

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**OTIMIZAÇÃO DO PROCESSO DE MONTAGEM DE COMPONENTES  
ELETRÔNICOS DE SUPERFÍCIE UTILIZANDO FERRAMENTA MULTICRITÉRIO**

**ANDERSON DE OLIVEIRA CASTRO**

**DM: 36/2015**

**UFPA / ITEC / PPGEE**  
**Campus Universitário do Guamá**  
**Belém-Pará-Brasil**  
**2015**

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**ANDERSON DE OLIVEIRA CASTRO**

**OTIMIZAÇÃO DO PROCESSO DE MONTAGEM DE COMPONENTES  
ELETRÔNICOS DE SUPERFÍCIE UTILIZANDO FERRAMENTA MULTICRITÉRIO**

**DM: 36/2015**

**UFPA / ITEC / PPGEE**  
**Campus Universitário do Guamá**  
**Belém-Pará-Brasil**  
**2015**

**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**ANDERSON DE OLIVEIRA CASTRO**

**OTIMIZAÇÃO DO PROCESSO DE MONTAGEM DE COMPONENTES  
ELETRÔNICOS DE SUPERFÍCIE UTILIZANDO FERRAMENTA MULTICRITÉRIO**

Dissertação de Mestrado Acadêmico apresentada à coordenação do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) do Instituto de Tecnologia da Universidade Federal do Pará (UFPA) como parte dos requisitos finais para obtenção do grau de Mestre em Engenharia Elétrica.

**UFPA / ITEC / PPGEE**  
**Campus Universitário do Guamá**  
**Belém-Pará-Brasil**

**2015**

Dados Internacionais de Catalogação-na-Publicação (CIP)  
Sistema de Bibliotecas da UFPA

---

Castro, Anderson de Oliveira, 1983-  
Otimização do processo de montagem de componentes  
eletrônicos de superfície utilizando ferramenta  
multicritério / Anderson de Oliveira Castro. - 2015.

Orientador: Ubiratan Holanda Bezerra;  
Coorientador: Jandecy Cabral Leite.  
Dissertação (Mestrado) - Universidade  
Federal do Pará, Instituto de Tecnologia,  
Programa de Pós-Graduação em Engenharia  
Elétrica, Belém, 2015.

1. Circuitos impressos - montagem. 2.  
Circuitos eletrônicos. 3. Automação industrial.  
4. Otimização combinatória. I. Título.  
CDD 22. ed. 621.38531

---

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“OTIMIZAÇÃO DO PROCESSO DE MONTAGEM DE COMPONENTES  
ELETRÔNICOS DE SUPERFÍCIE UTILIZANDO FERRAMENTA MULTICRITÉRIO”**

AUTOR: ANDERSON DE OLIVEIRA CASTRO

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO  
COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO  
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA  
ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA ELÉTRICA.

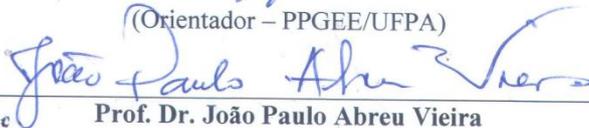
APROVADA EM: 30/10/2015

**BANCA EXAMINADORA:**



Prof. Dr. Ubiratan Holanda Bezerra

(Orientador – PPGEE/UFPA)



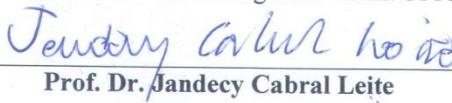
Prof. Dr. João Paulo Abreu Vieira

(Avaliador Interno – PPGEE/UFPA)



Prof.ª Dr.ª Carminda Célia Moura de Moura Carvalho

(Avaliadora Externa ao Programa – FEE/UFPA)



Prof. Dr. Jandecy Cabral Leite

(Avaliador Externo – ITEGAM)

**VISTO:**



Prof. Dr. Evaldo Gonçalves Pelaes  
(Coordenador do PPGEE/ITEC/UFPA)

## **DEDICATÓRIA**

Esta obra é um feito de Deus a quem devo toda honra e glória. Escrevi esta dissertação para meus filhos Pedro Henrique, Helloyse e Alice cujos sorrisos são minha força motriz e que tenham o exemplo que a educação é o caminho para pessoas do bem. Em especial aos meus pais que com amor se dedicaram a minha educação. A minha amada esposa Helhenice pela paciência e compreensão durante todo o período de Mestrado.

## AGRADECIMENTOS

Agradeço a DEUS, pois foi DELE que me foi dada a força, perseverança e a inteligência para prosseguir nesta jornada.

A minha família em especial ao meu pai pelo exemplo de superação e a minha mãe pelo empenho, dedicação e orientação seja no âmbito pessoal quanto no acadêmico. Aos meus avôs pelo exemplo de humanidade deixado e que carregarei pelo resto dos meus dias.

Aos meus queridos familiares e amigos que nos momentos de dificuldade estenderam a sua mão e me encorajaram pelo simples fato de acreditarem em mim.

Em especial aos professores Dr. Jandecy Cabral Leite e Dr. Ubiratan Holanda Bezerra pela confiança, apoio e dedicação durante o período de realização desta dissertação. Ao ITEGAM pelo excelente trabalho de difundir ciência na região amazônica cujo potencial é conhecido internacionalmente, mas não é reconhecido nacionalmente.

Ao convênio UFPA/ITEGAM que possibilitou esta oportunidade de realização de mais um sonho em minha vida. Aos professores do ITEC da UFPA pela sua dedicação nas várias horas tanto dentro quanto fora da sala de aula.

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>XII</b>
<b>LISTA DE TABELAS.....</b>	<b>XIII</b>
<b>LISTA DE SIGLAS.....</b>	<b>XIV</b>
<b>RESUMO .....</b>	<b>XV</b>
<b>ABSTRACT .....</b>	<b>XVI</b>

### CAPÍTULO I

<b>Introdução.....</b>	<b>1</b>
1.1 Uma Abordagem Geral .....	1
1.2 Motivação e Justificativa da Proposta.....	3
1.3 Objetivos .....	4
1.3.1 Objetivo Geral.....	4
1.3.2 Objetivos Específicos.....	4
1.4 Contribuições do Trabalho.....	4
1.5 Delimitações da Pesquisa.....	5
1.6 Organização desta Dissertação .....	5

### CAPÍTULO II

<b>ESTADO DA ARTE .....</b>	<b>7</b>
2.1 A Evolução dos Processos de Montagem de Placas de Circuito Impresso.....	7
2.2 Tipos de Máquinas de Montagem de Componentes de Superfície .....	8
2.3 Matemática Aplicada a Problemas de Combinação em Máquinas SMD .....	12
2.4 Heurísticas Conhecidas na Aplicação de Otimização de Máquinas SMD .....	13

### CAPÍTULO III

<b>METODOLOGIA APLICADA AO ESTUDO.....</b>	<b>18</b>
3.1 A Máquina de Montagem Modelo GXH-3 .....	18
3.1.1 Características do Modelo GXH-3.....	19
3.1.2 Cabeça de Acionamento Direto .....	20
3.1.3 Reconhecimento de Imagens em Movimento.....	21

3.1.4 Alta Densidade de Montagem de Componentes Pequenos.....	22
3.1.5 Retorno da Altura da Placa .....	22
3.1.6 Leitura da Altura da Parte Inferior do Componente .....	23
3.1.7 Contribuições do Software para Melhoria da Produtividade.....	23
3.2 Especificações do Modelo GXH-3 .....	24
3.3 O Problema de Atribuição Quadrática.....	29
3.4 O Problema do Caixeiro Viajante .....	31
3.5 Integrações do Problema de Atribuição e do Problema do Caixeiro Viajante .....	33
3.6 A Modelagem do Problema de Otimização da Máquina GXH-3 .....	35

## **CAPÍTULO IV**

<b>TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVO .....</b>	<b>39</b>
4.1. Como Administrar Vários Objetivos .....	39
4.2. Otimização de Vários Objetivos e Soluções Ótimas de Frente de Pareto .....	40
4.3 Técnicas de Resolução de Problemas de Otimização Multiobjetivo .....	42
4.3.1 A Técnica de Transformação Escalar .....	43
4.3.2 Método das Restrições $\epsilon$ .....	46
4.3.3 Programação por Objetivo .....	49
4.3.4 Programação Multinível.....	51
4.4 Problema de Otimização Multiobjetivo de Número Inteiro .....	54
4.4.1 Problema de Otimização Multiobjetivo dos Caminhos Mais Curtos.....	56
4.4.2 O Problema do Caixeiro Viajante Multiobjetivo .....	61
4.4.3 Outros Trabalhos em Problemas de Otimização Combinatorial Multiobjetivo.....	62
4.4.4 Otimização Combinatorial Multiobjetivo Utilizando Metaheurística .....	63
4.4.5 Otimização Combinatorial Multiobjetivo Utilizando Algoritmos Genéticos.....	66
4.4.5.1 Algoritmo Genético de Pareto Aglomerado (NPGA).....	67
4.4.5.2 Algoritmo Genético Baseado na Média (WBGA) .....	67
4.4.5.3 Algoritmo Genético de Ordenação Não-Dominada (NSGA-II) .....	68
4.4.5.4 Algoritmo Evolucionário Baseado na "Força" de Pareto (SPEA) .....	72

## **CAPÍTULO V**

<b>IMPLEMENTAÇÃO DA TÉCNICA PROPOSTA UTILIZANDO A FERRAMENTA DE ALGORITMO EVOLUTIVO – NSGA II .....</b>	<b>74</b>
---	-----------

5.1. Solução do Problema da Rota de Montagem dos Componentes .....	75
5.2. Solução do Problema de Atribuição de Feeder .....	78
5.3. Solução do Problema da Sequência de Montagem .....	80
5.4. A Resolução do Problema Aplicando o NSGA-II .....	82
5.4.1 Codificação do Problema .....	84
5.4.2 Seleção e cruzamento .....	86
5.5. O Código de Aplicação da Ferramenta Multicritério .....	88
5.6. Os Resultados da Aplicação .....	92

## **CAPÍTULO VI**

<b>CONCLUSÃO .....</b>	<b>95</b>
------------------------	-----------

6.1 Recomendações para Trabalhos Futuros .....	95
--	----

<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>97</b>
---	-----------

<b>ANEXOS .....</b>	<b>109</b>
---------------------	------------

<b>ARTIGOS PUBLICADOS .....</b>	<b>111</b>
---------------------------------	------------

## LISTA DE FIGURAS

<b>Figura 3.1:</b>	Configuração da máquina de montagem GXH-3	33
<b>Figura 3.2:</b>	Esquema da cabeça de acionamento direto	34
<b>Figura 3.3:</b>	Foto tirada para reconhecimento da condição de pickup dos componentes	35
<b>Figura 3.4:</b>	Ajuste de acordo com a altura da placa devido ao empenamento	37
<b>Figura 3.5:</b>	Ajuste da altura da superfície inferior do componente	37
<b>Figura 3.6:</b>	Decomposição horizontal e vertical	39
<b>Figura 3.7:</b>	Face do empenamento	39
<b>Figura 3.8:</b>	Conveyor de transporte de PCI dianteiro e traseiro	39
<b>Figura 3.9:</b>	Reconhecimento de fiducial	40
<b>Figura 3.10:</b>	Limites para a área da marca de fiducial	41
<b>Figura 3.11:</b>	Exemplo 1: Montagem de 3 componentes adjacentes	43
<b>Figura 3.12:</b>	Fluxograma de funcionamento da máquina de montagem GXH-3	50
<b>Figura 4.1:</b>	Exemplo de uma curva de Pareto	55
<b>Figura 4.2:</b>	Pontos fracos e restritos da frente de Pareto	56
<b>Figura 4.3:</b>	Soma ponderada no caso da curva de Pareto convexa	59
<b>Figura 4.4:</b>	Soma ponderada no caso da curva de Pareto não-convexa	60
<b>Figura 4.5:</b>	Método das restrições $\epsilon$ no caso da curva de Pareto não-convexa	62
<b>Figura 4.6:</b>	Ótimos de Pareto suportado e não suportado	69
<b>Figura 4.7:</b>	Representação gráfica da distância de multidão	84
<b>Figura 4.8:</b>	Procedimento do NSGA-II	85
<b>Figura 5.1:</b>	Distâncias internas da máquina modular GXH-3	89
<b>Figura 5.2:</b>	Rota de montagem de acordo com a restrição das alturas dos componentes	92
<b>Figura 5.3:</b>	Codificação do gene para o problema de atribuição	99
<b>Figura 5.4:</b>	Arranjo de alimentadores	99
<b>Figura 5.5:</b>	Codificação do gene para o problema da rota de montagem	100
<b>Figura 5.6:</b>	Fluxograma do algoritmo principal	102
<b>Figura 5.7:</b>	Solução caso 1: Arranjo de feeder dos quatro racks	107
<b>Figura 5.8:</b>	Sequência de montagem da primeira cabeça para o caso 1	107

## LISTA DE TABELAS

<b>Tabela 3.1:</b> Relação dos componentes e suas embalagens.....	27
<b>Tabela 3.2:</b> Quantidade máxima de nozzles por tipo de componente.....	28
<b>Tabela 3.3:</b> Número máximo de alimentadores por tamanho de componente.....	28
<b>Tabela 5.1:</b> Ordenação dos componentes pela sua altura (placa caso 1).....	76
<b>Tabela 5.2:</b> Tempo total de ciclo utilizando o software do fabricante.....	93
<b>Tabela 5.3:</b> Tempo total de ciclo utilizando o algoritmo proposto.....	94
<b>Tabela 5.4:</b> Comparativo do desempenho dos dois métodos.....	94

## LISTA DE SIGLAS

<b>SMT</b>	Surface Mounting Technology
<b>THT</b>	Through Hole Technology
<b>NSGA</b>	Non Sorting Genetic Algorithm
<b>NSGAI</b>	Non Sorting Genetic Algorithm II
<b>PCI</b>	Placa de Circuito Impresso
<b>CPH</b>	Components Per Hour
<b>NP</b>	Non Polynomial
<b>SMD</b>	Surface Mounted Device
<b>MELF</b>	Metal Electrode Leadless Face
<b>QFP</b>	Quad Flat Package
<b>PLCC</b>	Plastic Leaded Chip Carrier
<b>ISP</b>	Input Shaping Technique
<b>TDC</b>	Time Delay Control
<b>ABC</b>	Artificial Bee Colony
<b>MIP</b>	Mixed Integer Problem
<b>MOEA</b>	Multiobjective Evolutionary Algorithm
<b>PEC</b>	Photo Enhanced Camera
<b>CAD</b>	Computer Aided Design
<b>AP</b>	Attribution Problem
<b>QAP</b>	Quadratic Assignment Problem
<b>TSP</b>	Traveller Salesman Problem
<b>SPP</b>	Shortest Path Problem
<b>MOSPP</b>	Multiobjective Shortest Path Problem
<b>MQAP</b>	Multiobjective Quadratic Assignment Problem
<b>VEGA</b>	Vector Evaluated Genetic Algorithm
<b>NPGA</b>	Niched Pareto Genetic Algorithm
<b>MOGA</b>	Multiobjective Genetic Algorithm
<b>SPEA</b>	Strength Pareto Evolutionary Algorithm
<b>FAP</b>	Feeder Assignment Problem
<b>MSP</b>	Mounting Sequence Problem
<b>PMX</b>	Partially Matched Crossover
<b>CX</b>	Cyclic Crossover
<b>OX</b>	Ordered Crossover

## RESUMO

Os problemas de otimização de processos industriais atraíram muitas pesquisas no início dos anos 90 do século passado. O aumento do volume de produção, o aumento da competitividade do mercado e os avanços tecnológicos pressionaram as indústrias a buscar soluções de baixo custo e implementação rápida. Um dos processos que se tornou a razão principal para o aumento do volume de vendas foi o processo de montagem de placas de circuito impresso, cuja tecnologia de montagem em superfície composta por máquinas de serigrafia, montadoras de chips automatizadas e fornos de refusão representou um avanço, pois substituiu o processo anterior que empregava a tecnologia através de furos. Esta pesquisa formulou um novo ponto de vista para otimização de montadoras modulares de chip baseada no conhecido problema de arranjo de alimentadores e no problema de movimentação da cabeça de montagem aplicando de forma global um Algoritmo Genético de Multicritério Não Dominante (NSGA-II) cujo objetivo foi a redução do tempo total do ciclo de montagem. A modelagem das funções de aptidão foi apresentada e a ferramenta de otimização multicritério foi descrita utilizando as funções da máquina e suas respectivas restrições. O mesmo método poderia ser aplicado para descrever outros tipos de máquinas para auxiliar pesquisas futuras.

**PALAVRAS-CHAVES:** Otimização Multicritério, montadora de chips, algoritmo genético não dominante, função objetivo.

## ABSTRACT

The optimization problems of industrial process attracted many researches since the early 90's of the last century. The production volume increase, market competition increase and technological advances pushed the industries to seek for low cost and quick implementation solutions. One of the processes that became the core reason for increasing the sales volumes was the surface mount technology composed by printing, automated chip mounting and reflow oven which replaced the through hole technology. This research formulated a new point of view for optimization of modular chip mounters based on the already known *feeder* assignment problem and assembly head motion problem applying the global optimization using the non sorting dominance genetic algorithm the NSGAI in regard of the total cycle time reduction. The modeling of the fitness functions was presented and the multi criteria optimization tool was described using the machine functions and constraints. The same method could be applied to describe other type of machines to support future research.

**KEYWORDS:** *multicriteria optimization, chip moulder, nonsorting genetic algorithm, objective function.*

# CAPÍTULO I

## INTRODUÇÃO

### 1.1 Uma Abordagem Geral

O aumento da competitividade e a mudança do modo de pensar do consumidor que passa a exigir um produto de qualidade a um preço mais baixo levaram as indústrias do ramo de eletrônicos a buscar maior rapidez, eficiência e custo reduzido para se manterem no mercado. Para que isso ocorresse às indústrias recorreram a automatização de seus processos seguindo o modelo japonês. No entanto a automatização gerou um novo problema: "Qual é a melhor maneira de se utilizar a automação para alcançar os índices de qualidade e produtividade necessários?".

Em meio a um cenário competitivo onde a inovação tecnológica reduz cada vez mais a vida útil dos produtos e altas demandas surgem para serem atendidas em um curto espaço de tempo as empresas buscaram aperfeiçoar seus processos automatizados para desta forma conseguir obter os lucros. Apesar do investimento inicial em processos automatizados ser relativamente alto se comparado a processos manuais os ganhos obtidos ao passar do tempo justifica-o e a melhoria obtida nesses processos representa um ganho significativo no volume de produção.

Na indústria de eletrônicos os processos automatizados são amplamente aplicados na fabricação das partes que compõe os produtos até mesmo na integração final onde as partes são montadas e tornam-se o produto acabado. Em meio às partes que integram o produto acabado, as placas de circuito impresso (PCI) são as responsáveis por todas as funções dos produtos eletrônicos. Este componente passou por um processo evolutivo que visa o aumento de peças produzido no menor tempo possível. A PCI evoluiu de um processo manual passando por um processo automatizado baseado na tecnologia de montagem por furos *Through Hole Technology* (THT), ainda empregada em alguns tipos de placa, até chegar à tecnologia de montagem de superfície *Surface Mounting Technology* (SMT) (CRAMA et al., 1997). O processo de montagem de componentes em superfície compreende a serigrafia da pasta de solda onde uma liga de estanho e prata é depositada através de um gabarito de metal sobre a placa, este gabarito é perfurado de modo que os mesmos coincidam com os pontos de cobre contidos na placa de circuito impresso. Em seguida as placas passam pelo processo de

montagem dos componentes onde os mesmos são alocados na placa conforme a sequência de montagem determinada pela programação da máquina de montagem de chips (componentes em miniatura cujos terminais estão no corpo do componente). Já no último estágio as PCI's passam por um forno de refusão que é dividido em zonas de temperatura diferentes traçando um perfil para que todos os componentes possam ser soldados (LEU et al., 1993). Após a finalização do processo SMD a placa passa para o estágio de calibração e testes até ser montada no produto acabado.

Entre os processos que compõe a fabricação da placa de circuito impresso, a montagem de componentes é a que desperta maior interesse, pois quanto maior for o CPH (componentes por hora) das máquinas de montagem maior será a sua capacidade de produção. Este processo pode ser limitado por dois fatores inerentes à organização que são: a quantidade de setups, processos de configuração das máquinas para a produção de uma determinada placa, e a programação que depende da quantidade, tipo, tamanho e localização física de montagem dos componentes na placa. Deste modo a tarefa de montagem acaba se tornando um processo limitante que é conhecido como gargalo ou "*bottleneck*", que por sua vez acaba sendo o principal processo a ser otimizado (CSASZAR et al., 2000), (MOYER; GUPTA, 1997) e (TIRPAK et al, 2000).

Como mencionado por (GASTEL, 2002), (GROTZINGER, 1992) e (KHOO; LOH, 2000) as máquinas de montagem conhecidas como "*chip mounter*" passaram por uma evolução desde a configuração de cabeça de montagem única (*single mounting head*) e alimentadores fixos (*fixed feeders*) (BENTZEN, 2000), passando para o próximo estágio onde foram desenvolvidas as máquinas de cabeças de montagem múltiplas (*multi mounting head*) e múltiplos alimentadores fixos (*multi fixed feeders*), até o conceito atual de máquinas de montagem modulares onde a máquina é dividida por estágios, cada um contendo de uma à duas cabeças múltiplas de montagem e alimentadores frontais e/ou traseiros fixos. Devido as diferentes características de construção das máquinas, diferentes tipos de restrições devem ser aplicadas para melhor descrevê-la (WANG et al., 1999).

De fato como demonstrado por (SHIH et al., 1996) o desempenho do software de otimização, fornecido pelos fabricantes das máquinas de montagem, não é em muitos casos eficiente. Como afirma (MAGYAR et al., 1999) os fornecedores de máquinas de montagem ainda não conseguiram resolver um problema de máquina de maneira eficiente. Isto demonstra claramente a necessidade de pesquisa deste problema com o objetivo de melhorar o desempenho industrial.

## 1.2 Motivação e Justificativa da Proposta

A tecnologia SMT (Tecnologia de Montagem em Superfície) avançou devido o aumento da demanda por produtos cada vez mais portáteis. A miniaturização evoluiu de acordo com o surgimento de novas tecnologias que exigiram maior integração de componentes *in chip*, ou seja, componentes passivos antes espalhados pela placa de circuito impresso agora são integrados aos componentes multiprocessados.

A demanda baixa e a alta quantidade de produtos devido ao portfólio variado que as empresas de eletroeletrônicos apresentam no mercado representa um grande impacto na capacidade produtiva das placas de circuitos eletrônicos. Os diversos problemas que o processo SMD (Dispositivo de Montagem em Superfície) enfrenta vão desde o manuseio e controle correto dos componentes da solda e processos correlatos, passando pelo processo de montagem finalizando com o processo de inspeção de qualidade. Dentre estes o de montagem é o que desde o princípio da introdução do SMT vem despertando várias pesquisas.

Ao realizar o balanceamento de uma linha SMD é possível identificar que a seção de montagem de componentes representa o fator limitante da capacidade de produção, não somente pelo processo de montagem em si, mas pelo fato de ao realizar a troca de um modelo para o outro, a configuração de uma máquina de montagem é o processo que toma maior tempo. Deste modo, otimizar a performance de uma máquina de montagem significa aumentar a produtividade da linha como um todo.

O conceito de otimizar uma máquina de montagem conta com uma série de fatores que levam em consideração as características construtivas da máquina, ou seja, o procedimento é customizado de acordo com a máquina. Desta forma fica impossível definir uma única técnica capaz de definir o problema de otimização de todas as máquinas de montagem SMD. Em termos computacionais, a otimização de uma máquina de montagem é muito difícil e para tais problemas define-se um termo conhecido como *NP-hard*, onde a definição determina a complexidade do problema que para o caso estudado muitas vezes não é capaz de encontrar uma solução exata e sim aproximada em um tempo de processamento hábil.

O que desperta grande interesse neste campo é que não existe uma técnica mais apropriada e sim métodos aproximados, isto é, o procedimento de otimizar uma máquina de montagem SMD é tão complexo que apresenta um grande desafio para modelá-lo e resolvê-lo, isto representa um processo empírico onde a observação do processo de montagem de uma

máquina deve ser descrita em detalhes para que seja possível resolvê-lo ou pelo menos obter uma resposta que reflita em uma boa performance da máquina.

### **1.3 Objetivos**

#### **1.3.1 Objetivo Geral**

Propor uma metodologia de otimização para o processo SMT utilizando técnicas de algoritmos evolutivos multiobjetivo para aplicações industriais.

#### **1.3.2 Objetivos Específicos**

Esta dissertação tem os seguintes objetivos específicos:

- Realizar a descrição detalhada das características da máquina de montagem modelo GXH-3 como forma de embasar com maior fidelidade o funcionamento da máquina;
- Apresentar um modelo matemático da máquina modular fabricada pela Hitachi modelo GXH-3 como base para modelagem e aplicação do mesmo tipo de máquina modular com diferentes parâmetros;
- Apresentar as técnicas de otimização multicritério mais aplicadas para problemas combinatoriais;
- Desenvolver um novo método de otimização da mesma GXH-3 utilizando uma ferramenta multicritério que usa como base o algoritmo genético. Esta ferramenta multicritério é conhecida pela sigla NSGAI II (*NonSorting Genetic Algorithm II*, ou Algoritmo Genético Não Ordenado II) cuja capacidade de avaliar dois ou mais objetivos é necessária para representar todos os parâmetros e restrições desta "chip mounter".

### **1.4 Contribuições do Trabalho**

Até o presente momento, a literatura não apresenta uma única metodologia para resolução do problema de otimização de máquinas de montagem de componentes SMD, e sim um grupo de técnicas que abordam os problemas separadamente conhecida como metaheurística, assim como em outras máquinas, os problemas de atribuição na GXH-3 são

solucionados com algoritmos de nós e ramos (*branch-and-bound algorithm*) e algoritmo ganancioso (*greedy algorithm*), já para resolver o problema da sequência de montagem um algoritmo de pesquisa local foi empregado, formando assim a metaheurística pré-definida para problemas envolvendo combinações. A solução apresentada nesta dissertação utiliza o método de otimização multicritério baseado na técnica empregada pelo algoritmo genético não ordenado conhecido como NSGAI.

### 1.5 Delimitações da Pesquisa

Apesar da inter-relação entre os problemas que caracterizam as máquinas de montagem de componentes SMD, tornando difícil a tarefa da otimização da mesma, várias técnicas foram desenvolvidas para escolha da melhor condição inicial dentro do espaço de busca. Esta condição inicial visa reduzir o tempo computacional porém, influencia a pesquisa dentro do espaço de modo a "guiar" o algoritmo para a solução mais aceitável. No entanto, regiões do espaço de busca podem apresentar soluções mais interessantes e até melhores das que são encontradas pela pesquisa "guiada", não garantindo que o máximo ou o mínimo do espaço seja localizado de modo a tornar a pesquisa ineficiente.

A proposta apresentada nesta dissertação foi oriunda de uma extensiva pesquisa *in loco*, onde a observação do funcionamento da máquina de montagem e a entrevista com os especialistas da mesma resultaram na descrição matemática de seu funcionamento. Após formulação, as restrições foram consideradas para que o modelo fosse aplicado ao algoritmo baseado na técnica de algoritmo genético multiobjetivo para avaliação das condições de produção da mesma, comparando os resultados obtidos com a técnica proposta com os resultados já disponibilizados baseados na ferramenta fornecida pelo fabricante.

### 1.6 Organização desta Dissertação

Esta dissertação está organizada em seis capítulos e estão estruturados da seguinte forma:

- **Capítulo 1** - Introduz o conceito de produção industrial do ramo de eletroeletrônicos e seus processos produtivos enfatizando o processo de montagem de componentes em superfície e estabelece os objetivos a serem alcançados;

- **Capítulo 2** - Uma pesquisa extensa é realizada levantando um breve histórico sobre a tecnologia de montagem em superfície, onde o foco concentra-se nas máquinas de montagem de componentes SMD e as heurísticas aplicadas recentemente;
- **Capítulo 3** - Uma descrição extensa das características construtivas da máquina de montagem GXH-3 é apresentada seguida por uma formulação matemática utilizada como base para apresentar o problema de otimização da mesma;
- **Capítulo 4** - As técnicas de otimização multiobjetivo mais conhecidas atualmente para problemas de combinação são apresentadas, bem como técnicas híbridas ou envolvendo algoritmos evolutivos;
- **Capítulo 5** - O problema de otimização multiobjetivo é construído baseado no tratamento do problema de atribuição de alimentador (FAP) e no problema da sequência de montagem (MSP). Os resultados da pesquisa e da elaboração do código são expostos neste capítulo;
- **Capítulo 6** - Os resultados desta dissertação são discutidos neste capítulo bem como os trabalhos futuros a serem realizados tomando como ponto de partida esta dissertação.

## CAPÍTULO II

### ESTADO DA ARTE

A incessante busca do aumento da produtividade no chão de fábrica tem sido a força motriz que tem incentivado as empresas a investirem na melhoria de seus processos, seja ela na manutenção preventiva visando a redução de paradas ou na otimização de seus processos, seja ele automatizado ou não, com o intuito de alcançar as metas de melhoria.

#### 2.1 A Evolução dos Processos de Montagem de Placas de Circuito Impresso

A eletrônica, ramo da engenharia elétrica, teve seu nascimento no início do século 20 com a primeira transmissão de sinal feita por Marconi em dezembro de 1901 utilizando código Morse, desde então a eletrônica tem aplicado o princípio da criação, controle e transmissão de sinais. Desta forma o equipamento desenvolvido até então utilizava apenas componentes e laços condutivos entre os mesmos. Os componentes evoluíram e uma gama desde centelhadores, válvulas termiônicas e componentes passivos como indutores, capacitores e resistores eram todos conectados com uma grande variedade de fios muitas vezes conectados aos componentes por meio de parafusos.

Nos trabalhos de (STRAUSS, 1998) e (AYOB, 2005) é apresentada que a natureza tridimensional da montagem dos equipamentos eletrônicos do início trazia dois problemas graves: o primeiro foi a impossibilidade de utilização de processos mecânicos para aumentar a quantidade de equipamentos produzidos; e o segundo que a inspeção de qualidade era difícil de ser realizada e os testes sempre eram funcionais e localizar os problemas era uma tarefa bastante difícil. Para mudar este cenário difícil, Paul Eisler inventou a placa de circuito impresso em 1943, a qual substituiu as ligações de fios tridimensionais em um plano bidimensional obedecendo a um padrão formado por faixas de cobre fixadas em uma das faces de uma placa de papel fenolite. Toda vez que um condutor precisava ser conectado a um componente, um furo era feito na placa e o terminal do componente era inserido e logo em seguida era soldado. Todos os componentes sejam eles axiais ou radiais eram montados na face do fenolite e os seus terminais eram encaixados até que seus terminais ficassem expostos na face do cobre onde posteriormente eram soldados um a um, a esta tecnologia foi dada o nome de *Through Hole Technology* (THT).

O próximo passo na evolução da fabricação dos equipamentos eletrônicos foi o desenvolvimento da solda por onda, o que trouxe maior automatização para o processo produtivo e conseqüentemente aumentou o volume de produção. Porém a montagem das placas ainda se mantinha tridimensional em relação ao componente e bidimensional em relação ao plano dos condutores, ainda era uma tarefa difícil de ser realizada. Por volta de 1960 uma patente britânica menciona um dispositivo montado em superfície (*Surface Mounting Device* (SMD)), cujos terminais não eram inseridos na placa de fenolite.

Assim como a invenção de Eisler esta nova patente incentivou o desenvolvimento de placas híbridas contendo ambos componentes THT e SMD. Apesar das vantagens apresentadas pela nova tecnologia houve resistência por parte da indústria que inicialmente previu que um grande investimento deveria ser feito com equipamentos, porém, as vantagens apresentadas pela tecnologia suprimiram as desvantagens geradas pelo investimento inicial.

Como descrito por (GUPTA, 1996) devido ao advento dos circuitos integrados com múltiplas funções, o número terminais teve um crescimento para além 68 o que tornou o cenário ainda mais vantajoso para a tecnologia de componentes montados em superfície. Ao mesmo tempo em que os componentes foram sendo miniaturizados, os desenvolvedores das máquinas de montagem começaram a trocar informações para melhorar o desempenho do processo SMT como um todo.

Uma visão geral é abordada por (LEE, 2001) e (SCHMITT, 2006) que apresenta um levantamento geral da capacidade do processo SMT enumerando os principais processos como o de serigrafia de pasta de solda (*solder printing*), montagem de componentes (*pick-and-place machine*) e soldagem (forno de refusão). Para cada processo, os pontos de perda são levantados e métodos para suas medições são apresentados. Em específico, o processo de montagem de componentes apresenta perdas relacionadas aos desvios como os angulares e correção de posicionamento de placa ou global conhecido como ajuste de fiducial.

## **2.2 Tipos de Máquinas de Montagem de Componentes de Superfície**

Como Strauss comenta em seu trabalho, as máquinas de pegar e montar (*pick-and-place machine*) são os equipamentos que montam os componentes SMD nas placas de circuito impresso e são responsáveis pelas seguintes tarefas:

1. Pegar o componente correto de uma posição selecionada de um dos alimentadores;

2. Montar na posição correspondente ao tipo de componente adquirido de maneira correta e com a polaridade correspondente.

Estas tarefas ficam mais evidentes e demandam um nível de complexidade cada vez maior com o aumento da variedade de componentes e com a miniaturização cada vez maior dos mesmos, fazendo com que as trilas e ilhas da PCI sejam projetadas para serem cada vez menores e encurtando a distância entre os componentes montados. Alguns componentes como o *Metal Electrode Leadless Face* (MELF) está sendo miniaturizado, enquanto que outros componentes como o *Quad Flat Package* (QFP) estão aumentando em tamanho pelo fato do aumento do número de terminais devido à da integralização cada vez maior de funções em um circuito integrado, como por exemplo, os microcontroladores para sistemas embarcados. Minimelfs (0204 =  $3,6 \times 1,4 \text{ mm}$ ) e micromelfs (0102 =  $2,0 \times 1,1 \text{ mm}$ ) já são amplamente aplicados devido às suas características de confiabilidade, porém por possuir um formato cilíndrico, a montagem deste tipo de componente necessita ser tratada de forma diferenciada dos demais.

Em contraste, o *Plastic Leaded Chip Carrier* (PLCC) e o QFP aumentaram em tamanho atingindo dimensões além de  $29,4 \text{ mm}$ . São estas demandas que impulsionam o desenvolvimento de novos equipamentos capazes de realizar a montagem em meio às tendências do desenvolvimento dos componentes.

Dependendo do procedimento de soldagem da placa, seja ele por solda à onda ou solda por refusão, o método de montagem deve ser seguido por um processo a mais. No processo de solda à onda antes da montagem dos componentes SMD é necessário depositar um adesivo entre as ilhas ou *pads* (parte exposta do cobre da placa em contato com o terminal do componente), e logo em seguida a placa passa por um processo de cura onde o adesivo enrijece e mantém os componentes na posição correta, somente após o processo de cura é que as placas podem passar pelo processo de soldagem. Com relação ao método de soldagem SMD, é necessário depositar a solda em forma de uma pasta cobrindo o *pad* como um todo utilizando uma máscara, como em um processo de serigrafia. Logo em seguida a montagem é iniciada e após seu término a placa passa por um forno de refusão onde resistências elétricas de alta potência fornecem calor que é direcionado para a placa através de ventiladores instalados dentro do forno.

Ao realizar um comparativo entre os processos THT e SMT, pode-se observar que o processo de montagem SMT é mais simples, pois uma única máquina pode montar uma grande variedade de componentes enquanto que no processo THT existem máquinas

específicas para os tipos de componente, por exemplo, para componentes radiais e axiais duas máquinas são necessárias.

As máquinas de montagem de componentes em superfície (SMD) podem ser divididas de acordo com as características desenvolvidas para atender os requisitos de montagem. A máquina denominada como modelo básico e modelo de alcance médio é composta por até 100 alimentadores capazes de comportar uma variedade de tipos de embalagem como o rolo, a vareta e a bandeja de componentes.

Como uma linha é composta por várias máquinas os alimentadores podem ser montados individualmente com qualquer tipo de embalagem observando a limitação física dos espaços dos mesmos. Ainda como característica os alimentadores são montados em suporte móveis conhecidos como carrinhos de alimentação que são facilmente trocados fazendo com que o posicionamento dos alimentadores em um setup de linha seja feito antecipadamente. Basicamente as máquinas do modelo básico e médio são compostas por uma única cabeça de montagem e são conhecidas como “*pick-and-place machines*” (máquinas de aquisição e montagem), cuja característica de montar um componente por vez confere uma capacidade de montagem de componentes por hora - CPH de até 4.000 posições, porém levando em consideração o movimento realizado pela cabeça entre as posições de aquisição e de montagem e do início e parada de movimentos, na prática, este valor é reduzido para 2.400. Já nas máquinas de modelo Premium a quantidade de cabeças aumenta e até mesmo o tipo de cabeça muda para adquirir mais componentes em uma só operação, com isto, o CPH destas máquinas atinge facilmente valores superiores a 40.000.

Além das máquinas de montagem mencionadas anteriormente é possível distinguir outras três categorias que são divididas em:

- Máquinas de montagem de terminais próximos: A sua acuracidade de montagem varia em torno de  $0.4\text{ mm}$  e é capaz de montar componentes de até  $55\text{ mm}^2$ , o que representa uma acuracidade alta em termos de posicionamento  $XY$  e rotacional. Esta característica é proveniente dos controles de imagem do posicionamento dos componentes e reconhecimento do posicionamento da placa antes da montagem, esta rede de sensoriamento faz com que seja possível ajustar a posição do componente e a força com que o mesmo é montado na placa. Geralmente são compostas de uma única cabeça de montagem com atuadores únicos ou múltiplos para realizar a aquisição dos componentes;

- Máquinas de montagem de chips de alta velocidade: Os diferentes fabricantes usaram métodos distintos para desenvolver este tipo de máquina. Um exemplo pode ser dado de um sistema onde a cabeça de montagem é do tipo tambor ou rotacional composta de até 12 *nozzles* (atuador pneumático que utiliza o vácuo para pegar o componente), cada um adquirindo um tipo de componente. Para o caso de máquinas com duas cabeças na mesma configuração, o acesso às posições de montagem da PCI é feito separadamente, ou seja, enquanto a primeira cabeça está pegando os componentes a segunda está montando os componentes do ciclo anterior e vice versa, isto é feito para evitar colisões entre as mesmas. Para montagem de componentes com muitos terminais, ao invés de duas cabeças apenas uma pode ser usada e outra cabeça para deposição de adesivo pode ser instalada para aplicações mistas de placas THT e SMD.
- Máquinas de montagem de velocidade muito alta: Estas máquinas de alto desempenho podem atingir um CPH que varia entre 20.000 a 60.000 variando o número de módulos que constituem a máquina e por consequência o número de cabeças. As cabeças destas máquinas operam nos eixos XY cada uma montando regiões diferentes da mesma PCI. Já existe uma máquina onde duas cabeças acessam simultaneamente a PCI para realizar o procedimento de montagem.

