essa linguagem continua sendo uma grande realização na área da automação programável. Tal linguagem ainda é amplamente utilizada na indústria nos dias atuais; sendo que a maioria das linguagens de programação mais modernas é baseada nos conceitos da linguagem APT (GROOVER, 2008).

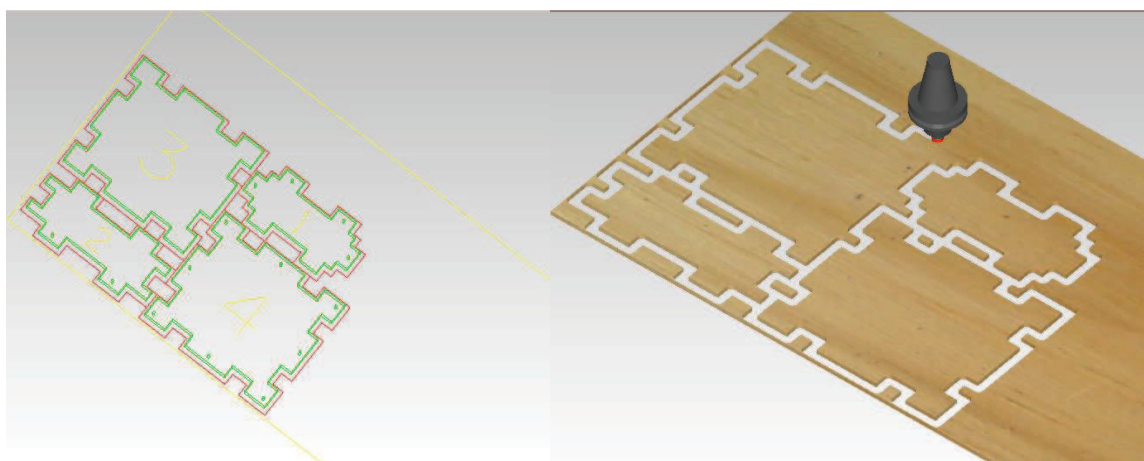
A partir da evolução da linguagem foi criada a atual tecnológica NC, abordada mais detalhadamente na seção 2.6 e, conseqüentemente, as máquinas CNC. A introdução ao uso do CAM iniciado nos anos 50, é considerada como o primeiro impulso para a entrada de produtos eletrônicos, e mais tarde a informática, sendo suporte a evolução dos sistemas de produção. No entanto, o desenvolvimento mais amplo de sistemas CAM foi possível ao se criar o conceito de controle numérico computadorizado (CNC) em máquinas de produção, que é datado de 1970 (NOVAK-MARCINCIN et al., 2013).

Com uma definição muito similar ao CAD, o CAM pode ser definido como o uso de ferramentas computacionais para auxiliar, controlar ou criar operações de um processo de manufatura (GROOVER, 2006). As definições similares não são a única coisa em comum entre essas tecnologias, mas também a grande integração entre os sistemas CAD e CAM ocorreu na década de 80, onde os sistemas começaram a ser também conhecidos por CAD/CAM. Sistemas CAM são dependentes dos sistemas CAD e a integração entre ambos é essencial para a eficiência do conjunto. Isso ocorre já que o sistema CAM permite transformar as informações, trazidas através de geometrias e sólidos criados através do sistema CAD, transformando-as diretamente em programas NC para finalmente serem utilizados em máquinas CNC (NOVAK-MARCINCIN et al., 2013).

Essencialmente, a função do CAM, quando utilizado para a criação de código NC, é criar caminhos de ferramenta. Caminhos de ferramentas tem a função de determinar os movimentos que serão utilizados pela ferramenta de corte no processo de usinagem. Groover (2008) define caminho de ferramenta como uma sequência de segmentos entre linhas e arcos conectados uns aos outros, utilizando uma geometria previamente definida em um sistema CAD para guiar uma ferramenta de corte.

Além do percurso por onde a ferramenta irá realizar o processo de usinagem, o caminho de ferramenta é responsável por carregar todas as informações que serão utilizadas nas etapas de usinagem. Essas informações, também chamadas de parâmetros de corte, dependem do tipo de processo de usinagem a ser utilizado. Todavia, na maior parte dos casos, caminhos de ferramentas carregam informações como: número da ferramenta, frequência de rotação, velocidade de corte, velocidade de avanço, diâmetro da ferramenta, taxa de remoção de material, número de dentes, entre outros parâmetros. Um exemplo de caminho de ferramenta e seus parâmetros de corte são ilustrados através da Figura 4.

**Figura 4 - Caminho de ferramenta e simulação do processo de usinagem**



Fonte: elaborado pelo autor (2014).

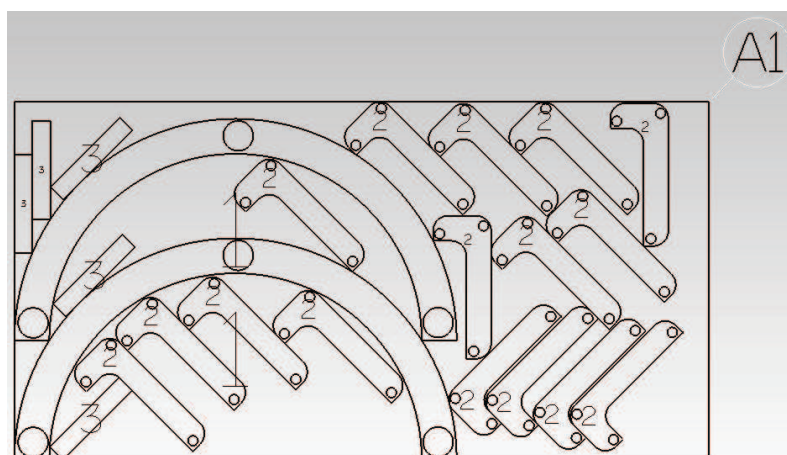
No processo analisado por este trabalho, a ferramenta CAM tem a função de criar os perfis de peças trazidos da etapa anterior (CAD) e atribuir caminhos de ferramenta às geometrias para o desenho utilizando duas dimensões (2D); ou sólidos, para desenhos utilizando três dimensões (3D). Essa é uma etapa importante do processo, já que através dela são definidos todos os parâmetros de usinagem, assim como a escolha de ferramentas. Ou seja, nessa etapa um simples desenho CAD se transforma em uma peça produzida. Uma vez aplicados os caminhos de ferramenta, é possível realizar o processo de otimização de peças (*Nesting*) e a sua parametrização.

### 2.3 Otimização de Peças - *Nesting*

Embora a otimização de peças não ocorra necessariamente após a criação dos caminhos de ferramenta, por fins didáticos, a seção de otimização de peças será apresentada antes da seção de parametrização do processo de *nesting*, uma vez que o melhor entendimento sobre o processo de *nesting* em geral, ajudará também na compreensão do mesmo.

O problema de *nesting*, também conhecido como problema de corte (*Cutting Problem*), problema de empacotamento (*Packing Problem*), ou ainda comumente tratado como problema da mochila (*Knapsack problem*); consiste em encontrar a melhor forma de organizar peças em uma superfície plana e encontrar o melhor *layout*, para que o corte das mesmas seja realizado visando a minimização de matéria prima. Podemos descrever o problema utilizando o seguinte exemplo: dada as dimensões de um retângulo (matéria prima), que são constantes, e definindo as peças que devem ser cortadas; tal problema pode ser definido como a melhor forma de alocar essas peças dentro do retângulo, sem sobrepor as geometrias, de forma a minimizar o comprimento da faixa de material utilizado longitudinalmente (JÚNIOR et al., 2013). Um exemplo prático é demonstrado através da Figura 5.

Figura 5 - *Nesting* aplicado



Fonte: elaborado pelo autor (2014).

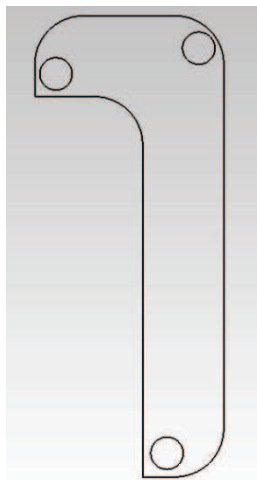
A etapa de *nesting* é de suma importância no processo de manufatura, já que este é responsável por otimizar a matéria prima utilizada, testando um número extremamente grande de possíveis distribuições que representam as formas como as peças podem ser acomodadas. Após essa análise, utilizando a melhor combinação encontrada com base nos parâmetros selecionados na etapa de parametrização (descrito pela seção 2.4), as peças serão encaixadas de forma eficiente no material bruto a ser usinado.

*Nesting* é um processo extremamente versátil e é aplicado a uma vasta variedade de áreas de manufatura, assim como pode ser aplicado a diversos tipos de matéria prima. É possível encontrar esse processo sendo aplicado a áreas extremamente diferentes; como por exemplo, na indústria têxtil (JÚNIOR et al., 2013) e também na indústria naval (NAGASHIMA et al., 2006). Wäscher et al. (2007) reforça essa ideia, afirmando que esse problema é um caso típico de dimensões abertas e interessa especialmente as empresas de produção em massa.

A redução de custos, proporcionada pelo sistema de *nesting*, não é mensurada apenas pelo uso de matéria prima. O sistema de *nesting* pode auxiliar também na redução do tempo de configuração de máquina (*setup*), reposição de chapas na mesa da máquina, dentre outros fatores. Uma vez que em um processo otimizado o operador possivelmente realizará um número reduzido destas ações (HERRMANN; DELALIO, 2001).

A otimização ocorre de forma que os perfis das peças, inseridas na primeira etapa, sejam analisados e tenham seus contornos extraídos. Uma vez que os contornos das peças sejam reconhecidos e se tenha a parametrização do processo de *nesting* esses perfis podem ser utilizados para identificar os limites da peça, assim como áreas internas onde outras peças podem ser inseridas. Um exemplo de peça, adequada para o processo de *nesting* é apresentada pela Figura 6.

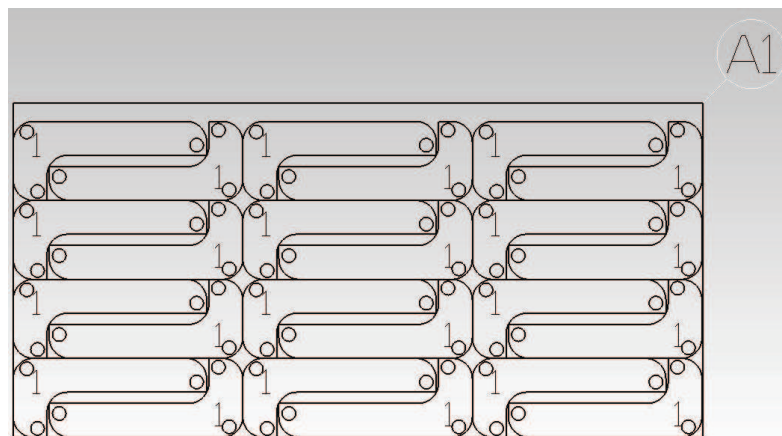
**Figura 6 - Nesting - Exemplo de peça a ser otimizada**



Fonte: elaborado pelo autor (2014).

Tendo as informações necessárias para iniciar o processo de otimização, o mesmo pode ser iniciado utilizando a parametrização, definida como: rotação da peça, número de peças, distância entre peças, etc. Todas as peças designadas ao projeto serão encaixadas em uma chapa de material bruto com dimensões que também são especificadas na parametrização do processo. Desta forma, é possível obter um resultado otimizado, onde a minimização do uso de matéria prima ocorra, conforme ilustrado pela Figura 7.

**Figura 7 - Nesting - Resultado obtido através da otimização**



Fonte: elaborado pelo autor (2014).

Todavia, uma vez que o trabalho proposto aborda o processo de otimização em um sistema CAM não apenas as geometrias e perfis das peças devem ser levados em consideração durante a otimização, mas também os caminhos de ferramentas a eles atribuídos. Considerar os caminhos de ferramenta é uma etapa importante já que, dependendo do diâmetro da ferramenta, esta pode sobrepor outra peça danificando o produto final.

No processo atual, a ferramenta de *nesting* é utilizada para sua função básica, ou seja; com base na parametrização realizada na etapa anterior, otimizar o uso de matéria prima e criar resultados que serão posteriormente analisados pelo usuário.

## 2.4 Parametrização *Nesting*

Processo realizado antes da etapa de *nesting*, a parametrização é uma parte de suma importância, pois visa definir como o sistema *nesting* será realizado. A parametrização define quais peças serão processadas pelo sistema de *nesting*, assim como suas informações específicas, tais como: quantidade de peças, rotação, prioridade, entre outros parâmetros. A parametrização, todavia, não envolve apenas as características relacionadas à peça. A parametrização pode ainda envolver decisões ligadas diretamente à forma como o processo de *nesting* será realizado e quais são as suas prioridades, assim como a escolha de quais métodos serão utilizados, usualmente algoritmos diferentes.

Outra escolha importante é o material a ser utilizado, assim como todas as suas respectivas características. A escolha do tipo de material, usualmente, não envolve grande grau de dificuldade, pois é definida pelo setor de *design* ou engenharia juntamente com o projeto inicial. Porém, a escolha da chapa do respectivo material pode ser uma etapa complexa, já que a escolha errada pode levar ao uso excessivo de matéria prima.

Essa escolha se torna ainda mais complexa quando o processo envolve sobras aproveitáveis trazidas de processos realizados anteriormente. O assunto sobre sobras aproveitáveis tem atualmente ganhado foco na literatura e recebido diversas contribuições tais como em: Cui e Yang (2010), Abrantes (2013) e Cherri e Vianna (2013).

A escolha dos parâmetros é uma etapa que demanda alta interação do usuário e pode variar o resultado drasticamente de acordo com as escolhas realizadas. Essa interação do usuário exige um alto nível de conhecimento sobre o processo que será realizado, ou seja, demanda que o usuário seja um especialista na respectiva área de manufatura, conhecendo as características de produção do produto assim como quais recursos deverão ser utilizados.

Dada a rotação da peça como exemplo, em diversos materiais tais como madeiras, não é possível rotacionar a peça, pois se deve obedecer ao veio da madeira. Por outro lado, em chapas de alumínio, por ser um material liso e homogêneo, as peças geralmente podem ser rotacionadas sem afetar o resultado final.

O uso de recursos que não são necessários para o processo produtivo pode aumentar drasticamente o tempo de processamento. Como na maioria dos problemas de otimização combinatória, é preciso preocupar-se com o tempo necessário para obter uma resposta de qualidade em tempo hábil, fator muito importante em processos de manufatura (CANELLIDIS et al., 2013).

Alguns dos parâmetros iniciais relacionados ao método de *nesting* a ser utilizado, peças e materiais encontrados em ferramentas de *nesting* podem ser observados na Tabela 2.

Tabela 2 - Parâmetros principais do *nesting*

| Parâmetros Primários     |                              |                                                                                                                                                      |
|--------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tipo                     | Nome                         | Descrição                                                                                                                                            |
| Método de <i>Nesting</i> | <i>True Shape</i>            | Quando utilizado, esse método leva em conta o formato real das peças.                                                                                |
|                          | Retangular                   | Quando utilizado, esse método enclausura a peça em um retângulo e leva em consideração esse retângulo quando realizando o processo de <i>nesting</i> |
| Parâmetros de Peça       | Nome                         | Nome ou código da peça                                                                                                                               |
|                          | Número de peças              | Número de peças a serem inseridas na chapa                                                                                                           |
|                          | Rotação (se possível)        | Define se as peças podem ser rotacionadas para aprimorar seu encaixe na chapa                                                                        |
|                          | Espelhamento da peça         | Define se é possível espelhar a peça para aprimorar seu encaixe na chapa                                                                             |
|                          | Definir como peça pequena    | Define se a peça pode ser inserida em áreas internas de peças grandes, ou alocada em áreas especiais para esse tipo de peça                          |
|                          | Prioridade                   | Define a prioridade da peça: peças com maior prioridade são acomodadas na chapa primeiro                                                             |
| Parâmetros de Material   | Número de lote               | Define um lote: peças com mesmo lote são acomodadas na chapa em conjunto, garantindo que estarão na mesma chapa e serão produzidas em conjunto       |
|                          | Tipo de material             | Define tipo de material a ser utilizado                                                                                                              |
|                          | Dimensões da chapa           | Define as dimensões do material que será utilizado no processo                                                                                       |
|                          | Número de chapas disponíveis | Define o número de chapas da dimensão definida disponíveis para o processo de otimização                                                             |
|                          | Espessura chapa              | Define a espessura da chapa                                                                                                                          |
|                          | Definir zonas                | Define zonas onde peças pequenas devem preferencialmente ser inseridas                                                                               |

Fonte: elaborado pelo autor (2014).

A parametrização se torna um fator mais agravante quando o usuário lida diretamente com as definições relacionadas ao processo de *nesting*, já que cada ferramenta ou algoritmo pode oferecer diferentes opções. Por esse motivo, para obter bons resultados, o usuário além de ser especialista no processo produtivo, também tem a necessidade de ser especialista no *software* que está utilizando para realizar a etapa de *nesting*. A Tabela 3 traz um levantamento dos parâmetros de *nesting* oferecidos por dois *softwares* CAM (Alphacam e Radan) que possuem essa ferramenta integrada.



Tabela 3 - Parametrização dos processos de *nesting*

| Parametrização Dos Processos de <i>Nesting</i>         |                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nome                                                   | Descrição                                                                                                                                                                                                                                                                                                |
| Borda ou porção da chapa para começar o <i>nesting</i> | Define em qual extremidade da chapa o processo de <i>nesting</i> será iniciado, Ex. Borda esquerda inferior ou borda direita superior                                                                                                                                                                    |
| Distância mínima entre peças                           | Define a distância mínima que uma peça deve ter até a outra peça                                                                                                                                                                                                                                         |
| Distância até a borda da chapa                         | Define a distância mínima que deve existir entre o limite da chapa e as peças                                                                                                                                                                                                                            |
| Tempo disponível para otimização em cada chapa         | Define, para cada chapa, o tempo máximo que o processo pode utilizar na busca de um resultado ótimo                                                                                                                                                                                                      |
| Cortar peças pequenas primeiro                         | Quando utilizado, reordena todas as operações de forma que a área externa das peças seja cortada da menor para a maior.                                                                                                                                                                                  |
| Ordenar peça por peça                                  | Força a ferramenta de <i>nesting</i> a realizar o processo peça por peça, dessa forma todas as peças estarão próximas possibilitando que sejam usinadas em conjunto posteriormente                                                                                                                       |
| Agrupar peças separadamente                            | Cria grupos independentes para cada peça, agrupando as geometrias e caminhos de ferramenta ligados à mesma                                                                                                                                                                                               |
| Utilizar distância entre ferramentas para corte        | Define se os caminhos de ferramenta podem ser acomodados nos espaços entre peças criados pela opção que define espaço mínimo entre elas                                                                                                                                                                  |
| Prevenir <i>nesting</i> em áreas internas              | Define se áreas internas das peças grandes podem ser utilizadas para acomodar outras peças pequenas que caibam nessas áreas                                                                                                                                                                              |
| Remover grupos                                         | Remove todo e qualquer grupo de geometrias ou caminhos de ferramenta que tenham sido criados antes do processo de <i>nesting</i>                                                                                                                                                                         |
| Repetir primeira linha e coluna                        | Após alocar a primeira peça, todas as demais seguirão esse padrão criando assim um resultado uniforme                                                                                                                                                                                                    |
| Prevenir atualização de tela                           | Possibilita esconder os resultados sendo processados, mostrando assim apenas o resultado final ao invés de atualizar a tela ao final do processamento de cada chapa                                                                                                                                      |
| Tentar rotacionar todas as peças                       | Quando utilizado, faz com que a ferramenta de <i>nesting</i> tente rotacionar todas as peças antes de acomodá-las na chapa                                                                                                                                                                               |
| Distancia extra para entrada da ferramenta             | Quando o corte aplicado à peça possui uma forma específica de entrada/saída da ferramenta ( <i>Lead-In/Out</i> ), essa distância extra é acrescentada para prevenir danos às outras peças                                                                                                                |
| Minimizar troca de ferramentas                         | Junta todos os processos de corte que utilizam a mesma ferramenta e os realiza de forma simultânea, evitando assim o excesso de retrações de ferramenta                                                                                                                                                  |
| Furar e então realizar operações internas              | Dá ao usuário a possibilidade de definir que os ciclos de furação serão os primeiros ciclos de usinagem realizados                                                                                                                                                                                       |
| <i>Onion Skin</i> em peças pequenas                    | Quando utilizado, essa opção cria um corte extra na peça com uma profundidade ligeiramente menor do que a espessura da chapa, deixando assim uma fina camada (conhecida como <i>Onion Skin</i> ) que prende o material à chapa. Um segundo corte é realizado posteriormente para liberar a peça da chapa |
| Ligar peças (Bridge Nesting)                           | Quando utilizado, faz com que uma ligação entre geometrias próximas seja criada. Quando aplicado a caminhos de ferramenta, pode evitar retrações                                                                                                                                                         |
| Cortar parte inteira                                   | Reordena todas as operações de usinagem de forma que cada peça seja completamente cortada antes de seguir para a próxima                                                                                                                                                                                 |
| Processar peças pequenas primeiro                      | Ao contrário do padrão, força a ferramenta de <i>nesting</i> a acomodar primeiramente as peças pequenas e depois as grandes                                                                                                                                                                              |
| Suprime verificação de contornos                       | Caso selecionado suprime a verificação que é realizada para avaliar a integridade dos contornos das peças e caminhos de ferramentas                                                                                                                                                                      |

Fonte: elaborado pelo autor (2014).

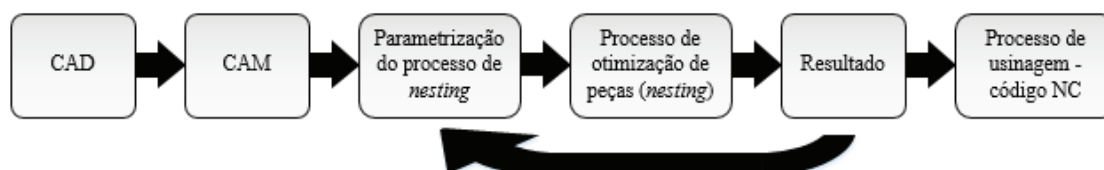
Atualmente, a parametrização das etapas é totalmente manual e demanda grande experiência do usuário, além de tempo para escolher os melhores parâmetros a serem utilizados. Além disso, muitas vezes é preciso realizar o processo de *nesting* diversas vezes utilizando parâmetros diferentes até que um resultado ótimo ou próximo do ótimo seja encontrado. Porém, muitas vezes, os sistemas de *nesting* não oferecem ferramentas para o controle de chapas disponíveis em estoque ou ainda qualquer ligação com algum sistema integrado de gestão empresarial (ERP), dificultado assim o processo de escolha da matéria prima a ser utilizada.

## 2.5 Resultados

A análise do resultado obtido pelo processo de *nesting*, utilizando a parametrização pré-selecionada, é o momento onde o usuário aceita a solução obtida através do sistema de *nesting* e, caso a mesma seja admitida, o usuário pode prosseguir para a criação do código NC e enviá-lo para a máquina CNC para posterior usinagem do produto.

Todavia, o resultado obtido pode não estar dentro do padrão de qualidade esperado pelo usuário fazendo assim com que seja necessário retornar à etapa de parametrização por diversas vezes até que o resultado esteja de acordo com o esperado ou ainda não exceda o custo máximo permitido. Caso seja necessário retornar às etapas anteriores, o processo de *nesting* e a análise dos resultados deverão ser repetidos conforme ilustra a Figura 8.

Figura 8 - Retorno ao processo de parametrização



Fonte: elaborado pelo autor (2014).

No problema abordado por esse trabalho, essa é uma etapa totalmente manual; todavia deverá permanecer assim mesmo após a otimização de etapas proposta pelo estudo, uma vez que esse resultado será apresentado para o usuário como resultado final.

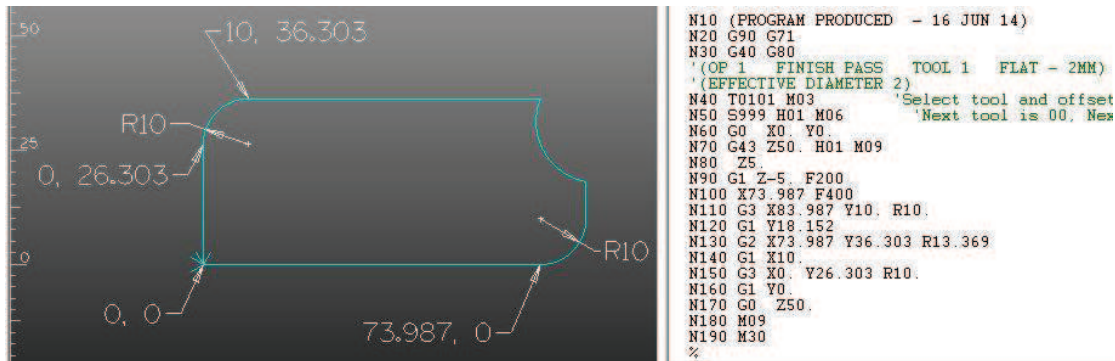
## 2.6 Código NC

Dado ao fato de que caso o usuário chegue a essa etapa de manufatura, significa que o resultado obtido pela etapa *nesting* foi aceita e, portanto; essa é a etapa final do processo de manufatura e o produto final será obtido através da criação do código NC, seguido de usinagem. O código NC possibilita realizar ações com base em informações numéricas. Mais precisamente, o controle numérico é utilizado para controlar máquinas desenvolvidas para obedecer instruções numéricas, usualmente através de códigos (OBERG et al., 2000).

O código mais utilizado é o código G, sendo este código formado por um bloco de dados que possui informações suficientes para enviar uma instrução completa para uma máquina. Cada bloco é constituído de uma letra como prefixo e um ou mais dígitos numéricos que indicam qual ação a máquina deverá realizar. A letra inicial é responsável pelo endereçamento da função, por exemplo: X é a letra que endereça a máquina a movimentar na

direção do eixo X, de forma similar Y corresponde à movimentação no eixo Y. Também de suma importância para o código G, a palavra preparatória consiste da letra G seguida de um ou dois dígitos numéricos em um bloco único. Esse bloco é responsável, por indicar para o sistema de controle, qual comando ou instrução deve ser realizada. Por exemplo, comando G01 corresponde a uma interpolação linear e moverá a ferramenta de corte em linha reta; já o código G02 moverá a ferramenta de corte de forma circular no sentido horário e G03 no sentido anti-horário. Um exemplo de código G utilizando os comandos G1, G2 e G3 é demonstrado pela Figura 9.

**Figura 9 - Código G utilizando os comandos G1, G2 e G3**



Fonte: elaborado pelo autor (2014).

Embora esta seja uma etapa muito importante, a criação de código NC será tratada ao longo deste estudo como a etapa final do processo e, uma vez que no processo atual o código NC é criado de forma eficiente; a operação de otimização proposta pelo trabalho não irá alterar a etapa de criação do código NC.

### 3 REVISÃO BIBLIOGRÁFICA: SISTEMAS INTELIGENTES E OTIMIZAÇÃO

Em tempos onde o escopo das grandes indústrias e empresas está diretamente focado no lucro, o ato de tomar decisões é algo comum e corriqueiro. Embora seja uma rotina trivial, o ato de decidir qual caminho deve ser seguido é de extrema importância para o bom andamento de uma empresa. Os problemas analisados pelos tomadores de decisões no ambiente comercial competitivo são de ritmo frenético. Os dias em um ambiente comercial são geralmente de extrema complexidade e os problemas podem ser resolvidos por vários cursos de ações possíveis. A avaliação dessas alternativas e a escolha do melhor curso de ação representam a essência da análise de decisão (RAGSDALE, 2011).

Embora, de uma forma geral, as principais tomadas de decisão sejam usualmente atribuídas aos gestores e executivos; outros trabalhadores, em diversos níveis hierárquicos e que desempenham atividades distintas em grandes empresas, possuem a necessidade de efetuar tais ações diariamente para o bom funcionamento de uma companhia. Freitas et al. (1996) justificam que, essencialmente, as atividades desempenhadas em uma organização são atividades de tomada de decisão e de resolução de problemas. Esses fatores agravam a complexidade das mesmas e para que as decisões sejam realizadas, o pleno entendimento e um vasto estudo sobre o processo se fazem obrigatórios.

Dada tal complexidade e a evolução da tecnologia da informação, os sistemas de apoio à tomada de decisão (SAD) são ferramentas poderosas para auxiliar a análise dos riscos e auxiliar na escolha de qual caminho deve ser seguido para chegar ao resultado esperado. Sprague e Carlson (1982) definem o SAD como sistemas computacionais que ajudam os responsáveis pelas tomadas de decisões a enfrentar problemas estruturais, através de uma interação direta com modelos de dados e análises.

Os primeiros conceitos relacionados à área de apoio à tomada de decisão foram elaborados na década de 70, sendo até tal momento chamado de sistemas de decisões gerenciais segundo Scott Morton (1971). Não obstante, em uma análise profunda das contribuições de Herbert Simon no campo da teoria organizacional, Balestrin (2001) demonstra o quão trivial o assunto se tornou para este campo, uma vez que apenas as obras de Simon relacionadas à área somam aproximadamente 845 (oitocentos e quarenta e cinco) títulos originais entre artigos e livros publicados no período de 1937 a fevereiro de 2001.

No que diz respeito ao desenvolvimento de ferramentas de apoio à tomada de decisão, esses são sistemas de difícil formulação, uma vez que se tratam de ferramentas dinâmicas e devem ser constantemente adequadas às mudanças do mercado. Sprague e Watson (1993) reforçam tal ideia, citando que as abordagens tradicionais para a análise e desenvolvimento de uma ferramenta de decisão mostraram-se inadequadas, uma vez que não existe teoria abrangente sobre o processo decisório. Por esse motivo, um SAD para ser desenvolvido precisa levar em consideração a experiência e o retorno dos usuários, com a finalidade de assegurar que a ferramenta está sendo desenvolvida de forma eficiente.

Por se tratar de uma ferramenta dinâmica, sistemas de apoio à decisão devem ser receptivos a atualizações, mudanças ou adequações ao longo de sua vida útil. Em curto prazo, é esperado que o sistema apresente respostas dentro de um escopo relativamente restrito. Em médio prazo, o sistema deve absorver e aprender sobre seus recursos, dessa forma modificando ou amentando o escopo ou ainda o domínio onde estas respostas se encontram. Após o amadurecimento do sistema, o sistema passa a tolerar e aceitar novos comportamentos e novos recursos, sendo esses muitas vezes fora do escopo “esperado” (SIMON, 1980). Para

tanto, o sistema pode receber ajuda de recursos estudados pela área da Inteligência Artificial, assim como apresentado na seção 3.3.

Por se tratar de um sistema dinâmico muitas vezes com diversas características estocásticas, a melhor forma de simular possíveis eventos é através da modelagem do mesmo. Todavia, deve-se criar um modelo válido, ou seja; um modelo que represente de forma precisa as variações e características ligadas ao problema de decisão em evidência. Modelos nos permitem ganhar o conhecimento e entendimento sobre o objeto ou problema de decisão que está sendo investigado. O objeto final de usar modelos é melhorar a análise de decisão (RAGSDALE, 2011).

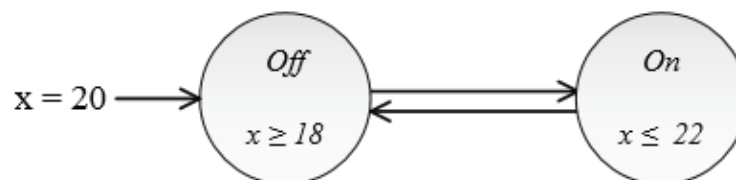
### 3.1 Sistemas Híbridos

Conforme Ghosh (1997), um sistema híbrido pode ser definido como um sistema que apresenta, simultaneamente, comportamentos dinâmicos de tempo discreto e tempo contínuo. Em seu conceito clássico, um sistema híbrido possui dois tipos distintos de componentes: um arranjo de sistemas dinâmicos responsáveis por definir o modo de cada sistema e um autômato, ou sistema de regras, responsável pelo chaveamento entre os sistemas dinâmicos.

Sistema dinâmico, em um sistema híbrido, é responsável por representar o comportamento de tempo contínuo. Muitas vezes é descrito em forma de equações diferenciais não lineares de certa dimensão, com parâmetros de controle. Por outro lado, os autômatos ou sistemas baseados em regras representam a componente discreta do sistema; e por esse motivo são responsáveis pelo chaveamento do mesmo, de forma que as regras impostas sejam respeitadas (GHOSH, 1997).

Henzinger (1996) exemplifica o funcionamento de um sistema híbrido através de um exemplo aplicado à automação de um termostato. Nesse exemplo, apresentado de forma simplificada pela Figura 10, a temperatura é representada pela variável  $X$ . Nesse termostato existe um chaveamento indicado por “*On*” quando ligado e “*Off*” quando desligado. Quando ligado, a temperatura tende a subir e desligado, diminuir. O sistema inicia desligado em uma temperatura de 20 graus e seguindo as regras de chaveamento, o sistema terá de ligar. Ou seja, o sistema deve mudar de estado quando a temperatura atingir 18 graus e esse estado mudará novamente para “*Off*” quando a temperatura atingir 22 graus.

Figura 10 - Exemplo de chaveamento em um sistema híbrido



Fonte: adaptado de Henzinger (1996).

Na área de Inteligência Artificial, sistemas híbridos são aqueles que possuem algum tipo de fusão, combinação ou ainda transformação de sistemas que até o momento funcionavam de forma individual e que apresentam características que não sejam coincidentes entre os mesmos. Essa junção os torna uma ferramenta robusta e eficiente quando comparada às suas funcionalidades individuais.

Devido à sua arquitetura que oferece uma estrutura básica e uma metodologia para a análise dos autômatos, sistemas híbridos são fortes candidatos a uma boa solução. Além disso, estudos de caso demonstram que a modelagem híbrida, utilizando o conceito de autômatos ou sistemas de regras, é eficiente na análise de sistemas envolvendo *software* e hardware; inclusive para processos de assincronia de relógios, sistemas de tempo real, protocolos para controle de plantas industriais, veículos, robôs e etc. (HENZINGER et al., 1998).

### 3.2 Subsistemas de um SAD

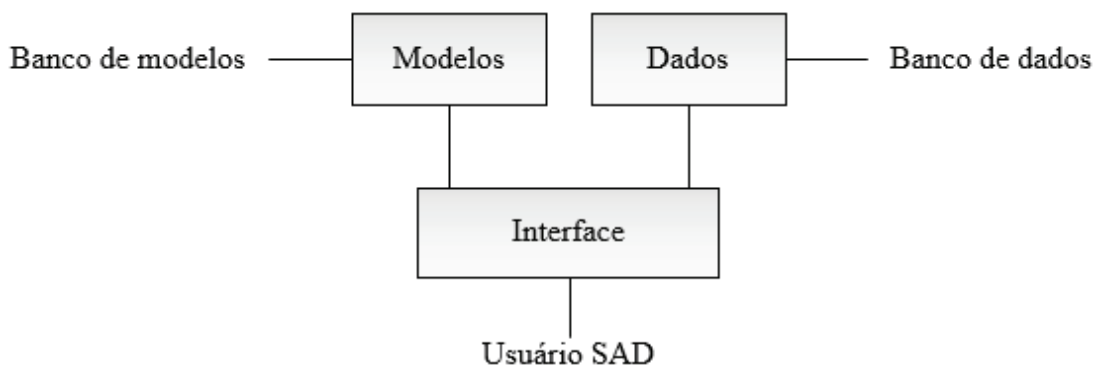
Sistemas de tomada de decisão devem possuir, no mínimo, três componentes, que podem ser separados, teoricamente, em categorias distintas (SPRAGUE; CARLSON, 1982); tais como:

- a) **Banco de dados:** Subsistema responsável por armazenar e gerenciar as informações relevantes para o auxílio na tomada de decisão.
- b) **Banco de modelos:** Subsistema responsável por armazenar modelos de decisões.
- c) **Interface:** Subsistema responsável pelo gerenciamento da interface entre usuário e o SAD.

O detalhamento dos subsistemas citados acima é realizado ao longo das seções 3.2.1, 3.2.2 e 3.2.3.

A Figura 11 ilustra a relação dos componentes de um SAD definidos posteriormente.

**Figura 11 - Ligação entre componentes de um SAD**



Fonte: elaborado pelo autor (2014).

#### 3.2.1 Banco de Dados

Tratando-se de um subsistema onde um grande volume de informações é armazenado, é necessário mais do que apenas alocar essas informações no banco de dados, mas também gerenciá-las de forma inteligente. Para tanto, um sistema de gerenciamento de dados deve efetuar a função de armazenamento e manutenção das informações que serão utilizadas no SAD. De acordo com Peter Rob e Carlos Coronel (2006), o banco de dados relacional é o mais utilizado quando o mesmo está relacionado ao desenvolvimento de um SAD. O principal motivo está na flexibilidade associada a esse tipo de sistema e também a sua facilidade na

normalização dos dados; ajudando de forma eficaz na redução de redundâncias quando o sistema possui um número drasticamente elevado de informações.

Habitualmente chamado de banco de dados DBMS, do inglês *Database Management System*, este é o componente do SAD responsável pelo gerenciamento do banco de dados entrelaçados ao sistema de apoio à decisão. Um DBMS deve estar sempre disponível e operante para auxiliar no compartilhamento e na manipulação de dados de forma eficiente (HIROUCHI; KOSAKA, 1984).

Para tanto, deve permitir a extração, obtenção e inserção de dados a qualquer momento. Além disso, o DBMS é interligado ao sistema como um todo; uma vez que no banco de dados devem estar contidas as informações fornecidas pelo subsistema onde se encontra a interface com o usuário, histórico de dados de entrada e seus respectivos resultados, cenários antes já simulados pelo SAD, entre diversas outras informações. Por se tratar de um sistema inteligente que aprende com as simulações anteriores, o DBMS acaba por viabilizar o funcionamento de toda a ferramenta, uma vez que sem os dados armazenados no banco de dados a comunicação entre os módulos que integram o SAD não seria possível (HENNEMANN et al., 2006).

No que diz respeito aos tipos de dados, existem algumas diferenciações entre as abordagens de bancos de dados dos sistemas tradicionais e as aplicáveis a um sistema de apoio à decisão. A principal diferença está ligada ao tipo de dados que se encontram em um SAD; nesses casos tem-se um conjunto de fontes de dados muito mais complexo e com maior riqueza vinda de fontes internas e externas (SPRAGUE; WATSON, 1993):

- **Internas:** Dados relacionados intrinsecamente à companhia que está fazendo uso do SAD, sendo usualmente informações relevantes ligadas aos custos de produção. Além disso, podem conter dados de compras, paradas do processo produtivo, informações sobre clientes e regiões promissoras (HIROUCHI; KOSAKA, 1984).
- **Externas:** Geralmente mais complexo, como listado por Vasile Prejmerean (2013), podem envolver dados de mercado, governamentais, informações econômicas, valores de matéria prima, entre outras informações que não estão intrinsecamente ligadas à companhia, mas que influenciam na tomada de decisão. Usualmente são dados com variação intensa e que dependem diretamente de terceiros.

Dada essa complexidade e grande volume de dados, as buscas e administração de dados devem passar por um processo rigoroso. Segundo Linoff e Berry (2011), por muitos anos empresas armazenaram uma grande quantidade de informações sobre seus processos e uma vez que esse grande montante de dados exista; as empresas desejam utilizá-los de forma que os mesmos possam criar uma vantagem comercial através do seu estudo utilizando ferramentas estatísticas ou ligadas à área da Inteligência Artificial.

Em virtude da grande quantidade de dados e de sua complexidade, processos de mineração de dados são constantemente aplicados a sistemas SAD. A mineração de dados é um processo que possibilita, aos setores gerenciais, encontrar padrões e relacionamentos ocultos entre dados a partir de dados já existentes (PHRIDVIRAJ; GURURAO, 2014). Além disso, conforme cita Daniel J. Power (2002), alguns especialistas acreditam que o uso da mineração de dados, por parte de uma empresa, proporciona para a mesma certa vantagem competitiva em relação às demais que não fazem uso de tais recursos. Os recursos associados à mineração de dados podem ser utilizados não só para testes de hipótese, mas também para o ganho de conhecimento.

### 3.2.2 Banco de Modelos

Segundo Goldbarg e Luna (2005), modelos podem ser definidos como a representação simplificada da realidade que preservam, para determinadas situações e enfoques, uma equivalência adequada. Uma vez que os modelos atribuídos ao SAD apresentado pelo trabalho visam maximizar ou minimizar critérios de otimização, por conseguinte, podemos definir os modelos atribuídos ao sistema como modelos de otimização.

Conforme Paul Gray (1994), modelos são um componente fundamental e chave para grande parte dos sistemas de apoio à decisão. Sistemas desenvolvidos para efetivamente auxiliar nos processos de decisão, devem, harmonicamente, adaptar os modelos que representam os motivos e caminhos tomados por um tomador de decisão ligado à área onde o problema está envolvido.

Um dos aspectos promissores de um SAD é sua capacidade de integrar acesso aos dados contidos no sistema e modelos de decisões. Isso pode ser realizado através da inserção de modelos de decisão em um sistema de informação que utilize banco de dados como mecanismo de integração e comunicação entre modelos (SPRAGUE; WATSON, 1993).

O modelo é uma etapa fundamental na análise de qual decisão se deve tomar. O processo de análise de decisão geralmente utiliza a construção de modelos a fim de efetuar diversas simulações e análises. Essas análises podem envolver diferentes cenários, usualmente ligados a perguntas do tipo “e se...”, em inglês conhecida como “*what-if*” (BOHANEK, 2001).

Muitas vezes os modelos são criados de forma errônea; isto acontece devido à modelagem individual de etapas distintas do processo produtivo, ao invés de analisá-lo como um problema único e integrado. A modelagem individual das etapas através de modelos separados é possível. Todavia, é nessa etapa onde a inserção desses modelos deve ocorrer em um sistema de informação, tornando assim as decisões integradas (SPRAGUE; WATSON, 1993).

De acordo com Paul Gray (1994), algumas afirmações e suposições podem ser feitas quando modelos são aplicados à tomada de decisão:

- a) Modelos são utilizados em todos os níveis do processo de tomada de decisão. Em todos os aspectos da tomada de decisão, modelos que simulam o sistema real são utilizados nos processos para se chegar a uma solução.
- b) Maior do que a esfera onde uma simples decisão é tomada, os tomadores de decisão executam muitas tarefas ligadas ao gerenciamento de modelos. Observações ao longo do tempo são responsáveis por determinar se essas tarefas são realizadas de forma consciente.
- c) Os tomadores de decisão envolvidos no processo devem reconhecer, explicitamente, quais modelos demonstraram-se úteis no passado e possuem confiabilidade suficiente para serem reutilizados em processos futuros de tomada de decisão.
- d) Modelos, em sua grande maioria, são independentes dos dados. Modelos podem ser manipulados independentemente dos dados que são utilizados no processo de tomada de decisão. Embora existam relações estruturais entre os dados e os procedimentos os modelos por si só são recursos a serem manipulados independentemente.



### 3.2.3 Interface

Interface pode ser definida como o local onde acontece toda e qualquer interação entre usuários e máquinas, sempre visando ser amigável e de fácil interação. Na literatura relacionada a sistemas de apoio à decisão, a interface também é conhecida como caixas de diálogo. Mais do que isso, como demonstrado por Daniel J. Power (2002), uma interface efetiva com o usuário é um dos componentes mais importantes de um SAD; porém é indubitavelmente importante quando esse sistema é utilizado diretamente por um gerente. Evidencia-se a importância através de uma pesquisa que mostra que quanto mais amigável e de fácil uso for um SAD, maior é a chance de um gerente realmente usar tal sistema.

A importância da interface com o usuário pode ser facilmente evidenciada no dia-a-dia das pessoas ligadas à tecnologia. Programas de computador e sistemas operacionais de dispositivos móveis trazem frequentemente atualizações e tem sua interface constantemente modificada. Isso acontece pela incessante busca de melhorar a interação e tornar essa interface cada vez mais amigável. Ben Shneiderman (2005) cita que a atuação humana e as experiências envolvendo computadores e sistemas de informação continuarão a ser alvos constantes de pesquisas e desenvolvimentos pelas próximas décadas.

Uma interface pode ser constituída de estilos específicos de diálogos, ou muitas vezes pode ser constituída de diversos estilos de diálogos diferentes. O conceito de estilo de diálogo pode ser de vários tipos, tais como: interação envolvendo perguntas e respostas, linguagem de comandos, menus ou ainda interação onde usuário tenha de preencher campos. Cada estilo tem suas vantagens e desvantagens, dependendo diretamente do usuário e das atividades que devem ser realizadas (SPRAGUE; WATSON, 1993).

Em contrapartida, para Druzdzel e Flynn (2002), uma boa interface é mais do que apenas um conjunto de diálogos. Uma interface eficiente deve conseguir manipular e lidar com a construção e a análise de modelos, levando em conta a estrutura do problema, assim como cálculos numéricos e a escolha e otimização das variáveis de decisão.

Outro fator importante é a clareza e facilidade de uso da interface. Usualmente, busca-se uma interface que faça uma aproximação fiel ou muito próxima do modelo real. Essa aproximação com a realidade e a facilidade de uso devem ser feitas de forma que vise melhorar o entendimento do usuário quanto ao sistema de apoio a decisão. Caso o usuário não possua um sólido entendimento, então o mesmo pode aceitar ou rejeitar as sugestões feitas pelo sistema por razões equivocadas; podendo fazer com que o desempenho do processo de decisão seja afetado (LEHNER et al., 1989).

### 3.3 Sistemas Especialistas

Dado o mercado atual, muitos acreditam que controlar o conhecimento conduz à liderança comercial e ao poder. Mesmo se essa crença for verdadeira, empresas só atingem bons resultados se o conhecimento for dividido entre seus funcionários e demais interessados (POWER, 2002). Para tanto, sistemas computacionais podem ser utilizados para gerir e armazenar essas informações. Esses sistemas não só necessariamente são utilizados para armazenar o conhecimento, mas também podem ser utilizados para criar relações lógicas, probabilísticas ou encontrar padrões comuns nas informações.

Quando discutindo sobre sistemas de apoio à tomada de decisão com base em conhecimento, é comum fazer um vínculo direto a sistemas especialistas. Ribeiro (1987) define sistema especialista como um programa que tem a capacidade de manipular problemas reais e complexos e reproduzir e simular o raciocínio de um especialista humano em sua área específica de atuação. Geralmente, consideramos uma pessoa especialista em um assunto quando essa pessoa possui um amplo conhecimento em uma área ou assunto.

Especialistas, por sua vez, sabem realizar as perguntas corretas para chegar a uma conclusão que diagnostica o problema e, por sua vez, geram conselhos que levam a uma potencial solução do mesmo. Sistemas especialistas, por sua vez, não são diferentes; eles devem ser capazes de, com base nas informações coletadas, gerar potenciais soluções para um problema (POWER, 2002). Por simular a mente humana, logo, é possível inferir que este se trata de um método ligado à área de Inteligência Artificial.

Sistemas tradicionais são projetados de forma que as soluções apresentadas sejam determinísticas, ou seja, o desenvolvimento de um sistema tradicional é focado na resolução de um problema bem determinado. Além disso, sistemas tradicionais são projetados para analisar dados de maneira repetitiva, chegando sempre a uma resposta correta. Diferente de um sistema tradicional, um sistema especialista é projetado de forma adaptiva ao problema, solucionando problemas que um sistema tradicional não resolveria. Caso gerasse uma solução, até a obtenção do resultado, um longo tempo de processamento seria necessário, em muitos casos inviabilizando o uso do sistema tradicional (RIBEIRO, 1987).

Dada a complexidade dos problemas submetidos aos sistemas especialistas, esses sistemas necessitam utilizar as técnicas e ferramentas disponíveis na área de atuação da Inteligência Artificial. Essas ferramentas tentam reproduzir o pensamento humano e sua inteligência tentando imitar seu aprendizado, compreensão de linguagem, raciocínio, reconhecimento de padrões, indução e decisões lógicas. De acordo com Rich e Knight (1994), Inteligência Artificial também pode ser definida como a área de estudo que analisa como fazer os computadores realizarem tarefas que, atualmente, humanos podem fazer melhor.

De acordo com Giarratano e Riley (1989), algumas das vantagens de se usar um sistema especialista são:

- a) Aumento da disponibilidade: Um sistema especialista pode estar disponível em qualquer computador.
- b) Redução de custos: O custo de prover apoio aos processos de tomada de decisão é reduzido drasticamente de acordo com o número de usuários, quando comparado a um especialista humano.
- c) Redução de risco: Sistemas especialistas podem trabalhar em condições onde um especialista humano não seria submetido.
- d) Permanência da informação: Diferente de um especialista humano que pode deixar a empresa, se aposentar ou se afastar por qualquer motivo, em um sistema especialista o conhecimento é permanente e acumulativo.
- e) Especialistas múltiplos: Em um sistema especialista, o conhecimento de diversos especialistas pode ser manipulado de forma que esses possam trabalhar simultaneamente. Harmon e King (1991) reforçam que, quando somado o conhecimento de diversos especialistas, esta união excede o conhecimento de um especialista único.

