

UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA – PIPCA

DIOGO ALBERTO BORGES

**UMA SOLUÇÃO INTEGRADA APLICADA AO PROBLEMA DE OTIMIZAÇÃO DO
CICLO DE MONTAGEM DE UMA INSERSORA AUTOMÁTICA DE
COMPONENTES UTILIZANDO UMA ABORDAGEM HÍBRIDA DE
METAHEURÍSTICAS**

São Leopoldo
2009

DIOGO ALBERTO BORGES

**UMA SOLUÇÃO INTEGRADA APLICADA AO PROBLEMA DE OTIMIZAÇÃO DO
CICLO DE MONTAGEM DE UMA INSERSORA AUTOMÁTICA DE
COMPONENTES UTILIZANDO UMA ABORDAGEM HÍBRIDA DE
METAHEURÍSTICAS**

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

Orientador: Prof. Dr. Arthur Tórgo Gómez

São Leopoldo
2009

AGRADECIMENTOS

Agradeço,

À minha esposa Bruna Frantz, pelas aulas de português, tempo dedicado a esta dissertação, apoio nos momentos difíceis e, principalmente, por doar muito tempo dedicado a ela, para redação desta dissertação;

À minha amiga Suzi, que me fez companhia nas muitas tardes de redação, pesquisa e codificação;

Ao professor Arthur Tórgo Gómez, pela grande motivação, por acreditar em minha capacidade de desenvolver esta pesquisa, mesmo nos momentos em que eu mesmo duvidei ser capaz, pelo apoio, sugestões e tudo que aprendi com ele nesta fase da minha vida;

Aos professores Leonardo Chiwiacowsky e José Vicente Canto dos Santos que, ao avaliarem a proposta e seminário de andamento, contribuíram com sugestões importantes que melhoraram a qualidade do trabalho e seus resultados, enfim ajudaram a tornar este trabalho como um todo melhor.

Muito obrigado a todos.

RESUMO

Este trabalho aborda o problema da otimização relacionado ao ciclo de operação de uma insersora automática de componentes (*pick-and-place*) em máquinas do tipo *multi-head*. O objetivo consiste em desenvolver uma técnica computacional capaz de encontrar uma boa solução para o problema de otimização, associado ao processo de inserção de componentes. A otimização consiste em resolver de forma conjunta, os Problemas de Escalonamento das Ferramentas, de Escalonamento da Seqüência de Inserção de Componentes e por fim, o Problema da Alocação dos Componentes no Alimentador, visando a redução do tempo total de operação. Como resultados, a técnica computacional permite elevar a produtividade das máquinas onde a mesma é aplicada. Neste trabalho, foi desenvolvida uma aplicação híbrida das metaheurísticas Busca Tabu e Algoritmos Genéticos. Através do uso de uma abordagem diferenciada e utilizando metaheurísticas contemporâneas, bons resultados são apresentados ao longo do trabalho.

Palavras-chave: *pick-and-place*, escalonamento, *multi-head*, metaheurísticas.

ABSTRACT

This work approaches the optimization Problem related to the operation cycle in an automatic multi-head SMT placement machine (pick-and-place). The objective of this work is developing a computational technique capable to find a good solution for the optimization problem associated with the component insertion process. The optimization process was planned to solve in a joint way the Tools Scheduling Problem, Schedule of Component Insertion Sequence Problem and finally the Component Feeder Allocation Problem. The main objective is to reduce the total operation time. As a result the computational technique can increase the productivity of the machines where it is applied. This study developed a hybrid application of Tabu Search and Genetic Algorithms. The use of a differentiated approach combined with modern metaheuristics, good results are presented in the work.

Keywords: pick-and-place, scheduling, multi-head, metaheuristics.

SUMÁRIO

RESUMO	4
ABSTRACT	5
SUMÁRIO	6
ÍNDICE DE FIGURAS	10
ÍNDICE DE TABELAS	12
LISTA DE ABREVIATURAS	13
INTRODUÇÃO	14
1 CONCEITOS DE TECNOLOGIA DE MONTAGEM DE CIRCUITOS ELETRÔNICOS...	17
1.1 TECNOLOGIA DE CIRCUITO IMPRESSO	17
1.2 TECNOLOGIAS PTH E TH.....	18
1.3 TECNOLOGIA DE MONTAGEM EM SUPERFÍCIE (SMT)	19
1.3.1 <i>Descrição</i>	20
1.3.2 <i>Processo de Montagem em Tecnologia Surface Mount (SMT)</i>	20
1.4 INSERSORAS AUTOMÁTICAS DE COMPONENTES (<i>PICK-AND-PLACE</i>)	22
1.4.1 <i>Máquinas Dual-Delivery</i>	23
1.4.2 <i>Máquinas Tipo Multi-Estação</i>	24
1.4.3 <i>Máquinas Tipo Turret</i>	25
1.4.4 <i>Máquinas Multi-Head</i>	25
1.4.5 <i>Máquinas com Operação Seqüencial de Carga e Inserção</i>	27
1.5 PROGRAMAS DE MONTAGEM UTILIZADOS EM MÁQUINAS INSERSORAS AUTOMÁTICAS DE COMPONENTES	27
2 FUNDAMENTAÇÃO TEÓRICA	28

2.1	COMPLEXIDADE COMPUTACIONAL DE ALGORITMOS.....	28
2.2	CIRCUITO HAMILTONIANO	30
2.3	HEURÍSTICAS	31
2.4	METAHEURÍSTICAS	32
2.4.1	<i>Busca Tabu</i>	33
2.4.2	<i>Algoritmos Genéticos</i>	35
2.5	TECNOLOGIA DE GRUPO	38
2.5.1	<i>Família de Partes</i>	38
2.5.2	<i>Separação em Famílias de Partes Utilizando o Método de Ordenação Rank-Order Clustering (ROC)</i>	39
2.6	ESCALONAMENTO DE PARTES	40
2.6.1	<i>Formulação Clássica</i>	41
3	REVISÃO BIBLIOGRÁFICA.....	43
3.1	TRABALHOS APLICADOS A MÁQUINAS <i>MULTI-HEAD</i>	44
3.2	TRABALHOS APLICADOS A MÁQUINAS DE INSERÇÃO SEQUENCIAL	47
3.3	ABORDAGENS A OUTROS TIPOS DE MÁQUINA UTILIZANDO ALGORITMOS GENÉTICOS.....	48
3.4	ABORDAGENS A OUTROS TIPOS DE MÁQUINA UTILIZANDO BUSCA TABU.....	48
3.5	CONSIDERAÇÕES DESTA PESQUISA EM RELAÇÃO AOS TRABALHOS APRESENTADOS	49
4	APRESENTAÇÃO DO PROBLEMA	50
4.1	PROJETO NO CAD DE PLACAS DE CIRCUITO IMPRESSO	50
4.2	INTEGRAÇÃO ENTRE O PROJETO DO CAD E A <i>PICK-AND-PLACE</i> (CAM)	51
4.3	O PROBLEMA DE OTIMIZAÇÃO A SER RESOLVIDO – CONTEXTO SEPARADO	52
4.3.1	<i>Determinação do Conjunto Ideal de Ferramentas e Escalonamento das Mesmas</i>	52
4.3.2	<i>Problema da Alocação de Componentes no Alimentador de Componentes</i>	52
4.3.3	<i>Ordem de Execução das Inserções de Componentes</i>	52
4.4	O PROBLEMA DE OTIMIZAÇÃO A SER RESOLVIDO – CONTEXTO GLOBAL	53
5	MODELO PROPOSTO	54
5.1	PROBLEMA ESTUDADO PARA DEFINIÇÃO DO MODELO.....	54
5.2	FORMULAÇÃO DO MODELO	57
5.2.1	<i>Considerações Sobre a Função Objetivo</i>	62
5.2.2	<i>Caracterização dos Parâmetros da Máquina Insersora</i>	62
5.3	ARQUITETURA DO MODELO.....	63
6	IMPLEMENTAÇÃO DO MODELO E ALGORITMO	67
6.1	PROCESSAMENTO INICIAL DOS ARQUIVOS DO CAD	67

6.2	ESTRUTURAS DE DADOS BÁSICAS PARA DEFINIÇÃO DOS PARÂMETROS DO PROBLEMA.....	67
6.2.1	<i>Posição dos Componentes na PCB</i>	68
6.2.2	<i>Alimentador onde cada Componente Está Localizado</i>	68
6.2.3	<i>Posição Física do Alimentador na Máquina</i>	69
6.2.4	<i>Ferramenta Associada a cada Componente</i>	69
6.2.5	<i>Correspondência entre Endereço lógico e Endereço Físico de um Alimentador</i>	70
6.3	ESTRUTURAS DE APOIO À IMPLEMENTAÇÃO DO ALGORITMO.....	70
6.4	FORMATO DAS SOLUÇÕES	71
6.4.1	<i>Solução Reduzida</i>	71
6.4.2	<i>Solução Completa</i>	73
6.5	AVALIAÇÃO DA SOLUÇÃO	75
6.6	AVALIAÇÃO DA SIMILARIDADE	75
6.7	AVALIAÇÃO DA VALIDADE DE UMA SOLUÇÃO	77
6.8	IMPLEMENTAÇÃO DA METAHEURÍSTICA HÍBRIDA	79
6.9	BUSCA TABU – VISÃO GERAL	80
6.9.1	<i>Lista Tabu</i>	81
6.9.2	<i>Geração de Vizinhanças</i>	81
6.9.3	<i>Sub-problema da Seqüência de Execução dos Carregamentos</i>	82
6.9.4	<i>Critério de Aspiração</i>	82
6.9.5	<i>Critério de Parada</i>	83
6.10	ALGORITMO GENÉTICO – VISÃO GERAL.....	83
6.10.1	<i>Codificação do Cromossomo</i>	84
6.10.2	<i>Operação de Cruzamento</i>	85
6.10.3	<i>Operação de Seleção</i>	85
6.10.4	<i>Operação de Mutação</i>	85
6.10.5	<i>Critério de Parada</i>	86
7	EXPERIMENTOS E VALIDAÇÃO DO MODELO.....	87
7.1	VALIDAÇÃO DO MODELO	87
7.1.1	<i>Ordem de Execução das Inserções</i>	88
7.1.2	<i>Problema da Alocação de Componentes no Alimentador de Componentes</i>	91
7.1.3	<i>Escalonamento das Ferramentas</i>	92
7.2	EXPERIMENTOS	94
7.2.1	<i>Análise de Significância dos Sub-problemas</i>	95
7.3	ANÁLISE DA INTERAÇÃO ENTRE A BT E O AG DURANTE A BUSCA.....	103
7.4	CASO REAL	104
7.4.1	<i>Máquina Utilizada</i>	104

7.4.2	<i>Obtenção dos Parâmetros Utilizados no Modelo Real</i>	105
7.4.3	<i>Parâmetros Utilizados no Modelo Real</i>	106
8	CONSIDERAÇÕES FINAIS	109
8.1	TRABALHOS FUTUROS	110
	REFERÊNCIAS BIBLIOGRÁFICAS	112
	ANEXO A – INSTÂNCIAS EXTERNAS UTILIZADAS NA VALIDAÇÃO	117