(AYOB; KENDALL, 2008) realizaram um trabalho de pesquisa onde classificam os diferentes tipos de máquina de montagem de componentes SMD de acordo com as suas características tais como: entrega dupla, multi-estações, tipo roleta, multi-cabeças e *pick-and-place* sequencial. De acordo com essa classificação os métodos heurísticos são associados para melhor descrever o seu funcionamento e obter o ponto ótimo de operação.

(FUKUSHIMA et al., 2008) apresentam a estrutura da máquina de montagem de componentes SMD modular modelo GXH-3, produzida pelo fabricante Hitachi. Devido a sua característica modular de multi cabeças e dois braços robóticos de montagem por módulo, a sua capacidade de montagem pode atingir 95.000 chips por hora. Outro fator que torna esta máquina atrativa quanto à qualidade e acuracidade de montagem é devido ao reconhecimento da imagem do componente depois do *pickup* (*gang-fly recognition*) de modo a visualizar todos os componentes de todas as cabeças de montagem faz com que a mesma seja amplamente aplicada no mercado. Entre todas as características a otimização realizada nos movimentos da cabeça de montagem e da sequência de montagem são aprimoradas com a adição da otimização da angulação dos componentes sem polaridade (ex: resistores, indutores

e capacitores cerâmicos). O software de balanceamento de máquinas também incorpora o sistema com o objetivo de redução do tempo de setup.

### 2.3 Matemática Aplicada a Problemas de Combinação em Máquinas SMD

A programação linear foi introduzida para representar por meios matemáticos os problemas reais. A programação linear tem como objetivo minimizar uma função linear sendo o domínio, ou seja, o espaço de busca definido por um poliedro (MONTICELI, 2010), em outras palavras, as restrições são funções que limitam o espaço de soluções e descrevem um poliedro fechado e a função objetivo remonta o processo principal a ser estudado onde a maximização ou minimização de um critério descrito pela função objetivo se torna o alvo da pesquisa. O método da eliminação Fourier-Motzkin introduzido no século XVIII, foi um dos primeiros a aplicar um algoritmo aplicando os conceitos da programação linear. Com o advento das indústrias, vários problemas surgiram e a programação linear foi empregada para resolvê-los. Este campo é conhecido como pesquisa operacional e envolve problemas como o de fluxo de rede e problemas de multi-comodites. Existem casos onde a solução exige que o resultado pertença ao conjunto dos números inteiros. Para estes casos o problema de programação linear é transformado para adaptar-se a esta condição, onde as variáveis pertencem ao conjunto dos números inteiros e as restrições são modificadas para refletirem um espaço de soluções de números inteiros. Quando temos variáveis inteiras e variáveis contínuas que definem o mesmo problema de otimização temos o que é conhecida como Programação Inteira Mista. Um algoritmo clássico proposto por (LAND; DOIG, 1960) em seu trabalho intitulado "*An Automatic Method of Solving Discrete Programming Problems*" um limite inferior e um limite superior são determinados, onde tais limites definem o espaço de busca para realizar a otimização. Com efeito, esta delimitação faz com que qualquer solução fora do intervalo seja descartada e o algoritmo alcance a convergência em um tempo aproximadamente polinomial. Tal procedimento é aplicado a problemas clássicos de otimização como o do caixeiro viajante e o problema da mochila.

Assim como problemas que envolvem a pesquisa operacional, os problemas de combinação são os mais presentes no âmbito industrial e para descrevê-los a teoria de atribuição foi desenvolvida como forma de modelagem matemática. Como apresentado por (SENNE et al., 2006), os problema de atribuição pode ser generalizado como sendo a atribuição de  $n$  tarefas a  $m$  agentes e que seja garantido que apenas uma tarefa seja atribuída a

um agente apenas e que todas as tarefas sejam atribuídas. O processo de atribuir uma tarefa  $n$  a um agente  $m$  gera certo custo denominado  $c$ . Os problemas de atribuição lidam com a questão de como atribuir  $n$  itens, por exemplo, tarefas a  $m$  máquinas ou trabalhadores da melhor maneira possível. Estes problemas consistem de dois componentes: a atribuição que se apresenta na forma de uma estrutura combinatorial e a função objetivo que indica o melhor caminho. Assim como comentado por (LOIOLA et al., 2004) Um problema clássico do âmbito industrial que envolve a aplicação de uma variação do problema de atribuição é o problema da cadeia de fornecimento também conhecido como problema de transporte. Neste caso é aplicado o problema de atribuição quadrática, onde ao atribuir a tarefa  $n$  ao agente  $m$  deve ser considerada a distância entre os pares e o fluxo de demanda entre os mesmos.

Uma variação do problema de atribuição que tem como objetivo encontrar a menor distância é mais conhecido como problema do caminho mais curto. A estes casos podem ser aplicadas técnicas de fluxo de rede onde o problema pode ser representado através de um gráfico onde o fluxo mais curto encontrado representa a solução. Na pesquisa de (BIRKHOFF, 1946) as diversas propriedades da representação gráfica do fluxo de rede são provadas e algumas propriedades foram demonstradas por (BALINSKI; RUSSAKOFF, 1974). Para que haja um caminho em um gráfico definido por um poliedro é necessário que todos vértices do poliedro sejam tomados em pares conectados por uma aresta. O teorema de (HALL, 1935) descreve as condições necessárias para que ocorra a máxima correspondência em gráficos delimitados pela forma geométrica de um poliedro.

Os algoritmos mais empregados para determinação de um fluxo de rede foram desenvolvidos por (BELLMAN, 1958), (FORD, 1956) e (MOORE, 1959). Este algoritmo ficou conhecido como algoritmo de Bellman-Ford-Moore e trata problemas onde existe a possibilidade das arestas apresentarem valores negativos. Para os casos onde o problema apresenta somente valores de arestas positivos o algoritmo de (DIJKSTRA, 1959) é o mais indicado.

#### **2.4 Heurísticas Conhecidas na Aplicação de Otimização de Máquinas SMD**

(CHANG; PARK, 2001) direcionam a sua pesquisa para a otimização do movimento ponto-a-ponto da cabeça de montagem onde um controle de malha é implementado visando uma redução da influência de vibrações externas e o aumento da acuracidade. Duas técnicas são aplicadas para obtenção de um controle robusto, a primeira foi denominada como a

técnica de manipulação da entrada *Input Shaping Technique* (ISP) e a segunda é o controle por atraso de tempo *Time Delay Control* (TDC).

(DEMIRKALE et al., 2010) desenvolveram uma aplicação de um algoritmo de colônia, baseado no comportamento das abelhas *Artificial Bee Colony* (ABC), empregado para descrever e otimizar a operação de uma máquina de montagem de componentes SMD.

(SUN et al., 2005) tratam o problema de otimização de uma máquina de montagem de componentes SMD com dois braços de montagem e múltiplas cabeças de montagem. Pelo tipo de configuração da máquina definiu-se que o arranjo dos componentes e a alocação dos alimentadores são os problemas que mais impactam no tempo total do ciclo de montagem da máquina. Como solução de otimização é proposto um algoritmo genético que otimiza as duas decisões de forma simultânea. Além da localização dos componentes e arranjo dos alimentadores foi necessário incluir outra heurística para tratar o balanceamento entre os atuadores e as decisões de *pick-and-place*.

(LI et al., 2008) desenvolvem um método de otimização de máquinas de montagem de componentes multi-cabeças, levando em consideração que a alocação dos alimentadores e a sequência de *pick-and-place* são os fatores que diretamente afetam no tempo de ciclo de montagem, sendo desta forma inter relacionados. Para resolver estes problemas combinatoriais, um algoritmo genético com os operadores de mutação e cruzamento modificados foi aplicado mostrando resultados satisfatórios.

(HO; JI, 2009) descrevem o funcionamento de uma máquina de montagem de componentes SMD de multi-cabeças que devido as suas características têm se tornado o principal tipo empregado nos processos industriais atualmente. Neste trabalho os autores consideram que o problema de otimização é afetado diretamente pela sequência de montagem dos componentes e pelos arranjos dos alimentadores. Levando em consideração estes problemas uma modelagem matemática é desenvolvida de forma integral tomando os pré-requisitos anteriores com o objetivo de reduzir as distâncias de *pick-and-place* da máquina.

(WILHELM et al., 2007) modelam uma máquina de roleta dupla com uma nova abordagem de geração de colunas para encontrar a menor distância de *pick-and-place* da máquina de montagem. Nesta abordagem os tipos de componente que influenciam o tipo de *nozzle* (atuador que gera um vácuo para pegar o componente) são levados em consideração pois este tipo de máquina pode realizar a troca do tipo de *nozzle* durante o ciclo de montagem se necessário porém, esta troca acarreta no aumento do tempo de ciclo.

(JULOSKI et al., 2004) apresentam a modelagem de um processo de montagem de uma máquina do tipo *pick-and-place* avaliando somente os impactos no processo de "*placement*" dos componentes. A heurística determinada é baseada em um modelo de regressão que analisa o sistema dinâmico da cabeça de montagem. Um protótipo foi desenvolvido para atestar os resultados teóricos.

(AYOB; KENDALL, 2005) abordam a otimização de uma máquina de montagem de componentes SMD cujos movimentos livres da placa, do robô e da estante de alimentadores são considerados como três funções objetivo a serem minimizadas. A heurística empregada utiliza a dinâmica de *pick-and-place* de Chebychev para minimizar os movimentos dos eixos X e Y.

(ASHAYERI; SOTIROV, 2011) desenvolvem uma nova abordagem de otimização de uma máquina de montagem *multi-head* que inclui a troca dos *nozzles* como parte do problema no aumento do ciclo de montagem. A heurística é abordada por um problema de número inteiro misto (Mixed Integer Problem - MIP) que considera a quantidade e tipos de componentes a serem montados. Do resultado desta heurística uma nova abordagem de sequenciamento foi desenvolvida para determinar a sequência de montagem final.

Na pesquisa de (CSASZAR et al., 2000) é descrita a otimização de um novo tipo máquina de montagem de componentes SMD modular de alta velocidade, com multi estações e feixe móvel para reconhecimento de pick-up de componentes. A modelagem é baseada em um sistema de conhecimento onde regras e condições são instituídas na tentativa de empregar via software o conhecimento dos especialistas responsáveis por tal operação.

Com o aumento do uso das máquinas de montagem modulares (PARK; KIM, 2010) desenvolveram um método de otimização tratando a heurística como sendo uma problema de programação integral e divide o problema em dois tipos: o problema de arranjo dos alimentadores e o problema da sequência de montagem. Para o problema do arranjo de alimentadores foi desenvolvido um algoritmo de limites e derivações (*branch-and-bound*) e para o problema da sequência de montagem foi desenvolvido um algoritmo de transporte para resolver o problema.

No caso da pesquisa de (YUAN et al., 2007) o foco da otimização está na redução do tempo total de *pickup* e de *placement* dos componentes de uma máquina de montagem de componentes SMD de alta velocidade sendo que o arranjo dos alimentadores é previamente conhecido. Dois algoritmos de pesquisa esparsa são apresentados para resolver o problema e

são comparados com o desempenho de um algoritmo genético aplicado a resolução de problemas relacionados distâncias e seus custos.

Apesar dos métodos aplicados para a otimização de máquinas de montagem de componentes de alta velocidade e precisão considerarem o problema como integral, na realidade, devido as suas características construtivas e à alta inter-relação entre os processos de aquisição e montagem dos componentes, é necessário desenvolver algoritmos que descrevam tais subprocessos, desta forma o resultado obtido se aproxima, mas não apresenta na maioria das vezes a melhor solução. No entanto diversas pesquisas como a aplicação de otimização de multicritério tem despertado grande interesse nas duas últimas décadas principalmente no âmbito industrial. Na pesquisa do estado da arte realizada por (ZHOU et al., 2011) os tipos de algoritmo evolucionário multiobjetivo (MOEA) mais recentes são abordados. Os mecanismos de funcionamento dos operadores de seleção, cruzamento e mutação são ilustrados para os diferentes MOEA's. Os algoritmos abordados são classificados em: coevolucionários, meméticos, de decomposição, manipulação de restrições, dinâmicos, ruidosos, combinatórios e discretos.

Como a solução dos problemas de otimização muitas vezes não são lineares o que dificulta a aplicação de uma solução ótima ou ligada a um limite de maneira determinística faz com que a busca por regiões de solução otimizada ou a utilização de uma heurística mais ampla seja necessária. Os problemas *NP-hard* são apresentados na pesquisa de (BOUSSAÏD et al., 2013) onde o estado da arte relacionado com os problemas de otimização foram classificados em contínuos ou discretos, restritos ou irrestritos, critério único ou multicritério, estático ou dinâmico. Para problemas de critério único as técnicas mais utilizadas são: recozimento simulado (*simulated annealing*), pesquisa Tabu (*tabu search*), algoritmo ganancioso (*greedy algorithm*), pesquisa em vizinhança variável (*variable neighborhood search*), pesquisa local assistida (*guided local search*) e pesquisa local iterativa (*iterated local search*). Para soluções multicritério temos a metaheurística baseada nas populações, conhecidos como algoritmos evolucionários, onde as técnicas mais aplicadas são: algoritmos genéticos, algoritmos de estimação de distribuição, evolução diferencial, algoritmos coevolucionários, algoritmos culturais e algoritmos de busca esparsa e conexão de rota. A inteligência de partículas ou comumente conhecida como metaheurística de colônia também são aplicadas a diversos problemas sejam eles mono ou multicritério. As técnicas mais conhecidas são: otimização baseada em colônia de formigas, otimização baseada em enxame de partículas, otimização baseada em colônia de abelhas, otimização por sistemas imunes

artificiais, otimização baseada em cultura de bactérias e algoritmo de otimização baseado na biogeografia.

## CAPÍTULO III

### METODOLOGIA APLICADA AO ESTUDO

Neste capítulo será abordado o tema máquina de montagem modelo GXH-3 do fabricante Hitachi. Uma descrição relevante de suas características é abordada e uma introdução sobre a teoria matemática básica sobre problemas de combinação, cuja classe de problema descreve o seu processo de montagem, é realizada como forma de embasar a formulação matemática da máquina de montagem GXH-3 que será utilizada na otimização multicritério.

#### 3.1 A Máquina de Montagem Modelo GXH-3

(FUKUSHIMA et al., 2008), descrevem em detalhes as características da máquina de montagem modular GXH-3 do fabricante Hitachi. Como desenvolvedores do equipamento explicam em detalhes as características que tornam esta máquina atrativa para os fabricantes de produtos eletrônicos interessados em alto volume e alta densidade na montagem de componentes em superfície (SMD). Este equipamento pode ser descrito como uma máquina de montagem modular, de entrega dupla cujas cabeças de montagem movem-se nos eixos X e Y, cujos atuadores pneumáticos (responsáveis por "pegar" os componentes e posicioná-los durante a montagem) estão dispostos em círculo conhecido como tambor ou barril (turret). A máquina possui dois módulos, onde por módulo duas cabeças realizam a montagem dos componentes.

A interação entre as cabeças de montagem ocorre de forma síncrona ao pegar o componente nos alimentadores e ao montá-los na posição requerida, ou seja, enquanto uma cabeça monta o componente a outra está pegando os componentes para montá-los no próximo ciclo, isto ocorre para que não haja colisão entre as cabeças. Este ciclo é repetido até que os componentes da lista de montagem deste módulo sejam finalizados.

As características deste modelo adaptam-se muito bem para o cenário onde a transição entre grandes volumes e baixas densidades para baixos volumes e grande diversificação de densidades ocorrem para atender uma gama alta de produtos. Atualmente a adaptação dos processos a uma constante mudança de produtos devido à redução do ciclo de vida dos modelos também é uma característica a que as máquinas de montagem devem atender. Outra realidade onde as máquinas de montagem desempenham um papel importante está no fato da

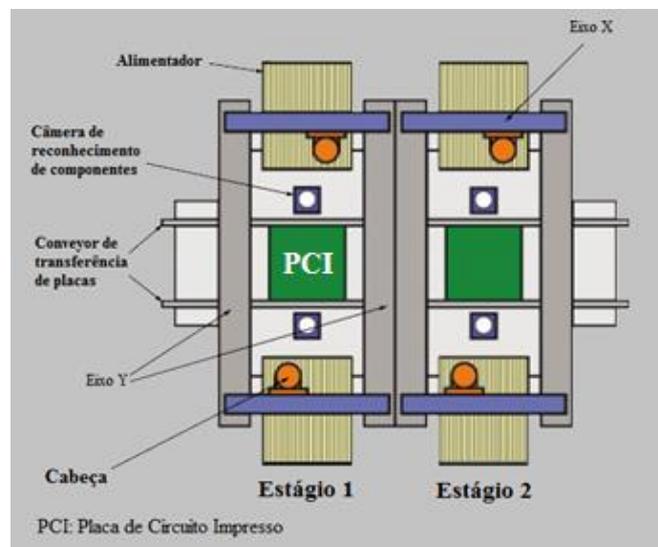
redução dos estoques de matéria prima para atender a demanda com reduzida margem de erro, exigindo assim uma melhor acuracidade na montagem de componentes e consequentemente menor desperdício dos mesmos.

### 3.1.1 Características do Modelo GXH-3

A vantagem principal da máquina de montagem modular é que a mesma foi implementada em uma única plataforma, que pode ser configurada para um grande range de aplicações durante a montagem dos diferentes tipos de componente. Devido a este range esta máquina é capaz de manipular uma grande variedade de componentes, e devido a esta adaptabilidade o processo de otimização também pode ser otimizado para atender diferentes modos de produção que variam entre um grande volume à baixos volumes com diferentes densidades de montagem.

A figura 3.1 mostra uma representação estrutural do modelo GXH-3, cujas principais características são listadas como:

Figura 3.1: Configuração da máquina de montagem GXH-3.



FONTE: Fukushima et al., 2008.

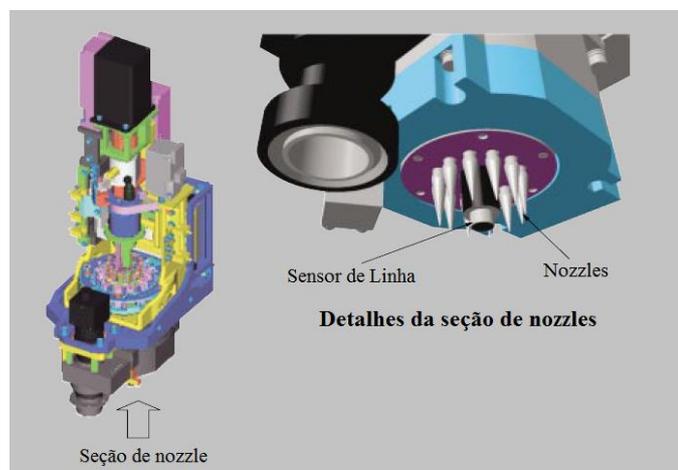
- As cabeças de montagem são movimentadas por motores de acionamento direto que cuja velocidade padrão de  $300\text{ mm/s}$  confere rapidez e acuracidade na montagem dos componentes.

- Alta confiabilidade na montagem devido aos sensores montados nas cabeças que identificam a presença e a espessura dos componentes durante o *pickup* e a distância da superfície da placa durante a montagem, garantindo uma montagem eficiente.
- Montagem com alto grau de acuracidade devido aos motores montados nos eixos X e Y.
- Alta velocidade e acuracidade de reconhecimento são atingidas devido ao sistema de reconhecimento de imagens de grupo em movimento denominado "gang fly" onde é possível realizar o reconhecimento do número máximo de 12 componentes ou o reconhecimento individual dos mesmos. Após o reconhecimento de cada componente, se houver qualquer deslocamento, uma correção de X, Y ou do ângulo  $\theta$  é realizada durante o processo de montagem do componente deslocado.
- Aumento da acuracidade dos alimentadores é devido à aplicação de um servo motor que melhoram a velocidade do incremento da fita durante o processo de *pickup*.
- Software de otimização capaz de equalizar o tempo de montagem dos estágios e reduzir o movimento dos motores para reduzir o tempo de fabricação.

### 3.1.2 Cabeça de Acionamento Direto

A cabeça de montagem de acionamento direto que é descrita na figura adota uma configuração de cabeça rotatória acionada por um servo motor CA e pode ser equipada com até 12 nozzle.

Figura 3.2: Esquema da cabeça de acionamento direto.



FONTE: Fukushima et al., 2008.

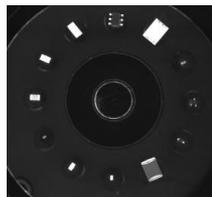
Os componentes são posicionados pela cabeça giratória até atingirem as suas coordenadas de montagem onde são montados pelos *nozzles*. Um sistema de controle quase em tempo real foi desenvolvido para melhorar a desempenho durante os ciclos de pegar e montar. Este sistema consegue uma resposta de 1/1.000 de segundos cujo objetivo é o de controlar de forma mais precisa o movimento de rotação da cabeça e o de entrada e saída das válvulas pneumáticas dos *nozzles*. Como uma forma de reduzir o tempo de fabricação, o impacto dos movimentos de subida e descida dos *nozzles* foi considerado bem como o movimento de rotação da cabeça no intuito de sobrepor os movimentos e reduzir o tempo de montagem.

O controle da válvula de vácuo durante o *pickup* é extremamente importante para atingir o manuseio correto dos componentes a altas velocidades. Para melhorar este controle as válvulas foram movidas para próximo dos *nozzles* que são localizados no rotor do servo motor.

### 3.1.3 Reconhecimento de Imagens em Movimento (*Gang Fly Recognition*)

As imagens são capturadas pela iluminação estroboscópica de uma lâmpada de xenon sem que a cabeça de montagem necessite ficar estática, isto permite o reconhecimento em conjunto dos 12 componentes (uma imagem típica é mostrada na figura 3.3). O processamento da imagem leva em torno de 5ms por chip quadrado (medida utilizada para medir a área de componentes SMD), resultando em um sistema com tempo de atraso 0 durante a montagem dos componentes. Além do reconhecimento de imagem dos componentes outras cameras posicionadas nas cabeças realizam a leitura de padrões de marcas nas PCI's o que acarreta em um sistema de correção duplo, ou seja, o "*gang fly*" corrige o deslocamento do componente e o PEC (reconhecimento de fiducial) corrige o deslocamento da PCI, este processo confere a GXH-3 uma acuracidade de  $\pm 0,005\text{ mm}$  durante o processo de montagem em alta velocidade.

Figura 3.3: Foto tirada para reconhecimento da condição de *pickup* dos componentes.



FONTE: Fukushima et al., 2008.

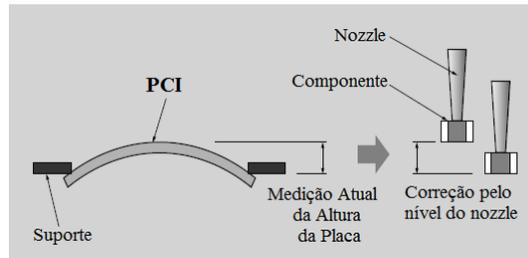
### 3.1.4 Alta Densidade de Montagem de Componentes Pequenos

Atualmente os componentes de tamanho 1005 e 0201 são muito pequenos e sua resistência ao impacto é limitada, e gera dois problemas relacionados a qualidade na montagem que seriam o defeito de montagem (deslocamentos) e a rachadura dos componentes. Quando o deslocamento é atribuído às máquinas de montagem, o problema geralmente ocorre pela insuficiência da força aplicada ao componente durante o processo de montagem. Por outro lado, as rachaduras ocorrem quando muita força é aplicada ao componente durante a montagem. Ou seja, pouca força causa deslocamento e muita força causa rachadura. Para que tais problemas sejam evitados é essencial que haja um controle eficiente durante o *pickup* e um controle da altura da placa durante a montagem. O problema do ajuste de força deve levar em consideração diversos fatores como a dimensão do componente que pode variar de lote para lote ou de fornecedor para fornecedor mesmo sendo o mesmo componente e até a diferença de altura devido ao empenamento da placa. Para resolver este tipo de problema o modelo GXH-3 foi implementado com um sistema de sensores de medição de altura capaz de realizar a leitura da altura dos componentes e da placa. Como resultado, a altura do componente é lida depois do *pickup*, a altura da placa é lida antes do processo de montagem e tais resultados são utilizados para corrigir a altura durante o processo de montagem. Além dos ajustes de altura a força de impacto é restrita durante o *pickup* e durante a montagem.

### 3.1.5 Retorno da Altura da Placa

Assim como explicado no tópico anterior a cabeça de montagem é equipada com um sensor a laser que realiza a leitura da altura da placa antes da montagem do componente com o objetivo de controlar o limite do movimento de descida do *nozzle* e assegurar a montagem apropriada do componente. A figura 3.4 apresenta como este processo é realizado, o sensor realiza a leitura atual da altura da placa e corrige a altura do *nozzle* para evitar a colisão com a PCI.

Figura 3.4: Ajuste de acordo com a altura da placa devido ao empenamento.

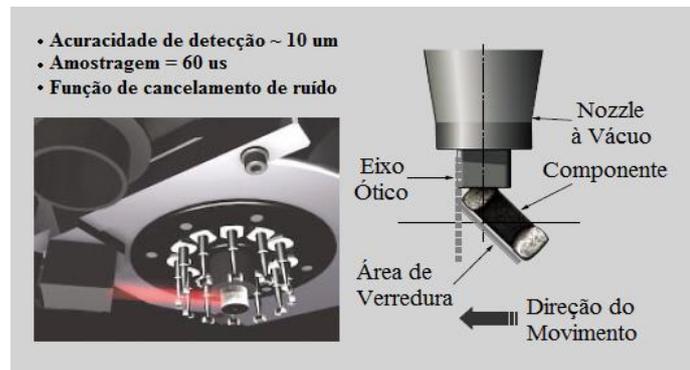


FONTE: Fukushima et al., 2008.

### 3.1.6 Leitura da Altura da Parte Inferior do Componente

Como abordado anteriormente, um dos controles realizados da altura de montagem é realizado por um sensor também montado na cabeça de montagem cuja função de medir a altura do componente é empregada. Este sensor além de medir a altura verifica se o componente foi pego da maneira correta, se sim o procedimento de montagem pode prosseguir, se não o componente é descartado e outro *pickup* do mesmo componente é realizado. A figura 3.5 apresenta como este processo é realizado. Antes da montagem o sensor realiza a leitura do componente e corrige a altura de montagem.

Figura 3.5: Ajuste da altura da superfície inferior do componente.



FONTE: Fukushima et al., 2008.

### 3.1.7 Contribuições do Software para Melhoria da Produtividade

Os ganhos em produtividade não são obtidos somente com as melhorias de hardware, mas também o software desempenha um importante papel. Como uma forma de auxiliar o processo produtivo com o menor tempo de configuração possível, ferramentas como o

software conversor de CAD, biblioteca de instruções offline e um sistema de instrução de coordenadas de montagem offline ajudam a desenvolver a programação para a introdução de novos produtos e fazer simulações do processo produtivo. A melhoria da produtividade também é obtida através do balanceamento da linha e do software de otimização, o que ajuda também na redução do tempo de setup.

Todas as condições possíveis devem ser levadas em consideração no desenvolvimento do software de otimização, como por exemplo, os vários tipos e tamanhos de componentes, os nozzles a serem usados, a interferência entre duas cabeças de um mesmo módulo e assim por diante. Todos estes fatores são necessários para determinar a atribuição otimizada dos nozzles para realizar o *pickup*, a atribuição de parte dos alimentadores, a sequência de *pickup* e de montagem do componente, a posição da placa no eixo Y e a sequência de produção.

Ao analisar somente a sequência de montagem, um grupo fatorial de possibilidades de montagem surge o que dificulta a criação de uma solução aceitável. Para o caso do otimizador empregado na GXH-3 uma metaheurística foi desenvolvida e empregou três métodos comumente aplicados em problemas de atribuição que são: o método da aproximação, o método do algoritmo ganancioso (greedy algorithm) e o método de pesquisa local.

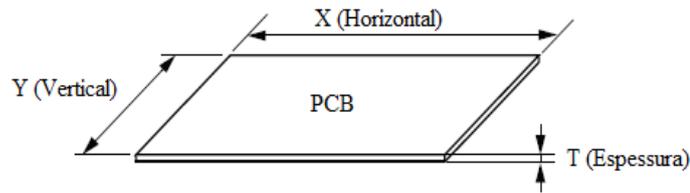
### 3.2 Especificações do Modelo GXH-3

A máquina de montagem GXH-3 tem suas limitações que são: dimensões mínimas e máximas da PCI a ser montada, número máximo de tipos de componente, número máximo de componentes por *pickup*, capacidade máxima de montagem de componente por hora, tamanho mínimo e máximo de componente a ser montado.

Estas especificações são descritas nos tópicos a seguir:

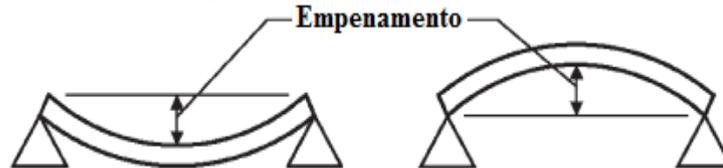
- Dimensões da PCI: As placas de circuito impresso podem variar em comprimento de 50 mm à 460 mm, em largura de 50 mm à 460 mm e em espessura de 0.5 mm à 5 mm. Além das dimensões físicas, existe a questão do empenamento, ou seja, a máxima altura, seja ela positiva ou negativa, que a placa pode ser deformada sem afetar a performance da montagem. Para máquina Hitachi este valor pode ser de até 0.2 mm a cada 50 mm de comprimento que a PCI possua. Em outras palavras, se o comprimento da PCI for de 150 mm o empenamento máximo aceitável é de 0.6 mm. Devido a este fato, o peso máximo admitido da placa após todos os componentes serem montados é o de 1,5 kg.

Figura 3.6: Decomposição horizontal e vertical.



FONTE: Hitachi, 2007.

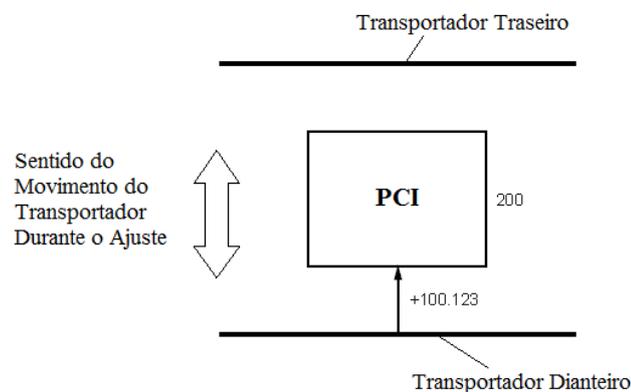
Figura 3.7: Face do empenamento.



FONTE: Hitachi, 2007.

- Capacidade de montagem: A capacidade de montagem de componentes em uma máquina de montagem SMD é medida pelo CPH (chips/componentes por hora). No caso da GXH-3 é definida em 95.000 CPH, desprezando-se o tempo de transição da PCI entre os módulos de entrada e de saída em condições ótimas.
- Posicionamento da posição do eixo Y da PCI: esta função é capaz de determinar qual é o melhor posicionamento da placa no eixo Y dependendo do programa padrão desenvolvido. Existem 4 modos de operação que são: base fixa, base livre, incrementos de 10mm ou incrementos de 20 mm. Cada modo de operação é utilizado para condições específicas. A figura mostra como o posicionamento ocorre.

Figura 3.8: Conveyor de transporte de PCI dianteiro e traseiro.

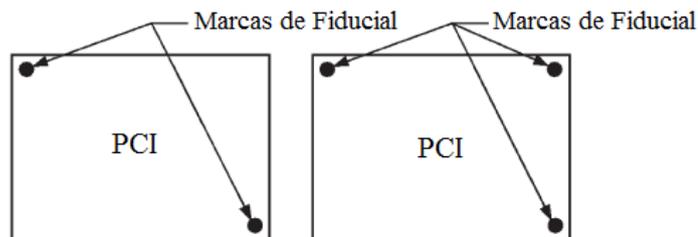


FONTE: Hitachi, 2007.

- Tempo de transição de PCI: É considerado aproximadamente 2.5 segundos em condições ótimas e para placas cujo comprimento seja menor que 155 mm.

- Direção de fluxo de placa e referência de transferência: A PCI segue o fluxo padrão da esquerda para direita e sua referência pode ser tomada pela parte frontal ou traseira da máquina de acordo com o requisito da programação.
- Método de correção da localização das posições de montagem: O reconhecimento de fiducial é responsável pelo ajuste de coordenadas de acordo com o posicionamento da placa no suporte antes que seja iniciado o processo de montagem. A marca de fiducial tem um formato que obedece a um padrão específico e está alocada na placa de circuito impresso (verificar a tabela 1 do Anexo I para obter maiores informações). Ao início do primeiro ciclo de montagem de cada cabeça o reconhecimento do fiducial é realizado. Este procedimento é realizado devido às coordenadas de montagem são referenciadas ao ponto de origem da placa e qualquer desvio de posição durante o posicionamento da placa pode fazer com que o processo de montagem seja realizado de maneira errada, ou seja, de acordo com a leitura de fiducial realizada o erro gerado serve para incrementar ou decrementar o valor de todas as coordenadas nos eixos X e Y. O processo de reconhecimento de fiducial é feito através do reconhecimento de dois ou três pontos no modelo GXH-3, porém o reconhecimento de dois pontos seja o mais utilizado conforme a figura 2.13 exemplifica.

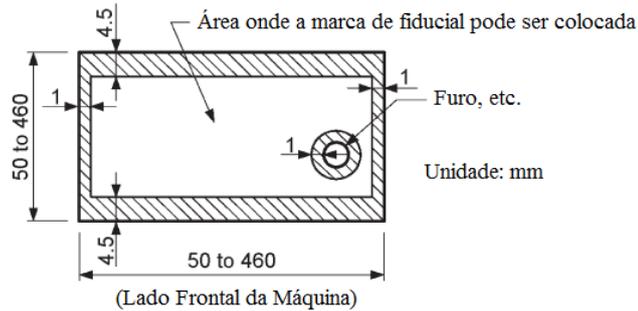
Figura 3.9: Reconhecimento de fiducial.



FONTE: Hitachi, 2007.

Devido ao fato de limitações quanto à leitura das marcas de fiducial pela câmera de reconhecimento (PEC), a área onde estas marcações podem ser colocadas na placa também é limitada conforme a figura 2.14.

Figura 3.10: Limites para a área da marca de fiducial.



FONTE: Hitachi, 2007.

- Componentes aplicáveis: Devido ao formato dos componentes e dos tipos limitados de *nozzle* disponíveis existem limitações quanto à dimensão e formato dos mesmos. A tabela 3.1 mostra a relação do tamanho do componente com o seu respectivo tipo de embalagem.

Tabela 3.1: Relação dos componentes e suas embalagens  
Especificação dos Componentes e das Embalagens

Dimensão dos Componentes	Valor da Medida (mm)	Tipo de Embalagem	Valor da Medida (mm)
Tamanho	$0,6 \times 0,3$ à $44 \times 44$	Componentes em fita	largura: 8 ~72 diâmetro externo do rolo: $\leq 382$
Altura	$\leq 12,7$	Componentes em vareta	largura: 8~60 comprimento: 400~600 altura: 3~16
Distância entre terminais	$\geq 0,4$	Componentes em bandeja (FP-G100/FP-G300)	$100 \times 100$ à $323 \times 136$
Largura dos Terminais	$\geq 0,15$	Componentes em bandeja (FP-G200)	$100 \times 100$ à $330 \times 230$
Comprimento dos terminais	$\geq 0,20$	-	-

FONTE: Hitachi, 2007.

- Cabeças de montagem e *nozzles*: O modelo GXH-3 está equipado com quatro cabeças de montagem montadas em quatro eixos em Y cujos movimentos são independentes uma da outra. O número de *nozzles* varia entre um até doze *nozzles* dependendo do tamanho dos componentes a serem montados. A tabela 3.2 a seguir apresenta uma relação entre a quantidade de *nozzles* máxima pelo tipo de componente.

Tabela 3.2: Quantidade máxima de nozzles por tipo de componente

Tamanho do Componente (mm)	Numero Máximo de
$1,0 \times 0,5 \leq C \times L \leq 5,0 \times 4,0$	12
$5,0 \times 4,0 \leq C \times L \leq 10,0 \times 10,0$	6
$10,0 \times 10,0 \leq C \times L \leq 12,0 \times 12,0$	4
$12,0 \times 12,0 \leq C \times L \leq 20,0 \times 20,0$	2
$20,0 \times 20,0 \leq C \times L \leq 44,0 \times 44,0$	1

Legenda:  $C \times L$  = comprimento  $\times$  largura

FONTE: Hitachi, 2007.

- Alimentadores: A máquina de montagem GXH-3 possui quatro bases (uma base por cabeça de montagem) onde os alimentadores podem ser alocados. Os alimentadores são mecanismos eletromecânicos onde os componentes em fita, em vareta ou em bandeja são posicionados. Durante o seu funcionamento, os alimentadores movimentam os carretéis de componentes em incrementos pré-determinados, cujo valor é definido pelo tamanho do componente, que sempre posicionam os componentes na coordenada de *pick* definida pela sua posição. A tabela 3.3 contém os valores máximos de alimentadores por tamanho de componentes.

Tabela 3.3: Número máximo de alimentadores por tamanho de componente

Tipo de Alimentador	Tamanho do Componente (mm)	Quant. de <i>feeders</i> / Quant. de tipos de Componentes
Rolo Duplo de 8 mm (tipo fita)	$1,0 \times 0,5 \leq C \times L \leq 5,0 \times 4,0$	50/100
Rolo Único de 12/16 mm (tipo fita)	$5,0 \times 4,0 \leq C \times L \leq 10,0 \times 10,0$	50/50
Rolo Único de 24/32 mm (tipo fita)	$10,0 \times 10,0 \leq C \times L \leq 12,0 \times 12,0$	24/24
Rolo Único de 44/56 mm (tipo fita)	$12,0 \times 12,0 \leq C \times L \leq 20,0 \times 20,0$	16/16
Rolo Único de 72 mm (tipo fita)	$20,0 \times 20,0 \leq C \times L \leq 44,0 \times 44,0$	12/12

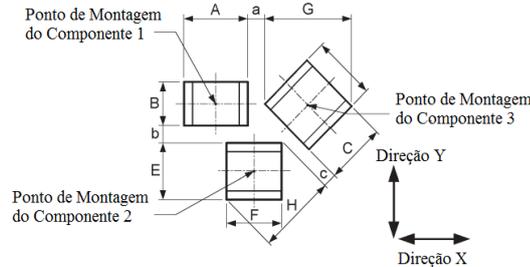
Legenda:  $C \times L$  = comprimento  $\times$  largura.

FONTE: Hitachi, 2007.

- Tipos de *nozzle* para montagem correta dos componentes: Quando os componentes são montados próximo de outros componentes montados anteriormente ou próximo a obstáculos, os formatos dos *nozzles* se tornam parte das condições que restringem a montagem. Para uma lista completa verifique a tabela de tipos de *nozzles* na seção de anexo.