- f) Resposta rápida: Sistemas especialistas conseguem gerar respostas rápidas ou em tempo real mesmo para problemas complexos. Dependendo das aplicações utilizadas, essa resposta pode ser mais rápida que de um especialista humano.
- g) Respostas balanceadas, não emocionais e completas a qualquer momento: Destacada como uma importante característica, é necessário ressaltar que humanos podem não atender como esperado ou não desempenhar sua máxima eficiência em situações de estresse ou fadiga. Entretanto, sistemas especialistas não são afetados pelo uso extremo ou situações de pressão.
- h) Sistema inteligente: Sistemas especialistas acessam bancos de dados de forma inteligente e possibilitam que um usuário “não especialista” simule diversos cenários, explicando o motivo e como o resultado foi obtido. Assim, auxiliando no aprendizado do usuário.

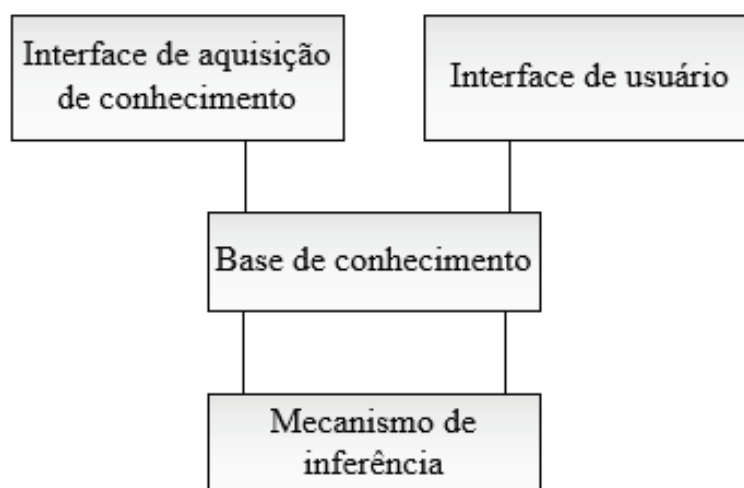
Sistemas especialistas, em sua implementação, podem auxiliar em situações complexas e ensinar “não especialistas” sobre determinado assunto, através de conselhos em momentos propícios. Ou seja, prover ao usuário uma segunda opinião quanto à forma que o processo está sendo executado, além de explicar os motivos e etapas que demonstram como essa segunda opinião foi formada (GRAY, 1994).

### 3.3.1 Arquitetura de um Sistema Especialista

Um sistema especialista, usualmente, é composto por quatro elementos básicos. Cada elemento tem sua função para que o sistema funcione de forma correta, sendo que cada elemento é indispensável para o bom funcionamento de um sistema especialista.

Conforme demonstrado pela Figura 12, tais elementos são: a base de conhecimento, mecanismo de inferência, interface de aquisição de conhecimento e interface com o usuário.

**Figura 12 - Elementos de um sistema especialista**



Fonte: elaborado pelo autor (2014).

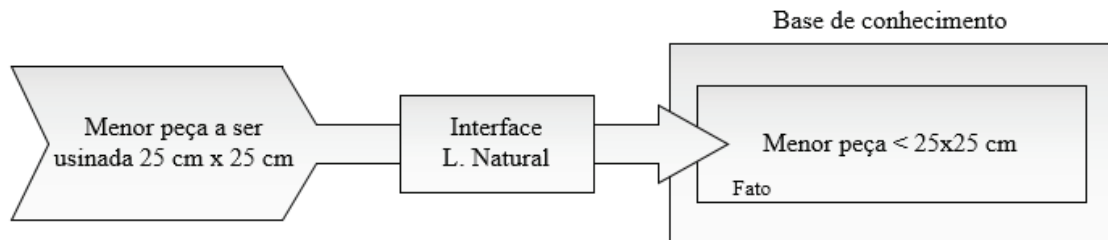
### 3.3.1.1 Base de Conhecimento

Pode-se considerar a base de conhecimento como o elemento mais importante de um sistema especialista, uma vez que ela dá as características de funcionamento do sistema. A base de dados é especializada e projetada para receber informações de determinada área de conhecimento, também chamada de domínio, tornando-a assim perita nessa área em particular.

Define-se a base de conhecimento como o elemento que possui conhecimento específico sobre um domínio real, usualmente o problema sendo abordado pelo sistema. Destaca-se ainda que a base de conhecimento difere de um banco de dados comum, dado o fato de que o conhecimento inserido na base de conhecimento está representada apenas implicitamente (BENFER et al., 1991).

A base de conhecimento é o local onde podemos armazenar fatos e regras que serão utilizadas pelo sistema (RIBEIRO, 1987). Um fato pode ser deduzido de uma afirmação inserida pelo usuário ou obtida de um sistema automático, como por exemplo, afirmando que: “a menor peça a ser usinada no processo atual possui 25 centímetros por 25 centímetros”, é fato que a menor peça possui 25 centímetros por 25 centímetros. Ou ainda, afirmando que “apenas uma ferramenta de 10 milímetros é utilizada no processo de usinagem”, é fato que apenas uma ferramenta é utilizada no processo. Essas informações são transformadas em fatos de forma que auxiliem no processo de inferência e no conselho gerado pelo sistema. A Figura 13 representa as afirmações e dados conhecidos do processo sendo transformados em fatos.

Figura 13 - Entrada sendo transformada em fatos



Fonte: elaborado pelo autor (2014).

De forma análoga, usuários com um nível superior de conhecimento podem inserir regras que modificam completamente o processo de inferência. Por exemplo, afirma-se que: “caso alguma peça possua menos de 26 centímetros de largura e possa ser inserida dentro de outra, então existe 70% de chance de se reduzir o custo do processo”.

De acordo com Nikolopoulos (1997), uma das formas mais comuns de se representar regras é através de estruturas “se-então” (*if-then*). Essas regras possuem o seguinte formato:

- Se (condições a ser respeitada);  
Então (realizar ação ou chegar a uma nova conclusão). Ou seja, quando aplicado ao exemplo acima teremos:
- Se (Peça < 26 cm e cabe dentro de outra);  
Então (0,7% de chance de reduzir custo do processo).

Logo, o sistema usa as informações de entrada como regra no processo de inferência.

### 3.3.1.2 Mecanismo de Inferência

Ribeiro (1987) define o mecanismo de inferência como o elemento de um sistema especialista capaz de buscar as regras que devem ser avaliadas, organizar de forma lógica e então realizar o processo de inferência. O funcionamento desse mecanismo está ligado diretamente à base de conhecimento, uma vez que as inferências são feitas de acordo com a técnica utilizada para o armazenamento do conhecimento. Embora existam muitos métodos para realizar inferência, o mais usado é o da avaliação de regras.

O mecanismo utiliza as regras de inferência de forma que o conhecimento explícito na base de conhecimento seja utilizado para inferir novas conclusões que não necessariamente estão implícitas na base de dados, mas podem ser inferidas através da junção do conhecimento já obtido. O mecanismo de inferência não é limitado apenas ao raciocínio matemático, ou ao raciocínio formal, ele é compatível a ambos (BENFER et al., 1991).

### 3.3.1.3 Interface de Aquisição de Conhecimento

A função principal da interface de aquisição de conhecimento é auxiliar na transformação do conhecimento em informações que possam ser, de fato, interpretadas e armazenadas pela base de conhecimento. De acordo com Benfer, Brent e Furbee (1991), usualmente os recursos utilizados para auxiliar na aquisição de conhecimento são:

- Traços: Listas de regras seguindo a ordem em que foram iniciadas ou executadas.
- Pesquisa: Comandos utilizados para encontrar e modificar regras específicas, fatos ou outros recursos disponíveis para inspeção no sistema.
- Escrituração: Através da escrituração das funções e índices utilizados, cria-se uma maneira de rastrear os recursos da base de conhecimento como variáveis ou regras.

O mecanismo de aquisição de conhecimento está ligado diretamente à forma como os dados são inseridos pelo usuário, uma vez que ele compartilha o seu conhecimento com o sistema. Além disso, a forma como esse conhecimento é compartilhado deve ser lida pelo sistema e transformada em recursos que possam ser identificados pelo mesmo.

O reconhecimento dos recursos de aquisição de dados, conforme listado acima, é indubitavelmente de extrema importância para a construção da justificativa para as ações tomadas. Através da união e do reconhecimento das regras, fatos e demais recursos que foram utilizados para obter a resposta, pode-se justificar e explicar o motivo da decisão tomada. As conclusões e explicações são apresentadas através da interface com o usuário.

Diferentes de métodos tais como, redes neurais que funcionam como uma “caixa preta” (embora a resposta seja obtida as razões ou passos utilizados para chegar a essa resposta são desconhecidos), sistemas especialistas possuem os recursos necessários para explicar a sugestão feita. O fato de que um sistema especialista possa justificar seu raciocínio e das decisões tomadas, é um fator de suma importância para a implementação bem sucedida do mesmo (NIKOLOPOULOS, 1997).

### 3.3.1.4 Interface com o Usuário

A interface com o usuário é um membro importante e responsável por toda e qualquer comunicação do usuário com o sistema especialista. A interface auxilia nas consultas feitas pelo usuário e deve possibilitar que o usuário insira dados, assim como deve prover as informações necessárias que expliquem a resolução do problema em análise, esclarecendo detalhadamente as ações realizadas e o raciocínio utilizado para chegar em um determinado resultado.

Em sua grande maioria, a comunicação com o usuário é feita através de uma interface gráfica, também conhecida como GUI do inglês *Graphical User Interface*, uma vez que esse tipo de interface permite a interação com o usuário através de componentes como caixas de texto, ícones, imagens, controles através de mouse, entre outros. Além de apresentar resultados e explicar as ações tomadas, a interface com o usuário pode permitir a entrada de dados através de recursos externos, tais como planilhas eletrônicas ou bancos de dados (NIKOLOPOULOS, 1997).

Para Benfer, Brent e Furbee (1991), essa interface deve, na maioria dos casos, simular e prover ao usuário uma experiência similar a se o mesmo estivesse questionando ou interagindo com um especialista humano. Dada a ordem de como as regras e suas possíveis ações são estruturadas em um sistema especialista, quando uma pergunta é realizada, é possível deduzir uma resposta através de diálogos autoexplicativos criados pelo desenvolvedor do sistema especialista. É também possível mostrar ao usuário o motivo de determinada regra ou ação ter sido tomada, podendo utilizar esse motivo como a justificativa para uma determinada tomada de decisão. A Tabela 4 demonstra a forma como as regras são estruturadas.

**Tabela 4 - Estrutura utilizada para armazenar regras**

| <b>Regra 1 - Método de <i>Nesting</i> 1</b>                                                                                                                                                             |                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| SE                                                                                                                                                                                                      | (90% das peças possuem formato retangular) |
| E                                                                                                                                                                                                       | (Não podem ser rotacionada)                |
| Então: Utilizar método de <i>nesting</i> retangular<br>Se não: Regra 2<br>Por que: Maioria das peças possui formato retangular, por esse motivo o método de <i>nesting</i> retangular é o mais indicado |                                            |
| <b>Regra 2 - Método de <i>Nesting</i> 2</b>                                                                                                                                                             |                                            |
| SE                                                                                                                                                                                                      | (70% das peças possuem formato livre)      |
| E                                                                                                                                                                                                       | (Podem ser rotacionada)                    |
| Então: Utilizar método <i>True Shape</i><br>Se não: Regra 3<br>Por que: Regra 1 não foi utilizada. Maioria das peças possui formato livre, portanto o método <i>True Shape</i> foi escolhido            |                                            |

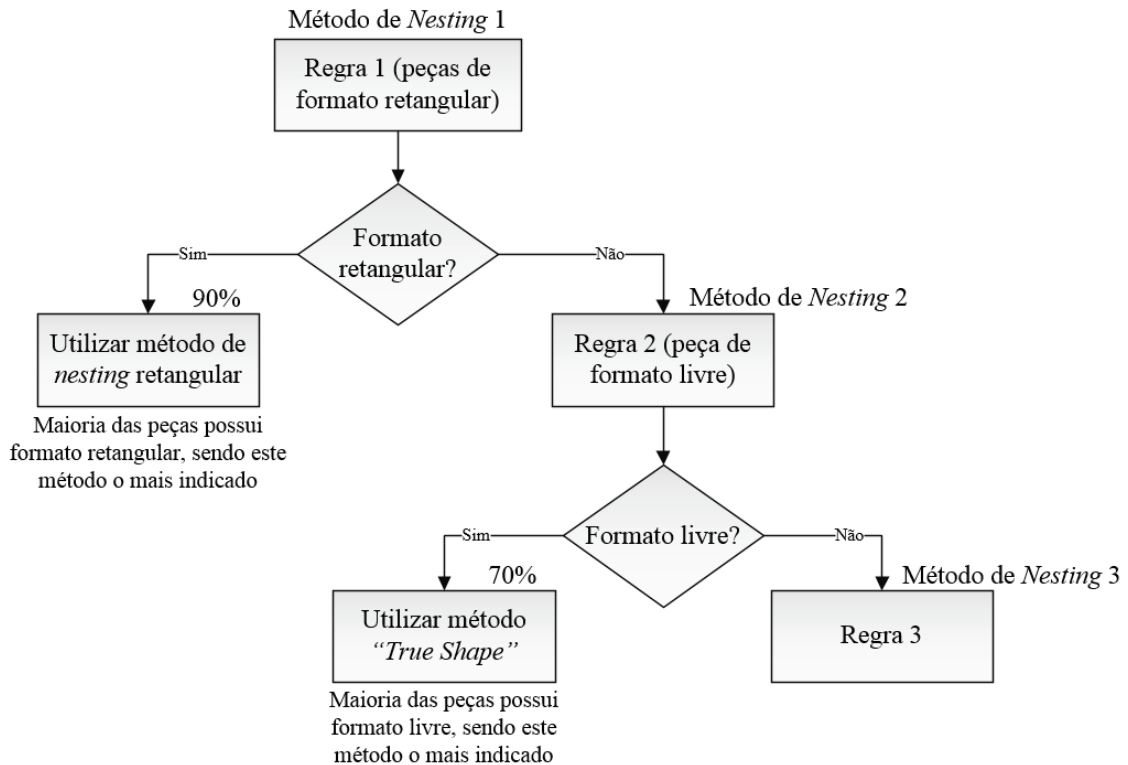
Fonte: elaborado pelo autor (2014).

Todavia, Benfer, Brent e Furbee (1991) reforçam que a respostas e sugestões apresentadas não necessariamente precisam ser realizadas através de texto. Para isso, é possível fazer uso de diversas estruturas, como: imagens, fluxogramas funcionais, grafos,

listas de comandos, entre outros recursos. A Figura 14 apresenta uma possível forma de apresentar ao usuário os passos utilizados na tomada de decisão.

O tipo de resposta trazido pelo sistema pode variar drasticamente de acordo com as especificações do sistema. Um exemplo é a situação onde o usuário deseja entender como a solução foi obtida e deseja verificar os passos adotados pelo programa. Para um maior entendimento, a interface com o usuário pode responder essa solicitação com fluxogramas funcionais que ilustrarão as ações tomadas de forma mais clara que utilizando texto.

**Figura 14 - Fluxograma demonstrando a escolha com base nas regras**



Fonte: elaborado pelo autor (2014).

### 3.4 Pesquisa Operacional

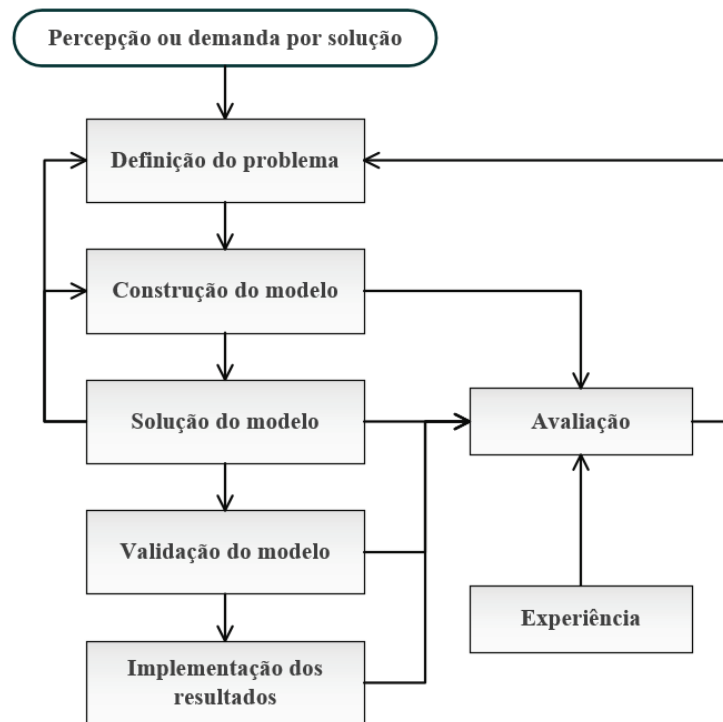
O ato de otimizar está ligado diretamente à área da Pesquisa Operacional. Andrade (1998) faz uma abordagem histórica da Pesquisa Operacional e afirma que esta historicamente teve início na Inglaterra, durante a Segunda Guerra Mundial. Pesquisadores desenvolveram métodos visando criar ferramentas de análise de decisão ligadas diretamente às decisões da guerra; tendo como foco principal o auxílio na alocação de radares antiaéreos e, posteriormente, sendo utilizado e aperfeiçoado no ramo acadêmico e empresarial.

A aplicação de Pesquisa Operacional está ligada tanto à construção de modelos matemáticos, econômicos e estatísticos de problemas de decisão e controle com o objetivo de resolver situações complexas e de incerteza, para análise de relações determinantes de consequências futuras de ações alternativas, quanto para planejar a eficácia das mesmas (WAGNER, 1986).

Segundo Silva et al. (1998), a Pesquisa Operacional (PO), na organização de um sistema através de um modelo, tem como objetivo descobrir a melhor maneira de operá-lo e auxiliar a tomada de decisões. Atualmente, utiliza-se Pesquisa Operacional em diversas áreas na resolução de problemas complexos e como auxílio na tomada de decisão do dia a dia empresarial.

A modelagem de um problema engloba suas diversas alternativas, restrições e um critério objetivo, a fim de representar o problema em variáveis de decisão e obter a solução ótima ou próxima, satisfazendo as devidas restrições e otimizando os valores da função objetivo (ou critério objetivo). Os principais passos para a construção desses modelos são descritos posteriormente (TAHA, 1997) e ilustrados conforme Figura 15:

**Figura 15 - Fases de um estudo de Pesquisa Operacional**



Fonte: adaptação (TAHA, 1997).

A definição do problema é o processo onde o escopo do problema a ser investigado é definido através da descrição de decisões alternativas, a determinação do objetivo e determinação dos fatores limitantes. A construção do modelo traduz, muitas vezes matematicamente, as definições atribuídas ao problema; de forma a possibilitar a aplicação de um algoritmo padrão para obter sua solução ou atribuir o uso de técnicas heurísticas, caso os algoritmos disponíveis não sejam compatíveis com o modelo puramente matemático.

A solução do modelo implica no uso de algoritmos de otimização já definidos. Porém, é necessário analisar a sensibilidade do modelo em relação aos parâmetros do problema, e verificar o comportamento do resultado ótimo. A validação do modelo, por sua vez, visa verificar se o modelo proposto cumpre com seu propósito, oferecendo assim a predição do comportamento do sistema modelado.

Um método comum de validação é a comparação de dados históricos, validando assim o modelo caso o mesmo seja capaz de reproduzir o desempenho histórico analisado sob as mesmas condições. A implementação pode ser classificada como a etapa onde a solução do



modelo validado é então traduzida pelo responsável no uso de PO, de forma a ser compreendida pelos indivíduos que administrarão o sistema modelado.

### 3.4.1 Otimização

Modelos de otimização constituem-se de problemas onde é necessário maximizar ou minimizar funções com uma ou diversas variáveis de uma região multidimensional. Algoritmos utilizados para a resolução de problemas de otimização podem ser classificados como determinísticos ou probabilísticos. Seguindo as afirmações de Harrel et. al. (2003), otimização pode ser definida como o processo de realizar distintas combinações de valores em variáveis de controle (independentes) em busca do resultado ótimo desejado.

Um problema de otimização consiste em encontrar a melhor solução, conhecida por solução ótima, ou conjunto de parâmetros que vise atingir um objetivo pré-definido. Para Papadimitriou (1998), o problema de otimização divide-se em duas categorias: problemas com variáveis contínuas e problemas com variáveis discretas, que também são chamados de problemas combinatoriais. Os problemas combinatoriais consistem em encontrar um objeto dentro de um espaço, finito ou infinito, contável de possibilidades. O objeto procurado pode ser um inteiro, um conjunto, uma permutação ou um grafo.

Conforme Goldberg e Luna (2005, p. 2), os modelos são representações simplificadas da realidade que preservam, para determinadas situações e enfoques, uma equivalência adequada com a mesma.

### 3.4.2 Metaheurísticas

Devido ao fato que problemas de otimização combinatória, em sua grande maioria, são NP-Difícil; sua respectiva complexidade e dificuldade em identificar um valor ótimo faz necessário o uso de heurísticas na busca de produzir um resultado de boa qualidade (HERTZ; WIDMER, 2003).

Uma metaheurística utiliza a combinação de heurísticas subordinadas dentro de um espaço definido de busca, e tem como objetivo encontrar a melhor solução possível do problema analisado. Sendo um método aproximativo, não existe a garantia de encontrar o resultado ótimo, porém devido à sua adaptabilidade e sua capacidade de interação com os resultados anteriores, possibilita soluções viáveis cada vez melhores dependendo do tempo hábil para o cálculo.

Metaheurísticas possuem combinações de escolhas aleatórias e fazem uso do conhecimento histórico de resultados adquiridos anteriormente pelo processo de busca, a fim de guiar e realizar buscas em vizinhanças dentro do espaço de pesquisa. Tais características são definidas como “Diversificação” (foco na expansão da busca) e “Intensificação” (foco no histórico) evitando assim manter-se com um resultado prematuro de um ótimo local, porém mantendo a coerência obtida através do histórico da pesquisa.

Segundo Blum e Roli (2003), uma metaheurística pode ser definida como uma estratégia que guia o processo de busca, sendo um algoritmo que pode apresentar comportamentos distintos; dito não determinístico. Conforme Boussaid et al. (2013), existem tipos distintos desses algoritmos, sendo estes:

- **Com base em uma única solução:** também chamadas de métodos de trajetória, este método se inicia em uma solução única inicial e conforme a necessidade de obter o resultado, afastando-se da mesma dentro do espaço de busca. Neste caso, os seguintes métodos compõem este grupo: Busca Tabu, *Simulated Annealing*, *Iterated Local Search*, *Variable Neighborhood Search*, método GRASP, etc.
- **Com base populacional:** métodos que utilizam um conjunto de soluções onde ocorre interação entre as mesmas, através de operadores de mutação ou evolutivos. Estes métodos são relacionados à Computação Evolutiva (teoria de evolução de Darwin) e *Swarm Intelligence* (comportamento coletivo), dentre eles: Estratégia Evolutiva, Algoritmos Genéticos, *Scatter Search*, Otimização por Colônia de Formigas, etc.

Além dos tipos de algoritmos citados acima, existem metaheurísticas híbridas e paralelas. O tipo híbrido pode combinar a metaheurística com algoritmos de programação matemática, programação de restrição, e aprendizagem automática. Já o tipo paralelo usa programação paralela para executar várias múltiplas pesquisas; com o objetivo de melhorar o ótimo global (TALBI, 2009).

### 3.4.2.1 Busca Tabu

O termo Busca Tabu (*Tabu Search*) foi criado sob as mesmas circunstâncias em que o termo metaheurística foi introduzido (GLOVER, 1986). A Busca Tabu tem como objetivo incorporar a memória adaptativa e a exploração responsiva na resolução de problemas, de forma inteligente.

O recurso de memória adaptativa deste algoritmo permite implementar procedimentos de busca de forma econômica e eficaz dentro do espaço da busca de soluções. Devido ao fato que a Busca Tabu usa opções locais guiadas por informações obtidas durante a busca, este método se diferencia de algoritmos sem memória que utilizam processos aleatórios que implementam uma forma de amostragem.

Além disso, a exploração responsiva da Busca Tabu (seja em uma execução determinística ou probabilística) acredita na suposição de que a escolha de uma estratégia ruim pode muitas vezes produzir mais informação que uma boa escolha aleatória (GLOVER; LAGUNA, 1998).

O uso do algoritmo Busca Tabu é um método iterativo para encontrar, em um conjunto  $X$  de possíveis soluções viáveis, uma solução  $s$  que otimiza a função objetivo  $f$ . A formulação (1) abaixo demonstra a função objetivo descrita:

$$\text{Minimizar } f(s): s \in X \text{ e } X \in R_v, \text{ sendo } R_v \text{ a região viável} \quad (1)$$

Neste caso, uma vizinhança  $N(s)$  é definida para cada solução  $s$  em  $X$ . O procedimento começa a partir de uma solução inicial viável ( $S_{\text{inicial}}$ ). Sempre que uma solução viável  $s$  for encontrada, uma coleção  $V^*$  de soluções  $s_i$  em  $N(s)$  é gerada, e um movimento feito para a melhor  $s_i$  em  $N(s)$ . Esta vizinhança melhor é denotada como  $s^*$  (HERTZ, 1991). Ou seja, a Busca Tabu encontra soluções ótimas locais e a partir destes e cria novas vizinhanças para melhorar o resultado encontrado. Para ilustrar o método, o algoritmo geral da Busca Tabu é mostrado na Figura 16 (RODRIGUES; GÓMEZ, 2000).

Figura 16 - Pseudocódigo da Busca Tabu

```

Inicialização
s := solução inicial em X;
Nbiter := 0;
Melhiter := 0;
Melhsol := s;
T := Ø;
Inicializar Função Critério de Aspiração A;
Enquanto (f(s) > f*) e (Nbiter - Melhiter < Nbmax) faça
    Nbiter := Nbiter+1;
    Gerar um conjunto V* de soluções Si em N(s) o qual não seja tabu ou que
        A(f(s)) >= f(Si);
    Escolha uma solução s* minimizando f de V*;
    Atualize a Função Critério de Aspiração A e a Lista tabu T;
    Se f(s*) < f(Melhsol) então
        Melhsol := s*;
        Melhiter := Nbiter;
    Fim_se;
    s := s*;
Fim_enquanto;

```

Sendo:  
 Nbiter: número da interação atual;  
 Melhiter: número da interação da melhor solução encontrada; \\  
 Melhsol: melhor solução;  
 X: conjunto de soluções possíveis;  
 f: função objetivo;  
 N(s): a vizinhança da solução s em X;  
 |T|: Lista Tabu;  
 |V|: número de soluções vizinhas geradas em cada interação;  
 f\*: o mais baixo limite de da função objetivo f;  
 A: Função critério de Aspiração;

Fonte: elaborado pelo autor (2014).

Para evitar percorrendo um caminho mais de uma vez e evitar que a mesma vizinhança seja visitada mais de uma vez, as informações sobre os movimentos realizados recentemente são armazenadas em uma ou mais listas tabu. Na sua forma mais simples, a construção da lista tabu registra os movimentos mais recentes realizados. Movimentos podem ser subdivididos em tipos e diferenciados por direção. Uma lista tabu pode ser separada e mantida para cada uma destas classificações, seja tipo ou direção (GLOVER, 1986).

Os movimentos são armazenados nessa lista e permanecem proibidos (movimentos tabu) por um dado número de iterações, chamado *tabu tenure*. Assim, quando explorando um subconjunto pertencente a atual vizinhança, ficam excluídos da busca vizinhos previamente realizados e que fazem parte da Lista Tabu.

No entanto, embora a lista tabu auxilie na não repetição de movimentos a mesma também pode proibir movimentos que levariam a melhores soluções. Por esse motivo um mecanismo adicional é utilizado, esse mecanismo é chamado de critério de aspiração. Esse mecanismo é responsável por remover itens da Lista Tabu quando o movimento respeita certas circunstâncias. Embora essas circunstâncias possam variar, o critério de aspiração mais comum é o que permite que um movimento tabu seja realizado quando e somente se esse movimento conduzir a um vizinho melhor do que o melhor vizinho já encontrado.

Desde a formulação inicial da Busca Tabu, diversas variações deste método foram geradas obtidas para alcançar resultados cada vez mais próximos do ótimo e com menor recurso computacional possível. Como exemplo, podemos citar alguns estudos da área:

- Gendreau et al. (1998) definem o Problema do Caixeiro Viajante (PCV) como um grafo (lucro) e arestas (distâncias); usando a Busca Tabu para inserir lotes (*clusters*) de vértices na rodada atual ou remover uma cadeia de vértices. Testes realizados em instâncias geradas aleatoriamente com até 300 vértices mostram que o algoritmo de forma consistente produz soluções quase ideais. Através de testes realizados em instâncias geradas aleatoriamente (até 300 vértices), é possível verificar que o algoritmo produz soluções perto do ideal de forma consistente e; provando sua capacidade de aplicação em diferentes problemáticas.

Além disso, outro recurso que recebem crescente enfoque em trabalhos acadêmicos é o processamento paralelo aplicado a esses métodos de Fiechter (1994) e He et al. (2005). Em ambos os casos, se faz uso da Busca Tabu com a estratégia paralela; permitindo o processamento do algoritmo de modo mais eficaz. Nesses casos, múltiplas tarefas podem ser executadas ao mesmo tempo por diferentes processadores.

### 3.5 Análise das Contribuições Encontradas na Literatura

Nesta seção são apresentadas as principais contribuições feitas por outros estudos e que de forma direta ou indireta contribuem para a formalização da ideia e implementação do modelo proposto por esse trabalho.

#### 3.5.1 Sistema de Apoio à Tomada de Decisão

Song (1999) visa desenvolver em seu artigo, um sistema de apoio à tomada de decisão inteligente focado em aprimorar a economia de recursos e reciclagem consciente em atividades ligadas à manufatura. O artigo visa minimizar os impactos causados pela manufatura de produtos, reduzindo o consumo de energia através do ciclo de vida do produto. O foco principal do SAD proposto é tornar o produto mais rentável, tendo seu custo benefício aprimorado. Para isso, faz uso de um sistema de apoio baseado em conhecimento que auxilia na escolha da melhor matéria prima e etapas de manufatura para serem utilizados no processo.

Zhou et al. (2008) introduz novos tipos de sistemas de apoio à decisão inteligentes tais como: SAD de grupo, SAD distribuído, SAD inteligente, SAD interativo e integrado. Ao longo do estudo, além da apresentação dos conceitos básicos e principais características sobre os sistemas introduzidos, também são abordadas as principais técnicas utilizadas por estes sistemas. A maior contribuição do artigo vem da discussão sobre os métodos, abordando o uso do conhecimento no SAD.

Bana e Costa et al. (1999) apresentaram uma proposta de solução para uma problemática real envolvendo a análise de decisão multicriterial na indústria têxtil, onde diversos sistemas de apoio à tomada de decisão foram integrados no processo de aprendizagem para a resolução do problema. Através do artigo é mostrado como a união de todos os SAD resultou em uma ferramenta poderosa e bem recebida pelos usuários.

Nepomuceno e Fontana (2013) apresentam um sistema híbrido de apoio à tomada de decisão é proposto com foco na área de gerenciamento de projetos de *software*. O método

proposto utiliza programação linear para auxiliar na decisão de quais recursos serão utilizados para finalizar a execução do projeto de *software* visando minimizar o tempo máximo de execução do mesmo.

Vargas (2012) apresenta um sistema de apoio à decisão fazendo uso de métodos ligados a Inteligência Artificial é proposto. O artigo aborda um problema relacionado à falta de qualidade na manufatura de produtos que possuem um baixo volume de produção e, através do artigo, busca tornar o processo viável, tornando-o flexível e ao mesmo tempo garantindo a qualidade do produto final. Para isso, o artigo faz uso de um sistema de apoio à decisão utilizando um sistema especialista, acoplado às técnicas de Sistema Multiagente responsáveis pelo diagnóstico e pela tomada de decisão. Em suas conclusões, o artigo reforça que a Inteligência Artificial no ramo industrial é um método válido que proporciona dinamismo e flexibilidade à linha de produção.

Amoako-Gyampah e Acquah (2008) analisam as relações entre as diferentes estratégias adotadas em empresas quanto ao seu processo de manufatura. A relação principal analisada pelo artigo está no relacionamento entre estratégias de manufatura e estratégias competitivas e suas influências nos resultados obtidos pela empresa. O artigo conclui que muitas estratégias adotadas não afetam efetivamente os resultados obtidos nas empresas, e aponta que o foco na qualidade da produção é uma das poucas estratégias que afeta consideravelmente o resultado final.

### 3.5.2 Sistema Especialista e Sistemas Baseados em Regras

Pereira et al. (2013) apresentam uma abordagem que permite especificar e realizar a detecção de situações onde é necessário o aprimoramento de sistemas que necessitem obter a percepção e o conhecimento sobre o que acontece nas suas imediações (*Situation-Aware Applications*). Para chegar aos resultados esperados, o artigo propõe o uso de um sistema utilizando regras como base para a ferramenta de gestão. Para a implementação e aprimoramento das regras já existentes, o *software* Jboss Drools® é utilizado.

Costa et al. (2012) são propostos métodos utilizando uma abordagem orientada a modelos para a especificação e detecção de situações. Dois métodos são propostos pelo trabalho, no primeiro sendo empregada uma linguagem artificial e gráfica para a modelagem de situações. Já no segundo método, essas linguagens são transformadas em um sistema baseado em regras. O artigo faz uso do *software* Drools® para a implementação das regras e exemplifica tais regras com base em um exemplo de fraude bancária.

Azevedo (2005) apresenta um sistema especialista de apoio à tomada de decisão é desenvolvido para auxiliar no gerenciamento de pessoas desempenhando as funções de um perito trabalhista. Para a implementação do sistema especialista, a *shell* expert Sinta é utilizada como ferramenta responsável na administração das regras utilizadas no sistema. Embora o trabalho não esteja ligado diretamente à área industrial, a implementação é detalhada e todas as regras utilizadas são expostas através dos anexos do trabalho.

### 3.5.3 Sobras Aproveitáveis e seu Reuso

Abrantes (2013) apresenta duas variações do problema de empacotamento são estudadas. O estudo é realizado de forma a satisfazer uma demanda de itens, minimizando o

custo dos objetos utilizados, e também buscando a resposta que maximize sobras aproveitáveis. Além disso, de forma contrária à primeira proposta, outra variação do problema de empacotamento é estudada, sendo que essa visa à minimização da quantidade de sobras. O estudo consegue resolver problemas de pequeno porte, uma vez que propõe soluções através de programação linear inteira e métodos baseados em *branch-&-bound*.

Cui e Yang (2010) apresentam uma heurística para resolução do problema de corte unidimensional com sobras aproveitáveis é proposta. O método proposto possui duas etapas: na primeira a programação linear é utilizada para satisfazer, o máximo possível, uma demanda de itens; já a segunda etapa utiliza uma heurística sequencial que preenche a parte restante do item de demanda. O método proposto ajuda no balanceamento da matéria prima consumida e o aumenta lucro a partir das sobras.

Cherri e Vianna (2013) apresentam o problema de corte bidimensional é abordado levando em consideração que uma demanda de itens gera sobras indesejáveis. Tais sobras, que de acordo com o trabalho não são grandes o suficiente para serem aproveitadas e nem tão pequenas para serem perdas aceitáveis, geram retalhos que podem ser utilizados para a produção de itens de demandas futuras. Dessa forma, chegam a resultados que não atendem somente ao critério de minimização do uso de matéria prima, mas também soluções que levem em consideração a sucata gerada. Para tanto, são empregados métodos envolvendo alterações na abordagem Grafo e/ou procedimentos heurísticos clássicos propostos pela literatura.

Ferrary et al. (2013) apresentam uma ferramenta computacional que visa possibilitar o reuso das sobras aproveitáveis geradas pelo processo de *nesting*, especialmente em *software* CAM. Para tanto, são empregados métodos distintos visando encontrar o perfil que representa a sucata (sobras) deixada pelo processo após a operação de *nesting*. No estudo são propostos dois métodos para a obtenção dos resultados, o primeiro um método envolvendo menor processamento e gerando resultados aproximados; uma vez que analisa os itens através de retângulos; e outro com alto nível de processamento, porém com resultados precisos, pois analisa o perfil real das peças.

### 3.5.4 Problema de Corte (*nesting*)

Carvalho e Rodrigues (1994) apresentam um sistema interativo computacional é desenvolvido para uma empresa fabricante de aço; sendo que a mesma possui processos de produção intermediários e não apenas o corte do aço. O sistema busca, além da minimização de perdas nos cortes e redução do tempo de *setup*, a redução dos cortes intermediários nas bobinas de aço; etapa necessária no sistema produtivo devido ao método manual de planejamento dos cortes. No artigo, o sistema proposto usa Programação Linear e uma heurística complementando com um procedimento de arredondamento para discretizar as soluções. O sistema é usado para obter múltiplas opções de planos de corte operacionais, respeitando os diferentes tipos de matéria prima. Os resultados obtidos, em relação à redução de perdas no corte, são ligeiramente melhores do que através do planejamento de corte manual. Porém, como o sistema é capaz de fornecer todas as informações necessárias para avaliar as diferentes alternativas de programação de produção; nesse aspecto, os resultados tornam evidente a capacidade do sistema em definir a ordem de cortes em menos tempo que o time de planejamento é capaz (programadores de produção experientes).

Outro exemplo é o estudo de Júnior et al. (2013), onde uma metodologia híbrida é proposta; unindo um Algoritmo Genético e uma heurística *Bottom-Left Greddy* para resolver

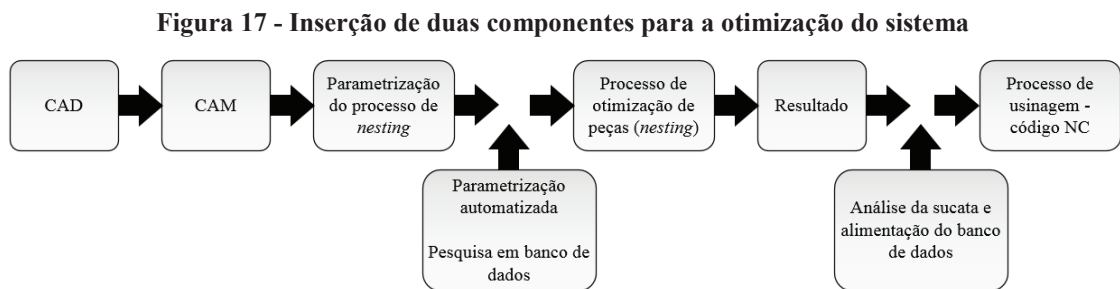
o problema de *Nesting* bidimensional em uma empresa do ramo têxtil. A ferramenta *No-Fit Polygon* é usada para obter ótimos locais em conjunto com um algoritmo de encolhimento, incorporado à metaheurística para identificar soluções de melhor qualidade. Os resultados obtidos através do uso do método híbrido proposto são comparados a demais estudos da literatura, onde a abordagem utilizada torna-se promissora ao ser testada como estudo de caso de uma empresa fabricante de logotipos de times de futebol, aplicando de modo prático a metodologia estudada.

## 4 METODOLOGIA E MODELO CONCEITUAL

### 4.1 Visão Geral do Sistema Proposto

Visando auxiliar a decisão na etapa de parametrização e buscar o resultado ótimo com base em um critério de otimização, a proposta é analisar todas as possíveis combinações de parâmetros e chapas que podem ser utilizadas ao longo do processo de manufatura por completo. Para tanto, é necessária a adição de dois componentes ao fluxo de manufatura original (mostrado na Figura 1). O primeiro componente a ser adicionado está relacionado diretamente à parametrização do processo, uma vez que a mesma é um procedimento realizado de forma manual na grande maioria dos *softwares* de *nesting*. Conforme apresentado pela Tabela 2, *softwares* tendem a carregar uma quantidade significativa de parâmetros que podem ser utilizados, exigindo do usuário uma determinada *expertise* sobre o processo de manufatura e o *software* utilizado. Para suprir essa necessidade, o primeiro componente é um sistema especialista (detalhado ao longo da seção 4.5.1) que deve guiar o usuário na escolha dos parâmetros. Todavia, a configuração do processo também envolve a escolha da matéria prima, sendo assim, a implementação de um banco de dados contendo as chapas disponíveis para serem utilizadas no processo é proposta.

O segundo componente a ser adicionado ao processo está ligado ao pós-processamento do resultado obtido. Uma vez que alguns projetos não preenchem totalmente a chapa e geram sobras que muitas vezes podem ser aproveitáveis, tal componente visa reconhecer essas sobras. O reconhecimento das mesmas possibilita caracterizá-las como aproveitáveis ou não aproveitáveis e, quando aproveitáveis, armazená-las no banco de dados de forma que seja possível sua reutilização futuramente. O novo ciclo de manufatura proposto é apresentado na Figura 17.



Fonte: elaborado pelo autor (2014).

A integração de todos os componentes propostos forma um sistema híbrido, uma vez que utiliza métodos de diversas áreas do conhecimento para chegar a um resultado que, quando unificado, possibilita a obtenção de resultados atrativos no âmbito industrial. O resultado e a interação com o usuário acontecem através do sistema híbrido de apoio a decisão.

### 4.2 Visão Geral do SAD Híbrido Proposto

Dada a complexidade e a variedade do processo como um todo, métodos individuais dificilmente chegariam a um resultado ótimo. Por esse motivo, o método híbrido proposto



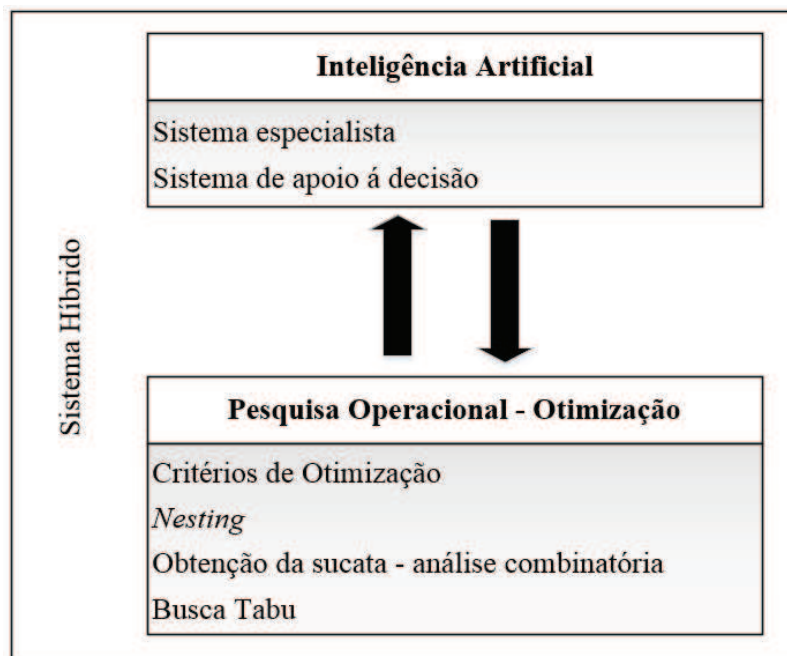
utiliza diversas áreas de conhecimento e, de acordo com cada etapa, aborda o problema utilizando um método indicado para a obtenção de um bom resultado.

Além disso, o uso de um sistema híbrido se faz necessário pela inviabilidade do uso dos métodos clássicos da Pesquisa Operacional no problema abordado. Tal inviabilidade acontece devido ao tempo de processamento que seria necessário para a obtenção de uma resposta ótima (ou quase ótima) caso metaheurísticas fossem utilizadas. Isso ocorre devido ao funcionamento das metaheurísticas, que buscam evolução da função objetivo através de diversas tentativas e possíveis permutações de parâmetros. Todavia, o sistema de *nesting* demanda considerável processamento e tempo para que seus cálculos sejam finalizados e por esse motivo, diversas tentativas utilizando diversos parâmetros resultariam em um tempo de resposta inviável.

O sistema híbrido proposto pode ser separado em dois blocos: os métodos ligados à área de Inteligência Artificial e o uso do conhecimento e os métodos ligados à Pesquisa Operacional, ou métodos de otimização. Nas etapas onde é necessário o uso da Inteligência Artificial para suprir a necessidade de um especialista, um sistema especialista é usado em conjunto com os componentes trazidos por um sistema de apoio à decisão.

Por outro lado, quando é necessário maximizar ou minimizar recursos, são utilizados critérios de otimização, modelados matematicamente, de forma que possibilitem chegar a um resultado ótimo ou próximo do ótimo com base no critério escolhido. Ainda analisando a área da Pesquisa Operacional ligada ao sistema híbrido proposto, as problemáticas voltadas à otimização utilizando *nesting* e à obtenção do perfil, que representa as sobras deixadas pelo processo de usinagem, podem ser classificadas como problema de otimização combinatória. Os componentes que compõem o sistema híbrido utilizado são ilustrados através da Figura 18.

**Figura 18 - Componentes do sistema híbrido proposto**



Fonte: elaborado pelo autor (2014).

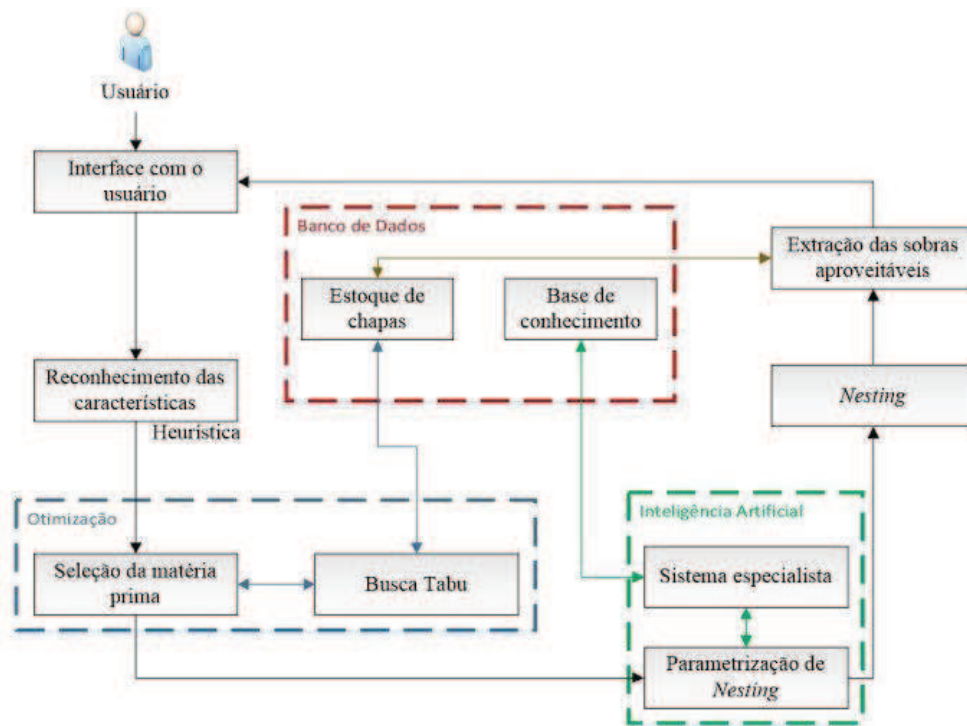
A comunicação entre os módulos deve ocorrer para que todos os componentes possam trabalhar de forma homogênea, utilizando o mesmo fluxo de dados. O sistema especialista é o principal responsável pela comunicação entre os componentes, uma vez que o mesmo utiliza informações obtidas no início do processo (reconhecimento das características das peças),

meio do processo (parametrização) e fim do processo (extração de sobras e armazenamento em banco de dados).

Já a comunicação com o usuário é dinâmica, podendo o mesmo interagir de forma intensa com o sistema ou, de forma oposta, ele pode aceitar todos os parâmetros sugeridos e reduzir ao mínimo sua interação. A interface com o usuário é responsável por apresentar, em cada etapa, as escolhas que estão sendo feitas e justificá-las quando solicitado, dando assim ao usuário um melhor entendimento da razão para aquela escolha.

A forma como cada etapa se comunica com as demais, assim como cada etapa ocorre pode ser evidenciada através da arquitetura do SAD proposto - apresentada na Figura 19.

Figura 19 - Arquitetura do SAD proposto



Fonte: elaborado pelo autor (2014).

Nos parágrafos a seguir, o comportamento dos componentes básicos de um sistema de apoio à tomada à decisão, e seus respectivos funcionamentos dentro do sistema proposto, são explicados:

- **Banco de dados:** O banco de dados é responsável por armazenar toda e qualquer informação utilizada pelas diversas etapas do processo de decisão. As informações contidas no banco de dados são variadas e representam desde informações físicas do processo; tais como chapas disponíveis em estoque ou até mesmo informações virtuais como o histórico das decisões tomadas, dados utilizados na parametrização do processo de *nesting*, entre outras informações possibilitando ao SAD extrair informações sobre cenários antes já simulados. Embora um banco de dados ligado a um SAD possa armazenar informações internas (relacionadas diretamente ao processo como, por exemplo, as chapas em estoque) e informações externas (informação que depende de terceiros, tais como preço das chapas utilizadas), na implementação inicial do SAD proposto, são ignoradas as informações externas, não considerando assim as variações de mercado.