ÍNDICE DE FIGURAS

Figura 1– Foto de uma placa de circuito impresso não montada.	18
Figura 2– Tecnologia PTH.	19
Figura 3– Tecnologia de montagem em superfície.	20
Figura 4– Diagrama em blocos de uma linha SMT.	21
Figura 5– Foto de uma placa montada em tecnologia SMT.	22
Figura 6– Vista superior de uma máquina tipo <i>dual-delivery</i>	24
Figura 7– Máquina tipo multi-estação.	24
Figura 8– Vista superior de uma máquina tipo <i>Turret</i>	25
Figura 9– Máquina tipo <i>multi-head</i>	26
Figura 10– Estrutura de classificação dos conjuntos de problemas.	29
Figura 11– Representação gráfica de um grafo.	30
Figura 12– Evolução da Busca Tabu implementada neste pesquisa aceitando soluções que causam piora na função objetivo.	33
Figura 13– Interação entre os componentes da estratégia de busca da BT.	34
Figura 14– Geração de vizinhanças para a BT - FONTE: (VIANA,1998).	34
Figura 15– Algoritmo de Busca Tabu adaptado (GLOVER; LAGUNA, 1997).	35
Figura 16– Estrutura básica de um Algoritmo Genético simples.	36
Figura 17– CAD para desenvolvimento de placas de circuito impresso.	51
Figura 18– Máquina comercial e posição dos alimentadores.	51
Figura 19– Vista superior da estrutura do sistema modelado.	54
Figura 20– Ciclo de montagem da máquina modelada.	55
Figura 21– Informações necessárias para implementação da metaheurística.	56
Figura 22– Casos possíveis para o modelo da distância percorrida pela cabeça de inserção.	59
Figura 23– Arquitetura do modelo proposto.	63
Figura 24– Solução inicial gerada pelo algoritmo proposto.	64
Figura 25– Encadeamento entre a Busca Tabu e o Algoritmo Genético.	66
Figura 26– Informação da posição dos componentes na PCB.	68
Figura 27– Informação do alimentador usado por cada componente.	68
Figura 28– Informação da ferramenta necessária para inserção de um componente.	69

Figura 29– Correspondência entre números lógicos e físicos de alimentadores.....	70
Figura 30– Solução reduzida e representação gráfica correspondente.....	72
Figura 31– Expansão de uma solução reduzida para solução completa.....	74
Figura 32– Caso de uma solução completa inválida.....	74
Figura 33– Avaliação da similaridade.....	76
Figura 34– Correção de problemas na seqüência de inserção.....	78
Figura 35– Correção de problemas de <i>setup</i>	79
Figura 36– Sub-problema da seqüência de carregamento.....	82
Figura 37– Codificação do cromossomo.....	84
Figura 38– Histograma das soluções encontradas para um total de 200 execuções da metaheurística para a instância Eil51	90
Figura 39– Histograma das soluções encontradas para um total de 200 execuções da metaheurística para a instância d493.....	90
Figura 40– Histograma das soluções encontradas para um total de 200 execuções da metaheurística para a instância d2103.....	91
Figura 41– Percentual de tempo da solução final gasto com cada sub-problema para o caso teste RECEPTOR_23.	98
Figura 42– Percentual de tempo da solução final gasto com cada sub-problema para o caso teste ALARME_57.....	100
Figura 43– Percentual de tempo da solução final gasto com cada sub-problema para o caso teste CENTRAL_315.	101
Figura 44– Interação entre a BT e o AG durante a busca da solução	103
Figura 45– Foto da insersora utilizada para testes no modelo (FONTE: site <i>TWS Automation</i> http://www.tws-automation.com).	105
Figura 46– Processo utilizado neste trabalho para medição dos tempos de operação da máquina insersora analisada.	106

INDÍCE DE TABELAS

TABELA 1 – Explosão combinatória no PCV.....	31
TABELA 2 – Parâmetros da modelagem de uma máquina insersora real.	62
TABELA 3 – Associação entre componentes, inserções e ferramentas.....	65
TABELA 4 – Análise comparativa com instâncias conhecidas aplicada ao problema da seqüência de inserção.	89
TABELA 5 – Análise comparativa entre as soluções iniciais e finais.	93
TABELA 6 – Características dos problemas teste dos experimentos.	96
TABELA 7 – Soluções geradas forçando limitações na metaheurística.	96
TABELA 8 – Gasto percentual de processamento na busca da solução.	104
TABELA 9 – Caracterização da máquina insersora testada.....	106
TABELA 10 – Otimização de um caso real.	107

LISTA DE ABREVIATURAS

AG	Algoritmo Genético
BT	Busca Tabu – O mesmo que TS, porém em português
CAD	<i>Computer Aided Design</i>
CAM	<i>Computer Aided Manufacturing</i>
FO	Função Objetivo
FP	Família de Partes
JSSP	<i>Job-Shop Scheduling Problem</i> - Problema do Escalonamento em um <i>job-shop</i>
PCB	<i>Printed Circuit Board</i> , o mesmo que PCI, porém em inglês
PCCD	Problema do Carteiro Chinês Direcionado
PCI	Placa de Circuito Impresso
PCV	Problema do Caixeiro Viajante
PO	Pesquisa Operacional
PTH	<i>Plated Through Hole</i>
QAP	<i>Quadratic Allocation Problem</i>
ROC	<i>Rank-order Clustering</i>
SMD	<i>Surface Mounted Device</i> - Componentes de Montagem em Superfície
SMT	<i>Surface Mount Technology</i> - Tecnologia de Montagem em Superfície
TG	Tecnologia de Grupo
TH	<i>Through Hole</i>
TS	<i>Tabu Search</i>
TSP	<i>Transport Salesmann Problem</i> – Mesmo que o PCV, porém em inglês
VRP	<i>Vehicle Routing Problem</i>

INTRODUÇÃO

O crescente aumento da quantidade de empresas fabricantes de produtos eletrônicos, aliada à demanda por maior qualidade e preços mais competitivos nos produtos produzidos, está forçando muitas empresas a automatizarem seus processos de montagem e manufatura. Máquinas produzem mais, com maior qualidade e precisão em processos repetitivos, aliando essas características a menores custos, quando comparados a processos equivalentes, essencialmente manuais.

Em uma indústria de produtos eletrônicos em processo de automação da produção, dentre as várias modificações realizadas, as linhas de montagem manuais são substituídas por máquinas comumente chamadas de *pick-and-place*, que são insersoras automáticas de componentes de montagem de superfície (SMD). Essas máquinas são capazes de inserir cada um dos componentes presentes em uma placa de circuito impresso (PCB) a partir de um alimentador de componentes.

Atualmente, todas as máquinas insersoras automáticas de componentes disponíveis no Brasil são importadas, devido à ausência de um fabricante nacional. Dada a demanda elevada por este tipo de equipamento, na data da publicação desta dissertação, abre-se a possibilidade de produção de uma máquina nacional. Há uma série de obstáculos mecânicos, eletrônicos e computacionais a serem vencidos para a produção desse equipamento. Na camada de software, a otimização do processo completo de inserção de componentes é o mais complexo. Esta complexidade deve-se ao fato de que o problema é classificado como NP-Difícil, conforme descrito por Crama em seu trabalho (CRAMA, 1990); sendo que a solução do mesmo, tem impacto direto no desempenho final do equipamento. Desenvolver uma técnica computacional capaz de encontrar uma boa solução para o problema de otimização, associado ao processo de inserção, é o objetivo deste trabalho. O problema é estudado, buscando gerar

uma abordagem que aumente a produtividade do equipamento, pela redução do tempo no processo de inserção de componentes, utilizando técnicas desenvolvidas para a solução dos problemas de Escalonamento das Ferramentas, Escalonamento da Sequência de Inserção de Componentes e Alocação dos Componentes no Alimentador de componentes. Como resultado, é esperado a geração de um programa de montagem que eleve a produtividade da máquina, quando comparada com soluções que utilizam técnicas determinísticas.

Esta dissertação está estruturada em 8 capítulos. No capítulo 1 são apresentados, de forma detalhada, os conceitos básicos de tecnologia de montagem de circuitos eletrônicos. Nele são apresentados os termos, itens e detalhes do processo onde o problema estudado está inserido. Também são apresentadas as tecnologias disponíveis para montagem de placas de circuito impresso, bem como uma pesquisa bibliográfica, apresentando os tipos de máquinas existentes. O capítulo 2 apresenta a fundamentação teórica de tópicos importantes na área da Ciência da Computação e da Pesquisa Operacional, cuja compreensão se faz necessária para acompanhar o desenvolvimento deste trabalho. A abordagem inicia com Complexidade de Algoritmos, passando por classes de Problemas de Otimização Combinatória, Heurísticas, Metaheurísticas e conceitos básicos de Tecnologia de Grupo. O capítulo 3 apresenta a pesquisa bibliográfica de trabalhos que apresentam temas semelhantes, refletindo o estado da arte do tema estudado. Também são apresentadas algumas técnicas e abordagens aplicadas a problemas relativos à otimização de máquinas insersoras de componentes. Esse capítulo é finalizado apresentando o diferencial desta pesquisa em relação aos demais trabalhos estudados. O capítulo 4 contextualiza e define o problema abordado, apresentando de forma modular cada um dos problemas envolvidos no processo de inserção de componentes. Ainda, com o objetivo de contextualizar o problema, é apresentada uma breve descrição do processo que antecede a fabricação da placa, ou seja, o projeto da mesma. O capítulo 5 apresenta o modelo matemático proposto e o método de otimização utilizado. Nele, a máquina modelada e todas as suas características e modos de operação são descritos de forma detalhada. Por fim, a arquitetura do modelo, dividida em cinco fases, é apresentada e detalhada. O capítulo 6 apresenta, de forma detalhada, como foi implementado todo o modelo e codificações da metaheurística híbrida. O capítulo 7 apresenta toda a metodologia de validação empregada e também a validação do modelo e da implementação realizada. São apresentados experimentos, utilizando problemas-teste disponíveis na literatura, aplicados a trabalhos de outros autores, o que permite comparar os resultados obtidos com outros trabalhos. Ainda nesse capítulo, são apresentados experimentos utilizando casos reais aplicados ao modelo de

uma máquina insersora real. Ao longo de todos os experimentos realizados no capítulo 7, os resultados apresentados são discutidos.

No último tópico, nas considerações finais, um resumo focando o objetivo principal de cada capítulo é apresentado, bem como possíveis linhas de pesquisa voltadas a trabalhos futuros.

1 CONCEITOS DE TECNOLOGIA DE MONTAGEM DE CIRCUITOS ELETRÔNICOS

O objetivo deste capítulo é apresentar os conceitos básicos de montagem de circuitos eletrônicos. Esses conceitos se fazem necessários, de modo a promover uma melhor compreensão deste trabalho, que consiste em desenvolver uma técnica computacional capaz de otimizar o funcionamento de uma máquina insersora automática de componentes de montagem de superfície. Essa máquina está presente em grande parte das linhas de montagem de produtos eletrônicos baseados no processo SMT (*surface mount technology*).