- Distância adjacente mínima entre componentes: Quando a coordenada de montagem dos componentes é levada em consideração, uma distância mínima de 0,2mm entre os componentes deve ser respeitada. Como forma ilustrativa a figura exemplifica um esquema de montagem de três componentes.

Figura 3.11: Exemplo 1: Montagem de 3 componentes adjacentes.



FONTE: Hitachi, 2007.

Para realizar o cálculo da distância mínima entre os componentes é necessário que os dados dos tamanhos dos componentes sejam conhecidos. Tendo como base o exemplo da figura para os eixos X e Y o cálculo segue uma determinada formulação. Para o cálculo do eixo X utilizam-se os componentes 1 e 3:  $X = \frac{A+G}{2} + 0,2$  (unidade em mm). Para o cálculo do eixo Y utilizam-se os componentes 1 e 2:  $Y = \frac{G+E}{2} + 0,2$  (unidade em mm).

### 3.3 O Problema de Atribuição Quadrática

Embora os problemas de programação linear sejam muito comuns e representam uma grande maioria de problemas reais que podem ser representados matematicamente por esta técnica, muitas vezes tais problemas apresentam funções ou restrições não lineares. Para tais condições a programação é descrita como problema de programação não linear.

O problema de atribuição quadrática (*Quadratic Attribution Problem* - QAP) é uma generalização do problema de atribuição (*Attribution Problem* - AP), porém o QAP apresenta uma função objetivo não linear.

A aplicação do problema de atribuição descreve parte do funcionamento de uma máquina de inserção de componentes composta por alimentadores, componentes e posições de montagem.

Desta forma para as seguintes variáveis são utilizadas para descrever o problema de atribuição quadrática temos:

$n$  = número de elementos de uma variável qualquer;

$k$  = alimentadores do primeiro módulo da máquina de inserção;

$l$  = alimentadores do segundo módulo da máquina de inserção;

$r$  = posições de alocação dos alimentadores do primeiro módulo;

$s$  = posições de alocação dos alimentadores do segundo módulo;

$\alpha$  = relação entre cada par de alimentadores do primeiro e segundo módulo;

$\beta$  = relação entre cada par de alocações de alimentadores entre o primeiro e segundo módulo;

$\tau$  = custo de associar um alimentador à uma posição do primeiro ou segundo módulo;

$\gamma$  = variável binária, ou seja, a variável assume o valor 1 se a posição  $k$  ou  $l$  for atribuída para a posição  $r$  ou  $s$ , 0 se não.

Considera-se que os grupos de alimentadores ( $k, l = 1, 2, \dots, n$ ) são atribuídos a um grupo único de locações ( $r, s = 1, 2, \dots, n$ ). A relação entre cada par de alimentadores pode ser definida como  $\alpha_{kl}$  e a relação entre cada par de locações como sendo  $\beta_{rs}$ . Um custo  $\tau_{kr}$  também é associado com a alocação do alimentador  $k$  na posição  $r$  então o problema de atribuição quadrática pode ser formulado como:

$$\min z = \sum_{k=1}^n \sum_{l=1}^n \sum_{r=1}^n \sum_{s=1}^n \alpha_{kl} \beta_{rs} \gamma_{kr} \gamma_{ls} + \sum_{k=1}^n \sum_{r=1}^n \tau_{kr} \gamma_{kr} + \sum_{l=1}^n \sum_{s=1}^n \tau_{ls} \gamma_{ls} \quad 3.1$$

sujeito à:

$$\sum_{k=1}^n \gamma_{kr} = 1 \quad r = 1, 2, \dots, n \quad 3.2$$

$$\sum_{r=1}^n \gamma_{kr} = 1 \quad k = 1, 2, \dots, n \quad 3.3$$

$$\sum_{l=1}^n \gamma_{ls} = 1 \quad s = 1, 2, \dots, n \quad 3.4$$

$$\sum_{s=1}^n \gamma_{ls} = 1 \quad l = 1, 2, \dots, n \quad 3.5$$

$$\gamma_{kr} \in \{0,1\}$$

$$\gamma_{ls} \in \{0,1\}$$

A variável de decisão  $\gamma_{kr}$  representa a atribuição do alimentador  $k$  na posição  $r$ . A variável de decisão  $\gamma_{ls}$  representa a atribuição do alimentador  $l$  na posição  $s$ . A função objetivo trata a alocação dos alimentadores de tal forma que a distância percorrida durante a operação de aquisição de materiais seja mínima. As restrições 3.2 e 3.4 determinam que a cada alocação deve-se conter apenas um alimentador e as restrições 3.3 e 3.5 garantem que cada alimentador deve ser designado para apenas uma alocação.

### 3.4 O Problema do Caixeiro Viajante

O problema do caixeiro viajante (*Traveller Salesman Problem* - TSP) tem sido um dos problemas mais abordados quando se trata de problemas relacionados a distâncias entre localizações. O TSP pode ser facilmente descrito como: um vendedor deve visitar  $n$  cidades diferentes e então deve retornar para a cidade inicial (casa). O vendedor quer determinar qual rota deve ser seguida para que todas as cidades em seu roteiro sejam visitadas uma vez apenas de modo que a distância total percorrida seja a mínima possível. Embora o conceito seja simples de entender a solução ótima é difícil de ser encontrada devido ao fato de o problema ser de natureza combinatorial, para  $n$  cidades a serem visitadas existem  $n!$  possibilidades de rota que devem ser avaliadas no espaço de busca.

De forma análoga, o problema do caixeiro viajante pode ser comparado ao problema de rota de montagem de componentes SMD, que representa parte do problema de otimização de uma máquina de montagem. Ao introduzir a variável  $x_{ij}$  que representam a rota da cabeça de montagem entre as posições de componentes  $i$  e  $j$ , um dos problemas mais comuns de formulação de problemas de números inteiros pode ser formulado de acordo com as seguintes variáveis:

$n$  = número de posições de montagem;

$u$  = número inteiro associado a posição de montagem conforme a ordem crescente das coordenadas de montagem a serem seguidas;

$i$  = posição de montagem da primeira posição da rota de montagem;

$j$  = posição de montagem da última posição da rota de montagem;

$d_{ij}$  = distância entre as posições de montagem  $i$  e  $j$ ;

$x_{ij}$  = variável binária, ou seja, ela assume o valor 1 se a distância entre  $i$  e  $j$  for a menor, 0 se não.

Assim sendo o problema do caixeiro viajante aplicado a modelagem matemática de uma máquina SMD pode ser descrito como:

$$\text{minimizar } a = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n d_{ij} x_{ij} \quad 3.6$$

sujeito à,

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \text{ para } i \neq j \quad 3.7$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \text{ para } i \neq j \quad 3.8$$

$$u_i - u_j + n x_{ij} \leq n - 1 \quad i, j = 1, 2, \dots, n \text{ para } i \neq j \quad 3.9$$

$$x_{ij} \in \{0, 1\}$$

$u_i \geq 0$  e é composto por números inteiros.

A variável  $d_{ij}$  representa a distância entre os componentes  $i$  e  $j$ . A função objetivo representa a distância mínima percorrida em uma rota. A restrição 3.7 garante que a cabeça de montagem chegue apenas uma vez em cada coordenada de montagem. A restrição 3.8 define que a cabeça de montagem deixe cada coordenada de montagem apenas uma vez. A restrição 3.9 evita a presença de sub-rotas.

Para muitos autores que tratam o problema do caixeiro viajante a expressão TSP euclidiano é bastante empregada devido ao fato da equação que descreve a distância entre duas cidades ser definida pela equação euclidiana da distância, na qual a matriz de distâncias  $A$  é simétrica e obedece a equação de igualdade  $d_{ij} = d_{ji}$  para todo  $i, j$  e satisfaz a desigualdade triangular  $d_{jk} \leq d_{ij} + d_{jk}$  para todo  $i, j, k$  distinto. Porém como descrito por (DEZA; DEZA, 2006) as representações das distâncias não dependerão somente do comprimento do arco que determina a distância entre dois pontos como também da rota que deve ser empregada para alcançar o destino.

### 3.5 Integrações do Problema de Atribuição e do Problema do Caixeiro Viajante

Existem problemas que mistos que envolvem os problemas de atribuição e o de sequenciamento, nestes casos o TSP e o QAP devem ser combinados para encontrar a solução destes tipos de problemas. Por exemplo, para modelar integralmente o problema de montagem de uma máquina SMD considera-se que um grupo de componentes ( $c = 1, 2, \dots, m$ ) deve ser montado em determinadas posições ( $i, j = 1, 2, \dots, n$ ). Cada tipo de componente  $m$  deve ser armazenado em um alimentador  $l$ , porém existe um número limitado que um carrinho de alimentadores pode armazenar. Assim como apenas um tipo de componente  $c$  pode ser armazenado em um alimentador,  $n$  alimentadores são necessários para armazenar  $n$  tipos de componentes.

Assim sendo as seguintes variáveis são utilizadas para descrever o problema:

$n$  = número de elementos de uma variável;

$c$  = tipo de componente;

$l$  = número de posições de alimentadores;

$i$  = coordenada de montagem do primeiro componente da rota de montagem;

$j$  = coordenada de montagem do último componente da rota de montagem;

$d_{il}$  = distância entre a coordenada de montagem do componente  $i$  e coordenada da posição do seu respectivo alimentador;

$d_{lj}$  = distância entre a coordenada de montagem do componente  $j$  e coordenada da posição do seu respectivo alimentador;

$\gamma_{cl}$  = variável binária, ou seja, a variável assume o valor 1 se o tipo de componente  $c$  da posição de montagem  $j$  é atribuída ao alimentador  $l$ , 0 se não;

$x_{ij}$  = variável binária, ou seja, ela assume o valor 1 se a distância entre  $i$  e  $j$  for a menor, 0

se não;

$d_{i0}$  = distância entre a posição inicial e a primeira posição de *pickup*;

$x_{i0}$  = variável binária que assume o valor 1 se a distância inicial é considerada durante o ciclo de montagem, 0 se não;

Algumas considerações devem ser feitas para descrever o problema:

1. No primeiro ciclo, a cabeça de montagem sai da posição de origem e vai até a posição para realizar a aquisição do componente (*pickup*);
2. Após adquirir o componente a cabeça de montagem move-se para a posição de montagem onde o componente  $c$  deve ser montado na posição  $i$ ;
3. O processo é repetido até que todos os componentes tenham sido montados com a cabeça retornando para a posição original.

Para minimizar a distância total percorrida pela cabeça para pegar e montar todos os componentes é necessário determinar não somente a sequência de montagem ( $x_{ij}$ ) mas também a atribuição dos tipos de componentes para os alimentadores ( $\gamma_{cjl}$ ).

O objetivo do problema é minimizar a distância total percorrida pela cabeça, que inclui a distância entre o ponto inicial e o primeiro alimentador  $d_{0l}$ , a distância entre o alimentador para a primeira posição de montagem  $d_{li}$ , a distância da última posição de montagem para o próximo alimentador  $d_{jl}$  e a distância entre a última posição de montagem para a posição inicial  $d_{i0}$ . É possível observar que a posição inicial da cabeça pode ser considerada como uma posição de montagem ( $i = j = 0$ ). Deste modo é possível formular o problema como:

$$\text{minimizar } d = \sum_{i=0}^n \sum_{\substack{j=1 \\ j \neq i}}^n \sum_{c=1}^n \sum_{l=1}^n (d_{il} + d_{lj}) x_{ij} \gamma_{cjl} + \sum_{i=1}^n d_{i0} x_{i0} \quad 3.10$$

sujeito à:

$$\sum_{i=0}^n x_{ij} = 1 \quad \text{para } j = 0, 1, \dots, n; i \neq j \quad 3.11$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \text{para } i = 0, 1, \dots, n; i \neq j \quad 3.12$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad \text{para } i, j = 1, 2, \dots, n; i \neq j \quad 3.13$$

$$\sum_{c=1}^n \gamma_{cl} = 1 \quad \text{para } l = 1, 2, \dots, m \quad 3.14$$

$$\sum_{l=1}^n \gamma_{cl} = 1 \quad \text{para } c = 1, 2, \dots, m \quad 3.15$$

$$x_{ij} \text{ e } \gamma_{cl} \in \{0, 1\}$$

$u_i \geq 0$  e é composto por números inteiros.

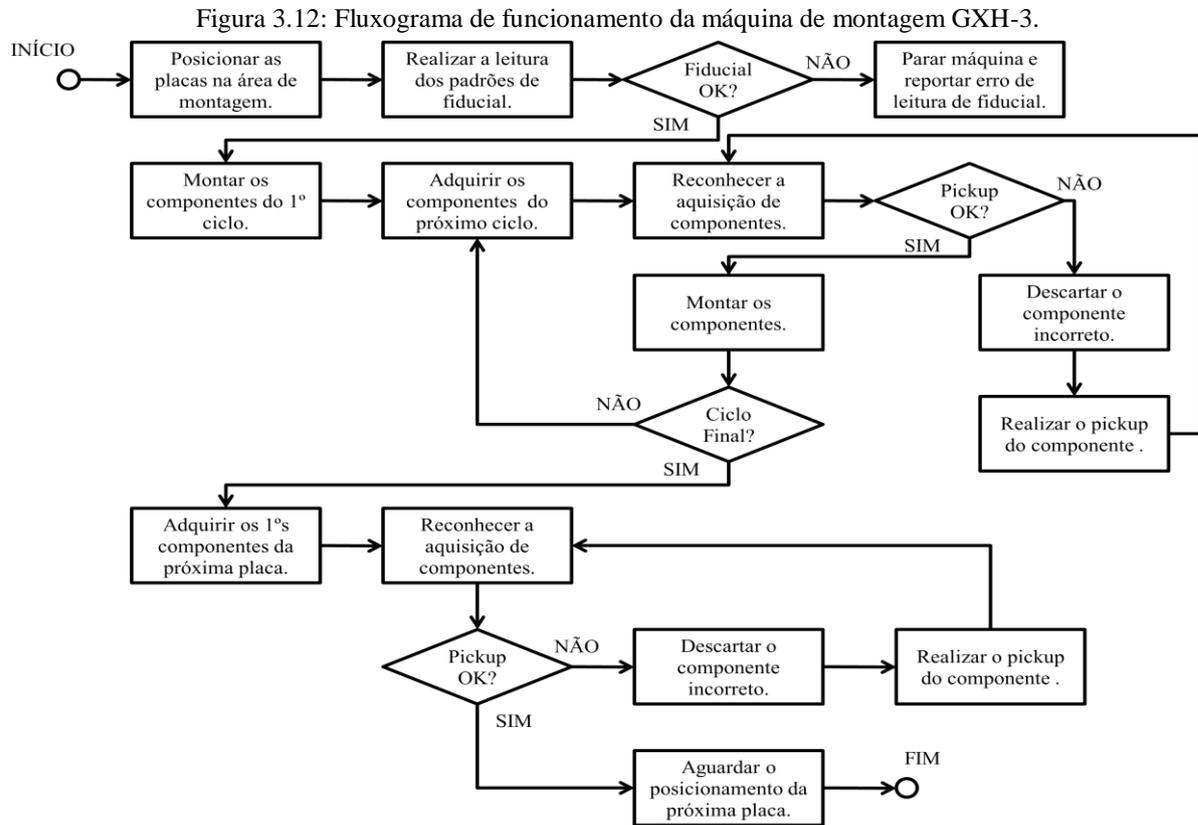
A função objetivo calcula a distância total percorrida pela cabeça de montagem. A restrição 3.11 mostra que uma posição  $i$  deve ser montada antes da posição  $j$ . A restrição 3.12 garante que a posição  $j$  deve ser montada depois da posição  $i$ . A restrição 3.13 elimina as sub-rotas. A restrição 3.14 garante que apenas um tipo de componente  $c$  pode ser armazenado em um alimentador  $l$ . Por fim a restrição 3.15 define que apenas um alimentador  $l$  pode conter um tipo de componente  $c$ .

O modelo torna-se muito complexo pelo fato de um problema de atribuição e de sequência serem considerados ao mesmo tempo, fazendo com que a resolução de ambos não possa ser feita separadamente. A este tipo de problema é dada a classificação de *NP-hard*, ou seja, um problema que não pode ser resolvido em tempo polinomial e sim em um tempo exponencial.

### 3.6 A Modelagem do Problema de Otimização da Máquina GXH-3

Com a teoria matemática introduzida algumas considerações devem ser feitas para descrever o funcionamento da máquina bem como as restrições impostas pelas características construtivas do modelo GXH-3. Antes de inicializar-se a modelagem matemática, é necessário descrever o processo de montagem para que a tradução deste processo para a linguagem

matemática seja realizada de maneira mais próxima da realidade. A seguir um fluxograma descreve como este processo ocorre.



FONTE: Autor, 2015.

O problema de otimização envolve a minimização do tempo de montagem pela atribuição apropriada dos alimentadores, pelo sequenciamento dos processos de aquisição e montagem dos componentes e por fim uma sequencia ótima de montagem dos componentes devido ao *pickup* em grupo capaz de realizar até 12 montagens por ciclo. Ao fazer a consideração de que a velocidade padrão de  $300\text{ mm/s}$  é constante é possível calcularmos o tempo bastando apenas encontrar as distâncias percorridas pela cabeça de montagem durante a sua operação. As distâncias entre todas as coordenadas  $(x, y)$  das posições dos componentes da placa, das posições dos componentes nos alimentadores e das posições dos dispositivos de reconhecimento de aquisição e de deslocamento de placa são calculadas através da distância euclidiana entre dois pontos em um plano. Dado 2 pares de coordenadas,  $(x_i, y_i)$  e  $(x_j, y_j)$  a distância entre estes pontos é dada por  $d_{ij}$  e é calculada como:

$$d_{ij} = \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}. \quad 3.16$$

Relembrando as características da máquina deve-se assumir que:

1. A GXH-3 possui 2 módulos, o de entrada e o de saída;
2. Cada módulo possui 2 cabeças de montagem;
3. Cada cabeça de montagem pode conter no máximo 12 *nozzles*;
4. As cabeças de montagem podem acessar a posição de montagem uma de cada vez para evitar colisões;
5. O número máximo de alimentadores depende das dimensões do componente (máx. 50 alimentadores);
6. O ciclo de aquisição de componentes pela cabeça pode ser no máximo de 12 componentes dependendo do tipo de *nozzle* que é definido pelo tipo de componente;
7. Logo após a aquisição o reconhecimento das imagens dos componentes adquiridos deve ser realizada independente do ciclo;
8. Antes do início de cada ciclo de montagem o tempo do primeiro ciclo de aquisição de componentes não deve ser considerado pois é realizado durante o período de transição da placa do carregador de placas até o sistema de fixação da placa antes do processo de montagem ser iniciado;
9. A coordenada da posição inicial da cabeça de montagem é a mesma da coordenada da câmera de reconhecimento de componentes;
10. Apenas durante o primeiro ciclo de montagem, o reconhecimento das marcas de fiducial é realizada antes de iniciar o processo de montagem. Desta forma a distância entre a posição inicial, a primeira marca de fiducial e a segunda marca de fiducial deve ser considerada nos cálculos de otimização;
11. Devido às restrições de montagem e ao sistema de sensores que medem o empenamento da placa, os componentes devem ser montados na ordem crescente em relação a sua altura;
12. É preferível que os componentes sejam montados em ordem decrescente em relação a sua quantidade, ou seja, os componentes com maior quantidade devem ser montados primeiro.

A cabeça de montagem adquire o componente no alimentador e o monta na posição designada na placa. A aresta (arco) de avanço define-se como o caminho da posição de

aquisição do componente, passando pelo reconhecimento dos componentes ( realizando o reconhecimento de fiducial se for o primeiro ciclo) para a posição designada de montagem na placa e a aresta (arco) de retorno designa-se como sendo o caminho realizado da última posição de montagem do ciclo máximo de 12 componentes para a próxima posição de aquisição de componentes. O ciclo de montagem além de contar com os arcos de avanço e de retorno também conta com o arco de montagem que vai da primeira posição de montagem do final do arco de avanço para a última posição de montagem que é a mesma posição inicial do arco de retorno.

Devido às restrições de montagens quanto à altura e quantidades de componentes é possível notar dois tipos de aquisição de componentes:

1. A aquisição de componentes quando a sequencia de montagem exige uma quantidade maior que 12 componentes deste mesmo tipo, a cabeça de montagem não se move durante a aquisição, pois a rotação dos atuadores (*nozzle*) se encarrega de realizar a aquisição dos componentes desejados. A esta condição denomina-se "aquisição sem movimento" (*no moving gang pick*);
2. A aquisição de componentes quando a quantidade do tipo de componente é inferior a 12 faz com que a cabeça se mova para a próxima posição de aquisição de acordo com a seqüência de montagem. A esta condição, dá-se o nome de "aquisição em movimento" (*moving gang pick*).

Até o presente momento, as soluções empregadas utilizam heurísticas onde a abordagem é feita em etapas para se alcançar um resultado através de métodos exatos como os empregados pelo fabricante da máquina em estudo. O conhecimento aprofundado da máquina, bem como a descrição matemática adquiridos neste capítulo são pré-requisitos importantes para desenvolver novas abordagens sobre o tema. A proposta desta dissertação tem como objetivo abordar este mesmo problema utilizando uma nova ferramenta como forma de resolução e melhoria do funcionamento tendo como resultado um aumento da produtividade da máquina de montagem.

## CAPÍTULO IV

### TÉCNICAS DE OTIMIZAÇÃO MULTIOBJETIVO

Neste capítulo será introduzido o conceito de otimização multiobjetivo, com o objetivo de explorar as técnicas mais empregadas nos diversos tipos de problemas atuais e a forma de modelar a solução multicritério para obter um grupo de soluções que auxiliem o tomador de decisão a adequar as respostas encontradas a seu problema de maneira a maximizar a sua eficiência.

#### 4.1. Como Administrar Vários Objetivos

Quando um projeto de um produto novo é aprovado, várias etapas devem ser cumpridas para que o produto chegue até o consumidor. Muitas destas etapas devem ser cumpridas ao mesmo tempo como por exemplo: contratar fornecedores de componentes e desenvolver uma lista de materiais de montagem deste produto. A otimização dos processos busca reduzir o tempo de execução do projeto gerando oportunidade da empresa obter uma vantagem ao oferecer um produto antes dos seus concorrentes.

O gerenciamento deste novo produto não pode ser modelado ou estipulado por uma única expressão matemática sujeita a várias restrições. Isto se deve ao fato que a otimização busca encontrar um mínimo global ou um máximo global e em todos os casos onde os objetivos divergem o objetivo da otimização, definir uma única função torna-se impossível. Exemplo: Deseja-se minimizar o custo dos componentes a contra deseja-se manter a desempenho maior possível.

De fato algumas técnicas tratam um problema multiobjetivo de forma parcial tratando cada objetivo e suas restrições de modo separado e por último cada solução é comparada, porém existem variáveis que influenciam não somente em um, mas em vários objetivos de modo a reduzir a eficiência das respostas de cada função levando a soluções muitas vezes locais e não globais.

Levando em consideração as variáveis inter-relacionadas a otimização multiobjetivo foi desenvolvida com o objetivo de alcançar um grupo de soluções mais aproximadas ao ótimo global deixando assim o tomador de decisões melhor amparado quanto à eficácia da solução empregada gerando uma estimativa mais próxima da realidade.

## 4.2. Otimização de Vários Objetivos e Soluções Ótimas de Frente de Pareto

Como apresentado por (CARAMIA; DELL'OLMO, 2008) um problema de otimização de objetivo único pode ser formulado conforme a expressão a seguir:

$$\min f(x) \quad 4.1$$

$$x \in S$$

onde  $f$  é uma função escalar e  $S$  é um grupo de restrições que podem ser definidas como  $S = \{x \in R^m: h(x) = 0, g(x) \geq 0\}$ .

A otimização multiobjetivo pode então ser escrita em termos matemáticos da seguinte forma:

$$\min[f_1(x), f_2(x), \dots, f_n(x),] \quad 4.2$$

$$x \in S,$$

onde  $n > 1$  e  $S$  é o grupo de restrições refere-se aos previamente definidos. O espaço denominado como "espaço de objetivos" contém o vetor com os vários objetivos e o espaço alcançado contém a imagem de todo grupo considerado apto pela função de aptidão  $F$ . Tal espaço é definido da seguinte forma:

$$C = \{y \in R^n: y = f(x), x \in S\}.$$

O conceito escalar de "ótimo" não se aplica diretamente a um grupo de otimização multiobjetivo. Ao invés de um número escalar indicando o "ótimo" a solução multiobjetivo introduz um conceito de "Pareto ótimo". O conceito consiste no fato que todo o vetor  $x^* \in S$  é considerado contido no Pareto ótimo de um problema multiobjetivo se e somente se todos os vetores  $x \in S$  tenham um valor maior de pelo menos uma das funções objetivo  $f_i$ , sendo  $i = 1, 2, \dots, n$ , ou tenha o mesmo valor de algumas funções objetivo. Levando este argumento em

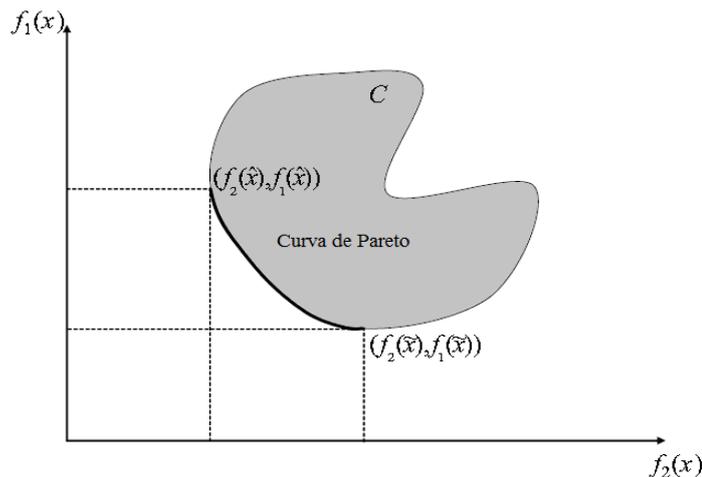
consideração as definições de "Pareto ótimo fraco" e "Pareto ótimo restrito" são definidas a seguir:

- Pareto ótimo fraco ou solução eficiente fraca de multiobjetivo é toda solução  $x^*$  tal que não exista um elemento  $x \in S$  que satisfaça  $f_i(x) < f_i(x^*)$  para todo  $i \in \{1, \dots, n\}$ .
- Pareto ótimo restrito ou solução eficiente restrita de um problema multiobjetivo é toda solução  $x^*$  tal que não exista um elemento  $x \in S$  que satisfaça  $f_i(x) \leq f_i(x^*)$  para todo  $i \in \{1, \dots, n\}$ , com pelo menos uma desigualdade restrita.

De posse das definições sobre as regiões do gráfico de Pareto, os pontos locais de Pareto ótimo podem ser definidos também de acordo com a vizinhança de aptidão do ponto  $x^*$ . Em outros termos, considere que se  $B(x^*, \varepsilon)$  for um círculo ou uma esfera de raio  $\varepsilon > 0$  em torno do ponto  $x^*$ , é necessário que para algum ponto de  $\varepsilon > 0$ , não exista um  $x \in S \cap B(x^*, \varepsilon)$  tal que  $f_i(x) \leq f_i(x^*)$  para todo  $i \in \{1, \dots, n\}$ , com pelo menos uma desigualdade restrita.

A que contém todas as soluções eficientes é conhecida como frente de Pareto, curva de Pareto ou superfície de Pareto. A forma da figura de Pareto indica a natureza das trocas entre as funções objetivo. A figura 4.1 mostra um exemplo de curva de Pareto, onde todos os pontos  $(f_2(\hat{x}), f_1(\hat{x}))$  e  $(f_2(\tilde{x}), f_1(\tilde{x}))$  definem a frente de Pareto. Estes pontos são conhecidos como pontos não dominados ou não inferiores.

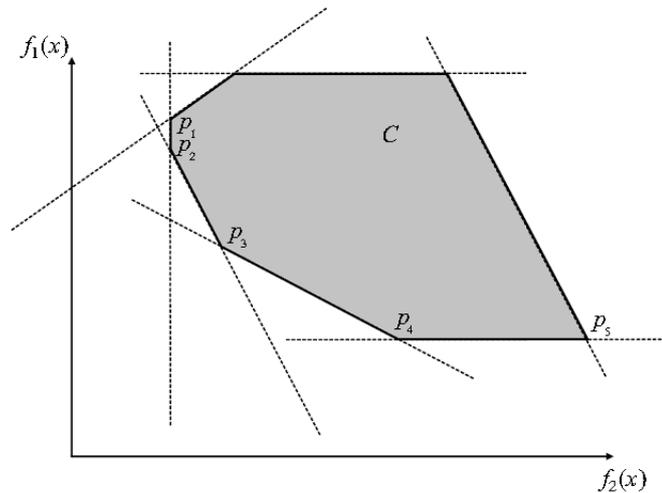
Figura 4.1: Exemplo de uma curva de Pareto.



FONTE: Caramia; Dell'olmo, 2008.

No próximo exemplo é mostrado como identificar os pontos fracos e restritos do Pareto ótimo.

Figura 4.2: Pontos fracos e restritos da frente de Pareto.



FONTE: Caramia; Dell'olmo, 2008.

No gráfico da figura 4.2 os pontos  $p_1$  e  $p_5$  são pontos fracos de Pareto ótimo enquanto que  $p_2$ ,  $p_3$  e  $p_4$  são pontos restritos de Pareto ótimo.

### 4.3 Técnicas de Resolução de Problemas de Otimização Multiobjetivo

Em muitos casos as curvas de Pareto não podem ser computadas de maneira eficiente. Embora na teoria seja possível encontrar exatamente todos os pontos, é possível que os mesmos apresentem um tamanho exponencial. Exemplo: uma redução na íntegra do problema da mochila demonstra que o mesmo é do tipo *NP-hard* para computar. Para estas condições existem métodos de aproximação que são frequentemente utilizados nos mesmos. No entanto, muitas vezes a aproximação não representa uma escolha secundária para o tomador de decisão. De fato, existem muitos problemas da vida real dos quais são muito difíceis para o tomador de decisão colher todas as informações para formular estes problemas de maneira correta. Levando em consideração que os tomadores de decisão tendem a entender mais sobre o problema assim que algumas soluções preliminares são apresentadas ter algumas soluções aproximadas pode ajudar o tomador de decisão em sua tarefa conforme apresentado por (RUZICA; WIECEK, 2005).

Técnicas de aproximação geralmente apresentam objetivos variados tais como: representação de um grupo de soluções gráficas que podem ser convertidos para números (problemas multiobjetivo convexos); representação de um grupo de soluções gráficas onde apenas algumas curvas de Pareto podem ser representadas numericamente (problemas

multiobjetivo não-lineares); representação de um grupo de soluções gráficas onde todo grupo eficiente de soluções não pode ser representado numericamente (problemas multiobjetivo discretos).

De fato a otimização multiobjetivo ou "multicritério" tem sido alvo de várias pesquisas. (RUZICA; WIECEK, 2005), (EHRGHOTT, 2005), (EHRGHOTT, 2006) e (T'KINDT; BILLAUT, 2005) relatam um estudo reunindo as mais variadas técnicas de otimização. No trabalho de (RUZICA; WIECEK, 2005) as diversas técnicas são aplicadas a duas funções objetivo separadamente e um caso onde o número de funções objetivo é estritamente maior que dois. No livro de (EHRGHOTT, 2005) e no seu artigo de 2006 diversas técnicas de transformação escalar são abordadas. Em (T'KINDT; BILLAUT, 2005) dedicam um capítulo do seu livro para "programação multicritério" no qual aborda vários métodos de otimização multiobjetivos.

#### 4.3.1 A Técnica de Transformação Escalar

Um problema multiobjetivo geralmente é solucionado através da combinação de vários objetivos em uma única função objetivo escalar. Esta técnica é denominada como soma ponderada ou método de transformação escalar. Mais detalhadamente, o método da soma ponderada minimiza positivamente a soma ponderada conexa das funções objetivo, tal que:

$$\min \sum_{i=1}^n \gamma_i \cdot f_i(x) \quad 4.3$$

$$\sum_{i=1}^n \gamma_i = 1 \quad 4.4$$

$$\gamma_i > 0, i = 1, \dots, n$$

$$x \in S,$$

que representa uma nova otimização de apenas uma função objetivo. A esta nova denominação da soma ponderada pode-se atribuir a função  $P_r(\gamma)$ .

É possível provar que ao minimizar a função objetiva única  $P_f(\gamma)$  encontra-se uma solução eficiente para o problema multiobjetivo original, devido ao fato da imagem desta função pertencer a curva de Pareto. De forma particular, se o vetor ponderado  $\gamma$  tiver todos os seus elementos maior que zero, então a minimização converte para a região ótima de Pareto restrita ( $P_r(\gamma)$ ), em contrapartida se ao menos 1 elemento de  $\gamma$  for igual a zero, então a minimização converte para a região ótima de Pareto fraca ( $P_f(\gamma)$ ).

Não existe uma relação a - priori entre o vetor ponderado  $\gamma$  e a solução do vetor, cabe ao tomador de decisão determinar os coeficientes apropriados, muito embora o coeficiente não se relacione diretamente com a importância das funções objetivo. Além do fato de o tomador de decisão não estar ciente de quais coeficientes são apropriados para obter a solução de forma satisfatória, ele ou ela, não sabe como alterar os coeficientes de maneira que haja uma mudança consistente na solução. Tendo isto em mente pode-se afirmar também que a tarefa de desenvolver uma heurística para auxiliar o tomador de decisão é significativamente difícil, pois se deve partir do conceito de inicializar a busca com determinados coeficientes e de forma interativa com os vetores ponderados alcançar determinada região da curva de Pareto.

Uma vez que a criação de um vetor ponderado leva à apenas um ponto na curva de Pareto, realizar várias otimizações exige bastante da parte computacional, no entanto cabe ao tomador de decisão determinar quais combinações diferentes dos coeficientes devem ser consideradas para que uma boa parte da frente de Pareto possa ser representada.

Além da possibilidade de um tempo de computação muito alto, o método de transformação escalar também apresenta duas deficiências técnicas que são explicadas a seguir:

- A relação entre os coeficientes da função objetivo e a curva de Pareto é tal que uma distribuição uniforme dos coeficientes, em geral, não reproduz uma distribuição uniforme na curva de Pareto. Isto pode ser observado pelo fato de que alguns pontos são agrupados em algumas partes da frente de Pareto enquanto algumas partes (às vezes as mais significantes) não são reproduzidas.
- As partes não convexas do grupo de Pareto não podem ser alcançadas pela combinação da minimização convexa das funções objetivo. Para ilustrar, considera-se uma interpolação geométrica do método de soma ponderada em duas dimensões, quando  $n = 2$ . Em um espaço bidimensional a função objetivo pode ser expressa da seguinte forma:

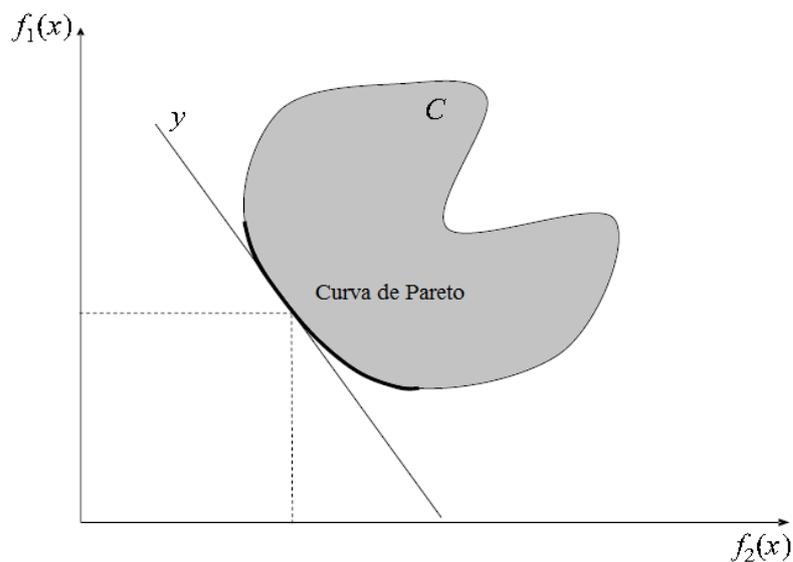
$$y = \gamma_1 \cdot f_1(x) + \gamma_2 \cdot f_2(x),$$

onde

$$f_2(x) = -\frac{\gamma_1 \cdot f_1(x)}{\gamma_2} + \frac{y}{\gamma_2}. \quad 4.6$$

A minimização de  $\gamma \cdot f(x)$  no método de soma ponderada pode ser interpretado como a tentativa de encontrar o valor de  $y$  tal que a reta cuja inclinação é determinada por  $-\frac{\gamma_1}{\gamma_2}$  seja tangente a região  $C$ . Obviamente, ao mudar os parâmetros dos coeficientes é possível alcançar diferentes pontos contidos na frente de Pareto. Se a curva de Pareto for convexa existe maior possibilidade de calcular tais pontos para diferentes vetores de  $y$  (como pode ser observado na figura 4.3).

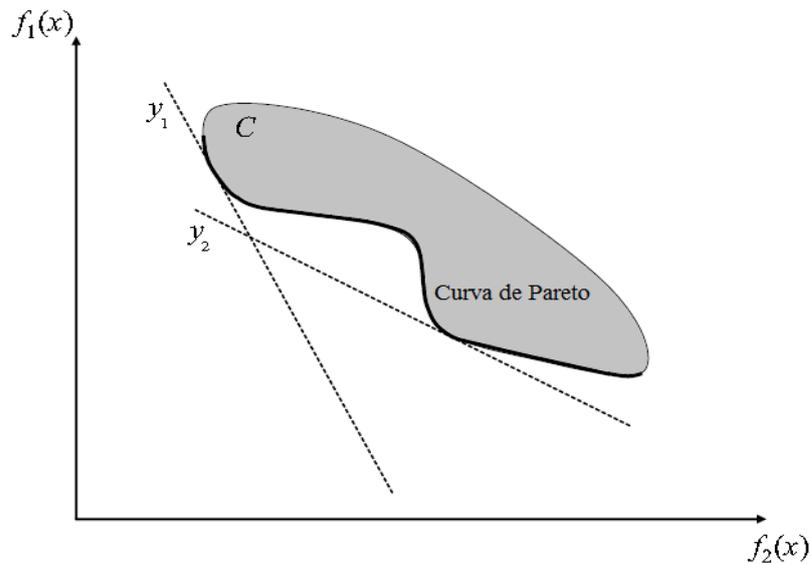
Figura 4.3: Soma ponderada no caso da curva de Pareto convexa.



FONTE: Caramia; Dell'olmo, 2008.

De forma análoga, quando a curva é não convexa existe um grupo de pontos que não pode ser alcançado por quaisquer combinações do vetor ponderado  $\gamma$  (conforme o gráfico da figura 4.4 a seguir).