- **Banco de modelos:** O banco de modelos é ligado diretamente ao sistema especialista, pois os modelos matemáticos propostos nesse trabalho são utilizados diretamente pelo sistema especialista para o processo de tomada de decisão. Os modelos propostos, demonstrados com mais detalhes na subseção 4.5.2., utilizam da modelagem matemática para atender um ou mais critérios de otimização estipulados pelo usuário. Neste caso, a responsabilidade é do sistema especialista em tomar as decisões corretas que deem ao usuário o resultado esperado com base nas suas necessidades. Por se tratar de um sistema híbrido, é importante que as ferramentas de apoio à decisão e de otimização trabalhem de forma unificada e, por esse motivo, os modelos de otimização contidos no banco de modelos dependem diretamente das regras criadas pelo sistema especialista e terão seu resultado baseado nas mesmas.
- **Interface:** Assim como os demais componentes do sistema de apoio a decisão proposto, a interface também é um componente integrado aos demais e em diversos momentos apresentará características específicas de etapas do processo. No início do processo, a interface com o usuário permite ao mesmo que sejam inseridas peças trazidas do *software* CAD e apresenta as características da peça. Em uma segunda etapa, a interface de usuário passa a ser integrada com a interface do sistema especialista e exibe as regras utilizadas, quais parâmetros específicos de *nesting* estão em uso, qual o material adotado entre outras características do processo. Em uma terceira etapa, são apresentados ao usuário os resultados obtidos e extraídas as sobras aproveitáveis do processo que, por conseguinte, são apresentadas ao usuário da mesma forma. Na etapa final, uma vez que todos os resultados sejam obtidos, a interface com o usuário confirma os resultados encontrados ao longo do processo e fornece ao usuário um resumo de como foram as etapas do processo de tomada de decisão. Neste caso, são demonstradas através da interface quais as escolhas feitas, regras que foram utilizadas e os motivos dessas escolhas. A interface tem por objetivo ser amigável e propor ao usuário que perguntas do tipo “*What-if*” sejam feitas. Sendo assim, caso o usuário não esteja completamente satisfeito com o mesmo, pode alterar o resultado e auxiliar o sistema em futuras tomadas de decisões.

Na sequência, são apresentados os detalhes de cada etapa da arquitetura proposta.

### 4.3 Reconhecimento das Características das Peças

Os *softwares* CAD e CAM possuem um comum acordo de integração, indiferente da empresa responsável pelos mesmos. Muitas empresas da área de Tecnologia da Informação são especializadas em apenas uma das ferramentas (CAD ou CAM), existindo a necessidade de mesmo assim manter a compatibilidade entre os *softwares*. Entretanto, como é bem conhecido dos usuários dessa área, não é possível manter 100% de integração (mesmo em *softwares* produzidos pela mesma empresa), existindo a perda de dados na transição de CAD para CAM.

Hoje em dia, a importação de um arquivo CAD em uma ferramenta CAM traz consigo, além do desenho 2D ou 3D, diversos outros dados embutidos tais como: dados de características, dados de furação e processos de rosqueamento, cores de *layers*, definição de linhas, etc. Na transição, alguns desses dados podem ser perdidos e necessitam ser adicionados novamente no arquivo produzido pelo *software* CAM. Devido a essa

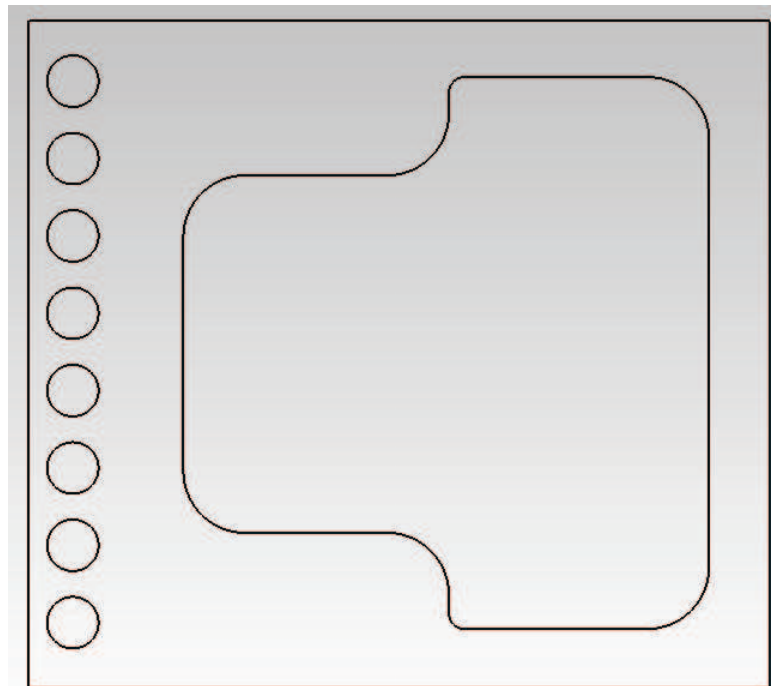
possibilidade, muitos usuários optam por manter-se com apenas uma empresa da área ou por soluções integradas (apenas um *software*) CAD/CAM.

Conforme as afirmações na seção 2.1 (etapa CAD), os desenhos trazidos através da ferramenta CAD para o processo de manufatura podem ser considerados como a entrada do sistema, já que os mesmos representam os itens que serão produzidos. Uma vez que o método proposto traz como foco a parametrização automática do processo, o que é diferente do processo atual, o reconhecimento dos padrões e características das peças passa a ser uma etapa obrigatória.

A importância do reconhecimento das características da peça se dá pelo fato de que as etapas de tomada de decisão propostas nesse método necessitam conhecer tais parâmetros para realizar a escolha correta. Essas características podem ser utilizadas no sistema especialista, que por sua vez faz uso de regras (descritas com mais detalhes na seção 4.5.1) para a escolha dos parâmetros.

Portanto, o reconhecimento dos aspectos de tais peças está diretamente ligado às exigências trazidas das regras utilizadas e por esse motivo, será analisado conforme a demanda de informações. Dado como exemplo a peça apresentada pela Figura 20, caso uma das regras necessite saber se a peça possui formato retangular para tomar uma decisão, então essa afirmação seria verdadeira. De forma análoga, caso necessário, ainda utilizando a peça apresentada pela Figura 20, poderia ser reconhecida a necessidade de furação ou ainda que a peça possua área interna utilizável para o processo de *nesting*.

**Figura 20 - Obtenção das características da peça**



Fonte: elaborado pelo autor (2014).

A abordagem utilizada para interrogar e descobrir as informações relacionadas às peças é uma abordagem heurística utilizando a leitura das geometrias contidas no desenho CAD, uma vez que as mesmas trazem todas as informações necessárias para que essa análise seja realizada.

#### 4.4 Seleção Automatizada da Matéria Prima

Dada a grande quantidade de elementos e a grande variedade de chapas (novas ou sobras aproveitáveis) que podem estar disponíveis em estoque, a questão da seleção automatizada de matéria prima deve ser analisada como um problema relacionado à área de análise combinatória.

Tendo em mente a similaridade dessa questão com os demais problemas de análise combinatória conhecidos, tais como caminho hamiltoniano, caixeiro viajante, problema do carteiro chinês, entre outros; podemos acreditar que o mesmo possui complexidade similar, ou seja, podemos inferir que esse problema é NP-completo (KARP, 1972). Tal complexidade está relacionada à fácil ascensão do número de possíveis combinações em relação ao número de chapas disponíveis a serem processadas.

O número de possibilidades está ligado diretamente à quantidade disponível de chapas de matéria prima que será analisada pelo processo, assim como o número de chapas necessárias para realizar o projeto em execução. Utilizando a fórmula (2) e simplificando o problema para um arranjo simples sem repetição e onde a ordem das chapas é importante, é possível estimar o número aproximado de combinações (GRAHAM et al., 1995).

$$\frac{n!}{(n - p)!} \quad (2)$$

onde:

- (1)  $n$  representa o número de chapas disponíveis no processo
- (2)  $p$  representa o número de chapas necessárias na realização do processo

Por exemplo, em um problema com 50 chapas disponíveis para o processo e aproximadamente 5 chapas necessárias para a obtenção do mesmo, teremos 254251200 combinações possíveis. Entretanto, é importante ressaltar que a área de cada chapa pode variar, fazendo assim com que o número de chapas necessárias na realização do processo tenha variações, e podendo aumentar ou reduzir o número de possíveis combinações.

Esse alto número de possíveis combinações e o complexo *mix* de matérias primas torna evidente o motivo para que a questão seja analisada como um problema de análise combinatória. Devido à natureza do problema e sua complexidade, podemos concluir que o mesmo não possui um algoritmo eficiente (polinomial) (ARORA; BARAK, 2007), e por esse motivo, a seleção automatizada de matéria prima será realizada através de metaheurísticas.

A metaheurística utilizada será a Busca Tabu, sendo que a implementação utilizada nessa etapa do processo não segue a implementação clássica da Busca Tabu. Embora o uso deste algoritmo, em uma visão geral, possua a mesma base teórica e mantenha as características elementares da Busca Tabu convencional; a implementação proposta adiciona à Busca Tabu clássica conceitos atuais de hibridização e programação paralela para gerar soluções próximas às regiões promissoras (GÓMEZ et al., 2014). A implementação proposta é detalhada ao longo da seção 5.3.2.

Diferentemente da parametrização do processo de *nesting*, nesta etapa do problema é possível o uso de metaheurísticas por se tratar de uma decisão basicamente matemática que analisa apenas as matérias primas que estão sendo levadas em conta através do critério de otimização escolhido. Na parametrização, é necessária a execução do algoritmo de *nesting* que envolve um alto tempo de processamento. Já a análise das chapas selecionadas para o

processo pode ser rapidamente realizada através de uma função objetivo. Por esse motivo, acrescido da complexidade do programa em análise, a implementação faz uso de metaheurísticas.

#### 4.5 Parametrização *Nesting*

O processo de parametrização de *nesting* é a etapa mais importante do processo, já que nela é determinada a qualidade da resposta obtida pelo processo. Os critérios adotados podem modificar os resultados obtidos variando a quantidade de matéria prima, horas-máquina, horas-homem, ferramentas utilizadas, tempo total de processo, entre outros fatores envolvidos no processo. Devido ao alto número de fatores associados ao método de parametrização do processo de *nesting* em um sistema CAM, o tempo para finalizar o processo de parametrização e o conhecimento exigido do usuário está acima do ideal.

Dados esses motivos, um sistema especialista é proposto como solução para otimizar essa parte específica do processo. O sistema especialista deve automatizar o processo de escolha de parâmetros tornando o mesmo capaz de se auto gerenciar e reduzindo assim, a interação do usuário com o processo de escolhas. Além disso, o sistema especialista faz com que o nível necessário de conhecimento do usuário seja menor, já que o sistema tem como objetivo substituir um usuário experiente e apontar ao usuário com menor experiência qual seria a melhor escolha.

Ao longo desse capítulo são apresentados mais detalhes sobre o funcionamento, teórico da automação e otimização da etapa de parametrização de *nesting* e do sistema especialista proposto para solucionar o problema.

##### 4.5.1 Sistema Especialista

A implementação do sistema especialista no método proposto visa auxiliar o usuário na decisão dos parâmetros de *nesting* a serem utilizados. O sistema especialista proposto utiliza regras criadas durante a etapa de implementação e criação do mesmo, em conjunto com as características obtidas através da análise dos itens demandados.

O sistema especialista deverá auxiliar o usuário em duas etapas da parametrização:

1. Parametrização da peça: Escolha dos parâmetros individuais da peça, como rotação ou espelhamento.
2. Escolha dos parâmetros de *nesting*: Escolha dos parâmetros específicos do processo de *nesting*.

A implementação do sistema especialista deverá ser feita através do *software* de código aberto *Drools.NET*. Por se tratar de uma versão reestruturada do conhecido *Drools Expert*, conta com o mesmo sistema de regras e mecanismo de inferência.

Uma vez que Benfer, Brent e Furbee (1991) afirmam que um sistema especialista pode oferecer ao usuário explicações sobre as decisões tomadas de diversas formas; os recursos oferecidos pelo *software* também são utilizados para justificar as escolhas tomadas, apresentando-as na interface com o usuário.

Nos parágrafos a seguir, o comportamento dos componentes básicos de um sistema especialista e seus respectivos funcionamentos dentro do sistema de apoio a decisão proposto pelo trabalho são explicados:

- **A base de conhecimento:** Esta é responsável por armazenar todas as regras e fatos obtidos pelas diversas execuções e inferências propostas pelo sistema especialista. A base de conhecimento está ligada, diretamente, com o banco de dados e necessita do mesmo para seu funcionamento. Tal importância se deve ao fato de que a mesma utiliza as informações disponíveis no banco de dados relacionadas a resultados anteriores, materiais disponíveis, entre outras características; e aplica essas informações diretamente nas regras que também são armazenadas pela base de conhecimento.
- **Mecanismo de inferência:** O *software* Drools.NET é responsável pelo mecanismo de inferência onde serão utilizadas as regras armazenadas pela base de conhecimento. Além disso, o mecanismo de inferência leva em consideração os critérios de otimização (descritos abaixo) para chegar a um resultado de acordo com o tipo esperado pelo usuário. Isso se dá pelas prioridades e necessidades de um projeto de manufatura, que podem variar de acordo com o tipo de projeto e o tempo para o mesmo ser realizado.
- **Interface de aquisição de conhecimento:** A interface de aquisição de conhecimento será responsável por transformar os dados resultantes obtidos em conhecimento. Através da análise das regras utilizadas para a obtenção do resultado final, a interface poderá armazenar os passos e quais as ações foram tomadas para a aquisição dessa solução. Dessa forma, o sistema terá seus resultados aprimorados ao longo do número de interações. Além disso, essa interface é responsável por armazenar regras e a forma como as mesmas foram utilizadas e futuramente, apresentar ao usuário de forma clara quais os motivos e quais opções de regras foram utilizadas para a obtenção do resultado. Dessa forma, o usuário tem a possibilidade de adicionar, editar ou ainda excluir regras que possam auxiliar o sistema especialista em futuras tomadas de decisão.
- **Interface com o usuário:** A interface com o usuário do sistema especialista está ligada diretamente com a interface de usuário apresentada pelo sistema de apoio à decisão. A diferença entre ambas não é perceptível ao usuário, porém toda e qualquer informação que necessite da interação ou auxílio do usuário será solicitada através da interface com o usuário do sistema especialista. A interface com o usuário acontece de forma distinta e poderá variar de acordo com o nível de interação com a rotina que o usuário deseja ter. A interface com o usuário deve mostrar ao mesmo os parâmetros que serão escolhidos no processo de *nesting*, tais como: materiais, funções especiais, entre outros. Além disso, embora a explicação e detalhes sobre as regras que foram utilizadas para chegar ao resultado final sejam produzidas pelo sistema especialista, as mesmas serão apresentadas no final do processo onde o sistema especialista já finalizou seu processo. Sendo assim, podemos considerar a explicação de como o resultado foi obtido será apresentada através da interface com o usuário do sistema de apoio à decisão e não da interface específica do sistema especialista.

Dado o grande número de parâmetros e variações de processo possíveis, é importante ressaltar que é inviável o desenvolvimento de um sistema especialista ótimo. Para que isso fosse possível, seria necessário explorar todas as possíveis permutações entre os parâmetros sendo analisados pelo sistema. Esta inviabilidade se dá, principalmente, pelo alto tempo de

processamento necessário para analisar todas essas possíveis combinações. Além disso, o número de regras a serem criadas para que as possíveis permutações fossem avaliadas seria extremamente alto. Portanto, dentro de sua plenitude, o sistema especialista proposto busca resultados otimizados analisando o maior número de variações possíveis, mas não todas as possíveis permutações entre parâmetros e características de usinagem.

#### 4.5.2 Critérios de Otimização

Os critérios de otimização propostos visam possibilitar ao usuário a escolha de suas prioridades em um projeto.

A importância dos diferentes critérios de otimização se dá também pelo fator de adaptabilidade da empresa em relação às metas a serem atingidas e as estratégias adotadas para que essas metas sejam alcançadas. A estratégia competitiva de uma empresa coloca exigências específicas sobre a produção, ao mesmo tempo a estratégia de manufatura da empresa deve ser devidamente concebida para cumprir as metas da estratégia competitiva adotada pela empresa (AMOAKO-GYAMPAH; ACQUAAH, 2008).

A estratégia competitiva adotada por uma empresa guia as estratégias do processo de manufatura e suas respectivas decisões operacionais e tomadas de decisão que fazem com que o processo de manufatura avance no caminho esperado. Para obter vantagens competitivas perante outras empresas, essas estratégias de manufatura devem estar alinhadas com a empresa (AMOAKO-GYAMPAH; ACQUAAH, 2008).

Diversos artigos abordam as estratégias de manufatura e seus possíveis resultados. Dentre os principais fatores analisados e utilizados para aprimorar ou criar práticas inovadoras estão: a qualidade, a flexibilidade, a entrega de produtos e os custos (FERREIRA DE LARA; NEVES GUIMARÃES, 2014). Todavia, a flexibilidade do processo não é relevante para o modelo proposto, uma vez que o sistema lida apenas com processos de manufatura envolvendo chapas como matéria prima. Por esse motivo o foco da otimização está sobre a entrega de produtos e seu tempo de processo e, fortemente, sobre o custo do processo.

Mesmo que os fatores e métodos de produção sejam diversos, inicialmente são propostos três diferentes critérios. Porém, para que seja possível a análise dos critérios de forma eficiente, é necessário que o usuário insira diretamente algumas informações relacionadas ao material utilizado, ferramentas de corte, ao custo de processo, tempo de *setup* entre outras informações. Uma limitação do modelo proposto está no fato de que o critério de otimização é analisado de forma única, não aceitando a escolha de critérios múltiplos.

É importante ressaltar que, já que as características analisadas para a avaliação dos critérios de otimização estão ligadas, em sua maioria, aos materiais e recursos gastos no processo de manufatura (e não ligados diretamente ao projeto que está sendo executado), não é necessária a inserção dos dados de materiais, máquinas ou ferramentas a cada novo projeto. Diferentemente, os dados podem ser inseridos somente quando cadastrados, como por exemplo, um novo material ou uma nova ferramenta.

##### 4.5.2.1 Redução do uso de Matéria Prima

A redução do consumo de matéria prima é um dos critérios mais comuns no meio acadêmico e na maioria dos casos, é um critério de otimização analisado por artigos



relacionados à área de *nesting*. Esse critério não possui a exigência de que o usuário informe os dados sobre o processo, uma vez que apenas a área utilizada é considerada, ignorando completamente custos e tempo de processo.

Portanto, quando utilizado esse critério de otimização, será solicitado à ferramenta *nesting* que utilize a mínima área de usinagem possível, podendo assim limitar a quantidade de chapas utilizadas no processo. Através das restrições, o modelo deve garantir que todas as peças pré-definidas a serem alocadas na chapa façam parte do processo. Além disso, o modelo deve garantir que a área total das peças não exceda a área total das chapas envolvidas no processo. Busca-se minimizar a equação (3):

$$Au = \sum_{p=1}^n Ap \quad (3)$$

onde:

- (1)  $Au$  representa a área total utilizada
- (2)  $p$  representa uma peça a ser processada;
- (3)  $Ap$  representa a área da peça  $p$ ;
- (4)  $n$  representa o número total de peças.

Respeitando (4):

$$\sum_{p=1}^n Ap < \sum_{c=1}^j Ac \quad (4)$$

onde:

- (1)  $c$  representa a chapa sendo utilizada;
- (2)  $Ac$  representa a área de chapa  $c$ ;
- (3)  $j$  representa o número total de chapas sendo utilizadas

#### 4.5.2.2 Minimização do Custo Total do Processo

O custo total envolvido em um processo de manufatura envolve um conjunto de fatores. Esses fatores estão ligados ao custo produtivo e à complexidade da análise realizada, sendo os mesmos ligados diretamente aos fatores que serão levados em consideração na avaliação. Em sua proposta inicial, o critério visa minimizar o custo com base nos principais gastos gerados pelo processo de manufatura, desconsiderando assim fatores indiretos ou de menor relevância.

Os principais fatores levados em consideração nesse critério são: custo da chapa (matéria prima), custo da ferramenta com base em sua vida útil, custo de operação da máquina CNC.

Tais fatores são demonstrados na equação (5):

$$CT = \sum_{c=1}^j CCc + \sum_{f=1}^l (CFf \times QFf) + CM \times TU \quad (5)$$

onde:

- (1)  $CT$  representa o custo total do processo;
- (2)  $c$  representa uma chapa utilizada no processo;
- (3)  $CC$  representa o custo de uma chapa  $c$ ;
- (4)  $f$  representa uma ferramenta utilizada no processo;
- (5)  $l$  representa o número total de ferramentas;
- (6)  $CF$  representa o custo da ferramenta  $f$ ;
- (7)  $QF$  representa a quantidade de ferramentas  $f$  utilizadas no processo;
- (8)  $CM$  representa o custo do uso da máquina por minuto;
- (9)  $TU$  representa o tempo de usinagem.

#### 4.5.2.3 Minimização do Tempo de Processo (*Makespan*)

Nesse critério de otimização, busca-se reduzir o tempo total do processo de manufatura. É importante ressaltar que, mesmo que optando por otimizar o processo com base em seu tempo total de manufatura, a ferramenta de *nesting* (mesmo que em segundo plano) sempre apresenta resultados que buscam minimizar a quantidade de matéria prima utilizada.

O tempo total de processo (6) é constituído pela soma do tempo de usinagem e o tempo de *setup*, incluindo o tempo de troca de chapas (quando utilizando mais de uma chapa) e troca de ferramentas quando necessário. O tempo total de configuração da máquina pode ser representado pelo número de chapas utilizadas no processo multiplicado pelo tempo de troca da chapa, somado ao número de ferramentas utilizadas no processo e multiplicado pelo tempo de troca de ferramenta (7).

$$TT = TU + TS \quad (6)$$

Sendo assim:

$$TS = TC \times QC + \sum_{f=1}^l Tf \times Qf \quad (7)$$

onde:

- (1)  $TT$  representa o tempo total do processo;
- (2)  $TU$  representa o tempo total de usinagem;
- (3)  $TS$  representa o tempo total de *setup* de máquina;
- (4)  $TC$  representa o tempo total para troca de chapas;
- (5)  $QC$  representa a quantidade de chapa  $c$  utilizadas no processo;
- (6)  $T$  representa o tempo de troca de uma ferramenta  $f$ ;
- (7)  $Q$  representa a quantidade de ferramentas  $f$  utilizadas no processo.

## 4.6 Extração das Sobras Aproveitáveis

Após o processamento e obtenção do resultado, é necessário processar o material caracterizado como sobra, podendo este ser considerado como aproveitável ou não. De forma geral, sobras aproveitáveis podem ser definidas como regiões, na grande maioria dos casos retangulares, que possuem altura ou largura igual ou superiores a de um ou mais itens manufaturados e são sobras de um processo de corte que podem ser utilizados na manufatura de novos itens no futuro (ABRANTES, 2013).

Utilizando as contribuições trazidas por Cherri e Vianna (2013), as perdas são caracterizadas como perdas pequenas, perdas intermediárias ou retalho. De acordo com a definição dos autores, uma perda pode ser considerada não tão pequena caso a mesma seja maior que uma perda pequena, porém, não grande o suficiente para ser um retalho. Na implementação proposta pelos autores, retalhos não são contabilizados como perdas e seus recursos podem ser realocados no estoque.

Conforme Cherri e Vianna (2013), os retalhos são classificados de acordo com as seguintes especificações. Sejam  $(L_k, W_k)$  as dimensões da chapa  $k$ ,  $k = 1, \dots, n$ , em estoque, e seja  $(l, w)$  uma sobra na chapa  $k$  ( $k = 1, \dots, n$ ) e  $As = lw$  a área correspondente. Então se define como retalho através de um único critério.

Caso para um dado  $\phi$ ,  $0 < \phi < 1$  as condições demonstradas abaixo sejam verdadeiras então a sobra  $(l, w)$  é um retalho.

$$l \geq \phi L_k \quad \text{e} \quad w \geq \phi W_k \quad (8)$$

Os autores reforçam a ideia de que  $\phi L_k$  e  $\phi W_k$  são valores arbitrários e por isso podem ser modificados e, ao invés de utilizar as dimensões mínimas para definir o retalho, os valores escolhidos pelo usuário podem ser utilizados. Na implementação proposta através desse trabalho, tais parâmetros são as dimensões do item de menor área envolvido no projeto.

Ainda sobre o estudo de Cherri e Vianna (2013), perdas aceitáveis (perdas consideradas pequenas para serem descartadas) são definidas através de dois critérios. O primeiro critério é dado através das condições abaixo e caso sejam verdadeiras  $(l, w)$  é uma perda aceitável.

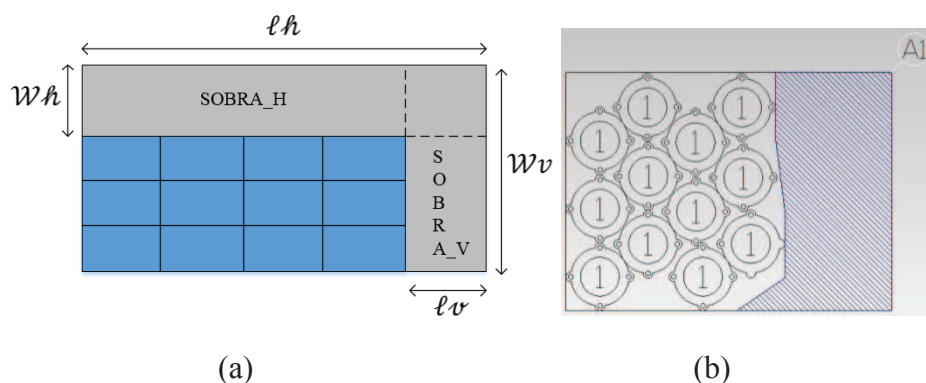
$$0 < \xi < 1 \quad \text{e} \quad As < \xi L_k W_k \quad (9)$$

Abstraindo o modelo proposto pelos autores,  $\xi$  é definido através da equação (3). Já  $\psi_k$  representa o percentual de perda mínima admissível para cada chapa  $n$  em estoque e é obtido a partir do percentual de perda obtida da solução original do problema, ignorando as sobras aproveitáveis para a chapa  $n$ .

$$\xi = \frac{\Psi_k}{4} \quad (10)$$

Uma vez que os métodos propostos por Cherri e Vianna (2013) sejam conhecidos, a sua integração com as contribuições feitas por Ferrary et al. (2013) passam a ser o foco do trabalho. A Figura 21 traz exemplos de sobras aproveitáveis apresentadas nos artigos de Cherri e Vianna (2013) (a) e Ferrary et al. (2013) (b).

**Figura 21 - Exemplos de extração de sobras aproveitáveis**



Fonte: adaptado de Cherri e Vianna (2013) (a) e Ferrary et al. (2013).

Neste trabalho, métodos heurísticos foram propostos para a obtenção do perfil que representa as sobras do processo de *nesting*. Todavia, as sobras não são qualificadas como aproveitáveis ou não, sendo este o motivo da união dos estudos e considerando como ideal para o método proposto.

A implementação do que é proposto em Ferrary et al. (2013) é importante para essa etapa, já que parte da melhoria proposta é a alimentação de um banco de dados contendo as chapas consideradas como sobras. Todavia, uma vez que o método está sendo aplicado a um *software* CAM, somente o cálculo matemático das sobras geradas não é suficiente, sendo assim o método apresentado no artigo contribui gerando o perfil que representa as sobras geradas pelo processo.

## 5 IMPLEMENTAÇÃO DO MODELO PROPOSTO

Para avaliar o modelo proposto, genérico e aplicável para qualquer ciclo de manufatura envolvendo *nesting* e chapas como matéria prima, a implementação das etapas descritas no capítulo 4 foi realizada no *software* CAM Alphacam. Tal escolha se dá pela facilidade de acesso às suas bibliotecas de programação API; biblioteca que possibilita utilizar os comandos do *software* através de linguagem de programação assim como coletar informações de recursos trazidos do CAD.

Uma vez que o trabalho visa otimizar o ciclo de manufatura existente de forma geral, não se faz necessário o desenvolvimento de um novo sistema de *nesting* para o estudo. Para as tarefas de *nesting*, a ferramenta oferecida pelo *software* Alphacam será utilizada. Por conseguinte, na etapa de parametrização são levados em consideração apenas os recursos oferecidos pelo *software* e não todas as possibilidades apresentadas pela Tabela 3 .

Através da biblioteca API do *software*, é possível, de forma automatizada ou em algumas etapas manualmente, passar por todas as etapas conforme descrito abaixo:

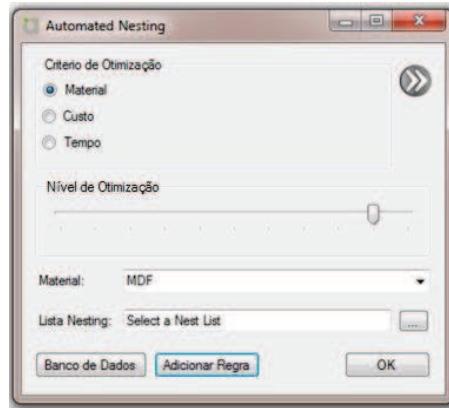
1. CAD - Importar desenhos trazidos do *software* CAD e utilizar a biblioteca API para obter as informações das geometrias.
2. CAM - Uso do *software* para aplicar caminhos de ferramenta e processos de usinagem em geral.
3. Parametrização *Nesting* - (Escolha da matéria prima) Uso da metaheurística Busca Tabu para encontrar, no banco de dados, a melhor escolha de matéria prima para ser utilizada no processo, com base no critério de otimização escolhida pelo usuário.
4. Parametrização *Nesting* - Uso do sistema especialista proposto visando obter a parametrização que encontre o melhor resultado possível, com base no critério de otimização escolhida pelo usuário.
5. *Nesting* - Uso da ferramenta oferecida pelo *software* para a otimização e acomodação das peças na chapa.
6. Resultado - Análise do resultado feita pelo usuário através da interface do *software*.
7. Processamento das sobras aproveitáveis - O resultado é analisado e, caso o mesmo gere sobras aproveitáveis, essas são analisadas pelo sistema proposto e armazenadas no banco de dados.
8. Código NC - O *software* é utilizado para gerar código NC e o processo de usinagem pode ser realizado.

### 5.1 Interface do *Software*

Embora o SAD proposto tenha como um dos objetivos minimizar a necessidade de intervenção do usuário com o sistema, uma interface com o usuário continua um componente obrigatório e muito importante para o SAD proposto. Todavia, essa interface visa ser simplificada e de fácil manuseio para que dessa forma não demande um alto nível de conhecimento do usuário para utilizar o sistema.

Através dessa interface, o usuário pode selecionar as principais informações que serão repassadas para o sistema. Na tela principal do SAD, é possível selecionar informações como: Critério de otimização, Nível de otimização, Material a ser utilizado, Lista de peças. A interface principal é demonstrada pela Figura 22.

**Figura 22 - Interface do sistema de tomada à decisão**



Fonte: elaborado pelo autor (2014).

Uma das escolhas importantes que é realizada é a seleção da “Lista *Nesting*” que é responsável por definir as peças que serão submetidas ao processo, assim como suas respectivas quantidades. Além disso, essa lista é responsável por definir alguns dos parâmetros que devem ser selecionados pelo usuário (definidos na seção 5.4.1.3). Essa lista é um arquivo de texto, que pode ser facilmente gerado pelo *software*, contendo essas informações. Um exemplo de “Lista *Nesting*” é apresentado na Figura 23.

**Figura 23 - Exemplo de “Lista de *Nesting*”**

```

$VERSION 2.40

$SETUP
TRUEADNEST Nesting method (TRUE = True Shape, R
2 Tool Paths (0=Geometry, 1=Tool Paths, 2=Tool
0 Gap Between Parts
0 Extra Lead-in Gap
0 Subroutines (0/1)
0 Nest Direction (0=Pack to left, 1=Pack to bot
0.920000 Search resolution (smaller is slower b
2.000000 Time per sheet, in seconds (0-9999). P
0 Gap at Sheet Edge

$ITEM 1
C:\Users\FFerrary\Google Drive\Unisinos\Computaçã
1 Number Required
1 Priority
0 Rotation Angle
0 Allow Mirror (0/1)
0 Try Rotated Part First (0/1)
0 Maximum number of parts on each sheet
0 Quality Zone (0 for none else 1 to 4)
0 Kit Number (or 0 for none)

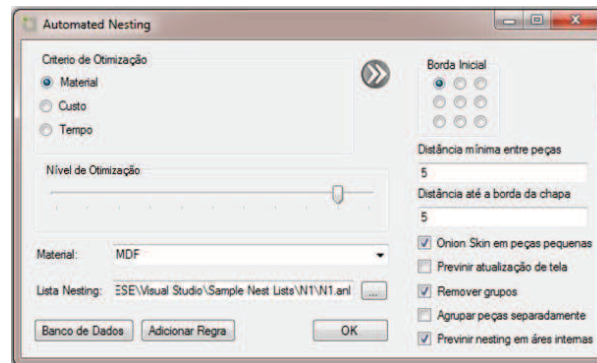
$ITEM 2
C:\Users\FFerrary\Google Drive\Unisinos\Computaçã
15 Number Required
1 Priority
0 Rotation Angle
0 Allow Mirror (0/1)
0 Try Rotated Part First (0/1)
0 Maximum number of parts on each sheet
0 Quality Zone (0 for none else 1 to 4)
0 Kit Number (or 0 for none)

```

Fonte: elaborado pelo autor (2014).

Além disso, parâmetros específicos do processo de *nesting* não relacionados às peças que serão submetidas ao trabalho e que não são otimizados também podem ser definidos através da interface com o usuário. Todavia, mesmo que esses parâmetros não sejam otimizados pelo processo, a seleção padrão dos mesmos é apresentada na interface com o usuário de forma que vise apresentar a seleção mais recomendada para processos em geral. Essas opções extras são apresentadas ao usuário somente quando o mesmo opta por expandir o diálogo responsável pela interface conforme apresentado na Figura 24.

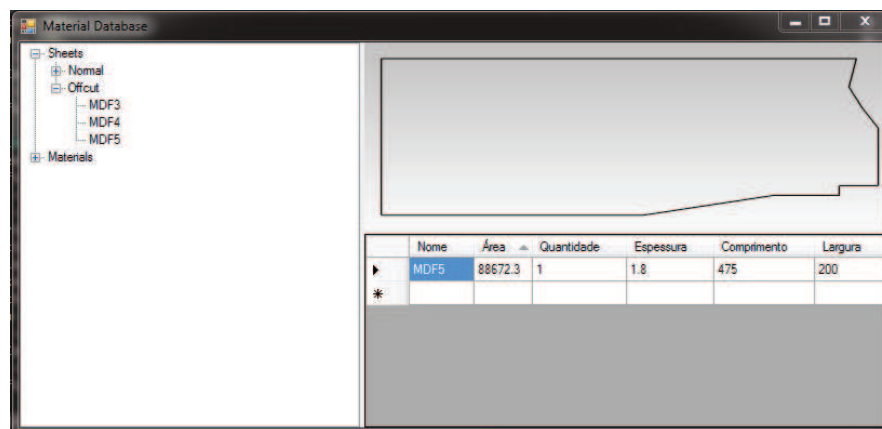
Figura 24 - Opções extras do diálogo



Fonte: elaborado pelo autor (2014).

Também através da interface com o usuário, é possível que o usuário manipule o banco de dados com a finalidade de verificar e controlar as chapas que estão disponíveis em estoque. Além disso, é possível adicionar novas chapas ao banco de dados manualmente, sendo assim possível para o usuário manter o mesmo atualizado. No banco de dados, as chapas são divididas entre chapas normais ou sobras aproveitáveis (*offcut*) e, quando o usuário seleciona uma das chapas, as informações sobre a mesma e uma visualização do perfil desta são apresentadas na interface. A interface do banco de dados pode ser observada na Figura 25.

Figura 25 - Interface do banco de dados



Fonte: elaborado pelo autor (2014).

## 5.2 Reconhecimento das Características

Em virtude de o SAD proposto suportar diversos formatos de arquivo CAD, é necessário que as características das peças sejam conhecidas, uma vez que para muitos arquivos e formatos as informações são perdidas durante o processo de tradução das informações. Para isso, é necessário realizar uma avaliação de cada peça submetida ao SAD.

Essas informações devem ser obtidas já que são relevantes para o processo de decisão, e são utilizadas diretamente pelo sistema especialista, que faz uso das informações para determinar como o processo será realizado; assim como determinar como será realizada a parametrização do processo.

Através da biblioteca API do *software*, é possível avaliar peça por peça obtendo informações relevantes para o processo de decisão. Essas informações podem ser de informações simples sobre o desenho CAD, assim como informações sobre os tipos de ferramenta aplicados a cada peça. A Tabela 5 traz um resumo das informações extraídas das peças para que o processo de decisão seja possível.

**Tabela 5 - Informações obtidas das peças**

| <b>Informações Necessárias</b>     | <b>Descrição</b>                                                                                                                                                                                                                                                                      |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Possui Arcos                       | É necessário descobrir se a peça é composta apenas por linhas, ou também possui arcos. Essa informação é utilizada para definir o formato da peça e se geometrias com formatos livres são utilizadas. Geometrias que não utilizam arcos possuem maior tendência de serem retangulares |
| Possui Área Interna                | É necessário definir se a peça possui ou não geometrias internas que podem ser utilizadas para acomodar outras peças menores                                                                                                                                                          |
| Permite Rotação                    | Determinar se a rotação das peças é permitida ou não através da avaliação dos materiais selecionados                                                                                                                                                                                  |
| Permite Espelhamento               | Determina se é possível espelhar peças ou não com base no material selecionado.                                                                                                                                                                                                       |
| Tamanho Médio das Peças            | Através da análise de todas as peças submetidas ao processo, determina qual o tamanho médio das peças                                                                                                                                                                                 |
| Número de Peças Pequenas           | Determina o número de peças pequenas com base no tamanho médio das peças. Para ser pequena, a peça deve possuir dimensões equivalentes a 30% do tamanho médio                                                                                                                         |
| Número de Peças Grandes            | Determina o número de peças grandes com base no tamanho médio das peças. Todas as peças que não são consideradas pequenas são automaticamente determinadas como grandes                                                                                                               |
| Máximo Raio de Ferramenta          | Após avaliar todas as ferramentas utilizadas no processo, determina qual a maior ferramenta utilizada e armazena o valor de seu raio                                                                                                                                                  |
| Existe Corte com Serra             | Determina se o tipo de corte utilizado nas peças é o corte com serra circular                                                                                                                                                                                                         |
| Possui Entrada/Saída de Ferramenta | Determina se as peças submetidas ao processo possuem ou não entrada/saída de ferramenta adicional                                                                                                                                                                                     |
| Possui Ciclo de Furação            | Determina se alguma das peças possui ciclo de furação sendo aplicada a mesma                                                                                                                                                                                                          |
| Possui Operação Interna            | Determina se existem operações internas sendo realizadas na peça ou se apenas operações externas estão sendo realizadas para que a peça seja destacada do material                                                                                                                    |
| Possui Apenas Operações Internas   | Determina se apenas operações internas estão sendo realizadas, ou seja, peças não serão destacadas da chapa durante o processo de corte                                                                                                                                               |
| Número de Caminhos de Ferramenta   | Determina quantos caminhos de ferramentas estão sendo aplicados nas peças. Quando selecionando apenas geometrias sem caminhos de ferramentas aplicados a elas, esse número será 0                                                                                                     |
| Número de Ferramentas              | Determina o número de ferramentas diferentes sendo utilizadas no processo                                                                                                                                                                                                             |
| Corte <i>Profiling</i>             | Determina se o corte é do tipo <i>Profiling</i> (Corte a laser, Jato d'água, etc.)                                                                                                                                                                                                    |
| Corte <i>Wire</i>                  | Determina se o corte é do tipo <i>Wire</i> (Corte a Fio, Eletro erosão, etc.)                                                                                                                                                                                                         |

Fonte: elaborado pelo autor (2014).

As informações obtidas nesta etapa são diretamente responsáveis por limitar o sistema especialista, uma vez que só podem ser desenvolvidas regras que utilizam as informações obtidas nesse processo. Ou seja, para aprimorar o sistema especialista sempre será necessária uma melhoria deste módulo para que as mesmas sejam utilizadas no processo de decisão.



### 5.3 Seleção Automatizada da Matéria Prima

A escolha da matéria prima deve ocorrer de forma que, com base no critério de otimização escolhido, seja possível melhorar o processo através da escolha do conjunto de chapas (matéria prima) que melhor represente tal critério e as respectivas definições do projeto de *nesting* usadas.

A seleção automatizada da matéria prima é realizada através de técnicas ligadas à área de Pesquisa Operacional. Dada a grande quantidade de combinações de matéria prima possíveis a serem utilizadas no processo podemos caracterizar essa etapa individualmente como um problema de análise combinatória. Neste caso, conforme apontado pela seção 3.4.2, metaheurísticas são métodos frequentemente utilizados para resolver problemas dessa natureza. Por esse motivo, a metaheurística Busca Tabu foi escolhida como método para a resolução deste problema e os detalhes de sua implementação são apresentados ao longo da seção 5.3.2.

O processo de escolha de quais serão as chapas utilizadas, acontece em 2 etapas:

- **Avaliação das matérias primas disponíveis banco de dados:** Através dessa etapa, toda ou qualquer chapa disponível no banco de dados será analisada e avaliada. Essa análise ocorre para que seja possível concluir se a mesma é adequada ou não para o projeto. Caso esta não seja adequada, a matéria prima será ignorada no processo de escolha.
- **Busca Tabu:** Esta etapa é responsável por otimizar o processo através do uso da metaheurística Busca Tabu, sendo essa responsável por realizar a seleção do melhor conjunto de chapas a ser utilizado no processo.

Ao longo das próximas seções, a implementação dessas etapas é descrita de forma mais detalhada.

#### 5.3.1 Avaliação das Matérias Primas Disponíveis no Banco de Dados

O processo de avaliação das matérias primas começa através da criação de uma lista envolvendo todas as chapas contidas no banco de dados, incluindo chapas novas ou chapas extraídas como sobras aproveitáveis. Nessa lista, são também armazenadas todas as informações de cada matéria prima, sendo que tais características são demonstradas pela Tabela 6.

**Tabela 6 - Características armazenadas no banco de dados**

| Parâmetros                            |                        |
|---------------------------------------|------------------------|
| Sobra aproveitável (verdadeiro/Falso) | Largura                |
| Nome                                  | Área                   |
| Quantidade                            | Custo                  |
| Espessura                             | Comentários do usuário |
| Comprimento                           |                        |

Fonte: elaborado pelo autor (2014).

As informações sobre cada material são de suma importância para o processo de avaliação; uma vez que as mesmas são utilizadas para definir se a chapa pode ou não ser utilizada no processo. O sistema avalia individualmente as chapas contidas nessa lista e, ao final de sua análise, deverá considerá-las como aptas ou não para o processo.

A primeira avaliação consiste em analisar se o tipo de material coincide com o requisito do projeto. Cada matéria prima possui características físicas e mecânicas distintas e, por esse motivo, o material selecionado para o projeto é definido pelo usuário e de forma alguma o sistema deverá tentar utilizar diferentes materiais.

A grande variedade de materiais no banco de dados é dependente diretamente do tipo de manufatura realizada pelo usuário, por exemplo: em um banco de dados de uma empresa do ramo de móveis, é possível encontrar variedades do material do tipo madeira com características diferente; tais como: MDFs (*Medium Density Fiberboard*), HDFs (*high-density fiberboard*) ou ainda, diferentes tipos de madeiras, tais como: Mogno, Pinheiro, Acácia, Pinus, entre outros. Lembrando, ainda, que o banco de dados pode incluir metais ou qualquer outro material em formato plano que seja adequado ao processo de *nesting*. A Figura 26 demonstra, brevemente, alguns tipos distintos de chapas que poderiam estar armazenadas no banco de dados.

Figura 26 - Diferentes tipos de madeira



Fonte: extraído da internet (2014)<sup>1</sup>.

Devido a todas essas variações possíveis na matéria prima, serão incluídas na seleção apenas chapas que correspondam exatamente ao especificado no projeto de manufatura. Esse

<sup>1</sup> Disponível em: <http://marinasadesign.blogspot.co.uk/2013/09/mdf-mdp-e-compensado-quando-e-como-usar.html>

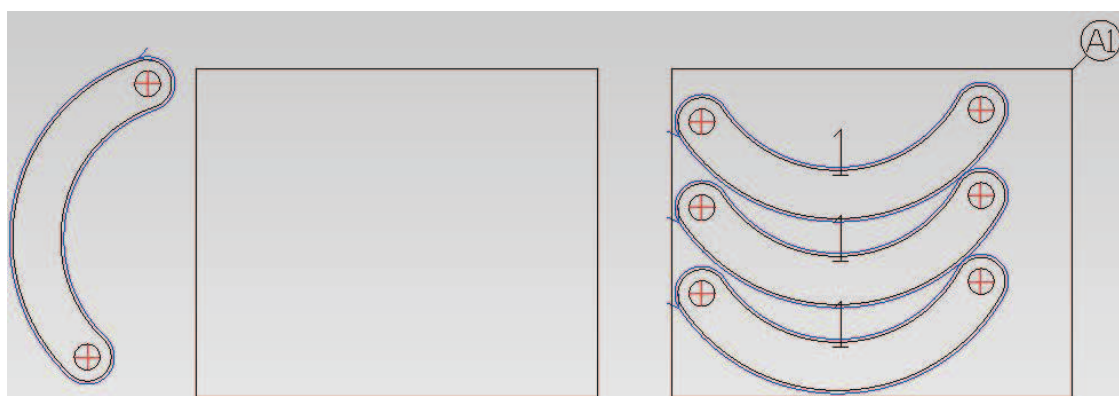
reconhecimento é feito através da característica definida como “tipo” no banco de dados. Ou seja, se é definido que no processo serão utilizados apenas materiais MDF revestido da cor Cerezo Athenas, então uma filtragem será aplicada para selecionar apenas materiais que respeitem tipo, cor e demais características definidas.

A segunda avaliação acontece de forma a garantir que as peças sejam compatíveis em relação ao tamanho (altura e largura) da chapa a ser utilizada. Ou seja, esta avaliação deve eliminar chapas que sejam pequenas demais para acomodar as peças submetidas ao processo de *nesting*. Para que a chapa seja compatível com o processo, esta deve ter capacidade de acomodar pelo menos uma peça ou então será desconsiderada na seleção. A avaliação acontece através da verificação individual de cada chapa e nessa avaliação, sua altura e largura são comparadas com a peça a ser produzida. Sendo assim, é possível utilizar apenas uma amostra de cada peça, já que mesmo que o projeto demande mais de uma peça do mesmo tipo; analisar apenas a amostra será suficiente para definir se a peça pode ou não ser encaixada na chapa.

Caso o tamanho seja insuficiente e nenhuma peça possa ser acomodada nesta chapa, não é necessário que a mesma seja analisada pelo processo. Contudo, em alguns casos as peças podem ser rotacionadas para melhor acomodação na chapa; em situações especiais não é possível fazer a avaliação apenas através da altura e largura das chapas e peças. Neste caso, é necessária uma segunda verificação utilizando o sistema de *nesting* para acomodar a peça dentro da chapa utilizando a rotação permitida pelo projeto de manufatura.

Caso, mesmo aplicando a rotação na peça, a chapa não possua espaço suficiente para acomodar a amostra, a chapa sendo avaliada deverá ser removida na seleção. A Figura 27 demonstra o exemplo de uma peça que não possui encaixe em uma chapa (primeira situação), devido à falta de permissão de rotação. Caso existisse a possibilidade de rotação (segunda situação), a chapa seria considerada na seleção e possibilitaria o encaixe de três peças.

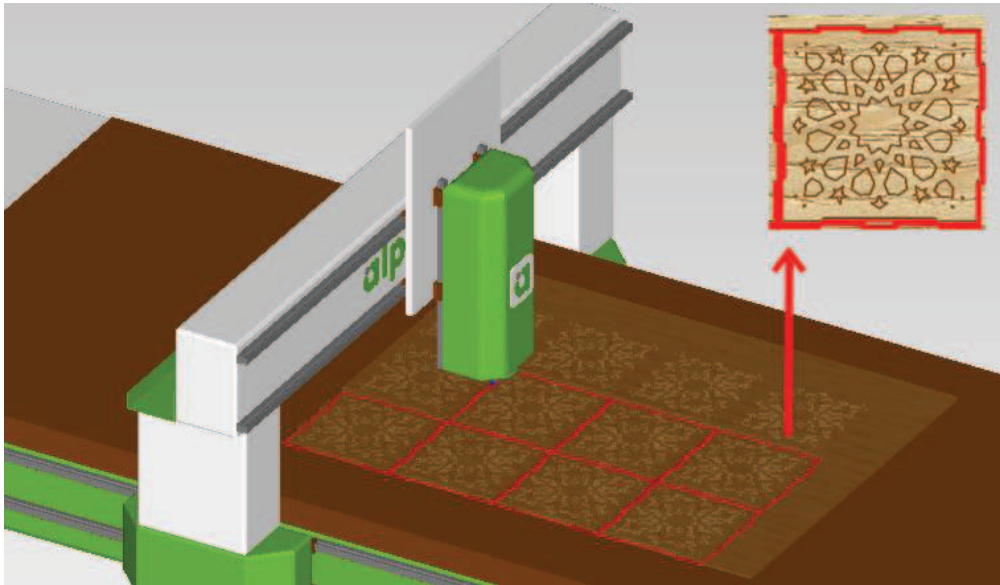
**Figura 27 - Peça que necessita de rotação para ser encaixada**



Fonte: elaborado pelo autor (2014).