1.1 Tecnologia de Circuito Impresso.

A tecnologia de circuito impresso foi criada com o objetivo de substituir os primeiros processos de montagem de circuitos eletrônicos que consistiam na interligação direta entre os componentes eletrônicos utilizando fios. Na tecnologia de circuito impresso, os componentes são montados em uma placa com denominação abreviada pela sigla PCB (*Printed Circuit Board*). Uma placa de circuito impresso, conforme Hoffart (HOFFART; WILEY, 1989), consiste em uma superfície plana que tem por objetivo suportar mecanicamente os componentes eletrônicos e realizar as ligações elétricas entre os mesmos. O núcleo da placa é fabricado com materiais isolantes, cuja função é dar resistência mecânica à placa. Em função do material ser isolante, as ligações entre os componentes são feitas utilizando finas lâminas de cobre, depositadas na superfície ou em camadas internas da placa de circuito impresso, geralmente, através de processos químicos. Uma placa de circuito impresso pode conter uma ou mais camadas de material condutor. A Figura 1 ilustra uma placa de circuito impresso sem componentes montados.

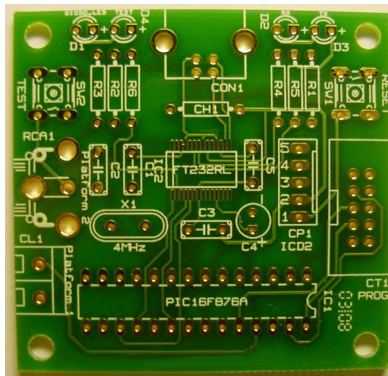


FIGURA 1– FOTO DE UMA PLACA DE CIRCUITO IMPRESSO NÃO MONTADA.

A Figura 1 apresenta uma placa de circuito impresso em seu estágio final de fabricação projetada para uso com tecnologia PTH. A placa possui duas camadas de material condutor, uma na superfície superior (visível na foto) e outra na superfície inferior. O texto gravado sob a placa tem a função de orientar os operadores nos processos de montagem e revisão da placa quando os mesmos forem manuais.

1.2 Tecnologias PTH e TH

A sigla PTH abrevia a tecnologia de montagem chamada de *Plated Through Hole*. Nesta tecnologia os componentes eletrônicos são montados na placa através de pinos presentes nos mesmos, chamados de terminais, inseridos em furos presentes na placa de circuito impresso. Quando os furos possuem conexão elétrica entre as camadas de material condutor da PCB, a tecnologia é chamada de PTH, este é o caso da Figura 1. Quando os furos não possuem conexão elétrica entre estas camadas, a tecnologia é chamada de *Through Hole* (TH). A palavra removida do nome da tecnologia, no caso, *plated*, em português, metalizado, é uma alusão ao processo químico necessário para depositar metal nas paredes do furo e torná-las condutoras, interligando com isso, as camadas de material condutor em diferentes profundidades da PCB. Na Figura 2 são ilustrados itens de tecnologia PTH.

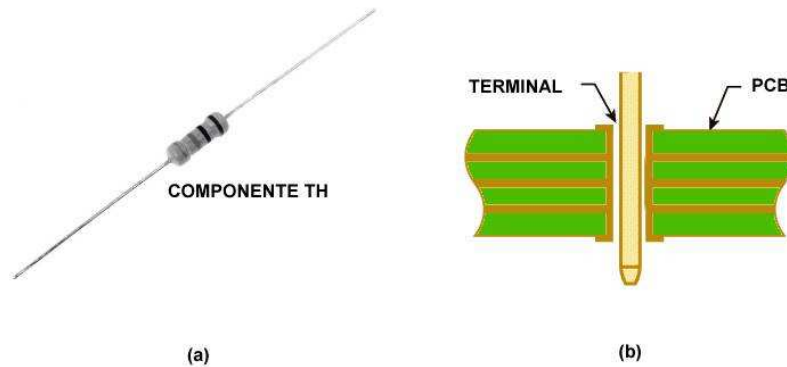


FIGURA 2– TECNOLOGIA PTH.

Na Figura 2, item (a), observa-se um componente eletrônico, mais especificamente um resistor, fabricado em tecnologia *Through Hole*. No item (b) é ilustrada a vista em corte de uma placa de circuito impresso de cinco camadas de material condutor, com o terminal de um componente inserido em um furo. Posteriormente para a ilustração 2(b), os espaços entre o terminal e furo serão preenchidos com solda.

1.3 Tecnologia de Montagem em Superfície (SMT)

A sigla SMT abrevia a tecnologia de montagem chamada de *Surface Mount Technology*, ou tecnologia de montagem em superfície. Essa tecnologia é uma evolução da tecnologia PTH e nasceu da demanda de um conjunto de necessidades: placas mais densas e robustas e componentes menores e mais compatíveis com processos automatizados de montagem. Conforme o livro *Surface Mount Technology: Principles and Practice* (PRASAD, 1989), no desenvolvimento desta tecnologia, os componentes foram projetados visando facilitar seu manuseio por processos automatizados e permitir a fixação dos mesmos em placas de circuito impresso sem a utilização de furos. Isso permitiu eliminar ou reduzir os terminais dos componentes minimizando seu tamanho, aumentar a densidade de trilhas na placa de circuito impresso e principalmente, foi possível aumentar o nível de automação da produção. O processo SMT foi desenvolvido no final da década de 60 e tornou-se extensivamente utilizada no mundo no final da década de 80 (PRASAD, 1989). Grande parte das pesquisas no desenvolvimento dessa tecnologia, foram feitas por grandes fabricantes mundiais japoneses e americanos.

1.3.1 Descrição

Em um processo SMT, conforme Moyer (MOYER; GUPTA, 1997), os componentes são inseridos na superfície de uma placa de circuito impresso, geralmente por uma máquina automatizada, chamada de inseridora automática de componentes ou *pick-and-place* (o foco desta dissertação consiste em desenvolver uma técnica computacional capaz de otimizar o funcionamento dessa máquina). Nessa tecnologia os componentes são soldados na placa através de uma pasta sólida, aplicada na própria placa, antes da inserção dos componentes pela máquina *pick-and-place*. Diferentemente da tecnologia PTH, onde o pé do componente fica ancorado em um furo, no processo SMT a ancoragem é realizada pela solda. Na Figura 3 é ilustrado, isoladamente, um componente fabricado em SMT e também o processo de ancoragem do componente na placa.

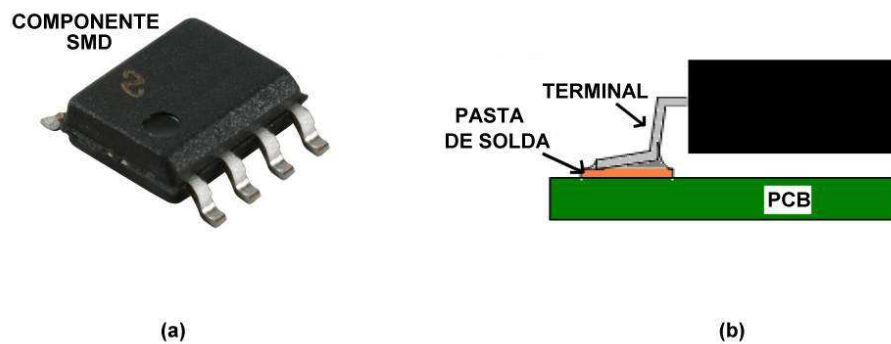


FIGURA 3– TECNOLOGIA DE MONTAGEM EM SUPERFÍCIE.

O componente ilustrado na Figura 3(a), consiste em circuito integrado com encapsulamento designado como *SOIC8*. Os componentes fabricados para uso em processos SMT são chamados de SMD (*surface mounted devices*). Na Figura 3(b) é possível observar a vista em corte de uma PCB com componente semelhante ao apresentando no item (a). A pasta de solda se funde quando levada ao forno, transformando-se em um bloco maciço de metal.

1.3.2 Processo de Montagem em Tecnologia Surface Mount (SMT)

Na Figura 4 é apresentado o diagrama de blocos de uma possível linha de montagem de placas de circuito impresso cujo processo é SMT. Assim como em qualquer outro processo produtivo, as opções de *layout* e organização de máquinas são inúmeras. O diagrama ilustrado na Figura 4 é apenas uma delas.

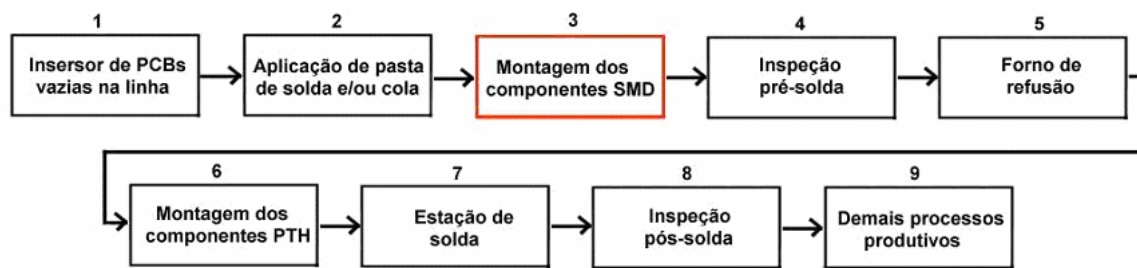


FIGURA 4– DIAGRAMA EM BLOCOS DE UMA LINHA SMT.

A primeira etapa (no primeiro módulo), do processo descrito na Figura 4, consiste em uma máquina ou um operador manual que insere na linha de montagem as placas de circuito impresso que serão montadas na seqüência de produção. O segundo módulo representa aplicação de pasta de solda e/ou cola. A cola é utilizada quando os componentes são montados na face inferior da PCB e se faz necessária para o processo posterior de solda. O terceiro módulo, destacado no diagrama, consiste na inserção automatizada de todos os componentes SMT, por uma ou mais máquinas insersoras. A elaboração de uma técnica computacional capaz de realizar a otimização do ciclo de funcionamento da máquina insersora automática de componentes presente apenas nesta etapa, é o objetivo desta dissertação. O quarto módulo consiste na inspeção visual do trabalho realizado pela insersora automática de componentes. Este módulo pode ser manual ou automatizado. O quinto módulo consiste em aquecer a PCB montada em um forno, chamado forno de refusão. O aquecimento tem dois objetivos: fundir a pasta de solda aplicada anteriormente, fixando componentes na PCB, ou secar a cola aplicada anteriormente para fixação dos componentes. O sexto módulo consiste na montagem, quando necessário, de todos os componentes em tecnologia PTH. Este módulo pode ser manual ou automatizado. O sétimo módulo tem como objetivo soldar os componentes inseridos no módulo anterior e também soldar os componentes SMD que foram apenas colados em fases anteriores do processo produtivo. O processo de solda é realizado por uma máquina conhecida como Solda-Onda. O oitavo módulo consiste na inspeção visual da solda realizada no módulo anterior em busca de falhas. Este módulo também pode ser manual ou automatizado. Ao final dele, a placa já estará montada com todos os componentes. O nono módulo implica em todos os demais processos necessários para finalizar o produto, como testes, inspeções extras, montagem de partes mecânicas, entre outros. Essa etapa é altamente customizada e depende da fábrica e tipo de produto montado.