Figura 4.4: Soma ponderada no caso da curva de Pareto não-convexa.



FONTE: Caramia; Dell'olmo, 2008.

O seguinte resultado alcançado por (GEOFFRION, 1968) as condições necessárias e suficientes para casos de convexidade tal que:

"Se um grupo de soluções  $S$  for convexo e os  $n$  objetivos de  $f_i$  são convexos em  $S$ , onde  $x^*$  é considerado uma solução restrita de Pareto ótimo se e somente se existir um  $\gamma \in \mathbb{R}^n$  tal que  $x^*$  é uma solução ótima do problema  $P_r(\gamma)$ ". Da mesma forma: "Se um grupo de soluções  $S$  for convexo e os  $n$  objetivos de  $f_i$  são convexos em  $S$ , onde  $x^*$  é considerado uma solução fraca de Pareto ótimo se e somente se existir um  $\gamma \in \mathbb{R}^n$  tal que  $x^*$  é uma solução ótima do problema  $P_f(\gamma)$ ".

Se a hipótese de convexidade não for satisfeita, apenas as condições necessárias permanecem válidas, isto é, as soluções ótimas de  $P_r(\gamma)$  e  $P_f(\gamma)$  são respectivamente os ótimos de Pareto restrito e fraco.

### 4.3.2 Método das Restrições $\epsilon$

Além da abordagem da transformação escalar, outra técnica de solução para otimização do método das restrições  $\epsilon$  proposto por (CHANKONG; HAIMES, 1983). Neste método, o tomador de decisão escolhe um entre  $n$  objetivos a ser minimizado enquanto que o restante dos objetivos é limitado a serem menores ou iguais a um determinado valor definido.

Matematicamente falando, se determinada função  $f_2(x)$  for escolhida para ser encontrado o seu valor mínimo, o seguinte problema de otimização é apresentado na forma de:

$$\min f_2(x) \quad 4.7$$

$$f_i(x) \leq \varepsilon_i, \forall i \in \{1, \dots, n\} \setminus \{2\} \quad 4.8$$

$$x \in S.$$

É possível observar que a formulação do método das restrições  $\varepsilon$  foi derivada de uma forma mais generalizada por (MIETTINEM, 1994), que por sua vez provou que:

"Se um vetor  $j$  e um vetor  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{j-1}, \varepsilon_{j+1}, \dots, \varepsilon_n) \in R^{n-1}$  existir, tal que  $x^*$  é uma solução ótima para o seguinte problema  $P(\varepsilon)$ :

$$\min f_i(x) \quad 4.9$$

$$f_i(x) \leq \varepsilon_i, \forall i \in \{1, \dots, n\} \setminus \{j\} \quad 4.10$$

$$x \in S,$$

*então  $x^*$  é um ótimo de Pareto fraco.*"

Por sua vez, o teorema de Miettinen é derivado de um teorema mais geral apresentado por (YU, 1974) que afirma que:

*" $x^*$  é uma solução ótima de Pareto restrita se e somente se para cada objetivo  $j$ , com  $j = \{1, \dots, n\}$  exista um vetor  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{j-1}, \varepsilon_{j+1}, \dots, \varepsilon_n) \in R^{n-1}$  tal que  $f(x^*)$  é o vetor objetivo único que corresponde à solução ótima do problema  $P(\varepsilon)$ ."*

De fato o teorema de Miettinen é uma versão de mais fácil implementação do que o resultado da pesquisa de Yu pois, o resultado *fraco* de Miettinen permite que seja usada apenas a condição necessária para calcular a solução ótima de Pareto fraca independente da singularidade das soluções ótimas. Contudo, se o grupo  $S$  e os objetivos forem convexos, esta condição torna-se unicamente necessária para se calcular o ótimo de Pareto fraco. Desta forma ao analisarmos o problema de  $P(\varepsilon_2)$  onde o objetivo dois é fixado temos conforme exposto

pelo teorema de Miettmen uma versão mais simplificada e portanto uma versão mais facilmente implementada para sistemas automatizados de apoio a decisão, no entanto, não é possível afirmar que na presença de um grupo  $S$  conexo e de funções  $f_i$  convexas tal que  $i = 1, \dots, n$ , seja possível calcular todas as soluções ótimas de Pareto fraco apenas variando o vetor  $\varepsilon$ .

Uma vantagem do método das restrições  $\varepsilon$  está na possibilidade de alcançar eficientemente os pontos de uma curva de Pareto não-convexa. Por exemplo, considere que temos duas funções objetivo onde a função  $f_1(x)$  é escolhida para ser minimizada, então a formulação do problema de otimização se apresenta da seguinte forma:

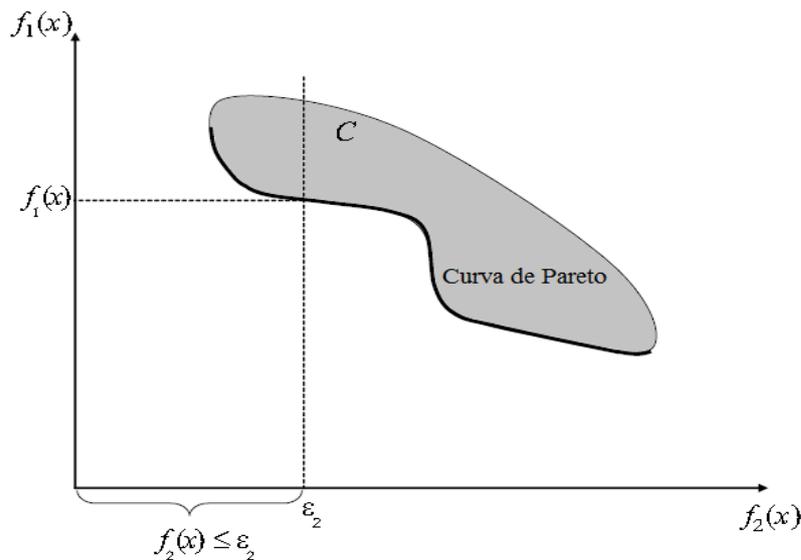
$$\min f_1(x) \quad 4.11$$

$$f_2(x) \leq \varepsilon_2 \quad 4.12$$

$$x \in S,$$

onde é possível que seja encontrada a situação descrita no gráfico da figura 4.5 tal que: quando  $f_2(x) = \varepsilon_2$ ,  $f_1(x)$  é um ponto eficiente da curva de Pareto não-convexa.

Figura 4.5: Método das restrições  $\varepsilon$  no caso da curva de Pareto não-convexa.



FONTE: CARAMIA; DELL'OLMO, 2008.

Portanto, o método proposto por (STEURER, 1986) estabelece que o tomador de decisão pode variar os limites superiores  $\varepsilon_i$  para obter o ótimo de Pareto fraco o que claramente deriva do método de (YU, 1974), isto quer dizer que, o tomador de decisão deve escolher os limites superiores apropriados para as restrições que são os valores de  $\varepsilon_i$ . Apesar destas vantagens o método deixa de ser eficiente quando existem mais de duas funções objetivo sendo que uma será minimizada e as restrições que determinam o limite superior variam entre si tornando impossível determinar um limite superior único.

Por esta razão, (ERGHOTT; RUSIKA, 2005), propõe duas modificações ao método para melhorá-lo, dando atenção particular as dificuldades computacionais que o novo método agrega.

### 4.3.3 Programação por Objetivo

A programação por objetivo inicia-se com (CHARNESET et al., 1955) e (CHARNES; COOPER, 1961). Este método não impõe uma questão de maximizar múltiplos objetivos, mas ao invés, tenta encontrar valores de objetivos específicos para estas funções. Como exemplo temos o seguinte programa:

$$f_1(x) \geq v_1 \quad 4.13$$

$$f_2(x) = v_2 \quad 4.14$$

$$f_3(x) \leq v_3 \quad 4.15$$

$$x \in S.$$

Pode-se distinguir claramente dois casos, no primeiro se existe a interseção entre o grupo da imagem  $C$  e o grupo de utopia (objetivo ideal), e o no segundo se a imagem admissível dos objetivos é nula ou não existe. Para o primeiro caso, o problema transforma em um dos quais se deve encontrar a solução cujos valores são os mais próximos possíveis do grupo de utopia. Para que isto seja alcançado, variáveis adicionais e restrições devem ser adicionadas. Em particular para cada tipo de restrição no formato da desigualdade a seguir  $f_1(x) \geq v_1$  uma variável  $s_1^-$  tal que a primeira restrição possa ser reescrita da seguinte forma:

$$f_1(x) + s_1^- \geq v_1. \quad 4.16$$

Para cada restrição do tipo  $f_2(x) = v_2$  são introduzidas duas variáveis excedentes  $s_2^+$  e  $s_2^-$  de modo que a segunda restrição pode ser reescrita como:

$$f_2(x) + s_2^- - s_2^+ = v_2. \quad 4.17$$

E finalmente para as restrições do tipo  $f_3(x) \leq v_3$  uma variável  $s_3^+$  deve ser introduzida de modo que a restrição anterior pode ser reescrita tal que:

$$f_3(x) - s_3^+ \leq v_3. \quad 4.18$$

Considere que  $s$  seja o vetor das variáveis adicionais. Uma solução  $(x, s)$  para o problema anterior é definida como ótimo de Pareto-lento restrito se e somente se uma solução  $(x', s')$ , onde para cada  $x' \in S \forall s'_i \leq s_i$  com pelo menos uma desigualdade, não exista.

Existem diferentes métodos de aplicar a otimização de excedente/redução de variáveis. Como exemplo pode-se considerar a programação por objetivo de Arquimedes, onde o problema torna-se a minimização da combinação linear das variáveis excedentes e de redução de cada coeficiente ponderado por um coeficiente positivo  $\alpha$  como é possível observar a seguir:

$$\min \alpha_{s_1} - s_1^- + \alpha_{s_2} + s_2^+ + \alpha_{s_2} - s_2^- + \alpha_{s_3} + s_3^+ \quad 4.19$$

$$f_1(x) + s_1^- \geq v_1 \quad 4.20$$

$$f_2(x) + s_2^- - s_2^+ = v_2 \quad 4.21$$

$$f_3(x) - s_3^+ \leq v_3 \quad 4.22$$

$$s_1^- \geq 0$$

$$s_2^+ \geq 0$$

$$s_2^- \geq 0$$

$$s_3^+ \geq 0$$

$$x \in S.$$

Conforme o teorema de Geoffrion o resultado da otimização por objetivo gera um ótimo de Pareto lento restrito ou fraco.

Além da programação por objetivo de Arquimedes, existem outros métodos como a programação por objetivo lexográfica, programação por objetivo interativa, a programação por objetivo de referência e a programação por objetivo de multicritérios como exposto por (T'KINDT; BILLAUNT, 2005).

#### 4.3.4 Programação Multinível

A programação multinível é outra abordagem de otimização multiobjetivo e procura encontrar um ponto ótimo em toda a superfície de Pareto. A programação multinível ordena os  $n$  objetivos de acordo com a hierarquia. Primeiramente, os minimizadores da primeira função objetivo são encontrados, em seguida os minimizadores do segundo objetivo mais importante são encontrados e assim por diante até que todas as funções tenham sido otimizadas a grupos menores sucessivamente.

Quando a ordem hierárquica entre os objetivos deve ser considerada a programação multinível se torna bastante útil ainda mais quando o usuário não tem interesse nas trocas contínuas entre as funções. Uma desvantagem da programação multinível reside no fato de que em determinados problemas de otimização que são solucionados próximos do fim da hierarquia podem ser altamente restringidos e podem tornar a solução inapta restringindo assim os objetivos menos importantes de exercerem influência na solução ótima global.

Em uma programação matemática de dois níveis o tomador de decisão está lidando com dois problemas de otimização onde a região de aptidão do primeiro problema, conhecido como problema de nível superior (ou líder), é determinada pelo conhecimento prévio de outro problema de otimização, conhecido como problema de nível baixo (ou seguidor). Problemas que podem ser modelados utilizando a programação de dois níveis são aqueles em que as

variáveis do primeiro problema são restritas pelas soluções ótimas dos problemas de nível baixo.

Em geral a otimização de dois níveis é indicada para ser aplicada a problemas com dois tomadores de decisão onde a solução otimizada de um deles (líder) é restrita pela decisão otimizada do segundo tomador de decisão (seguidor). Em outras palavras, o tomador de decisão do segundo nível (líder) otimiza a sua função objetivo dentro da região de aptidão definida pelo tomador de decisão de primeiro nível (seguidor).

Um programa de dois níveis pode ser formulado como:

$$\min f(x_1, x_2) \quad 4.23$$

$$x_1 \in X_1$$

$$x_2 \in \arg \min g(x_1, x_2)$$

$$x_2 \in X_2$$

O analista deve prestar atenção particular ao empregar a otimização de dois níveis ou a otimização multinível em geral ao avaliar as soluções únicas do problema do seguidor. Assume-se que deve ser calculada uma solução ótima  $x_1^*$  para o modelo do líder. Deixe que  $x_2^*$  seja uma solução ótima do problema do seguidor associada com  $x_1^*$ . Se  $x_2^*$  não for única, isto é,  $|\arg \min g(x_1^*, x_2)| > 1$ , é possível que ocorra uma situação onde o tomador de decisão do seguidor fique livre para escolher para o seu problema, sem violar as restrições do líder, outra solução ótima diferente de  $x_2^*$ , isto é,  $\hat{x}_2 \in \arg \min g(x_1^*, x_2)$  sendo  $\hat{x}_2 \neq x_2^*$ , o que possivelmente resultará em  $f(x_1^*, \hat{x}_2) > f(x_1^*, x_2^*)$  no grupo de soluções do líder, forçando o mesmo a realizar uma análise mais específica dos valores obtidos pela sua função objetivo com relação a todas as soluções ótimas em  $\arg \min g(x_1^*, x_2)$ .

A programação de dois níveis está intimamente relacionada ao problema de equilíbrio de (STACKELBERG, 1952) e a programação matemática com restrições de equilíbrio exposta por (LUO et al., 1996). Os casos mais estudados de problemas de programação de dois níveis tem sido na aplicação da resolução de problemas lineares, e, portanto esta subclasse tem sido alvo de vários estudos dentre eles o exposto por (WEN; HSU, 1991) apresenta o maior número de casos estudados.

Com o passar dos anos, programas de dois níveis mais complexos tem sido estudados e até aqueles que incluem variáveis discretas receberam alguma atenção como no estudo realizado por (VICENTE et al., 1996). Desde que mais pesquisas generalizadas surgiram tais como a de (VICENTE; CALAMAI, 1994) e (FALK; LIU, 1995) em programação não-linear de dois níveis a pesquisa do formato combinatorial deste método tem sido intensificada como exposto por (MARCOTTE; SAVARD, 2005).

Programas de dois níveis são difíceis de resolver. Em particular na pesquisa de (HANSEN et al., 1992), mostra que a programação linear de dois níveis é um problema difícil de ordem (*Non Polynomial* - NP), ou seja, o tempo de processamento não segue uma função de tempo no formato polinomial e que foi reforçada pela pesquisa de (VICENTE et al., 1996) que mostra que confirmar o encontro de um ponto local ótimo também apresenta a mesma complexidade computacional.

Os programas de dois níveis podem ser distintos em duas classes: os lineares e não lineares. Embora sejam duas classes ambos convergem em algoritmos para programas bi-nível mais generalizados com propriedades teóricas que garantem as condições estacionárias adequadas. Como exemplo temos o método da função implícita proposto por (OUTRATA et al., 1998), a reformulação quadrática de nível único proposta por (SCHOLMES; STOHR, 1999) e o método de suavização estudados por (FUKUSHIMA; PANG, 1999) e reforçado por (DUSSAULT et al., 2006)

A respeito dos problemas de otimização com restrições complementares, que representam outra maneira de resolver programas de dois níveis. Nesta área pode-se mencionar os artigos de (KOCVARA; OUTRATA, 2004) que introduz um novo cenário teórico com o método da programação implícita, (BOUZA; STILL, 2007) apresentam o estudo das propriedades de convergência do método de suavização que permite a caracterização dos minimizadores locais onde todas as funções que definem o modelo são duplamente diferenciáveis e por fim (LIN; FUKUSHIMA, 2003) e (LIN; FUKUSHIMA, 2005) estudam dois métodos de relaxação.

Algoritmos exatos têm sido propostos para algumas classes especiais de programas de dois níveis, por exemplo, pode-se citar os métodos de enumeração de vértices propostos por (CANDLER; TOWNSLEY, 1982), (BIALAS; KARWAN, 1984) e (TUY et al., 1993) onde aplicam esta propriedade em programas lineares de dois níveis quando apenas somente uma solução extrema for mantida. Métodos de articulação complementares como propostos por (BIALAS et al., 1980) e (JÚDICE; FAUSTINO, 1992) têm sido apresentados para obter a

otimização dos problemas de nível-único pela substituição do problema do segundo nível de otimização pelas suas condições otimizadas.

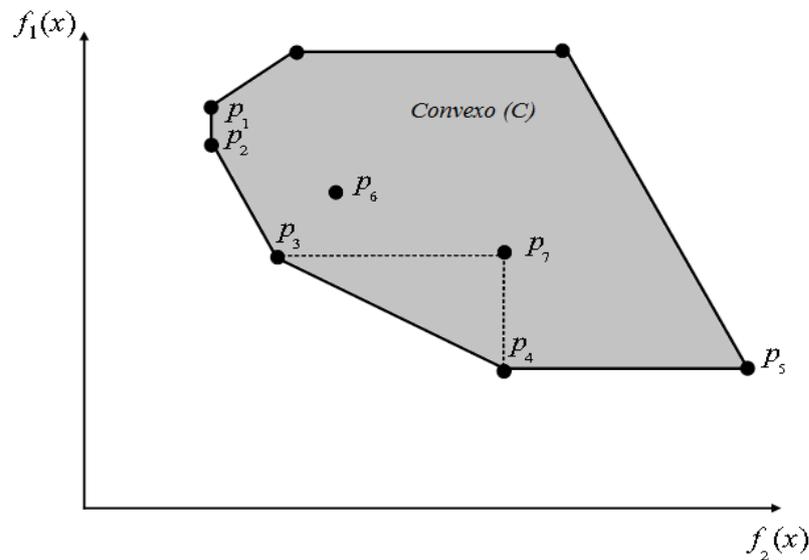
Seguindo esta linha de exploração das estruturas complementares de reformulação do nível-único, (BARD; MOORE, 1990) e (HANSEN et al., 1992) propuseram a implementação de algoritmos de ramificação e limites conhecidos como "branch-and-bound" que destacam-se por serem os mais eficientes para este tipo de formulação. Tipicamente o algoritmo de "branch-and-bound" é aplicado quando o problema de baixo nível for convexo e regular, sendo que este último caso possa ser substituído pelas condições de Karush-Kuhn-Tucker, produzindo uma reformulação de nível-único. Ao lidar com programas bi-nível, as condições complementares são intrinsecamente combinatoriais, e para tais casos o método de "branch-and-bound" é o melhor para resolver este tipo de problema conforme (COLSON et al., 2005).

O método de corte de plano (cutting-plane) não são frequentemente utilizados para resolver problemas lineares de dois níveis. Os métodos de corte de plano que são encontrados na literatura são baseados essencialmente nos cortes côncavos de (TUY, 1964). (WHITE; ANANDALINGAM, 1993) usaram estes cortes na forma de funções de penalidade para solucionar problemas lineares de dois níveis. No trabalho de (MARCOTTE et al., 1993) é proposto um algoritmo de corte de plano para resolver problemas lineares bi-nível com a garantia de uma terminação finita. Recentemente (AUDET et al., 2007) exploram a equivalência do modelo proposto por (MARCOTTE et al., 1993) com um problema linear misturado com números inteiros e propuseram um novo algoritmo de branch-and-bound com os cortes de Gomory incluídos para resolver problemas de programação linear de dois níveis.

#### **4.4 Problema de Otimização Multiobjetivo de Número Inteiro**

Na seção anterior, um aparato geral das técnicas de otimização multiobjetivo contínuas foi discutido. No entanto esta seção foca no que acontece quando os problemas de otimização apresentam restrições integralidade nas variáveis. Embora as técnicas anteriores possam ser adaptadas com um certo grau de capacidade de construir inteiramente a frente de Pareto. De fato, estes métodos são geralmente muito difíceis de aplicar em situações reais ou são incapazes de encontrar todas as soluções eficientes. Quando restrições de integralidade surgem, um dos principais fatores limitantes está no fato destas técnicas serem inaptas para obterem algumas soluções ótimas de Pareto; por isso, é possível que se apresente a situação ótima de Pareto *suportada* ou *não suportada*.

Figura 4.6: Ótimos de Pareto *suportado* e *não suportado*.



FONTE: CARAMIA; DELL'OLMO, 2008.

A figura 4.6 dá um exemplo destas situações: os pontos  $p_6$  e  $p_7$  são pontos ótimos de Pareto não suportado, enquanto  $p_1$  e  $p_5$  são pontos ótimos de Pareto fraco suportados, enquanto que os pontos  $p_2$ ,  $p_3$  e  $p_4$  são pontos ótimos de Pareto restrito suportados.

Dado um problema de otimização multiobjetivo de número inteiro, a transformação escalar em um problema de objetivo único com variáveis ou parâmetros adicionais para encontrar o subgrupo das soluções eficientes do problema de otimização original, tem a mesma complexidade computacional que o problema de transformação escalar no modo contínuo.

Em (EHRGOTT, 2006) é tratada também a satisfação dos requerimentos impostos pela programação multiobjetivo de números inteiros apresentando mais métodos como o abrandamento de Lagrange e o método das restrições elásticas.

Pela análise de Ehrgott, ao utilizar o método de abrandamento de Lagrange para resolver os problemas escalonados não traria resultados além dos já obtidos pela técnica das somas ponderadas. Também por sua análise constata que a formulação generalizada da transformação escalar apresenta uma complexidade computacional do tipo não polinomial difícil (*NP-hard*). Então Ehrgott apresenta uma nova técnica denominada método das restrições elásticas, que é capaz de superar a deficiência de pesquisar por soluções eficientes contidas nas técnicas mencionadas anteriormente pela combinação das vantagens dos métodos

da soma ponderada e das restrições- $\varepsilon$ . Além disso, uma aplicação prática é apresentada com um tempo de computação razoável. De fato os resultados obtidos por Ehrgott com o método das restrições elásticas são aplicados no problema de alocação de tripulação em companhias aéreas, cuja quantidade de restrições varia entre 500 a 2000 restrições, provando assim a eficácia da técnica proposta.

#### 4.4.1 Problema de Otimização Multiobjetivo dos Caminhos Mais Curtos

Considere um gráfico  $G = (V, A)$ , uma origem  $s \in V$  e uma destinação  $t \in V$ , é denominado problema do caminho mais curto (Shortest Path Problem - SPP) todo aquele onde o objetivo é encontrar a menor distância do caminho em  $G$  da origem  $o$  para o destino  $d$ . Este problema tem sido estudado por mais de 50 anos e vários algoritmos polinomiais tem sido propostos assim como em (CORMEM et al., 2001).

Do ponto de vista do problema de custos de operação da cadeia de suprimentos, o significado do termo mais curto pode variar e adquirir diferentes significados como o mais rápido ou mais rapidamente, para o mais seguro e assim por diante, o tomador de decisão deve condicionar o problema sempre levando em consideração o significado da representação do grupo de arcos  $A$ . Por esta razão em alguns casos é mais simples definir cada legenda dos arcos e suas características (por exemplo, comprimento, tempo percorrido, risco estimado, etc). O problema de encontrar os caminhos mais curtos de problemas multiobjetivo (*Multiobjective Shortest Path Problem* - MOSPP) é conhecido como sendo de complexidade computacional do tipo *NP-hard* assim como exposto por (SERAFINI, 1986), sendo que os algoritmos propostos nas literaturas relacionadas encontram dificuldade de gerenciar um número alto de caminhos não dominados o que resulta em um tempo computacional relativamente alto, mesmo considerando casos de poucos nós. (HANSEN, 1979) prova que o número de caminho não dominado cresce exponencialmente com o aumento do número de nós.

Em um cenário multiobjetivo, cada arco  $(i, j)$  tem um vetor de custos no gráfico denominado  $c_{ij} \in R^n$  composto por  $c_{ij} = (c_{ij}^1, \dots, c_{ij}^n)$  componentes onde  $n$  representa o número de critérios.

Um caminho  $P_{si}$  da origem  $s$  para o nó  $i$  representa uma sequência de nós (e arcos) tal que  $P_{si} = (s \equiv n_1, \dots, n_h \equiv i)$  cujo comprimento  $h > 2$ , para cada arco  $(n_l, n_{l+1}) \in A$  para

$l = 1, \dots, h - 1$ . Tal caminho  $P_{si}$  é avaliado de acordo com o desempenho do vetor  $c(P_{si}) = (c^1(P_{si}), \dots, c^n(P_{si}))$  onde:

$$c^l(P_{si}) = \sum_{(p,q) \in P_{si}} c_{pq}^l \quad 4.24$$

sendo  $l = 1, \dots, n$ .

Seja  $x_{ij}$  uma variável binária tal que:

$$x_{ij} = \begin{cases} 1 & \text{se } (i, j) \text{ está no caminho;} \\ 0 & \text{se não.} \end{cases} \quad 4.25$$

Levando em consideração a equação 4.24 e a variável binária  $x_{ij}$  é possível definir o problema do caminho mais curto multiobjetivo como sendo:

$$\min f^l(x) = \sum_{(i,j) \in A} c_{ij}^l x_{ij} \quad \forall l \in \{1, \dots, n\} \quad 4.26$$

sujeito à:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ij} = \begin{cases} 1 & i = s \\ 0 & \forall i \in V \setminus \{s, t\} \\ -1 & i = t \end{cases} \quad 4.27$$

$$x_{ij} \geq 0, \forall (i, j) \in A.$$

Seja um espaço de solução da equação 4.26 denominado por  $S_{st}$ . Considera-se  $P_{st}^1$  e  $P_{st}^2$  dois caminhos aptos cujos vetores  $c(P_{st}^1)$  e  $c(P_{st}^2)$  representam suas performances respectivamente.

Desta maneira para analisar a dominância entre as soluções considera-se que dado um caminho  $P_{st}^1 \in S_{st}$  e o vetor  $c(P_{st}^1)$ , considera-se que a solução é não-dominada se não existir outro vetor  $c(P_{st}^2)$ , com  $P_{st}^2 \in S_{st}$ , tal que  $c^l(P_{st}^2) \leq c^l(P_{st}^1)$ , sendo  $l = 1, \dots, n$  e  $c^l(P_{st}^2) \neq c^l(P_{st}^1)$  para alguns  $l$ ; se não  $c^l(P_{st}^2)$  domina a solução  $c^l(P_{st}^1)$ .

De forma correspondente as soluções eficientes são classificadas na frente de Pareto da seguinte forma:

- Um caminho  $P_{st}^1$  é considerado estritamente eficiente (ou solução ótima de Pareto restrito) se e apenas se  $P_{st}^1 \in S_{s,t}$  e não exista outro caminho  $P_{st}^2 \in S_{s,t}$  cujo vetor de desempenho  $c^l(P_{st}^2)$  domine  $c^l(P_{st}^1)$ .
- Um caminho  $P_{st}^1$  é considerado fracamente eficiente (ou solução ótima de Pareto fraco) se e apenas se  $P_{st}^1 \in S_{s,t}$  e não exista outro caminho  $P_{st}^2 \in S_{s,t}$  cujo vetor de desempenho  $c^l(P_{st}^2)$  domine restritamente  $c^l(P_{st}^1)$ .

Em seu trabalho de 1979, Hansen apresentou um dos primeiros estudos em problemas de caminhos mais curtos multiobjetivo. Hansen analisou alguns problemas de dois objetivos e demonstrou que, nos piores casos, o número de caminhos não dominados cresce exponencialmente com o tamanho da rede. Então fica claro que, no pior caso, a geração de uma frente de Pareto eficiente pode requerer um grande esforço computacional. Desta forma um método que pode ser utilizado para encontrar o subgrupo dos caminhos não dominados seria o uso da combinação linear dos objetivos estudados, mudando apenas os coeficientes utilizados. Seguindo esta linha de pesquisa (HENIG, 1986) propõe a utilização do método anterior para encontrar todos os caminhos não-dominados, provando por este novo método que a média de tempo de computação obedece a um padrão polinomial com o crescimento da rede. No mesmo contexto (KOSTREVA; WIECEK, 1993) apresentaram métodos de programação dinâmica porém, os estudos dos algoritmos multiobjetivos para encontrar a solução do problema dos caminhos mais curtos foram dedicados unicamente a duas vertentes : na primeira vertente o algoritmo é baseado no ajuste de designação e a segunda vertente baseia o algoritmo na correção da designação. Na primeira classe, a maioria dos trabalhos dedicou-se a problemas onde todas as funções objetivo são do tipo somatório como proposto por (HANSEN, 1979) e (MARTINS, 1984). (MARTINS, 1999) propôs um algoritmo multiobjetivo baseado nas técnicas de designação dos vértices que generaliza o princípio de “otimalidade” de Bellman. Martins assume que os atributos associados aos arcos são não-negativos, determinísticos e aditivos ao longo da rota. O algoritmo de Martin usa um método de designação múltipla. Cada nó  $i \in V$  é associado a várias legendas e a  $l$ -ésimo legenda contém os valores dos  $n$  objetivos e dois ponteiros. A legenda pode ser representada como

$$[c^1(P_{si}), \dots, c^r(P_{si}), j, l_1]l, \quad 4.28$$

onde  $c^h(P_{si})$  representa o comprimento do caminho  $P_{si}$  da origem  $s$  para o nó  $i$ , para  $h = 1, \dots, n$ ,  $j \neq i$  é algum nó de  $G$ , e  $l_1$  indica certa legenda do nó  $j$  no qual,

$$c_{[h,l]}(P_{si}) = c_{[h,l_1]}(P_{sj}) + c_{ij}^h, \quad 4.29$$

onde  $c_{[h,l]}(P_{si})$  é o  $h$ -ésimo componente da  $l$ -ésima legenda do nó  $i$ .

A cada interação, existem dois tipos de legenda classificadas como: permanentes e temporárias. Um algoritmo seleciona uma legenda temporária em um nó  $i$ , a converte em permanente e atualiza todas as legendas dos sucessores  $j$  de  $i$ , para cada  $(i, j) \in A$ ; então o mesmo deleta todas as legendas que representam um caminho dominado  $P_{sj}$ . O algoritmo para quando todas as legendas temporárias acabam, sendo que cada legenda permanente representa um caminho eficiente único.

O passo da seleção de nós é feito levando-se em consideração a, entre todas as legendas de cada nó, lexicograficamente (modo de representação dos caminhos em um gráfico) menor. Para ilustrar melhor o termo de representação gráfica de distâncias considere que para alguns nós  $i$ , uma legenda,

$$[c^1(P_{si}), \dots, c^n(P_{si}), -, -]\xi \quad 4.30$$

é dita ser lexicograficamente menor do que a legenda

$$[c^1(P'_{si}), \dots, c^n(P'_{si}), -, -]\delta, \quad 4.31$$

se  $c_{[1,\xi]}(P_{si}) = c_{[1,\delta]}(P'_{si}), \dots, c_{[k-1,\xi]}(P_{si}) = c_{[k-1,\delta]}(P'_{si})$  e  $c_{[k,\xi]}(P_{si}) < c_{[k,\delta]}(P'_{si})$  são mantidos para alguns termos de  $k \in \{1, \dots, n\}$ .

(GANDIBLEUX et al., 2006) propuseram em seu trabalho um problema envolvendo um objetivo max-min e dois objetivos min-soma resolvido com uma versão atualizada do algoritmo de Martins. Seguindo a mesma linha de pesquisa de Martins pode-se citar (BRUMBAUGH-SMITH; SHIER, 1989), (MOTE et al., 1991), e (SKRIVER; ANDERSEN, 2000) cujas pesquisas focam em diferentes algoritmos onde as legendas dos nós tornam-se permanentes apenas na última interação do algoritmo.

Para os casos de dois objetivos, (MARTINS; CLIMACO, 1981) e (MARTINS; CLIMACO, 1982) desenvolveram algoritmos para o problema de ranking de caminhos e para

os problemas de ordenação máxima (EHRGOTT; SKRIVER; 2003) desenvolveram suas soluções também de modo satisfatório. Na pesquisa de (PAIXÃO et al., 2003) o ranking dos caminhos é obtido através de um algoritmo multiobjetivo do caminho mais curto utilizando uma função utilitária onde o valor normal foi associado a cada caminho.

Como (TARAPATA, 2007) descreve em seu trabalho, um dos métodos mais utilizados para resolver problemas multiobjetivos dos caminhos mais curtos é o da construção das curvas  $(1 + \varepsilon)$  de Pareto e revisto também pelas pesquisas de (PAPADIMITRIOU; YANNAKAKIS, 2000) e (VASSILVITSKII; YANNAKAKIS, 2004). Explicando a técnica de uma maneira simplificada, uma curva  $P_\varepsilon$  da frente  $(1 + \varepsilon)$  de Pareto representa um subgrupo de soluções aptas tal que para qualquer solução ótima de Pareto exista uma solução em  $P_\varepsilon$  que não está a mais de  $(1 - \varepsilon)$  de distância nos objetivos conforme exposto por Tarapata (2007). No trabalho de (PAPADIMITRIOU; YANNAKAKIS, 2000) foi descrito que para qualquer problema de otimização multiobjetivo existe uma curva de tamanho (polinomial)  $P_\varepsilon$  da frente  $(1 + \varepsilon)$  de Pareto. Algumas extensões deste método são apresentadas na pesquisa de (VASSILVITSKII; YANNAKAKIS, 2004) que demonstram o método capaz de produzir uma aproximação constante para a menor curva possível da frente  $(1 + \varepsilon)$  de Pareto para os casos de dois ou três objetivos, enquanto que para objetivos maiores que três os resultados da aproximação constante não é satisfatório.

Ainda na pesquisa de (PAPADIMITRIOU; YANNAKAKIS, 2000) foi demonstrado como uma rotina de espaço pode ser construída (baseada em um algoritmo pseudo-polinomial para processar os caminhos exatos) e conseqüentemente, fornece uma técnica de aproximação polinomial do tempo total para este tipo de problema visando reduzir a sua complexidade computacional. Para o caso de técnicas de aproximação polinomial do tempo de computação para problemas multiobjetivo do caminho mais curto para o caso de dois objetivos como demonstrado por (HANSEN, 1979), e para o caso de problemas multiobjetivos de gráficos acíclicos direcionados o trabalho de (WARBURTON, 1987) é tido como referência. Atualmente, para o caso particular envolvendo somente dois objetivos as pesquisas para transformação polinomial do tempo computacional têm sido amplamente estudadas, como pode ser visto na pesquisa de (EHRGOTT; GADIBLEUX, 2002). Contudo para os casos com mais de dois objetivos muito pouco têm sido alcançado. Atualmente, neste campo de transformação polinomial, os trabalhos de (WARBURTON, 1987), (PAPADIMITRIOU; YANNAKAKIS, 2000) e por fim (TSAGGOURIS; ZAROLIAGIS, 2005) são tidos como sendo as melhores técnicas de aproximação polinomial conhecidas.

Referenciando a pesquisa de (WARBURTON, 1987) o método de aproximação polinomial é responsável por gerar todos os caminhos "quasi" ótimos de Pareto. Uma outra abordagem deste problema seria tratar cada objetivo menos importante como uma restrição, assim como exposto por (SANCHO, 1988), desta forma quando considera-se uma problema com três objetivos, os dois menos significativos são tratados como restrições. Desta forma para qualquer solução gerada não podemos considerá-la como sendo uma solução ótima de Pareto se a relacionarmos aos três objetivos.

Uma abordagem utilizando atributos determinísticos foi proposta por (LIST et al., 1991). Em sua pesquisa em particular List desenvolveu um método que encontra, dada uma origem e um destino, os caminhos não dominados substituindo os seus atributos estocásticos (supostos normalmente distribuídos) de média e variância, reduzindo desta forma o número de caminhos não dominados pois, um critério de dominância estocástica é utilizado para reduzir o número de caminhos não dominados tomados dois a dois.

#### **4.4.2 O Problema do Caixeiro Viajante Multiobjetivo**

O problema do caixeiro viajante (conhecido na literatura como TSP) é um problema que envolve distâncias mais curtas, porém, considerado diferente do problema do caminho mais curto. Agora o grupo de nós  $V$  do gráfico  $G = (V, A)$  deve ser visitado uma única vez. Se os arcos são legendados com os seus respectivos comprimentos o problema torna-se o de encontrar o caminho total de menor comprimento. Relembrando que, para o problema do caminho mais curto, as legendas dos arcos podem conter diferentes significados. Como referência mais atual sobre o estudo do problema do caixeiro viajante e suas variações o livro de (GUTIN; PUNNEL 2006) apresenta uma explanação considerável sobre o tópico.

(MANTHLEY; RAM, 2006) propuseram um vasto estudo do estado da arte de problemas multiobjetivo envolvendo o problema do caixeiro viajante. (EHRGOTT, 2000) também contribuiu significativamente com a sua pesquisa no campo do problema do caixeiro viajante com múltiplos critérios. Além destes trabalhos resultados expressivos na resolução do problema do caixeiro viajante multiobjetivo também foram alcançados na pesquisa feita por (ANGEL et al., 2004) e (ANGEL et al., 2005).

Na pesquisa de (EHRGOTT, 2000) uma generalização do algoritmo de Christofides foi feita e aplicada a um TSP com desigualdades triangulares e de propriedade simétrica conhecida como  $\Delta$ -STSP. Ao invés de considerar as curvas de Pareto, Ehrgott mediu a

qualidade da solução como sendo uma norma do vetor das funções objetivo, ou seja, Ehrgott realizou uma transformação escalar dos vários objetivos em um só. A taxa de aproximação obtida ficou entre  $\frac{3}{2}$  e 2, dependendo da norma usada para combinar os diferentes critérios.

Seguindo a linha de pesquisa de Ehrgott, (ANGEL et al., 2004) também considerou um TSP simétrico com dois critérios onde os mesmos foram considerados como o peso um e peso dois. O algoritmo alcançado apresentou uma aproximação de  $\frac{3}{2}$  para o problema em questão através do uso de uma heurística de pesquisa local.

Logo em seguida, (ANGEL et al., 2005) generalizaram os resultados de sua pesquisa anterior para um TSP simétrico de  $k$ -critérios cujos critérios foram nomeados de peso um e peso dois que apresentaram uma aproximação de  $2 - \frac{2}{k+1}$ , sendo  $k \geq 3$ .

Na pesquisa mais recente sobre o TSP simétrico multiobjetivo (MANTHLEY; RAM, 2006) apresentaram novos resultados obtidos através da aproximação  $(1 - \gamma - \varepsilon)$  sendo  $\gamma \in \left[\frac{1}{2}, 1\right]$ .

#### 4.4.3 Outros Trabalhos em Problemas de Otimização Combinatorial Multiobjetivo

No artigo intitulado "Um método de ajuste de nível para otimização combinatorial multiobjetivo: aplicação ao problema de designação quadrática", (EHRGOTT et al., 2006) estudaram o problema de designação em um caso multiobjetivo.