A terceira avaliação tem como finalidade garantir que a espessura da matéria prima seja suficiente em relação a profundidade de corte utilizada no processo de usinagem. Essa avaliação é dependente do processo de usinagem em situações que as chapas estão fixas à mesa da máquina CNC. Tal cautela existe para evitar que a profundidade de corte exceda a espessura da chapa, criando a possibilidade de colisão entre a mesa da máquina e podendo causar a perda da ferramenta ou em alguns casos danos à máquina CNC. A Figura 28 demonstra, através de simulação, um processo onde a ferramenta excede a espessura da chapa e colide contra a mesa da máquina CNC (colisão destacada em vermelho).

**Figura 28 - Exemplo de corte excede a espessura do material**



Fonte: elaborado pelo autor (2014).

Por esse motivo, a profundidade de corte de cada peça será verificada e após realizar a análise de todas as peças, a maior profundidade de corte entre as mesmas é comparada com a espessura de cada chapa. Caso a profundidade máxima de corte exceda a espessura da chapa, a mesma será removida da seleção para o processo de manufatura. Através da interface com o usuário é possível definir uma tolerância extra para tal análise, já que muitas vezes é possível fixar a peça de forma distinta ou ter uma determinada distância entre a chapa e a mesa da máquina CNC.

Em conjunto, essas etapas de avaliação ajudam não só a alcançar o resultado esperado pelo usuário, mas também auxiliam na redução do número de chapas a serem analisadas pela Busca Tabu; já que são eliminadas antes de iniciar o processamento realizado pelo algoritmo. Com a redução do número de chapas a serem analisadas pelo processo, torna-se mais fácil chegar a um resultado ótimo e reduz o tempo de processamento para obtê-lo.

### 5.3.2 Busca Tabu

Na implementação realizada, os principais métodos propostos pela implementação clássica de Glover (1986) são utilizados. Ao longo desse capítulo, são descritos os métodos utilizados para realizar os movimentos de uma solução candidata para outra dentro de uma vizinhança; métodos utilizados para gerar uma nova vizinhança, métodos utilizados para intensificação e diversificação da busca, além da definição do critério de aspiração e da lista Tabu utilizada.

Assim como em Gómez et al. (2014), a Busca Tabu proposta nesse estudo utiliza métodos modernos para aprimorar a implementação clássica do algoritmo. Deste modo, utiliza técnicas de processamento paralelo e processamento multitarefa com a finalidade de aprimorar o tempo de resposta e permitir um número maior de tentativas na busca de uma resposta otimizada. Ao longo dessa seção são detalhadas as principais características sobre a implementação da Busca Tabu proposta.

### 5.3.2.1 Solução Inicial

O tipo de solução gerada varia drasticamente de acordo com o critério de otimização escolhido. Ao longo dessa seção os algoritmos adotados serão explicados.

Quando selecionada a opção “**redução do uso de matéria prima**”, a área necessária para acomodar todas as peças é utilizada, pois esse critério busca reduzir a quantidade de material utilizada no processo de manufatura. O algoritmo utilizado para definir a solução inicial da Busca Tabu é um algoritmo não determinístico. Esse comportamento se dá devido à forma como a primeira chapa a ser utilizada é selecionada.

Conforme descrito previamente, as chapas aptas para o processo são armazenadas em uma lista sendo que cada uma possui seu próprio índice único. Portanto, é possível selecionar de forma randômica uma dessas chapas através do sorteio de um número entre 0 e n-1 (onde n-1 o número de chapas nessa lista), onde o número sorteado representa o índice da primeira chapa que será utilizada no processo.

Inicialmente, a área da primeira chapa selecionada é reduzida da área total necessária, obtendo-se a quantidade de área para acomodar as demais peças. Caso o valor resultante seja negativo, sabemos que a área da primeira chapa é suficiente para acomodar todas as peças e, por esse motivo, não é necessária a seleção de uma segunda chapa. Todavia, caso esse valor seja positivo, o que significa que mais chapas são necessárias para acomodar todas as peças, o algoritmo irá buscar na lista de chapas disponíveis uma opção que possua a quantidade de área mais próxima da necessária. Essa seleção acontece de forma similar ao que ocorre nos algoritmos gulosos, onde a escolha é sempre realizada com base na situação atual e sempre busca o melhor mínimo local (BLACK, 2014). Esse processo se repete até que a área resultante seja negativa e ao fim de cada iteração, a nova chapa escolhida é adicionada a solução.

É importante ressaltar que pela natureza do processo, onde peças precisam ser encaixadas da melhor forma possível na matéria prima, é praticamente nula a chance de preencher 100% do material utilizado. Por essa perda natural do processo, é necessário adicionar um fator de perda na área total necessária para acomodar as peças. O valor extra adicionado é armazenado no banco de dados e consiste na quantidade de sucata gerada nos últimos 10 processos realizados pelo sistema de apoio à decisão. Caso não existam execuções suficientes para determinar o valor médio de perda, o padrão utilizado é  $\frac{1}{4}$  da área total necessária.

Quando selecionando “**minimização do custo total do processo**”, o custo total das chapas selecionadas é analisado. Embora a função objetivo descrita na seção 4.5.2.2 leve em consideração as ferramentas, nessa etapa da automação apenas a seleção de material é realizada. Portanto, essa escolha não afetará nas ferramentas utilizadas no processo, tornando assim possível ignorar a seleção das mesmas. Similarmente à escolha que visa a redução do uso de matéria prima, a área total das peças é utilizada para verificar se mais chapas são necessárias após a adição de cada nova chapa à seleção.

A natureza desse algoritmo faz com que ele busque o conjunto de chapas com menor preço para serem utilizadas no processo. O algoritmo adotado é a Busca Gulosa, onde se busca sempre a próxima chapa mais barata disponível. Todavia, esse não é o único fator a ser analisado já que a função objetivo também leva em consideração o custo do uso da máquina. A seleção, portanto, acontece de forma que ambos os custos sejam analisados.

A seleção da chapa, apenas considerando o custo da mesma, poderia fazer com que diversas chapas pequenas fossem selecionadas, aumentando significativamente o tempo de usinagem. Isso ocorre porque, para cada chapa selecionada, é necessário adicionar o tempo de troca da mesma ao processo e em alguns casos, o tempo de parada de máquina para esse processo ser feito de forma manual. Por esse motivo, é estipulado que para ser selecionada, a chapa necessita ter pelo menos 30% da área restante para que as peças sejam encaixadas.

Inicialmente, a chapa com menor custo na lista é selecionada e seu custo é computado. Após essa seleção, a próxima chapa com menor custo e que respeite a regra estipulada para a área será selecionada. Esse processo irá se repetir até que a área restante seja negativa.

Quando selecionando a opção “**minimização do tempo de processo**”, o número de chapas a ser utilizado é a principal variável a ser analisada. Isso ocorre já que um dos principais fatores que afetam o tempo de processamento é a troca entre chapas (*setup*). Além disso, reduzir o número de chapas reduz, conseqüentemente, o número de troca de ferramentas já que devem existir menos retrações.

Visando reduzir o número de chapas, esse algoritmo tem a propensão de selecionar as maiores chapas disponíveis no banco de dados. Nesse caso, o custo do processo será totalmente ignorado, já que se considera que o usuário deseja que o processo seja finalizado rapidamente.

O algoritmo, assim como os demais, faz a busca da melhor chapa a ser utilizada através de uma Busca Gulosa. Sendo que esta opção de busca sempre procura pela maior chapa possível. Para evitar um alto nível de sucata, é estipulado que a área da chapa selecionada não exceda um determinado limite. Esse limite adotado é o mesmo limite de sucata definido como área extra na otimização, com base no uso de matéria prima, e será gerado com base nas últimas 10 execuções do sistema. Caso não existam execuções suficientes para determinar o valor médio de perda, o padrão utilizado também será  $\frac{1}{4}$  da área total necessária.

O resultado obtido ao final de todas as iterações, independente de seu critério de otimização, é considerado como a primeira solução candidata na resolução do problema. A Figura 29 mostra, de forma simplificada, o algoritmo utilizado.

**Figura 29 - Pseudocódigo do algoritmo (solução inicial)**

```

Inicialização
A := Área total das peças
MLista := Lista de todos os materiais no banco de dados
Enquanto A < AreaDasChapasSelecionadas faça
  Se critério de otimização = material então
    Realiza busca gulosa, na lista MLista, pelas chapas que minimizem a
    área total utilizada
  Fim se;
  Se critério de otimização = custo então
    Realiza busca gulosa, na lista MLista, pelas chapas com menor custo
  Fim se;
  Se critério de otimização = tempo então
    Realiza busca gulosa, na lista MLista, pelos maiores materiais
    disponíveis
  Fim se;
Fim Enquanto;
Cria solução com base no critério de otimização;
Fim

```

### 5.3.2.2 Intensificação da Busca

A intensificação da busca procura por vizinhos próximos da solução candidata. Isso acontece através da troca de um número pequeno de chapas dentro da solução candidata. Ao total, são geradas dez soluções com a finalidade de formar uma nova vizinhança e que será analisada no final de cada iteração. Dessas 10 soluções, 5 são geradas através da troca de uma chapa e as demais 5 são geradas através da troca de 2 chapas.

O algoritmo utilizado sorteia, de forma aleatória, a chapa e/ou chapas que serão removidas da solução candidata. Embora um número pequeno de chapas seja removido, o número de chapas que serão utilizadas não pode ser estipulado. Isso ocorre porque, de forma similar ao algoritmo utilizado para gerar a solução inicial, é necessário garantir que a solução de seleção de chapas sendo criada possua área suficiente para acomodar todas as peças.

Por esse motivo, caso após eliminar essa chapa da seleção a área das chapas não seja suficiente para acomodar as peças; é necessário re-selecionar outras chapas a fim de substituir aquela que foi removida. A re-seleção acontece de forma distinta, dependendo do critério de otimização selecionado.

Independentemente do critério de otimização selecionado pelo usuário, o algoritmo de intensificação será o mesmo. Apenas variando de forma aleatória a solução candidata atual, a seleção da chapa a ser reinserida acontece através do sorteio de um número aleatório e a chapa com índice sorteado será adicionada à solução gerada. Esse processo se repete até que a área resultante seja negativa. Ao final da escolha das chapas, a lista Tabu é verificada de forma a validar se a solução criada é ou não um movimento Tabu. Caso a mesma seja um movimento Tabu, esta será descartada. A Figura 30, mostra de forma simplificada, o algoritmo utilizado.

**Figura 30 - Pseudocódigo do algoritmo (intensificação da busca)**

```

Inicialização
A := Área total das peças
N = Número de chapas a serem removidas
Enquanto N > 0 faça
  Remove chapa aleatória
  N--
Fim Enquanto
Enquanto A < AreaDasChapasSelecionadas faça
  Adiciona uma chapa aleatória a solução
Fim Enquanto;
Cria solução com base no critério de otimização;
Fim

```

Fonte: elaborado pelo autor (2014).

### 5.3.2.3 Diversificação da Busca

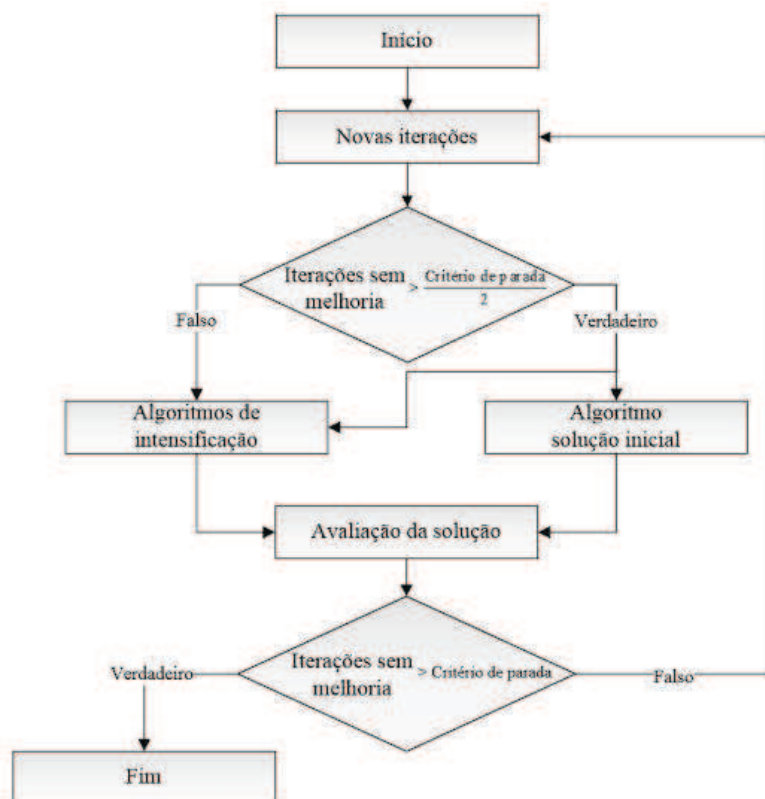
Com a finalidade de fugir de mínimos locais, o algoritmo utilizado busca perturbar a vizinhança local e realizar movimentos mais bruscos. Através dessa perturbação é possível alcançar regiões ainda não exploradas pela Busca Tabu. Ao total, são geradas 10 soluções de diversificação.

Diferentemente dos movimentos realizados pelo algoritmo de intensificação, onde apenas uma quantidade pequena de chapas é removida; no algoritmo de diversificação um número elevado de chapas é removido do processo visando gerar uma nova solução. O número de chapas removidas pode variar desde a metade das chapas até todas as chapas da solução candidata.

Os algoritmos utilizados para diversificação são apenas utilizados quando a Busca Tabu atinge metade do número de iterações máximas estipuladas como critério de parada. Na implementação proposta, o critério de parada utilizado é de 750 iterações sem melhoria. Sendo este número de iterações escolhido com base em diversas execuções do sistema proposto. Essa quantidade de iterações faz com o sistema não execute um número excessivo de iterações e que o processo seja finalizado em tempo hábil, porém, ainda execute iterações suficientes para que um resultado ótimo ou quase ótimo seja obtido. Sendo assim, a partir de 375 iterações sem melhoria os algoritmos de diversificação serão utilizados.

As 5 primeiras soluções que irão compor a vizinhança são geradas usando o mesmo algoritmo utilizado pela intensificação da busca; todavia o número de chapas removidas é superior. A re-seleção das chapas que serão reinseridas no processo também acontece de forma similar e é realizada com base no critério de otimização selecionado. As 5 soluções restantes passam por um processo mais complexo. De forma similar a Busca Tabu proposta por Gómez et al. (2014) e Ferrary et al. (2014), a diversificação acontece utilizando processamento paralelo; permitindo que algoritmos de maior complexidade sejam utilizados. Utilizando tais recursos, torna-se possível o uso de algoritmos de maior complexidade já que o tempo de processamento pode ser reduzido. A Figura 31 demonstra a arquitetura da Busca Tabu implementada.

**Figura 31 - Arquitetura da Busca Tabu proposta**





Além disso, por analisar de forma mais profunda as chapas que podem ser inseridas no processo buscando a chapa com melhor área, o resultado obtido não irá apenas substituir as chapas removidas de forma aleatória. Essa análise mais criteriosa faz com que uma nova solução, potencialmente próxima a um mínimo local ou até mesmo global, seja encontrada (BLACK, 2014).

O processamento paralelo acontece de forma que as tarefas sejam separadas e processadas de forma independente. Quando utilizando o algoritmo de diversificação, as tarefas são divididas em duas, sendo uma tarefa responsável por processar os algoritmos de baixa complexidade e outra tarefa responsável por criar apenas as soluções utilizando o algoritmo Guloso proposto na etapa de criação da solução inicial.

Por se tratar de um processamento paralelo, é importante que a sincronia do processamento seja mantida. Por esse motivo, no final de cada iteração é necessário que ambas as tarefas sejam finalizadas. A análise da vizinhança criada só ocorre após o fim do processamento das duas tarefas.

#### 5.3.2.4 Avaliação da Função Objetivo

Ao final de cada iteração da Busca Tabu, ocorre a avaliação da função objetivo. Essa avaliação tem por finalidade determinar qual a melhor solução encontrada até o momento além de determinar qual será a solução candidata utilizada na próxima iteração da Busca Tabu; ou seja, determinar o próximo movimento.

A avaliação inicia através da escolha do melhor resultado na vizinhança criada. Essa escolha varia de acordo com o critério de otimização escolhido pelo usuário. A melhor chapa para o processo é determinada através das funções objetivo definidas nas seções 4.5.2.1, 4.5.2.2 e 4.5.2.3.

Uma vez que a melhor solução na vizinhança seja escolhida, é possível determinar se existe uma solução mais otimizada do que a melhor solução obtida até o momento. Caso nenhuma resposta na vizinhança seja superior, quando comparada a melhor solução já encontrada, o contador de iterações sem melhoria será acrescido em uma unidade. Caso exista uma resposta melhor, esse contador é reiniciado voltando a 0. Esse contador é responsável por determinar se serão ou não executados movimentos que diversificam as soluções geradas, assim como responsável por controlar se o critério de parada foi alcançado.

Finalizada a avaliação das soluções obtidas, é possível determinar o próximo movimento realizado. Esse movimento a ser realizado, em sua grande maioria, é em direção a melhor resposta encontrada já que esta guia o caminho para um resultado ótimo ou aproximadamente ótimo. Todavia, antes que o próximo movimento seja realizado nessa direção, um sorteio é realizado.

Nesse sorteio, um número aleatório inteiro entre 0 e 100 é gerado e, caso o mesmo seja superior a 94, o próximo movimento será realizado na direção do pior resultado encontrado nessa vizinhança. Essa pequena chance (5%) de realizar um movimento para uma solução que não é a ótima, faz com que mínimos locais sejam evitados.

Além disso, antes que uma nova iteração seja iniciada, é preciso garantir que o tamanho da lista Tabu, definida como 10% do número de chapas na lista, não seja excedido. Por esse motivo, caso esse limite seja excedido, a solução mais antiga é removida da lista

Tabu para que a solução candidata escolhida seja adicionada a essa lista. A Figura 32, mostra de forma simplificada, o algoritmo utilizado.

Figura 32 - Pseudocódigo do algoritmo (avaliação da função objetivo)

```

Inicialização
  Se critério de otimização = material então
    MelhorSoluçãoLocal := Selecciona melhor solução com base na área utilizada
  Fim se;
  Se critério de otimização = custo então
    MelhorSoluçãoLocal := Selecciona melhor solução com base no custo
  Fim se;
  Se critério de otimização = tempo então
    MelhorSoluçãoLocal := Selecciona melhor solução com no número de chapas
  Fim se;
  Se MelhorSoluçãoLocal < MelhorSoluçãoJáEncontrada então
    MelhorSoluçãoJáEncontrada := MelhorSoluçãoLocal
  Fim se;
  Sorteio = número aleatório entre 0 e 100
  Se Sorteio > 94 então
    SoluçãoCandidata = PiorSoluçãoLocal
  Senão
    SoluçãoCandidata = MelhorSoluçãoLocal
  Fim se;
  Atualiza Lista Tabu
Fim

```

Fonte: elaborado pelo autor (2014).

## 5.4 Sistema Especialista

O sistema especialista proposto pelo trabalho tem como seu objetivo principal realizar a seleção dos parâmetros de *nesting* com base no seu banco de regras. O processo de escolha é realizado através do sistema de regras criado especialmente para o sistema em análise. Por esse motivo, é necessário o uso das informações obtidas através das características das peças obtidas previamente, conforme demonstrado pela seção 5.2.

Ao longo dessa seção, são detalhadas as regras utilizadas para o desenvolvimento do sistema especialista, dividindo as regras em: parâmetros ligados a peça e parâmetros especiais da ferramenta de *nesting*. Além disso, são detalhadas as restrições de todos os parâmetros, restrições que, algumas vezes, fazem com que não seja viável a automação do cálculo de um determinado parâmetro.

Ao longo dessa seção também é apresentada a implementação que torna possível as melhorias e o aprendizado do sistema.

### 5.4.1 Regras

Embora seja uma adaptação do software Jboss Drools®, o Drools.NET traz consigo a grande variedade de recursos existentes no sistema original. A adaptação trás do sistema original toda a estrutura de regras e sintaxe. Assim como no sistema original, regras são identificadas por seu nome e possuem sua estrutura conforme ilustrado pela Figura 33.

Seguido do nome da regra, é definida a condição sobre a qual a regra deve ser utilizada. Conhecida no sistema original como condição “*Left Hand Side (LHS)*”, é responsável por armazenar uma ou mais condições para que a regra seja utilizada. Essa condição é identificada, de forma simplificada, através da palavra-chave “*when*”.

A descrição das condições é através de uma palavra chave, determinada como “Modelo” pela Figura 33, seguida da condição lógica a ser considerada. O modelo é o nome utilizado para definir o tipo de regra e torna possível agrupar diversas regras que abordam uma área específica.

Caso a condição imposta seja verdadeira, então o sistema atinge a ação que deverá ser realizada nesse caso específico. Conhecido como “*Right Hand Side (RHS)*” no sistema original, essa parte da regra define quais as modificações serão realizadas nas variáveis de processo em análise. Essa parte da regra é responsável por determinar quais ações serão tomadas, assim como determinar o comportamento do sistema especialista, como, por exemplo, determinar que todas as regras devem ser reavaliadas devido à mudança de uma ou mais variáveis de processo. Essa condição é identificada, de forma simplificada, através da palavra-chave “*then*”.

**Figura 33 - Sintaxe das regras (Drools.NET)**

```
rule "Nome da regra"
  when
    Modelo : Condição (LHS)
  then
    "Ações" (RHS)
  end
```

Fonte: elaborado pelo autor (2014).

A sintaxe exemplificada na Figura 33 é utilizada para a implementação do sistema de regras.

#### 5.4.1.1 Parâmetros Principais

Para que as regras sejam criadas, é importante saber quando e em quais circunstâncias cada parâmetro disponível deve ser utilizado. Ao longo deste capítulo serão apresentadas as circunstâncias favoráveis para cada parâmetro, assim como suas respectivas regras.

A primeira escolha no processo de *nesting* é o método que o mesmo utilizará. São oferecidos dois métodos: o primeiro é o método *True Shape*, que é utilizado em situações onde o formato da peça deve ser levado em consideração, não tolerando simplificações da peça. Embora seja mais complexo, este método é o mais indicado quando as peças submetidas ao processo de *nesting* possuem formato livre ou áreas internas.

Já o segundo método, chamado de retangular, é calculado de forma que um retângulo seja criado ao redor da peça e ao invés de analisar o formato da real da peça - tal retângulo será utilizado nos cálculos de *nesting*, simplificando o cálculo. Por esse motivo, toda e qualquer área interna dentro desse retângulo será ignorada, tornando esse método não

recomendável para situações onde a área interna da peça pode ser utilizada para alocar peças menores.

A Tabela 7, apresenta as regras utilizadas pelo sistema especialista para definir qual método será utilizado.

**Tabela 7 - Regras utilizadas para escolha do método de *nesting***

| Método de <i>Nesting</i> |                                                                                                                                                                          |                             |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| Nome da regra            | Descrição das regras                                                                                                                                                     | Método escolhido pela regra |
| Metodo.TrueShape         | Analisa se a peça possui geometrias ou caminhos de ferramenta com formatos livres ou arcos. Caso possua, dará preferência à escolha do método <i>True shape</i> .        | <i>True Shape</i>           |
| Metodo.Retangular        | Analisa se a peça possui é formada por apenas geometrias ou caminhos de ferramentas criados a partir de linhas retas.                                                    | Retangular                  |
| Metodo.AreaInterna       | Analisa se a peça possui ou não áreas internas. Caso possua, indica que o método <i>True Shape</i> seja utilizado já que o método retangular irá ignorar áreas internas. | <i>True Shape</i>           |

Fonte: elaborado pelo autor (2014).

Parte dos parâmetros descritos como “ligados à peça”, tais como a rotação e o espelhamento, são parâmetros que na maioria dos casos podem ser automatizados já que estão ligados diretamente ao tipo de material utilizado no processo de usinagem. Usualmente, peças submetidas ao processo de *nesting* em materiais uniformes (sem ranhuras, veios ou qualquer tipo de marca), podem ser rotacionadas ou espelhadas sem que a qualidade e resultado final sejam alterados. Um exemplo de matéria prima sem marcas é demonstrado pela Figura 34 (a), neste caso uma chapa de alumínio sem qualquer marca.

Todavia, em casos onde tais marcas existem, podemos inferir que esteticamente a peça ficará fora dos padrões esperados já que, na grande maioria das vezes, respeitam-se essas marcas. Um caso comum onde marcas no material devem ser respeitadas está no mercado de móveis de madeira, uma vez que chapas de madeira possuem sentido do veio definido. Neste caso, todas as peças dentro dessa chapa possuem a tendência de respeitar o sentido do veio e serem alocadas da chapa sem serem rotacionadas. Um exemplo de matéria prima com marcas é demonstrado pela Figura 34 (b), ilustrando uma chapa de madeira com veios no sentido vertical.

**Figura 34 - Amostra de material com e sem marcas**



(a)



(b)

Fonte: extraído da internet (a)<sup>2</sup> e (b)<sup>3</sup> (2014).

Por esse motivo, as regras aplicadas ao movimento de rotação e espelhamento aplicado ao sistema especialista são baseadas no material e através do material o usuário terá a liberdade de definir se o mesmo permite ou não a rotação do posicionamento das peças dentro da matéria prima. A Tabela 8 descreve as regras aplicadas à parametrização da rotação e espelhamento das peças.

**Tabela 8 - Regras aplicadas aos parâmetros da peça**

| Parâmetros da Peça       |                                                                                                                                                                                                                 |                            |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| Nome da Regra            | Descrição Das Regras                                                                                                                                                                                            | Ação Utilizada pela Regra  |
| Peça.PermiteRotação      | Considerando a rotação da peça dependente do tipo de material, essa regra define se a peça pode ser rotacionada com base no material escolhido.                                                                 | Rotação em qualquer angulo |
| Peça.PermiteEspelhamento | Considerando a rotação da peça dependente do tipo de material, essa regra define se a peça pode ser espelhada com base no material escolhido.                                                                   | Pode ser espelhada         |
| Peça.DefinirPequena      | Analisa o tamanho da peça em comparação com as demais peças sendo submetidas ao processo de <i>nesting</i> . A peça será considerada como pequena, caso a peça possua metade do tamanho médio das demais peças. | É considerada peça pequena |

Fonte: elaborado pelo autor (2014).

Ainda parte dos parâmetros principais, os parâmetros associados à peça não podem ser facilmente automatizados, já que possuem características únicas e que devem ser definidas pelo usuário. Características tais como o nome da peça, número de peças, número do lote e prioridade são algumas que devem ser informadas pelo usuário no momento em que as peças são escolhidas como parte do processo. Tais dados são considerados parte das informações de entrada do sistema e serão utilizadas no processo de tomada de decisão. Os parâmetros que não serão automatizados e necessitam a inserção de dados do usuário serão detalhados ao longo da seção 5.4.1.3.

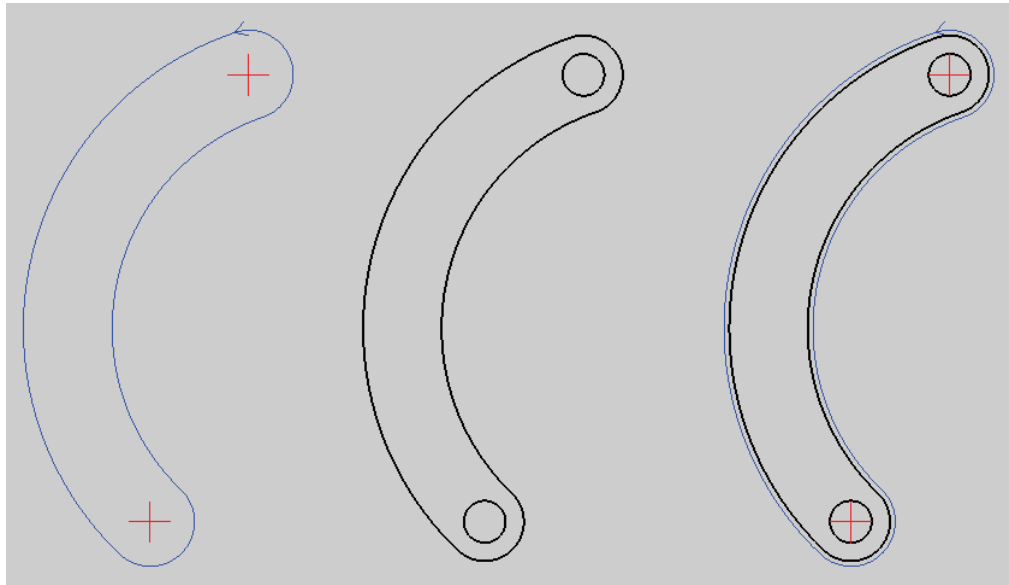
#### 5.4.1.2 Parâmetros Especiais

Uma vez determinados os parâmetros principais do processo, a próxima etapa é definir os parâmetros específicos da ferramenta de *nesting*. Por se tratar de uma aplicação desenvolvida para *softwares* CAM, são oferecidos três diferentes tipos de seleções para realizar a otimização: a primeira opção possibilita a seleção apenas de caminhos de ferramenta; a segunda escolha possibilita a seleção de geometrias; a terceira opção possibilita selecionar geometrias e seus respectivos caminhos de ferramenta. Esses três tipos distintos de escolha são demonstrados pela Figura 35.

<sup>2</sup> Disponível em <http://solucionesmobiariasmetalicas.com/portafolio.html>

<sup>3</sup> Disponível em <http://dicasparadecorar.com/condominios/texturas-de-madeira>

**Figura 35 - Diferentes tipos de seleção de peça**



Fonte: elaborado pelo autor (2014).

O tipo de escolha é importante para a seleção dos parâmetros especiais, porque diversos desses parâmetros se aplicam a apenas um tipo de seleção; sendo assim possível simplificar o processo de escolha através da redução do número de variáveis de escolha. Por esse motivo, na inicialização da escolha dos parâmetros o tipo de seleção é a primeira característica levada em consideração.

A restrição de parâmetros acontece principalmente quando o usuário faz a escolha de utilizar a ferramenta de *nesting* para otimizar apenas geometrias. Isso ocorre porque aproximadamente 1/3 dos parâmetros especiais realizam mudanças ou otimização em caminhos de ferramentas, comportamento esperado por se tratar de um *software* CAM. Na Tabela 9 são listados os parâmetros que não devem ser utilizados na ausência de caminhos de ferramenta. Tendo em mente essa lista de variáveis torna-se possível, através de uma variável booleana, controlar quando essas regras serão ignoradas.

**Tabela 9 - Parâmetros que necessitam caminhos de ferramenta**

| Nome do Parâmetro                          |
|--------------------------------------------|
| Ligar peças ( <i>Bridge Nesting</i> )      |
| Cortar peças pequenas primeiro             |
| Cortar peça inteira                        |
| Furar e então realizar operações internas  |
| Distância extra para entrada da ferramenta |
| Suprime verificação de contornos           |
| Ordenar peça por peça                      |

Fonte: elaborado pelo autor (2014).

Algumas restrições que tornam possível simplificar o processo de escolha estão relacionadas ao número de diferentes peças que estejam sendo submetidas ao processo. Nesse caso, existem tanto parâmetros que só podem ser utilizados quando existem peças distintas,

porém, existem parâmetros que só funcionam quando considerada uma única peça. Seja por um motivo lógico, ou seja, por uma restrição do *software*, é importante integrar essa restrição ao sistema especialista. A Tabela 10 mostra as regras com essa restrição, assim como a sua respectiva restrição.

**Tabela 10 - Regras com restrições quanto ao número de peças**

| Nome do Parâmetro                 | Número de Peças |
|-----------------------------------|-----------------|
| Processar peças pequenas primeiro | > 1             |
| Cortar peças pequenas primeiro    | > 1             |
| Repetir linhas e colunas          | = 1             |

Fonte: elaborado pelo autor (2014).

Tendo em mente essas restrições, e o benefício que as mesmas trazem para a simplificação do processo de escolha, é possível iniciar o uso das regras que definirão quando cada parâmetro especial será utilizado. Ao longo da Tabela 12 são descritos parâmetros que serão avaliados pelo sistema especialista, assim como suas respectivas regras.

Para efeito de identificação dos itens da tabela Tabela 12, a Tabela 11 atribui um número de identificação para cada parâmetro; de modo a facilitar o entendimento das regras utilizadas.

**Tabela 11 - Identificação de parâmetros**

| Nome do parâmetro                       | ID |
|-----------------------------------------|----|
| Nesting.Tempo                           | 1  |
| Nesting.CortarPequenasPrimeiro          | 2  |
| Nesting.OrdenarPeça                     | 3  |
| Nesting.UsarDistanciaEntrePeças         | 4  |
| Nesting.RepetirLinhasEColunas1          | 5  |
| Nesting.RepetirLinhasEColunas2          | 6  |
| Nesting.TentarRotacionarPecas           | 7  |
| Nesting.DistanciaExtraEntrada           | 8  |
| Nesting.MinimizarTrocasFerramenta1      | 9  |
| Nesting.MinimizarTrocasFerramenta2      | 10 |
| Nesting.FurarEOperaçõesInternasPrimeiro | 11 |
| Nesting.CortarPecaInteira               | 12 |
| Nesting.BridgeNesting1                  | 13 |
| Nesting.BridgeNesting2                  | 14 |
| Nesting.BridgeNesting3                  | 15 |
| Nesting.ProcessarPequenasPrimeiro       | 16 |
| Nesting.SuprimeVerificaçãoContornos     | 17 |

Fonte: elaborado pelo autor (2014).

Tabela 12 - Regras utilizadas para selecionar parâmetros especiais

| ID | Descrição das Regras                                                                                                                                                                                                                                                                                                                                                     | Ação da Regra                                                                                                                                                                                                         | Restrições                                                                                |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| 1  | O tempo para cada chapa é definido de acordo com o nível de otimização selecionado pelo usuário. O tempo mínimo para cada chapa é 10 segundos, quando utilizando nível de otimização 1. Já o tempo máximo é 100 segundos, quando utilizando nível de otimização 10.                                                                                                      | Seleciona o tempo com base no nível de otimização solicitado                                                                                                                                                          | Nenhuma                                                                                   |
| 2  | Embora o uso dessa função não auxilie no número de peças que serão encaixadas, quando o número de peças pequenas é superior ao número de peças grande, é sempre recomendável que as peças pequenas sejam processadas e cortadas primeiro. Por esse motivo, sempre que o número de peças pequenas representa 30% ou mais das peças totais, essa opção é ativa.            | Utiliza o parâmetro " <b>Cortar peças pequenas primeiro</b> " se 30% das peças ou mais são pequenas                                                                                                                   | Deve existir ao menos 1 caminho de ferramenta                                             |
| 3  | Essa opção é utilizada sempre que o método de otimização escolhido é o tempo. Essa opção faz que o corte seja otimizado e faz com que lotes de peças iguais sejam finalizados por etapas. Isso faz com que apenas que esse lote seja finalizado, o usuário possa remover essas peças e prosseguir com os demais processos de manufatura que essas peças serão submetidas | Seleciona o parâmetro " <b>Ordenar peça por peça</b> " quando o critério de otimização é "tempo"                                                                                                                      | Deve existir ao menos 1 caminho de ferramenta                                             |
| 4  | Essa opção é utilizada sempre que o método de otimização escolhido é o material. Todavia, a mesma só deve ser utilizada quando existe espaço suficiente para a ferramenta entre as peças. Por isso a regra define que a ferramenta possua diâmetro igual ou menor do que a distância entre peça definida pelo usuário.                                                   | Seleciona o parâmetro " <b>Utilizar distância entre peças</b> " quando o critério de otimização é a redução de material. Todavia, somente quando a ferramenta possui diâmetro compatível com a distância entre peças. | Nenhuma                                                                                   |
| 5  | Quando realizando corte com serra, utilizar esta opção. Assim, é possível cortar uma seção inteira de uma só vez utilizando linhas em comum. Só pode ser utilizado quando cortando um único tipo de peça.                                                                                                                                                                | Seleciona o parâmetro " <b>Repetir primeira linha e coluna</b> " quando realizando corte com serra e apenas uma peça está sendo processada                                                                            | Função só pode ser utilizada quando processando apenas um tipo de peça                    |
| 6  | Utilizar quando otimizando por tempo ou custo. Não utilizar quando critério escolhido é o material, já que a área utilizada para manter linhas e colunas alinhadas é geralmente superior quando não alinhando as mesmas. Para qualquer tipo de corte, utilizar quando a rotação não é permitida. Só pode ser utilizado quando cortando um único tipo de peça.            | Seleciona o parâmetro " <b>Repetir primeira linha e coluna</b> " quando o critério de otimização é o tempo ou custo e se a rotação não é permitida                                                                    | Função só pode ser utilizada quando processando apenas um tipo de peça                    |
| 7  | Utilizado quando a rotação de uma ou mais peças é permitida e nível de otimização é 5 ou superior                                                                                                                                                                                                                                                                        | Seleciona o parâmetro " <b>Tentar rotacionar todas as peças</b> " quando as peças podem ser rotacionadas e o nível de otimização é 5 ou superior                                                                      | Função só pode ser utilizada quando processando apenas um tipo de peça                    |
| 8  | Essa função é utilizada sempre que existem caminhos de ferramenta com entrada/saída de ferramenta. No entanto essa opção só pode ser utilizada quando existe espaço suficiente para que esse movimento seja realizado. Essa opção só é utilizada quando o raio da maior ferramenta utilizada é igual ou inferior à distância mínima entre peças definida pelo usuário    | Seleciona o parâmetro " <b>Distância extra para entrada da ferramenta</b> " quando caminhos de ferramenta possuem entrada/saída e existe distância suficiente entre as peças para que uma entrada extra seja definida | Caminho de ferramenta deve possuir entrada ou saída de caminhos de ferramentas adicionais |
| 9  | Sempre utilizar quando otimizando tempo ou custo                                                                                                                                                                                                                                                                                                                         | Seleciona o parâmetro " <b>Minimizar troca de ferramentas</b> " quando o critério de otimização selecionado é tempo ou custo                                                                                          | Só pode ser utilizados quando existem operações internas ou ciclo de furação              |



Tabela 12 - Regras utilizadas para selecionar parâmetros especiais (continuação)

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                      |                                                              |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| 10 | Não utilizar quando Cortar parte inteira ou Agrupar peças separadamente, pois quando essas opções são utilizadas é preferível que a ordem em que os cortes são realizados permaneça sem alterações                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Torna proibitivo o uso de "Minimizar troca de ferramentas" quando as opções "Cortar parte inteira" ou "Agrupar peças separadamente" estão sendo utilizadas                                                           | Deve existir ao menos 1 caminho de ferramenta                |
| 11 | Em geral, sempre que existam caminhos de ferramentas internos ou ciclos de furação, é recomendável o uso desse parâmetro. Ele irá fazer com que todas as operações internas sejam realizadas primeiro; fazendo com que a peça não seja removida do material bruto até que todas as operações de usinagem sejam realizadas. Sempre será mais eficiente realizar as operações internas na chapa intacta, ao invés de em uma chapa fragmentada e com diversas peças já destacadas, pois assim o acabamento das peças é aprimorado. Além disso, a fixação a vácuo possui dificuldade para manter peças pequenas presas à mesa. Não pode ser utilizado quando "Cortar parte inteira" ou "Agrupar peças separadamente" estão sendo utilizados | Seleciona o parâmetro "Furar e então realizar operações internas" quando as opções "Cortar parte inteira" ou "Agrupar peças separadamente" estão sendo utilizadas                                                    | Deve existir ao menos 1 caminho de ferramenta                |
| 12 | Utilizado sempre que o número de peças pequenas for superior o número de peças grandes. Todavia, por restrição do <i>software</i> , esse recurso não deve ser utilizado quando a opção "furar e realizar operações internas primeiro". Além disso, pode aumentar o tempo de usinagem, portanto nunca deve ser utilizado quando critério de otimização selecionado for "tempo".                                                                                                                                                                                                                                                                                                                                                          | Seleciona o parâmetro "Cortar peça inteira" quando as opções "Cortar parte inteira" ou "Agrupar peças separadamente" não estão sendo utilizadas. Além disso, nunca é utilizado quando otimizando o tempo de processo | Deve existir ao menos 1 caminho de ferramenta                |
| 13 | Utilizar quando buscando otimizar o tempo, exceto se cortando com serra                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Seleciona o parâmetro " <i>Bridge Nesting</i> " quando critério de otimização seleciona é tempo. Não utilizada quando realizando cortes com serra                                                                    | Só pode ser utilizado quando existem caminhos de ferramentas |
| 14 | Independente do tipo de otimização, caso usando apenas uma ferramenta e realizando apenas cortes de área internas; utilizar esse recurso                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Seleciona o parâmetro " <i>Bridge Nesting</i> " apenas uma ferramenta está sendo utilizada e apenas cortes externos são realizados                                                                                   | Só pode ser utilizado quando existem caminhos de ferramentas |
| 15 | Utilizar sempre que cortando através do <i>Profiling</i> (Laser, Plasma, Waterjet, Wire)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Seleciona o parâmetro " <i>Bridge Nesting</i> " quando utilizando corte a Laser, Plasma, Waterjet, Wire.                                                                                                             | Só pode ser utilizado quando existem caminhos de ferramentas |
| 16 | O uso dessa função, de forma geral, não auxilia no número de peças que serão encaixadas. Quando o número de peças pequenas é superior ao número de peças grande é sempre recomendável que as peças pequenas sejam processadas e cortadas antes. Por esse motivo, sempre que o número de peças pequenas representa 30% ou mais das peças totais, essa opção é ativa                                                                                                                                                                                                                                                                                                                                                                      | Seleciona o parâmetro "Processar peças pequenas primeiro" quando o número de peças pequenas representa mais do que 30% das peças                                                                                     | Devem existir no mínimo 2 peças diferentes                   |
| 17 | Verificação de contornos é apenas desligada quando nível de otimização é menor que 5 (escolhido conforme demonstrado na seção 5.1 )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Seleciona o parâmetro "Suprime verificação de contornos" quando nível de otimização solicitado é 5 ou menor                                                                                                          | Deve existir ao menos 1 caminho de ferramenta                |

Fonte: elaborado pelo autor (2014).

Assim como nos parâmetros principais, após a análise aprofundada de cada parâmetro, foi observado que alguns deles não podem ser selecionados pelo sistema especialista e devem ser selecionados pelo usuário. As justificativas para a escolha de remover esses parâmetros do sistema especialista são abordadas ao longo da próxima seção.

#### 5.4.1.3 Parâmetros Selecionados pelo Usuário

Embora através da implementação proposta têm-se como objetivo elaborar um sistema o mais automatizado possível, em um sistema real o número de processos de manufatura que podem ser submetidos ao sistema especialista pode variar drasticamente. Por esse motivo, alguns dos parâmetros não serão automatizados intencionalmente, por razões que são abordadas ao longo dessa seção.

Todavia, é importante ressaltar que todas as opções continuam disponíveis para o usuário através da interface do SAD proposto. A não supressão de opções é importante, pois assim, a ferramenta automatizada não possuirá perda de funcionalidade quando comparada a ferramenta original.

Quando lidando com as informações das peças, deve-se levar em conta que as mesmas são consideradas a entrada do sistema e por isso dificilmente podem ser automatizadas, já que estas serão utilizadas no processo de tomada de decisão.

Parâmetros tais como: número de peças, nome da peça, prioridade para ser produzida e número de lote são características únicas, estipuladas pelo processo de produção ou produto final almejado. Sendo assim, esses parâmetros não deverão ser modificados ou automatizados pelo sistema, tal como listados na Tabela 13.

**Tabela 13 - Parâmetros de peça não automatizados**

| Parâmetros da Peça |                                                                                                                                                                                                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parâmetro          | Descrição                                                                                                                                                                                                                                                             |
| Nome da peça       | Nome é uma característica única da peça e não deve ser modificado pelo sistema. Por padrão a peça, possui o mesmo nome do modelo importado do sistema CAD.                                                                                                            |
| Número de peças    | Número de peças está ligado diretamente ao produto final esperado. Por esse motivo, o número de peças deve ser especificado pelo usuário.                                                                                                                             |
| Prioridade         | Prioridade das peças em relação às outras também é uma variável que depende do processo. Partes do processo podem facilitar o processo de montagem e usinagem caso sejam produzidas antes das demais. Por isso, cabe ao usuário determinar a prioridade de cada peça. |
| Número do lote     | A definição se as peças devem ser produzidas em lotes também é uma característica ligada ao processo de produção e não pode ser facilmente automatizada.                                                                                                              |

Fonte: elaborado pelo autor (2014).

De forma similar aos parâmetros ligados à peça, alguns dos parâmetros especiais não deverão ser submetidos ao sistema especialista. Esses motivos vão desde opções que não afetam o resultado (prevenir atualização de tela) até peculiaridades do processo de manufatura (*Onion Skin*). Os parâmetros que não serão incluídos no processo de decisão do sistema especialista são demonstrados na Tabela 14.

Tabela 14 - Parâmetros especiais escolhidos pelo usuário

| Parametrização dos Processos de <i>Nesting</i>         |                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Borda ou porção da chapa para começar o <i>nesting</i> | Na grande maioria dos testes, o início não realizou uma melhor performance no sistema de <i>nesting</i> , variando apenas o posicionamento das peças. Por esse motivo a escolha desse parâmetro depende principalmente das preferências do usuário assim como a máquina CNC sendo utilizada                 |
| Distância mínima entre peças                           | A distância mínima entre peças deve ser definida pelo projeto e não pelo sistema especialista. Caso o usuário opte por não definir uma distância mínima, o valor utilizado é 0 o que permite que peças possuam caminhos de ferramenta com corte em comum                                                    |
| Distância até a borda da chapa                         | Assim como a distância mínima entre peças, a distância entre a peça e a borda da chapa é uma escolha do usuário. Em muitos casos depende do usuário utilizar ou não a borda do material como parte do processo                                                                                              |
| <i>Onion Skin</i> em peças pequenas                    | Quando utilizado, esse processo cria um corte extra na peça com uma profundidade ligeiramente menor do que a espessura da chapa, deixando assim uma fina camada (conhecida como <i>Onion Skin</i> ) que prende o material a chapa. Um segundo corte é realizado posteriormente para liberar a peça da chapa |
| Prevenir atualização de tela                           | Possibilita esconder os resultados sendo processados, mostrando assim apenas o resultado final ao invés de atualizar a tela ao final do processamento de cada chapa                                                                                                                                         |
| Remover grupos                                         | O parâmetro não afeta o processo de forma alguma. Por ser um parâmetro de escolha pessoal, e não modificar o resultado obtido, é oferecido ao usuário usar ou não utilizar essa opção através da interface com o usuário                                                                                    |
| Agrupar peças separadamente                            | Similar à opção de remover grupos, essa opção não afeta o processo e cabe ao usuário determinar ou não se quer que as geometrias e caminhos de ferramenta sejam colocados dentro de um único grupo                                                                                                          |
| Prevenir <i>nesting</i> em áreas internas              | Não utilizar as áreas internas das peças causa perda excessiva e sempre que possível será utilizado. Todavia, é possível que por motivos específicos o usuário não deseje utilizar essas áreas. Por esse motivo essa opção continua disponível através da interface com o usuário                           |

Fonte: elaborado pelo autor (2014).

#### 5.4.1.4 Módulo de Aprendizado

Ao final do processo completo do SAD, onde a resposta obtida pelo sistema é demonstrada ao usuário; caso o usuário rejeite o resultado, por qualquer que seja o motivo, é possível para o mesmo auxiliar no processo de aprendizado do sistema. Deste modo, o usuário tem a oportunidade de criar uma nova regra, sendo que esta será integrada ao sistema. Sendo assim, o sistema exerce o conhecimento obtido através das definições inseridas manualmente pelo usuário.

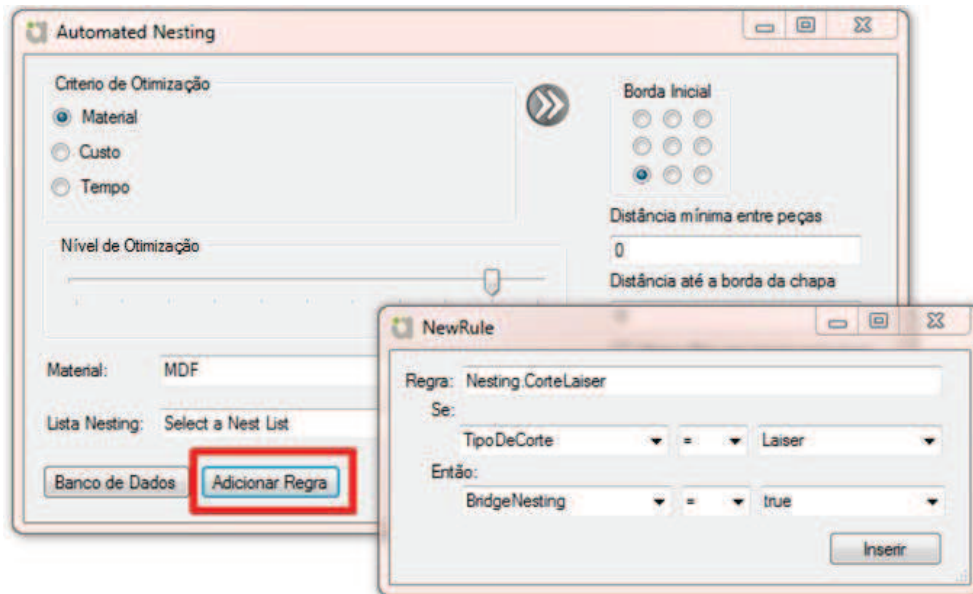
Essa opção de escolha tem por objetivo aprimorar o sistema para qualquer tipo de usuário, uma vez que, após diversas execuções do *software* o mesmo poderá inserir características peculiares ao seu processo produtivo. Isso é importante devido à grande variedade de processos aos quais a ferramenta de *nesting* se aplica.