No diagrama da linha de montagem que utiliza um processo SMT é possível observar a presença de uma etapa de montagem de componentes em tecnologia PTH. Na grande

Não existe na literatura um consenso quanto a classificação das máquinas. Neste trabalho, optou-se por utilizar uma classificação baseada no método de operação da máquina, separando as mesmas em cinco tipos diferentes (AYOB, 2005): entrega dupla (*dual-delivery*), multi-estação (*multistation*), tipo-torre (*turret-type*), multi-cabeça (*multi-head*) e carga e inserção seqüencial (*sequential pick-and-place*).

Independente do tipo, geralmente as insersoras automáticas de componentes são compostas de quatro itens principais:

a) Uma cabeça insersora, com uma ou mais pinças a vácuo utilizadas para carregar os componentes SMD, sendo esta cabeça fixa ou móvel. O formato e tamanho das pinças presentes na cabeça de inserção dependem do tipo de componente a ser inserido, isso faz com que seja necessário um mecanismo automatizado para troca das pinças caso a máquina opere com componentes de diversos formatos;

b) Uma mesa para fixação da placa de circuito impresso, podendo esta também ser fixa ou móvel;

c) Um ou mais grupos de alimentadores automáticos de componentes, subdivididos em *slots*, sendo cada *slot*, alocado a um tipo de componente. Os alimentadores de componentes, assim como os itens descritos anteriormente, também podem ser fixos ou móveis;

d) Por último, existe um alimentador de ferramentas, onde são armazenadas as diferentes pinças a vácuo necessárias para carregar os componentes SM. O alimentador de ferramentas costuma ser fixo na grande maioria das máquinas (AYOB, 2005).

1.4.1 Máquinas Dual-Delivery

Este tipo de máquina se caracteriza por possuir duas cabeças insersoras, com um alimentador individual de componentes para cada uma delas. As cabeças possuem movimentação apenas no eixo Y e operam de forma alternada. Enquanto uma cabeça insersora está inserindo um componente na PCB, a outra está carregando componentes no alimentador definido para ela. Estas máquinas utilizam uma mesa de fixação da PCB com liberdade de movimentação nos eixos X e Y e alimentadores de componentes com liberdade

de movimentação no eixo X. Na Figura 6 pode-se observar a vista superior de uma máquina deste tipo.

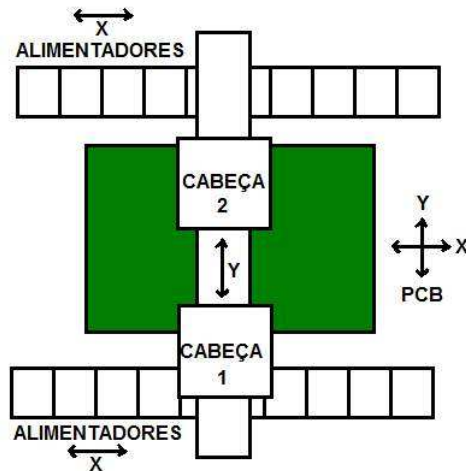


FIGURA 6– VISTA SUPERIOR DE UMA MÁQUINA TIPO *DUAL-DELIVERY*.

Conforme a Figura 6, uma vez que ambas as cabeças inseridoras utilizam a mesma PCB para inserção, as movimentações da mesa de fixação, visando posicionar a placa no ponto correto para inserção, somente podem ocorrer quando nenhuma operação de inserção estiver em andamento.

1.4.2 Máquinas Tipo Multi-Estação

Esse tipo de máquina se caracteriza por possuir mais de um módulo de inserção. Esses módulos consistem em uma cabeça inseridora e um alimentador de componentes operando em paralelo ao longo de uma esteira. Cada módulo opera de forma independente e possui sua própria programação e seqüência de montagem, porém o término e início das operações de todos os módulos são sincronizados entre si. Basicamente, ela se comporta como diversas máquinas com apenas um módulo de inserção, dispostas em série, em uma linha de produção. Na Figura 7 é ilustrada a vista em perspectiva de uma máquina desse tipo.

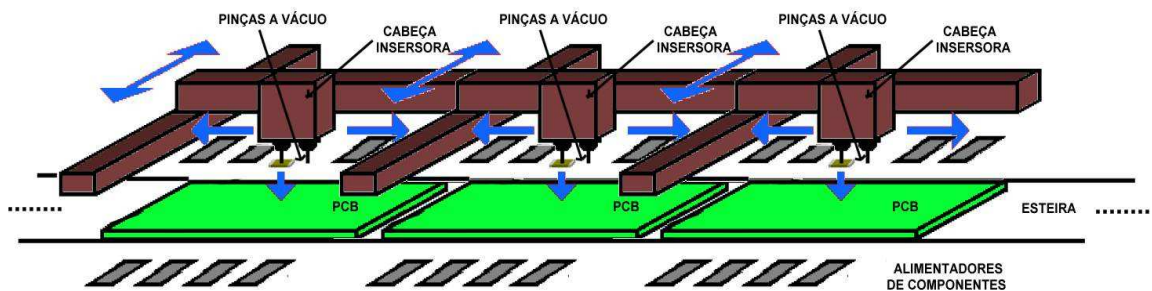


FIGURA 7– MÁQUINA TIPO MULTI-ESTAÇÃO.

Na Figura 7 pode-se observar também a disposição dos itens em uma máquina tipo multi-estação com 3 módulos inseridores. Quando os três módulos estiverem inserindo componentes de forma simultânea nas placas, o *pipeline* estará completo e a máquina irá operar com o máximo de sua capacidade.

1.4.3 Máquinas Tipo Turret

Conforme a publicação de Moyer (MOYER; GUPTA, 1997), as máquinas do tipo Turret, também são chamadas de *chip-shooters*. Essa máquina tem como principal característica sua cabeça de inserção, que é montada em uma torre giratória fixa em forma de carrossel, com múltiplas pinças a vácuo. Essas pinças alteram entre o carregamento de componentes a partir de uma posição fixa e inserção de componentes em outra localização fixa.

Nesse tipo de máquina, uma vez que a posição de carga e inserção é fixa, os alimentadores de componentes e a mesa de fixação da placa de circuito impresso precisam ser móveis. Na Figura 8 é ilustrada a vista superior de uma máquina tipo *Turret*.

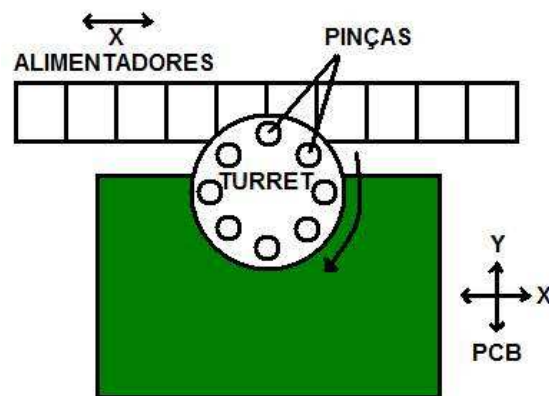


FIGURA 8– VISTA SUPERIOR DE UMA MÁQUINA TIPO *TURRET*.

A Figura 8 apresenta uma máquina tipo do *turret* ou *chip-shooter* com oito pinças a vácuo na cabeça de inserção giratória. Ao girar a cabeça, a máquina, simultaneamente, insere componentes na placa e busca novos componentes no alimentador.

1.4.4 Máquinas Multi-Head

As máquinas *multi-head*, foco da pesquisa desta dissertação, são consideradas como as mais versáteis inseridoras automáticas de componentes existentes (AYOB, 2005) ; (BENTZEN, 2000). Segundo o trabalho de Jeevan e colaboradores (JEEVAN *et al*, 2002), existem dois modelos básicos de máquinas *multi-head*. Em um dos modelos a cabeça de inserção tem

liberdade para se movimentar por toda superfície da máquina (eixos X e Y), atingindo os alimentadores de ferramentas e de componentes e PCB. A cabeça de inserção pode conter duas ou mais pinças a vácuo, que operam de forma independente e tem liberdade de movimento no eixo Z , com o objetivo de carregar e descarregar componentes. No segundo modelo, os movimentos da cabeça de inserção são mais restritos, porém a mesa de fixação da PCB e o alimentador de componentes são móveis, compensando as restrições da cabeça de inserção. A ilustração de uma máquina tipo *multi-head* é apresentada na Figura 9.

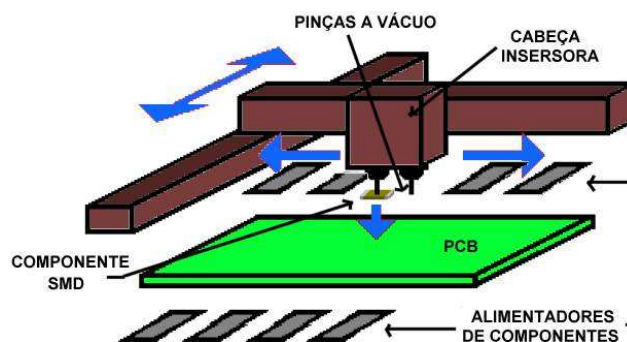


FIGURA 9– MÁQUINA TIPO *MULTI-HEAD*.

Cabe salientar que apesar deste tipo de máquina ser classificada pela literatura como *multi-head* (multi-cabeça), pelo diagrama da Figura 9 pode-se verificar que ela possui apenas uma cabeça de inserção, porém esta cabeça possui no mínimo duas pinças a vácuo, capazes de transportar e inserir componentes.

O ciclo de trabalho de uma máquina *multi-head* com mesa de fixação da PCB e alimentador de componentes fixos, inicia com a cabeça de inserção se deslocando até o alimentador de ferramentas e realizando o *setup* inicial de ferramentas. O passo seguinte consiste em um deslocamento da cabeça até o alimentador de componentes, onde os componentes a serem inseridos nos passos seguintes são carregados pelas pinças a vácuo. Na próxima etapa a cabeça de inserção se desloca até as posições da PCB, onde os componentes devem ser inseridos e insere os componentes de forma seqüencial. Estando a cabeça de inserção vazia, quando necessário, a máquina se desloca até o alimentador de ferramentas e realiza uma nova troca de ferramentas para inserção dos próximos componentes. Não havendo necessidade de troca de ferramentas, a cabeça inseridora desloca-se diretamente ao alimentador de componentes iniciando um novo ciclo de carga e inserção. O processo se repete até que todos os componentes tenham sido inseridos.

1.4.5 Máquinas com Operação Seqüencial de Carga e Inserção

As máquinas classificadas como de operação seqüencial de carga e inserção são apenas um caso particular das máquinas tipo *multi-head*. Neste tipo de máquina, a cabeça de inserção possui apenas uma pinça a vácuo, sendo necessário realizar uma nova carga de componente a cada inserção. O modelo apresentado neste trabalho é generalista o suficiente para tratar com máquinas tipo *multi-head*, onde o número especificado de pinças a vácuo é um, englobando este caso em particular. Uma máquina deste tipo pode ser descrita pela Figura 9, se considerarmos a presença de apenas uma pinça a vácuo e não duas, como apresentado.