A versão básica do bastante conhecido problema de atribuição (assignment problem - AP) pode ser definido em um gráfico de duas partes. Uma maneira interessante de tratar um problema de atribuição seria o de encontrar o número de arestas que não possuem vértices em comum, por exemplo, para modelar o problema considera-se a atribuição de cargas para caminhões ou pessoas (motoristas) para veículos então um modo de maximizar a atribuição de cargas seria o mesmo que tentar encontrar o número máximo de cargas atribuídas para os caminhões (onde as arestas entre uma carga e um caminhão significam um grau de compatibilidade). Em outros casos as arestas podem ser legendadas, por exemplo, com uma estimativa do pagamento obtido através de certa atribuição, então o objetivo seria pesquisar pelo grupo de arestas sem vértices em comum que maximize o pagamento total estimado.

De modo similar, se as legendas das arestas representarem um custo, o objetivo torna-se o de encontrar a atribuição que minimize o custo total. (AHUJA et al., 1993) apresentam em

seu livro intitulado "Network Flows: Theory, Algorithms, and Applications" uma série de algoritmos e variações dos problemas de atribuição e de fluxo de rede.

No estudo da versão de atribuição quadrática do problema com múltiplos objetivos, (EHRGOTT et al., 2006) propôs uma meta dupla: a primeira delas propõe o desenvolvimento de um procedimento para a determinação das soluções ótimas de Pareto em problemas de otimização combinatorial multiobjetivo e a segunda meta demonstra a eficácia do método, discutido anteriormente no capítulo 3, aplicado ao QAP. O método proposto para alcançar a primeira meta tem como base o resultado da pesquisa realizada por (EHRGOTT et al., 1997). A rotina, baseada na noção de níveis de grupos e níveis de curvas, trabalha com um número arbitrário de objetivos, cujo propósito utilizava um algoritmo que resolvesse o problema de encontrar a melhor solução  $K$  em um problema de otimização combinatorial multiobjetivo. Com relação à segunda meta de resolver o problema de atribuição quadrática multiobjetivo (*Multiobjective Quadratic Attribution Problem - MQAP*), o método apresentado demonstrou bastante eficiência e também através da literatura tornou-se o primeiro método capaz de lidar com este tipo de problema combinatorial. Ainda na mesma pesquisa de 2006 foram apresentados dois algoritmos que classificavam as soluções do QAP e reportaram as experiências comparativas realizadas utilizando vários exemplos aleatórios cujos objetivos variavam entre dois a seis.

#### **4.4.4 Otimização Combinatorial Multiobjetivo Utilizando Metaheurística**

O primeiro trabalho compreensivo no campo da otimização combinatorial multiobjetivo que utiliza a metaheurística foi publicado por Andrzej Jaskiewicz em 2001 em seu livro intitulado "*Objective Metaheuristic Algorithms for Combinatorial Optimization*".

O primeiro trabalho proposto em metaheurística multiobjetivo foi apresentado por Schaffer em 1985 que apresenta o algoritmo genético avaliado por vetor (VEGA). O VEGA é apresentado como uma variação de um algoritmo genético de objetivo único cujo mecanismo de seleção foi alterado. Foi observado por (COELLO, 1999) que as soluções geradas pelo VEGA apresentaram, no modo geral, um fraco espalhamento uniforme na frente de Pareto. De fato, este algoritmo tende a ter um baixo desempenho na representação das regiões médias do grupo de pontos não-dominados.

(FONSECA; FLEMING, 1995) e (TAMAKI et al., 1996) propuseram uma modificação do VEGA para melhorar a deficiência de representação das regiões ótimas de Pareto.

Recentemente, os algoritmos evolucionários tem sido adaptados para resolver problemas com múltiplos objetivos conflitantes utilizando a aproximação da frente de Pareto para tais problemas. Um tutorial completo em otimização evolucionária multiobjetivo pode ser encontrado nos artigos (ZITZLER et al., 2004), (DEB, 2001) e (DEB, 2005).

O algoritmo genético opera com um grupo de soluções candidatas que são manipuladas por dois operadores básicos: a *seleção* e a *variação*. A seleção está a cargo de modelar o mecanismo de reprodução das soluções aptas, enquanto que a variação imita a capacidade natural de criar novos indivíduos utilizando a recombinação e a mutação. Dois objetivos, tem que ser levados em consideração ao desenvolver um algoritmo genético multiobjetivo: o primeiro objetivo procura guiar a busca da formação do grupo de Pareto, enquanto que o segundo objetivo mantém a diversidade do grupo de soluções não-dominadas.

O primeiro objetivo está relacionado diretamente com a seleção para reprodução, mais especificamente ao problema de atribuir valores de aptidão escalares na presença de múltiplos objetivos. O segundo objetivo preocupa-se com a seleção em geral devido à preocupação de evitar que a população seja composta por soluções idênticas. Pode-se citar também um último fator que influencia os dois objetivos anteriores que seria o elitismo, cuja responsabilidade reside na prevenção da perda de soluções não dominadas.

Em contraste com a otimização de objetivo único, onde as funções objetivo e de aptidão são diretamente relacionadas, a atribuição da aptidão e da seleção na otimização multiobjetivo deve levar em consideração todos os diferentes objetivos. Entre as diferentes estratégias de atribuição de aptidão, as que são usadas com mais frequência são aquelas baseadas em aglomeração, objetivo único e dominância de Pareto.

O primeiro método, que imita o método da soma ponderada, aglomera os objetivos em uma única função objetivo parametrizada. Tais parâmetros são sistematicamente variados durante o processo de otimização com o objetivo de encontrar um grupo de soluções não dominadas ao invés de uma única solução de troca. Para aumentar o range de busca para todas as regiões dos grupos não-dominados, vários coeficientes dos objetivos devem ser usados. (HAJELA; LIN, 1992) propuseram uma codificação dos coeficientes na descrição das soluções. Deste modo, os coeficientes evoluem como resultado dos operadores de recombinação e mutação. A aptidão compartilhada é utilizada para alcançar a diversificação dos coeficientes. Os coeficientes são utilizados em funções lineares com transformação escalar. No trabalho de (MURATA et al., 1996) propôs um método onde os coeficientes são

representados por um vetor aleatório usado para cada interação composto por uma única recombinação.

Os métodos baseados em um único objetivo realizam uma troca de objetivos durante a fase de seleção. A cada interação que um indivíduo for avaliado durante a fase de seleção, potencialmente um objetivo diferente da interação anterior será utilizado para indicar se os indivíduos são aptos ou não para realizar a reprodução.

A estratégia de dominância de Pareto (ranking) é baseada na exploração parcial da população. Alguns métodos usam um ranking de dominância, isto é, o número de indivíduos pelo qual um indivíduo é dominado, para determinar o valor de aptidão; outros métodos utilizam a intensidade de dominância, onde a população é dividida em várias frentes de Pareto e sua intensidade define a que frente o indivíduo pertence.

De forma alternativa aos dois métodos anteriores, a contagem de dominância também pode ser considerada como uma estratégia, pois leva em conta o número de indivíduos dominados por certo indivíduo. Independente da técnica utilizada, a aptidão está relacionada com toda a população, contrastando com os métodos baseados em aglomeração que calculam o valor de aptidão bruto independente dos demais indivíduos. Apenas o ranking de Pareto não garante por si só que a população irá se disseminar uniformemente sobre o grupo de pontos não dominados. De fato no caso da seleção baseada no ranking de Pareto as populações finitas geralmente convergem para um único ótimo, tal fenômeno é denominado como flutuação genética (*genetic drift*), termo este determinado na pesquisa de (GOLDBERG; SEGRAST, 1987), que implica na convergência para pequenas regiões do grupo ótimo de Pareto. Alguns autores como (SRINIVAS; DEB, 1994) e (FONSECA; FLEMING, 1995) desenvolveram pesquisas para implementar soluções capazes de minimizar o efeito da flutuação genética.

A técnica de recozimento simulado (*simulated annealing*) também tem sido empregada para lidar com problemas de otimização de combinatórios com múltiplos objetivos. O artigo de (SERAFINI, 1992) foi o primeiro a apresentar este método na forma de múltiplos objetivos. Em seu método Serafini demonstra que a implementação multiobjetivo não se difere muito da padrão utilizada para objetivo único, e que produz um grupo soluções não dominadas por nenhum outro grupo gerado pelo próprio algoritmo. A técnica do recozimento simulado no modo objetivo único determina que para cada nova solução um índice de probabilidade é atribuído dependendo do valor obtido ao ser comparado com a solução atual. O valor da probabilidade se iguala a um quando a aptidão da nova solução for melhor que a solução atual e no caso da aptidão pior um valor de probabilidade menor do que é atribuído.

Em um cenário multiobjetivo três situações podem ocorrer quando comparadas a nova solução  $s'$  com a solução atual  $s$ :  $s'$  domina  $s$ ,  $s'$  é dominada por  $s$  e as duas soluções  $s'$  e  $s$  são mutuamente não dominadas. No primeiro caso a probabilidade de aceitação é igual a um, no segundo caso a probabilidade de aceitação é menor do que um e finalmente no terceiro caso Serafine propõe regras diferentes que correspondem a algumas funções com transformação escalar ponderada, demonstrando que estas regras garantem que o algoritmo alcance uma das soluções ótimas de Pareto se a temperatura diminuir vagarosamente, porém esta técnica não garante a dispersão sobre todos os pontos do grupo não dominado. Para garantir a dispersão Serafini propõe uma modificação aleatória dos coeficientes ao longo do tempo. Seguindo a linha de pesquisa de Serafini, (ULUNGU et al., 1999) propôs regras de aceitação multiobjetivo utilizando vetores de coeficientes pré definidos, onde cada vetor está associado com um processo de recozimento independente. Como no caso do algoritmo de (SERAFINI, 1992), o algoritmo de (ULUNGU et al., 1999) um grupo potencial de soluções ótimas de Pareto contendo todas as soluções não dominadas por qualquer outra solução gerada pelo processo de recozimento. O artigo de (SUPPAPITNARM; PARKS, 1999) também implementa outro algoritmo baseado na técnica do recozimento simulado aplicando a problemas multiobjetivos.

(GANDIBLEUX et al., 1997) propõe em seu trabalho uma outra abordagem multiobjetivo utilizando a técnica de pesquisa tabu (*tabu search*). O seu método baseava-se na modificação periódica dos coeficientes a serem atribuídos nas funções objetivo com transformação escalar, e fazendo uso de listas tabu, preveniu o tour a soluções já consideradas em interações anteriores. Uma segunda lista relacionava todos os coeficientes dos vetores. Na mesma área de pesquisa (HANSEN, 1998) desenvolveu uma pesquisa por tabu multiobjetivo baseado na atribuição de coeficientes para objetivos com transformação escalar. Este método emprega algumas ideias semelhantes ao do Pareto com recozimento simulado devido ao fato deste método usar o mesmo mecanismo de dispersão das soluções atingido pela modificação apropriada dos coeficientes.

#### **4.4.5 Otimização Combinatorial Multiobjetivo Utilizando Algoritmos Genéticos**

Conforme (LEITE, 2013) Uma vertente que tem sido bastante pesquisada desde a introdução do conceito multiobjetivo utilizando algoritmos genéticos tem sido o refinamento dos dois conceitos básicos de qualquer algoritmo de busca que são: a exploração (*exploration*)

e a investigação (*exploitation*). Ambos os termos significam pesquisar, encontrar, investigar, etc, porém tem aplicações diferentes no âmbito do espaço de busca da região de aptidão. O termo exploração busca a maior dispersão possível da população dentro da região de aptidão, aumentando assim a possibilidade de exploração do maior número de regiões possíveis em busca do mínimo ou máximo global. Já a investigação utiliza uma técnica mais refinada e bem mais restrita que leva a uma pesquisa minuciosa do espaço apontado de modo a encontrar um mínimo/máximo global em meio a um ambiente com muitos mínimos/máximos locais.

#### **4.4.5.1 Algoritmo Genético de Pareto Aglomerado (NPGA)**

O NPGA foi primeiramente proposto por (HORN et al., 1993). Ao invés de outros métodos (VEGA, NSGA e o MOGA) que aplicava uma seleção proporcional de aptidão, o NPGA combina as técnicas de compartilhamento do valor de aptidão e do mecanismo de seleção via torneio. Este algoritmo não usa qualquer estratégia para manter a população dos melhores indivíduos encontrados em cada interação. No entanto, existem vários mecanismos para preservar estas soluções que são classificadas como métodos elitistas.

#### **4.4.5.2 Algoritmo Genético Baseado na Média (WBGA)**

Assim como o método da média ponderada, o algoritmo genético baseado na média também aplica, para cada função objetivo  $f_m(x)$  uma multiplicação por um coeficiente  $w_m$  tal que  $m = (1, \dots, M)$ . Esta implementação junto ao algoritmo genético foi proposta no trabalho de (HAJELA; LIN, 1992) onde cada indivíduo da população não representa apenas o valor das variáveis de decisão, mas também as médias associadas de diferentes combinações de coeficientes. Este mecanismo prevê uma solução não-dominada correspondente a combinação de uma média específica em cada interação do algoritmo genético e um grupo de várias soluções não-dominadas geradas simultaneamente pela combinação aleatória de médias.

#### 4.4.5.3 Algoritmo Genético de Ordenação Não-Dominada (NSGA-II)

Em sua pesquisa (FONSECA; FLEMING, 1993), expuseram o procedimento de ranking de soluções baseadas em um sistema de nível de dominância conhecido como NSGA que também foi abordado por (GOLDBERG, 1989) e (SRINIVAS; DEB, 1994).

Em (DEB et al, 2002) é dito que encontrar a convergência para o grupo ótimo de Pareto não é a única meta de um algoritmo evolucionário pois, manter uma boa distribuição de soluções dentro do grupo de soluções obtidas na frente de Pareto também é de grande importância para um algoritmo evolucionário. Por se tratar de um algoritmo genético multiobjetivo *Multiobjective Genetic Algorithm* (MOGA), o NSGAI utiliza um conceito baseado na dominância para realizar o ranking das soluções durante o processo de seleção. Existem dois mecanismos que visam realizar a distribuição as soluções por níveis de dominância e preservar a diversidade entre cada solução de frente não-dominada. Os demais operadores como o cruzamento e a mutação são aplicados da mesma maneira em toda a população.

O pseudo-código a seguir ilustra um método rápido de classificação de soluções em diferentes níveis dominância descrito por (DEB et al, 2002).

para cada  $p \in P$

$$S_p = \emptyset$$

$$n_p = 0$$

para cada  $q \in P$

if ( $p < q$ ) então                      Se  $p$  dominar  $q$

$S_p = S_p \cup \{q\}$                       Adicionar  $q$  ao grupo de soluções dominadas por  $p$

else if ( $q < p$ ) então

$n_p = n_p + 1$                       Incrementar o contador de dominância de  $p$

if  $n_p = 0$  então                       $p$  pertence a primeira frente de Pareto

$$p_{rank} = 1$$

$$\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$$

$$i = 1$$

Inicializar o contador de frente de Pareto

while  $\mathcal{F}_i \neq \emptyset$

$$Q = \emptyset$$

Armazenar os membros da próxima frente de Pareto

for each  $p \in \mathcal{F}_i$

```

for each  $q \in S_p$ 
   $n_q = n_q - 1$ 
  if  $n_q = 0$  então            $q$  pertence a próxima frente de Pareto
     $q_{rank} = i + 1$ 
     $Q = Q \cup \{q\}$ 
 $i = i + 1$ 
 $\mathcal{F}_i = Q$ 

```

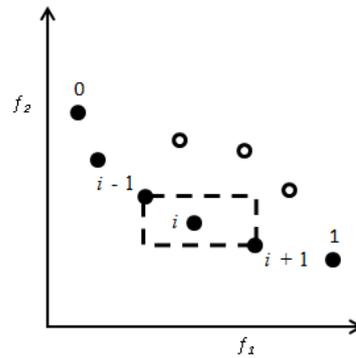
O algoritmo inicia na interação  $t = 0$  com a geração aleatória de uma população  $P_0$ , e tamanho  $N$ . Nesta população muitas frentes de Pareto não-dominadas são identificadas e as soluções são ordenadas em diferentes níveis de dominância (ranking). O valor de aptidão de cada solução atribuída é igual ao seu nível de dominância, iniciando com 1 para as soluções da primeira frente (que representam as soluções não dominadas da população). Este método é conhecido como sistema não-dominado ou ordenação não dominada.

No NSGA, a diversidade das soluções era mantida através do índice de compartilhamento  $\sigma_{share}$ , porém este método apresentava dois problemas eminentes:

1. Manter a diversidade das soluções (espalhamento) dependia do valor de  $\sigma_{share}$  determinado pelo usuário;
2. Devido ao fato de cada solução ser comparada com todas as outras soluções da população, o método da função  $\sigma_{share}$  empregava uma complexidade computacional de  $O(N^2)$ .

Para resolver o problema de diversidade de solução foi desenvolvida uma função que calcula a distância média entre dois pontos que representam as soluções próximas de uma determinada solução denominada  $i_{distance}$ . O valor determinado pelo  $i_{distance}$  serve para estimar o perímetro de um cubóide formado pelas soluções mais próximas. Tais soluções representam os vértices do cubóide representados na figura 4.7 pelos pontos no retângulo vazado e a distância de multidão (*crowding distance*) é representado pela sua aresta.

Figura 4.7: Representação gráfica da distância de multidão.



FONTE: Deb et al, 2002.

O código a seguir mostra como a distância de multidão é calculada:

$l =  \mathbb{I} $	Número de soluções em $\mathbb{I}$
<i>for each</i> $i$ , <i>set</i> $\mathbb{I}[i]_{distance} = 0$	Inicializar a distância
<i>for each objective</i> $m$	
$\mathbb{I} = \text{sort}(\mathbb{I}, m)$	Ordenar usando o valor de cada objetivo
$\mathbb{I}[1]_{distance} = \mathbb{I}[l]_{distance} = \infty$	De modo que os limites sejam selecionados
<i>for</i> $i = 2$ até $(l - 1)$	Para os demais pontos
$\mathbb{I}[i]_{distance} = \mathbb{I}[i]_{distance} + (\mathbb{I}[i + 1].m - \mathbb{I}[i - 1].m) / (f_m^{max} - f_m^{min})$	

A abordagem do NSGA-II aplica o operador de comparação de multidão ( $<_n$ ) para direcionar as soluções para uma distribuição uniforme na frente de Pareto. Este operador assume que cada solução  $i$  possui dois atributos:

1. Rank de não dominância ( $i_{rank}$ );
2. Distância de multidão ( $i_{distance}$ ).

De posse da descrição destes atributos o operador de comparação ordena as soluções seguindo a seguinte definição:

$$i <_n j$$

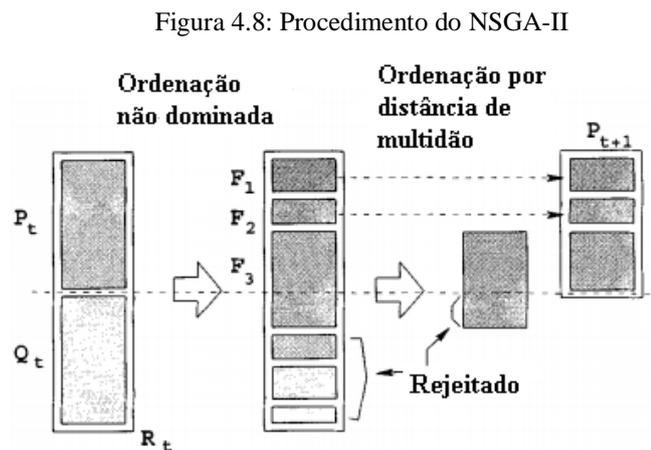
$$\text{if}(i_{rank} < j_{rank}) \text{ ou } ((i_{rank} = j_{rank}) \text{ e } (i_{distance} > j_{distance}))$$

O operador  $\prec_n$  compara duas soluções  $i$  e  $j$ . A solução que apresentar o menor rank (maior dominância) é escolhida para compor a frente de Pareto ou para o caso dos ranks das duas soluções serem iguais, a solução que apresentar maior distância de multidão é escolhida.

Depois de definidos os métodos de ordenação e de diversificação é possível apresentar o laço principal do código proposto por (DEB et al, 2002):

$R_t = P_t \cup Q_t$	Combinar os pais e gera a nova população
$\mathcal{F} = \text{fast non dominated sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$ , frentes não dominadas de $R_t$
$P_{t+1} = \emptyset$ and $i = 1$	
until $ P_{t+1}  +  \mathcal{F}_i  \leq N$	Até que a população dos pais seja preenchida
crowding distance assignment ( $\mathcal{F}_i$ )	Calcular a distância de multidão $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	Incluir a $i$ -ésima frente de Pareto não dominada
$i = i + 1$	Verificar a próxima frente para inclusão
sort( $\mathcal{F}_i, \prec_n$ )	Ordenar na forma decrescente usando $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1:(N -  P_{t+1} )]$	Escolher o primeiro $(N -  P_{t+1} )$ elemento de $\mathcal{F}_i$
$Q_{t+1} = \text{make new pop}(P_{t+1})$	Criar a nova população $Q_{t+1}$
$t = t + 1$	Incrementa o contador de geração

A figura 4.8 ilustra o processo realizado pelo código do NSGA-II:



FONTE: Deb et al, 2002.

O algoritmo realiza um procedimento comum a todas as interações até que a condição de parada seja alcançada. Tal procedimento está dividido nos seguintes passos para cada interação  $t$ :

1. Um mecanismo de seleção por torneio é aplicado baseado no valor de aptidão e na distância de multidão (*crowd distance*), cujo objetivo é o de preservar a diversidade da população. A solução  $i$  ganha o torneio se: a mesma possuir uma melhor classificação (em termos da ordem da frente de Pareto não dominada que a solução  $i$  pertence) em relação a solução  $j$  ou ambas tem a mesma classificação e a solução  $i$  possuir um distância de multidão maior do que a solução  $j$ . A distância de multidão é utilizada para estimar a densidade das soluções na vizinhança de uma solução  $i$ .
2. Os operadores genéticos e mutação são aplicados para construir uma população de descendentes  $Q_t$ , de tamanho  $N$  que foi combinada com a população  $P_t$  para formar a terceira população  $R_t$  composta por pais e filhos.
3. As soluções resultantes da população  $R_t$  são classificadas de acordo com o nível de dominância para a atribuição das diferentes frentes de Pareto não-dominadas.
4. Uma nova população  $P_{t+1}$  é selecionada da população das soluções  $R_t$ . Como a nova população  $P_{t+1}$  representa a metade de  $R_t$  nem todas as soluções podem ser adicionadas às frentes de Pareto. Ou seja, as piores frentes serão descartadas. Se não houver mais a possibilidade de adicionar mais soluções um mecanismo baseado na distância de multidão é aplicado para encontrar a melhor solução.
5. Se a interação atual for menor que o número de gerações pré-definidas então, um incremento à interação atual  $t = t + 1$  é realizado e o algoritmo retorna ao passo 1 se não o software para.

#### 4.4.5.4 Algoritmo Evolucionário Baseado na "Força" de Pareto (SPEA)

O MOEA conhecido como SPEA foi inicialmente proposto por (ZITZLER; THIELE, 1998). Zitzler utiliza a técnica de manter uma população externa de tamanho fixo que armazena o valor das soluções não- dominadas assim como um buffer ou uma memória volátil. Em cada interação, as novas soluções encontradas são comparadas com o valor das previamente armazenadas na população externa e os resultados são armazenados na mesma memória. Como o tamanho da memória é limitada, o algoritmo utiliza uma técnica de

agrupamento conhecida como clusterização que preserva as soluções mais esparsas. As soluções não-dominadas conhecidas como “elite” também são utilizadas como referência na comparação da população externa utilizando operadores genéticos de forma a influenciar a população a se espalhar por regiões mais específicas dentro do espaço de busca. Poucos anos mais tarde, após a introdução do SPEA, (ZITZLER et al., 2001) apresentam uma versão melhorada que ficou conhecida como SPEA II cujas principais diferenças foram:

- Para determinar o valor de aptidão da solução, o método de cálculo é baseado no número de soluções que a dominam e no número de soluções dominadas por ela.
- Se for necessário, no caso da classificação das soluções com características de dominância idênticas, o método de cálculo de aptidão adiciona uma nova técnica de estimação da densidade das soluções naquela região do espaço de busca.
- Como o tamanho da população externa é constante, é possível que não haja soluções não-dominadas suficientes para preenchê-la, com isso a população externa também pode ser composta por soluções dominadas.
- Apenas a população externa participa no processo de seleção.

## CAPÍTULO V

### IMPLEMENTAÇÃO DA TÉCNICA PROPOSTA UTILIZANDO A FERRAMENTA DE ALGORITMO EVOLUTIVO – NSGA II

Neste capítulo são apresentadas as soluções para cada problema de otimização existente na máquina de montagem SMD GXH-3. A aplicação do fabricante desta máquina emprega uma metaheurística baseada em técnicas clássicas de resolução de problemas de números inteiros comumente conhecidos como: *branch & bound*, o algoritmo ganancioso (*greedy*) e a pesquisa local. A relação entre estas técnicas ocorre de modo que os dados obtidos por uma aplicação serve de condição inicial para a próxima técnica. Isto se dá devido a inter relação entre os problemas de arranjo de alimentadores e a sequência de montagem. Em meio a este cenário uma abordagem multicritério pode reduzir o tempo computacional e melhorar a pesquisa no espaço de soluções ótimas haja vista que as técnicas clássicas dependem muito da condição inicial e dos limites impostos.

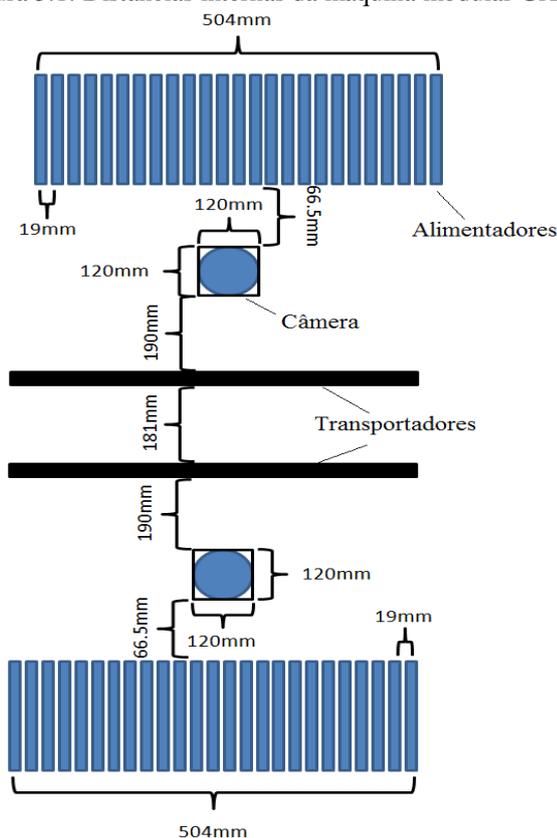
No contexto das técnicas computacionais aplicadas para a resolução de problemas de máquinas de montagem SMD a utilização de algoritmos evolucionários tem sido empregada porém, apenas na condição de mono objetivo para otimizar a atribuição dos alimentadores.

Antes de se aplicar o procedimento de otimização, um parâmetro muito importante ainda falta ser apresentado, o qual se refere às distâncias internas da máquina de montagem GXH-3. Durante o processo de introdução de um novo modelo apenas as coordenadas de montagem da placa são passadas com referência a um ponto na placa.

Como o software do fabricante já possui estas informações não é necessário obtê-las. Mas para o caso de uma otimização multicritério isto é necessário, pois as distâncias entre a posição de repouso da cabeça dos alimentadores, dos alimentadores para a câmera de reconhecimento de componentes e da câmera para a posição de montagem da placa descrevem a distância percorrida do processo de montagem da máquina.

A figura 5.1 ilustra as distâncias internas da máquina. As medidas foram realizadas *in loco* em uma máquina que faz parte de um processo fabril onde o procedimento de otimização aqui proposto foi aplicado.

Figura 5.1: Distâncias internas da máquina modular GXH-3.



FONTE: Autor, 2015

### 5.1. Solução do Problema da Rota de Montagem dos Componentes

De acordo com a restrição descrita no capítulo três onde os componentes devem ser montados seguindo a ordem crescente de altura, ou seja, o componente de menor altura deve ser montado primeiro, todos os componentes foram agrupados para compor um sequenciamento ordenado para atender a esta especificação. A tabela 5.1, a seguir mostra como os componentes para o caso 1 foram ordenados. Como pode ser observado os componentes foram ordenados levando em consideração a altura crescente dos componentes e a quantidade do tipo de componente em ordem decrescente.

Tabela 5.1: Ordenação dos componentes pela sua altura (placa caso 1)

<b>Tipo</b>	<b>Quantidades por Tipo</b>	<b>Altura em Ordem Crescente (mm)</b>
4	14	0.35
2	13	0.35
5	4	0.35
3	2	0.35
1	2	0.35
6	1	0.35
7	1	0.45
8	38	0.5
10	3	0.5
9	2	0.5
11	2	0.5
13	2	0.8
12	1	0.8
14	1	0.8
15	1	0.8
16	1	1.25

Fonte: Autor, 2015.

De acordo com a especificação do tamanho dos componentes é possível determinar o *feeder* a ser utilizado e o *nozzle* apropriado para manipular o tipo de componente  $c$  durante os ciclos de *pick* e *placement*. A tabela 2 na seção de anexos apresenta a organização do tipo de componente pela sua especificação e coordenada de montagem do caso 1 estudado.

Como a abordagem da rota de montagem dos componentes deve obedecer ao critério da altura, o problema passa a ser considerado como um problema de caminho mais curto com rótulos onde o algoritmo de Dijkstra disponível no site da *Mathworks* pode ser aplicado de forma a encontrar esta sequência. Sejam os nós  $s$  e  $t$  considerados respectivamente como os nós de fonte e destino, ou seja, as coordenadas de montagem são tratadas como vértices de gráfico onde um caminho é traçado entre a coordenada inicial e a coordenada final de cada ciclo de montagem. Atribui-se um custo como sendo a distância entre as coordenadas com a mesma rotulação que no caso é atribuída a altura do componente ao rótulo da coordenada, ou seja,  $\pi_s \leq \pi_i \leq \pi_t$ . Desta forma é possível garantir que sempre o componente de menor altura será montado primeiro.

O problema que envolve o menor caminho, (*Shortest Path Problem* - SPP) com rótulos é considerado um método ganancioso (*greedy*) que nem sempre garante o ótimo global, pois a busca é guiada pelas distâncias e restrita pelos rótulos. A abordagem sem rótulos é a mais

apropriada para encontrar as rotas mais curtas, porém não garante que o componente de menor altura será montado primeiro. Considerando  $i$  e  $j$  como as posições de montagem com os rótulos  $\pi_i$  e  $\pi_j$  respectivamente é possível descrever as variáveis do problema como:

$\pi_i$  = rótulo atribuído à primeira coordenada da rota de montagem, ou seja, altura do primeiro componente do caminho;

$\pi_j$  = rótulo atribuído à próxima coordenada da rota de montagem, ou seja, altura do próximo componente do caminho;

$d_{ij}$  = distância entre os coordenadas  $i$  e  $j$ , análogo ao custo no problema clássico do caminho mais curto;

$x_{ij}$  = variável utilizada para definir a ordenação das coordenadas de montagem;

$x_{ji}$  = variável complementar à variável  $x_{ij}$ .

Então para a formulação do problema do caminho mais curto para definição da rota de montagem dos componentes consideram-se cada componente  $i$  e  $j$  como sendo dois conjuntos distintos de vértices onde  $v_i$  é pertence à  $P = (v_1, v_2, \dots, v_n) \in V$  e  $v_j$  pertence à  $W = (v_1, v_2, \dots, v_n) \in W$ , o caminho que une dois vértices adjacentes é a aresta  $e$  que representa a distância  $d_{ij}$  e representa a distância entre os vértices  $v_i$  e  $v_j$ , então o problema de rota pode ser formulado como:

$$\text{minimizar } s = \sum_{i=1}^{n-1} d_{ij} x_{ij} \quad 5.1$$

sujeito à,

$$\pi_i \leq \pi_j \text{ para todo } i, j = 1, 2, \dots, n \quad 5.2$$

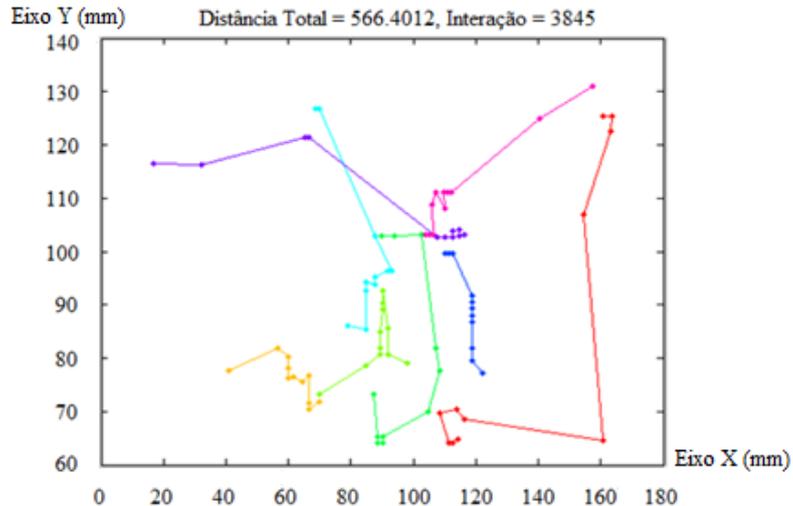
$$\sum_{j=1}^{n-1} (x_{ij} - x_{ji}) = \begin{cases} 1 & i = s \text{ ou } j = t \\ 0 & i \neq s, t \end{cases} \quad 5.3$$

$$x_{ij} \geq 0 .$$

A equação 5.1 determina o caminho mais curto a ser seguido pela cabeça de montagem. A restrição 5.2 garante que os componentes serão montados em altura crescente e a restrição determina a ordenação que as coordenadas devem seguir.

Ao ser empregado o método do caminho mais curto, a rota dos componentes na seqüência de alturas desejadas é encontrada conforme ilustrado na figura 5.2, a seguir.

Figura 5.2: Rota de montagem de acordo com a restrição das alturas dos componentes.



FONTE: Autor, 2015.

Cada cor representa um tour ou caminho no gráfico de rede, ao qual obedece a ordenação definida pela restrição da altura do componente e pela quantidade de componentes montados pela cabeça a cada ciclo de montagem. Como a cabeça de montagem pode pegar e montar até 12 componentes por vez, o número máximo de pontos visitados por ciclo de cabeça também fica delimitado a 12.

## 5.2. Solução do Problema de Atribuição de *Feeder*

De acordo com o trabalho demonstrado por (PARK; KIM, 2010) a modelagem do problema de máquinas de montagem modulares pode ser descrita como um problema de programação linear de números inteiros, assim sendo para uma linha com  $n$  montadoras modulares, é possível definir o conjunto de módulos ( $M$ ), alocações de alimentadores ( $L$ ), posições de montagem ( $P$ ) e tipos de componente ( $C$ ) como sendo:

$$M = \{1, \dots, n_m\}$$

$$L = \{1, \dots, n_l\}$$

$$P = \{1, \dots, n_p\}$$

$$C = \{1, \dots, n_c\}$$

O problema de otimização pode ser dividido em dois subproblemas: o problema de arranjo de alimentadores (*Feeder Arrangement Problem* - FAP) onde a atribuição do alimentador (*feeder*) a uma posição é feita de tal forma que a distância percorrida pela cabeça de montagem durante o ciclo de aquisição (*pick*) e montagem (*placement*) seja mínima e o problema da seqüência de montagem (*Mounting Sequence Problem* - MSP) onde a distância percorrida pela cabeça da última posição de montagem para o próximo alimentador no próximo ciclo de montagem seja mínima.

O FAP pode representado com a seguinte notação:

$$T_f^m = \sum_{c=1}^{n_c} \sum_{l=1}^{n_l} \sum_{j=1}^{n_p} t_f^m(l, j) x_{cl} \quad 5.3$$

onde,

$T_f^m$  = tempo total que a cabeça de montagem leva para pegar um componente do tipo  $c \in C$  de um *feeder*  $l \in L$  e montá-lo na posição  $j \in P$ ;

$n_c$  = número de tipos de componente  $c$ ;

$n_l$  = número de alocações de alimentadores  $l$ ;

$n_p$  = número de posição de montagens do componente  $j$ ;

$L^m$  = grupo de alocações de alimentadores  $l$  pertencentes ao módulo  $m$ ;

$x_{cl}$  = variável binária da alocação do tipo de componente  $c$  na posição do alimentador  $l$ , ou seja, a variável assume o valor 1 se o componente  $c$  for alocado na posição do alimentador  $l$ , 0 se não.

Desta forma a otimização do arranjo de *feeder* se dá pela:

$$\min\{\max T_f^m\} \quad 5.4$$

$$\sum_{c \in C} x_{cl} \leq 1, \forall l \in L \quad 5.5$$

$$\sum_{l \in L} x_{cl} = 1, \forall c \in C \quad 5.6$$

$$\sum_{c \in C} \sum_{l \in L^m} x_{cl} \leq |L^m|, \forall m \in M \quad 5.7$$

a restrição 3.19 garante que cada locação de alimentadores pode conter no máximo um alimentador, a restrição 3.20 garante que cada tipo de componente está armazenado em um alimentador apenas e a restrição 3.21 garante que cada módulo possui pelo menos  $L^m$  alocações de *feeder*.

A solução gerada para o problema de atribuição de alimentadores é uma matriz  $n_c \times n_l$  onde a variável  $x_{cl} \in \{0,1\}^{n_c \times n_l}$  determina que o tipo de componente  $c$  alocado no *feeder*  $l$  que reduz o movimento da cabeça durante o processo de *pickup*.

### 5.3. Solução do Problema da Seqüência de Montagem

Como demonstrado na pesquisa de *Ho e Ji (2009)* a modelagem do problema de seqüência de montagem é muito semelhante ao problema do caixeiro viajante, porém o vendedor não visita todas as cidades (tour completo) e retorna para a cidade de origem, pois a cabeça de montagem (vendedor) tem um limite máximo de componentes a serem montados por ciclo. Na abordagem feita, o problema da rota de montagem é resolvido e seu resultado é utilizado como dados de entrada para realizar a seqüência de montagem, ou seja, o problema da seqüência é resolvido com o intuito de minimizar as seguintes distâncias: a distância percorrida pela cabeça entre os alimentadores (*feeders*) durante o ciclo de aquisição (*pickup*), a distância do último *feeder* para a câmera de reconhecimento da imagem do *pickup*, a distância da câmera de reconhecimento para a primeira posição de montagem  $i$  e a distância da última posição de montagem  $j$  para o primeiro *feeder* do próximo ciclo de aquisição.