Para tornar o processo de inserção de novas regras algo simplificado para o usuário, a inserção de novas regras é realizada através de uma caixa de diálogo. Nessa caixa de diálogo, são apresentadas todas as características que podem ser obtidas através do reconhecimento de características das peças.

Utilizando o mesmo formato de regra apresentado na seção 5.4.1, o diálogo tem como principal objetivo possibilitar ao usuário informar uma condição na qual a regra deve ser utilizada (“*when*”) e qual o parâmetro que deverá ser utilizado quando essa condição for verdadeira (“*then*”). A mesma lógica pode ser utilizada para evitar que parâmetros sejam

utilizados. A Figura 36 demonstra o diálogo apresentado para o usuário inserir uma nova regra para o banco de regras.

Figura 36 - Diálogo de inserção de novas regras



Fonte: elaborado pelo autor (2014).

É importante ressaltar que é possível, a qualquer momento, inserir novas regras através da interface com o usuário. Todavia, o ato de rejeitar um resultado proposto pelo sistema de apoio à decisão dá indícios fortes de que alguma decisão não foi tomada de forma correta ao longo do processo; sendo este o momento mais indicado para que essa falha seja corrigida pelo usuário, fazendo com que o sistema aprenda com as falhas.

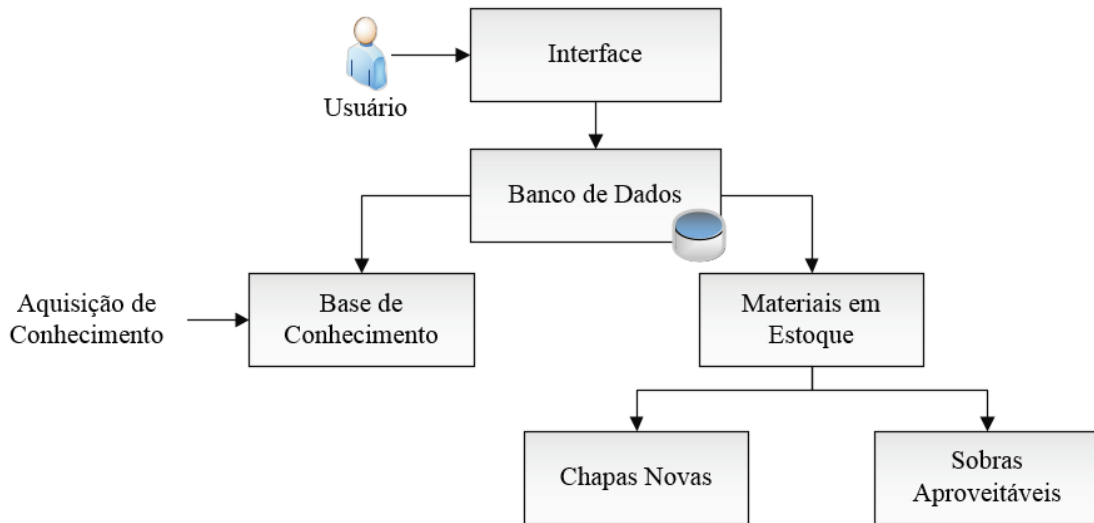
## 5.5 Banco de Dados

Para que o sistema continue operante e realizando a sua função principal, apoiando o usuário na tomada de decisão, é necessário que o banco de dados esteja sempre atualizado. Para isso, o banco de dados é responsável por manter todas as informações que serão utilizadas no processo pelo SAD.

Implementado utilizando o gerenciador de banco de dados *SQL Server*®, desenvolvido pela Microsoft, o banco de dados do SAD proposto é responsável por armazenar todo o conhecimento obtido pelo sistema ao longo das execuções do mesmo. Além disso, o banco de dados é responsável por manter o estoque atual de chapas que podem ser utilizadas no processo.

Em sua arquitetura, o banco de dados implementado é dividido em dois conjuntos. O primeiro conjunto armazena a base de conhecimento e nessa estão contidas todas as regras e informações sobre os processos a serem utilizados pelo SAD. Já o segundo conjunto armazena o estoque de chapas que podem ser utilizadas pelo usuário. A Figura 37 apresenta a arquitetura do banco de dados implementado.

Figura 37 - Arquitetura do banco de dados



Fonte: elaborado pelo autor (2014).

A base de conhecimento armazena as informações referentes ao processo e regras que são utilizadas para auxiliar no processo de decisão. Por esse motivo, o módulo de aquisição de conhecimento exerce forte influência na base de conhecimento. O módulo de aquisição de conhecimento é responsável por manter a base de conhecimento atualizada e inserir novas regras à base de conhecimento sempre que solicitado pelo usuário.

Já o estoque de chapas é responsável por armazenar toda e qualquer matéria prima que poderá ser utilizada no processo de *nesting*. Dentro do banco de dados, essas chapas são divididas em dois subgrupos. O primeiro grupo são as chapas novas e que nunca foram submetidas a qualquer processo de manufatura, definidas como “normais”. Já o segundo grupo são chapas extraídas da sucata, ou seja, sobras aproveitáveis que poderiam ser utilizadas novamente no processo. Para cada chapa, informações tais como: nome, tipo de material, quantidade, espessura, comprimento, largura, área, custo e comentários do usuário são armazenados no banco de dados. São informações que o SAD utiliza para realizar a seleção automatizada de matéria prima.

Quanto à manutenção do banco de dados, a comunicação do usuário com o banco de dados não acontece de forma direta. Para que o sistema seja de fácil uso para o usuário, toda e qualquer alteração é realizada através da interface com o usuário. Essas interfaces foram descritas ao longo da seção 5.1.

## 5.6 Extração das Sobras Aproveitáveis

Ao longo dessa seção, são apresentados detalhes da implementação realizada para que seja possível o reconhecimento e extração das sobras aproveitáveis, geradas pelo sistema de apoio a decisão. São descritos ao longo da seção, a etapa de extração do perfil que representa a sucata, assim como o método avaliativo para determinar se a sucata é ou não é aproveitável.

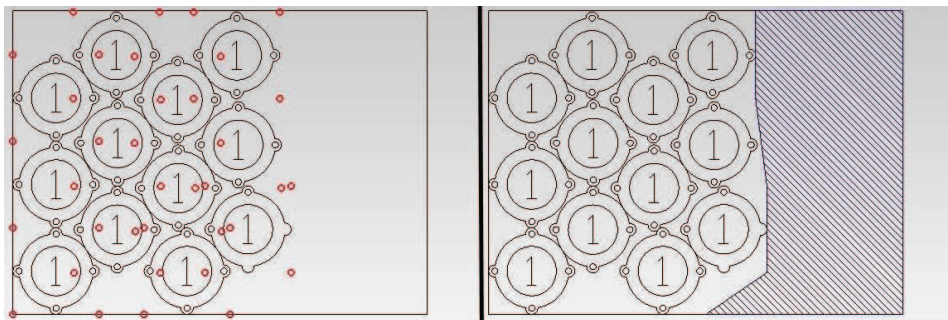
### 5.6.1 Extração do Perfil das Sobras

Seguindo a aplicação proposta por Ferrary et al. (2013), dois métodos de extração das sobras aproveitáveis são utilizados. Esses métodos são determinados ao longo do trabalho como: método Retangular e método de Varredura. Embora ambos os métodos possuam o mesmo objetivo, a grande diferença entre ambos está na forma como a sucata é obtida.

No método Retangular, peças complexas são colocadas dentro de retângulos de medidas semelhantes a da mesma, isso acontece para que seja possível reduzir a complexidade do processo de análise dessas peças. Uma vez que esse retângulo é criado é possível para o algoritmo utilizar apenas os 4 vértices do retângulo para determinar o perfil da peça ao invés de diversos vértices da peça original. Através dos vértices desses retângulos o algoritmo encontra apenas os pontos localizados entre as peças e a área onde a sucata está localizada. Através desses pontos, o algoritmo então desenha o perfil que representa a sucata.

Por possuir um algoritmo que simplifica o formato das peças, consecutivamente o resultado também é simplificado fazendo com que a representação do resultado seja uma aproximação do perfil real da sucata gerada pelo processo. A Figura 38 demonstra os pontos obtidos pelo algoritmo assim como o respectivo perfil de sucata gerado, demonstrado na figura pela área achurada.

**Figura 38 - Extração das sobras aproveitáveis através do método Retangular**



Fonte: Adaptado de Ferrary et al. (2013).

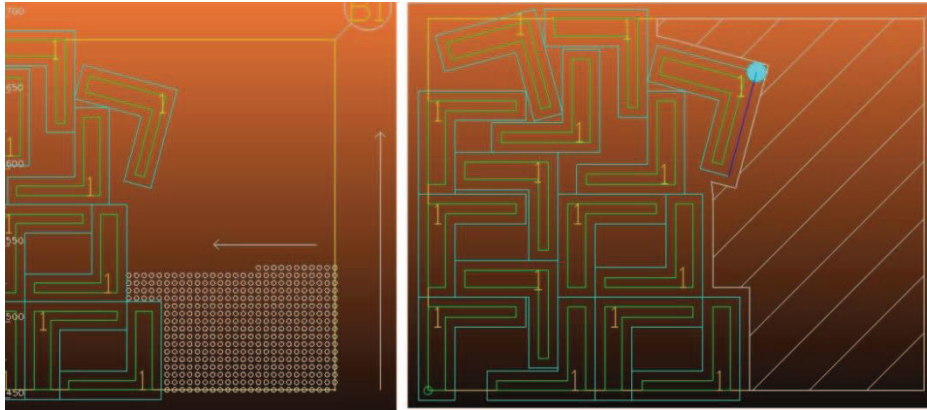
No método de varredura, diferentemente do método Retangular, o algoritmo não irá simplificar o formato das peças e irá considerar toda e qualquer geometria utilizada para criar as peças pelo *software* CAD, assim como todos os caminhos de ferramentas atribuídos a essas peças. Esse método possui um algoritmo que realiza uma varredura em toda a chapa, com base em uma tolerância, e armazena os pontos que determinam a fronteira entre as peças e a sucata gerada pelo processo.

Por ser uma análise mais complexa, o algoritmo cria segmentos dessas geometrias e analisa os pontos iniciais e finais dos mesmos, criando assim um grande número de pontos que são utilizados como referência para determinar o perfil da sucata. Tendo os pontos de referência definidos, o algoritmo inicia uma busca ao longo de toda a chapa, iniciando em uma das extremidades da peça o algoritmo realiza pequenos passos em direção das peças acomodadas na chapa, realizando uma busca longitudinal (eixo X) ao longo da peça. Esses passos são realizados até que uma colisão com uma peça seja detectada e, quando isso ocorre, esse ponto será utilizado como um dos pontos responsáveis por determinar o perfil da sucata gerada pelo processo.

Essa análise de colisão com as peças é realizada através do comparativo do ponto atual com os pontos extraídos das geometrias e caminhos de ferramenta seccionados previamente

pelo algoritmo. Uma vez que esse ponto seja encontrado é possível realizar um passo na direção Y e repetir a busca por um novo ponto que colida com uma peça. Esse processo é repetido até que toda a chapa seja analisada. A Figura 39 demonstra o funcionamento desse algoritmo.

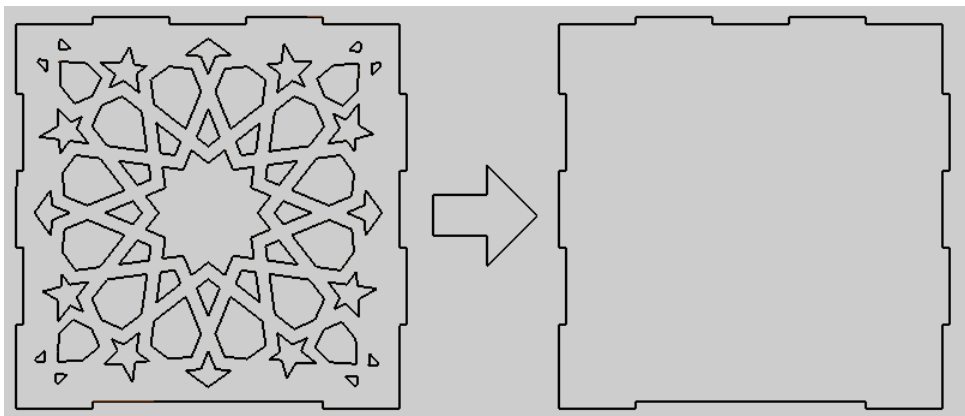
**Figura 39 - Extração das sobras aproveitáveis através do método de Varredura**



Fonte: Adaptado de Ferrary et al. (2013).

Como melhorias do processo, Ferrary et al. (2013) aponta que, para tornar mais amigável e eficaz a experiência do usuário quando utilizando a ferramenta desenvolvida, a implementação dos métodos propostos deveria ser realizada utilizando plataforma .NET, abandonando a antiga plataforma VBA. Levando em consideração essa proposta de melhoria, assim como o sistema especialista, o módulo responsável por extrair o perfil que representa as sobras aproveitáveis também foi desenvolvido utilizando linguagem de programação C#. Além de possuir recursos mais modernos, isso evita um dos principais problemas da implementação original: a ocorrência constante de problemas de *overflow* dependendo da tolerância utilizada no processo. O controle do *overflow* é possível através da classe *OverflowException* que possibilita que erros desse tipo sejam diagnosticados e tratados de forma eficiente. Além disso, a nova implementação simplifica o processo de análise das peças utilizado na implementação original. Ao invés de segmentar geometrias e caminhos de ferramenta em diversos pedaços, a implementação realizada nesse estudo coleta apenas as informações referente à geometria ou caminho de ferramenta externa de cada peça, ignorando assim todas as geometrias e caminhos de ferramentas internos. A Figura 40 demonstra, sucessivamente, a peça sendo analisada e a sua versão simplificada utilizada pelo algoritmo.

**Figura 40 - Simplificação de peça**

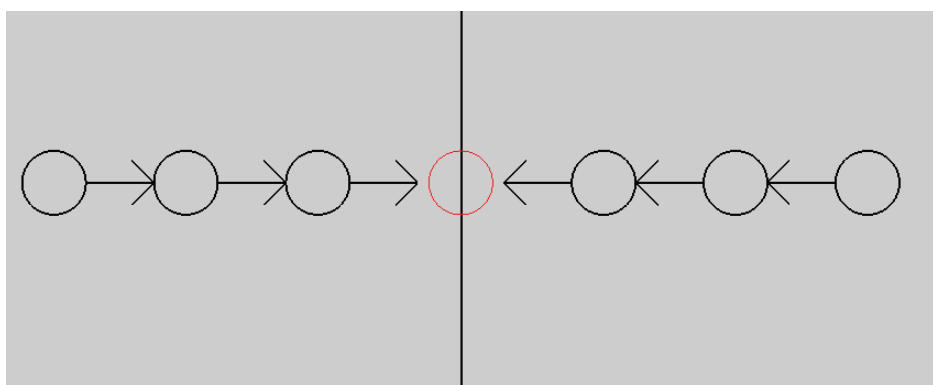


Fonte: elaborado pelo autor (2014).

Essa simplificação é possível já que, mesmo no algoritmo original, a parte externa da peça sempre será a primeira a coincidir com um ponto durante o processo de varredura. Além disso, na implementação proposta por esse estudo, a verificação que antes ocorria através da análise de todos os pontos extraídos dos segmentos criados, agora é realizada simplesmente através da verificação se o ponto está dentro de alguma das geometrias que representam o perfil externo das peças em análise. Esse método de análise simplificado torna possível que a etapa de segmentação das geometrias e caminhos de ferramenta sejam completamente removidos do processo, não criando a grande quantidade de pontos responsável pelos problemas de *overflow* citados anteriormente.

Todavia, uma vez que não há interesse em um ponto interior da peça; assim que o algoritmo encontra o ponto dentro da peça o mesmo retrocede um passo e então, realiza uma busca fina entre esses dois pontos pelo exato ponto que colida com a geometria. Essa busca fina é realizada através de diversas tentativas onde a distância entre os dois pontos é gradativamente sendo reduzida até que os dois pontos estejam dentro ou fora da peça. A Figura 41 representa graficamente como a busca fina é realizada.

Figura 41 - Busca fina sendo realizada



Fonte: elaborado pelo autor (2014).

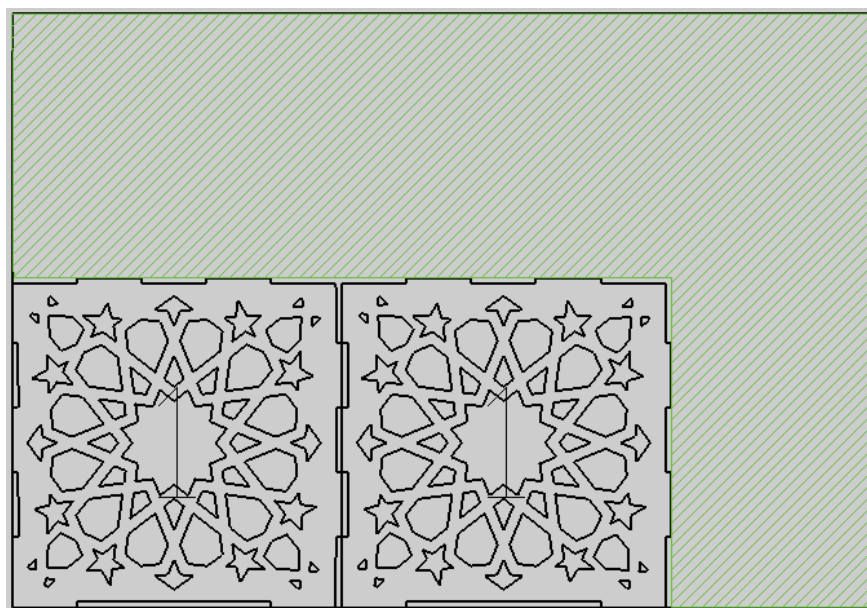
Uma vez que esse ponto seja obtido, é importante determinar se o mesmo é uma geometria ou um caminho de ferramenta, já que para caminhos de ferramenta é preciso levar em consideração que o mesmo representa o centro de uma ferramenta cilíndrica. Sendo assim, a ferramenta tem a necessidade de retrair, em passos, a quantidade equivalente ao raio da ferramenta para que a sucata seja corretamente representada.

Outra melhoria, em comparação à implementação inicial, é a escolha do método antes realizado de forma direta pelo usuário. Todavia, já que o SAD proposto reconhece diversas características do processo, é possível também determinar qual o melhor método a ser utilizado para realizar a extração do perfil que representa as sobras do processo. A decisão do método é realizada com base no tipo de *nesting* que foi realizado. Quando o método de *nesting* escolhido foi o retangular, então a extração da sucata também será realizada através do método retangular. Consequentemente, se o método de *nesting True Shape*, então o método de varredura é utilizado.

Ao final do processo, caso seja considerada uma sobra aproveitável, o perfil da sucata gerada é apresentado ao usuário e cabe ao mesmo aceitar ou não o resultado obtido. Caso, o perfil seja aceito, então o mesmo será adicionado ao banco de dados para que seu uso seja possível em um processo futuro. A Figura 42 apresenta um exemplo de sucata criada pelo SAD proposto.



Figura 42 - Perfil de sucata extraído pelo SAD



Fonte: elaborado pelo autor (2014).

### 5.6.2 Avaliação das Sobras Aproveitáveis

A avaliação se as sobras geradas são aproveitáveis ou não acontece posteriormente ao processo onde o perfil de sucata é extraído. Essa pós-análise tem a finalidade de encontrar partes da perda gerada que não podem ser utilizadas em um novo projeto, por serem muito pequenas.

Em Cherri e Vianna (2013) é proposta uma métrica que torna possível classificar as sobras obtidas em diferentes tipos e através dessa métrica é possível definir se uma sobra é aproveitável ou não.

Embora uma métrica similar seja utilizada, a ação tomada quando uma sobra não aproveitável é encontrada é diferente. Na implementação original, sempre que uma sobra é classificada como perda indesejável, alterações são realizadas no padrão de corte de modo a gerar retalhos e reduzir a perda. No entanto, na implementação proposta por este estudo, a etapa de verificação se as sobras são aproveitáveis ou não ocorre posteriormente ao processo de *nesting* e não durante.

Essa avaliação é realizada através de dois testes. O primeiro teste tem a finalidade de definir se a área e as dimensões do perfil de sucata extraído são suficientes para realizar processos no futuro. Para que esse teste seja realizado, um comparativo entre a área total da porção de material definida como sucata é comparada a um valor mínimo de área, estipulado pelo usuário, através da interface com o usuário. Além disso, as dimensões médias da sucata são comparadas às medidas mínimas, também estipuladas pelo usuário. Para que seja considerada uma sobra aproveitável, a sucata extraída deve possuir área maior ou igual à área mínima definida pelo usuário, assim como as dimensões mínimas definidas.

Sejam  $(L_{su}, W_{su})$ , respectivamente, o comprimento e a largura mínima definida pelo usuário, como as dimensões mínimas para a sobra ser considerada aproveitável e  $(L_s, W_s)$  as medidas da sucata extraída de uma chapa  $c$  ( $k = 1, \dots, n$ ), onde  $n$  é o número de chapas a serem

avaliadas. E ainda, seja  $A_{su}$  a área mínima estipulada pelo usuário e  $A_s$  a área da sucata extraída da chapa  $c$ , é necessário garantir que:

$$L_{s,c} \geq L_{su} \quad \text{e} \quad W_{s,c} \geq W_{su} \quad (11)$$

assim como:

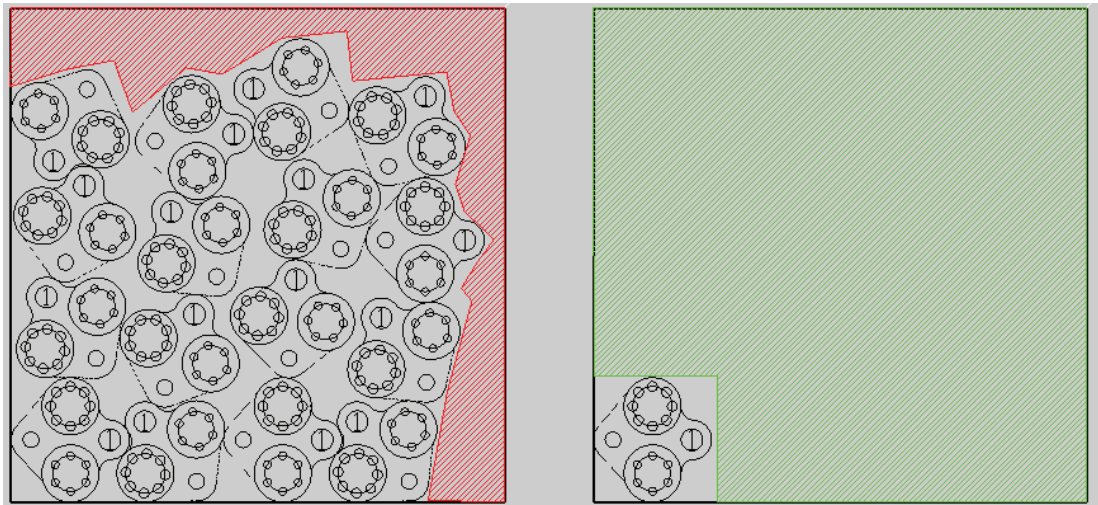
$$A_{s,c} \geq A_{su} \quad (12)$$

O segundo teste garante que, quando comparado ao tamanho do material, as perdas sejam substanciais. Essa opção permite que perdas pequenas, quando comparadas ao tamanho total da chapa, sejam ignoradas. Seja  $\xi$  o percentual de perda mínima estipulada pelo usuário e, seja  $A_k$  a área total da chapa  $k$  sendo analisada, para que a sobra seja aproveitável é preciso garantir que:

$$A_s > \xi A_k \quad (13)$$

Quando a chapa  $k$  passa pelos dois testes, a mesma é então considerada uma sobra aproveitável e será apresentada para o usuário. Caso contrário, o usuário será apenas informado ao fim do processo, o motivo pelo qual a sucata não foi extraída, seja por falhar no primeiro teste ou falhar no segundo teste. A Figura 43 demonstra, respectivamente, uma sobra rejeitada (vermelho) e uma sobra aceita pelo processo (verde), onde 10% é estipulado como área mínima para perda e dimensões mínimo de 1000 mm por 1000 mm.

**Figura 43 - Sobra rejeitada e sobra aceita pela avaliação**



Fonte: elaborado pelo autor (2014).

## 6 TESTES E RESULTADOS

Para validar o modelo proposto, duas métricas de teste são propostas. A primeira é a métrica comum e mais utilizada pela comunidade acadêmica, ou seja, o comparativo dos resultados obtidos neste estudo com resultados obtidos na literatura. Para realizar tais testes, as instâncias ligadas ao problema de corte encontrado em Abrantes (2013) são utilizadas.

Todavia, por estar lidando com um sistema de apoio à decisão, o uso e avaliação do usuário acabam sendo tão importantes quanto a avaliação através das instâncias disponíveis na literatura. Por esse motivo, uma segunda métrica de teste é utilizada e nela usuários deverão avaliar o sistema.

A segunda métrica de teste proposta visa comparar os resultados obtidos pelo sistema implementado com o sistema original (sem automação) com base na opinião do usuário. Para isso, serão solicitados a diversos usuários especialistas e não especialistas que realizem uma série de projetos de manufatura envolvendo um conjunto finito de peças.

Uma vez finalizados, serão analisados critérios como: tempo de processamento, número de tentativas para chegar ao resultado esperado, quantidade de chapas utilizadas, percentual de sobras gerado, percentual de perdas, número de tentativas entre outros parâmetros relevantes para o bom funcionamento do sistema. Através desse levantamento será possível avaliar qualitativamente o sistema proposto.

### 6.1 Análise das Instâncias Encontradas na Literatura

A primeira métrica tem como objetivo principal garantir que, com a implementação do sistema de apoio à tomada de decisão, sejam produzidos resultados otimizados quando comparado à literatura atual. A métrica utilizada aqui compara os resultados obtidos pelo sistema proposto através da simulação de alguns cenários já testados na literatura.

O primeiro conjunto de testes é realizado através do comparativo das instâncias utilizadas por Abrantes (2013). Em seu trabalho, o autor realiza testes em 10 diferentes instâncias retangulares e analisa além dos resultados obtidos, as sobras aproveitáveis geradas pelo processo.

Para efeito de identificação do tamanho das geometrias utilizadas nos testes; a Tabela 15 demonstra os detalhes sobre cada instância, onde são apresentadas as dimensões e quantidades das peças que serão submetidas ao processo de *nesting*, assim como as chapas que estão disponíveis para uso.

**Tabela 15 - Lista de instâncias para testes**

| Descrição das Instâncias |                     |             |                        |             |
|--------------------------|---------------------|-------------|------------------------|-------------|
| #                        | peças               |             | chapas                 |             |
|                          | Qtde. Total         | Qtde. Tipos | Qtde. Total            | Qtde. Tipos |
| 1                        | 5                   | 2           | 2                      | 2           |
|                          | 3 (2x11)<br>2 (5x5) |             | 1 (22x17)<br>1 (14x30) |             |
| 2                        | 2                   | 1           | 2                      | 2           |
|                          | 2 (2x4)             |             | 1 (10x17)<br>1 (10x24) |             |

Tabela 15 - Lista de instâncias para testes (continuação)

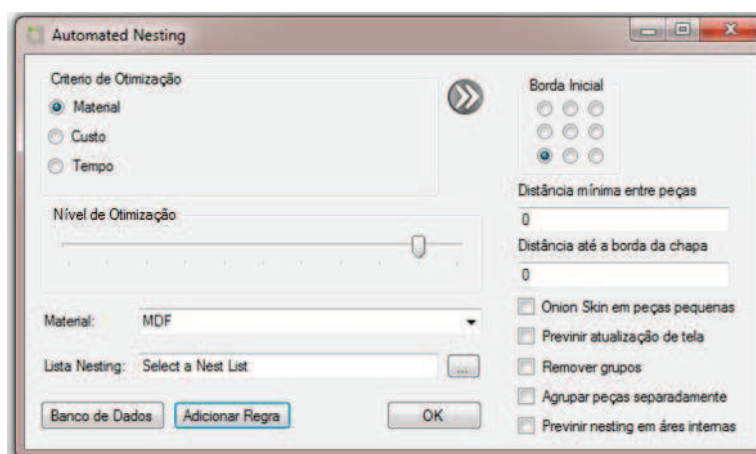
| #  | peças                            |             | chapas                              |             |
|----|----------------------------------|-------------|-------------------------------------|-------------|
|    | Qtde. Total                      | Qtde. Tipos | Qtde. Total                         | Qtde. Tipos |
| 3  | 3                                | 1           | 2                                   | 2           |
|    | 3 (5x4)                          |             | 1 (18x19)<br>1 (26x22)              |             |
| 4  | 7                                | 3           | 3                                   | 3           |
|    | 4 (3x3)<br>1 (4x3)<br>2 (7x1)    |             | 1 (24x12)<br>1 (15x18)<br>1 (17x13) |             |
| 5  | 5                                | 2           | 2                                   | 2           |
|    | 4 (7x1)<br>1 (11x1)              |             | 1 (20x10)<br>1 (29x12)              |             |
| 6  | 7                                | 3           | 2                                   | 2           |
|    | 2 (11x11)<br>3 (2x11)<br>2 (5x5) |             | 1 (22x17)<br>1 (14x30)              |             |
| 7  | 9                                | 2           | 5                                   | 3           |
|    | 5 (9x6)<br>4 (5x3)               |             | 1 (27x23)<br>2 (19x17)<br>2 (19x19) |             |
| 8  | 9                                | 2           | 2                                   | 2           |
|    | 5 (3x2)<br>4 (3x1)               |             | 1 (19x17)<br>1 (16x11)              |             |
| 9  | 10                               | 2           | 2                                   | 2           |
|    | 4 (3x4)<br>1 (3x1)               |             | 1 (18x20)<br>1 (13x10)              |             |
| 10 | 10                               | 2           | 2                                   | 2           |
|    | 7 (4x5)<br>3 (5x2)               |             | 1 (22x14)<br>1 (18x22)              |             |

Fonte: Adaptado de Abrantes (2013).

Para que o teste seja possível, utilizando as informações contidas na tabela, foram criadas geometrias no *software* CAD que representam as mesmas. Uma vez que as geometrias foram criadas, é possível definir uma lista *nesting* onde serão definidas as quantidades de cada peça, e submeter essa lista ao sistema de apoio à decisão.

No sistema de apoio à decisão, uma configuração constante foi utilizada para todos os testes e de forma a coincidir com a implementação proposta por Abrantes (2013), o critério de otimização escolhido foi o material. Já o nível de otimização escolhido foi o máximo, sendo que; por estarmos lidando apenas com geometrias, é definido que a distância entre as peças e a distância até a borda da chapa devem ser desconsideradas nulas (0). A Figura 44 demonstra a parametrização básica escolhida.

Figura 44 - Parametrização do SAD



Fonte: elaborado pelo autor (2015).

Cada instância foi submetida ao sistema de apoio à decisão 75 vezes e dessas execuções foram coletadas informações tais como: número de chapas utilizadas, área total das chapas utilizadas, área perdida e área contendo sobras aproveitáveis. Os resultados dos testes, apresentados pela Tabela 16 foram obtidos em um computador cuja configuração possui 8.00 GB de memória RAM, processador Intel(R) Core (TM) i5-3210 CPU @2.50 Ghz com 4 *threads* e Windows 7 64 bits.

Tabela 16 - Resultados dos testes realizados com as instâncias 1 à 10

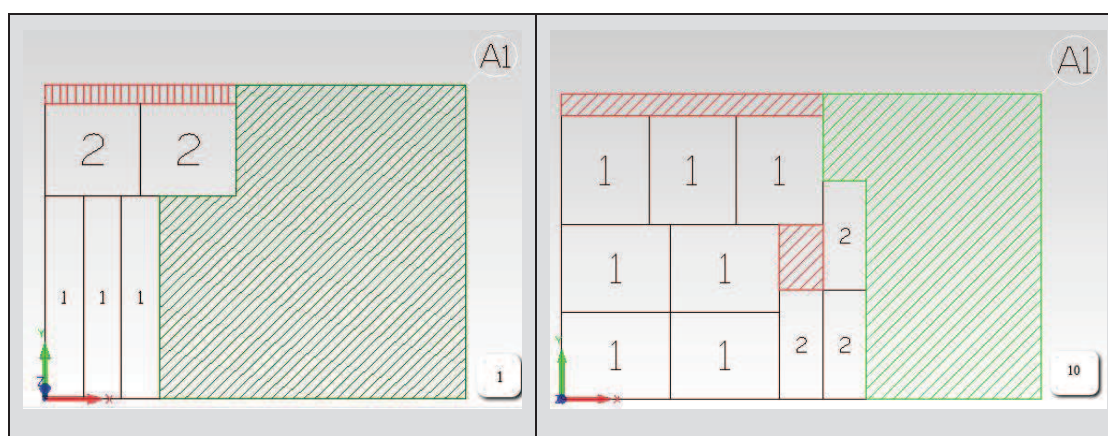
| #                                                 | 1       | 2       | 3       | 4       | 5       | 6      | 7       | 8           | 9       | 10          |
|---------------------------------------------------|---------|---------|---------|---------|---------|--------|---------|-------------|---------|-------------|
| <b>Número de chapas utilizadas</b>                | 1       | 1       | 1       | 1       | 1       | 1      | 1       | 1           | 1       | 1           |
| <b>Área total das chapas</b>                      | 374     | 240     | 342     | 288     | 200     | 374    | 621     | 323         | 360     | 308         |
| <b>Perda média (Área)</b>                         | 10      | 0       | 0       | 1,143   | 0       | 16,1   | 0       | 11,77142857 | 0       | 17,71428571 |
| <b>Desvio Padrão - Perdas <math>\sigma</math></b> | 0       | 0       | 0       | 1,069   | 0       | 0      | 0       | 20,1035415  | 0       | 2,058663459 |
| <b>% médio de perdas</b>                          | 2,6738  | 0,0000  | 0,0000  | 0,3968  | 0,0000  | 4,3048 | 0,0000  | 3,6444      | 0,0000  | 5,7514      |
| <b>Sobras aproveitáveis média (Área)</b>          | 248,2   | 160     | 281,96  | 225,46  | 161     | 0      | 291,2   | 269,4285714 | 309     | 119,8342857 |
| <b>Desvio Padrão - Sobras <math>\sigma</math></b> | 0       | 0       | 0       | 1,03    | 0       | 0      | 0       | 20,1035415  | 0       | 2,058663459 |
| <b>% médio reutilizável</b>                       | 66,3636 | 66,6667 | 82,4444 | 78,2857 | 80,5000 | 0,0000 | 46,8921 | 83,4144     | 85,8333 | 38,9072     |
| <b>Tempo médio de execução (segundos)</b>         | 22      | 44,2    | 22,42   | 61,44   | 37,1    | 24,7   | 17      | 37,43       | 24,2    | 45,7        |

Fonte: elaborado pelo autor (2015).

Conforme apresentado pela Tabela 16, podemos notar que todo o processo de decisão acontece rapidamente, sendo que, no pior resultado obtido, acontece em pouco mais de 1

minuto. Além disso, o percentual de material considerado como reaproveitável é relativamente grande (reaproveitamento médio de 62,93%). Isso demonstra a importância do módulo de extração de sobras aproveitáveis, e possibilita ao usuário a reutilização da chapa pelo *software* CAM, uma vez que o sistema alimenta automaticamente o banco de dados com o material reutilizável. Alguns dos resultados obtidos são apresentados pela Figura 45, onde áreas vermelhas são áreas que serão perdidas da chapa e áreas verdes representam sobras aproveitáveis.

**Figura 45 - Resultados das chapas obtidas**



Fonte: elaborado pelo autor (2015).

Visando realizar um confronto dos resultados apresentados por Abrantes (2013), uma comparação entre os resultados obtidos pelo autor e os resultados obtidos nesse estudo foi realizada. Nesse comparativo, foram analisados alguns aspectos importantes do processo tais como: número de chapas utilizadas, área perdida e área contendo sobras aproveitáveis. Esse comparativo é demonstrado através da Tabela 17.

**Tabela 17 - Comparativo de resultados entre literatura e estudo**

| #  | Resultados em Abrantes (2013)     |                      |                                   | Resultados no SAD proposto  |                      |                                   |
|----|-----------------------------------|----------------------|-----------------------------------|-----------------------------|----------------------|-----------------------------------|
|    | Número de chapas utilizadas       | Área perdida (média) | Área sobras aproveitáveis (média) | Número de chapas utilizadas | Área perdida (média) | Área sobras aproveitáveis (média) |
| 1  | 1                                 | 5                    | 253                               | 1                           | 10                   | 248                               |
| 2  | 1                                 | 0                    | 160                               | 1                           | 0                    | 160                               |
| 3  | 1                                 | 0                    | 282                               | 1                           | 0                    | 282                               |
| 4  | 1                                 | 4                    | 161                               | 1                           | 1,142857143          | 225,4628571                       |
| 5  | 1                                 | 3                    | 178                               | 1                           | 0                    | 161                               |
| 6  | 2                                 | 5                    | 431                               | 1                           | 16,1                 | 0                                 |
| 7  | 1                                 | 37                   | 0                                 | 1                           | 0                    | 291,2                             |
| 8  | Não foram apresentados resultados |                      |                                   | 1                           | 11,77142857          | 269,4285714                       |
| 9  | 1                                 | 0                    | 76                                | 1                           | 0                    | 309                               |
| 10 | Não foram apresentados resultados |                      |                                   | 1                           | 17,71428571          | 119,8342857                       |

Fonte: elaborado pelo autor (2015).

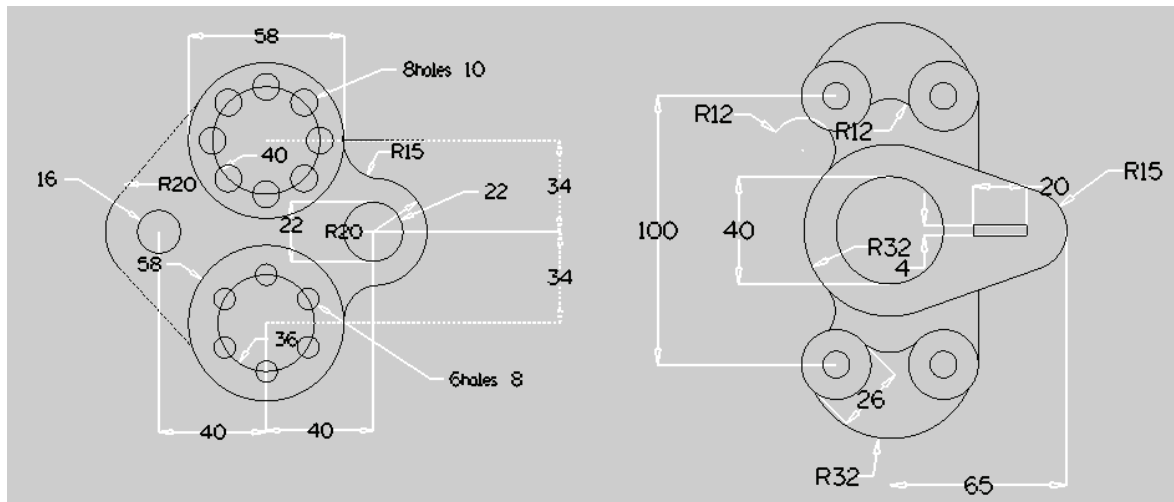
Quando comparando, quantitativamente, a área perdida, podemos observar que em diversas instâncias como 4, 5 e 7, esse parâmetro foi inferior ao resultado encontrado em Abrantes (2013). Além disso, em diversas instâncias a quantidade de perdas foi a mesma, como nas instâncias 2, 3 e 9.

Embora na instância 6 a área perdida total foi considerada superior, quando observando o número de chapas utilizadas é possível verificar que o SAD utilizou apenas uma chapa, enquanto a solução do autor utilizou duas. O resultado do autor traz a segunda chapa praticamente sem uso, gerando 2 blocos grandes de material e por esse motivo podemos concluir que a quantidade de área considerada como sobras aproveitáveis é muito maior do que a obtida pelo SAD, onde toda a segunda chapa não é utilizada.

Quando comparando a quantidade de sobras aproveitáveis geradas, podemos observar que, na maioria dos casos, essa quantidade é a mesma ou inferior, como por exemplo, nas instâncias 2, 3, 4, 7 e 9. Nas instâncias 8 e 10 não foram apresentados resultados na solução proposta por Abrantes (2013).

Conforme pode ser observado nas instâncias extraídas de Abrantes (2013), o número máximo de peças submetidas ao processo é relativamente pequeno, sendo o máximo de 10 peças. Por esse motivo, uma nova combinação de peças não retangulares, compostas pelas peças P1 e P2, demonstradas pela Figura 46, é utilizada para realizar testes com instâncias maiores, variando entre 10 e 100 peças.

Figura 46 - Novos testes com peças não retangulares



Fonte: elaborado pelo autor (2015).

Sendo as peças P1 e P2 de maior complexidade, estas se tratam de geometrias não retangulares. Tais peças, mesmo sendo utilizadas como um segundo teste de validação, não são facilmente descritas por uma tabela. A Tabela 18 apresenta novas instâncias criadas, considerando um número definido de ambas as peças para os testes. As medidas são consideradas as mesmas ilustradas na Figura 46.

**Tabela 18 - Listas de instâncias para testes definidas neste estudo**

| #  | Quantidades    | Número Total de peças | Chapas disponíveis           |
|----|----------------|-----------------------|------------------------------|
| 11 | 5xP1<br>5xP2   | 10                    | 5 (1000x1000)<br>3 (750x300) |
| 12 | 12xP1<br>13xP2 | 25                    | 5 (1000x1000)<br>3 (750x300) |
| 13 | 25xP1<br>25xP2 | 50                    | 5 (1000x1000)<br>3 (750x300) |
| 14 | 37xP1<br>38xP2 | 75                    | 5 (1000x1000)<br>3 (750x300) |
| 15 | 50xP1<br>50xP2 | 100                   | 5 (1000x1000)<br>3 (750x300) |

Fonte: elaborado pelo autor (2015).

Todas as peças foram submetidas aos mesmos testes, utilizando o mesmo número de iterações e a mesma seleção de parâmetros apresentado nas instâncias anteriores testadas. Os resultados observados são apresentados pela Tabela 19.

**Tabela 19 - Resultados dos testes realizados utilizando as instâncias 11 à 15**

| #                                                 | 11          | 12         | 13          | 14          | 15          |
|---------------------------------------------------|-------------|------------|-------------|-------------|-------------|
| <b>Número de chapas utilizadas</b>                | 1           | 2          | 4           | 5           | 7           |
| <b>Área total das chapas</b>                      | 250000      | 500000     | 1000000     | 1250000     | 1700000     |
| <b>Perda média (Área)</b>                         | 17800,6954  | 57138,9051 | 97898,4971  | 150898,4971 | 150898,4971 |
| <b>Desvio Padrão - Perdas <math>\sigma</math></b> | 219,56      | 272,65     | 245,6       | 792,50      | 1120,489    |
| <b>% médio de perdas</b>                          | 7,1203      | 11,4278    | 9,7898      | 12,0719     | 8,8764      |
| <b>Sobras aproveitáveis média (Área)</b>          | 112136,9758 | 101927,975 | 233435,1099 | 109539,53   | 213104,8847 |
| <b>Desvio Padrão - Sobras <math>\sigma</math></b> | 187,543     | 225        | 234,21      | 742,00      | 1032,456    |
| <b>% médio reutilizável</b>                       | 44,8548     | 20,3856    | 23,3435     | 8,7632      | 12,5356     |
| <b>Tempo médio de execução (segundos)</b>         | 67,32       | 90,5       | 120         | 175,4       | 220,5       |

Fonte: elaborado pelo autor (2015).

Conforme observamos na Tabela 19, o sistema continua estável e fornecendo respostas em um tempo aceitável mesmo quando processando um número elevado de peças e com maior grau de dificuldade, já que se trata de peças com formatos livres e áreas internas.

Além disso, em alguns casos, a área considerada como sobra aproveitável é razoavelmente alta, possibilitando assim a reutilização desse material e sua respectiva entrada no banco de dados.



## 6.2 Análise Feita por Usuários

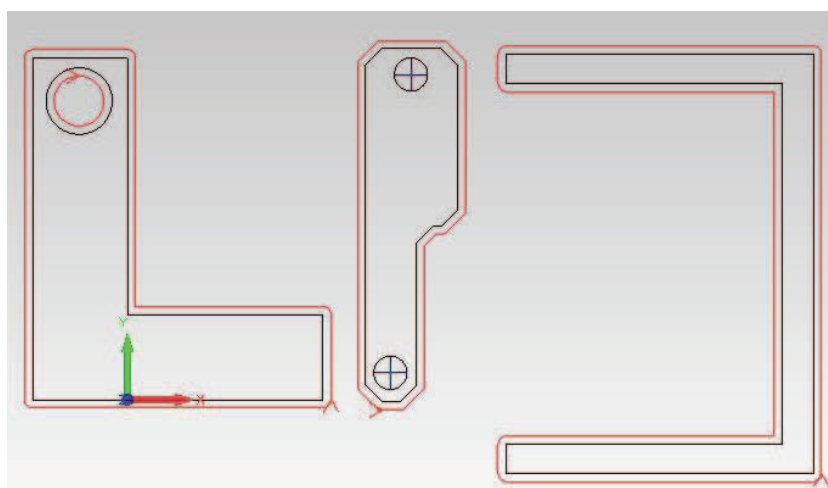
Embora a primeira métrica seja muito importante para verificar os resultados obtidos e validar o sistema criado, considerando que o estudo trata da implementação de um SAD, onde existe uma necessidade natural de intervenção e comunicação com o usuário, a avaliação feita pelos mesmos acaba sendo uma métrica de teste muito importante para o sistema.

O SAD proposto foi desenvolvido com o objetivo de realizar tarefas distintas propostas pelo usuário. Por esse motivo, a avaliação realizada por usuários reais do *software* que utilizam, até então de forma não automática, os recursos que estão sendo automatizados por esse estudo, pode dar fortes indícios sobre o sucesso da implementação realizada. Isso é dado pelo fato que os usuários irão comparar o novo sistema automatizado com o sistema original.

Além disso, nessa métrica poderão ser testados todos os recursos oferecidos pelo sistema, diferentemente da métrica inicial onde os recursos especiais de *nesting* não são vastamente utilizados, já que apenas geometrias estavam sendo submetidas ao processo. Nesta nova métrica, todos os recursos especiais de automação, ligados aos caminhos de ferramentas, deverão ser testados.

A etapa de teste aqui proposta visa avaliar os resultados obtidos quando usuários de diversos níveis de conhecimento fazem uso da ferramenta criada. Para tanto, um conjunto de 10 tarefas foram enviadas para 2 usuários do *software* Alphacam. Sendo desses 2, um especialista na área de CAM e gestor de produção (denominado neste estudo de usuário 1), e o outro, um usuário com conhecimento inicial do *software* (denominado neste estudo de usuário 2). Para cada problema, foi solicitado aos usuários que realizassem as tarefas de forma manual e coletasse os resultados obtidos e em seguida realizassem uma segunda tentativa, com as mesmas tarefas, porém desta vez utilizando o SAD proposto por este estudo. A Figura 47 demonstra uma das peças das 10 tarefas.

Figura 47 - Exemplo de instância de teste utilizada na segunda métrica de testes

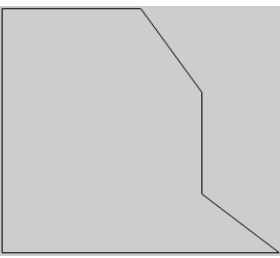
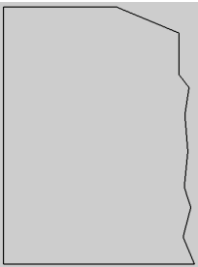
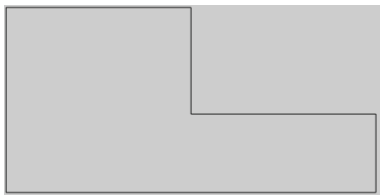
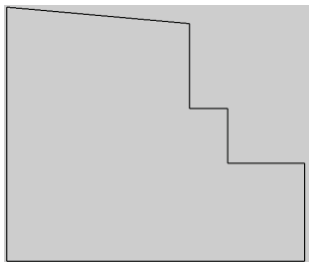


Fonte: elaborado pelo autor (2015).

Em conjunto com as peças, foi solicitado o preenchimento de uma tabela demonstrando algumas das informações relevantes para qualificar os resultados obtidos. Nesse formulário, foram solicitadas para cada tarefa, as seguintes informações: tempo para selecionar as chapas utilizadas, tempo para realizar a parametrização, número de tentativas, área total utilizada de material, área das sobras aproveitáveis e número de chapas utilizadas.