1.5 Programas de Montagem Utilizados em Máquinas Insersoras Automáticas de Componentes

Ao montar uma placa de circuito impresso, uma insersora automática de componentes segue uma seqüência de operações descrita em uma lista, chamada nesta dissertação, de instruções de montagem. O objetivo desta pesquisa consiste em desenvolver uma técnica que otimize os parâmetros da lista, de modo a minimizar o tempo total de montagem de uma PCB.

As instruções de montagem são listas de dados que contêm as quatro informações necessárias ao processo de montagem: posição do componente na placa, rotação do componente, tipo de componente (utilizado para determinar a ferramenta correta de inserção) e posição do mesmo no alimentador de componentes. A ordem de execução das operações pela máquina insersora segue a ordem que as mesmas aparecem na lista. A técnica desenvolvida nesta pesquisa pode trabalhar em três parâmetros desta lista: dois parâmetros são diretos. Um é caracterizado pela operação efetuada, outro pela posição dos componentes no alimentador de componentes. O terceiro parâmetro, que é indireto, é a seqüência de operações efetuadas, ou seja o modo como as operações estão ordenadas na lista.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo referenciar os conceitos utilizados no trabalho, relacionando tópicos sobre otimização e os problemas clássicos citados no capítulo 4, onde é apresentada a modelagem do problema.

2.1 Complexidade Computacional de Algoritmos

Em Ciência da Computação, a Teoria da Complexidade Computacional estuda os recursos necessários durante um cálculo para resolver um determinado problema. Em geral, os recursos analisados costumam ser o tempo (número de cálculos ou passos que um determinado algoritmo leva para resolver um problema) e memória, também chamada de espaço, (quantidade de memória necessária para resolver um determinado problema). Conforme Cormen (CORMEN, 2001), um algoritmo pode ser definido como uma seqüência bem definida de passos que, dada uma determinada entrada, produz uma determinada saída.

Os problemas são classificados agrupando os mesmos em conjuntos cuja dificuldade para encontrar uma dada solução seja semelhante, ou seja, necessitam de um algoritmo de complexidade semelhante.

Conforme Viana (VIANA, 1998), algoritmos de tempo polinomial são preferíveis aos algoritmos de tempo exponencial. Em geral, problemas com complexidades $O(m^n)$, onde o grau n é superior a 3, $O(x^n)$ para qualquer $x > 1$ e $O(c.n^2)$ para $c > 106$ são problemas complicados, denominados intratáveis.

Pode-se considerar, de maneira geral, um problema como tratável quando existe um algoritmo de complexidade polinomial capaz de resolvê-lo e intratável quando todos os outros algoritmos conhecidos para resolvê-lo possuem complexidade exponencial ou não polinomial.

A classificação dos problemas em P e NP é baseada na teoria desenvolvida por Cook (COOK, 1971). Em termos de complexidade, pode-se dizer que problemas de otimização de complexidade de tempo de ordem polinomial constituem a classe P, enquanto que os problemas de complexidade de tempo de ordem exponencial constituem a classe NP. Utilizando outra abordagem para efetuar a classificação, pode-se dizer que problemas de decisão que permitem o uso de algoritmos determinísticos polinomiais para a sua solução formam a classe P, enquanto que os problemas de decisão resolvidos com algoritmos não-determinísticos polinomiais constituem a classe NP. São classificados como determinísticos os algoritmos que, para uma dada entrada de um problema, sempre retornam a mesma solução. Algoritmos que, para uma mesma entrada, apresentem soluções diferentes a cada execução, são chamados de não-determinísticos.

A classe NP também representa os problemas de decisão em que existe um algoritmo não-determinístico polinomial para justificar a resposta “sim”, enquanto que a justificativa da resposta “não” geralmente requer uma busca exaustiva. A classe NP-Completo foi sugerida por Cook (COOK, 1971), que definiu o problema de satisfabilidade (SAT), na forma normal conjuntiva, como o problema mais difícil da classe NP e que todos os outros problemas podem ser reduzidos polinomialmente a ele.

Se um problema $\pi_0 \alpha \pi$ para todo $\pi_0 \in NP - Completo$, então o problema π é considerado NP-Difícil (*NP-Hard*), ou seja, pode-se afirmar que este problema é pelo menos tão difícil, quanto qualquer problema da classe NP. A Figura 10 mostra a estrutura de classificação dos conjuntos de problemas.

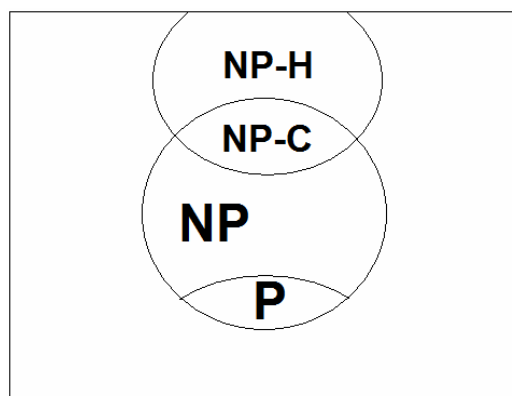


FIGURA 10– ESTRUTURA DE CLASSIFICAÇÃO DOS CONJUNTOS DE PROBLEMAS.

Na Figura 10, as siglas NP-H e NP-C referem-se respectivamente a NP-Hard (NP-Difícil) e NP-Completo. Os círculos e arcos delimitam os conjuntos de problemas classificados.

Os problemas NP-Completo constituem os mais difíceis da classe NP. Pela Figura 10 é possível observar que os problemas NP-Difíceis são classificados à parte da classe NP, sendo que a dificuldade em resolvê-los é não menos difícil que de problemas da classe NP-Completo.

O problema tratado por esta dissertação pode ser modelado separando o mesmo em três sub-problemas NP-Difíceis, o que justifica o uso de uma metaheurística para tratamento do mesmo.

2.2 Circuito Hamiltoniano

Um grafo, representado pela letra G consiste em um conjunto “V” de elementos chamados vértices ou nós, e outro conjunto “E” de elementos formados pela união de pares de vértices (v,w) chamados bordas, arestas ou arcos. A Figura 11 mostra um grafo composto por 6 vértices.

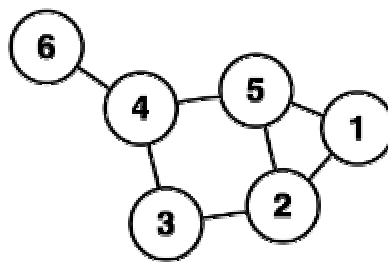


FIGURA 11– REPRESENTAÇÃO GRÁFICA DE UM GRAFO.

Na Figura 11, o grafo apresentado é formado por um conjunto de vértices $\{1,2,3,4,5,6\}$ e um conjunto de arestas, formado pelos pares $\{(1,2),(1,5),(2,3),(2,5),(3,4),(4,5),(4,6)\}$. Como exemplo ilustrativo, em um dos problemas abordados nesta pesquisa, cada vértice representa a posição onde um componente eletrônico deve ser montado.

Um caminho fechado que passa por todos os vértices, exatamente uma vez por cada vértice, é chamado de circuito Hamiltoniano. O problema do caixeiro viajante, amplamente

utilizado para testes de algoritmos em PO, consiste em encontrar o circuito hamiltoniano de menor comprimento possível em um grafo. Determinar se existem circuitos deste tipo em um grafo é um problema da classe NP-Completo. O Problema do Caixeiro Viajante (PCV) foi apresentado em torno do ano de 1800 pelos matemáticos Sir William Rowan Hamilton e Thomas Penyngton Kirkman. Uma discussão sobre o trabalho inicial de Hamilton e Kirkman, relativa ao problema do caixeiro viajante, pode ser encontrada no trabalho de Biggs (BIGGS et al, 1976).

Nesta pesquisa, partes específicas do algoritmo implementado foram validadas separando o problema principal em sub-problemas, estes sub-problemas foram modelados durante a validação como PCV.

2.3 Heurísticas

Problemas NP, NP-Completo, devido a sua natureza, não podem ser resolvidos por algoritmos exatos quando o número de elementos analisados é grande, devido ao problema da explosão combinatória. Um exemplo simples de algoritmo exato que, teoricamente, é capaz de encontrar a solução, consiste em verificar todas as soluções viáveis, classificá-las e escolher a solução que otimiza a função objetivo.

Conforme Rayward (Rayward *et al*, 1996), o problema da explosão combinatória cria restrições relativas ao tempo de processamento necessário para encontrar a solução. Usando como exemplo o Problema do Caixeiro Viajante, onde o número de rotas passíveis de análise é dado por $(n-1)!$, tal que n representa o número de cidades. Na Tabela 1 pode-se ver o problema causado pela explosão combinatória. Os dados foram escolhidos de forma a permitir a clara visualização da explosão.

TABELA 1 – EXPLOSÃO COMBINATÓRIA NO PCV.

Número de cidades	Rotas possíveis (n-1)!	Tempo de processamento (1µs por rota)
9	40320	0,04 s
16	1307674368000	363 horas
21	$2,43 \times 10^{18}$	77146 anos
26	$1,55 \times 10^{25}$	491857243890 anos

A Tabela 1, a coluna mais a esquerda representa o número de cidades. A coluna central representa o número de rotas possíveis baseadas na equação $(n-1)!$. A coluna a direita

representa o tempo gasto para verificação de todas as rotas possíveis, considerando que a análise de cada rota leva apenas um microsegundo.

O problema da explosão combinatória, que torna alguns problemas extremamente complexos, fez com que métodos de solução aproximada não exatos fossem desenvolvidos, sendo estes chamados de Heurísticas. Segundo Viana (VIANA, 1998), a definição para heurística é um método ou técnica criada para resolver um problema em um tempo computacional aceitável, sem garantia de otimalidade do resultado. As heurísticas podem ser classificadas em dois tipos: heurísticas construtivas e heurísticas gulosas. As heurísticas construtivas elaboram uma solução passo a passo, onde em cada passo se adicionam componentes individuais: vértices, arestas, variáveis, etc. As heurísticas gulosas funcionam de forma diferente, elas procuram a cada passo maximizar a melhoria local.

As heurísticas, quando projetadas, normalmente são ligadas ao problema que se propõem a resolver (o que não impede que a mesma seja utilizada em outros problemas), sendo assim, o sucesso ou qualidade da solução de uma heurística depende em grande parte da experiência de quem está desenvolvendo ou utilizando uma heurística conhecida para resolver um problema.

2.4 Metaheurísticas

Os métodos heurísticos tem como grande deficiência o fato de que normalmente ficam presos a ótimos locais, quando encontrados. Com o objetivo de superar esta limitação surgiu o conceito de Metaheurísticas. As metaheurísticas são estratégias que guiam e modificam heurísticas subordinadas, de forma a produzir outras soluções, além das encontradas através de simples buscas locais (GLOVER; LAGUNA, 1997).

As metaheurísticas possuem métodos para não ficarem presas em ótimos locais. Estes métodos incluem aceitar soluções que piorem a função objetivo, como ocorre nos Algoritmos Genéticos, Busca Tabu e outros. Especificamente a BT possui ainda uma memória adaptativa (lista tabu), que evita que caminhos já percorridos sejam percorridos com frequência. Visando representar a aceitação de resultados ruins, um exemplo de evolução da BT implementada nesta pesquisa é ilustrado na Figura 12.