O problema da seqüência de montagem (*Mounting Sequence Problem – MSP*) pode ser definido pela seguinte formulação:

$$T_b^m = \sum_{i=1}^{n_p} \sum_{l=1}^{n_l} t_b^m(i, l) y_{il} \quad 5.8$$

onde,

$T_b^m$  = tempo total que a cabeça de montagem leva para se movimentar da posição de montagem do último componente da rota para o primeiro alimentador do próximo ciclo de aquisição (*pickup*);

$n_p$  = número de posição de montagens do componente  $i$ ;

$n_l$  = número de alocações de alimentadores  $l$ ;

$P_{\xi(l)}$  = rota de montagem definida por tipo de componentes para o grupo de alimentadores  $l$  do módulo  $m$ ;

$S^m$  = grupo de posições de montagens definidas para o módulo  $m$ ;

$x_{\tau(i)l}^*$  = variável binária cuja função  $\tau(i)$  define previamente a alocação dos tipos de componentes para as posições de alimentadores  $l$ ;

$y_{il}$  = variável binária da posição de montagem  $i$  na posição do alimentador  $l$ , ou seja, a variável assume o valor 1 se a posição de montagem  $i$  for alocado na posição do alimentador  $l$ , 0 se não.

Assim sendo, o problema de otimização da seqüência de montagem é descrito como:

$$\min T_b^m, \forall m \in M \quad 5.9$$

$$\sum_{l=1}^{n_l} y_{il} = 1, \forall i \in P^m \quad 5.10$$

$$\sum_{i=1}^{n_p} y_{il} = |P_{\xi(l)}|, \forall l \in L^m \quad 5.11$$

$$\sum_{l=1}^{n_l} \sum_{i=1}^{s^m \in n_p} x_{\tau(i)l}^* y_{il} \leq |S^m| - 1 \quad 5.12$$

a restrição 3.24 garante que cada *feeder* contém apenas 1 tipo de componente, a restrição 3.25 garante que apenas um componente pertencente a rota de montagem pode ser montado onde  $P_{\xi(l)}$  é definida como a função que determina o tipo de componente para cada coordenada de montagem e a restrição 3.26 indica que a quantidade de tipos de componentes  $i$  do subgrupo de posições de montagem pertencente ao módulo  $S^m$  está armazenado no *feeder*  $l$  pertencente ao módulo é inferior à quantidade de posições de montagem do módulo em  $|S^m| - 1$  pois o último ciclo de *pickup* da placa não é considerado.

De forma a garantir o menor deslocamento da cabeça de montagem no eixo X, um ponto médio entre as coordenadas deste eixo foi determinado como ponto de início e término do ciclo de montagem. Deste modo é possível determinar uma condição inicial para a sequência de montagem a fim de oferecer um parâmetro inicial ótimo para resolução do problema de atribuição que depende intrinsecamente da solução do problema da sequência de montagem. Assim como na rota de montagem, as cores representam um ciclo de montagem. Já com a rota de montagem pré-definida, a sequência de montagem está focada em reduzir a distância entre a câmera de inspeção e o primeiro componente do ciclo de montagem e entre o último componente do ciclo e o primeiro alimentador do próximo ciclo.

A figura 5.3 apresenta cada ciclo de montagem, considerando cada ponto do gráfico uma posição de montagem. Este gráfico da figura 5.3 representa o ciclo de montagem de apenas uma cabeça de montagem, sendo a máquina GXH-3 composta de quatro cabeças de montagem (duas cabeças por módulo), um balanceamento é determinado para que todas as cabeças tenham uma carga de componentes igual ou aproximada.

Desta forma a solução gerada para o problema de sequência de montagem é uma matriz  $n_p \times n_l$  onde a variável  $(y_{il} \in \{0,1\}^{n_p \times n_l})$  determina que o tipo de componente  $c$  alocado no *feeder*  $l$  pode ser montado na coordenada  $p$ .

#### 5.4. A Resolução do Problema Aplicando o NSGA-II

Toda e qualquer otimização baseada em algoritmos genéticos depende principalmente da representação das soluções através dos cromossomos. Assim como na teoria evolutiva, onde o indivíduo mais apto ao meio sobrevive, no algoritmo os indivíduos com os melhores valores de aptidão são selecionados. Este resultado da otimização depende da interpretação do valor gerado pela representação da solução na forma do cromossomo. Nesta dissertação o cromossomo representa parcialmente a atribuição do *feeder* e a sequência de montagem

avaliando a distância percorrida pela cabeça de montagem. Como demonstrado no capítulo 3 a solução do problema de atribuição de *feeder* gera uma matriz  $n_c \times n_L$ , desta forma o cromossomo é composto pela posição do gene que representa a posição do *feeder* e o seu valor representa o tipo de componente. Assim como a solução da seqüência de montagem gera uma matriz  $n_p \times n_L$ , cujo gene representa a posição do *feeder* e o seu valor representa a posição de montagem.

Como exposto no capítulo 3 o primeiro ciclo de montagem é atípico se comparado com os demais ciclos, devido ao fato de no primeiro ciclo ocorrer o reconhecimento das marcas de fiducial (procedimento utilizado para correção das coordenadas X e Y das posições de montagem devido ao deslocamento da PCI durante o seu posicionamento). Apenas uma cabeça de montagem por módulo realiza este procedimento antes de iniciar o seu processo de *placement*. Como padrão, a máquina de montagem GXH-3 utiliza o reconhecimento de dois pontos em diagonal, e desta forma a primeira seqüência de montagem é definida pelo componente mais próximo da segunda posição de fiducial. Desta forma o primeiro ciclo foi minimizado como:

$$\text{minimizar } f = \sqrt{\left((x_{f2} - x_i)^2 + (y_{f2} - y_i)^2\right)}. \quad 5.13$$

Este procedimento é realizado para cada cabeça após a definição da rota de montagem através da resolução do problema do caminho mais curto.

A definição dos componentes por cabeça é determinada pela distância da primeira posição de montagem para a câmera de reconhecimento de componentes. Como cada cabeça possui uma câmera e as rotas de montagem estão previamente definidas resta definir a distribuição das seqüências de montagem para as cabeças de montagem com o intuito de balancear os ciclos. Sendo assim, o ponto médio da placa com relação à largura da mesma é encontrado tomando como referência o seu valor e dividi-lo por dois, isto é,  $P_m = W/2$  onde  $W$  representa a largura da placa.

Este procedimento aplica uma penalidade para os valores da coordenada Y acima do valor médio, sendo estes componentes atribuídos para o *rack* de *feeder* da próxima cabeça do mesmo módulo. A única exceção à esta atribuição ocorre no primeiro ciclo onde não importa se a coordenada da posição de montagem possui valor maior que o ponto médio e sim se é a mais próxima da coordenada da segunda marca de fiducial. Para reduzir o movimento do eixo

X durante a atribuição do *feeder* do componente a ser montado deve apresentar a menor diferença possível, porém, após o processo de *pickup* (aquisição dos componentes pela cabeça de montagem) a cabeça deve mover-se para a câmera de reconhecimento de componentes. Então para garantir a menor diferença no eixo X durante o *pickup* a seguinte inequação deve ser obedecida:

$$d_{l,l} + d_{l,CAM} \leq d_{j,l}, \quad 5.14$$

onde:

$d_{l,l}$  = distância entre o primeiro *feeder* e o último *feeder* do ciclo de *pickup*;

$d_{l,CAM}$  = distância entre o último *feeder* do ciclo de *pickup* e a câmera de reconhecimento;

$d_{j,l}$  = distância entre o último componente da rota de montagem para a primeira posição do *feeder* do ciclo de *pickup*.

A definição dada pela equação 5.14 é aplicada para construir os indivíduos da população inicial de modo que não haja violações onde um tipo de componente atribuído a um *rack* de *feeder* do módulo de entrada não entre na sequência de montagem do módulo de saída e vice-versa.

#### 5.4.1 Codificação do Problema

O comprimento do cromossomo é igual ao número máximo de posições disponíveis no *rack* de *feeder* de cada cabeça de montagem, ou seja, o cromossomo é composto por 25 genes. Cada gene é candidato a receber um tipo de componente, porém, as suas dimensões podem reduzir a quantidade de genes disponíveis, pois para componentes de dimensões maiores a largura do *feeder* tende a aumentar na mesma proporção, ou seja, quanto maior for o componente, maior será a largura do *feeder* o que faz com que este mesmo *feeder* chegue a ocupar mais de uma posição de alimentador disponível. A figura 5.3 mostra um exemplo desta codificação:

Figura 5.3: Codificação do gene para o problema de atribuição.

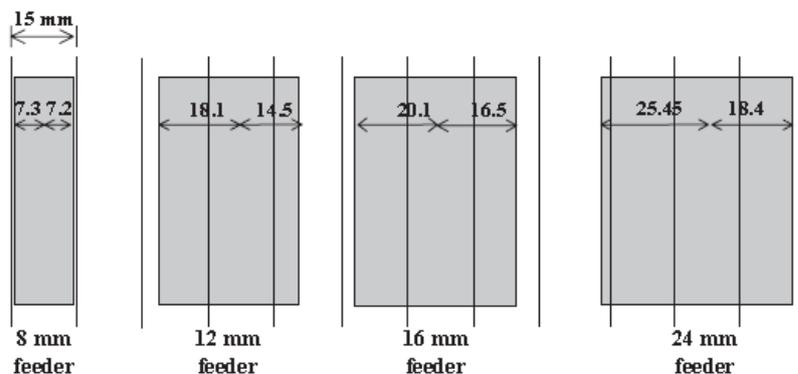
0	0	0	0	0	0	0	0	-1	3	-1	2	1	5	4	-1	8	0	0	0	0	0	0	0	0	0
$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$	$l_8$	$l_9$	$l_{10}$	$l_{11}$	$l_{12}$	$l_{13}$	$l_{14}$	$l_{15}$	$l_{16}$	$l_{17}$	$l_{18}$	$l_{19}$	$l_{20}$	$l_{21}$	$l_{22}$	$l_{23}$	$l_{24}$	$l_{25}$	

FONTE: Autor, 2015.

Onde 0 significa posição disponível,  $-1$  representa posição ocupada por *feeder* adjacente e qualquer número  $> 0$  representa o tipo de componente  $c$  atribuído à posição no *rack* de *feeder*.

No exemplo da figura 5.4 a codificação do cromossomo é feita para o problema de atribuição de *feeder*. No caso estudado por (SUN et al, 2005), devido às características construtivas dos *feeders*, é possível que o mesmo tenha uma largura assimétrica que pode ocupar mais espaços disponíveis em um lado do que em outro, a figura 5.4 apresenta um esquema de associação de diferentes tipos de alimentadores sendo que as linhas verticais representam a largura de um espaço disponível e os retângulos representam a largura e disposição dos alimentadores.

Figura 5.4: Arranjo de alimentadores.



FONTE: Sun et al, 2005.

Para o caso da codificação do cromossomo para o problema do TSP é considerado da mesma forma, porém o tipo de componente que é atribuído à posição no *rack* de *feeder* deve conter o tipo de componente da primeira posição de montagem e o tipo de componente da última posição de montagem de um ciclo. Por exemplo, considere que o componente tipo 4 e tipo 6 pertencem à primeira e à última posição de montagem do ciclo e assim por diante até que todos os tipos de componente das sequências de montagem sejam atribuídos para formar um cromossomo assim como mostrado na figura 5.5.

Figura 5.5: Codificação do gene para o problema da rota de montagem.

0	0	0	0	0	0	0	0	0	0	0	0	0	6	4	0	0	0	0	0	0	0	0	0	0	0
$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$	$l_8$	$l_9$	$l_{10}$	$l_{11}$	$l_{12}$	$l_{13}$	$l_{14}$	$l_{15}$	$l_{16}$	$l_{17}$	$l_{18}$	$l_{19}$	$l_{20}$	$l_{21}$	$l_{22}$	$l_{23}$	$l_{24}$	$l_{25}$	

FONTE: Autor, 2015.

### 5.4.2 Seleção e cruzamento

De acordo com descrito no código de (SONG, 2014) o método empregado na seleção dos indivíduos foi o torneio.

O procedimento da seleção por torneio segue o seguinte pseudo-código:

1. Definir a quantidade de elementos que participarão do torneio ao atribuir o número  $t = \{1, 2, \dots, n\}$ ;
2. Escolher o indivíduo de melhor aptidão resultante do torneio;
3. Laço: Realizar o torneio até que todos os indivíduos passem pelo torneio;
4. Resultado: A nova população resultante que venceu o torneio.

Como descrito por (KUMAR et al., 2012) para o operador de cruzamento existem três métodos conhecidos: cruzamento de código parcial (PMX), cruzamento cíclico (CX) e cruzamento ordenado (OX). O algoritmo desenvolvido emprega o PMX, onde dois pontos intermediários do cromossomo de cada pai são inteiramente trocados para obter a próxima geração.

Desta forma aplicando os operadores de seleção por torneio e cruzamento por PMX temos o seguinte pseudo-código:

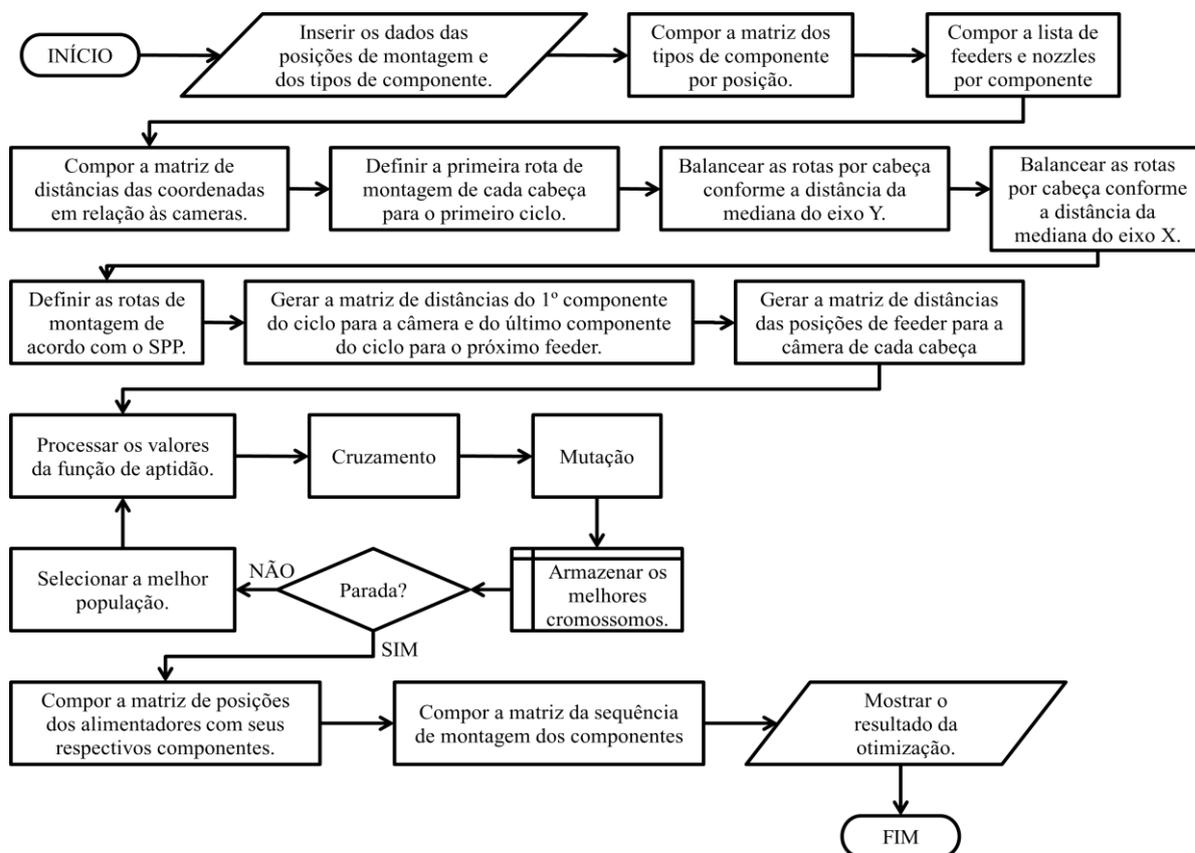
1. Iniciar;
2. Geração da população inicial utilizando a função de permutação randômica (*randperm*);
3. Crie um contador ( $K=1$ );
4. Enquanto  $K! < 100$  faça;
  - 4.1 Avaliar a aptidão de cada cromossomo usando a função de aptidão no qual o peso de cada posição de montagem seja somado;
  - 4.2 Selecionar o indivíduo através do método de torneio;
  - 4.3 Aplicar o cruzamento PMX com a probabilidade de geração  $P_C = 1$ .

Após a definição da codificação do gene, do método de seleção e cruzamento, o operador “mutação” introduz uma modificação do código genético da nova geração tornando

possível explorar soluções locais para efetivamente encontrar a solução global. Este operador seleciona aleatoriamente indivíduos desta nova população e aplica a mutação a uma probabilidade  $P_M = 0.01$ , em outras palavras, apenas 1% da população fica sujeita a sofrer uma mutação (mudança). No caso apresentado nesta dissertação, o conteúdo do gene representa o tipo de componente e sua posição representa a posição do *feeder* no rack de *feeders*. Em outras palavras, ao aplicar o operador “mutação” não somente uma posição de *feeder* é trocada, mas também o seu conteúdo que representa o tipo de componente.

O fluxograma mostrado na figura 5.6 descreve o funcionamento passo a passo do algoritmo de otimização multicritério, porém, é necessário definir os parâmetros iniciais. Primeiramente é necessário dar informações sobre o componente como: a coordenada XY dos componentes por tipo, a quantidade total de componentes por tipo, o tamanho do componente e o tamanho da PCI. Os dados de coordenadas XY e a quantidade total de componentes por tipo definem a matriz de distâncias. O tamanho dos componentes define o tipo de *nozzle* que deve ser usado assim como o tipo de *feeder*. O tamanho dos componentes também é necessário para definir o rótulo dos mesmos para a aplicação da rota do caminho mais curto para cada ciclo de montagem. Após a definição das rotas de montagem o primeiro ciclo de cada cabeça é definido pela menor distância entre uma posição de montagem e a posição da leitura da segunda marca de fiducial, ou seja, a rota de montagem do primeiro ciclo de cada cabeça é definida pela menor distância entre a posição da leitura de fiducial e a posição do componente mais próximo. Em seguida com base na informação das coordenadas dos componentes são definidos dois pontos de mediana, o primeiro no eixo X e o segundo no eixo Y, visando um balanceamento de montagem dos componentes pelas cabeças. As coordenadas de cada eixo então são subtraídas em relação à sua respectiva mediana. Se o valor for negativo, para o caso do eixo X, o componente é atribuído ao ciclo de montagem do módulo de entrada (*in*) e se o valor for positivo, o componente é atribuído para o módulo de saída (*out*). Para o caso do eixo Y, é feita a mesma operação, o valor negativo indica que o componente pertence à sequência de montagem da cabeça frontal (*front*) e o valor positivo indica que o componente pertence à sequência de montagem da cabeça traseira (*rear*).

Figura 5.6: Fluxograma do algoritmo principal.



FONTE: Autor, 2015.

## 5.5. O Código de Aplicação da Ferramenta Multicritério

O código que aplica o NSGA II foi baseado no código disponibilizado por (SONG, 2014). Ele é composto pelos arquivos *callOutputfuns.m*, *crossoverOp.m*, *evaluate.m*, *extractPop.m*, *initpop.m*, *loadpopfile.m*, *mutationOp.m*, *ndsort.m*, *nsga2.m*, *nsgaopt.m*, *output2file.m*, *plotnsga.m*, *selectOp.m*, *statpop.m*, *varlimit.m*, *verifyOpt.m*, *TP\_CONSTR.m* e *TP\_CONSTR\_objfun.m*.

Os arquivos principais que foram modificados *TP\_CONSTR.m* e *TP\_CONSTR\_objfun*, descrevem como as variáveis devem ser declaradas e como as funções objetivo e as restrições devem ser inseridas.

O primeiro arquivo *TP\_CONSTR.m* contém a seguinte estruturação:

**clc; clear; close all**

```

options = nsgaopt( );           % cria a estrutura de opções padrão.

options.popsiz = ;           % tamanho da população,

options.maxGen = ;           % número máximo de gerações (critério de parada).

options.numObj = ;           % número de funções objetivo.

options.numVar = ;           % número das variáveis do problema de otimização.

options.numCons = ;         % número de restrições.

options.lb = [ ];           % limite inferior de x.

options.ub = [ ];           % limite superior de x.

options.objfun = @objfun;    % operador da função objetivo

result = nsga2(options);     % inicia o processo de otimização.

```

Já o segundo arquivo contém a definição das funções objetivo que serão utilizadas pela função @objfun no processo de otimização. O arquivo *TP\_CONSTR\_objfun.m*, contém um exemplo para a resolução de um problema contendo restrições e segue a seguinte estrutura:

```

function [y, cons] = objfun(x) % define os parâmetros para o nsgaopt ( ) usado na
                                % otimização.

y = [0,0];                    % vetor dos valores das funções objetivo.

cons = [0,0];                 % vetor dos valores das restrições.

y(1) = x(1);                  % primeira função objetivo.

y(2) = (1+x(2)) / x(1);       % segunda função objetivo.

c = x(2) + 9*x(1) - 6;       % restrição da primeira função objetivo.

```

```

if (c<0)

    cons(1) = abs(c);           % loop para garantir a desigualdade da restrição

end                            $x_2 + 9x_1 \geq 6.$ 

c = - x(2) + 9*x(1) - 1;     % restrição da segunda função objetivo.

if (c<0)

    cons(2) = abs(c);         % loop para garantir a desigualdade da restrição

end                            $-x_2 + 9x_1 \geq 1.$ 

```

O código desenvolvido por Song permite a utilização de variáveis reais e inteiras, para que isto seja possível basta especificar no arquivo *nsgaopt.m* onde está indicado '*vartype*', [ ] indicar da seguinte forma: se o valor for deixado em branco o tipo de variável padrão utilizada é a real e se assinalado o número 2 indica que o tipo de variável inteira é utilizada. Por exemplo na especificação das variáveis a seguir como '*vartype*', [ 1 1 2 ] indica que as duas primeiras variáveis são reais e a terceira é inteira. Outra opção como a computação paralela da função objetivo também pode ser ativada ao efetuar a modificação do parâmetro '*useParallel*', '*no*' para '*yes*'.

Após a configuração do modelo de otimização é necessário indicar o formato da população inicial. O código de Song admite três possibilidades:

1. Utiliza uma distribuição uniforme de números aleatórios entre os limites inferior e superior definidos para as funções objetivo;
2. Utiliza um arquivo de população gerado de uma otimização anterior e continua a otimização a partir destes indivíduos;
3. Utiliza a estrutura de um resultado populacional de uma otimização existente, por exemplo, uma otimização do mesmo problema pode ser rodada utilizando a estrutura anterior da população e torná-la a sua população inicial.

Estes parâmetros também são configurados no arquivo *nsgaopt.m* e segue a seguinte sintaxe: *options.initfun* = { }. A sintaxe que especifica o primeiro caso é a definida como *options.initfun* = {*@initpop*}. Para o segundo caso, a sintaxe é definida como *options.initfun*

=  $\{@initpop, strFileName, ngen\}$ , onde *strFileName* indica o nome do arquivo onde se encontra a otimização anterior e *ngen* representa um número inteiro que indica a geração partir da qual os indivíduos serão utilizados como a população inicial da otimização atual, se o valor for deixado em branco, entende-se que a última geração será utilizada. Para o terceiro caso, a sintaxe é definida como *options.initfun* =  $\{@initpop, oldresult, ngen\}$  onde *oldresult* representa a estrutura de uma otimização realizada anteriormente e guardada na área de trabalho do matlab e como no segundo caso o termo *ngen* também indica a partir de qual geração os indivíduos serão utilizados como população inicial da nova otimização.

Definida a população inicial a otimização é realizada e os operadores seleção, cruzamento e mutação são executados para chegar ao resultado de otimização final. No código utilizado o operador seleção emprega a técnica baseada em torneio binário onde o valor de *t* é igual a dois, ou seja, os indivíduos são tomados dois a dois. O cruzamento, como mencionado no subitem 5.4.2, utiliza a técnica de conhecida como cruzamento aritmético ou (PMX).

Os parâmetros para o cruzamento podem ser configurados na linha *options.crossover* do arquivo *nsgaopt.m* com a seguinte sintaxe: *options.crossover* =  $\{type, ratio\}$ . A variável *type* utilizar um *string* que define o tipo de *crossover*. A variável *ratio* utiliza um número escalar que define a posição dos filhos gerados em relação aos pais, por exemplo se  $0 \leq ratio \leq 1$  os filhos estão em uma posição entre os pais. Se ocorrer uma convergência prematura este valor deve ser configurado como sendo maior que um. O código utilizado no crossover é baseado na estrutura a seguir:

$$\begin{aligned} child1 &= parent1 + rand \times ratio \times (parent2 - parent1) \\ child2 &= parent2 - rand \times ratio \times (parent2 - parent1). \end{aligned}$$

O operador mutação utiliza a mutação gaussiana e também pode ser configurado no arquivo de opções do NSGAI o *nsgaopt.m* seguindo a sintaxe: *options.mutation* =  $\{type, scale, shrink\}$ . A variável *type* é uma *string* que define o tipo de mutação, a variável *scale* é um parâmetro que determina o desvio padrão do número aleatório gerado e a variável *shrink* é um número escalar entre 0 e 1 utilizado como parâmetro para pesquisa local. O operador mutação segue a seguinte estrutura:

$$\begin{aligned} child &= parent + S \times randn \times (ub - lb) \\ S &= scale \times (1 - shrink \times currGen/maxGen) \end{aligned}$$

Como critério de parada o código utiliza o parâmetro *options.maxGen* = [ ]. O valor utilizado foi o de 1000 gerações.

Os resultados da otimização são obtidos do arquivo *output2file.m* e segue a seguinte sintaxe: *function output (opt, state, pop, type, varargin)*.

Onde:

*opt* = estrutura do *options*;

*state* = estrutura do *state*;

*pop* = população atual;

*type* = utilizado para identificar a última chamada. O valor -1 indica a última chamada, qualquer outro valor ou vazio indicam a saída normal.

*varargin* = parâmetro especificado no vetor *options.outputfuns*.

## 5.6. Os Resultados da Aplicação

Como resultado da otimização, duas matrizes são geradas, como mostra a figura 5.7 a matriz 25 X 4 informa o tipo de componente por posição de *feeder*. A primeira coluna representa o *rack* de *feeder* do módulo de entrada do lado frontal, a segunda coluna representa o *rack* de *feeder* do módulo de entrada do lado traseiro, a terceira coluna representa o *rack* de *feeder* do módulo de saída do lado frontal e a quarta coluna representa o *rack* de *feeder* do módulo de saída do lado traseiro. As linhas 1 a 25 representam a posição do *feeder* e o número representa o tipo de componente alocado na posição do *feeder*. A segunda matriz informa a sequência de aquisição e de montagem dos componentes. Como a máquina de montagem possui quatro cabeças, a figura 5.8 apresenta uma matriz que é gerada para cada uma e seu tamanho varia com o número de ciclos de montagem, esta quantidade varia com a quantidade total de componentes a ser montada, devido ao fato de cada cabeça poder pegar até 12 componentes, a matriz de sequência é de dimensão 12 X N, onde  $N = n^{\circ}$  de ciclos por cabeça.

Figura 5.7: Solução caso 1: Arranjo de *feeder* dos quatro racks.
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 4 & 8 & 10 \\ 1 & 6 & 7 & 15 \\ 5 & 11 & 12 & 16 \\ 3 & 9 & 14 & 13 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

FONTE: Autor, 2015

Figura 5.8: Sequência de montagem da primeira cabeça para o caso 1.

$$\begin{pmatrix} 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 2 & 1 & 1 & 5 \\ 2 & 2 & 1 & 1 & 5 \\ 2 & 2 & 1 & 1 & 5 \\ 2 & 2 & 1 & 3 & 5 \\ 2 & 2 & 1 & 3 & 0 \end{pmatrix}$$

FONTE: Autor, 2015

Logo após a aplicação do procedimento de otimização os seus resultados foram comparados com os obtidos pelo software de otimização do fabricante. A tabela 5.2 apresenta o desempenho do software empregado pelo fabricante para realizar a otimização.

Tabela 5.2: Tempo total de ciclo utilizando o software do fabricante.

PCI	Tamanho: C x L (mm)	Número de tipos de componente	Ciclos por Cabeça				Tempo total (seg)
			H1	H2	H3	H4	
1	192 x 163	28	5	5	4	5	20.296
2	195 x 138	31	5	6	5	5	18.569
3	192 x 158	17	8	8	9	8	38.804
4	290 x 240	19	16	16	16	16	65.109
5	192 x 158	13	7	6	7	7	19.372

Fonte: Autor, 2015.

A tabela 5.3 apresenta os resultados obtidos com o procedimento proposto nesta dissertação, utilizando as mesmas condições dos resultados obtidos pelo fabricante, e apresentados na tabela 5.2.

Tabela 5.3: Tempo total de ciclo utilizando o algoritmo proposto

PCI	Tamanho: C x L (mm)	Número de tipos de componente	Ciclos por Cabeça				Tempo total (seg)
			H1	H2	H3	H4	
1	192 x 163	28	5	4	4	4	17.823
2	195 x 138	31	5	5	5	5	15.992
3	192 x 158	17	8	8	8	8	32.214
4	290 x 240	19	15	14	15	15	59.745
5	192 x 158	13	6	6	6	6	14.282

Fonte: Autor, 2015.

A tabela 5.4 apresenta o desempenho dos dois métodos e uma estimativa de quanto é possível aumentar a produtividade ao aplicar o método proposto.

Tabela 5.4: Comparativo do desempenho dos dois métodos.

PCI	Tamanho: C x L (mm)	Número de tipos de componente	Tempo total fabricante Hitachi (seg)	Tempo total método proposto (seg)	Melhoria (%)	Ganho de placas produzidas
1	192 x 163	28	20.296	17.823	12,18	147
2	195 x 138	31	18.569	15.992	13,88	187
3	192 x 158	17	38.804	32.214	16,98	113
4	290 x 240	19	65.109	59.745	8,24	29
5	192 x 158	13	19.372	14.282	26,28	397

\*Para o cálculo do ganho de placas utilizou-se um turno de produção cuja duração é de 21600 segundos (considerando o tempo de pausas).

Fonte: Autor, 2015.

## CAPÍTULO VI

### CONCLUSÃO

O método proposto utilizou uma abordagem pré-definida para determinar o melhor arranjo para a população inicial. Estas condições iniciais ótimas contribuem para redução do tempo de processamento computacional, os resultados obtidos melhoraram o processo de montagem, porém o método torna-se limitado devido às restrições construtivas da máquina. O menor ganho de 29 placas para o exemplo 4 demonstra que o método apresenta um ganho, porém inexpressivo quando o número de ciclos de montagem por cabeça for próximo de 15.

Ao passo que o método proposto alcançou um melhor desempenho para os exemplos 1, 2 e 5 onde o número de ciclos por cabeça é próximo de 5. Foi possível observar que ao aplicar a técnica proposta foi possível reduzir o número de ciclos por cabeça pela melhor distribuição dos componentes utilizando as técnicas baseadas na disposição dos componentes na placa de circuito impresso.

A modelagem matemática desempenhou papel crucial onde, com objetivos bem definidos, foi possível aplicar a técnica computacional NSGAI e oferecer uma solução para o problema de eficiência na montagem de placas em seu processo de produção.

#### 6.1 Recomendações para Trabalhos Futuros

A empresa estudada possui outras linhas de montagem que empregam máquinas de diferentes fabricantes como *Fuji* e *Panasonic*. Estas máquinas também podem ser modeladas conforme as suas características construtivas e de acordo com as mesmas modelar as condições iniciais. Estas máquinas são modulares e compostas por cabeças rotatórias sendo duas para cada módulo. Desta forma é possível realizar a modelagem tornando-se necessário apenas coletar os dados da máquina como: posicionamento da câmera de reconhecimento, dimensões de largura e comprimento úteis, etc.

Uma forma de melhorar o método proposto pode ser a aplicação do agrupamento dos componentes com referência à distribuição normal dos mesmos, ou seja, o ponto de referência não mais se basearia no ponto médio total no eixo X e sim no ponto médio dos componentes tomados em grupos de 12 independentemente da altura de componentes. A restrição de altura

pode ser ainda aplicada a esta distribuição, porém uma restrição quanto ao deslocamento da cabeça pode ser aplicada para prevenir a colisão de componentes. Este agrupamento também pode ser estendido ao eixo Y incluindo assim componentes desse eixo que minimizem o deslocamento da cabeça no eixo X.

Outra abordagem para melhorar o método seria a adoção da mediana das coordenadas do eixo Y como forma de melhorar a atribuição dos componentes para os *racks* de *feeders* dentro de um mesmo módulo.

Um próximo passo para aumentar a eficiência da linha de montagem SMT seria a integração do método proposto à uma heurística mais abrangente de forma a reduzir o tempo de configuração da linha por modelo, haja vista que estas linhas são flexíveis e podem montar diversos tipos de PCI.

Outro estudo que pode ser conduzido para reafirmar a importância do método seria o acompanhamento do impacto que o método gera sobre o consumo das máquinas e do seu efeito sobre a qualidade de energia, sendo que as novas normas brasileiras apontam para multas pesadas que serão aplicadas à empresa que violar os limites impostos pela norma.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Ahuja K., T. Magnanti, J. Orlin, Network flows, Prentice Hall, New York (1993).
- Angel E., E. Bampis, L. Gourvès, Approximating the Pareto curve with local search for the bi-criteria TSP (1,2) problem. *Theoretical Computer Science* 310, pp. 135-146 (2004).
- Angel E., E. Bampis, L. Gourvès, J. Monnot, (Non-) approximability for the multi-criteria TSP (1,2), in Liskiewicz M., R. Reischuk (eds), *Proc. of the 15th International Symposium on Fundamentals of Computation Theory, Lecture Notes in Computer Science* 3623, pp. 329-340, Springer-Verlag, Berlin, Heidelberg (2005).
- Ashayeri, J. Sotirov, N. An aggregated optimization model for multi-head SMD placements. *Computers & Industrial Engineering* 60, pages 99–105, Elsevier (2011).
- Audet C., G. Savard, W. Zghal, New branch-and-cut algorithm for bi-level linear programming, *Journal of Optimization Theory and Applications* 134, pp. 353–370 (2007).
- Ayob, M., Kendall, G., A triple objective function with a Chebychev dynamic pick-and-place point specification approach to optimise the surface mount placement machine. *European Journal of Operational Research* 164, 609–62 (2005).
- Ayob, M., Optimisation of Surface Mount Device Placement Machine in Printed Circuit Board Assembly, Ph.D. thesis, University of Nottingham, The School of Computer Science and Information Technology, Nottingham (2005).
- Ayob, Masri. Kendall, Graham. A triple objective function with a Chebychev dynamic pick-and-place point specification approach to optimise the surface mount placement machine. *European Journal of Operational Research* 164, pages 609–626, Elsevier (2005).
- Balinski, M. L., A. Russakoff., On the assignment polytope, *SIAM Rev.* 16, 516-525, (1974).
- Bard J.F., J. Moore, J., A branch-and-bound algorithm for the bilevel programming problem, *SIAM Journal of Scientific and Statistical Computing* 11, pp. 281–292 (1990).
- Bellman, Richard. "On a routing problem". *Quarterly of Applied Mathematics* 16: 87–90 (1958).
- Bentzen, B. (2000). SMD placement, in the *SMT in FOCUS*. Disponível em: <[http://www.smtinfocus.com/PDF/SMD\\_placement.pdf](http://www.smtinfocus.com/PDF/SMD_placement.pdf)> Acesso em: 19 Out. 2014.
- Bialas W., M. Karwan, J. Shaw, A parametric complementarity pivot approach for two-level linear programming, *Operations Research Program Report* 80-2, State University of New York at Buffalo (1980).
- Bialas W., M. Karwan, Two-level linear programming, *Management Science* 30, pp. 1004–1020 (1984).

- Bialas W., M. Karwan, Two-level linear programming, *Management Science* 30, pp. 1004-1020 (1984).
- Birkhoff, Garrett. "Tres observaciones sobre el algebra lineal", *Univ. Nac. Tucumán. Revista A*. **5**: 147–151, (1946).
- Boussaïd, Ilhem. Lepagnot, Julien. Siarry, Patrick. A survey on optimization metaheuristics. *Information Sciences* 237, pages 82–117, Elsevier (2013).
- Bouza G., G. Still, Mathematical programs with complementarity constraints: convergence properties of a smoothing method, *Mathematics of Operations Research* 32, pp. 467–483 (2007).
- Brumbaugh-Smith J., D. Shier, An empirical investigation of some bi-criteria shortest path algorithms, *European Journal of Operational Research* 43 (2), pp. 216–224 (1989).
- Burkard, R.E.; Karisch, S. & Rendl, F. QAPLIB – A quadratic assignment problem library. *European Journal of Operational Research*, 55, 115-119 (1991).
- Candler W., R. Townsley, A linear two-level programming problem, *Computers and Operations Research* 9, pp. 59–76 (1982).
- Chang, Pyung Hun. Park, Juyi. A concurrent design of input shaping technique and a robust control for high-speed/high-precision control of a chip mounter. *Control Engineering Practice* 9, pages 1279–1285, Elsevier (2001).
- Chankong V., Y. Haimes, *Multiobjective decision making theory and methodology*, Elsevier Science, New York (1983).
- Charnes A., W.W. Cooper, R.O. Ferguson, Optimal estimation of executive compensation by linear programming, *Management Science* 1 (2), pp. 138–151 (1955).
- Charnes A., W.W. Cooper, *Management models and industrial applications of linear programming*, John Wiley & Sons volumes I and II, New York (1961) .
- Climaco J.C.N., E.Q.V. Martins, A bicriterion shortest path algorithm, *European Journal of Operational Research* 11, pp. 399–404 (1982).
- Coello C.A., A comprehensive survey of evolutionary-based multi-objective optimization techniques, *Knowledge and Information Systems. An International Journal* 1 (3), pp. 269– 308 (1999).
- Colson B., P. Marcotte, G. Savard, Bilevel programming: A survey, *4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 3, pp. 87–107 (2005).
- Cormen T.H., C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to algorithms*, Second Edition. MIT Press and McGraw-Hill, London (2001).

Crama, Y., Flippo, O. E., Klundert, J. J. V. D. and Spieksma, F. C. R. The assembly of printed circuit boards: A case with multiple machines and multiple board types. *European Journal of Operational Research*, 98, 457-472 (1997).

Csaszar, Peter. Nelson, Peter C. Rajbhandari, Rajan R. Tirpak, Thomas M. Optimization of Automated High-Speed Modular Placement Machines Using Knowledge-Based Systems. *IEEE transactions on systems, man, and cybernetics—part c: applications and reviews*, vol. 30, no. 4, 2000.

D. Angluin, L.G. Valiant., Fast probabilistic algorithms for Hamiltonian circuits and matchings. *J. Comput. System Sci.*, 18, pp. 155–193 (1979).