Para a realização dos testes, foi disponibilizado para ambos os usuários um banco de dados padrão, contendo determinadas chapas disponíveis para os testes. Dessa forma, é possível comparar resultados obtidos de acordo com os usuários e entre métodos não automatizado e automatizado. Nesse banco de dados, são disponibilizados 8 tipos distintos de chapas que quando somadas, representam um total de 90 chapas. A Tabela 20 demonstra as chapas utilizadas durante o processo de testes.

**Tabela 20 - Dados das chapas disponíveis para segunda métrica de testes**

| Dimensões | Quantidade | Área     | Custo | Espessura | Extraída de Sobras | Perfil da Chapa                                                                       |
|-----------|------------|----------|-------|-----------|--------------------|---------------------------------------------------------------------------------------|
| 1000x1800 | 15         | 1800000  | 180   | 2,5       | Não                | (retangular)                                                                          |
| 2000x1000 | 15         | 2000000  | 210   | 1,8       | Não                | (retangular)                                                                          |
| 750x350   | 20         | 262500   | 75    | 1,8       | Não                | (retangular)                                                                          |
| 750x750   | 20         | 562500   | 115   | 2,5       | Não                | (quadrado)                                                                            |
| 1700x1500 | 3          | 1826385  | 235   | 1,8       | Sim                |   |
| 150x200   | 10         | 27619,47 | 50    | 1,8       | Sim                |  |
| 700x350   | 4          | 173950   | 70    | 1,8       | Sim                |  |
| 2000x1750 | 3          | 4449260  | 250   | 2,5       | Sim                |  |

Fonte: elaborado pelo autor (2015).

Ao longo desse capítulo são apresentados os resultados obtidos através dessa etapa de testes.

### 6.2.1 Usuário 1 (Especialista)

O usuário 1 possui 10 anos de experiência na área de CAD e CAM, sendo atualmente gestor de uma empresa do ramo metalúrgico e liderando uma equipe de 6 pessoas. O gestor é responsável pela área de processos de manufatura aplicados às chapas, e dentre sua equipe, existem usuários com diversos níveis de conhecimento sobre o processo. O usuário 1 e sua equipe realizaram os testes solicitados utilizando as 10 instâncias de testes criadas, assim como as chapas estipuladas como matéria prima. As respostas obtidas podem ser resumidas na Tabela 21.

**Tabela 21 - Resultados pelo método de teste manual (usuário 1)**

| #  | Tempo Seleção do Material (minutos) | Tempo na Parametrização do <i>nesting</i> (minutos) | Número de Tentativas | Total de Perdas (Área) | Sobras Geradas (Área) | Tempo de Usinagem | Chapas Utilizadas        |
|----|-------------------------------------|-----------------------------------------------------|----------------------|------------------------|-----------------------|-------------------|--------------------------|
| 1  | 2                                   | 1                                                   | 3                    | 89136                  | 46105,9115            | 2m 50s            | 1x 750x350               |
| 2  | 3                                   | 1                                                   | 2                    | 260466,026             | 4692413,65            | 18m 14s           | 7x 2000x1000             |
| 3  | 3                                   | 1                                                   | 2                    | 744972,305             | 737783,5              | 29m 04s           | 2x 750x350<br>1x 750x750 |
| 4  | 2                                   | 1                                                   | 1                    | 70004,6028             | 842385,786            | 40m 58s           | 2x 2000x1000             |
| 5  | 3                                   | 1                                                   | 2                    | 130233,013             | 2815448,19            | 12m 14s           | 5x 2000x1000             |
| 6  | 3                                   | 1                                                   | 1                    | 68408,1714             | 27079,4336            | 1m 32s            | 1x 750x350               |
| 7  | 3                                   | 1                                                   | 1                    | 280018,411             | 3959213,19            | 1h 24m            | 5x 2000x1000             |
| 8  | 4                                   | 2                                                   | 3                    | 140009,206             | 1684771,57            | 2h 12m            | 4x 2000x1000             |
| 9  | 3                                   | 1                                                   | 2                    | 324472,305             | 987743,5              | 37m 04s           | 3x 750x750<br>1x 750x350 |
| 10 | 5                                   | 3                                                   | 3                    | 2207303,66             | 660236,653            | 45m 32s           | 15x 750x350              |

Fonte: elaborado pelo autor (2015).

Analisando os resultados, é possível notar que o usuário possui a tendência de não utilizar uma grande variedade de chapas distintas. Esse tipo de escolha é racional se considerarmos que o banco de dados oferece 8 tipos distintos de chapas e quando somando suas quantidades, possuímos 90 chapas disponíveis para realizar a escolha; criando um número muito grande de possíveis combinações.

Além disso, é possível perceber que, independentemente da instância de teste, o usuário gastou tempo similar na etapa de parametrização do processo de *nesting*. Isso acontece devido à forte tendência do usuário em realizar da mesma forma a definição de parâmetros em diversos itens, realizando apenas pequenas variações de acordo com o formato e caminhos de ferramenta aplicados às mesmas.

Depois de concluídos os testes não automatizados (método manual), o usuário realizou as mesmas tarefas para o segundo teste sendo que dessa vez utilizando o SAD proposto por esse estudo. Os resultados obtidos na realização desses testes são apresentados pela Tabela 22.

Tabela 22 - Resultados obtidos através do SAD (usuário 1)

| #  | Tempo Total (minutos) | Número de Tentativas | Sobras Geradas (Área) | Critério de Otimização Utilizado | Tempo de Usinagem | Nível de Otimização Utilizado | Chapas Utilizadas            |
|----|-----------------------|----------------------|-----------------------|----------------------------------|-------------------|-------------------------------|------------------------------|
| 1  | 1                     | 2                    | 44315,24              | material                         | 2m 50s            | 7                             | 1x 750x350                   |
| 2  | 1,5                   | 1                    | 4692413,646           | material                         | 18m 14s           | 7                             | 7x 2000x1000                 |
| 3  | 1                     | 1                    | 724231,44             | tempo                            | 20m 32s           | 10                            | 2x 750x350<br>1x 750x750     |
| 4  | 2                     | 1                    | 842385,7858           | tempo                            | 27m 28s           | 7                             | 1x 2000x1750<br>1x 1700x1500 |
| 5  | 3                     | 1                    | 26046,60261           | custo                            | 13m 27s           | 7                             | 3x 2000x1750                 |
| 6  | 1                     | 1                    | 24335,3649            | material                         | 1m 12s            | 10                            | 1x 750x350                   |
| 7  | 3                     | 1                    | 24004,89              | custo                            | 1h 12m            | 10                            | 3x 2000x1750                 |
| 8  | 2                     | 2                    | 140009,2056           | material                         | 2h 12m            | 7                             | 4x 2000x1000                 |
| 9  | 2                     | 2                    | 831423,5              | material                         | 32m 04s           | 7                             | 2x 750x750<br>2x 750x350     |
| 10 | 4                     | 3                    | 52211,823             | custo                            | 32m 32s           | 10                            | 1x 2000x1750                 |

Fonte: elaborado pelo autor (2015).

Conforme podemos observar na Tabela 22, e comparando a mesma aos resultados obtidos sem apoio do SAD, um dos fatores evidentes é a redução do tempo para se realizar o processo. Nos resultados gerados pelo SAD é possível observar que, na grande maioria dos casos, 1 minuto é o suficiente para realizar todo o processo em sua plenitude, diferentemente dos resultados obtidos sem apoio do sistema proposto neste trabalho. Nos resultados obtidos manualmente, observa-se que, na grande maioria das tarefas, foi necessário tempo superior a 3 minutos para realizar todo o processo. Isso mostra fortes indícios de que o SAD será uma maneira mais rápida de realizar todo o processo de decisão.

Além disso, em alguns casos (instâncias 2 e 8) os conjuntos de chapas escolhidas, assim como os parâmetros, foram os mesmos selecionados pelo usuário pelo método não automatizado. Em ambos os casos onde isso ocorreu, o critério de otimização escolhido foi o material. Por esse motivo, é possível concluir que, quando realizando a escolha de forma não automatizada, o usuário também estava determinado a reduzir a área utilizada, assim como a quantidade de chapas utilizadas.

Também é possível observar que nas instâncias 5, 7 e 10 houve opção pela mesma chapa. Isso acontece pelo fato da chapa “2000x1750” possuir área suficiente para acomodar todas as peças e possuir custo reduzido por ser resultado de extração de sobras aproveitáveis. Porém, essa chapa requer maior tempo de cálculo para a acomodação das peças na etapa *nesting* devido ao seu formato irregular, aumentando assim o tempo de processamento. Além disso, é possível observar que, em alguns casos, o tempo de usinagem é reduzido, devida à redução do número de chapas.

Nas instâncias 3 e 4, embora a seleção de chapas não varie muito da seleção realizada pelo usuário, a grande diferença dos resultados está no tempo de usinagem. Isso ocorre devido à escolha do critério de otimização do tempo, que força a seleção de opções da parametrização como *bridge nesting* e redução das retrações de ferramenta. As junções dessas opções fazem com que o processo seja realizado sem que a ferramenta retraia, diminuindo consideravelmente o tempo de usinagem.

Quando aplicando ao seu próprio processo de manufatura, o usuário reportou que com o SAD e seu recurso de extração de sobras aproveitáveis, é possível ter um maior controle sobre o estoque. Esse usuário reportou ter problemas para controlar as sobras aproveitáveis e que em diversas ocasiões no passado, tais sobras foram desperdiçadas pela falta de informação das mesmas (disponibilidade e área aproveitável) e acabaram por ser descartadas. Além disso, o usuário reportou que, mesmo utilizando o *software* CAM, muitas vezes não é possível compreender a grande variedade de opções oferecidas pelo mesmo e, com o objetivo de otimizar o seu tempo, optava por sempre utilizar o mesmo conjunto de opções.

Com o novo sistema, o usuário foi capaz de inserir novas regras e adequar o sistema aos seus processos fabris. O usuário reportou que 3 novas regras foram criadas para garantir que o processo fosse realizado de acordo com o padrão esperado pela empresa e, uma vez que essas regras vão sendo incorporadas pelo sistema, o usuário acredita que mesmo outros usuários da equipe sejam capazes de utilizar o sistema sem auxílio de um especialista humano. A não necessidade de treinamento para lidar com o sistema também demonstrou a versatilidade do mesmo e a capacidade de gerar resultados adequados à realidade, mesmo considerando que estes usuários não possuem vasta experiência em processos de fabricação com uso de *nesting*.

#### 6.2.2 Usuário 2 (Não-especialista)

O usuário 2, sendo este com menor experiência na área, trabalha na área de suporte de *software* CAD/CAM e possui em torno de 2 anos de experiência anterior como programador de produção de uma empresa do ramo metal-mecânico. Este especialista possui conhecimento com o processo de chapas como usuário, porém o mesmo não exercia a programação da máquina CNC para efetuar os cortes de chapas. Entretanto, o usuário foi capaz de presenciar as dificuldades do dia a dia no controle de sobras e é de seu conhecimento a necessidade de obter mão de obra qualificada quanto à programação do *nesting*.

Sendo sua atual posição na área de suporte a *softwares* que possuem a ferramenta *nesting*, o usuário pode ser considerado como capacitado para avaliar a ferramenta SAD pelo conhecimento que o mesmo possui das duas pontas da área de corte de chapas (usuário e analista).

Quando observando o tempo para realizar os processos de forma não automatizada, é possível observar pela Tabela 23, a curva de aprendizado do usuário através do tempo necessário para realizar os processos. É possível notar que o tempo para realizar a parametrização foi reduzindo gradativamente ao longo dos testes. O usuário reportou que isso ocorreu devido à necessidade de parar o processo para que fosse possível ler a documentação do *software* e tentar entender as consequências de cada parâmetro.

Ainda analisando o tempo necessário para realizar o processo, quando comparando os resultados obtidos pelo usuário 2 com os resultados obtidos pelo usuário 1, podemos notar que nas instâncias 1 e 6 a mesma seleção de chapas e resultados similares foram obtidos. No entanto, podemos observar que o usuário 2 teve de realizar diversas tentativas até que esse resultado fosse obtido. Em muitos casos, esse número maior de tentativas fez com que o usuário chegasse a bons resultados.

**Tabela 23 - Resultados pelo método de teste manual (usuário 2)**

| #  | Tempo Seleção do Material (minutos) | Tempo na Parametrização do Nesting (minutos) | Número de Tentativas | Total Perdas (Área) | Sobras Geradas (Área) | Tempo de Usinagem | Chapas Utilizadas            |
|----|-------------------------------------|----------------------------------------------|----------------------|---------------------|-----------------------|-------------------|------------------------------|
| 1  | 3                                   | 5                                            | 3                    | 89136               | 46105,9115            | 3m 22s            | 1x 750x350                   |
| 2  | 5                                   | 5                                            | 3                    | 343815,1545         | 5161655,01            | 22m 32s           | 1x 2000x1750<br>6x 2000x1000 |
| 3  | 7                                   | 4                                            | 4                    | 886517,0431         | 848451,025            | 32m               | 3x 750x750                   |
| 4  | 6                                   | 3                                            | 1                    | 69304,55677         | 732875,6336           | 50m 58s           | 1x 2000x1750<br>1x 150 x 200 |
| 5  | 5                                   | 3                                            | 2                    | 130233,0131         | 2815448,187           | 14m 32s           | 3x 2000x1000<br>2x 1700x1500 |
| 6  | 3                                   | 5                                            | 3                    | 68408,1714          | 27079,43363           | 1m 50s            | 1x 750x350                   |
| 7  | 3                                   | 2                                            | 4                    | 294019,3318         | 1853248,729           | 1h 33m            | 7x 1700x1500                 |
| 8  |                                     | 2                                            | 3                    | 154010,1262         | 1516294,414           | 2h 10m            | 4x 2000x1000                 |
| 9  | 3                                   | 2                                            | 3                    | 350430,0895         | 96351,13735           | 38m 24s           | 4x 750x750                   |
| 10 | 2                                   | 3                                            | 2                    | 1295194,712         | 52211,823             | 32m 32s           | 1x 2000x1750                 |

Fonte: elaborado pelo autor (2015).

Analisando as escolhas de chapas realizadas pelo usuário 2, podemos observar que, o mesmo realizou um maior número diferente de combinações de chapas, quando comparando ao usuário 1. De certa forma, isso é esperado já que o usuário realizou diversas tentativas e, possivelmente, cada tentativa utilizou um conjunto diferente de materiais.

**Tabela 24 - Resultados obtidos através do SAD (usuário 2)**

| #  | Tempo Total (minutos) | Número de Tentativas | Sobras Geradas (Área) | Critério de Otimização Utilizado | Tempo de Usinagem | Nível de Otimização Utilizado | Chapas Utilizadas                            |
|----|-----------------------|----------------------|-----------------------|----------------------------------|-------------------|-------------------------------|----------------------------------------------|
| 1  | 1                     | 1                    | 44105,7               | material                         | 3m                | 5                             | 1x 750x350                                   |
| 2  | 2                     | 1                    | 4694201,73            | material                         | 18m 14s           | 5                             | 7x 2000x1000                                 |
| 3  | 2                     | 1                    | 347900                | custo                            | 23m 24s           | 7                             | 4x 700x350<br>1x 750x750                     |
| 4  | 2                     | 1                    | 35065.0539            | material                         | 32m 28s           | 7                             | 1x 2000x1750<br>3x 750x750<br>1x 750x350     |
| 5  | 3                     | 1                    | 25265.2045            | tempo                            | 13m 26s           | 7                             | 3x 2000x1750                                 |
| 6  | 1                     | 1                    | 24335,3649            | material                         | 1m 12s            | 10                            | 1x 750x350                                   |
| 7  | 3                     | 2                    | 2400.489              | material                         | 1h 02m            | 10                            | 1x 2000x1750<br>3x 1700x1500<br>2x 2000x1000 |
| 8  | 2                     | 1                    | 75645.0221            | custo                            | 2h 32m            | 10                            | 1x 2000x1750<br>1x 1700x1500<br>1x 1000x1800 |
| 9  | 2                     | 1                    | 327100                | custo                            | 33m 12s           | 7                             | 1x 1000x1800                                 |
| 10 | 3                     | 2                    | 28716.5027            | material                         | 35m 32s           | 10                            | 1x 1000x1800<br>1x 2000x1000<br>1x 700x350   |

Fonte: elaborado pelo autor (2015).

Uma vez concluídos os testes não automatizados, o usuário realizou as mesmas tarefas para o segundo teste, sendo dessa vez utilizado o SAD proposto por esse estudo. Os resultados obtidos na realização desses testes são apresentados na Tabela 24.

Conforme podemos observar, da mesma forma que com o usuário 1, os resultados obtidos através do uso do SAD foram produzidos de forma mais rápida do que os resultados criados sem o mesmo. Essa situação fica ainda mais evidente quando comparando os resultados obtidos pelo SAD quando executado pelo usuário 2. Conforme descrito previamente, o usuário 2 teve de paralisar o processo algumas vezes para entender os parâmetros de *nesting*, o que foi totalmente eliminado quando realizando o processo através do SAD. A média de tempo para realizar as tarefas sem o sistema automatizado foi de aproximadamente 7 minutos, porém quando realizando o processo através do sistema automatizado, essa média foi reduzida para aproximadamente 2 minutos.

Quando comparando o número de tentativas, podemos verificar, também uma drástica redução das mesmas. O número médio de tentativas pelo método não automatizado foi de aproximadamente 3 tentativas, já com o sistema SAD esse número caiu para aproximadamente 1 tentativa. Isso demonstra fortes indícios de que o usuário ficou satisfeito com os resultados obtidos, eliminando assim a necessidade de realizar diversas tentativas.

Outro fator interessante que pode ser observado, quando comparando os resultados entre ambos os usuários, é que o sistema apresentou resultados similares para usuários com diferentes níveis de experiência. Em casos onde os usuários selecionaram o mesmo tipo de critério de otimização para as mesmas instâncias, conforme pode ser observado nas instâncias 1, 2 e 6, a escolha do critério de otimização selecionado como material faz com que o mesmo conjunto de chapas seja selecionado, e conseqüentemente resultados muito similares sejam obtidos. Embora os resultados sejam similares, devido ao diferente nível de automação utilizado nos resultados obtidos pelo usuário 2, e por esse motivo podemos observar uma pequena redução na quantidade de sobras aproveitáveis e um pequeno aumento no tempo de usinagem.

O usuário 2 reportou que, com o ritmo atual dos negócios e o dia-a-dia industrial, *softwares* CAD/CAM não deveriam exigir dos usuários o entendimento de um alto número de parâmetros, como no caso do processo de *nesting*. Nesse aspecto, o usuário 2 considerou que o SAD é de grande ajuda, uma vez que o tempo gasto para se habituar a todo o ciclo de escolha de material, escolha de parâmetros de *nesting* e obtenção da sucata do processo, é mínimo.

Concluindo, o usuário 2 reportou a importância do módulo de extração das sobras aproveitáveis. De acordo com o mesmo, em experiências passadas, passou por situações onde as sobras aproveitáveis eram abandonadas pelo operador da máquina CNC, que por sua vez sempre optava por utilizar chapas novas disponíveis no estoque. Com o módulo de extração das sobras aproveitáveis e o banco de dados, é possível para gestores e coordenadores controlar e cobrar dos operadores que esses materiais sejam reutilizados.

## 7 CONCLUSÕES

No presente estudo, um sistema híbrido de apoio à tomada de decisão foi desenvolvido para auxiliar e automatizar o processo de manufatura envolvendo chapas como matéria prima. O estudo visa reduzir a intervenção do usuário nesse processo de manufatura e, para tanto, propõe um SAD responsável por automatizar diversas etapas do processo. O SAD proposto auxilia o usuário desde a primeira etapa (CAD), até a última, onde o código NC é enviado para a máquina CNC.

Com a implementação do sistema proposto, todo o processo de seleção de material e criação de arranjos da melhor maneira possível passa a exibir baixa intervenção humana. Assim, potencializando o índice de resultados ótimos através da redução do uso desnecessário de matéria prima e da eliminação das atividades que não agregam valor e que resultam em gastos de tempo e recursos. Tais perdas adicionam custos desnecessários aos produtos (BRINSON, 1996), como no caso do tempo utilizado pelo usuário na etapa de parametrização, que se torna desnecessário devido à alta automação do SAD.

Analisando o SAD implementado, foi possível observar que os resultados obtidos foram satisfatórios e, em muitos casos, superiores aos obtidos em outros testes realizados na literatura. Além disso, a aceitação do sistema pelos usuários que realizaram os testes de performance e viabilidade de uso foi considerada excelente. Conforme apontado pelos usuários, o número de parâmetros a ser selecionado reduziu drasticamente, tornando assim o sistema mais simples de ser utilizado.

A análise realizada por usuários experientes apontou que, mesmo para esse tipo de usuário, o sistema pode ser benéfico já que o mesmo consegue resultados tão eficientes quanto o obtido de forma convencional. Além disso, a possibilidade de passar conhecimento para o sistema especialista foi considerada um ponto positivo para esses usuários, uma vez que eles podem dividir tal conhecimento com o sistema especialista.

A análise realizada por novos usuários do *software* aponta um fator ainda mais benéfico já que os mesmos indicaram que por não entenderem as opções disponíveis na ferramenta de *nesting*, muitas vezes acabam deixando de usá-las, optando sempre pela configuração padrão do *software*. Outro fator interessante apontado, pelo usuário 2, foi a quantidade de tempo que pode ser poupada quando utilizando o sistema já que, uma vez que a lista *nesting* seja criada, a mesma pode ser utilizada inúmeras vezes para realizar um dos seus processos.

Como melhorias futuras, um dos principais fatores que foi observado ao longo desse trabalho foi a necessidade de escolher mais de um critério de otimização. Por esse motivo, uma das principais melhorias futuras para o SAD proposto seria transformá-lo em um sistema que possibilita a escolha de múltiplos critérios de otimização. Além disso, o aumento da quantidade de características obtidas através da análise das peças trazidas do CAD deve ser executada, uma vez que essas afetam diretamente o sistema especialista. Cabe lembrar que o aumento da quantidade de características obtidas das peças influenciaria diretamente no sistema especialista e nas regras atribuídas ao mesmo. Por esse motivo, é de suma importância que o módulo de reconhecimento de características seja aprimorado no futuro.

Quanto a melhorias futuras no módulo de extração de sucata, uma possível melhoria é a implementação de um algoritmo responsável por remover pequenas partes do material que não poderão ser utilizadas no futuro e que, automaticamente realizem operações de manufatura que cortassem essas pequenas partes indesejadas do material. No sistema atual, a



análise realizada para a obtenção da sucata não avalia esses pequenos pedaços sobressalentes de matéria, analisando apenas as sobras aproveitáveis geradas como um todo.

## REFERÊNCIAS

- ABRANTES, R. L. DE A. **Problemas de corte com sobras aproveitáveis e eliminação de simetrias**, 20. Sep. 2013. Tese de Doutorado. Universidade de São Paulo.
- AKUNURU, R.; BABU, N. R. A semi-discrete geometric representation for nesting problems. **International Journal of Production Research**, v. 51, n. 14, p. 4155–4174, 2013. Taylor & Francis. Disponível em: <<http://dx.doi.org/10.1080/00207543.2012.751508>>. Acesso em: 13/8/2014.
- AMOAKO-GYAMPAH, K.; ACQUAAH, M. Manufacturing strategy, competitive strategy and firm performance: An empirical study in a developing economy environment. **International Journal of Production Economics**, v. 111, n. 2, p. 575–592, 2008. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925527307001004>>. Acesso em: 4/12/2014.
- ANDRADE, E. L. DE. **Introdução à pesquisa operacional: métodos e modelos para a análise de decisão**. Livros Técnicos e Científicos, 1998.
- ARORA, S.; BARAK, B. **Computational complexity: a modern approach**. Cambridge University Press, 2007.
- AZEVEDO, T. D. A. **Seattrc: Um Sistema Especialista de Apoio a Decisão dos Tipos de Trabalhadores e Rescisões de Contratos**, 2005. Trabalho de Conclusão. UNIVERSIDADE FEDERAL DA BAHIA. Disponível em: <<http://www.egov.ufsc.br/portal/sites/default/files/anexos/6508-6507-1-PB.pdf>>. Acesso em: 26/8/2014.
- BALESTRIN, A. **UMA ANÁLISE DA CONTRIBUIÇÃO DE HERBERT SIMON ÀS TEORIAS ORGANIZACIONAIS**, 2001. UFRGS.
- BANA E COSTA, C. A.; ENSSLIN, L.; CORNÊA, É. C.; VANSNICK, J.-C. Decision Support Systems in action: Integrated application in a multicriteria decision aid process. **European Journal of Operational Research**, v. 113, n. 2, p. 315–335, 1999. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221798002197>>. Acesso em: 26/8/2014.
- BENDOLY, E.; ROSENZWEIG, E. D.; STRATMAN, J. K. The efficient use of enterprise information for strategic advantage: A data envelopment analysis. **Journal of Operations Management**, v. 27, n. 4, p. 310–323, 2009. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0272696308000958>>. Acesso em: 23/12/2014.
- BENFER, R. A.; BRENT, E. E.; FURBEE, L. **Expert Systems (Quantitative Applications in the Social Sciences)**. 1st Ed. ed.SAGE Publications, Inc, 1991.
- BLACK, J. T. **O Projeto da Fábrica Com Futuro**. Porto Alegre: Bookman, 1998.
- BLACK, P. E. Dictionary of Algorithms and Data Structures. Disponível em: <<http://www.nist.gov/dads/>>. Acesso em: 10/11/2014.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization. **ACM Computing Surveys**, v. 35, n. 3, p. 268–308, 2003. ACM. Disponível em: <<http://dl.acm.org/citation.cfm?id=937503.937505>>. Acesso em: 15/1/2015.

BOHANEC, M. What is Decision Support? **Proc. Information Society IS-2001: Data Mining and Decision Support in Action**, p. 86–89, 2001. Disponível em: <[http://www.academia.edu/6376630/Proceedings\\_of\\_the\\_4th\\_International\\_Multi-Conference\\_on\\_Society\\_Cybernetics\\_and\\_Informatics\\_IMSCI\\_2010\\_Vol.\\_1](http://www.academia.edu/6376630/Proceedings_of_the_4th_International_Multi-Conference_on_Society_Cybernetics_and_Informatics_IMSCI_2010_Vol._1)>. Acesso em: 25/8/2014.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, v. 237, p. 82–117, 2013. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025513001588>>. Acesso em: 10/7/2014.

BRINSON, J. A. **Contabilidade por Atividades: uma Abordagem de Custeio Baseado em Atividades**. Brasil: Atlas, 1996.

CANELLIDIS, V.; GIANNATIS, J.; DEDOISSIS, V. Efficient parts nesting schemes for improving stereolithography utilization. **Computer-Aided Design**, v. 45, n. 5, p. 875–886, 2013. Butterworth-Heinemann. Disponível em: <<http://dl.acm.org/citation.cfm?id=2461557.2461907>>. Acesso em: 30/7/2014.

CARVALHO, J. M. V DE; RODRIGUES, A. J. G. A computer based interactive approach to a two-stage cutting stock problem. **Infor**, v. 32, n. 4, p. 243–243, 1994. UNIVERSITY OF TORONTO PRESS. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Computer-Based+Interactive+Approach+to+a+two-Stage+Cutting+Stock+Problem#0>>. .

CHERRI, A.; VIANNA, A. Sobras aproveitáveis para o problema de corte bidimensional. XLV Simpósio Brasileiro de Pesquisa Operacional. **Anais...**, 2013. Natal - RN: Sobrapo. Disponível em: <<http://www.din.uem.br/sbpo/sbpo2013/pdf/irq0188.pdf>>. Acesso em: 15/8/2014.

CHRYSSOLOURIS, G.; PAPAKOSTAS, N.; MOURTZIS, D. A decision-making approach for nesting scheduling: A textile case. **International Journal of Production Research**, v. 38, n. 17, p. 4555–4564, 2000. Taylor & Francis. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/00207540050205299>>. Acesso em: 14/8/2014.

CIMDATA. **Annual CIMdata Market Assessment Report 2005**. USA, 2005.

COSTA, P. D.; MIELKE, I. T.; PEREIRA, I.; ALMEIDA, J. P. A. A Model-Driven Approach to Situations: Situation Modeling and Rule-Based Situation Detection. 2012 IEEE 16th International Enterprise Distributed Object Computing Conference. **Anais...** p.154–163, 2012. IEEE. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6337246>>. Acesso em: 26/8/2014.

CUI, Y.; YANG, Y. A heuristic for the one-dimensional cutting stock problem with usable leftover. **European Journal of Operational Research**, v. 204, n. 2, p. 245–250, 2010. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221709007966>>. Acesso em: 14/8/2014.

DRUZDZEL, M. J.; FLYNN, R. R. Decision Support Systems. **Encyclopedia of Library and Information Science**, 2002. Springer. Disponível em: <<http://www.pitt.edu/~druzdzel/psfiles/dss.pdf>>. Acesso em: 25/8/2014.

ELANCHEZHIAN, C.; SUNDAR, G. S. **Computer Aided Manufacturing**. Firewall Media, 2007.

FERRARY, F. R.; GERHARDT, R.; CANTO DOS SANTOS, J. V. REUTILIZAÇÃO DE CHAPAS DE MATERIAL BRUTO PÓS SISTEMA DE OTIMIZAÇÃO DE PEÇAS EM CHAPAS. XLV Simpósio Brasileiro de Pesquisa Operacional. **Anais...** . p.129–140, 2013. Natal - RN: Sobrapo. Disponível em: <<http://www.din.uem.br/sbpo/sbpo2013/pdf/arf0270.pdf>>. Acesso em: 1/9/2014.

FERRARY, F. R.; GÓMEZ, A. T.; SANTOS, J. V. C. DOS; CHIWIACOWSKY, L. A. MULTITHREADING TABU SEARCH FOR SETUP TIME OPTIMIZATION FOR LIMITED MACHINE MAGAZINES IN THE CELL FORMATION PROBLEM. Simpósio Brasileiro De Pesquisa Operacional. **Anais...** . v. XLVI, 2014. Salvador: Sobrapo.

FERREIRA DE LARA, F.; NEVES GUIMARÃES, M. R. Competitive Priorities and Innovation in SMEs: A Brazil Multi-Case Study. **Journal of technology management & innovation**, v. 9, n. 3, p. 51–64, 2014. Universidad Alberto Hurtado. Facultad de Economía y Negocios. Disponível em: <[http://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718-27242014000300004&lng=es&nrm=iso&tlng=en](http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-27242014000300004&lng=es&nrm=iso&tlng=en)>. Acesso em: 28/12/2014.

FIECHTER, C.-N. A parallel tabu search algorithm for large traveling salesman problems. **Discrete Applied Mathematics**, v. 51, n. 3, p. 243–267, 1994. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0166218X92000331>>. Acesso em: 21/1/2015.

FREITAS, H. M. R.; MACADAR, M. A.; MOSCAROLA, J. Na busca de um método quanti-qualitativo para estudar a percepção do tomador de decisão. **ANPAD, Administração da Informação**, v. Anais do 2, p. 253 – 276, 1996. Disponível em: <[http://www.ufrgs.br/gianti/files/artigos/1996/1996\\_038\\_ENANPAD.pdf](http://www.ufrgs.br/gianti/files/artigos/1996/1996_038_ENANPAD.pdf)>. Acesso em: 15/8/2014.

GENDREAU, M.; LAPORTE, G.; SEMET, F. A tabu search heuristic for the undirected selective travelling salesman problem. **European Journal of Operational Research**, v. 106, n. 2-3, p. 539–545, 1998. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221797002890>>. Acesso em: 27/1/2015.

GHOSH, J. Network Ensembles and Hybrid Systems, tutorial lecture. **IEEE International Conference on Neural Networks**, v. 4, 1997.

GIARRATANO, J. C.; RILEY, G. **Expert systems: principles and programming**. PWS-KENT Pub. Co., 1989.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, v. 13, n. 5, p. 533–549, 1986. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0305054886900481>>. Acesso em: 13/1/2015.

GLOVER, F. W.; LAGUNA, M. **Tabu Search, Volume 1**. Springer Science & Business Media, 1998.

GOLDBARG, M. C.; LUNA, H. P. L. **OTIMIZAÇÃO COMBINATÓRIA E PROGRAMAÇÃO LINEAR**. 2nd Ed. ed. Editora Campus-Elsevier, 2005.

GÓMEZ, A. T.; FERRARY, F. R.; SANTOS, J. V. C. DOS; CHIWIACOWSKY, L. Cell Formation Problem: An Multithreading Tabu Search for Setup Time Optimization for Limited Machine Magazines - A New Solution for a Classical Problem. 11th International Conference on Informatics in Control, Automation and Robotics. **Anais...** . p.332–339, 2014. Disponível em: <<http://www.scitepress.org/DigitalLibrary/Index/DOI/10.5220/0005063403320339>>. Acesso em: 14/1/2015.

GONÇALVES, E. L. Z. **Manufatura Integrada por Computador: sistemas integrados de produção: estratégia, organização, tecnologia e recursos humanos**. Rio de Janeiro: Ed. Campus, 1995.

GRAHAM, R. L.; GROTSCHER, M.; LOVASZ, L. **Handbook of combinatorics**. 1st editio ed. North Holland, 1995.

GRAY, P. **Decision Support and Executive Information Systems**. Englewood Cliffs, New Jersey: Prentice Hall, 1994.

GROOVER. **CAD/CAM: Computer-Aided Design and Manufacturing**. Pearson Education, 2006.

GROOVER, M. P. **Automation, Production Systems, and Computer-integrated Manufacturing**. Prentice Hall, 2008.

HAIJIREZAIE, M.; MOHAMMAD, S.; HUSSEINI, M.; ABDOLLAHZADEH, A.; KARIMI, B. Modeling and evaluating the strategic effects of improvement programs on the manufacturing performance using neural networks. **African Journal of Business Management**, v. 4, n. 4, p. 414–424, 2010. Disponível em: <<http://www.mendeley.com/catalog/modeling-evaluating-strategic-effects-improvement-programs-manufacturing-performance-using-neural-ne/>>. Acesso em: 27/12/2014.

HARMON, P.; KING, D. **Expert Systems and Intelligent Computer-aided Instruction**. 2nd Ed. ed. Englewood Cliffs, New Jersey: Educational Technology, 1991.

HARRELL, C. R.; GHOSH, B. K.; BOWDEN, R. O. **Simulation Using Promodel**. McGraw-Hill Science/Engineering, 2003.

HENNEMANN, F. A.; SANTOS, J. V. C. DOS; RABELO, R. J.; CURY, J. E. R. Um sistema híbrido de apoio à decisão formado por redes de Petri, simulação e sistema especialista. **Sba: Controle & Automação Sociedade Brasileira de Automática**, v. 17, n. 1, p. 10–23, 2006. Sociedade Brasileira de Automática. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-17592006000100002&lng=en&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592006000100002&lng=en&nrm=iso&tlng=pt)>. Acesso em: 25/8/2014.

HENZINGER, T. A. The theory of hybrid automata. Proceedings 11th Annual IEEE Symposium on Logic in Computer Science. **Anais...** . p.278–292, 1996. New Brunswick, NJ:

IEEE Comput. Soc. Press. Disponível em:

<<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=561342>>. Acesso em: 15/8/2014.

HENZINGER, T. A.; KOPKE, P. W.; PURI, A.; VARAIYA, P. What's Decidable about Hybrid Automata? **Journal of Computer and System Sciences**, v. 57, n. 1, p. 94–124, 1998.

Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022000098915811>>.

Acesso em: 15/8/2014.

HERRMANN, J. W.; DELALIO, D. R. Algorithms for sheet metal nesting. **IEEE**

**Transactions on Robotics and Automation**, v. 17, n. 2, p. 183–190, 2001. IEEE. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=928563>>. Acesso em: 14/8/2014.

HERTZ, A. Tabu search for large scale timetabling problems. **European Journal of Operational Research**, v. 54, n. 1, p. 39–47, 1991. Disponível em:

<<http://www.sciencedirect.com/science/article/pii/037722179190321L>>. Acesso em: 21/1/2015.

HERTZ, A.; WIDMER, M. Guidelines for the use of meta-heuristics in combinatorial optimization. **European Journal of Operational Research**, v. 151, n. 2, p. 247–252, 2003.

Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221702008238>>.

Acesso em: 15/1/2015.

HIROUCHI, T.; KOSAKA, T. An effective database formation for decision support systems. **Information & Management**, v. 7, n. 4, p. 183–195, 1984. Disponível em:

<<http://www.sciencedirect.com/science/article/pii/0378720684900181>>. Acesso em: 25/8/2014.

JÚNIOR, B. A.; PINHEIRO, P. R.; SARAIVA, R. D.; PINHEIRO, P. C. A HYBRID APPROACH FOR THE NESTING PROBLEM: CASE STUDY ON A TEXTILE

INDUSTRY. **XLV Simpósio Brasileiro de Pesquisa Operacional**, 2013. Disponível em:

<<http://www.din.uem.br/sbpo/sbpo2013/pdf/arq0316.pdf>>. Acesso em: 14/8/2014.

KARP, R. M. **Reducibility Among Combinational Problems, Complexity of Computer Computations**. New York: Plenum, 1972.

LEHNER, P. E.; MULLIN, T. M.; COHEN, M. S. A Probability Analysis of the Usefulness of Decision Aids. AI '89: Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence. **Anais...** p.427 – 436, 1989. Windsor, Ontario, Canada: Elsevier Science. Disponível em:

<[https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article\\_id=1894&proceeding\\_id=1005](https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1894&proceeding_id=1005)>. Acesso em: 25/8/2014.

LINOFF, G. S.; BERRY, M. J. A. **Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management**. 3rd Edition: John Wiley & Sons, 2011.

METAGROUP. **The State of ERP Services**. Stamford, CT., 2004.

MORTON, M. S. S. **Management Decision Systems: Computer-Based Support of Decision Making**. Harvard University Press, 1971.

NAGASHIMA, T.; MATSUOKA, S.; TAWARA, H.; NIWA, F.; OKADA, T. Automatic nesting of ship hulls using the FINEST algorithm. **IHI engineering**, v. 39, n. 1, p. 27–32,

2006. Ishikawajima-Harima Heavy Industries. Disponível em:

<<http://cat.inist.fr/?aModele=afficheN&cpsidt=17559861>>. Acesso em: 14/8/2014.

NEPOMUCENO, V. S.; FONTANA, M. E. Decision Support System to Project Software Management. 2013 IEEE International Conference on Systems, Man, and Cybernetics.

**Anais...** . p.964–969, 2013. IEEE. Disponível em:

<<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6721922>>. Acesso em: 26/8/2014.

NIKOLOPOULOS. **Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems**. 1st Ed. ed.CRC Press, 1997.

NOVAK-MARCINCIN, J.; BARNA, J.; JANAK, M.; NOVAKOVA-MARCINCINOVA, L. Augmented Reality Aided Manufacturing. **Procedia Computer Science**, v. 25, p. 23–31, 2013. Disponível em:

<<http://www.sciencedirect.com/science/article/pii/S187705091301209X>>. Acesso em: 14/8/2014.

OBERG, E.; JONES, F. D. .; RYFFEL, H. H. .; MCCAULEY, C. J.; HEALD, R. M. . **Machinery's Handbook**. 26th Editi ed.Industrial Press, Inc, 2000.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. Dover Publications, 1998.

PEREIRA, I. S. A.; COSTA, P. D.; ALMEIDA, J. P. A. A rule-based platform for situation management. 2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). **Anais...** . p.83–90, 2013. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6523827>>. Acesso em: 26/8/2014.

PHRIDVIRAJ, M. S. B.; GURURAO, C. V. Data Mining – Past, Present and Future – A Typical Survey on Data Streams. **Procedia Technology**, v. 12, p. 255–263, 2014. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2212017313006683>>. Acesso em: 9/7/2014.

POWER, D. J. **Decision Support Systems: Concepts and Resources for Managers**. Greenwood Publishing Group, 2002.

PREJMEREAN, V. Decision Support System. Disponível em:

<[http://www.cs.ubbcluj.ro/~per/Dss/Dss\\_3.pdf](http://www.cs.ubbcluj.ro/~per/Dss/Dss_3.pdf)>. Acesso em: 25/8/2014.

RAGSDALE, C. T. **MODELAGEM E ANÁLISE DE DECISÃO**. São Paulo: Cengage Learning, 2011.

RIBEIRO, H. DA C. E S. **Introdução aos sistemas especialistas**. Rio de Janeiro: Embrapa, 1987.

RICH, E.; KNIGHT, K. **Inteligência Artificial**. 2nd Ed. ed.Rio de Janeiro: McGraw-Hill, 1994.

ROB, P.; CORONEL, C. **Database Systems: Design, Implementation, and Management**. 7th ed. Cengage Learning, 2006.

RODRIGUES, A. G.; GÓMEZ, A. T. Utilização de metaheurísticas para a modelagem de problemas de seleção de partes e “scheduling” em um sistema de manufatura flexível. **Revista Scientia, Unisinos**, 2000. São Leopoldo.

SARCAR, M. M. M.; RAO, K. M.; NARAYAN, K. L. **Computer Aided Design and Manufacturing**. PHI Learning Pvt. Ltd., 2008.

SHINGO, S. **O Sistema Toyota de Produção do Ponto de Vista da Engenharia de Produção**. Porto Alegre: Bookman, 1996.

SHNEIDERMAN, B. **Designing the User Interface: Strategies for Effective Human-computer Interaction**. Pearson/Addison Wesley, 2005.

SILVA, E. M.; SILVA, E. M.; GONÇALVES, V.; MUROLO, A. C. **Pesquisa Operacional: programação linear**. São Paulo: Atlas, 1998.

SIMON, H. A. **A racionalidade do processo decisório em empresas**. v.1, n. 1 ed. Edições Multiplic, 1980.

SMID, P. **CNC Programming Handbook: A Comprehensive Guide to Practical CNC Programming**. Industrial Press, 2000.

SONG, S.-J. Intelligent decision support system for continuous improvement of resource-saving and recycling-conscious manufacturing. Proceedings First International Symposium on Environmentally Conscious Design and Inverse Manufacturing. **Anais...** . p.723–727, 1999. IEEE. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=747705>>. Acesso em: 26/8/2014.

SPRAGUE, R. H.; CARLSON, E. D. **Building effective decision support systems**. Prentice-Hall, 1982.

SPRAGUE, R. H.; WATSON, H. **Decision Support Systems: Putting Theory Into Practice**. USA: Prentice Hall International, 1993.

TAHA, H. A. **Operations Research: An Introduction**. Prentice Hall, 1997.

TALBI, E.-G. **Metaheuristics: From Design to Implementation**. John Wiley & Sons, 2009.

VARGAS, J.; STEMMER, M. R. SISTEMA ESPECIALISTA DE DIAGNOSTICO E TOMADA DE DECISÃO PARA GARANTIA DE QUALIDADE NA PRODUÇÃO DE PLACAS DE CIRCUITO IMPRESSO EM PEQUENAS SERIES. XIX Congresso Brasileiro de Automática, CBA 2012. **Anais...** , 2012. Campina Grande – PA. Disponível em: <<http://www.eletrica.ufpr.br/anais/cba/2012/Artigos/99026.pdf>>. Acesso em: 26/8/2014.

WAGNER, H. M. **Pesquisa operacional**. Prentice-Hall do Brasil, 1986.

WÄSCHER, G.; HAUSSNE, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, n. 3, p. 1109–1130, 2007. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S037722170600292X>>. Acesso em: 7/8/2014.



XIE, S. Q.; WANG, G. G.; LIU, Y. Nesting of two-dimensional irregular parts: an integrated approach. **International Journal of Computer Integrated Manufacturing**, v. 20, n. 8, p. 741–756, 2007. Taylor & Francis. Disponível em: <http://dx.doi.org/10.1080/09511920600996401>>. Acesso em: 14/8/2014.

YI, H.; YUHUI, Q.; GUANGYUAN, L.; KAIYOU, L. A Parallel Adaptive Tabu Search Approach for Traveling Salesman Problems. 2005 International Conference on Natural Language Processing and Knowledge Engineering. **Anais...** . p.796–801, 2005. IEEE. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1598845>>. Acesso em: 21/1/2015.

ZHOU, F.; YANG, B.; LI, L.; CHEN, Z. Overview of the New Types of Intelligent Decision Support System. 3rd International Conference on Innovative Computing Information and Control. **Anais...** . p.267–267, 2008. IEEE. Disponível em: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4603456>>. Acesso em: 26/8/2014.

**ANEXO A - TRABALHOS PÚBLICADOS ATÉ JANEIRO/2015**

## REUTILIZAÇÃO DE CHAPAS DE MATERIAL BRUTO PÓS SISTEMA DE OTIMIZAÇÃO DE PEÇAS EM CHAPAS

**Felipe Rodrigues Ferrary**

PIPCA - Unisinos

Av. Unisinos, 950 - São Leopoldo - RS

feliperferrary@gmail.com

**Raul Gerhardt**

Fundação Escola Técnica Liberato S. V. da Cunha

Rua Inconfidentes, 395 - Novo Hamburgo - RS

raul\_rag@hotmail.com

**José Vicente Canto dos Santos**

PIPCA - Unisinos

Av. Unisinos, 950 - São Leopoldo - RS

jvcanto@unisinos.br

### RESUMO

O trabalho a seguir tem por objetivo a implementação de uma ferramenta automatizada com a finalidade de reutilizar chapas de material bruto após serem submetidas ao processo de *nesting*, processo onde peças são alocadas dentro de uma chapa plana visando minimizar a perda de matéria prima, seguido de um processo de usinagem. É calculada de forma automatizada a porção de material da chapa que não foi utilizada durante o processo e será possivelmente descartada em forma de sucata. Como resultado, é oferecida ao usuário a possibilidade de utilizar dois métodos de extração: um método envolvendo menor processamento e resultados aproximados e outro com alto nível de processamento, porém com resultados precisos. Após a extração do perfil que representa a chapa de sucata é possível reutilizar e gerenciar o estoque de chapas, dessa forma, uma vez que essa chapa se encaixe ao projeto é possível ao programador CAM reutilizá-la.

**Palavras-chave.** *Nesting*; Reutilização de Chapas; CAM.

**Área Principal.** AD&GP - PO na Administração & Gestão da Produção

### ABSTRACT

The following study aims implementing a feature that saves the scrap area of raw material sheets submitting them to a nesting process, process used in order to minimize the amount of scrap in a flat raw material sheet, followed by a machining process. In this study the part of the material in a raw sheet that was not used during the nesting process is calculated and defined as scrap. Users can choose between two different methods to extract the scrap profile. One method involves a low processing level and outputs approximate results. The other involves a high processing level, however the method outputs truthful results. After finishing the profile extraction that represents the new scrap sheet, it is possible to reuse these sheets and manage the sheet inventory. Once one of the sheets in the inventory fits in a new project, it is possible for the CAM programmer to reuse it.

**Key words.** *Nesting*; Save Off-Cut/Scrap; CAM;

**Main area.** AD & GP - OR in Administration & Production Management

## 1. Introdução

Uma questão conhecida pelos administradores dos processos estratégicos da manufatura, a perda de matéria prima, é um problema que atinge diretamente o custo da produção de um produto. Segundo Brinson (1996), perdas e desperdícios são constituídos pelas atividades que não agregam valor e que resultam em gastos de tempo e recursos, além de adicionarem custos desnecessários aos produtos. Atividades que não agregam valor são as que podem ser eliminadas sem que haja deterioração no desempenho da empresa (custo, função, qualidade e valor agregado). Empresas que buscam manterem-se líderes de mercado necessitam de sistemas automatizados e eficazes para evitar perdas, assim produzindo mais com um custo reduzido.

Quando as perdas são relacionadas à manufatura assistida por computador, pode-se claramente afirmar que empresas de grande porte investem em ferramentas automatizadas que evitam essas perdas. No processo de manufatura de peças que possuem chapas de metal ou madeira como matéria prima, o processo de *nesting* é o comumente adotado para se reduzir a quantidade de matéria prima utilizada e, por conseguinte o custo do processo como um todo. Este processo pode ser descrito como um problema clássico, onde peças são alocadas em uma chapa plana visando minimizar a perda de matéria prima (Rajasekhar e Ramesh, 2013). Este é um problema conhecido (Wascher, 2007) e pode frequentemente ser encontrado nas indústrias de metal-mecânica, vidro, construção naval, fabricação de móveis (Xie, 2007) ou até mesmo na indústria têxtil (Chryssolouris, 2000) com a finalidade de evitar perdas devido à sua eficácia em alocar o maior número ou a quantidade desejada de peças dentro de uma chapa utilizando a menor quantidade de material possível.

Devido à importância desse processo, ao longo dos anos muitos artigos estudaram algoritmos em busca da forma ótima de se alocar as peças em uma chapa. Todavia, em muitos processos, de acordo com as características do projeto, a chapa não é totalmente populada e poucos artigos abordam a análise da chapa após a otimização. Sendo assim, é necessário um sistema que eficientemente extraia a porção de material determinada como sucata e administre o estoque de chapas de uma linha de produção. Seguindo o conceito de Brinson (1996), a implementação de uma ferramenta automatizada de reconhecimento e reutilização de chapas proporciona ao usuário, não só a economia de tempo do programador CAM, mas também a economia de matéria prima; por conseguinte, evita o aumento dos custos de produção.