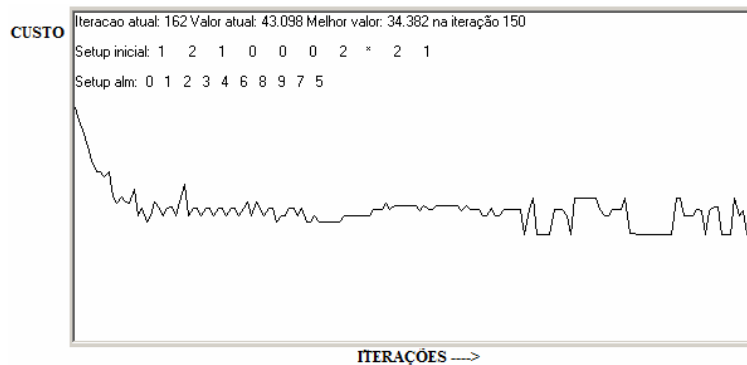


FIGURA 12– Evolução da Busca Tabu implementada neste pesquisa aceitando soluções que causam piora na função objetivo.

Em geral, apesar das metaheurísticas possuírem um caráter generalista se comparado às heurísticas puras, o conhecimento do problema e a forma como a metaheurística foi aplicada para resolver o problema impactam na qualidade da solução. Frequentemente é necessário o ajuste dos parâmetros utilizados na metaheurística para adequação ao problema aplicado. Os ajustes normalmente são feitos de forma empírica, alterando parâmetros e executando o algoritmo diversas vezes. Com isso é possível analisar o comportamento de uma metaheurística, quando aplicada a um problema qualquer.

2.4.1 Busca Tabu

A metaheurística *Tabu Search* (TS ou BT, em português) foi apresentada por Glover (GLOVER, 1986) como uma metaheurística voltada à solução de problemas de programação inteira. No ano seguinte, em outra obra, (GLOVER, 1987) ele descreveu o método de forma geral, para uso em diversos problemas na área de Pesquisa Operacional. Outro autor propôs na mesma época uma técnica semelhante, porém Glover é considerado o criador da técnica em função dos inúmeros trabalhos publicados na área. Conforme Viana (VIANA, 1998), a metaheurística Busca Tabu foi criada para encontrar uma boa aproximação da solução ótima global de qualquer problema de otimização. A técnica tem por base três princípios:

- a) Possui uma estrutura de dados organizada em forma de fila que armazena o histórico do processo de busca;
- b) Possui um mecanismo que baseado nas informações da fila, aceita ou não uma nova solução. A análise é realizada com base nas restrições e aspirações desejadas;
- c) Pode utilizar mecanismos que permitem ao método alternar entre estratégias de diversificação e intensificação.

A Figura 13 ilustra a interação entre os componentes da estratégia de busca da BT.

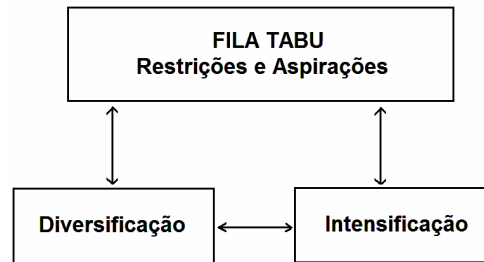


FIGURA 13– INTERAÇÃO ENTRE OS COMPONENTES DA ESTRATÉGIA DE BUSCA DA BT.

Glover salienta que o método faz uma exploração agressiva, referindo-se à busca, sempre, do melhor movimento possível, podendo melhorar a solução atual ou não (GLOVER, 1990). Isso permite alternar entre estratégias de intensificação e diversificação baseada na análise de atributos proibidos, chamados de tabu. No caso da diversificação, o objetivo é analisar superficialmente todo o espaço de soluções possíveis, com isso evita-se o bloqueio em soluções ótimas locais, que corresponde a repetições cíclicas. No caso da intensificação, é realizado um reforço na busca de uma solução historicamente considerada como boa. Nesta dissertação a estrutura utilizada foi simplificada e o papel da diversificação ficou a cargo do Algoritmos Genéticos.

A Figura 14 apresenta um esboço da evolução do processo de busca da BT.

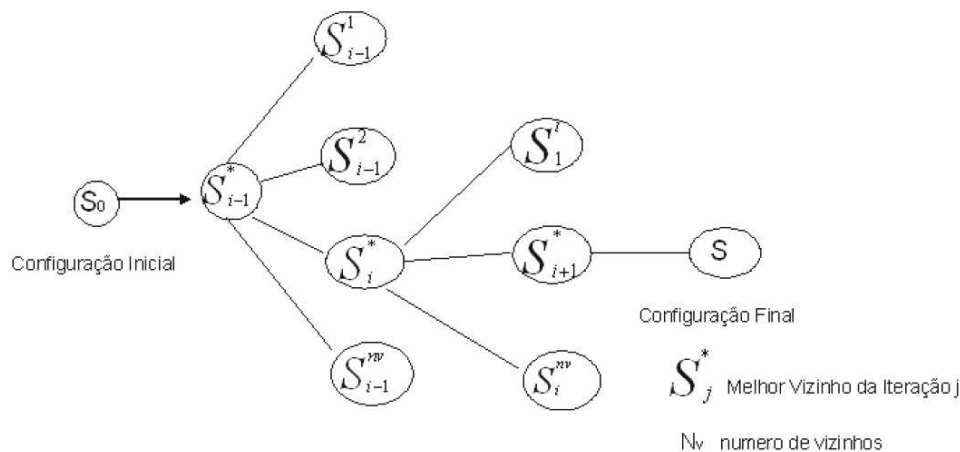


FIGURA 14– GERAÇÃO DE VIZINHANÇAS PARA A BT - FONTE: (VIANA,1998).

A evolução do processo de busca ocorre através da geração de N_v configurações vizinhas a atual. A notação S_i^j indica, para “ j ” variando de 1 a N_v , é o vizinho da ordem “ j ” de S_{i-1} ; e que S_i^* é a “melhor” solução entre eles, para algum $1 \leq j \leq N_v$. No caso de um problema de maximização, a designação “melhor” indica o maior valor encontrado.

A Figura 15 ilustra o algoritmo BT para um problema de minimização.

```

s = solução inicial obtida através de uma regra de despacho;
niter = 0;
melhiter = 0;
nbmax = numero de iterações consecutivas sem melhora na solução;
melhsol = s;
Fmelhor = F(s);
Fmin = valor mínimo estimado da função;
Inicializar a lista Tabu;
Inicializar a função do critério de aspiração  $A(z = F(s))$ ;

ENQUANTO (F(s) > Fmin) OU ( niter - melhiter < nbmax) FAÇA
    niter = niter + 1;
    Gere um conjunto  $V^*$  de soluções em  $N(s)$ ;
    Escolha a melhor solução  $s^*$  em  $V^*$  que não
    seja tabu ou  $F(s^*) < A(F(s))$ ;
    Atualize a função do critério de aspiração  $A(z = F(s))$  e a lista tabu retirando o movimento mais
    antigo;
        SE  $F(s^*) < F(melhsol)$  ENTÃO
            melhsol =  $s^*$ ;
            melhiter = niter;
            Fmelhor =  $F(s^*)$ ;
        FIM SE
        s =  $s^*$ ;
FIM ENQUANTO

Onde,
S = conjunto de soluções viáveis;
s = solução inicial pertencente a S;
niter = número de iterações;
melhiter = número da iteração correspondente à melhor solução encontrada;
nbmax = número máximo de iterações sem obter melhora na melhor solução obtida;
melhsol = melhor solução;
Fmelhor = valor da F.O. da melhor solução;
Fmin = valor mínimo estimado da função;
 $V^*$  = vizinhança gerada;
 $s^*$  = melhor solução da vizinhança  $V^*$ ;
F = função objetivo;

```

FIGURA 15– ALGORITMO DE BUSCA TABU ADAPTADO (GLOVER; LAGUNA, 1997).

A cada iteração um ótimo local é escolhido e dele é gerada a nova vizinhança. Para evitar ciclos e mínimos locais, é implementada uma lista de movimentos proibidos, chamada de lista tabu. Caso a iteração gere uma melhora na função objetivo F , utiliza-se a função critério de aspiração A , que admite o movimento proibido. Para as condições de parada do algoritmo, pode-se utilizar um número máximo de iterações ($nbmax$) sem que ocorra melhora na função objetivo F ou executar até que alcance o valor mínimo ($Fmin$).

2.4.2 Algoritmos Genéticos

Os algoritmos genéticos (AG) foram criados em 1975 por Holland, baseando-se nos processos observados na evolução natural das espécies: os indivíduos mais aptos sobrevivem e geram descendentes com suas características hereditárias (VIANA, 1998). A idéia principal dos AGs é explorar uma população de soluções, criando novas soluções a partir da recombinação de bons atributos das soluções que serviram como base (soluções pai). A Figura 16 ilustra a estrutura básica de um Algoritmo Genético simples.

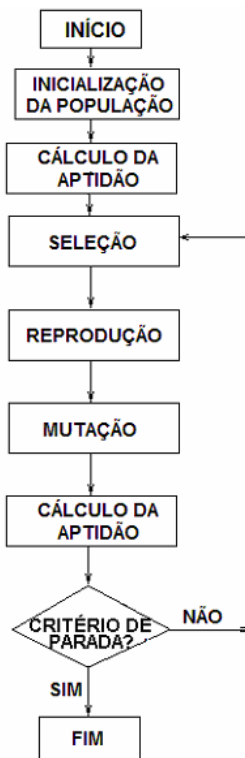


FIGURA 16– ESTRUTURA BÁSICA DE UM ALGORITMO GENÉTICO SIMPLES.

De forma geral, um Algoritmo Genético parte de uma população de indivíduos (soluções ou cromossomos) gerados de forma aleatória ou através de heurísticas (OSMAN; LAPORTE, 1996). Estas soluções correspondem às configurações iniciais do problema. Posteriormente é realizado o cálculo da aptidão da desta população. Na operação seguinte é realizada seleção dos melhores indivíduos para a operação posterior de cruzamento, a seleção visa separar os melhores indivíduos (elitização). A seleção é baseada no cálculo de aptidão de cada solução baseado nos critérios estabelecidos na função objetivo. Nesta fase, as soluções são ordenadas em função da qualidade. Com os melhores indivíduos selecionados, são executadas algumas manipulações genéticas, como cruzamento e mutação (existem outras operações utilizadas em algoritmos mais complexos). Após a nova geração de indivíduos ser criada, é realizado o cálculo de aptidão de cada individuo novo. O processo é repetido até que um determinado critério de parada seja alcançado, que pode ser o valor da FO, o número máximo de gerações, o número máximo de gerações sem melhora na solução, dentre outros.