Deb K., A tutorial on evolutionary multi-objective optimization (EMO), Dagstuhl Seminar Proceedings 04461 Practical Approaches to Multi-Objective Optimization disponível em: <<http://drops.dagstuhl.de/opus/volltexte/2005/252>> Acesso em: 12 Dez. 2014.

Deb K., Multi-objective optimization using evolutionary algorithms, Wiley, Chichester, UK (2001).

Deineko VG, Filonenko VL., On the reconstruction of specially structured matrices. *Aktualnyje Problemy EVM: programirovaniye*, Dnepropetrovsk, DGU, 43-45, Russian (1979).

Demirkale, Hüseyin. Alkaya, Ali Fuat. Duman, Ekrem. Exact and Metaheuristic Approaches for Optimizing the Operations of Chip Mounter Machines. 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM).

Derigs U, Goecke O, Schrader R., Monge sequences and a simple assignment problem. *Dis Appl Math*, 5:241-248 (1986).

Dial, R. B. Algorithm 360: Shortest Path Forest with Topological Ordering. *Communications of the ACM*, 12, 632-633, (1969).

Dijkstra, E. W. "A note on two problems in connexion with graphs". *Numerische Mathematik* 1: pp. 269–271, (1959).

Dussault J.-P., P. Marcotte, S. Roch, G. Savard, A smoothing heuristic for a class of bilinear bilevel programs, *European Journal of Operational Research* 174 (3), pp. 1396–1413 (2006).

E. Deza, M. M. Deza, *Dictionary of Distances*, Elsevier, Amsterdam, 2006, 391 pp, EUR 125, ISBN 0-444-52087-2.

Ehrgott M., A discussion of scalarization techniques for multiple objective integer programming, *Annals of Operational Research* 147, pp. 343–360 (2006).

Ehrgott M., A discussion of scalarization techniques for multiple objective integer programming, *Annals of Operational Research* 147, pp. 343–360 (2006).

- Ehrgott M., Approximation algorithms for combinatorial multi-criteria optimization problems, *International Transactions in Operational Research* 7 (1), pp. 5-31 (2000).
- Ehrgott M., H.W. Hamacher, K. Klamroth, S. Nickel, A. Schöbel, M.M. Wiecek, A note on the equivalence of balance points and Pareto solutions in multiple-objective programming, *Journal of Optimization Theory and Applications* 92 (1), pp. 209–212 (1997).
- Ehrgott M., *Multicriteria optimization*, Springer-Verlag, Berlin, Heidelberg (2005).
- Ehrgott M., T.P. Dagmar, S. Thomas, A level set method for multi-objective combinatorial optimization: application to the quadratic assignment problem, *Pacific Journal of Optimization* 2 (3), pp. 521–544 (2006).
- Ehrgott M., X. Gandibleux, *Multiple criteria optimization: state of the art annotated bibliographic survey*, Kluwer Academic Publishers, Boston (2002).
- Ehrgott, M., A.J.V. Skriver, Solving biobjective combinatorial max-ordering problems by ranking methods and a two-phases approach, *European Journal of Operational Research* 147, pp. 657–664 (2003).
- Ehrgott M., S. Rusika, An improved  $\varepsilon$ -constraint method for multiobjective programming, *Report in Wirtschaftsmathematik (WIMA Report)* (2005).
- Erkut E., F. Gzara, Solving the hazmat transport network design problem, *Computers and Operations Research* (35), pp. 2234–2247 (2008).
- Falk J.E., J. Liu, On bilevel programming, part I: general nonlinear cases, *Mathematical Programming* 70, pp. 47–72 (1995).
- Fonseca C.M. & P.J. Fleming P.J. "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization". *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 316-323, 1993.
- Fonseca C.M., P.J. Fleming, An overview of evolutionary algorithms in multi-objective optimization, *Evolutionary Computation* 3 (1), pp. 1–16 (1995).
- Ford Jr., Lester R. *Network Flow Theory*. Paper P-923. Santa Monica, California: RAND Corporation (1956).
- Ford, L. R., Jr.; Fulkerson, D. R., "Maximal flow through a network" (PDF), *Canadian Journal of Mathematics* 8: 399–404, (1956).
- Fred Glover, "Maximum Matching in a Convex Bipartite Graph," *Naval Research Logistics Quarterly* 15, 3, 313-316, September 1967.
- Fredman, Michael Lawrence; Tarjan, Robert E. "Fibonacci heaps and their uses in improved network optimization algorithms". *Journal of the Association for Computing Machinery* 34 (3): 596–615, (1987).

Fukushima M. and Pang, J.S. Complementarity constraint qualifications and simplified b-stationary conditions for mathematical programs with equilibrium constraints, *Computational Optimization and Applications* 13, pp. 111–136 (1999).

Fukushima, H., Usui, Y., Takemura, I., Katsuta, S. Direct Drive Modular Mounter “GXH-3” Contributing to High Productivity and High Density Placement, *Hitachi Review* Vol. 57, No. 3, Japan (2008).

Gabow, H.N., and R.E. Tarjan,. A linear-time algorithm for a special case of disjoint set union. *J. Comput. System Sci.* 30, 209-221, (1985).

Gallo, G., and Pallottino, S., “Shortest Path Algorithms,” *Annals of Operations Research*, Vol. 7, pp. 3-79, (1988).

Gandibleux X., F. Beugnies, S. Randriamasy, Martins’ algorithm revisited for multi-objective shortest path problems with a MaxMin cost function, *4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 4, pp. 47–59 (2006).

Gandibleux X., N. Mezdaoui, A. Frville, A multi-objective tabu search procedure to solve combinatorial optimization problems, in Caballero R., F. Ruiz, R. Steuer (eds), *Advances in Multiple Objective and Goal Programming. Lecture Notes in Economics and Mathematical Systems* 455, Springer Verlag Berlin Heidelberg, pp. 291–300 (1997).

Gastel, S. V. (2002). A comparison of SMD placement machine concepts, in the *SMT in FOCUS*. Disponível em : <[http://www.smtinfocus.com/PDF/SMD\\_machine\\_concept\\_Assembleon.pdf](http://www.smtinfocus.com/PDF/SMD_machine_concept_Assembleon.pdf)> Acesso em: 02 Out 2014.

Geoffrion A.M., Proper efficiency and the theory of vector maximization, *Journal of Mathematical Analysis and Applications* 22, pp. 618–630 (1968).

Goldberg D.E., P. Segrest, Finite Markov chain analysis of genetic algorithms, in Grefenstette H.-M. (ed.), *Genetic algorithms and their applications. Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, NJ, pp. 1–8 (1987).

Goldberg, D.E. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

Goldberg, A.V. Scaling algorithms for the shortest paths problem, In *Proc. 4nd ACM-SIAM Symposium on Discrete Algorithms*, pp. 222-231, (1993).

Grotzinger, S. Feeder assignment models for concurrent placement machines. *IIE Transactions*, 24, 31-46 (1992).

Gutin G., A.P. Punnel, *The traveling salesman problem and its variations*, Springer-Verlag, Berlin (2006).

Hajela P., C.Y. Lin, Genetic search strategies in multicriterion optimal design, *Structural Optimization* 4, pp. 99–107 (1992).

Hajela, P. Lin, C.Y. "Genetic Search strategies in multicriterion optimal design". *Structural and Multidisciplinary Optimization*, 3 (2). Springer, 99-107, 1992.

Hall, Philip., "On Representatives of Subsets", *J. London Math. Soc.* 10 (1): 26–30 (1935).

Hansen M.P., *Metaheuristics for multiple objective combinatorial optimization*, Ph.D. Thesis, IMM-PHS-1998-45, Technical University of Denmark, Lyngby (1998).

Hansen P., Bicriteria path problems, in Fandel G., T. Gal (eds), *Multiple criteria decision making theory and applications*, Lecture Notes in Economics and Mathematical Systems 177, pp. 109–127 Springer-Verlag, Berlin (1979).

Hansen P., Jaumard, B., Savard, G., New branch-and-bound rules for linear bi-level programming, *SIAM Journal of Scientific and Statistical Computing* (13), pp. 1194–1217 (1992).

Hansen P., Jaumard, B., Savard, G., New branch-and-bound rules for linear bilevel programming, *SIAM Journal of Scientific and Statistical Computing* (13), pp. 1194–1217 (1992).

Hardy, G. H. Littlewood, J. E. and Polya, G. "Inequalities," Cambridge Univ. Press, New York, (1952).

Henig M., The shortest path problem with two objective functions, *European Journal of Operational Research* 25, pp. 281–291 (1986).

Hitachi High-Tech Instruments Co., Ltd. *Instruction Manual Chip Mounter GXH-3*, Volume 3: Programming and Machine Data. 2nd edition, Dec., 2007.

Ho, William. Ji, Ping. An integrated scheduling problem of PCB components on sequential pick-and-place machines: Mathematical models and heuristic solutions. *Expert Systems with Applications* 36, pages 7002–7010, Elsevier (2009).

Hoffman, A. J. "On the Polynomial of a Graph." *Amer. Math. Monthly* 70, 30-36, (1963).

Hopcroft, John E.; Karp, Richard M., "An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs", *SIAM Journal on Computing* 2 (4): 225–231 (1973).

Horn, J, N. Nafploitis e D. Goldberg. A niched Pareto Genetic Algorithm for multiobjective optimization. *Proceedings of the first IEEE Conference on Evolutionary Computation (CEC'93)* 1, IEEE Press, 82-87, 1993.

<http://www.mathworks.com/matlabcentral/fileexchange/5550-dijkstra-shortest-path-routing>.

Jaszkiewicz A., multiple objective metaheuristic algorithms for combinatorial optimization, Technical Report, Politechnika Poznanska, Poznan (2001).

Júdice J., A. Faustino, A sequential LCP method for bilevel linear programming, *Annals of Operations Research* 34, pp. 89–106 (1992).

Juloski, A. Lj. Heemels, W.P.M.H. Ferrari-Trecate, G. Data-based hybrid modelling of the component placement process in pick-and-place machines. *Control Engineering Practice* 12, pages 1241–1252, Elsevier (2004).

Kara B.Y., V. Verter, Designing a road network for hazardous materials transportation, *Transportation Science* 38 (2), 188–196 (2004).

Khoo, L. P. and Loh, K. M. A genetic algorithms enhanced planning system for surface mount pcb assembly. *International Journal of Advanced Manufacturing Technology*, 16(4), 289–296 (2000).

Kocvara M., J.V. Outrata, Optimization problems with equilibrium constraints and their numerical solution, *Mathematical Programming* 101, pp. 119–149 (2004).

Kőnig, Dénes. "Gráfok és mátrixok", *Matematikai és Fizikai Lapok* 38: 116–119 (1931).

Kostreva M.M., M.M. Wiecek, Time dependency in multiple objective dynamic programming, *Journal of Mathematical Analysis and Applications* 173 (1), pp. 289–307 (1993).

Land A. H, Doig A. G. An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, Vol. 28, No. 3. pp. 497-520. Jul, 1960.

Leu, M.C., Wong, H. and Ji. Z. Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm. *Journal of Electronic Packaging*, 115, 424-432 (1993).

Levit, B. Ju. Livshits, B. N. Neleneinye Setevye Transportnye Zadachi Transport, Moscow, 1972.

Li, Shaoyuan. Hu, Chaofang. Tian, Fuhou. Enhancing optimal feeder assignment of the multi-head surface mounting machine using genetic algorithms. *Applied Soft Computing* 8 , pages 522–529, Elsevier (2008).

Lin G. H., Fukushima M., New relaxation method for mathematical programs with complementary constraints, *Journal of Optimization and its Applications* 118, pp. 81–116 (2003).

Lin, G.H., M. Fukushima, A modified relaxation scheme for mathematical programs with complementarity constraints, *Annals of Operations Research* 133, pp. 63–84 (2005).

Song, L. NGPM - A NSGA-II Program in Matlab, disponível no seguinte site: <<http://www.mathworks.com/matlabcentral/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4>> Acesso em: 10 Fev. 2014.

List G.F., P.B. Mirchandani, M.A. Turnquist, K.G. Zografos, Modeling and analysis for hazardous materials transportation: Risk analysis, routing/scheduling and facility location, *Transportation Science* 2, pp. 100–114 (1991).

LOIOLA, Eliane Maria; ABREU, Nair Maria Maia de; BOAVENTURA NETTO, Paulo Oswaldo. Uma revisão comentada das abordagens do problema quadrático de alocação. *Pesqui. Oper.*, Rio de Janeiro, v. 24, n. 1, p. 73-109, Apr. 2004

Luo Z.Q., J.S. Pang, D. Ralph, *Mathematical programs with equilibrium constraints*, Cambridge University Press, Cambridge (1996).

Manthley B., Ram L.S., *Approximation algorithms for multi-criteria traveling salesman problems*, Proceedings of the 4th Workshop on Approximation and Online Algorithms, Springer-Verlag, Berlin, Heidelberg (2006).

Marcotte P., G. Savard, *Bilevel programming: A combinatorial perspective*, in Avis D. et al. (eds), *Graph Theory and Combinatorial Optimization*. Kluwer Academic Publishers, Boston (2005).

Marcotte, P., W. Shiquan, Y. Chen, *A cutting-plane algorithm for the linear bi-level programming problem*, CRT Report 925 (1993).

Magyar, G., Johnsson, M. And Nevalainen, O. *On solving single machine optimization problems in electronics assembly*. *Journal of Electronics Manufacturing*, 9(4), 249-267 (1999).

Martins E.Q.V., J.C.N. Climaco, *On the determination of the nondominated paths in a multi-objective network problem*, *Mathematical Methods of Operations Research* 40, pp. 255–258 (1981).

Martins E.Q.V., J.L.E. Santos, *The labeling algorithm for the multi-objective shortest path problem*, (link para download <http://www.mat.uc.pt/~eqvm/cientificos>) (1999).

Martins E.Q.V., *On a special class of bicriterion path problems*, *European Journal of Operational Research* 17, pp. 85–94 (1984).

Miettinen K., *On the methodology of multiobjective optimization with applications*, Ph.D. thesis, University of Jyväskylä, Department of Mathematics, Jyväskylä, Finland (1994).

Monticeli, A. R. Título: Um estudo sobre equações lineares. Março, 2010. 123 páginas. Dissertação - Universidade Estadual de Campinas. Campinas, 25 de março de 2010.

Moore, Edward F. *The shortest path through a maze*. Proc. Internat. Sympos. Switching Theory 1957, Part II. Cambridge, Mass.: Harvard Univ. Press. pp. 285–292 (1959).

Mote J., I. Murthy, D.L. Olson, *A parametric approach to solving bi-criteria shortest path problems*, *European Journal of Operational Research* 53, pp. 81–92 (1991).

Moyer, L.K. and Gupta, S. M. *SMT feeder slot assignment for predetermined component placement paths*. *Journal of Electronics Manufacturing*, 6, 173-192 (1996).

- Moyer, L.K. and Gupta, S. M. An efficient assembly sequencing heuristic for printed circuit board configurations. *Journal of Electronics Manufacturing*, 7, 143-160 (1997).
- Murata T., H. Ishibuchi, H. Tanaka, Multi-objective genetic algorithm and its application to flowshop scheduling, *Computers and Industrial Engineering* 30 (4), pp. 957–968 (1996).
- N. Kumar, B. Karambir, R. Kumar. *International Journal of Latest Research in Science and Technology* Vol.1, Issue 2 :Page No.98-101 , July (2012).
- Ning-Cheng Lee, *Introduction to Surface Mount Technology, Reflow Soldering Processes*, Pages 1-18, 2001.
- Outrata J.V., M. Kocvara, J. Zowe, *Nonsmooth Approach to optimization problems with equilibrium constraints*, Kluwer Academic, Dordrecht (1998).
- Paixão J.M.P., E.Q.V. Martins, M.S. Rosa, J.L.E. Santos, The determination of the path with minimum-cost norm value, *Networks* 41 (4), pp. 184–196 (2003).
- Papadimitriou C.H., M. Yannakakis, On the approximability of trade-offs and optimal access of web sources, in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pp. 86-92, IEEE Computer Society (2000).
- Pape, U. “Implementation and Efficiency of Moore - Algorithms for the Shortest Path Problem,” *Math. Programming*, Vol. 7, pp. 212-222, (1974).
- Park, Tae-Hyoung. Kim, Kyung-Min. *Integer Programming Approach to PCB Assembly Optimization of Modular Mounters*. *Evolutionary Computation (CEC)*, 2010 IEEE Congress on Barcelona.
- Rote, Günter. Rendl, Franz., Minimizing the density of terminal assignments in layout design. *Operations Research Letters* 5 111–118 (1986).
- Ruzika S., Wiecek M., Approximation methods in multiobjective programming, *Journal of Optimization Theory and Applications* 126 (3), pp. 473–501 (2005).
- Sancho N.G.F., A new type of multiobjective routing problem, *Engineering Optimization* 14, pp. 115–119 (1988).
- Schaffer J.D., Multiple objective optimization with vector evaluated genetic algorithms, in: Grefenstette J.J. (ed.), *Genetic Algorithms and Their Applications: Proceedings of the Third International Conference on Genetic Algorithms*, pp. 93–100 Lawrence Erlbaum, Hillsdale, NJ (1985).
- Schmitt R., *Machine and Process Capability Analysis in SMT Manufacturing - Electronics System Integration Technology Conference*, Dresden, Germany (2006).
- Scholtes S., M. Stohr, Exact penalization of mathematical programs with equilibrium constraints *SIAM Journal on Control and Optimization* 37, pp. 617–652 (1999).

Serafini P., Simulated annealing for multiple objective optimization problems, in volume 1 of the Proceedings of the Tenth International Conference on Multiple Criteria Decision Making, Taipei, pp. 87–96 (1992).

Serafini P., Some considerations about computational complexity for multi objective combinatorial problems, in Jahn J., W. Krabs (eds), Recent advances and historical development of vector optimization, Lecture Notes in Economics and Mathematical Systems 294, pp. 222-232, Springer-Verlag, Berlin (1986).

Shih, W., Srihari, K. and Adriance, J. Expert system based placement sequence identification for surface mount PCB assembly. *International Journal of Advanced Manufacturing Technology*, 11, 413-424 (1996).

Skriver A.J.V., K.A. Andersen, A label correcting approach for solving bi-criteria shortest-path problems, *Computers and Operations Research* 27, pp. 507–524 (2000).

Srinivas N., K. Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation* 2, pp. 221–248 (1994).

Steuer R., Multiple criteria optimization: theory, computation and application, John Wiley, New York (1986).

Strauss, R., Surface Mount Technology - SMT Soldering Handbook, 2nd edition, 621.3815, Newnes, Oxford (1998).

Sun, Dong-Seok. Lee, Tae-Eog. Kim, Kyung-Hoon. Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool. *International Journal of Production Economics* 95, pages 245–264, Elsevier (2005).

T'Kindt V., J.-C. Billaut, Multicriteria scheduling, theory, models and algorithms, Springer-Verlag, Berlin, Heidelberg (2005).

Tae-Hyoung Park; Kyung-Min Kim, "Integer programming approach to PCB assembly optimization of modular mounters," in *Evolutionary Computation (CEC), 2010 IEEE Congress on* , vol., no., pp.1-6, 18-23 July 2010.

Tamaki H., H. Kita, S. Kobayashi, Multi-objective optimization by genetic algorithms: A review, in Fukuda T., T. Furuhashi (eds), Proceedings of the 1996 International Conference on Evolutionary Computation, pp. 517–522, Nagoya, Japan, IEEE (1996).

Tarapata Z., Selected Multicriteria Shortest Path Problems: An Analysis of Complexity, Models and Adaptation of Standard Algorithms, *International Journal of Applied Mathematics International Journal of Applied Mathematics* (2007).

Tirpak, T. M. Design to manufacturing information management for electronics assembly. *International Journal of Flexible Manufacturing Systems*, 12, 189-205 (2000).

- Tsaggouris G., C. Zaroliagis, Improved FPTAS for multiobjective shortest paths with applications, Technical Report TR 2005/07/03, Research Academic Computer Technology Institute, Petras, Greece (2005).
- Tutte, W. T. "The factorization of linear graphs". The Journal of the London Mathematical Society, Ser. 1 22 (2): 107–111 (1947).
- Tuy H., A. Migdalas, P. Varbrand, A global optimization approach for the linear two-level program, Journal of Global Optimization 3, pp. 1–23 (1993).
- Tuy H., Concave programming under linear constraints, Soviet Mathematics 5, pp. 1437-1440 (1964).
- Ulungu E.L., J. Teghem, P. Fortemps, D. Tuyttens, MOSA method: a tool for solving multi-objective combinatorial optimization problems, Journal of Multi-Criteria Decision Analysis 8, pp. 221–236 (1999).
- van Stackelberg H., The theory of market economy. Oxford University Press, Oxford (1952).
- Vassilvitskii S., M. Yannakakis, Efficiently computing sufficient trade-off curves, Lecture Notes in Computer Science 3142, pp. 1201-1213 (2004).
- Vicente L.N., G. Savard, J. Júdice, Discrete linear bilevel programming problem, Journal of Optimization Theory and Applications 89, pp. 597–614 (1996).
- Vicente L.N., P.H. Calamai, Bilevel and multilevel programming: a bibliography review, Journal of Global Optimization 5, pp. 291–306 (1994).
- Wang, W., Nelson, P. C. and Tirpak, T. M. Optimization of high-speed multistation SMT placement machines using evolutionary algorithms. *IEEE Transactions on Electronics Packaging Manufacturing*, 22(2), 137-146 (1999).
- Warburton A., Approximation of Pareto optima in multi-objective shortest path problem, Operations Research 35, pp. 70–79 (1987).
- Wen U., S. Hsu, Linear bi-level programming problems - A review, Journal of the Operational Research Society 42, pp. 125–133 (1991).
- White D., G. Anandalingam, A penalty function approach for solving bi-level linear programs, Journal of Global Optimization 3, pp. 397–419 (1993).
- Wilhelm, Wilbert E. Choudhry, Nilanjan D. Damodaran, Purushothaman. A model to optimize placement operations on dual-head placement machines. Discrete Optimization 4, pages 232–256, Elsevier (2007).
- Yu P.L., Cone convexity, cone extreme points and nondominated solutions in decision problems with multi-objectives, Journal of Optimization Theory and Applications 14, pp. 319–377 (1974).

Yuan, Peng. Hu, Yue-ming, Liu Hai-ming. Gao, Hong-xia. Scatter Search algorithm for multi-headed Mouter Networking, Sensing and Control. ICNSC 2008. IEEE International Conference on Sanya.

Zhou, Aimin. Qu, Bo-Yang. Li, Hui. Zhao, Shi-Zheng. Suganthan, Ponnuthurai Nagaratnam. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, pages 32–49, Elsevier (2011).

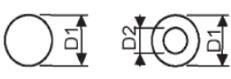
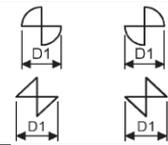
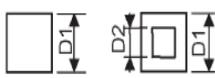
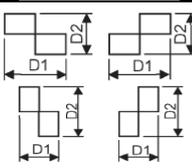
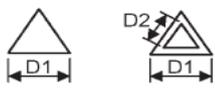
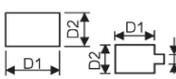
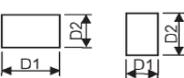
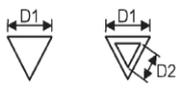
Zitzler E., M. Laumanns, S. Bleuler, A Tutorial on Evolutionary Multi-Objective Optimization, Workshop on Multiple Objective Metaheuristics, Springer-Verlag, Berlin (2004).

Zitzler, E. & Thiele, L. “An evolutionary algorithm for multiobjective optimization: The strength Pareto approach”. Technical Report 33, Computer Engineering and Networks Laboratory (TIK). Swiss Federal Institute of Technology (ETH) Zurich, 1998.

Zitzler, E. Laumanns, M. & Thiele, L. “SPEA 2: Improving the Strength Pareto evolutionary algorithm for multiobjective optimization”. Proceedings of the EUROGEN 2001 – Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. K. C. Giannakoglou et al. (Eds.), 95-100, 2001.

## ANEXO I – TABELA DE MARCAS DE FIDUCIAL

Tabela 1 - Padrões de marcação de fiducial que a máquina GXH-3 pode reconhecer.

Tipo de Marca	Diâmetro 1 (mm)	Diâmetro 2 (mm)	Tipo de Marca	Diâmetro 1 (mm)	Diâmetro 2 (mm)
<b>Circular</b> 	0,5 à 3,0	0 à 2,8 ( $D1 - 0,2 \geq D2$ )	<b>Gravata</b> 	0,5 à 3,0	-
<b>Quadrado</b> 	0,5 à 3,0	0 à 2,8 ( $D1 - 0,2 \geq D2$ )	<b>Xadrez Retângulo</b> 	0,5 à 3,0	0,5 à 3,0
<b>Triângulo Equilátero</b> 	0,5 à 3,0	0 à 2,8 ( $D1 - 0,2 \geq D2$ )	<b>Furo Passante (Circular)</b> 	0,5 à 3,0	0,5 à 1,5
<b>Cruz</b> 	0,5 à 3,0	0 à 1,5 ( $D1/2 \geq D2$ )	<b>Marca de Ilha</b> 	0,5 à 3,0	0,5 à 2,0
<b>Xadrez Quadrado</b> 	0,5 à 3,0	-	<b>Retângulo</b> 	0,5 à 3,0	0,5 à 3,0
<b>Diamante</b> 	0,5 à 3,0	0 à 2,8 ( $D1 - 0,2 \geq D2$ )	<b>Triângulo Equilátero (Invertido)</b> 	0,5 à 3,0	0 à 2,8 ( $D1 - 0,2 \geq D2$ )

FONTE, Hitachi, 2007.

## ANEXO II – DADOS SOBRE COORDENADAS DE MONTAGEM

Tabela 1 - Organização das coordenadas de montagem caso 1.

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
C1024	163.70	-125.32	1.0 X 0.5 X 0.5	10	C7057	93.20	-96.42	1.0 X 0.5 X 0.5	8
C4010	40.85	-77.57	1.0 X 0.5 X 0.5	8	C7058	92.00	-96.42	1.0 X 0.5 X 0.5	8
C6004	112.40	-102.72	1.0 X 0.5 X 0.5	8	R1007_LCD	69.85	-126.87	1.0 X 0.5 X 0.35	1
C6017	119.00	-81.82	1.0 X 0.5 X 0.5	11	R1008_LCD	68.70	-126.87	1.0 X 0.5 X 0.35	6
C6023	119.00	-79.42	1.0 X 0.5 X 0.5	11	R1009_LCD	66.40	-121.52	1.0 X 0.5 X 0.35	4
C6039	119.00	-88.02	1.0 X 0.5 X 0.5	8	R1011_LCD	65.35	-121.52	1.0 X 0.5 X 0.35	4
C6040	119.00	-86.82	1.0 X 0.5 X 0.5	8	R1030	32.00	-116.22	1.0 X 0.5 X 0.35	1
C6042	119.00	-91.72	1.0 X 0.5 X 0.5	8	R6051	112.40	-103.92	1.0 X 0.5 X 0.35	4
C6043	119.00	-90.52	1.0 X 0.5 X 0.5	8	R6060	110.10	-99.62	1.0 X 0.5 X 0.35	4
C6044	119.00	-89.32	1.0 X 0.5 X 0.5	8	R6061	112.60	-99.62	1.0 X 0.5 X 0.35	4
C6047	116.40	-68.62	1.0 X 0.5 X 0.5	8	R6065	111.10	-111.12	1.0 X 0.5 X 0.35	4
C6050	112.40	-64.02	1.0 X 0.5 X 0.5	10	R6072	110.20	-108.02	1.0 X 0.5 X 0.35	4
C6051	111.20	-64.02	1.0 X 0.5 X 0.5	10	R6073	109.80	-111.12	1.0 X 0.5 X 0.35	4
C6057	88.60	-65.22	1.0 X 0.5 X 0.5	9	R6080	112.20	-111.12	1.0 X 0.5 X 0.35	2
C6058	90.40	-65.22	1.0 X 0.5 X 0.5	8	R6082	111.40	-99.62	1.0 X 0.5 X 0.35	2
C7002	114.60	-104.02	1.0 X 0.5 X 0.5	8	R6099	106.20	-103.22	1.0 X 0.5 X 0.35	2
C7003	107.40	-81.82	1.0 X 0.5 X 0.5	8	R6101	105.00	-103.22	1.0 X 0.5 X 0.35	2
C7005	108.60	-77.62	1.0 X 0.5 X 0.5	8	R6103	103.80	-103.22	1.0 X 0.5 X 0.35	2
C7007	87.50	-73.27	1.0 X 0.5 X 0.5	8	R6115	102.60	-103.22	1.0 X 0.5 X 0.35	2
C7008	93.80	-102.82	1.0 X 0.5 X 0.5	8	R6126	105.90	-108.72	1.0 X 0.5 X 0.35	4
C7009	89.60	-102.82	1.0 X 0.5 X 0.5	8	R6128	107.20	-111.12	1.0 X 0.5 X 0.35	4
C7010	114.60	-102.82	1.0 X 0.5 X 0.5	8	R6130	90.40	-64.02	1.0 X 0.5 X 0.35	3
C7012	107.80	-102.62	1.0 X 0.5 X 0.5	8	R6131	88.60	-64.02	1.0 X 0.5 X 0.35	3
C7013	87.80	-93.82	1.0 X 0.5 X 0.5	8	R6151_NFBE	90.20	-92.62	1.0 X 0.5 X 0.35	2
C7014	85.00	-85.42	1.0 X 0.5 X 0.5	8	R6160	91.80	-85.62	1.0 X 0.5 X 0.35	2
C7015	116.20	-103.12	1.0 X 0.5 X 0.5	8	R6161	89.40	-85.02	1.0 X 0.5 X 0.35	2
C7018	113.85	-70.42	1.0 X 0.5 X 0.5	8	R6165	89.40	-81.82	1.0 X 0.5 X 0.35	2
C7019	98.15	-79.02	1.0 X 0.5 X 0.5	8	R6168	91.80	-80.62	1.0 X 0.5 X 0.35	2
C7023	104.85	-69.87	1.0 X 0.5 X 0.5	8	R6169	89.40	-80.62	1.0 X 0.5 X 0.35	2
C7024	122.30	-77.22	1.0 X 0.5 X 0.5	8	R6186	90.20	-90.22	1.0 X 0.5 X 0.35	4
C7026	140.20	-124.82	1.0 X 0.5 X 0.5	8	R6187	90.20	-89.12	1.0 X 0.5 X 0.35	4
C7027	108.40	-69.67	1.0 X 0.5 X 0.5	8	R6196	85.00	-92.62	1.0 X 0.5 X 0.35	4
C7030	110.20	-102.62	1.0 X 0.5 X 0.5	8	R6216	87.80	-95.22	1.0 X 0.5 X 0.35	2
C7032	85.00	-78.62	1.0 X 0.5 X 0.5	8	R7001	69.80	-73.22	1.0 X 0.5 X 0.35	5
C7035	84.90	-94.22	1.0 X 0.5 X 0.5	8	R7003	69.80	-71.82	1.0 X 0.5 X 0.35	5
C7037	87.60	-102.82	1.0 X 0.5 X 0.5	8	R7004	66.40	-71.62	1.0 X 0.5 X 0.35	5
C7046	56.75	-81.92	1.0 X 0.5 X 0.5	9	R7006	66.40	-70.42	1.0 X 0.5 X 0.35	5
C7048	64.50	-75.52	1.0 X 0.5 X 0.5	8	R7040	160.60	-64.62	1.0 X 0.5 X 0.35	4
C7049	59.75	-80.32	1.0 X 0.5 X 0.5	8	D6001	79.00	-86.02	1.6 X 0.8 X 0.8	12
C7050	61.40	-76.42	1.0 X 0.5 X 0.5	8	R1017	157.30	-130.92	1.6 X 0.8 X 0.45	7
C7051	59.75	-76.32	1.0 X 0.5 X 0.5	8	C1001	160.70	-125.32	1.6 X 0.8 X 0.8	13
C7052	66.40	-76.62	1.0 X 0.5 X 0.5	8	C1017	163.10	-122.52	1.6 X 0.8 X 0.8	14
C7054	59.75	-78.12	1.0 X 0.5 X 0.5	8	C1053	16.75	-116.52	1.6 X 0.8 X 0.8	13

FONTE: Autor, 2015.

Tabela 2 - Organização das coordenadas de montagem caso 2.

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
R491	64.90	147.65	1.0 X 0.5 X 0.35	1	C1805	123.25	35.45	1.0 X 0.5 X 0.5	3
BD1106	89.05	58.65	1.0 X 0.5 X 0.5	5	C1806	123.25	24.25	1.0 X 0.5 X 0.5	6
BD1107	94.45	58.65	1.0 X 0.5 X 0.5	5	C1807	122.25	19.05	1.0 X 0.5 X 0.5	6
BD1108	99.65	58.65	1.0 X 0.5 X 0.5	5	C1815	126.45	35.65	1.0 X 0.5 X 0.5	2
BD1201	23.30	88.45	1.0 X 0.5 X 0.5	5	C1817	126.45	24.05	1.0 X 0.5 X 0.5	2
BD1202	36.60	106.00	1.0 X 0.5 X 0.5	5	C202	133.85	123.05	1.0 X 0.5 X 0.5	10
BD1602	152.15	119.75	1.0 X 0.5 X 0.5	5	C206	147.65	125.45	1.0 X 0.5 X 0.5	10
BD201	37.35	140.70	1.0 X 0.5 X 0.5	8	C210	86.45	138.85	1.0 X 0.5 X 0.5	10
BD202	59.05	144.25	1.0 X 0.5 X 0.5	8	C214	80.05	142.05	1.0 X 0.5 X 0.5	10
BD203	37.85	138.45	1.0 X 0.5 X 0.5	8	C215	85.85	143.05	1.0 X 0.5 X 0.5	9
BD204	42.45	127.45	1.0 X 0.5 X 0.5	8	C220	87.65	152.65	1.0 X 0.5 X 0.5	9
BD205	75.85	151.25	1.0 X 0.5 X 0.5	8	C221	140.25	123.05	1.0 X 0.5 X 0.5	9
BD206	187.45	85.45	1.0 X 0.5 X 0.5	8	C222	57.65	98.25	1.0 X 0.5 X 0.5	4
BD207	144.05	125.05	1.0 X 0.5 X 0.5	8	C223	61.45	94.45	1.0 X 0.5 X 0.5	4
BD209	31.25	136.85	1.0 X 0.5 X 0.5	8	C226	71.65	148.25	1.0 X 0.5 X 0.5	4
BD212	131.05	129.85	1.0 X 0.5 X 0.5	8	C227	144.65	123.05	1.0 X 0.5 X 0.5	7
BD215	63.45	93.85	1.0 X 0.5 X 0.5	8	C228	71.65	152.25	1.0 X 0.5 X 0.5	7
BD217	169.85	132.55	1.0 X 0.5 X 0.5	8	C229	103.65	96.65	1.0 X 0.5 X 0.5	21
BD219	62.85	141.65	1.6 X 0.8 X 0.8	8	C230	147.65	127.45	1.0 X 0.5 X 0.5	25
BD220	39.00	140.70	1.6 X 0.8 X 0.8	8	C231	63.45	97.05	1.0 X 0.5 X 0.5	25
BD221	37.85	136.85	1.6 X 0.8 X 0.8	6	C232	67.35	148.45	1.0 X 0.5 X 0.5	29
BD222	59.05	146.05	1.6 X 0.8 X 0.8	6	C233	68.55	148.45	1.0 X 0.5 X 0.5	29
BD303	177.55	139.85	1.6 X 0.8 X 0.8	6	C234	68.55	150.75	1.0 X 0.5 X 0.5	29
BD308	177.40	125.30	1.6 X 0.8 X 0.8	6	C235	106.45	97.25	1.0 X 0.5 X 0.5	12
BD350	88.45	142.65	1.6 X 0.8 X 0.8	6	C236	65.25	97.25	1.0 X 0.5 X 0.5	12
BD351	81.85	148.45	1.6 X 0.8 X 0.8	6	C237	69.75	148.45	1.0 X 0.5 X 0.5	19
BD352	69.20	94.20	1.6 X 0.8 X 0.8	6	C238	69.75	150.75	1.0 X 0.5 X 0.5	19
BD701	125.65	94.85	1.6 X 0.8 X 0.8	6	C239	105.25	97.25	1.0 X 0.5 X 0.5	19
BD806	135.45	80.85	1.6 X 0.8 X 0.8	6	C240	83.45	97.25	1.0 X 0.5 X 0.5	15
BD808	132.35	74.15	1.6 X 0.8 X 0.8	6	C241	85.85	141.85	1.0 X 0.5 X 0.5	15
BD810	124.30	77.55	1.6 X 0.8 X 0.8	6	C242	89.25	139.25	1.0 X 0.5 X 0.5	15
BD862	111.85	68.05	1.6 X 0.8 X 0.8	6	C243	85.85	140.65	1.0 X 0.5 X 0.5	15
BD863	115.50	69.30	1.6 X 0.8 X 0.8	6	C247	39.05	128.05	1.0 X 0.5 X 0.5	30
C1150	101.45	67.05	1.0 X 0.5 X 0.5	11	C251	77.25	143.05	1.0 X 0.5 X 0.5	30
C1201	45.35	98.10	1.0 X 0.5 X 0.5	11	C252	88.65	140.85	1.0 X 0.5 X 0.5	28
C1205	54.55	58.45	1.0 X 0.5 X 0.5	11	C253	39.05	131.55	1.0 X 0.5 X 0.5	28
C1210	54.50	79.70	1.0 X 0.5 X 0.5	11	C254	147.85	122.65	1.0 X 0.5 X 0.5	13
C1214	23.50	86.55	1.0 X 0.5 X 0.5	11	C256	42.25	125.58	1.0 X 0.5 X 0.5	13
C1216	34.80	86.40	1.0 X 0.5 X 0.5	11	C270	33.65	141.05	1.0 X 0.5 X 0.5	14
C1219	29.35	105.80	1.0 X 0.5 X 0.5	11	C278	40.90	129.80	1.0 X 0.5 X 0.5	16
C1221	25.20	98.00	1.0 X 0.5 X 0.5	11	C279	38.25	135.03	1.0 X 0.5 X 0.5	16
C1307	164.45	12.25	1.0 X 0.5 X 0.5	22	C281	62.15	129.35	1.0 X 0.5 X 0.5	17
C1360	160.25	89.25	1.0 X 0.5 X 0.5	22	C282	62.45	130.95	1.0 X 0.5 X 0.5	24
C1365	154.05	99.25	1.0 X 0.5 X 0.5	22	C301	83.05	88.85	1.6 X 0.8 X 0.8	19
C1366	154.05	100.45	1.0 X 0.5 X 0.5	22	C302	84.65	89.45	1.6 X 0.8 X 0.8	19
C1367	156.85	106.45	1.0 X 0.5 X 0.5	22	C303	179.55	140.45	1.6 X 0.8 X 0.8	21
C1370	158.05	106.45	1.0 X 0.5 X 0.5	31	C304	179.40	125.70	1.6 X 0.8 X 0.8	21
C1372	159.25	106.45	1.0 X 0.5 X 0.5	31	C317	185.85	48.65	1.6 X 0.8 X 0.8	18
C1502	146.55	119.95	1.0 X 0.5 X 0.5	26	C321	128.45	116.05	1.6 X 0.8 X 0.8	20
C1508	151.95	104.65	1.0 X 0.5 X 0.5	26	C322	110.05	133.05	1.6 X 0.8 X 0.8	27
C1510	150.45	100.45	1.0 X 0.5 X 0.5	31	C323	106.85	133.05	1.6 X 0.8 X 0.8	24
C1518	132.55	116.55	1.0 X 0.5 X 0.5	31	C324	105.65	133.05	1.6 X 0.8 X 0.8	26
C1521	132.75	110.85	1.0 X 0.5 X 0.5	31	C325	107.65	97.25	1.6 X 0.8 X 0.8	26
C1523	144.05	119.95	1.0 X 0.5 X 0.5	31					
C1524	131.65	119.55	1.0 X 0.5 X 0.5	31					
C1525	150.45	98.45	1.0 X 0.5 X 0.5	3					
C1605	152.15	116.35	1.0 X 0.5 X 0.5	3					
C1804	120.65	36.25	1.0 X 0.5 X 0.5	3					

FONTE: Autor, 2015.