Ao longo das últimas décadas, muitas análises e propostas já foram realizadas para minimizar as perdas no processo de *nesting*. Os estudos vão das análises mais breves e simplificadas, onde apenas formas retangulares são analisadas (Chryssolouris, 2010) até as análises mais complexas, utilizando meta-heurísticas em conjunto com algoritmos conhecidos como o algoritmo genético (Hopper, 2001).

Embora pesquisas sejam comumente encontradas na área de otimização para alocação de peças, as pesquisas na análise de pós-otimização, por se tratar de uma área de pesquisa recente, deixam a desejar. Na bibliografia atual, encontramos estudos relacionados à área que abordam os mais distintos ramos, tais como: o caráter econômico da minimização da sucata (Lee, 2000); auxílio na escolha da matéria prima (Jayavel, 1993) ou ainda propostas de modelos para determinar a melhor estratégia maximizando o lucro em indústrias que reutilizam a sucata gerada (Tang, 2009). Todavia, estudos que analisem e viabilizem a reutilização da porção que não foi utilizada em processos de manufatura envolvendo chapas como matéria prima não são comumente encontrados, sendo esta uma das motivações deste trabalho.

## 2. Problema Enfocado

Neste trabalho trata-se principalmente da reutilização do material após o processo de usinagem, o que não é frequente na área, automatizando a extração do perfil de material bruto considerado sucata ou sobra. Além disso, em muitos casos reais, como no *software* Alphacam® utilizado na etapa de testes, o banco de dados onde os perfis de sucata são armazenados aceitam apenas materiais retangulares e não perfis com diferentes formatos geométricos. Portanto, buscando uma interface que melhor represente ao usuário os resultados obtidos após o processo

de *nesting*, a pesquisa busca obter resultados reais de material bruto que restariam em uma chapa – independentemente do tipo do material - após esse ser submetido ao processo de *nesting*.

*Nesting* pode ser definido como o processo que busca agrupar o maior número ou o número desejado de peças dentro de uma chapa plana de material bruto. Esta é uma etapa importante no processo de manufatura, uma vez que o sistema é responsável por otimizar matematicamente diversas distribuições possíveis das peças selecionadas, inserindo-as da forma mais eficiente dentro dessa chapa a fim de escolher a distribuição que melhor represente os valores pré selecionados pelo usuário. O sistema de *nesting* pode reduzir o tempo de configuração de máquina, reposição de chapas e reduzir os custos com materiais. O operador realiza apenas uma configuração ao invés de duas ou mais (Herrmann, 2001). O sistema utiliza como base os contornos de peças, de forma que seja possível através desses contornos delimitar os limites máximos da peça e também a melhor forma de colocá-las na chapa para a redução de perdas de material. Após a parametrização do mesmo, o processo de *nesting* é iniciado e assim que finalizado o cálculo de alocação das peças o resultado é apresentado ao usuário, a figura 1 ilustra a aplicação de um sistema de *nesting*.

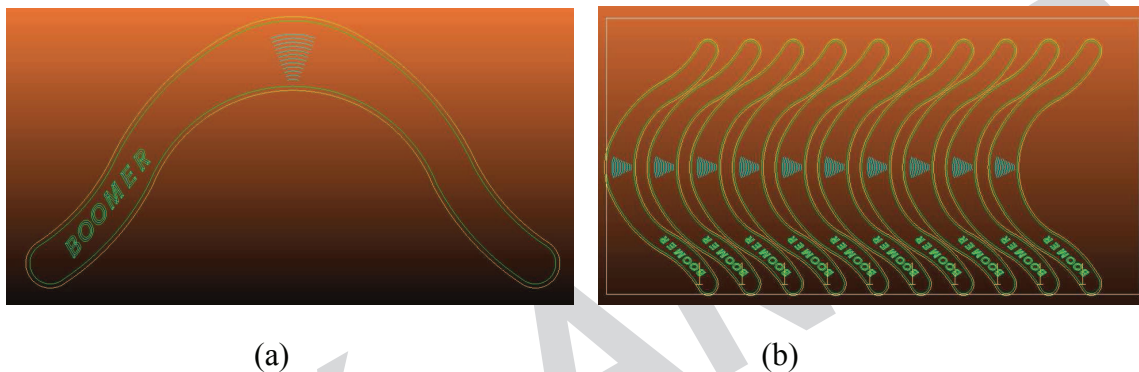


Figura 1. Peça submetida ao processo de *nesting* (a) e seu respectivo resultado (b).

Neste estudo, serão ignoradas as sobras internas de materiais devido à complexidade de seu cálculo. Além disso, a terceira dimensão (plano Z) nos desenhos criados através de *softwares* de desenho assistido por computador (CAD) será ignorada tendo em mente que o sistema de *nesting* se trata basicamente de uma alocação de peças criadas a partir de duas dimensões (2D) nos planos X e Y.

### 3. Sistema proposto

O sistema de reutilização de sobras de chapas tem como objetivo identificar, de forma automática, os segmentos de chapas classificados como sucata e separá-los para uma possível reutilização criando um perfil 2D que represente a sucata. Dessa forma, boa parte da matéria prima é salva, facilitando o processo de reutilização.

Assim como na problemática de alocação de peças, a complexidade computacional envolvida na busca pela porção de sucata no material também pode ser determinada como um problema NP-difícil (Garey, 1979), uma vez que a complexidade computacional envolvida na análise das infinitas possibilidades de peças alocadas é alta. Usualmente, *softwares* recebem um algoritmo de baixa complexidade e retornam resultados que extraem apenas restos de material em formato retangular, sendo boa parcela dos resultados apresentados errôneos, visto que os algoritmos consistem apenas em encontrar o maior ponto onde não há caminhos de ferramenta no eixo X e criar um retângulo até o ponto superior direito da chapa. A Figura 2 apresenta resultados obtidos, consecutivamente, por um algoritmo de baixa complexidade (a) e outro de alta complexidade (b). Pode-se notar claramente que o resultado de baixa complexidade, ignora completamente a área superior aos caminhos de ferramenta – canto superior esquerdo. O resultado esperado deveria considerar toda a área por onde o inserto ou *laser* não extraiu material,

ou seja, a área onde o material continua intacto e habilitado para ser reutilizado. Os resultados esperados devem levar em consideração o perfil 2D de todas as peças e não apenas o perfil da peça mais próxima à borda direita, apresentando a área resultante sem ignorar porções relevantes do material.

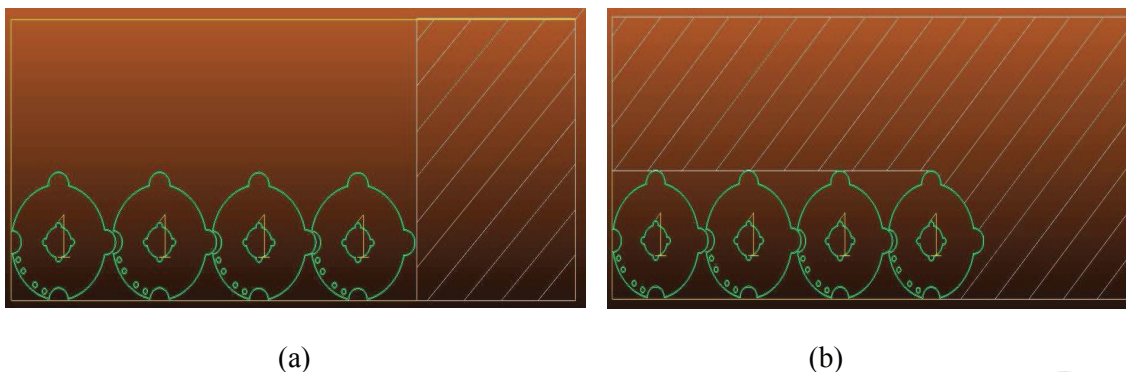


Figura 2 - Resultados obtidos com algoritmos de baixa e alta complexidades

Para a obtenção dos perfis de sucata esperados foram desenvolvidos dois métodos de cálculo: Retangular e de Varredura. O primeiro, Método Retangular, consiste em criar um retângulo na parte externa de cada caminho de ferramenta. Esses retângulos representam os pontos máximos e mínimos de X e Y no plano cartesiano por onde essa ferramenta passou. Assim, esses pontos serão adotados como pontos aproximados a área de usinagem. O segundo, Método de Varredura, consiste em criar 2 coleções de pontos e com base em uma tolerância selecionada pelo usuário, realizar um busca ponto a ponto dentro da chapa previamente otimizada pelo sistema de *nesting*.

### 3.1 Método Retangular

O Algoritmo consiste em encontrar os pontos máximos e mínimos ao longo dos eixos X e Y por onde a ferramenta irá remover material com base nos caminhos de ferramenta aplicados a uma chapa. Para tanto, o aplicativo faz uma análise de todos os elementos a fim de armazenar os valores máximos e mínimos no sistema de coordenadas X e Y no plano cartesiano por onde a chapa foi cortada. Contudo, os pontos de máximo e mínimo desses caminhos de ferramenta não necessariamente representam o local exato onde o material foi extraído e sim, um ponto aproximado, visto que os pontos de máximo e mínimo apenas criam virtualmente um retângulo que engloba o caminho de ferramenta externamente. Os pontos de máximo e mínimo extraídos das peças contidas nessa chapa são representados por circunferências brancas na Figura 3.

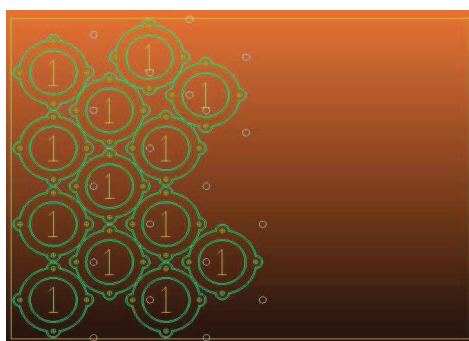


Figura 3 - Pontos de máximo e mínimo de cada geometria

Todavia, após a extração dos pontos em questão, os mesmos precisam passar por um processamento de filtragem, eliminando pontos considerados como indesejáveis. Pontos serão

determinados como indesejáveis para o processo caso os mesmos sejam pertencentes a uma peça que possui outra peça localizada entre ela e o limite entre peças e sucata. Após esta etapa, restarão apenas os pontos localizados entre a peça e a porção de material considerada sucata. Para isso, foi criado um algoritmo que lê todos os pontos criados nas peças e elimina os mesmos caso não sejam relevantes (pontos que não pertencem a fronteira entre peças e sucata) para o processo de extração do perfil de sucata. No processo, são analisados dois parâmetros, sendo que, caso ambos sejam verdadeiros o ponto é eliminado do conjunto. O primeiro parâmetro a ser verificado é a existência de outra peça alocada entre o ponto e extremidade da chapa onde a sucata está localizada. Caso esse parâmetro seja falso é suficiente para concluir que o ponto pertence à fronteira entre a porção de material que será desperdiçado e as peças alocadas na chapa. No entanto, caso essa afirmação seja verdadeira, e haja uma peça alocada entre a extremidade da chapa e o ponto, uma segunda verificação será realizada a fim de verificar se o ponto se encontra entre o valor máximo e mínimo de Y da peça em análise. Caso o ponto esteja alocado acima ou abaixo dos valores citados acima, o mesmo é considerado válido e é utilizado na criação do perfil da sucata. Em posse dos vetores contendo as coordenadas dos pontos filtrados é possível iniciar a rotina responsável por criar o perfil da sucata. Considerando que no algoritmo o perfil é criado a partir do ponto que possui menor valor em Y (parte inferior da chapa), é necessário ainda ordenar os vetores de forma que os índices apresentem-se em ordem crescente em relação aos valores de Y. Contudo, mesmo após ordenar os valores em ordem crescente para iniciar o desenho do perfil da sucata, é necessário verificar se o ponto inicial está alocado fora das extremidades da chapa, caso esteja, o algoritmo mantém o valor ponto inicial no eixo X e projeta o ponto inicial em Y para coincidir com a parte inferior da chapa. Após definido o ponto inicial, o mesmo é interligado aos demais pontos criando como resultante o perfil aproximado que define as fronteiras entre as peças e a sucata. Com a obtenção desse perfil, o último passo é interligá-lo à extremidade superior do material que, por conseguinte deve ser interligada às arestas inferior e superior da chapa e por último, ao ponto inicial seguindo o perfil da chapa bruta. Resultando, por exemplo, no perfil apresentado em branco na Figura 4.

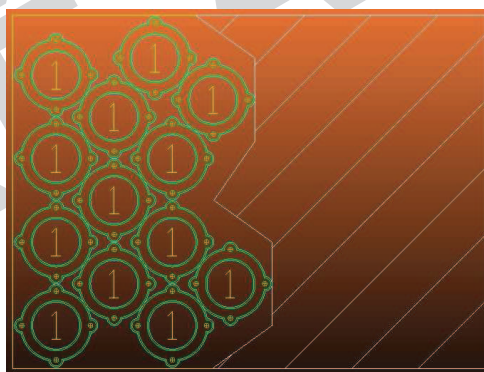


Figura 4 - Perfil final obtido utilizando o método retangular

É importante ressaltar que no sistema retangular, embora os resultados apresentados não sejam fielmente o perfil de sucata real, o processamento é realizado de forma rápida, dando ao usuário uma resposta imediata.

Os caminhos de ferramenta contidos nas chapas são criados a partir de diversos elementos, sendo tais elementos arcos e linhas. A análise realizada pelo método retangular diferencia arcos de linhas para extrair o ponto com maior valor em Y e X. A análise em linhas basicamente consiste em analisar todos os elementos, partindo do primeiro elemento do caminho de ferramenta e comparar os pontos inicial e final de cada elemento ao maior valor já obtido, armazenando assim sempre o maior valor. Dessa forma, após a análise ser finalizada, a combinação das coordenadas de maior valor em X e Y no plano cartesiano é utilizada para obter os pontos máximos. O processo para obter os menores valores é o mesmo, todavia durante o

processo onde os elementos são comparados, o menor valor é armazenado ao contrário do processo descrito acima. As quatro arestas do retângulo que engloba virtualmente o caminho de ferramenta podem ser obtidas utilizando as combinações de coordenadas.

Quando a análise é realizada em arcos, cálculos trigonométricos e vetoriais devem ser realizados para que seja possível fracionar o arco em diversas porções e analisar seus pontos individualmente, tornando assim a análise desses elementos mais complexa e exigindo maior tempo de processamento de dados. O algoritmo consiste em alocar o arco a ser analisado em um círculo trigonométrico de raio e centro iguais aos do arco em questão e dessa forma criar dois vetores, sendo que o primeiro vetor tenha como origem o centro do círculo trigonométrico e seu ponto final seja o ângulo inicial do círculo trigonométrico (zero grau). O segundo vetor assim como o primeiro possui sua origem no centro do círculo trigonométrico, porém seu ponto final é análogo ao ponto inicial do arco.

Após a criação desses vetores é possível calcular o ângulo interno entre ambos através do produto escalar dos mesmos. O ângulo obtido representa não só o ângulo inicial do arco a ser analisado em relação ao ângulo zero do círculo trigonométrico, mas também o ponto inicial onde o arco deve começar a ser fracionado pelo algoritmo. O ângulo é obtido utilizando a equação contida na Figura 5, considerando que  $U$  e  $V$  são vetores não nulos em  $R^2$  ou  $R^3$  e  $\theta$  é o ângulo entre estes vetores. (Anton, 2008).

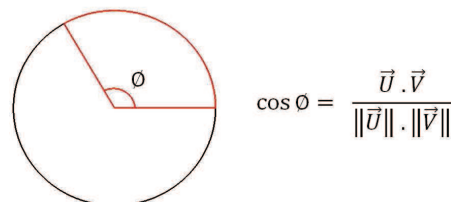


Figura 5 - Obtenção do ângulo entre dois vetores

Através do ângulo encontrado, inicia-se o processo de fracionamento do arco tendo como incremento o valor selecionado pelo usuário no início do processo. Após o final de cada incremento, os pontos final e inicial dessa secção de arco são calculados utilizando as identidades triangulares em conjunto com aplicações de trigonometria. Uma vez que esses pontos sejam obtidos é possível analisar os pontos inicial e final da secção do arco, e assim, de forma similar ao processo aplicado na análise de linhas, comparar esses valores aos demais pontos inicial e final de todas as secções de arco, tendo como produto final o maior ou menor ponto em X e Y de acordo com a comparação aplicada no algoritmo.

### 3.2 Método de Varredura

Diferentemente do método retangular, no método de varredura o algoritmo não busca apenas pelos pontos de máximo e mínimo, mas todos os pontos exatos por onde o caminho de ferramenta percorreu. Para realizar essa análise, é necessário segmentar todos os caminhos de ferramenta com base em uma tolerância pré-selecionada pelo usuário no início do processo. Esse método é afetado diretamente pela tolerância e de acordo com esses valores são criados milhares ou até mesmo milhões de segmentos de caminho de ferramenta, que por sua vez, exigem um grande nível de processamento e que por ventura podem criar problemas relacionados à sobrecarga de variáveis de programação (*Overflow*). A fim de evitar essa espécie de problemas, o algoritmo de varredura é iniciado coletando todos os caminhos de ferramenta contidos na chapa e analisando seus pontos máximos e mínimos.

Depois de finalizado o processo descrito acima, todas as peças são analisadas novamente para verificar se atendem a dois parâmetros. Caso ambos os parâmetros sejam verdadeiros, o caminho de ferramenta será eliminado não afetando o processo. O primeiro parâmetro consiste em analisar se há outro caminho de ferramenta alocado entre a peça em análise e a extremidade da chapa ao longo do eixo X. Caso esse parâmetro seja verdadeiro, o



caminho de ferramenta possui uma grande possibilidade de não afetar o processamento já que pode estar alocada entre outras peças e não entre a porção de material entre peças e sucata.

Com base na peça que teve o primeiro parâmetro verdadeiro, o segundo parâmetro consiste em analisar se:

- O ponto **mínimo** no eixo Y do caminho de ferramenta em análise é **superior ou igual ao ponto mínimo** no eixo Y da peça que passou pelo primeiro parâmetro.
- O ponto **mínimo** no eixo Y do caminho de ferramenta em análise é **inferior ou igual ao ponto mínimo** no eixo Y da peça que passou pelo primeiro parâmetro.
- O ponto **máximo** no eixo Y do caminho de ferramenta em análise é **superior ou igual ao ponto mínimo** no eixo Y da peça que passou pelo primeiro parâmetro.
- O ponto **máximo** no eixo Y do caminho de ferramenta em análise é **inferior ou igual ao ponto de máximo** no eixo Y da peça que passou pelo primeiro parâmetro.

Caso todos os parâmetros acima sejam atendidos, a peça sendo analisada será eliminada da coleção, uma vez que o caminho de ferramenta está localizado em uma posição que não influenciará no cálculo, evitando assim processamento desnecessário. Depois de eliminados os caminhos de ferramenta desnecessários, os elementos contidos nos caminhos de ferramentas devem ser seccionados com base na tolerância pré-selecionada pelo usuário. Diferente do método retangular, onde apenas arcos precisam ser rigorosamente analisados e linhas possuem uma análise superficial já que apenas são coletados os valores máximos e mínimos dos caminhos de ferramenta, no método de varredura arcos e linhas devem ser seccionados e analisados ponto a ponto.

Com base na equação da reta obtida, primeiramente é analisado o tipo de reta e então a reta é classificada como reta horizontal, vertical e reta inclinada. Determinar o tipo da reta é necessário, uma vez que o algoritmo utilizado para o seccionamento da reta percorre a linha do seu ponto inicial até o seu ponto final e o tipo de incremento utilizado para percorrer a reta é distinto para os três tipos de reta. O tipo de reta é determinado através do seu coeficiente angular e de seu termo independente. Retas do tipo horizontais são incrementadas e seccionadas ao longo do eixo X já que seu valor de Y é constante. Por outro lado, retas do tipo vertical tem apenas o valor de Y incrementado e seccionado uma vez que o valor X é constante. Retas do tipo inclinado tem o valor incrementado ao longo de X, todavia seu valor de Y será alterado de acordo com a equação da reta. A análise realizada em elementos do tipo arco é similar à esplanada no método retangular.

Após todos os pontos serem seccionados e as informações de todos os pontos serem armazenadas, a varredura é iniciada. A varredura se inicia na parte inferior da chapa e é realizada de forma que a busca se prolongue até a parte superior da chapa. Para cada vez que a busca é incrementada em direção à extremidade superior da chapa e o valor de Y é incrementado com a tolerância selecionada pelo usuário, uma busca longitudinal ao longo do eixo X é realizada. Essa busca tem início no ponto no eixo X correspondente à extremidade da peça mais próxima onde a sucata está localizada (já que não é necessário analisar a área já conhecida como sucata, assim evitando um grande número desnecessário de interações e por conseguinte diminuindo os índices de *overflow*). Iniciando a análise nesse ponto, cada ciclo é incrementado de forma que o ponto a ser analisado seja projetado em direção das peças aplicadas a chapa.

Prosseguindo, depois de cada incremento ao longo do eixo X é verificado se o ponto atual coincide com algum dos segmentos de caminho de ferramenta criados previamente. Caso o ponto em análise possua o valor de Y entre o valor inicial e final do segmento do caminho de ferramenta, então uma segunda verificação é realizada. A segunda verificação irá certificar-se de que o ponto em análise não apenas coincide ao longo do eixo Y com o segmento de caminho de ferramenta, mas também coincide com a posição em X. Para ser considerado o ponto desejado, o valor do ponto no eixo X deve ser superior ou igual ao ponto inicial do segmento - subtraindo o valor da tolerância pré-estabelecida pelo usuário e também o ponto deve ser inferior ou igual ao

ponto final do segmento somado ao valor da tolerância pré-estabelecida pelo usuário. O valor de tolerância selecionado pelo usuário influencia diretamente o resultado obtido, além disso, evita que o ponto próximo ao caminho de ferramenta não seja encontrado. Em casos onde nenhum segmento de caminho de ferramenta se encaixa nos parâmetros acima, é determinado que não existem peças ao longo do trecho de X e portanto o ponto para criação da sucata é a extremidade da chapa. As circunferências brancas na Figura 6 (a) representam cada ponto que já foi processado e a Figura 6 (b) o resultado obtido através desse método.

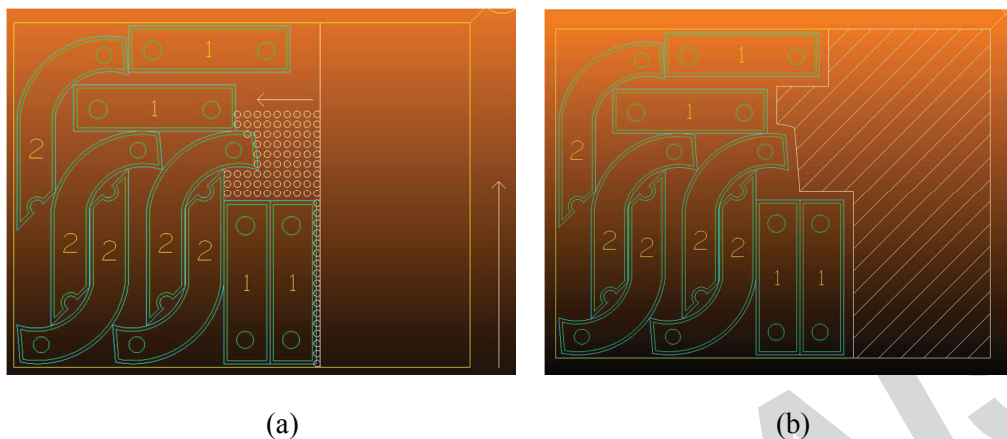


Figura 6 – Representação gráfica da busca e resultado obtido pelo método de varredura

Uma vez que seja encontrado o ponto que coincide com os demais, os pontos inicial e final do segmento do caminho de ferramenta são armazenados para serem posteriormente utilizados na criação do perfil que representa a sucata. Depois de concluída a análise de todos os pontos, os mesmos são interligados aos demais pontos e as extremidades da chapa, dessa forma criando o perfil de sucata que pode ser reutilizado para um processo futuro.

#### 4. Testes e resultados

Para analisar ambos os métodos foram utilizados um número total de 150 chapas de *nesting* com tipos distintos de caminho de ferramenta. Como ferramenta de teste, ambos os algoritmos foram aplicados ao *software* de manufatura assistida por computador Alphacam©.

Os resultados obtidos foram classificados de acordo com a qualidade do perfil de sucata obtido. Os níveis de qualidade foram classificados como: Excelente; Bom; Regular e Insatisfatório; descritos a seguir.

- Excelente: Perfil de sucata é extraído completamente, incluindo pequenas áreas.
- Bom: Perfil de sucata é extraído completamente, todavia pequenas porções de material são ignoradas.
- Regular: Perfil de sucata é extraído parcialmente, perdendo parte relevante do material.
- Insatisfatório: Perfil é extraído de forma mecanicamente impossível ou erro é causado durante a execução do algoritmo.

A fim de testar a eficiência do método retangular, o mesmo foi aplicado as 150 chapas criadas para teste. Após todas as chapas serem submetidas ao método, os resultados foram classificados em níveis de desempenho. Os resultados obtidos através dos testes podem ser observados na Tabela 1. Assim como esperado, peças que possuem formas livres fazem com que o resultado do método retangular ignore uma porção relevante de material, por outro lado o método se mostrou muito eficaz e com tempo de resposta excelente para peças simétricas ou de natureza retangular assim como portas, janelas e degraus de escada. Além disso, assim como

demonstrado pela Figura 7, o método se mostrou eficaz quando a chapa possui uma grande quantidade de caminhos de ferramenta pequenos em relação à chapa.

Tabela 1 – Avaliação dos resultados obtidos através do método retangular

|                |        |
|----------------|--------|
| Excelente      | 85.71% |
| Bom            | 9.52%  |
| Regular        | 2.86%  |
| Insatisfatório | 1.90%  |

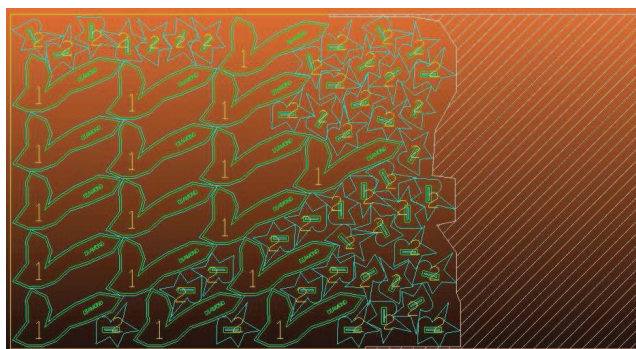


Figura 7 - Método retangular aplicado a chapa com peças pequenas

Finalizada a rotina de testes utilizando as mesmas chapas aplicadas ao método retangular, os resultados obtidos com o Método de Varredura foram classificados conforme a Tabela 2.

Tabela 2 - Avaliação dos resultados obtidos através do método de varredura

|                |        |
|----------------|--------|
| Excelente      | 85.45% |
| Bom            | 7.29%  |
| Regular        | 1.81%  |
| Insatisfatório | 5.45%  |

O nível de resultados insatisfatórios está diretamente ligado ao alto nível de processamento exigido por esse método, já que muitas vezes erros ligados ao sistema operacional do usuário causam problemas de memória que são dificilmente tratados utilizando, nesse caso, *Visual Basic for Application*. Embora ainda seja necessário reduzir ao máximo o nível de erros, os mesmos foram tratados de forma que uma mensagem de erro é mostrada ao usuário solicitando que entre em contato com o suporte técnico. Esse tipo de erro pode ser facilmente corrigido restringindo que o usuário insira valores de tolerância muito pequenos em relação ao tamanho da chapa. Além disso, durante os testes foi observado que o tempo de processamento é elevado de forma significativa quando o caminho de ferramenta contém um número elevado de elementos do tipo arco. Todavia, elementos do tipo linha possuem um resultado preciso e são processados com mais velocidade. O método se mostrou extremamente eficaz para peças que possuem formas livres, criando um perfil de sucata real ou muito próximo quando comparado à chapa que será usinada. Além disso, foi constatado que o método possui grande eficácia para chapas que possuem peças grandes ou uma pequena quantidade de peças.

Durante a rotina de testes aplicados ao algoritmo de remoção de caminhos de ferramenta desnecessários, o principal elemento a ser analisado foi o número de peças que foram eliminados para evitar processamento desnecessário. Aplicando o algoritmo a 21 chapas com características distintas foram obtidos os resultados apresentados na Tabela 3.

Tabela 3 - Resultados do algoritmo de remoção de peças

| Redução de peças feita pelo algoritmo |        |            |       |        |            |       |        |            |
|---------------------------------------|--------|------------|-------|--------|------------|-------|--------|------------|
| Antes                                 | Depois | % removido | Antes | Depois | % removido | Antes | Depois | % removido |
| 15                                    | 9      | 40.00%     | 120   | 55     | 54.16%     | 33    | 24     | 27.27%     |
| 15                                    | 11     | 26.66%     | 52    | 10     | 80.76%     | 14    | 12     | 14.28%     |
| 52                                    | 48     | 7.69%      | 5     | 5      | 0.00%      | 6     | 5      | 16.66%     |
| 44                                    | 30     | 31.81%     | 7     | 5      | 28.57%     | 3     | 3      | 0.00%      |
| 30                                    | 6      | 80.00%     | 10    | 3      | 70.00%     | 44    | 33     | 25.00%     |
| 151                                   | 61     | 59.60%     | 14    | 13     | 7.14%      | 7     | 3      | 57.14%     |
| 56                                    | 20     | 64.28%     | 37    | 7      | 81.08%     | 20    | 6      | 70.00%     |

Os resultados variaram de 0% de peças removidas até 81.08% de peças removidas obtendo uma média de 40,10% de peças removidas do processo. Além disso, pode-se observar que quanto maior o número de peças distintas na chapa, menor é o número de peças removidas, uma vez que as peças diferentes tendem a não ser alocadas na chapa pelo processo de *nesting* de uma forma regular. Portanto, processos de *nesting* realizados utilizando apenas um modelo de peça, assim como o apresentado na Figura 8, devem ser os mais afetados pelo algoritmo de redução de caminhos de ferramenta desnecessários.

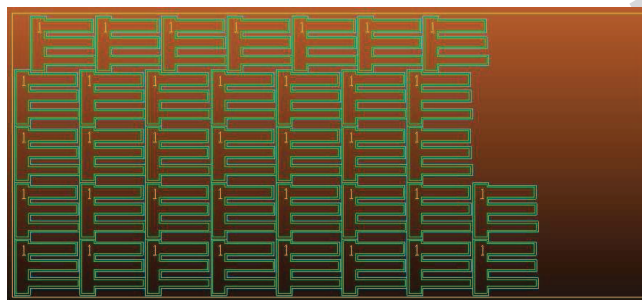


Figura 8 - Chapa que obteve maior índice de remoção de caminhos de ferramenta

Ao longo da análise, foi possível concluir que os algoritmos irão atender necessidades diferentes uma vez que o método Retangular se mostrou eficaz quando aplicado a peças de natureza retangular. Por outro lado, o método de varredura apresentou eficácia quando aplicado a peças com formato livre. Outro contraste importante quando os dois métodos são comparados é que o método Retangular é capaz de analisar uma grande quantidade de caminhos de ferramenta em milissegundos, por outro lado o método de varredura pode levar cerca de minutos para apresentar o resultado quando aplicado a esse tipo de chapa. Portanto, o método mais indicado a ser utilizado para chapas com grande quantidade de peças é o método Retangular. Contudo, embora o método retangular seja mais estável e proporcione ao usuário um resultado mais rápido, o método de varredura continua sendo o mais indicado já que sempre apresenta o resultado condizente com a chapa real. Uma amostra comparativa entre ambos é apresentada na figura 9.

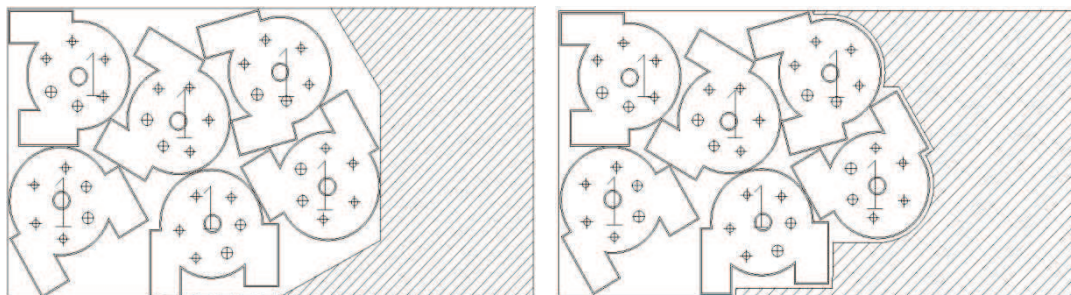


Figura 9 - Consecutivamente, resultados obtidos no método retangular e de varredura

Embora a obtenção dos resultados seja a parte mais complexa deste trabalho, a reutilização das geometrias obtidas utilizando os algoritmos descritos de maior relevância, no aspecto prático, é de maior importância. Independentemente do método utilizado para a obtenção do perfil de sucata, ao final do método, é criado um arquivo de desenho do *software* compatível com o módulo que o usuário possui e esse arquivo é automaticamente armazenado na pasta padrão de informações do *software*. Em conjunto ao arquivo de desenho, também é armazenado uma imagem no formato .jpeg que será utilizada para a representação gráfica do perfil. Além disso, todas as informações mecânicas dessa chapa tais como: espessura, material, comprimento, largura entre outras são armazenadas em um banco de dados. Para reutilizar a chapa, o usuário deve utilizar o visualizador de chapas, ferramenta criada especificamente para essa finalidade.

Utilizando a plataforma de desenvolvimento .NET, a ferramenta de visualização de chapas foi desenvolvida não apenas com o propósito de apresentar visualmente as possibilidades de chapas que o programador CAM possui para serem utilizadas no processo de usinagem, mas também com o propósito de ser a ferramenta onde o usuário pode gerenciar e administrar o seu estoque de chapas. A ferramenta de visualização do banco de dados oferece não só a possibilidade de reutilizar as chapas obtidas através da sucata, mas também a possibilidade adicionar e modificar chapas novas, retangulares, nunca antes utilizadas. Na tela principal, os tipos de chapas são separados de acordo com a sua origem em duas categorias “Novas chapas” e “chapas extraídas de sucata”. Assim que o usuário optar por expandir o nó de visualização selecionando o tipo de chapa que gostaria de utilizar, uma nova lista será apresentada. Essa lista é ordenada de forma que o mesmo material não seja demonstrado duas vezes, já que por questões logísticas o usuário normalmente procura primeiro pelo tipo de material estipulado previamente no projeto a ser realizado no *software* CAM e então procura quais as chapas que melhor se adaptam ao processo. Como representado pela Figura 10, assim que selecionado o material desejado, a listagem completa de chapas disponíveis é apresentada em conjunto com todas as suas características. A imagem de amostra do perfil também é apresentada, ao lado direito, no momento em que o usuário seleciona a chapa.

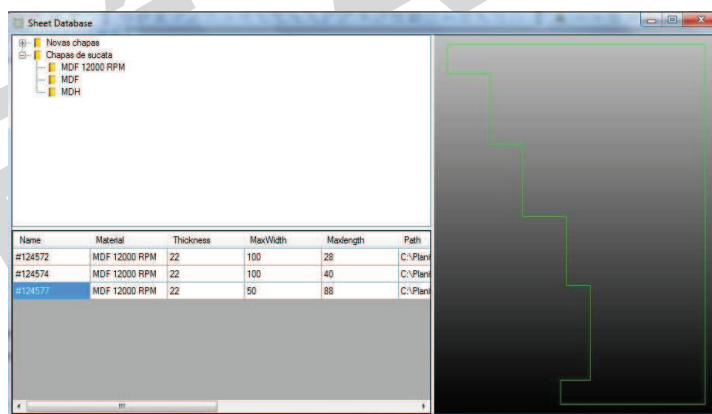


Figura 10 - Interface da ferramenta de reutilização de chapas

## 5. Considerações Finais

Com este trabalho percebeu-se que embora para chegar ao objetivo de extrair os mais distintos contornos de sucata seja necessário realizar a complexa tarefa de analisar as infinitas possibilidades de combinações de geometrias e peças que podem ser criadas pela mente humana, os resultados alcançados foram satisfatórios e economizam importante quantidade de matéria prima e tempo de processo. Além de tornar possível a gestão sobre as chapas de sucata em estoque, cabe lembrar que buscar a redução dos custos do processo de manufatura é um objetivo de todo administrador. No caso deste trabalho houve um avanço importante no sentido de agregar valor aos sistemas computacionais habitualmente utilizados para a gestão de chapas na indústria.

Melhorias futuras devem ser implementadas, tornando mais amigável e eficaz a experiência de utilizar a ferramenta. Além disso, o código deve ser aprimorado a fim de reduzir os erros causados pelo alto processamento do método de varredura, o mesmo deve ser transcrito para a plataforma “C# .NET”, abandonando assim a plataforma *Visual Basic for Application*.

### Referências

- Anton, H., Busby, R.** Álgebra Linear Contemporânea. Porto Alegre: Bookman, 2006. Cap. 1.
- Brinson, J.** Contabilidade por atividades: uma abordagem de custeio baseado em atividades. São Paulo: Atlas, 1996.
- Chryssolouris, G., Papakostas, Mourtzis, D.,** A decision-making approach for nesting scheduling: A textile case, *International Journal of Production Research*, 38:17, 4555-4564, 2000.
- Garey, M.R., Johnson, D.,** Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
- Herrmann, J.W., Delalio, D.R.,** Algorithms for sheet metal nesting, *IEEE transactions on robotics and automation*, 2001.
- Hopper, E., Turton, B.,** An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem, *European Journal of Operational Research* 128, 2001.
- Jayavel, S., Dileep, G. D.,** Scrap minimization in production processes with stochastic outputs, *International Journal of Production Economics* 29, 1993.
- Lee, Y., Wei, C., Chen C., Tsaia C.,** Minimization of scrap and rework costs by process tolerances allocation, *Engineering Optimization*, DOI:10.1080/03052150008941315 (2000)
- Rajasekhar A., Ramesh, N.,** A semi-discrete geometric representation for nesting problems, *International Journal of Production Research*, DOI:10.1080/00207543.2012.751508 (2013)
- Tang J., Liua Y., Fung R. Y.K., Luo X.,** Industrial waste recycling strategies optimization problem: mixed integer programming model and heuristics, *Engineering Optimization*, DOI:10.1080/03052150802294573 (2008)
- Wascher, G., H. Hauber, and H. Schumann.** “An improved typology of cutting and packing problems.” *European Journal of Operational Research* 183: 1109–1130, 2007.
- Xie, S. Q. , G. G. Wang, Y. Liu,** Nesting of two-dimensional irregular parts: an integrated approach, *International Journal of Computer Integrated Manufacturing*, 20:8, 741-756, 2007.

# Cell Formation Problem: An Multithreading Tabu Search for Setup Time Optimization for Limited Machine Magazines

## *A New Solution for a Classical Problem*

Arthur T. Gómez, Felipe R. Ferrary, José V. Canto dos Santos and Leonardo Chiwiacowsky  
*Universidade do Vale do Rio dos Sinos – PIPCA, Av. Unisinos, 950, São Leopoldo – RS, Brazil*

**Keywords:** Tabu Search, Machine Setup Time Optimization, Manufacturing Cell Formation.

**Abstract:** This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search that uses deterministic methods to effectively explore local optimum areas. Manufacturing Cell Formation problem involves the process of analysing parts and groups them according to their similarity. This paper aims maximize the production efficiency, by minimizing the machine setup time in a limited magazine size through the reduction of tool changes by creating clusters of parts that share machining tools and present an initial scheduling based on tool changes reduction. In order to valid the proposed algorithm, the results obtained are compared against other Tabu Search solutions proposed in the literature.

## 1 INTRODUCTION

The process of batching and automate manufacturing processes is an essential process for companies that want to make competitive products. The Batch manufacturing is estimated to be the most common form of production, constituting more than 50% of the total manufacturing activity in the US. In addition, there is a growing need to make batch manufacturing more efficient and productive (Groover, 2001).

In the batching process, Group Technology (GT) has an important role and offers relevant contribution helping to increase the efficiency of production operations and reducing the requirement of facilities (Xu et al., 2014). As part of GT, the Manufacturing Cell Formation problem aims to define an efficient structure to group machines (James et al., 2007).

The Manufacturing Cell Formation problem is a NP-Hard (Spiliopoulos and Sofianopoulou, 2008). Thus, simple heuristics has the propensity to not present satisfactory results. Therefore, several methods making use of artificial intelligence techniques are proposed to solve the problem. Being the manufacturing cell formation a problem of combinatorial optimization, metaheuristic like Ant

Colony (Li et al., 2011; Spiliopoulos and Sofianopoulou, 2008), Genetic algorithms (Xiaodan et al., 2007), Tabu Search (Gómez et al., 2011) are commonly used to find a good solution. On this paper the metaheuristic Tabu Search is used to solve the problem.

Beyond that, another resource that currently receives increasing focus on academic papers is the parallel processing applied to these methods (Fiechter, 1994; He et al., 2005). In such cases, different processors can perform multiple calculations simultaneously.

Another method gaining increased attention is the hybridization of metaheuristics (Kaur and Murugappan, 2008). The success of the hybrid approach comes from the union factor of the strategic advantages of each method in a single metaheuristic; providing a better performance compared to the pure method (Tsai et al., 2009).

In this paper, a variation of the classical implementation of Tabu Search is proposed. Being part of this modification, concepts of hybridization and parallel programming are used to provide solutions near promising regions.

This paper is divided into 5 sections. Section 2 presents concepts of the Manufacturing Cell Formation problem as well as review from its literature. Section 3 presents details about the Tabu

Search proposed to solve the problem, also further details about the multithreading concept used on this method are provided. Section 4 presents the methods used for general testing, as well as the metric used to compare the result against other solution found in the literature. Section 5 presents the conclusions.

## 2 MANUFACTURING CELL FORMATION LITERATURE REVIEW

The cellular formation for a manufacturing system is an application of the Group Technology (GT) – which is a tool to identify similar parts and group them together regarding similarities between them (Selim et al, 1998). The Cell Formation (CF) aims to reduce the setup and flow times – minimizing the inventory and manufacturing lead times (Wemmerlov and Hyer, 1989; Wu et al., 2010).

CF is a binary matrix machines versus parts which reorganizes rows and columns with the intention to group parts (part families) and machines (machine cell). The binary element in the Figure 1 represents the relationship between part and machine indicating:

- 1 – the part  $p$  uses the resource (or machine)  $m$ ;
  - 0 – the part  $p$  can't use the resource  $m$ ;
- See the example below:

|                | Part Number |   |   |   |   |
|----------------|-------------|---|---|---|---|
| Machine Number | 1           | 2 | 3 | 4 | 5 |
| 1              | 0           | 1 | 0 | 1 | 1 |
| 2              | 1           | 0 | 1 | 0 | 0 |
| 3              | 0           | 1 | 0 | 1 | 0 |
| 4              | 1           | 0 | 1 | 0 | 0 |

Figure 1: Relationship between part and machine.

Then, apply the Cluster Identification Algorithm (Kusiak and Chow, 1987) to find any relationship between parts and machines – rearranging the cells and resulting in two cellular formation for manufacturing systems and two part families, as described in the Figure 2.

|      | FP-1 |   | FP-2 |   |   |
|------|------|---|------|---|---|
|      | 1    | 3 | 2    | 4 | 5 |
| CM-1 | 2    | 1 | 1    |   |   |
|      | 4    | 1 | 1    |   |   |
| CM-2 | 1    |   | 1    | 1 | 1 |
|      | 3    |   | 1    | 1 |   |

Figure 2: Cells created by Kusiak Algorithm.

## 3 PROPOSED Tabu SEARCH

Firstly proposed in 1986, Tabu Search is a method that can be used to solve different problems on combinatorial optimization environment (Glover, 1986). According with Glover (1986), Tabu Search may be viewed as a metaheuristic superimposed on another heuristic. This method uses a list in order to forbid movements (“tabu movement”) that drives to solution areas already explored.

Sumanta Basu (2012) brings a brief and updated literature review about the Tabu Search being used as a problem resolution tool related to combinatorial area. On this paper, the main methods used to perform movements from one solution to another movements used to generate the neighbourhood, Tabu List size and type, search intensification and diversification, aspiration criteria as well as a couple of other relevant details pointed along the article.

The different methods overviewed by this article are evaluated qualitatively (through the quality of the solution) as well as quantitatively (by checking the number of times this method is used in the studied articles). The methods employed in the proposed Multithreading Tabu search take in account of the methods pointed as the most efficient and appropriate to the resolution of combinatorial problems.

The solution quality is evaluated using by metric named “effective clustering” (Kumar and Chandrasekharan, 1990). This metric is represented by the equation(1).

$$\tau = \frac{e - e_0}{e + e_v} \quad (1)$$

In the given equation,  $e$  represents the number of 1's in the given matrix,  $e_v$  represents the number of voids (number of 0's inside the clusters in the main diagonal)  $e_0$  is the number of exception (the number of 1's outside the main diagonal).



### 3.1 Initial Solution

As initial solution, cluster identification is used. This algorithm is an iterative process that selects rows and columns in a matrix in order to simultaneously create Family Parts and Machine Clusters. Kusiak and Chow (1987) have initially proposed an algorithm that identifies the similarity between manufacturing processes and creates separable groups. This method uses a binary part-machine incidence matrix  $A = [a_{ij}]$  and decomposes  $A$  into sub matrices  $A_1, A_2, A_n$ . Each sub matrix can be defined as a machine cell.

However, once this paper aims to control and reduce the number of tool changes in a limited magazine, it is necessary taking in account of the machine magazine limit. For this reason, the algorithm used on this paper to generate the initial solution is a method proposed by Gómez (Gómez, 1996) because this method considers the similarity of tools used in the manufacturing process, as well as the machine magazine limit.

Despite this kind of method hardly returns the optimal solution, it is frequently used because it points to an optimum solution. Usually an exact algorithm points to a local optimum solution instead of a global one. However, this algorithm can in some particular cases point to the optimal solution as well as it can point to a near-optimal solution that can be considered an acceptable result (Black, 2005).

### 3.2 Search Intensification

Proposed by (Croes, 1958) as a method to solve the travelling salesman problem, the 2-opt is widely used to modify a current solution and generate a new neighborhood until the stop criteria requirements are met (Lim, Yong, Ramli and Khalid, 2011). Also, it is used as a local search method in many other combinatorial problems (Kothari and Ghosh, 2013; Hasegawa, Ikeguchi and Aihara, 1997).

The 2-opt movement occurs through the removal of two non-adjacent parts or machines from a cluster. After the removal of these components, those parts or machines have their position exchanged thus preventing the need of perform an evaluation of a subgroup again. In addition to, 2-opt movement is also used to exchange the families orders, thus changing the current scheduling. This process helps in reducing the number of tool change between Family parts, consequently reducing the setup time.

The intensification occurs due the non-substantial change of only two parts or machines,

thus creating a solution near of the current solution. Considering it is a simple movement, consequently it has a low computational complexity  $\sim \theta(n^2)$  for 2-opt algorithm (Croes, 1958). For this reason, this method can be commonly find in Tabu Search implementations. In this paper, the 2-opt movement is used to generate 20 neighbours and potential solution candidates.

Another mehod used in the implementation and on search intensification is a column insertion method. In this method, as initial step an part or machine is randomly selected and then, this component is removed from its cluster. In a next step, the algorithm should find the best cluster to insert this part or machine back (Semet and Taillard, 1993).

Suppose that the Families FP-1, FP-2 and FP-3 (Figure 3(a)) are clusters that represent the current solution for a machine supports 4 tools and the part 4 that belongs to the Cluster FP-3 is removed. Considering the, algorithm should find the best Family to insert this part back aiming to minimize the associated objective function as well as its restrictions. On this particular example, once the part 4 shares resources with the FP-1 the part 4 must be moves this cluster reducing the FP-3(Figure 3(b)). In a second iteration the part 7 could be moved to FP-2 cutting the FP-3 off.

|                |      | FP-1 |   |   | FP-2 |   |   | FP-3 |   |
|----------------|------|------|---|---|------|---|---|------|---|
|                |      | 1    | 5 | 3 | 6    | 2 | 8 | 7    | 4 |
| Machine Number | CM-1 | 1    |   | 1 |      |   |   |      |   |
|                | CM-2 | 2    | 1 | 1 |      |   |   |      |   |
|                | CM-3 | 3    |   |   |      | 1 |   |      |   |
|                | CM-2 | 4    |   |   |      | 1 | 1 | 1    |   |
|                | CM-3 | 5    |   |   |      |   | 1 | 1    |   |
|                | CM-3 | 6    |   |   | 1    |   |   |      | 1 |

Figure 3(a): Incidence matrix  $[a_{ij}]$  before insertion movement.

|                |      | FP-1 |   |   | FP-2 |   |   | FP-3 |   |
|----------------|------|------|---|---|------|---|---|------|---|
|                |      | 1    | 5 | 3 | 4    | 6 | 2 | 8    | 7 |
| Machine Number | CM-1 | 1    |   | 1 |      |   |   |      |   |
|                | CM-2 | 2    | 1 | 1 |      |   |   |      |   |
|                | CM-2 | 6    |   |   | 1    | 1 |   |      |   |
|                | CM-3 | 3    |   |   |      |   | 1 |      |   |
|                | CM-3 | 4    |   |   |      |   | 1 | 1    | 1 |
| CM-3           | 5    |      |   |   |      | 1 |   | 1    | 1 |

Figure 3(b): Incidence matrix  $[a_{ij}]$  after insertion movement.