O termo “cromossomos” é aplicado às soluções do problema. Geralmente a forma de representação dos cromossomos é feita com a utilização de *strings* binárias, embora existam outras formas de representação (PHAM; KARABOGA, 2000). Os “genes” são as posições dentro do cromossomo. Nas manipulações genéticas, o cruzamento é utilizado para criar novos indivíduos a partir dos pais selecionados. Neste trabalho sempre são gerados dois novos

filhos. A criação de novos indivíduos pode ser feita através de diversas maneiras, a mais simples é o cruzamento de um ponto. Neste caso, dois indivíduos chamados de pais são selecionados e um ponto na *string* é escolhido para ser o ponto de corte. Com isso, cada indivíduo é dividido em dois segmentos. A criação dos novos indivíduos ocorre através da união do segmento 1 de um indivíduo com o segmento 2 do outro. O exemplo a seguir mostra um caso de cruzamento entre dois pais:

Pais: $S1 = 10001001$ e $S2 = 11110011$, com ponto de corte a partir do gene 4, ou seja, 10001001 e 11110011 .

Indivíduos gerados: $S3 = 10000011$ e $S4 = 11111001$.

A mutação corresponde a perturbações e tem por objetivo evitar que o algoritmo fique preso em ótimos locais. A mutação pode ser realizada de diversas formas. Uma das técnicas mais simples é através da reversão randômica de “genes” (*bits*), de acordo com uma taxa específica relativa ao problema modelado (PHAM; KARABOGA, 2000). No exemplo a seguir é apresentada uma mutação:

Indivíduo original: $S = 10010010$.

Invertendo os bits 3 e 4 o resultado da mutação será:

Indivíduo alterado: $S = 10001010$.

Em seu livro, Viana afirma que os algoritmos genéticos possuem 4 características principais (VIANA, 1998): a) Usam técnica de randomização; b) São robustos em relação aos problemas que podem resolver (abrangentes); c) Trabalham exclusivamente com o valor da função objetivo e d) Utilizam somente regras probabilísticas (não-determinísticas) para encontrar a solução.

Os AGs podem ser utilizados para resolver qualquer tipo de problema de otimização e são classificados como metaheurísticas analógicas.

2.5 Tecnologia de Grupo

A Tecnologia de Grupo (TG) nasceu da necessidade de organizar e otimizar o funcionamento de um processo produtivo. Apesar do conceito estar ligado a otimização de linhas de produção inteiras, com diversas máquinas, peças e processos distintos, alguns dos conceitos utilizados na TG podem ser relacionados e aplicados, especificamente, na otimização do processo de funcionamento de apenas uma máquina, o que é o objetivo desta dissertação. A Tecnologia de Grupo é uma filosofia em que problemas semelhantes são agrupados segundo alguma similaridade, de forma que se possa obter uma única solução para todos os elementos do grupo (LORINI,1993). Aplicada ao ambiente de manufatura, a TG é utilizada principalmente na tarefa de reunir partes em grupos, segundo similaridades de projeto e/ou produção. Aplicada ao trabalho apresentado nesta dissertação, a TG será empregada na tarefa de formar os grupos de componentes associados as ferramentas utilizadas na cabeça insersora da máquina.

2.5.1 Família de Partes

Uma Família de Partes (FP) é um conjunto de peças ou partes agrupadas segundo alguma similaridade. As partes dentro de uma FP não precisam ser iguais, mas devem possuir similaridade suficiente para que possam ser reunidas, respeitando limitações impostas, como restrições físicas relativas às máquinas que irão processá-las. Em um ambiente de manufatura, geralmente o critério de similaridade pelo qual as partes são agrupadas, é o de forma geométrica ou o de processo produtivo. Aplicando o conceito de Família de Partes ao trabalho proposto, os critérios de similaridade podem ser ferramentas comuns a determinados componentes e/ou localização geométrica na PCB.

Segundo Lorini (LORINI, 1993), para a formação das FPs, existem três métodos amplamente conhecidos: inspeção visual, classificação e codificação e análise de fluxo de produção. Os métodos de inspeção visual e classificação e codificação não possuem aplicação direta ao trabalho proposto por esta dissertação, porém algumas técnicas utilizadas na formação das FPs utilizando a análise de fluxo de produção, como formação de famílias através do método de ordenação *Rank-order clustering* (ROC), podem ser diretamente aplicadas ao trabalho proposto.

2.5.2 Separação em Famílias de Partes Utilizando o Método de Ordenação Rank-Order Clustering (ROC)

O autor King em sua publicação desenvolveu um algoritmo para separação em famílias de partes, utilizando matrizes de incidência baseados em métodos de ordenação (KING,1980). O algoritmo é simples e eficiente e consiste em cinco passos apresentados a seguir:

a) Para cada linha da matriz de incidência, definir um peso binário e calcular o peso decimal equivalente;

b) Organizar as linhas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes;

c) Para cada coluna da matriz obtida no passo b, designar um peso binário e calcular o peso decimal equivalente;

d) Organizar as colunas da matriz na ordem decrescente dos valores dos pesos decimais equivalentes;

e) Repetir os passos de (a) a (d) até não haver mais mudanças de posições dos elementos em cada linha ou coluna. Os pesos para cada linha i e coluna j são calculados utilizando a equações (1) e (2).

i = indexador da linha;

j = indexador da coluna;

n = número de partes;

m = número de máquinas;

$$\text{linha } i : \sum_{k=1}^n a_{ik} \times 2^{n-k} \quad (1)$$

$$\text{coluna } j : \sum_{k=1}^m a_{jk} \times 2^{m-k} \quad (2)$$

Sujeito a:

$$a_{ij} = 0,1 \quad (3)$$

$$i = 1, \dots, n \quad (4)$$

$$j = 1, \dots, m \quad (5)$$

A restrição imposta pela equação (3) assegura que a matriz de cálculo seja uma matriz de incidência. A restrição imposta pela equação (4) especifica que o indexador i deve ser limitado ao número de partes. A restrição imposta pela equação (5), especifica que o indexador j deve ser limitado ao número de máquinas. É importante salientar que este método não precisa que a quantidade de FPs seja pré-definida. Esta técnica posteriormente será utilizada para definir os grupos de componentes associados às pinças a vácuo (ferramentas) durante a criação da solução inicial.

2.6 Escalonamento de Partes

O problema do escalonamento consiste em alocar recursos de forma a ordená-los em tarefas a serem executadas seqüencialmente. Esses recursos são limitados de diversas formas, como exemplo pode-se citar número de máquinas e tempo disponível. Conforme Garey (GAREY, et al, 1976), os problemas de escalonamento têm sua complexidade classificada como NP-Difícil. Aplicado ao ambiente de manufatura, o escalonamento de partes pode ser definido através de diversos objetivos, dependendo fortemente do contexto onde está inserido. Independente do objetivo proposto, como a minimização de troca de ferramentas, minimização do tempo de atraso de uma parte, maximização a produção, o escalonamento, conforme cita Jain (JAIN, 1998), tem como grande objetivo reduzir o tempo total de produção (*makespan*). Blazewicz divide o problema do escalonamento em três modelos (BLAZEWICZ; KOVALYOV, 2002):

Flow-shop: Modelo onde a produção é considerada contínua. É caracterizado pelas linhas de produção em massa, ou seja, o fluxo do processamento das partes é seqüencial e linear e a quantidade de partes diferentes é muito pequena;

Open-shop: É um ambiente aberto. A ordem de processamento não é especificada e o número de operações é igual ao número máquinas;

Job-shop: O fluxo de produção é realizado de acordo com a rota de produção definida na parte. Caracteriza-se por um alto *mix* de partes e fluxo de produção variado.

Aplicado ao trabalho proposto por esta pesquisa, o problema do escalonamento de um *job-shop* (JSSP), permite modelar o processo de funcionamento de uma insersora automática de componentes, no qual consideradas as partes (geralmente com *mix* elevado). As diferentes pinças a vácuo necessárias para transportar as partes podem ser consideradas como as máquinas e a rota de produção definida pela parte, pode ser modelada através de uma função matemática que correlacione a posição de destino do componente na placa com seu tempo de operação. Os autores Müller e Gómez (MULLER; GÓMEZ, 2005) definem um *job-shop* como sendo um conjunto de “*n*” *jobs* a serem processados num conjunto de “*m*” máquinas. Cada *job* é composto de “*i*” operações, que devem ser processadas através de um roteiro; para cada operação é definida uma rota com tempo de processamento padrão. Como já mencionado anteriormente, o objetivo é minimizar o tempo total de produção.

2.6.1 Formulação Clássica

No ambiente de manufatura, em função de ser um problema amplamente estudado, existem diversas formulações possíveis para o JSSP, porém será adotado o modelo da formulação de Adams apud (BLAZEWICZ *et al*, 1996). O conjunto de operações é representando por $V = \{0, 1, \dots, n\}$, onde “0” é a primeira operação de todos os *jobs* e “*n*” é a última operação para todos os *jobs*. “*M*” representa o conjunto de “*m*” máquinas e “*A*” é a representação para o conjunto de pares ordenados das restrições de operações pela precedência das relações de cada *job*. Para cada máquina “*k*”, o conjunto de “ E_k ” descreve todos os pares de operações fornecidos pela máquina “*k*”. Cada operação “*i*” é processada num tempo “ p_i ” (fixo) e o processo inicial de “*i*” é “*t*”, uma variável que tem sido determinada durante a otimização. “*j*” descreve o indexador que indica qual máquina está sendo analisada. A partir destas definições, o modelo é definido como:

$$\begin{aligned} \min t_n \\ t_j - t_i \geq p_i \quad \forall (i, j) \in A, \quad i = 1, \dots, n; \quad j = 1, \dots, n \end{aligned} \quad (6)$$

$$t_j - t_i \geq p_i \text{ ou } t_i - t_j \geq p_i \quad \forall \{i, j\} \in E_k, \quad (7)$$

$$\forall k \in M \quad (8)$$

$$t_i \geq 0 \quad \forall i \in V. \quad (9)$$

A função objetivo, apresentada na equação (6), para o modelo proposto, busca minimizar o tempo total de produção. A restrição apresentada na equação (7) garante que a seqüência de processamento das operações para cada *job* corresponde a uma ordem pré-determinada. A restrição apresentada na equação (8) garante que haja somente um *job* em cada máquina num determinado tempo, e a equação (9) assegura as restrições de não negatividade.

O autor também apresenta algumas restrições ao modelo, que são definidas para os *jobs* e para as máquinas:

a) Não existe regra para a precedência de operações entre as diferentes partes;

b) Ao iniciarem o processo, as operações não podem ser interrompidas e uma máquina processa apenas um *job* por vez;

c) Um *job* pode somente ser processado em uma máquina por vez.

No modelo proposto, as rotas de *jobs* para cada máquina são fixas, assim o problema consiste em encontrar a rota de cada um dos *jobs* em cada máquina, de modo a minimizar o tempo total de produção do primeiro ao último *job* (JAIN, 1998).

Ao aplicar este modelo, no trabalho proposto por esta dissertação, algumas considerações devem ser feitas, como por exemplo: cada parte usa apenas uma máquina (pinça à vácuo), porém um novo ciclo de trabalho nas máquinas somente pode ser iniciado assim que todas as máquinas tenham completado suas tarefas, uma vez que o carregamento das peças é executado simultaneamente em todas as máquinas.

3 REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica contempla os trabalhos realizados para otimização de máquinas do tipo *multi-head* e de operação seqüencial. Conforme descrito no capítulo 1, estes tipos de máquinas consistem em um caso particular de máquinas *multi-head*, onde apenas uma pinça é montada na cabeça inserora. Também são contemplados trabalhos baseados em outros tipos de máquinas, mas que entretanto, utilizam metaheurísticas semelhantes a adotada nesta pesquisa.