Tabela 3 - Organização das coordenadas de montagem caso 3.

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
C1017	10	136.6	1.0 X 0.5 X 0.5	2	C3025	108.18	178.5	1.0 X 0.5 X 0.5	3
C1018	97.13	138.45	1.0 X 0.5 X 0.5	2	C3026	117.88	91.45	1.0 X 0.5 X 0.5	3
C1022_HN	114.78	136.8	1.0 X 0.5 X 0.5	2	C3100	135.38	170.8	1.0 X 0.5 X 0.5	7
C1023_HN	114.78	135.7	1.0 X 0.5 X 0.5	2	C3101	101.73	170.15	1.0 X 0.5 X 0.5	7
C1101	108.88	108.95	1.0 X 0.5 X 0.5	2	C3102	135.38	169.3	1.0 X 0.5 X 0.5	7
C1201	266.18	121.5	1.0 X 0.5 X 0.5	2	C3103	131.88	182.6	1.0 X 0.5 X 0.5	7
C1206_RST	240.38	18	1.0 X 0.5 X 0.5	2	C3106	105.58	169	1.0 X 0.5 X 0.5	7
C1207	261.18	144.1	1.0 X 0.5 X 0.5	2	C3109	105.58	164.6	1.0 X 0.5 X 0.5	7
C1208	267.38	119.5	1.0 X 0.5 X 0.5	2	C3112	129.93	179.55	1.0 X 0.5 X 0.5	7
C1208_RST	243.68	16.15	1.0 X 0.5 X 0.5	2	C3113	105.58	167.9	1.0 X 0.5 X 0.5	7
C1215	236.13	8.8	1.0 X 0.5 X 0.5	2	C3114	131.73	178.45	1.0 X 0.5 X 0.5	7
C1224_RS	233.88	127.2	1.0 X 0.5 X 0.5	2	C3115	105.58	165.7	1.0 X 0.5 X 0.5	7
C1403	89.88	109	1.0 X 0.5 X 0.5	5	C3116	140.18	169.4	1.0 X 0.5 X 0.5	7
C1404	84.28	100.6	1.0 X 0.5 X 0.5	5	C3117	129.93	178.45	1.0 X 0.5 X 0.5	7
C1405	82.58	109	1.0 X 0.5 X 0.5	5	C3118	145.68	171.6	1.0 X 0.5 X 0.5	7
C1406	89.88	100.35	1.0 X 0.5 X 0.5	5	C3119	141.28	169.4	1.0 X 0.5 X 0.5	7
C1407	66.48	181.8	1.0 X 0.5 X 0.5	5	C312	88.28	49.85	1.0 X 0.5 X 0.5	7
C1412	64.68	180.3	1.0 X 0.5 X 0.5	5	C3120	131.73	180.65	1.0 X 0.5 X 0.5	7
C1413	63.58	180.3	1.0 X 0.5 X 0.5	5	C3121	141.28	171.6	1.0 X 0.5 X 0.5	7
C1414	63.18	181.8	1.0 X 0.5 X 0.5	5	C3122	137.98	169.4	1.0 X 0.5 X 0.5	7
C1415	65.78	180.3	1.0 X 0.5 X 0.5	5	C3123	137.98	171.6	1.0 X 0.5 X 0.5	7
C1417	86.68	109	1.0 X 0.5 X 0.5	5	C3124	143.48	169.4	1.0 X 0.5 X 0.5	7
C1418	85.18	109	1.0 X 0.5 X 0.5	5	C3125	139.08	171.6	1.0 X 0.5 X 0.5	7
C1419	83.68	109	1.0 X 0.5 X 0.5	5	C3126	142.38	169.4	1.0 X 0.5 X 0.5	7
C1420	87.63	100.35	1.0 X 0.5 X 0.5	5	C3127	147.88	171.6	1.0 X 0.5 X 0.5	7
C1422	48.68	142.3	1.0 X 0.5 X 0.5	5	C3128	127.38	182.6	1.0 X 0.5 X 0.5	7
C1423	90.08	141.75	1.0 X 0.5 X 0.5	5	C3129	103.48	157.7	1.0 X 0.5 X 0.5	7
C1424	46.28	144.7	1.0 X 0.5 X 0.5	5	C3130	144.58	169.4	1.0 X 0.5 X 0.5	7
C1426	45.08	145.9	1.0 X 0.5 X 0.5	5	C3131	143.48	171.6	1.0 X 0.5 X 0.5	7
C1427	90.83	144.8	1.0 X 0.5 X 0.5	5	C3132	145.68	169.4	1.0 X 0.5 X 0.5	7
C1428	45.58	102.3	1.0 X 0.5 X 0.5	5	C3133	140.18	171.6	1.0 X 0.5 X 0.5	14
C1429	90.08	142.85	1.0 X 0.5 X 0.5	5	C3134	146.78	169.4	1.0 X 0.5 X 0.5	14
C1430	89.73	144.8	1.0 X 0.5 X 0.5	5	C3137	144.58	171.6	1.0 X 0.5 X 0.5	17
C1431	84.28	99.45	1.0 X 0.5 X 0.5	5	C3138	105.58	160.15	1.0 X 0.5 X 0.5	17
C1432	97.48	148.6	1.0 X 0.5 X 0.5	5	C3139	125.68	180.65	1.0 X 0.5 X 0.5	17
C2956	123.78	184	1.0 X 0.5 X 0.5	1	C3140	103.28	160.45	1.0 X 0.5 X 0.5	17
C2957	125.28	184	1.0 X 0.5 X 0.5	1	C3141	125.68	178.45	1.0 X 0.5 X 0.5	17
C3000	142.98	167.5	1.0 X 0.5 X 0.5	4	C3142	105.58	163.45	1.0 X 0.5 X 0.5	17
C3001	133.03	144.55	1.0 X 0.5 X 0.5	4	C3143	127.78	180.65	1.0 X 0.5 X 0.5	17
C3003	146.58	167.1	1.0 X 0.5 X 0.5	4	C3144	105.58	159.05	1.0 X 0.5 X 0.5	17
C3005	145.48	167.1	1.0 X 0.5 X 0.5	4	C3145	125.68	181.75	1.0 X 0.5 X 0.5	17
C3006	148.78	167.1	1.0 X 0.5 X 0.5	4	C3146	105.58	161.25	1.0 X 0.5 X 0.5	17
C3007	149.88	167.1	1.0 X 0.5 X 0.5	4	C3147	127.78	179.55	1.0 X 0.5 X 0.5	17
C3008	133.53	147	1.0 X 0.5 X 0.5	4	C3148	103.63	161.95	1.0 X 0.5 X 0.5	17
C3009	133.53	145.9	1.0 X 0.5 X 0.5	4	C3149	125.68	179.55	1.0 X 0.5 X 0.5	17
C3012	106.98	184.2	1.0 X 0.5 X 0.5	11	C3200	135.38	173.8	1.0 X 0.5 X 0.5	6
C3013	105.08	178.8	1.0 X 0.5 X 0.5	11	C3201	101.73	179.85	1.0 X 0.5 X 0.5	6
C3014	121.08	178.85	1.0 X 0.5 X 0.5	11	C3203	101.73	171.65	1.0 X 0.5 X 0.5	6
C3015	122.18	178.85	1.0 X 0.5 X 0.5	11	C3205	137.98	174.1	1.0 X 0.5 X 0.5	6
C3016	113.43	178.8	1.0 X 0.5 X 0.5	11	C3207	103.43	180.3	1.0 X 0.5 X 0.5	9
C3017	114.53	178.8	1.0 X 0.5 X 0.5	11	C3208	143.48	174.1	1.0 X 0.5 X 0.5	9
C3018	115.03	180.7	1.0 X 0.5 X 0.5	11	C3209	105.58	172.3	1.0 X 0.5 X 0.5	9
C3019	116.13	180.7	1.0 X 0.5 X 0.5	11	C3210	103.83	177.65	1.0 X 0.5 X 0.5	9
C3020	116.63	178.8	1.0 X 0.5 X 0.5	11	C3211	144.58	174.1	1.0 X 0.5 X 0.5	9
C3021	117.73	178.8	1.0 X 0.5 X 0.5	11	C3212	105.58	173.4	1.0 X 0.5 X 0.5	9
C3022	119.23	180.7	1.0 X 0.5 X 0.5	3	C3213	105.58	176.7	1.0 X 0.5 X 0.5	9
C3023	120.33	180.7	1.0 X 0.5 X 0.5	3	C3214	139.08	174.1	1.0 X 0.5 X 0.5	9
C3024	106.58	178.8	1.0 X 0.5 X 0.5	3	C3215	105.58	170.1	1.0 X 0.5 X 0.5	9

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
C3216	104.23	183.65	1.0 X 0.5 X 0.5	9	C622	185.48	169.95	1.6 X 0.8 X 0.8	12
C3217	135.38	172.3	1.0 X 0.5 X 0.5	9	C625	178.23	191.7	1.6 X 0.8 X 0.8	12
C3218	105.58	171.2	1.0 X 0.5 X 0.5	9	C633	204.78	194.5	1.6 X 0.8 X 0.8	12
C3219	101.73	176.15	1.0 X 0.5 X 0.5	9	C634	205.18	197.2	1.6 X 0.8 X 0.8	12
C3220	141.28	174.1	1.0 X 0.5 X 0.5	9	C635	205.58	180.3	1.6 X 0.8 X 0.8	12
C3221	140.18	174.1	1.0 X 0.5 X 0.5	9	C636	217.38	185.6	1.6 X 0.8 X 0.8	12
C3222	104.23	184.75	1.0 X 0.5 X 0.5	9	C637	216.11	194.48	1.6 X 0.8 X 0.8	12
C3223	104.23	185.85	1.0 X 0.5 X 0.5	9	C638	202.98	191.7	1.6 X 0.8 X 0.8	12
C3224	101.73	177.65	1.0 X 0.5 X 0.5	9	C639	202.68	193	1.6 X 0.8 X 0.8	12
C3227	105.58	175.6	1.0 X 0.5 X 0.5	9	C640	204.48	180.3	1.6 X 0.8 X 0.8	12
C3228	103.43	181.4	1.0 X 0.5 X 0.5	9	C642	217.83	193.6	1.6 X 0.8 X 0.8	15
C323_CPU	47.88	110.3	1.6 X 0.8 X 0.8	8	C645	196.58	178	1.6 X 0.8 X 0.8	15
C324_CPU	47.68	114.2	1.6 X 0.8 X 0.8	8	C646	202.68	194.9	1.6 X 0.8 X 0.8	15
C325_CPU	47.68	113.1	1.6 X 0.8 X 0.8	8	C647	213.86	194.48	1.6 X 0.8 X 0.8	15
C327_CORE	67.78	106.9	1.6 X 0.8 X 0.8	8	C648	206.68	180.3	1.6 X 0.8 X 0.8	15
C328_CORE	47.68	115.3	1.6 X 0.8 X 0.8	8	C649	216.78	196.15	1.6 X 0.8 X 0.8	15
C330_CPU	47.88	111.8	1.6 X 0.8 X 0.8	8	C650	207.78	180.3	1.6 X 0.8 X 0.8	15
C403	231.63	31	1.6 X 0.8 X 0.8	8	C651	206.93	178.4	1.6 X 0.8 X 0.8	15
C405	232.73	36.2	1.6 X 0.8 X 0.8	8	C652	214.98	194.48	1.6 X 0.8 X 0.8	15
C406	237.43	37.8	1.6 X 0.8 X 0.8	8	C653	202.98	182.4	1.6 X 0.8 X 0.8	15
C414	244.38	37.8	1.6 X 0.8 X 0.8	8	C654	209.28	180.3	1.6 X 0.8 X 0.8	15
C416	237.68	23.5	1.6 X 0.8 X 0.8	8	C655	203.08	180.4	1.6 X 0.8 X 0.8	15
C421	232.88	27	1.6 X 0.8 X 0.8	10	C656	217.38	189.2	1.6 X 0.8 X 0.8	15
C422	246.38	37.8	1.6 X 0.8 X 0.8	10	C657	214.38	179.3	1.6 X 0.8 X 0.8	15
C423	246.38	22.1	1.6 X 0.8 X 0.8	10	C658	215.68	179.75	1.6 X 0.8 X 0.8	15
C429	249.28	37.8	1.6 X 0.8 X 0.8	10	C659	202.68	196.8	1.6 X 0.8 X 0.8	15
C430	249.28	22.1	1.6 X 0.8 X 0.8	10	C660	209.58	195.65	1.6 X 0.8 X 0.8	15
C445	135.08	194.5	1.6 X 0.8 X 0.8	10	C661	206.83	197.7	1.6 X 0.8 X 0.8	15
C600	179.98	164.7	1.6 X 0.8 X 0.8	12	C662	217.38	184.4	1.6 X 0.8 X 0.8	15
C601	161.38	164.7	1.6 X 0.8 X 0.8	12	C663	217.38	190.3	3.2 X 1.6 X 1.6	13
C602	164.03	191.7	1.6 X 0.8 X 0.8	12	C664	217.38	183.3	3.2 X 1.6 X 1.6	13
C603	162.48	164.7	1.6 X 0.8 X 0.8	12	C665	202.98	183.5	3.2 X 1.6 X 1.6	13
C604	185.48	182.45	1.6 X 0.8 X 0.8	12	C666	217.18	179.75	3.2 X 1.6 X 1.6	13
C606	163.58	164.7	1.6 X 0.8 X 0.8	12	C667	202.98	190.1	3.2 X 1.6 X 1.6	13
C607	158.38	169.25	1.6 X 0.8 X 0.8	12	C668	216.38	177.35	3.2 X 1.6 X 1.6	16
C609	164.68	164.7	1.6 X 0.8 X 0.8	12	C704	198.38	199.4	3.2 X 1.6 X 1.6	16
C616	166.23	191.7	1.6 X 0.8 X 0.8	12	C705	219.33	154.4	3.2 X 1.6 X 1.6	16
C619	185.48	180.45	1.6 X 0.8 X 0.8	12	C706	221.33	154.65	3.2 X 1.6 X 1.6	16

FONTE: Autor, 2015.

Tabela 4 - Organização das coordenadas de montagem caso 4.

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
BD1100	11.25	192	1.0 X 0.5 X 0.5	19	C1358	117.15	129.5	1.0 X 0.5 X 0.5	4
BD1101	11.25	195.6	1.0 X 0.5 X 0.5	19	C1360	119.95	130.5	1.0 X 0.5 X 0.5	4
BD1102	11.25	199.2	1.0 X 0.5 X 0.5	19	C1362	114.35	119.1	1.0 X 0.5 X 0.5	4
BD1120	11.43	218.75	1.0 X 0.5 X 0.5	11	C1363	114.35	115.3	1.0 X 0.5 X 0.5	4
BD1121	17.35	218.75	1.0 X 0.5 X 0.5	11	C1365	118.35	129.5	1.0 X 0.5 X 0.5	4
BD1122	14.35	218.75	1.0 X 0.5 X 0.5	11	C1366	123.45	129.5	1.0 X 0.5 X 0.5	4
BD1302	129.45	123.2	1.0 X 0.5 X 0.5	9	C1367	127.7	126.8	1.0 X 0.5 X 0.5	4
BD1305	80.15	93.9	1.0 X 0.5 X 0.5	9	C1369	127.65	118.7	1.0 X 0.5 X 0.5	4
BD1401	89.25	76	1.0 X 0.5 X 0.5	2	C1370	127.65	113.1	1.0 X 0.5 X 0.5	4
BD1402	87.45	76	1.0 X 0.5 X 0.5	2	C1401	83.25	75.8	1.0 X 0.5 X 0.5	4
BD201	127.6	68.7	3.2 X 1.6 X 1.6	5	C1403	85.45	75.8	1.0 X 0.5 X 0.5	4
BD202	124.45	72.7	3.2 X 1.6 X 1.6	5	C1601	196.75	207.45	1.0 X 0.5 X 0.5	4
BD203	127.6	65.3	3.2 X 1.6 X 1.6	5	C1602	181.95	207.45	1.0 X 0.5 X 0.5	4
BD204	127.6	58.7	3.2 X 1.6 X 1.6	5	C1607	200.65	217.95	1.0 X 0.5 X 0.5	4
BD220	84.95	90.4	3.2 X 1.6 X 1.6	5	C1611	188.75	216.9	1.0 X 0.5 X 0.5	4
BD230	90.15	195.2	3.2 X 1.6 X 1.6	5	C1618	197.85	213.85	1.0 X 0.5 X 0.5	4
BD231	71.15	195.55	3.2 X 1.6 X 1.6	5	C1619	182.85	214.2	1.0 X 0.5 X 0.5	4
BD240	95.85	81.4	3.2 X 1.6 X 1.6	5	C1690	154.8	214.8	1.0 X 0.5 X 0.5	4
BD250	107.9	138.25	3.2 X 1.6 X 1.6	5	C202	125.85	73.8	1.0 X 0.5 X 0.5	4
BD270	32.6	199.3	3.2 X 1.6 X 1.6	5	C203	129.65	65.6	1.0 X 0.5 X 0.5	4
BD280	41.65	150.6	1.6 X 0.8 X 0.8	7	C206	129.65	53.4	1.0 X 0.5 X 0.5	6
BD281	41.65	154.6	1.6 X 0.8 X 0.8	7	C220	94.85	86.2	1.0 X 0.5 X 0.5	6
BD408	46.65	59.2	1.6 X 0.8 X 0.8	7	C221	92.05	85.3	1.0 X 0.5 X 0.5	6
BD410	45.55	189.95	1.6 X 0.8 X 0.8	7	C232	83.05	195.2	1.0 X 0.5 X 0.5	6
BD420	45.55	192.5	1.6 X 0.8 X 0.8	7	C234	71.15	192.4	1.0 X 0.5 X 0.5	6
BD430	82.45	159	1.6 X 0.8 X 0.8	7	C235	73.15	192.2	1.0 X 0.5 X 0.5	6
BD440	43.75	200	1.6 X 0.8 X 0.8	7	C236	73.15	193.3	1.0 X 0.5 X 0.5	6
BD450	43.35	185.7	1.6 X 0.8 X 0.8	7	C240	92.75	78.3	1.0 X 0.5 X 0.5	6
BD460	43.35	183.9	1.6 X 0.8 X 0.8	7	C241	92.75	76.25	1.0 X 0.5 X 0.5	6
BD470	158.85	212.2	1.6 X 0.8 X 0.8	7	C256	94.25	142.4	1.0 X 0.5 X 0.5	6
BD630	79.65	172.9	1.6 X 0.8 X 0.8	18	C257	98.65	142.6	1.0 X 0.5 X 0.5	6
BD650	73.85	198	1.6 X 0.8 X 0.8	18	C258	95.85	141.6	1.0 X 0.5 X 0.5	6
BD651	76.65	196	1.6 X 0.8 X 0.8	18	C259	98.65	140.6	1.0 X 0.5 X 0.5	6
BD810	65.65	143.8	1.6 X 0.8 X 0.8	18	C261	97.05	152.8	1.0 X 0.5 X 0.5	6
BD900	41.55	121.3	1.6 X 0.8 X 0.8	18	C262	91.85	152.4	1.0 X 0.5 X 0.5	6
BD920	47.45	129.6	1.6 X 0.8 X 0.8	18	C263	116.55	150.9	1.0 X 0.5 X 0.5	6
BD940	62.75	112.5	1.6 X 0.8 X 0.8	18	C270	36.02	201.86	1.0 X 0.5 X 0.5	6
BD941	66.65	99.7	1.6 X 0.8 X 0.8	18	C271	39.45	201.87	1.0 X 0.5 X 0.5	6
BD960	30.53	134.9	1.6 X 0.8 X 0.8	18	C276	39.9	182.1	1.0 X 0.5 X 0.5	6
BD961	43.45	135.8	1.6 X 0.8 X 0.8	18	C277	40.15	185.8	1.0 X 0.5 X 0.5	6
C1030	159.15	210.8	1.0 X 0.5 X 0.5	3	C278	40.15	187.85	1.0 X 0.5 X 0.5	6
C1031	169.15	215.3	1.0 X 0.5 X 0.5	3	C279	40.15	183.75	1.0 X 0.5 X 0.5	6
C1100	22.65	176.45	1.0 X 0.5 X 0.5	3	C280	27.65	152	1.0 X 0.5 X 0.5	6
C1140	109.05	156.1	1.0 X 0.5 X 0.5	3	C284	30.3	177.25	1.0 X 0.5 X 0.5	8
C1150	21.3	185.8	1.0 X 0.5 X 0.5	3	C286	46.75	153.5	1.0 X 0.5 X 0.5	8
C1210	28.05	210.35	1.0 X 0.5 X 0.5	3	C287	46.75	154.6	1.0 X 0.5 X 0.5	8
C1220	53.15	214.6	1.0 X 0.5 X 0.5	3	C302	117.75	192.4	1.0 X 0.5 X 0.5	8
C1230	80.65	215	1.0 X 0.5 X 0.5	3	C305	117.85	171.95	1.0 X 0.5 X 0.5	8
C1301	74.35	94	1.0 X 0.5 X 0.5	15	C306	117.85	169.4	1.0 X 0.5 X 0.5	8
C1302	79.55	90.35	1.0 X 0.5 X 0.5	15	C307	91.45	159.2	1.0 X 0.5 X 0.5	8
C1303	69.55	94	1.0 X 0.5 X 0.5	15	C308	92.05	161	1.0 X 0.5 X 0.5	8
C1304	66.05	90.6	1.0 X 0.5 X 0.5	15	C313	117.05	153.8	1.0 X 0.5 X 0.5	8
C1308	71.95	96	1.0 X 0.5 X 0.5	15	C399	117.05	152.6	1.0 X 0.5 X 0.5	10
C1309	114.35	124.6	1.0 X 0.5 X 0.5	15	C400	45.05	56.8	1.0 X 0.5 X 0.5	10
C1310	70.75	94	1.0 X 0.5 X 0.5	15	C401	45.05	62	1.0 X 0.5 X 0.5	10
C1326	66.15	84.1	1.0 X 0.5 X 0.5	4	C410	45.35	188.1	1.0 X 0.5 X 0.5	10
C1327	66.15	80.5	1.0 X 0.5 X 0.5	4	C411	48.05	188	1.0 X 0.5 X 0.5	10
C1331	79.55	77.4	1.0 X 0.5 X 0.5	4	C412	49.25	188	1.0 X 0.5 X 0.5	10
C1334	79.55	83.7	1.0 X 0.5 X 0.5	4	C413	50.25	190.3	1.0 X 0.5 X 0.5	10

## Continuação

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
C414	48.05	190.3	1.0 X 0.5 X 0.5	10	C961	34.65	135.3	1.6 X 0.8 X 0.8	12
C415	49.15	190.3	1.0 X 0.5 X 0.5	10	R1001	13.85	136.2	1.0 X 0.5 X 0.35	9
C420	45.35	194.35	1.0 X 0.5 X 0.5	10	R1002	14.5	127	1.0 X 0.5 X 0.35	9
C421	48.05	192.1	1.0 X 0.5 X 0.5	10	R1003	14.5	130.6	1.0 X 0.5 X 0.35	9
C422	49.15	192.1	1.0 X 0.5 X 0.5	10	R1013	26.35	166.8	1.0 X 0.5 X 0.35	9
C423	48.05	194.5	1.0 X 0.5 X 0.5	10	R1023	26.35	172.8	1.0 X 0.5 X 0.35	9
C424	49.15	194.5	1.0 X 0.5 X 0.5	10	R1100	11.55	190.5	1.0 X 0.5 X 0.35	17
C430	80.65	158.9	1.0 X 0.5 X 0.5	10	R1101	11.55	194.1	1.0 X 0.5 X 0.35	17
C431	80.65	162.1	1.0 X 0.5 X 0.5	10	R1102	11.55	197.7	1.0 X 0.5 X 0.35	17
C432	81.75	162.1	1.0 X 0.5 X 0.5	10	R1105	15.45	183.4	1.0 X 0.5 X 0.35	17
C440	43.95	201.4	1.0 X 0.5 X 0.5	10	R1109	15.45	184.6	1.0 X 0.5 X 0.35	14
C441	43.95	202.5	1.0 X 0.5 X 0.5	10	R1110	15.45	185.8	1.0 X 0.5 X 0.35	14
C450	47.65	185.95	1.0 X 0.5 X 0.5	10	R1111	15.45	182.2	1.0 X 0.5 X 0.35	14
C451	50.55	185.5	1.0 X 0.5 X 0.5	10	R1120	16.65	217	1.0 X 0.5 X 0.35	14
C452	51.65	185.5	1.0 X 0.5 X 0.5	10	R1121	14.55	215.2	1.0 X 0.5 X 0.35	14
C453	52.75	185.5	1.0 X 0.5 X 0.5	10	R1131	14.9	211.4	1.0 X 0.5 X 0.35	14
C461	51.9	181.3	1.0 X 0.5 X 0.5	10	R1132	14.9	210.3	1.0 X 0.5 X 0.35	14
C462	51.9	183.6	1.0 X 0.5 X 0.5	10	R1133	17.65	208.8	1.0 X 0.5 X 0.35	14
C463	45.65	182.15	1.0 X 0.5 X 0.5	10	R1141	117.05	157.1	1.0 X 0.5 X 0.35	14
C465	44.55	182.15	1.0 X 0.5 X 0.5	10	R1144	109.05	155	1.0 X 0.5 X 0.35	14
C466	43.45	182.15	1.0 X 0.5 X 0.5	10	R1145	109.05	153.9	1.0 X 0.5 X 0.35	13
C467	51.9	182.45	1.0 X 0.5 X 0.5	10	R1146	109.05	157.2	1.0 X 0.5 X 0.35	13
C580	50.45	192.65	1.6 X 0.8 X 0.8	16	R1147	109.45	167.6	1.0 X 0.5 X 0.35	13
C640	39.05	124.1	1.6 X 0.8 X 0.8	16	R1150	20.1	185.8	1.0 X 0.5 X 0.35	13
C641	82.25	172.3	1.6 X 0.8 X 0.8	16	R1212	33.65	219	1.0 X 0.5 X 0.35	13
C642	84.85	172.3	1.6 X 0.8 X 0.8	16	R1214	34.85	219	1.0 X 0.5 X 0.35	13
C651	74.25	196	1.6 X 0.8 X 0.8	16	R1215	27.65	217.7	1.0 X 0.5 X 0.35	13
C660	29.25	186	1.6 X 0.8 X 0.8	16	R1222	60.65	219	1.0 X 0.5 X 0.35	13
C700	30.75	160.8	1.6 X 0.8 X 0.8	12	R1224	61.75	219	1.0 X 0.5 X 0.35	13
C701	28.65	160.8	1.6 X 0.8 X 0.8	12	R1225	53.15	218.8	1.0 X 0.5 X 0.35	13
C702	32.4	160.6	1.6 X 0.8 X 0.8	12	R1232	87.65	219	1.0 X 0.5 X 0.35	13
C707	37.85	160.6	1.6 X 0.8 X 0.8	12	R1234	88.75	219	1.0 X 0.5 X 0.35	13
C708	31.15	163.1	1.6 X 0.8 X 0.8	12	R1235	80.65	218.8	1.0 X 0.5 X 0.35	13
C709	37.45	177.7	1.6 X 0.8 X 0.8	12	R1244	115.75	219	1.0 X 0.5 X 0.35	13
C720	53.95	133.3	1.6 X 0.8 X 0.8	12	R1301	66.05	91.8	1.0 X 0.5 X 0.35	13
C721	48.45	138.4	1.6 X 0.8 X 0.8	12	R1319	114.35	125.7	1.0 X 0.5 X 0.35	13
C722	58.55	133.4	1.6 X 0.8 X 0.8	12	R1601	182.45	209.15	1.0 X 0.5 X 0.35	9
C727	46.75	150.2	1.6 X 0.8 X 0.8	12	R1602	198.05	209.15	1.0 X 0.5 X 0.35	9
C728	46.75	151.3	1.6 X 0.8 X 0.8	12	R1690	152.65	214.8	1.0 X 0.5 X 0.35	9
C729	49.25	145	1.6 X 0.8 X 0.8	12	R201	125.85	71.6	1.0 X 0.5 X 0.35	19
C810	65.25	139.6	1.6 X 0.8 X 0.8	12	R205	121.65	64.75	1.0 X 0.5 X 0.35	1
C811	65.25	140.7	1.6 X 0.8 X 0.8	12	R206	129.65	63.6	1.0 X 0.5 X 0.35	1
C812	65.25	138.5	1.6 X 0.8 X 0.8	12	R208	124.05	61.45	1.0 X 0.5 X 0.35	1
C902	25.45	121.7	1.6 X 0.8 X 0.8	12	R212	128.25	95.25	1.0 X 0.5 X 0.35	1
C903	27.35	120.8	1.6 X 0.8 X 0.8	12	R242	84.45	83.4	1.0 X 0.5 X 0.35	1
C904	24.25	124.5	1.6 X 0.8 X 0.8	12	R243	84.45	84.6	1.0 X 0.5 X 0.35	1
C905	43.35	120.6	1.6 X 0.8 X 0.8	12	R250	101.5	140.9	1.0 X 0.5 X 0.35	1
C906	43.45	123	1.6 X 0.8 X 0.8	12	R251	105.1	140.9	1.0 X 0.5 X 0.35	1
C908	36.25	120.7	1.6 X 0.8 X 0.8	12	R252	102.7	140.9	1.0 X 0.5 X 0.35	1
C909	33.05	120.7	1.6 X 0.8 X 0.8	12	R254	108.9	140.9	1.0 X 0.5 X 0.35	1
C910	48.55	122.6	1.6 X 0.8 X 0.8	12	R255	107.5	140.9	1.0 X 0.5 X 0.35	1
C911	49.05	121	1.6 X 0.8 X 0.8	12	R260	95.05	152.8	1.0 X 0.5 X 0.35	1
C912	61.45	87.9	1.6 X 0.8 X 0.8	12	R270	41.42	190.31	1.0 X 0.5 X 0.35	1
C913	61.95	86.3	1.6 X 0.8 X 0.8	12	R271	41.45	193.3	1.0 X 0.5 X 0.35	1
C914	62.65	81.1	1.6 X 0.8 X 0.8	12	R274	41.42	196.31	1.0 X 0.5 X 0.35	1
C915	62.25	78.8	1.6 X 0.8 X 0.8	12	R300	117	197.35	1.0 X 0.5 X 0.35	1
C923	48.15	132.2	1.6 X 0.8 X 0.8	12	R301	104.25	185.4	1.0 X 0.5 X 0.35	1
C950	44.85	72.35	1.6 X 0.8 X 0.8	12	R302	105.45	185.4	1.0 X 0.5 X 0.35	1
C951	59	72.15	1.6 X 0.8 X 0.8	12	R303	106.65	185.4	1.0 X 0.5 X 0.35	1

FONTE: Autor, 2015.

Tabela 5 - Organização das coordenadas de montagem caso 5.

Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo	Posição Mecânica	Eixo X	Eixo Y	Dimensões	Tipo
C1002	73.68	66.8	1.0 X 0.5 X 0.5	5	C1165	48.21	126.8	1.0 X 0.5 X 0.5	12
C1003	86.37	120.8	1.0 X 0.5 X 0.5	5	C1167	80.26	132.8	1.0 X 0.5 X 0.5	12
C1008	86.41	123	1.0 X 0.5 X 0.5	9	C1203	173.01	19.4	1.0 X 0.5 X 0.5	10
C1013	64.27	90.35	1.0 X 0.5 X 0.5	7	C1213	155.61	22.2	1.0 X 0.5 X 0.5	10
C1014	69.56	68.5	1.0 X 0.5 X 0.5	7	C1218	154.41	115.6	1.0 X 0.5 X 0.5	10
C1015	79.41	69.9	1.0 X 0.5 X 0.5	7	C1222	141.81	125	1.0 X 0.5 X 0.5	10
C1016	77.01	69.9	1.0 X 0.5 X 0.5	7	C1302	41.96	82.35	1.0 X 0.5 X 0.5	4
C1017	76.16	90.2	1.0 X 0.5 X 0.5	7	C1416	68.71	87.5	1.0 X 0.5 X 0.5	4
C1019	68.16	64.85	1.0 X 0.5 X 0.5	7	C1501	173.61	24.35	1.0 X 0.5 X 0.5	4
C1021	99.51	118.65	1.0 X 0.5 X 0.5	11	C210	110.66	147.45	1.0 X 0.5 X 0.5	4
C1024	80.26	127.65	1.0 X 0.5 X 0.5	11	C231	94.36	142.3	1.6 X 0.8 X 0.8	6
C1025	57.01	137.8	1.0 X 0.5 X 0.5	11	C301	87.41	13.2	1.6 X 0.8 X 0.8	6
C1027	64.36	69.95	1.0 X 0.5 X 0.5	13	C305	5.61	140.4	1.6 X 0.8 X 0.8	6
C1028	67.06	64.85	1.0 X 0.5 X 0.5	13	C314	160.81	51	1.6 X 0.8 X 0.8	6
C1029	67.11	87.25	1.0 X 0.5 X 0.5	13	C315	158.81	52.6	1.6 X 0.8 X 0.8	6
C1030	66.91	90.1	1.0 X 0.5 X 0.5	13	C318	167.91	57.15	1.6 X 0.8 X 0.8	6
C1031	99.51	120.85	1.0 X 0.5 X 0.5	13	C319	167.91	55.35	1.6 X 0.8 X 0.8	6
C1032	100.61	120.85	1.0 X 0.5 X 0.5	13	C322	161.21	52.6	1.6 X 0.8 X 0.8	6
C1033	100.61	118.65	1.0 X 0.5 X 0.5	13	C339	164.21	41.4	1.6 X 0.8 X 0.8	6
C1034	83.61	119.8	1.0 X 0.5 X 0.5	13	C340	161.01	41.4	1.6 X 0.8 X 0.8	6
C1035	83.61	121	1.0 X 0.5 X 0.5	13	C341	167.81	41.2	1.6 X 0.8 X 0.8	6
C1036	83.61	122.2	1.0 X 0.5 X 0.5	13	C342	170.81	41.2	1.6 X 0.8 X 0.8	6
C1037	83.61	123.4	1.0 X 0.5 X 0.5	13	C367	159.21	50.6	1.6 X 0.8 X 0.8	6
C1038	88.81	119.8	1.0 X 0.5 X 0.5	13	C368	162.41	55.35	1.6 X 0.8 X 0.8	6
C1039	88.81	121	1.0 X 0.5 X 0.5	13	C404	114.56	86.4	1.6 X 0.8 X 0.8	6
C1040	88.81	122.2	1.0 X 0.5 X 0.5	13	C407	140.01	120.6	1.6 X 0.8 X 0.8	6
C1041	88.81	123.4	1.0 X 0.5 X 0.5	13	C601	167.56	95.15	1.6 X 0.8 X 0.8	6
C1105	67.61	145	1.0 X 0.5 X 0.5	1					
C1106	28.61	114.6	1.0 X 0.5 X 0.5	1					
C1107	59.21	138.6	1.0 X 0.5 X 0.5	1					
C1108	35.61	104.45	1.0 X 0.5 X 0.5	1					
C1117	74.41	141.2	1.0 X 0.5 X 0.5	3					
C1118	77.41	138.75	1.0 X 0.5 X 0.5	3					
C1123	74.41	139.6	1.0 X 0.5 X 0.5	8					
C1124	53.11	115	1.0 X 0.5 X 0.5	8					
C1125	74.41	131.9	1.0 X 0.5 X 0.5	8					
C1126	80.26	135.15	1.0 X 0.5 X 0.5	8					
C1142	43.61	106.8	1.0 X 0.5 X 0.5	2					
C1143	42.51	120	1.0 X 0.5 X 0.5	2					
C1144	38.16	120	1.0 X 0.5 X 0.5	2					
C1145	74.41	138.2	1.0 X 0.5 X 0.5	2					
C1146	35.46	120	1.0 X 0.5 X 0.5	2					
C1147	74.41	135.8	1.0 X 0.5 X 0.5	2					
C1148	34.01	120	1.0 X 0.5 X 0.5	2					
C1149	61.61	139.2	1.0 X 0.5 X 0.5	2					
C1150	32.41	120	1.0 X 0.5 X 0.5	2					
C1151	39.86	125.3	1.0 X 0.5 X 0.5	2					
C1152	36.01	106.8	1.0 X 0.5 X 0.5	2					
C1153	68.91	121.75	1.0 X 0.5 X 0.5	2					
C1154	34.21	106.8	1.0 X 0.5 X 0.5	2					
C1155	28.81	106.6	1.0 X 0.5 X 0.5	2					
C1156	32.41	106.8	1.0 X 0.5 X 0.5	2					
C1157	34.46	125.3	1.0 X 0.5 X 0.5	2					
C1158	42.41	106.8	1.0 X 0.5 X 0.5	2					
C1159	36.81	125.3	1.0 X 0.5 X 0.5	2					
C1160	61.61	129.6	1.0 X 0.5 X 0.5	2					
C1161	61.61	141	1.0 X 0.5 X 0.5	2					
C1162	62.61	145.4	1.0 X 0.5 X 0.5	2					
C1163	45.81	126.8	1.0 X 0.5 X 0.5	2					

FONTE: Autor, 2015.

## ARTIGOS PUBLICADOS

### ARTIGO PUBLICADO EM CONGRESSO INTERNACIONAL

Castro, A. O.; Bezerra, U. H.; Leite, J.C. - **Metodología propuesta para la optimización multicriterio mediante algoritmos genéticos en las aplicaciones industriales** - 17<sup>a</sup> CUJAE - Convención Científica de Ingeniería y Arquitectura, Havana, Cuba (2014).

### PUBLICADO EM CONGRESSO NACIONAL

Castro, A.O.; Bezerra, U.H.; Leite, J.C.; Azevedo, M.S.S. - **Methodology proposal for multicriteria optimization using NSGA-II in industrial applications** - Industry Applications (INDUSCON), 2014 11th IEEE/IAS International Conference, Juiz de Fora, Brasil (2014).