When comparing both proposed intensification methods, it is possible to notice that both methods are very similar. More than this, it is possible to affirm that the insert column method is an improved 2-opt once in both cases a random part or machine is selected. However, on insert column algorithm, when returning this part to a cluster an objective function must be respected giving always the best solution. Due the second method requires more computational resources than the 2-opt method, it is used to generate only 5 neighbours and potential solution candidates.

### 3.3 Search Diversification

Different from the movements proposed in order to intensify the search (usually have simpler and subtle movements), in situations where it is necessary diversify the search and then to reach a solution far from the current solution then diversification movements are used. The diversification movements should perturb the neighbourhood increasing its candidate solution range consequently reaching solutions areas not explored before.

In the proposed model, the diversification movements occur only after a pre-defined number of iterations without improve of the current value of the actual solution or the value of the objective function. By default, this value is set as half of the value proposed as stopping criteria (The stopping criteria is defined as a finite number of iteration without improvement the value of objective function).

Once the iteration counter reaches half of the stopping criteria, automatically diversification movements are made perturbing the current neighbourhood. In this paper the chosen stopping criteria is 500 rounds without improvement, consequently the diversification movements only creates solution candidates after 250 iterations. Two different methods are proposed in order to generate solution candidates.

As previously cited in this paper, the Tabu Search is a method that improves the results through several iterations using a list of movements that should not be repeated and uses these resources to avoid becoming stuck in local optima area. However, the performance and the time to obtain an optimal or near-optimal solution are strongly affected by the proposed initial solution (Liu, Xiong and Liu, 2009).

Bearing in mind that the diversification algorithm is only used after several rounds without improvements, we can assume that it is potentially

stucked in a local optimal. If so, the result hardly will be improved. For this reason, the implemented Tabu Search proposes the usage of the same deterministic method previously used to generate the initial solution again, generating a unique solution candidate that points to a new optimal or suboptimal region, region which can possibly be the global optimal solution (Black, 2005).

It is known that the deterministic algorithm used to find an initial solution has a higher complexity -  $\Theta(2mn)$  (Kusiak, 1987) - than the standard algorithm used during the search intensification like 2-opt ( $\Theta(n^2)$ ). Nevertheless, with the advancement from studies related to the parallel processing area in addition to technological advancement of the computer hardware, it becomes possible to use algorithms with higher complexity instead of save processing time in combinatorial problems. Thus, it is proposed the use of the modified kusiak algorithm in parallel to the Tabu Search algorithm to create a solution candidate. The Figure 4 shows the architecture used to run the initial solution method in parallel to the intensification and diversification movements. Once it is a multithread processing method, an iteration should not be finished until both processes are completely done, otherwise there is a risk of loose the synchronization between both.

The implemented alternative Kusiak method can be defined as a deterministic algorithm, being non deterministic only when selecting the initial part represented by a row in the matrix. For this reason, depending of the initial part selected, the same result matrix can be obtained more than once. It occurs every time the initial part previously selected is selected again. For this reason, the implementation of second Tabu List is proposed in order to prevent that the same initial part be selected several times. This secondary Tabu List is responsible for storing a set of initial parts that were already used. This special list size is proportional to the problem size and has a size equivalent to 10% of the problem size.

### 3.4 Tabu List

When generating a collection of solutions, it can be observed that the new neighbourhood can present only worse solutions in comparison with the best solution found. The Tabu Search has resources that can prevent the method to visit these worse solutions again, as well as it has resources avoid becoming stuck in a local optimal region (Sorlin and Solnon, 2005). A solution is added to the Tabu List every time it was already visited.

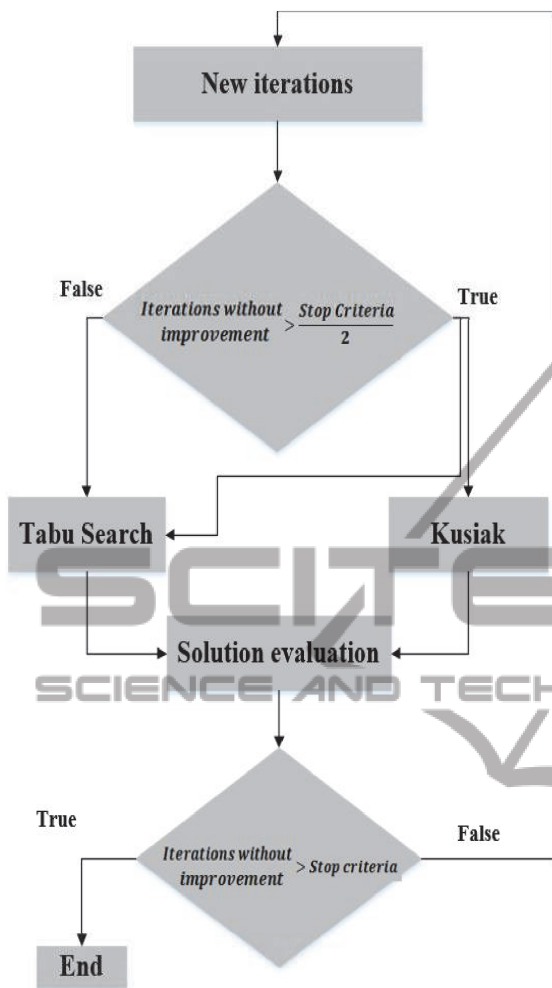


Figure 4: Multithreading Tabu Search architecture.

The Tabu Search is a method with memory structures. This memory structure is used in order to store a finite collection of movements made during its processing. Therefore, a list is necessary to store these movements, being this list known as the Tabu List (Glover, 1986). Bearing in mind that whether it is an finite size list, each movement added to the Tabu List must remain in this list for a limited and pre-defined time. This limitation occurs through the definition of the Tabu List Size (Basu, 2012).

The size of the Tabu List allows the user to intensify or diversify the search. The Tabu List memory types can be divided into three different categories Short-term, Intermediate-term or Long-term.

The Tabu List size can also be static, dynamic or random (Tang and Hooks, 2005). According with (Basu, 2012) survey, lists with static size are the most common and its size usually varies from 0 to 4 like as proposed in (Crainic, Gendreau, Soriano and

Toulouse, 1993). On the other hand, there are articles that also present static lists, however with a larger list allowing more than 30 values (Lau, Sim and Teo, 2003). In this paper as well as presented by (Ting, Li and Lee, 2003; Greistorfer, 2003) the Tabu List size is proportional the number of parts and machines being analyzed, the chosen proportion on this implementation is 25% of the problem size.

#### 4 EXPERIMENTS AND RESULTS

In order to test the Tabu Search implementation, a set of 11 problems extracted from the literature is used. This set of problems was previously tested by James et. al. (2007) and Gómez et. Al. (2011). The best solutions found in both papers are compared against the solutions obtained on this paper. The list of test instances was extracted from James et. al. (2007).

Table 1: Minimum, maximum and average efficiency.

| ID | Problem Source                          | Min   | Max   | Average | $\sigma$ |
|----|-----------------------------------------|-------|-------|---------|----------|
| 1  | King and Nakornchai (1982)              | 82,35 | 82,35 | 82,35   | 0,00     |
| 2  | Waghodekar and Sahu (1984)              | 70,00 | 80,00 | 74,08   | 2,44     |
| 3  | Seifoddini (1989)                       | 79,59 | 79,59 | 79,59   | 0,00     |
| 4  | Kusiak and Cho (1992)                   | 80,00 | 80,00 | 80,00   | 0,00     |
| 5  | Kusiak and Chow (1987)                  | 58,62 | 60,71 | 58,97   | 0,75     |
| 6  | Boctor (1991)                           | 70,37 | 73,08 | 71,18   | 1,31     |
| 7  | Seifoddini and Wolfe (1986)             | 76,00 | 76,00 | 76,00   | 0,00     |
| 8  | Chandrasekharan and Rajagopalan (1986a) | 66,67 | 67,44 | 66,89   | 0,36     |
| 9  | Chandrasekharan and Rajagopalan (1986b) | 18,18 | 26,77 | 22,47   | 4,52     |
| 10 | Mosier and Taube (1985)                 | 70,59 | 76,00 | 71,87   | 2,26     |
| 11 | Chan and Milner (1982)                  | 92    | 92    | 92      | 0,00     |

Once the proposed Tabu Search uses multithreading processing – a multi core processor must be used to evaluate the processing time gain, therefore the tests were made in a computer with an Intel Core i5-3210M processor with 2.5Ghz and 6GB of RAM memory.

Each problem was submitted to 75 executions considering a machine magazine limited to 4 tools. In the Table 1 are shown the obtained: worse and best efficiency result, average efficiency result and its standard deviation. Like in James et. al. (2007) the problems are organized by ID and problem source.

An relevant point observed during the tests are the low processing time. In most part of the cases, the processing time have not exceed 2 seconds. The lowest processing time is observed on problem 1 – average processing time of 1 second and 200 milisenconds – and the biggest processing time is observed on problem 8 taking 2 seconds and 79 miliseconds. This low processing time even using the Kusiak algorithm to diversificate the search proves that with the new hardware improvements, methods with a higher complexity can be used in metaheuristics without performance reduction.

The method proved it self powerful when grouping machines that do not have to process more parts than the magazine limit. Once we are imposing a restriction, on this particular case imposing a magazine limit of 4 tool. Consequently, if a machine have to process more than 4 parts – it tends to create a bottleneck because this part have to be processed in more than one step. This situation is observed on problem 9, this problem has several parts that requires 5 tools on its production. For this reason, it is expected to have a result above the found on literature where this restriction is not impose.

In other hand, in problems where 4 or less tools are used during the process, like in problems 1, 2, 3, 4 and 5, the results are better or the same when compared to the literature. Table 2 shows the results in the literature in comparison to the obtained on this paper. Proposed Tabu Search represents the results found on this paper. Results better than the ones found in literature are in bold.

Though the Table 2 it is possible to observe that the solutions found to the problems 2, 4, 6, 7 and 10 are better than the solutions presented by the literature. On problems 1, 3 and 11 the same result was found. On problems 5, 8 and 9 the results found in the literature are better than the ones found by the proposed Tabu Search.

Table 2: Comparison between literature results and proposed Tabu Search results.

| ID | Best Solution in source Problem | James et. al. (2007) | Gómez et. Al. (2011) | Proposed Tabu Search |
|----|---------------------------------|----------------------|----------------------|----------------------|
| 1  | 73,68                           | 82,35                | 82,35                | <b>82,35</b>         |
| 2  | 68                              | 69,57                | 69,57                | <b>74,08</b>         |
| 3  | 79,59                           | 79,59                | 79,59                | 79,59                |
| 4  | 76,92                           | 76,92                | 76,92                | <b>80,00</b>         |
| 5  | 53,13                           | 60,87                | 60,87                | 58,97                |
| 6  | 70,37                           | 70,83                | 70,83                | <b>71,18</b>         |
| 7  | 68,3                            | 69,44                | 69,44                | <b>76,00</b>         |
| 8  | 85,25                           | 85,25                | 85,25                | 66,89                |
| 9  | 58,72                           | 58,72                | 56,70                | 22,47                |
| 10 | 72,79                           | 75                   | 70,35                | 71,87                |
| 11 | 92                              | 92                   | 92                   | 92                   |

## 5 CONCLUSIONS

This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search, that uses deterministic methods to effectively explore loca optimum areas. On this paper, the magazine is considered with limited capacity. The magazine capacity considered is four tools.

The method obtained better solutions when compared against other solutions found in the literature when using 4 or less tools on the manufacturing process. proving to be a powerfull method to create manufacturing cells for limited number of tools or resources.

However, due the magazine limit restriction added to the problem the better results are limited only to parts that use 4 or less tools on its processing. Thus, an independent method to deal with this restriction could be implemented in order

to improve the result efficiency. Consequently reducing the number of tool changes and setup time. Future works will present other studies on the problem addressed in this article.

## REFERENCES

- Basu, S., 2012. *Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey*, American Journal of Operations Research, Vol. 2 No. 2, pp. 163-173.
- Black, P. E., 2005. *Greedy algorithm*, in *Dictionary of Algorithms and Data Structures [online]*, Vreda Pieterse and Paul E. Black, eds.
- Boctor, F. F., 1991. *A linear formulation of the machine-part cell formation problem*. International Journal of Production Research; 29(2):343-56.
- Chan, H. M., Milner, D. A., 1982. *Direct clustering algorithm for group formation in cellular manufacture*. Journal of Manufacturing Systems; 1:65-75.
- Chandrasekharan, M. P., Rajagopalan, R., 1986. *An ideal seed non-hierarchical clustering algorithm for cellular manufacturing*. International Journal of Production Research; 24(2):451-64.
- Chandrasekharan M. P., Rajagopalan R., 1986. *MODROC: an extension of rank order clustering for group technology*. International Journal of Production Research; 24(5):1221-64.
- Crainic, T. G., Gendreau, M., Soriano P., Toulouse, M., 1993. *A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements*, Annals of Operations Research, Vol. 41, pp. 359-383.
- Croes, G. A., 1958. *A method for solving traveling salesman problems*. Operations Res. 6, pp. 791-812.
- Fiechter, C. N., 1994. *A parallel tabu search algorithm for large traveling salesman problems*, Discrete Applied Mathematics, Volume 51, Issue 3.
- Glover, F., 1986. *Future paths for integer programming and links to artificial intelligence*. Computers and Operations Research, 1, p. 533-549.
- Gómez, A. T., 1996. *Modelo para o sequenciamento de partes e ferramentas em um sistema de manufatura flexível com restrições às datas de vencimento e à capacidade do magazine*, PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, Brasil.
- Gómez, A. T., Linck, I., Galafassi, C., Wickert, T., 2011. *A Study for Manufacturing Cell Formation Approach Considering Setup*, 8th International Conference, ICINCO 2011 Noordwijkerhout, The Netherlands, p. 43-48.
- Greistorfer, P., 2003. *A Tabu Scatter Search Metaheuristic for the Arc Routing Problem*, Computers and Industrial Engineering, Vol. 44, No. 2.
- Groover, M. P., 2001. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Upper Saddle River, New Jersey: Prentice-Hall, USA.
- Hasegawa, M., Ikeguchi, T., Aihara, K., 1997. *Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problem*. Physical Review Letters, pp. 2344-2347.
- He, Y., Qiu, Y., Liu, G., Lei, K., 2005. *A parallel adaptive tabu search approach for traveling salesman problems*, Natural Language Processing and Knowledge Engineering.
- James, T. L., Brown, E. C. and Keeling, K. B., 2007. *A hybrid grouping genetic algorithm for the cell formation problem*, Computers and Operations Research, 34, 2059-2079.
- Kaur, D., Murugappan, M. M., 2008. *Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm*, in Proceedings of the 2008 annual meeting of the North American fuzzy information processing society, New York, pp.1-6.
- King, J. R., Nakornchai V., 1982. *Machine-component group formation in group technology: review and extension*. International Journal of Production Research; 20(2):117-33.
- Kothari, R., Ghosh, D., 2013. *Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighbourhoods*, European Journal of Operational Research, Volume 224, Issue 1, Pages 93-100.
- Kumar, K. R., Chandrasekharan, M. P., 1990. *Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology*. International Journal of Production Research, 28(2), 233-243.
- Kusiak, A., Cho, M., 1992. *Similarity coefficient algorithm for solving the group technology problem*, International Journal of Production Research; 30(11):2633-46.
- Kusiak, A., Chow, W. S., 1987. *Efficient solving of the group technology problem*, Journal of Manufacturing Systems, v. 6(2), p. 117-24.
- Lau, H., Sim M., and Teo K., 2003. *Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles*, European Journal of Operational Research, Vol. 148, No. 3.
- Li, X., Baki, M. F., Aneja, Y. P., 2011. *Flow shop scheduling to minimize the total completion time with a permanently present operator: Models and ant colony optimization metaheuristic*, Computers & Operations Research, Volume 38, Issue 1, Pages 152-164.
- Lim, Y. F., Hong, P. Y., Ramli, R., Khalid, R., 2011. *An improved tabu search for solving symmetric traveling salesman problems*, Humanities, Science and Engineering (CHUSER), 2011 IEEE Colloquium on , vol., no., pp.851,854.
- Liu, Y., Xiong, S., and Liu, H., 2009. *Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP*, Proceedings of GEC'09, Shanghai, pp.895-898.
- Mosier, C. T., Taube, L., 1985. *The facets of group technology and their impact on implementation*. OMEGA; 13(5):381-91.

- Seifoddini, H., Wolfe, P. M., 1986. *Application of the similarity coefficient method in group technology*. IIE Transactions; 18(3):271-7.
- Seifoddini, H., 1989. *A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications*. International Journal of Production Research; 27(7):1161-5.
- Selim, M. S., Askin, R. G., Vakharia, A. J., 1998. *Cell formation in group technology: review evaluation and directions for future research*. Computers and Industrial Engineering 34 (1), 3-20.
- Semet, F., Taillard, E. D., 1993. *Solving real-life vehicle routing problems efficiently using Tabu Search*, Annals of Operations Research. Vol. 41, pp. 469-488.
- Sorlin, S., Christine, S., 2005. *Reactive Tabu Search for Measuring Graph Similarity*, Graph-Based Representations in Pattern Recognition, pp. 172-182.
- Spiliopoulos, K., Sofianopoulou, S., 2008. *An efficient ant colony optimization system for the manufacturing cells formation problem*, International Journal of Advanced Manufacturing Technology, 36, 589-597.
- Tang H., Hooks E., 2005. *A Tabu Search Heuristic for the Team Orienteering Problem*, Computers & Operations Research, Vol. 32, No. 6.
- Ting, C., Li, S. and Lee, C., 2003. *On the Harmonious Mating Strategy through Tabu Search*, Information Sciences, Vol. 156, No. 3-4.
- Tsai, C. W., Tseng, S. P., Chiang, M. C., Yang, C. S., 2009. *A Time-Efficient Method for Metaheuristics: Using Tabu Search and Tabu GA as a Case*, Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on, vol.2, no., pp.24-29.
- Waghodekar, P. H., Sahu S., 1984. *Machine-component cell formation in group technology MACE*. International Journal of Production Research;22:937-48.
- Wemmerlöv, U., Hyer, N. L., 1989. *Cellular manufacturing in the U.S. industry: a survey of users*, International Journal of Production Research 27(9), 1511-1530.
- Wu, T. H., Chung, S. H., Chang, C. C., 2010. *A water flow-like algorithm for manufacturing cell formation problems*. European Journal of Operational Research 205(2): 346-360.
- Wu, X., Chu, C.H., Wang, Y., Yue, D., 2007. *Genetic algorithms for integrating cell formation with machine layout and scheduling*, Computers & Industrial Engineering, 53, Issue 2, Pages 277-289.
- Xu, Y. T., Zhang, Y., Huang, X., 2014. *Single-machine ready times scheduling with group technology and proportional linear deterioration*, Applied Mathematical Modelling, Volume 38, Issue 1, pp. 384-391.



## **A MULTITHREADING TABU SEARCH FOR SETUP TIME OPTIMIZATION FOR LIMITED MACHINE MAGAZINES IN THE CELL FORMATION PROBLEM**

**Felipe Rodrigues Ferrary**

Universidade do Vale do Rio dos Sinos  
Av. Unisinos - 950, São Leopoldo, Brasil  
feliperferrary@gmail.com

**Arthur Tórgo Gómez**

Universidade do Vale do Rio dos Sinos  
Av. Unisinos - 950, São Leopoldo, Brasil  
breno@unisinos.br

**José Vicente Canto dos Santos**

Universidade do Vale do Rio dos Sinos  
Av. Unisinos - 950, São Leopoldo, Brasil  
jvcanto@unisinos.br

**Leonardo Chiwiacowsky**

Universidade do Vale do Rio dos Sinos  
Av. Unisinos - 950, São Leopoldo, Brasil  
ldchiwiacowsky@unisinos.br

### **ABSTRACT**

This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search that uses deterministic methods to effectively explore local optimum areas. Manufacturing Cell Formation problem involves the process of analysing parts and groups them according with their similarity. This paper presents a method focused in maximize the production efficiency, by minimizing the machine setup time in a limited magazine size through the reduction of tool changes by creating clusters of parts that share machining tools and present an initial scheduling based on tool changes reduction. In order to valid the proposed algorithm, the results obtained are compared against other Tabu Search solutions proposed in the literature. In order to validate the proposed method, the results obtained are compared against other solutions found in the literature and during this process, in some instances results better than the ones found in the literature are obtained.

**KEYWORDS.** Tabu Search; Machine Setup Time Optimization; Manufacturing Cell Formation

**Main area:** IND - OR in Industry

## 1. Introduction

The process of batching and automate manufacturing processes is an essential process for companies that want to make competitive products. The Batch manufacturing is estimated to be the most common form of production, constituting more than 50% of the total manufacturing activity in the US. In addition, there is a growing need to make batch manufacturing more efficient and productive (Groover, 2001).

In the batching process, Group Technology (GT) has an important role and offers relevant contribution helping to increase the efficiency of production operations and reducing the requirement of facilities (Xu et al., 2014). As part of GT, the Manufacturing Cell Formation problem aims to define an efficient structure to group machines (James et al., 2007).

The Manufacturing Cell Formation problem is NP-Hard (Spiliopoulos and Sofianopoulou, 2008). Thus, simple heuristics has the propensity to not present satisfactory results. Therefore, several methods making use of artificial intelligence techniques are proposed to solve the problem. Being the manufacturing cell formation a problem of combinatorial optimization, metaheuristic like Ant Colony (Li et al., 2011; Spiliopoulos and Sofianopoulou, 2008), Genetic Algorithms (Xiaodan et al., 2007), Tabu Search (Gómez et al., 2011) are commonly used to find a good solution. On this paper the metaheuristic Tabu Search is used to solve the problem.

Beyond that, another resource that currently receives increasing focus on academic papers is the parallel processing applied to these methods (Fiechter, 1994; He et al., 2005). In such cases, different processors can perform multiple calculations simultaneously.

Another method gaining increased attention is the hybridization of metaheuristics (Kaur and Murugappan, 2008). The success of the hybrid approach comes from the union factor of the strategic advantages of each method in a single metaheuristic; providing a better performance compared to the pure method (Tsai et al., 2009).

In this paper, a variation of the classical implementation of Tabu Search is proposed. Being part of this modification, concepts of hybridization and parallel programming are used to provide solutions near promising regions.

This paper is divided into 5 sections. Section 2 presents concepts of the Manufacturing Cell Formation problem as well as review from its literature. Section 3 presents details about the Tabu Search proposed to solve the problem, also further details about the multithreading concept used on this method are provided. Section 4 presents the methods used for general testing, as well as the metric used to compare the result against other solution found in the literature. Section 5 presents the conclusions.

## 2. Manufacturing Cell Formation Literature Review

The cellular formation for a manufacturing system is an application of the Group Technology (GT) – which is a tool to identify similar parts and group them together regarding similarities between them (Selim et al., 1998). The Cell Formation (CF) aims to reduce the setup and flow times – minimizing the inventory and manufacturing lead times (Wemmerlov and Hyer, 1989; Wu et al., 2010).

CF is a binary matrix machines versus parts which reorganizes rows and columns with the intention to group parts (part families) and machines (machine cell). The binary element in the Figure 1 represents the relationship between part and machine indicating:

- 1 – the part  $p$  uses the resource (or machine)  $m$ ;
- 0 – the part  $p$  can't use the resource  $m$ ;



| Machine Number | Part Number |   |   |   |   |
|----------------|-------------|---|---|---|---|
|                | 1           | 2 | 3 | 4 | 5 |
| 1              | 0           | 1 | 0 | 1 | 1 |
| 2              | 1           | 0 | 1 | 0 | 0 |
| 3              | 0           | 1 | 0 | 1 | 0 |
| 4              | 1           | 0 | 1 | 0 | 0 |

Figure 1: Relationship between part and machine.

Then, apply the algorithm proposed by Kusiak and Chow (1987) to find any relationship between parts and machines – rearranging the cells and resulting in two cellular formation for manufacturing systems and two part families, as described in the Figure 2.

|      | FP-1 |   | FP-2 |   |   |
|------|------|---|------|---|---|
|      | 1    | 3 | 2    | 4 | 5 |
| CM-1 | 2    | 1 | 1    |   |   |
|      | 4    | 1 | 1    |   |   |
| CM-2 | 1    |   | 1    | 1 | 1 |
|      | 3    |   | 1    | 1 |   |

Figure 2: Cells created by Kusiak Algorithm.

### 3. Proposed Tabu Search

Firstly proposed in 1986, Tabu Search is a method that can be used to solve different problems on combinatorial optimization environment (Glover, 1986). According with Glover (1986), Tabu Search may be viewed as a metaheuristic superimposed on another heuristic. This method uses a list in order to forbid movements (“tabu movement”) that drives to solution areas already explored.

Basu (2012) brings a brief and updated literature review about the Tabu Search being used as a problem resolution tool related to combinatorial area. On this paper, the main methods used to perform movements from one solution to another( in order to generate the neighbourhood), Tabu List size and type, search intensification and diversification, aspiration criteria as well as a couple of other relevant details are pointed along the article.

The different methods overviewed by this article are evaluated qualitatively (through the quality of the solution) as well as quantitatively (by checking the number of times this method is used in the studied articles). The methods employed in the proposed Multithreading Tabu search take in account of the methods pointed as the most efficient and appropriate to the resolution of combinatorial problems.

The solution quality is evaluated by using a metric named “effective clustering” (Kumar and Chandrasekharan, 1990). This metric is represented by the equation(1).

$$\tau = \frac{e - e_0}{e + e_r} \quad (1)$$

In the given equation,  $e$  represents the number of 1's in the given matrix,  $e_v$  represents the number of voids (number of 0's inside the clusters in the main diagonal)  $e_o$  is the number of exception (the number of 1's outside the main diagonal).

### 3.1. Initial Solution

As initial solution, the Kusiak Algorithm is used. This algorithm is an iterative process that selects rows and columns in a matrix in order to simultaneously create Family Parts and Machine Clusters. Kusiak and Chow (1987) have initially proposed an algorithm that identifies the similarity between manufacturing processes and creates separable groups. This method uses a binary part-machine incidence matrix  $A = [a_{ij}]$  and decomposes  $A$  into sub matrices  $A_1, A_2, \dots, A_n$ . Each sub matrix can be defined as a machine cell.

However, once this paper aims to control and reduce the number of tool changes in a limited magazine, it is necessary taking in account of the machine magazine limit. For this reason, the algorithm used on this paper to generate the initial solution is a method proposed by Gómez (1996), because this method considers the similarity of tools used in the manufacturing process, as well as the machine magazine limit.

Despite this kind of method hardly returns the optimal solution, it is frequently used because it points to an optimum solution. Usually an exact algorithm points to a local optimum solution instead of a global one. However, this algorithm can in some particular cases point to the optimal solution as well as it can point to a near-optimal solution that can be considered an acceptable result (Black, 2005).

### 3.2. Search Intensification

Proposed by Croes (1958) as a method to solve the travelling salesman problem, the 2-opt is widely used to modify a current solution and generate a new neighborhood until the stop criteria requirements are met (Lim, Yong, Ramli and Khalid, 2011). Also, it is used as a local search method in many other combinatorial problems (Kothari and Ghosh, 2013; Hasegawa, Ikeguchi and Aihara, 1997).

The 2-opt movement occurs through the removal of two non-adjacent parts or machines from a cluster. After the removal of these components, those parts or machines have their position exchanged thus preventing the need of perform an evaluation of a subgroup again. In addition to, 2-opt movement is also used to exchange the families orders, thus changing the current scheduling. This process helps in reducing the number of tool change between Family parts, consequently reducing the setup time.

The intensification occurs due the non-substantial change of only two parts or machines, thus creating a solution near of the current solution. Considering it is a simple movement, consequently it has a low computational complexity –  $\theta(n^2)$  for 2-opt algorithm (Croes, 1958). For this reason, this method can be commonly find in Tabu Search implementations. In this paper, the 2-opt movement is used to generate 20 neighbours and potential candidate solutions.

Another method used in the implementation and on search intensification is a column insertion method. In this method, as initial step a part or machine is randomly selected and then, this component is removed from its cluster. In a next step, the algorithm should find the best cluster to insert this part or machine back (Semet and Taillard, 1993).

Suppose that the Families FP-1, FP-2 and FP-3 (Figure 3) are clusters that represent the current solution for a machine that supports 4 tools and the part 4 that belongs to the Cluster FP-3 is removed. Considering the, algorithm should find the best family to insert this part back aiming to minimize the associated objective function as well as its restrictions. On this particular example, since the part "4" shares resources with FP-1, then the part "4" must be moved to this cluster reducing the FP-3(Figure 4). In a second iteration the part 7 could be moved to FP-2 cutting the FP-3 off.

|                |      | FP-1 |   |   | FP-2 |   |   | FP-3 |   |
|----------------|------|------|---|---|------|---|---|------|---|
|                |      | 1    | 5 | 3 | 6    | 2 | 8 | 7    | 4 |
| Machine Number | CM-1 | 1    |   | 1 |      |   |   |      |   |
|                | CM-2 | 1    | 1 |   |      |   |   |      |   |
|                | CM-3 |      |   |   | 1    |   |   |      |   |
|                | CM-4 |      |   |   | 1    | 1 | 1 |      |   |
|                | CM-5 |      |   |   | 1    |   | 1 | 1    |   |
|                | CM-6 |      |   | 1 |      |   |   |      | 1 |

Figure 3: incidence matrix  $[a_{ij}]$  before insertion movement.

|                |      | FP-1 |   |   | FP-2 |   |   | FP-3 |   |
|----------------|------|------|---|---|------|---|---|------|---|
|                |      | 1    | 5 | 3 | 4    | 6 | 2 | 8    | 7 |
| Machine Number | CM-1 | 1    |   | 1 |      |   |   |      |   |
|                | CM-2 | 1    | 1 |   |      |   |   |      |   |
|                | CM-6 |      |   | 1 | 1    |   |   |      |   |
|                | CM-3 |      |   |   |      | 1 |   |      |   |
|                | CM-4 |      |   |   |      | 1 | 1 | 1    |   |
| CM-5           |      |      |   |   | 1    |   | 1 | 1    |   |

Figure 4: incidence matrix  $[a_{ij}]$  after insertion movement.

When comparing both proposed intensification methods, it is possible to notice that both methods are very similar. More than that, it is possible to affirm that the insert column method is an improved 2-opt once in both cases a random part or machine is selected. However, on insert column algorithm, when returning this part to a cluster an objective function must be respected giving always the best solution. Due the second method require more computational resources than the 2-opt method, it is used to generate only 5 neighbours and potential candidate solutions.

### 3.3. Search Diversification

Different from the movements proposed in order to intensify the search (movements that usually are simpler), in situations where it is necessary to diversify the search and then to reach a solution far from the current solution then diversification movements are used. The diversification movements should perturb the neighbourhood increasing its candidate solution range consequently reaching solutions areas not explored before.

In the proposed model, the diversification of movements occur only after a pre-defined number of iterations without improving the current value of the actual solution or the value of the objective function. By default, this value is set as half of the value proposed as stopping criteria (The stopping criteria is defined as a finite number of iteration without improvement the value of objective function).

Once the iteration counter reaches half of the stopping criteria, automatically diversification movements are made perturbing the current neighbourhood. In this paper the chosen stopping criteria is 500 rounds without improvement, consequently the diversification movements only creates candidate solutions after 250 iterations. Two different methods are proposed in order to generate candidate solutions.

As previously cited in this paper, the Tabu Search is a method that improves the results through several iterations using a list of movements that should not be repeated and it has

resources to avoid becoming stuck in local optima area. However, the performance and the time to obtain an optimal or near-optimal solution are strongly affected by the proposed initial solution (Liu et al., 2009).

Bearing in mind that the diversification algorithm is only used after several rounds without improvements, we can assume that it is potentially stuck in a local optimal. If so, the result hardly will be improved. For this reason, the implemented Tabu Search proposes the usage of the same deterministic method previously used to generate the initial solution again, generating a unique candidate solution that points at a new optimal or suboptimal region, region which can possibly be the global optimal solution (Black, 2005).

It is known that the deterministic algorithm used to find an initial solution has a higher complexity -  $\theta(2mn)$  (Kusiak, 1987) - than the standard algorithm used during the search intensification like 2-opt ( $\theta(n^2)$ ). Nevertheless, with the advancement from studies related to the parallel processing area in addition to technological advancement of the computer hardware, it becomes possible to use algorithms with higher complexity instead of saving processing time in combinatorial problems. Thus, it is proposed the use of the modified Kusiak Algorithm in parallel to the Tabu Search algorithm to create a candidate solution. The Figure 5 shows the architecture used to run the initial solution method in parallel to the intensification and diversification movements. Once it is a multithread processing method, an iteration should not be finished until both processes are completely done, otherwise there is a risk of losing the synchronization between both.

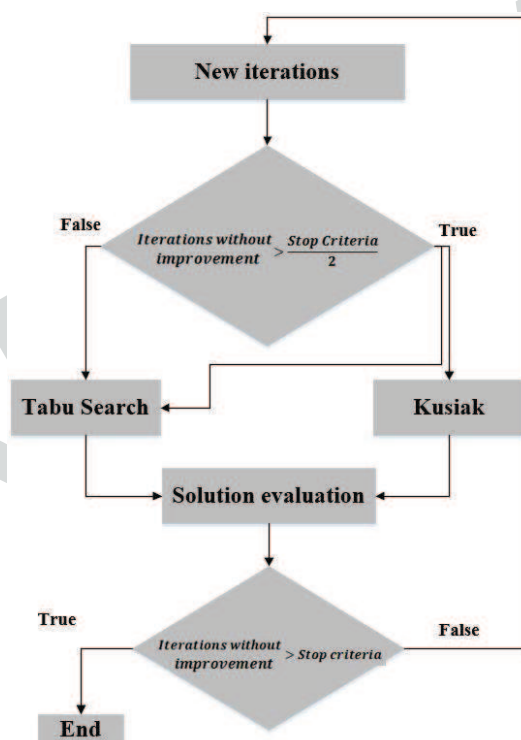


Figure 5: multithreading Tabu Search architecture

The implemented alternative Kusiak Algorithm can be defined as a deterministic algorithm, being non deterministic only when selecting the initial part represented by a row in the matrix. For this reason, depending on the initial part selected, the same resulting matrix can be obtained more than once. It occurs every time the initial part previously selected is selected again. For this reason, the implementation of the second Tabu List is proposed in order to prevent that the same initial part is selected several times. This secondary Tabu List is responsible for storing a set of initial parts that were already used. This special list size is proportional to the problem size and has a size equivalent to 10% of the problem size.

### 3.4. Tabu List

When generating a collection of solutions, it can be observed that the new neighbourhood can present only worse solutions in comparison with the best solution found. The Tabu Search has resources that can prevent the method to visit these worse solutions again, as well as it has resources avoid becoming stuck in a local optimal region (Sorlin and Christine, 2005). A solution is added to the Tabu List every time it was already visited.

The Tabu Search is a method with memory structures. This memory structure is used in order to store a finite collection of movements made during its processing. Therefore, a list is necessary to store these movements, being that list known as the Tabu List (Glover, 1986). Bearing in mind that whether it is an finite size list, each movement added to the Tabu List must remain in this list for a limited and pre-defined time. This limitation occurs through the definition of the Tabu List Size (Basu, 2012).

The size of the Tabu List allows the user to intensify or diversify the search. The Tabu List memory types can be divided into three different categories Short-term, Intermediate-term or Long-term.

The Tabu List size can also be static, dynamic or random (Tang and Hooks, 2005). According to Basu (2012) survey, lists with static size are the most common and its size usually varies from 0 a 4 like as proposed in (Crainic et al., 1993). On the other hand, there are articles that also present static lists, however with a larger list allowing more than 30 values (Lau, Sim and Teo, 2003). In this paper as well as presented by Ting, Li and Lee (2003) and Greistorfer (2003), the Tabu List size is proportional to the number of parts and machines being analyzed, the chosen proportion on this implementation is 25% of the problem size.

## 4. Experiments and Results

In order to test the Tabu Search implementation, a set of 11 problems extracted from the literature is used. This set of problems was previously tested by James et al. (2007) and Gómez et al. (2011). The best solutions found in both papers are compared against the solutions obtained on this paper. The list of test instances was extracted from James et al. (2007).

Once the proposed Tabu Search uses multithreading processing – a multi core processor must be used to evaluate the processing time gain, therefore the tests were made in a computer with an Intel Core i5-3210M processor with 2.5GHz and 6GB of RAM memory.

Each problem was submitted to 75 executions considering a machine magazine limited to 4 tools. In the Table 1 it is shown the obtained: worse and best efficiency results, average efficiency result and its standard deviation. Like in James et al. (2007) the problems are organized by ID and problem source. In addition to , the problem size and the maximum number of tools used in a individual part are also present on Table 1.

An relevant point observed during the tests is the low processing time. In most cases, the processing time has not exceed 2 seconds. The lowest processing time is observed on problem 1 – average processing time of 1 second and 200 milisenconds – and the biggest processing time is observed on problem 8 taking 2 seconds and 79 miliseconds. When comparing this processing time to the average time to obtain a solution in James et al. (2007) it is possible to observe that the processing time does not increase significantly even using the proposed method.

This low processing time - even using the Kusiak Algorithm to diversificate the search - shows that with the new hardware improvements, methods with a higher complexity can be used in metaheuristics without performance loss.

The method proved itself powerful when grouping jobs where the number to tools used do not exceed the magazine limit. On this particular case the method is imposing a magazine limit of 4 tools. Consequently, if a machine has to process more than 4 parts – it tends to create a bottleneck because this part has to be processed in more than one step. This situation is observed on problem 9, this problem has several parts that requires 5 tools on its production. For that

reason, it is expected to have a result above the found on the literature where this restriction is not impose.

Table 1: Minimum, maximum and average efficiency.

| ID | Problem Source                          | Problem Size | Max number of tool used | Min   | Max   | Average | $\sigma$ |
|----|-----------------------------------------|--------------|-------------------------|-------|-------|---------|----------|
| 1  | King and Nakornchai (1982)              | 7x5          | 3                       | 82,35 | 82,35 | 82,35   | 0        |
| 2  | Waghodekar and Sahu (1984)              | 7x5          | 4                       | 70    | 80    | 74,08   | 2,44     |
| 3  | Seifoddini (1989)                       | 18x5         | 3                       | 79,59 | 79,59 | 79,59   | 0        |
| 4  | Kusiak and Cho (1992)                   | 8x6          | 4                       | 80    | 80    | 80      | 0        |
| 5  | Kusiak and Chow (1987)                  | 11x7         | 3                       | 58,62 | 60,71 | 58,97   | 0,75     |
| 6  | Boctor (1991)                           | 11x7         | 3                       | 70,37 | 73,08 | 71,18   | 1,31     |
| 7  | Seifoddini and Wolfe (1986)             | 12x8         | 5                       | 76    | 76    | 76      | 0        |
| 8  | Chandrasekharan and Rajagopalan (1986a) | 20x8         | 5                       | 66,67 | 67,44 | 66,89   | 0,36     |
| 9  | Chandrasekharan and Rajagopalan (1986b) | 20x8         | 6                       | 18,18 | 26,77 | 22,47   | 4,52     |
| 10 | Mosier and Taube (1985)                 | 10x10        | 4                       | 70,59 | 76    | 71,87   | 2,26     |
| 11 | Chan and Milner (1982)                  | 15x10        | 4                       | 92    | 92    | 92      | 0        |

In other hand, in problems where 4 or fewer tools are used during the process, like in problems 1, 2, 3, 4 and 5, the results are better or the same when compared to the literature. Table 2 shows the results in the literature in comparison to the obtained on this paper. Proposed Tabu Search represents the results found on this paper. Results better than the ones found in the literature are in bold.

Table 2: Comparison between literature results and proposed Tabu Search results.

| ID | Best Solution in source Problem | James et al. (2007) | Gómez et al. (2011) | Proposed Tabu Search |
|----|---------------------------------|---------------------|---------------------|----------------------|
| 1  | 73,68                           | 82,35               | 82,35               | 82,35                |
| 2  | 68                              | 69,57               | 69,57               | <b>74,08</b>         |
| 3  | 79,59                           | 79,59               | 79,59               | 79,59                |
| 4  | 76,92                           | 76,92               | 76,92               | <b>80,00</b>         |
| 5  | 53,13                           | 60,87               | 60,87               | 58,97                |
| 6  | 70,37                           | 70,83               | 70,83               | <b>71,18</b>         |
| 7  | 68,3                            | 69,44               | 69,44               | <b>76,00</b>         |
| 8  | 85,25                           | 85,25               | 85,25               | 66,89                |
| 9  | 58,72                           | 58,72               | 56,70               | 22,47                |
| 10 | 72,79                           | 75                  | 70,35               | <b>71,87</b>         |
| 11 | 92                              | 92                  | 92                  | 92                   |

Though the Table 2 it is possible to observe that the solutions found to the problems 2, 4, 6, 7 and 10 are better than the solutions presented by the literature. On problems 1, 3 and 11

the same result was found. On problems 5, 8 and 9 the results found in the literature are better than the ones found by the proposed Tabu Search.

## 5. Conclusions

This paper aims to present a solution for the Manufacturing Cell Formation Problem through the use of a multithreading Tabu Search, that uses deterministic methods to effectively explore local optimum areas. On this paper, the magazine is considered with limited capacity. The magazine capacity considered is four tools.

The method obtained better solutions when compared against other solutions found in the literature when using 4 or fewer tools on the manufacturing process, proving to be a powerful method to create manufacturing cells for limited number of tools or resources.

However, due the magazine limit restriction added to the problem the better results are limited only to parts that use 4 or fewer tools on its processing. Thus, an independent method to deal with this restriction could be implemented in order to improve the result efficiency. Consequently reducing the number of tool changes and setup time. Future works will present other studies on the problem addressed in this article.

Also, the method proved itself powerful to obtain solutions in real time without reducing the solution quality. Thus being possible to use this method in situations where a real-time solution for a cell formation problem is necessary - like in stochastic scheduling problems.

Future works will present other studies on the problem addressed in this article. Also, the limited magazine should change from a fixed number of tools - on this paper 4 - to the maximum number of tool being used in a given problem, thus testing the grouping efficiency without creating a bottleneck.

## REFERENCES

- Basu, S.** (2012), Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey, *American Journal of Operations Research*. Vol. 2 No. 2, pp. 163-173.
- Black, P. E.**, Greedy algorithm, in Dictionary of Algorithms and Data Structures [online], Vreda Pieterse and Paul E. Black, eds., 2005.
- Boctor, F. F.** (1991), A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*. 29(2):343-56.
- Chan, H. M., Milner, D. A.** (1982), Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*; 1:65-75.
- Chandrasekharan, M. P., Rajagopalan, R.** (1986), An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24(2):451-64.
- Chandrasekharan M. P., Rajagopalan R.** (1986), MODROC: an extension of rank order clustering for group technology. *International Journal of Production Research*, 24(5):1221-64.
- Crainic, T. G., Gendreau, M., Soriano P. and Toulouse, M.** (1993), A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements, *Annals of Operations Research*, Vol. 41, pp. 359-383.
- Croes, G. A.** (1958), A method for solving traveling salesman problems. *Operations Res.* 6. , pp., 791-812.
- Fiechter, C. N.** (1994), A parallel Tabu search algorithm for large traveling salesman problems, *Discrete Applied Mathematics*, Volume 51, Issue 3.
- Glover, F.** (1986), Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 1, p. 533-549.

- Gómez, A. T.** (1996), Modelo para o sequenciamento de partes e ferramentas em um sistema de manufatura flexível com restrições às datas de vencimento e à capacidade do magazine, PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, São Paulo, Brasil.
- Gómez, A. T., Linck, I., Galafassi, C. and Wickert, T.** (2011), A Study for Manufacturing Cell Formation Approach Considering Setup, 8th International Conference, *ICINCO 2011* Noordwijkerhout, The Netherlands, p. 43-48.
- Greistorfer, P.** (2003), A Tabu Scatter Search Metaheuristic for the Arc Routing Problem, *Computers and Industrial Engineering*, Vol. 44, No. 2.
- Groover, M. P.**, Automation, Production Systems, and Computer-Integrated Manufacturing. Upper Saddle River, New Jersey: Prentice-Hall, USA, 2001.
- Hasegawa, M., Ikeguchi, T., Aihara, K.** (1997), Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problem. *Physical Review Letters*, pp. 2344-2347.
- He, Y., Qiu, Y., Liu, G., Lei, K.** (2005), A parallel adaptive tabu search approach for traveling salesman problems, *Natural Language Processing and Knowledge Engineering*, pp.796,801.
- James, T. L., Brown, E. C. and Keeling, K. B.** (2007), A hybrid grouping genetic algorithm for the cell formation problem, *Computers and Operations Research*, 34, 2059-2079.
- Kaur, D., Murugappan, M. M.** (2008), Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm, *Proceedings of the 2008 annual meeting of the North American fuzzy information processing society*, New York, pp.1-6.
- King, J. R., Nakornchai V.** (1982), Machine-component group formation in group technology: review and extension. *International Journal of Production Research*, 20(2):117-33.
- Kothari, R., Ghosh, D.** (2013), Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods, *European Journal of Operational Research*, Volume 224, Issue 1, Pages 93-100.
- Kumar, K. R., & Chandrasekharan, M. P.** (1990), Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, 28(2), 233-243.
- Kusiak, A., Cho, M.** (1992), Similarity coefficient algorithm for solving the group technology problem, *International Journal of Production Research*, 30(11):2633-46.
- Kusiak, A., Chow, W. S.** (1987), Efficient solving of the group technology problem, *Journal of Manufacturing Systems*, v. 6(2), p. 117-24.
- Lau, H., Sim M., Teo K.** (2003), Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles, *European Journal of Operational Research*, Vol. 148, No. 3.
- Li, X., Baki, M. F., Aneja, Y. P.** (2011), Flow shop scheduling to minimize the total completion time with a permanently present operator: Models and ant colony optimization metaheuristic, *Computers & Operations Research*, Volume 38, Issue 1, Pages 152-164.
- Lim, Y. F., Hong, P. Y., Ramli, R., Khalid, R.** (2011), An improved Tabu Search for solving symmetric traveling salesman problems, *Science and Engineering*, 2011 IEEE Colloquium on, vol., no., pp.851,854.
- Liu, Y., Xiong, S., and Liu, H.** (2009), Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP, *Proceedings of GEC'09*, Shanghai, pp.895-898.
- Mosier, C. T., Taube, L.**, (2009), the facets of group technology and their impact on implementation. *OMEGA*; 13(5):381-91.
- Seifoddini, H., Wolfe, P. M.** (1986), Application of the similarity coefficient method in group technology. *IIE Transactions*; 18(3):271-7.
- Seifoddini, H.** (1989), A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. *International Journal of Production Research*; 27(7):1161-5.
- Selim, M. S., Askin, R. G., Vakharia, A. J.** (1998), Cell formation in group technology: review evaluation and directions for future research. *Computers and Industrial Engineering*, 34 (1), 3-20.
- Semet, F., Taillard, E. D.** (1993), Solving real-life vehicle routing problems efficiently using Tabu Search, *Annals of Operations Research*. Vol. 41, pp. 469-488.



- Sorlin, S., Christine, S.** (2005), Reactive Tabu Search for Measuring Graph Similarity, Graph-Based Representations in Pattern Recognition, *Lecture Notes in Computer Science*, 3434, pp 172-182.
- Spiliopoulos, K., Sofianopoulou, S.** (2008), An efficient ant colony optimization system for the manufacturing cells formation problem, *International Journal of Advanced Manufacturing Technology*, 36, 589-597.
- Tang H., Hooks E.** (2005), A Tabu Search Heuristic for the Team Orienteering Problem, *Computers & Operations Research*, Vol. 32, No. 6.
- Ting, C., Li, S. and Lee, C.** (2003), On the Harmonious Mating Strategy through Tabu Search, *Information Sciences*, Vol. 156, No. 3-4.
- Tsai, C. W., Tseng, S. P., Chiang, M. C. and Yang, C. S.** (2009), A Time-Efficient Method for Metaheuristics: Using Tabu Search and Tabu GA as a Case, *Hybrid Intelligent Systems*, 2009. HIS '09. Ninth International Conference on, vol.2, no., pp.24-29.
- Waghodekar, P. H., Sahu S.** (1984), Machine-component cell formation in group technology MACE. *International Journal of Production Research*, 22, pp. 937-48.
- Wemmerlöv, U., Hyer, N. L.** (1989), Cellular manufacturing in the U.S. industry: a survey of users, *International Journal of Production Research* 27(9), 1511-1530.
- Wu, T. H., Chung, S. H., Chang, C. C.** (2010), A water flow-like algorithm for manufacturing cell formation problems. *European Journal of Operational Research*, 205(2): 346-360.
- Xu, Y. T., Zhang, Y., Huang, X.** (2014), Single-machine ready times scheduling with group technology and proportional linear deterioration, *Applied Mathematical Modeling*, Volume 38, Issue 1, pp. 384-391.
- Xiaodan Wu, Chao-Hsien Chu, Yunfeng Wang, Dianmin Yue** (2007), Genetic algorithms for integrating cell formation with machine layout and scheduling, *Computers & Industrial Engineering*, 53, Issue 2, Pages 277-289.

PRÉ-ANÚNCIO