O autor Crama (CRAMA *et al*, 2002) dividiu o problema de otimização aplicável a uma máquina simples em quatro sub-problemas:

- a) Alocação dos componentes no alimentador de componentes;
- b) Seqüência de inserção de componentes na PCB;
- c) Plano de carga dos componentes;
- d) Controle de movimento.

Posteriormente, Ayob (AYOB, 2005) acrescentou o sub-problema (e): a otimização do conjunto de ferramentas (pinças a vácuo) utilizadas no processo de inserção dos componentes. Esse sub-problema é importante na busca por boas soluções, uma vez que as trocas de ferramentas constituem a operação que mais demanda tempo da máquina (CRAMA, 1990).

A pesquisa desta dissertação estuda a resolução de quatro dos cinco sub-problemas anteriormente citados, a saber: a alocação de componentes no alimentador, a seqüência de inserção dos componentes na placa de circuito impresso, o plano de carga de componentes na cabeça inserora e a otimização do conjunto de ferramentas. Para o controle de movimento,

assume-se que o sistema eletrônico da máquina fica responsável pela tarefa, pois o problema está intrinsecamente ligado ao projeto eletro-mecânico da máquina. Não sendo considerado o controle de movimento, o trabalho proposto abordará somente operações de alto nível, como comandos de troca de ferramentas, posicionamento da cabeça de inserção, carga e inserção de componentes. Essa abordagem é utilizada na grande maioria dos trabalhos estudados sobre o tema (CRAMA *et al*, 2002). Em função do modelo de máquina abordada neste trabalho ser do tipo *multi-head*, o plano de carga e seqüência de inserção foram combinados em um só problema. Com base nas considerações, a dissertação propõe apresentar um método eficiente de resolver três problemas associados à otimização de uma insersora *multi-head*. São eles:

- a) Alocação dos componentes no alimentador de componentes;
- b) Seqüência de inserção de componentes na PCB;
- c) Otimização do conjunto de ferramentas e escalonamento do mesmo.

A pesquisa bibliográfica foi delimitada de modo a apresentar trabalhos de outros autores, cuja abordagem é feita em um ou mais dos três problemas citados anteriormente, excluindo o controle de movimento.

3.1 Trabalhos Aplicados a Máquinas *Multi-head*

Alguns trabalhos foram desenvolvidos com o objetivo específico de otimizar a seqüência de inserção de máquinas insersoras automáticas de componentes do tipo *multi-head*. Dentre eles podem ser destacados os trabalhos de Lee S. (LEE *et al*, 1999), Lee W. (LEE, W.; *et al*, 1999), Magyar (MAGYAR *et al*, 1999), Altinkemer (ALTINKEMER, 2000) e Burke (BURKE *et al*, 2001).

Em seu trabalho, Lee (LEE *et al*, 1999), desenvolveu heurísticas baseadas em programação dinâmica e o algoritmo do vizinho mais próximo aplicado ao Problema do Caixeiro Viajante para resolver o problema de otimização associado a máquinas do tipo *multi-head*. Os autores, para encontrar a solução, escolheram um método hierárquico onde primeiramente os componentes eram dispostos em grupos e em seguida o problema da alocação dos componentes no alimentador de componentes era resolvido, e por fim a seqüência ótima de inserção era determinada. Como a troca das pinças consumia muito tempo

e o número de trocas de pinças era proporcional ao número de pinças utilizadas, eles escolheram efetuar trocas de pinça uma vez para cada componente diferente, assim o número de mudanças seria minimizado. O próximo passo foi atribuir um tipo de componente para cada pinça disponível na cabeça insersora, de forma a manter balanceada a utilização de cada pinça. Como a troca de pinças era a operação que mais consumia tempo, primeiramente toda a seqüência de troca de pinças era determinada e posteriormente era definida a seqüência de inserção. Segundo seu trabalho, as simulações demonstraram que o método proposto por eles reduzia em média 18% o tempo de montagem em relação ao algoritmo utilizado pelo fabricante (Yamaha) da máquina modelada.

No mesmo ano, Lee e colaboradores (LEE, W.; *et al*,1999) aplicaram um Algoritmo Genético na solução do mesmo problema proposto em seu trabalho anterior (LEE *et al*, 1999). Desta vez, ao invés de tratar cada um dos problemas separadamente, algoritmos genéticos foram utilizados e o problema da alocação de componentes no alimentador e o problema da seqüência de inserção foram tratados em conjunto. Neste trabalho, a abordagem reduziu em média 10,7% o tempo de montagem em relação ao algoritmo utilizado pelo fabricante da máquina modelada. Pode-se observar neste caso, que a abordagem conjunta dos problemas, contrariando a expectativa, piorou a qualidade da solução em relação ao trabalho anterior.

Magyar (MAGYAR *et al*,1999) abordou os problemas de determinar a seqüência ótima de inserção, o da alocação dos componentes no alimentador e o do escalonamento de trocas na cabeça de inserção, utilizando uma abordagem hierárquica. Em seu trabalho ele estudou uma máquina *mutli-head* com cabeça de inserção que continha quatro pinças a vácuo, sendo cada uma delas capaz de lidar com um componente. Nessa obra, ele resolveu o problema da alocação de componentes no alimentador, utilizando uma heurística do tipo *greedy local search*, onde a mesma procurava maximizar o número de componentes pinçados de forma simultânea pela cabeça de inserção no alimentador de componentes. O resultado desse problema era utilizado como entrada para o algoritmo que otimizava os grupos de pinças utilizados na cabeça de inserção, sendo o resultado desse último utilizado como entrada no algoritmo que determinava a seqüência ótima de inserção. Para o algoritmo que buscava a seqüência ótima de inserção, novamente foi utilizada uma heurística do tipo *greedy local search*. Segundo o trabalho, sua abordagem melhorava de forma significativa o tempo de montagem quando testado em produtos industriais reais.

Em outro trabalho é proposta uma solução para uma máquina tipo *multi-head* com alimentadores de componentes móveis e fixos. O autor, definindo que máquina se movia nas direções X e Y de forma simultânea (ALTINKEMER, 2000), calculou a distância percorrida pela máquina como a soma do máximo percorrido em X e o máximo percorrido em Y. Ele considerou duas situações, uma onde o alimentador de componentes era fixo, e outra onde o alimentador de componentes era móvel. Na situação onde o alimentador era móvel, ele definiu um ponto próximo ao alimentador de ferramentas, como sendo o ponto onde os componentes seriam carregados pela cabeça de inserção. Nessa condição, enquanto a cabeça inseria um componente, o alimentador posicionava o próximo componente no ponto determinado para carga. Com isso, todos os componentes possuíam o mesmo ponto de origem e também o mesmo ponto destino, assim o problema pode ser formulado como um problema de roteamento de veículos com capacidades independentes. Para este modo de operação, como a distância do componente ao ponto onde ele seria inserido na PCB, não dependia da posição do mesmo no alimentador, o problema de alocação de componentes no alimentador não precisaria estar integrado ao problema da seqüência de inserção de componentes. Para o caso onde o alimentador de componentes era fixo, eles formularam o problema como uma combinação de problemas de atribuição e problemas de roteamento de veículos (VRP) para cada componente. Primeiramente eles resolviam o VRP associado a cada componente em todas as posições possíveis no alimentador. Então, eles utilizam o resultado do problema de VRP como custo para a atribuição de um dado componente a uma dada posição. O trabalho não publica comparações dos resultados com outros trabalhos. Funcionalmente a abordagem adotada não é interessante, pois ela altera o modo de operação da máquina, e do modo como foi formulada cria uma série de movimentos desnecessários no alimentador de componentes.

Em outra publicação, o autor modelou uma situação bastante realista em linhas de produção com *mix* elevado e pouco volume. A utilização da solução ótima para cada PCB implica sempre em um novo *setup* de componentes nos alimentadores de componentes. O *setup* é uma operação manual e muito lenta. Burke (BURKE *et al*, 2001) propôs um trabalho onde o *setup* de componentes no alimentador não era alterado para a produção de diferentes placas. O autor utilizou uma abordagem hierárquica, assim como os demais trabalhos descritos anteriormente, porém com três fases, tendo obtido bons resultados experimentais quando comparado à utilização da solução ótima para cada PCB.

3.2 Trabalhos Aplicados a Máquinas de Inserção Seqüencial

Os autores Ball e Magazine (BALL; MAGAZINE, 1988) desenvolveram um algoritmo para otimização do processo de inserção automática em uma máquina insersora, com apenas uma cabeça de inserção, com placa de circuito impresso e alimentador em posição fixa. O problema formulado por eles para modelagem da seqüência das operações de inserção é um equivalente do carteiro chinês direcionado. A partir deste artigo, outras técnicas de solução para problemas semelhantes em máquinas com diferentes configurações mecânicas foram desenvolvidas. A máquina modelada por esse artigo é a mais simples possível e a redução do problema a um problema do carteiro chinês direcionado não resolveu os problemas do sequenciamento de ferramentas nem alocação dos componentes no alimentador.

Leipälä e Nevalainen (LEIPÄLÄ; NEVALAINEN, 1989) investigaram uma máquina de inserção seqüencial com cabeça simples, alimentadores móveis e uma mesa XY fixa. Em seu artigo eles separaram o problema principal em dois sub-problemas: o problema do caixeiro viajante assimétrico (PCV), representando a otimização da seqüência de inserção, e o problema quadrático da alocação (QAP), relativo à alocação dos componentes no alimentador. A heurística utilizada não garantia a solução ótima global, mas soluções muito próximas a ela podiam ser garantidas. Assim como a grande maioria dos trabalhos iniciais sobre o tema, a pesquisa tratou o problema da seqüência de inserção como um PCV e desconsiderou problemas importantes como a otimização do conjunto de ferramentas.

Em outro trabalho, o problema da alocação de componentes no alimentador e o problema da seqüência de inserção foram abordados por Kumar e Li (KUMAR; LI, 1995) como uma instância de um problema de programação linear inteira. Eles resolveram o problema atribuindo a posição dos componentes no alimentador e a seqüência de montagem para cada pinça a vácuo (quando o máquina possuir mais de uma) individualmente. Para geração da solução inicial foram utilizadas heurísticas como inserção mais próxima, vizinho mais próximo e randomização. Para melhorar a solução inicial, heurísticas como *2-opt* e *3-opt* foram utilizadas. As simulações no trabalho mostraram uma melhora de 25% no tempo de montagem, quando comparadas à abordagem utilizada na fábrica onde a máquina modelada estava instalada.

Em uma abordagem completamente diferenciada de outros trabalhos, os autores Ahmadi e Mamer (AHMADI; MAMER, 1999) modelaram o problema da seqüência de inserção como uma série de pequenos Problemas do Caixeiro Viajante co-relacionados. Segundo os resultados apresentados no artigo, a abordagem como seqüência de PCVs melhorou de forma significativa as soluções iniciais. O trabalho não abordou os problemas do sequenciamento de ferramentas nem alocação dos componentes no alimentador.

3.3 Abordagens a Outros Tipos de Máquina Utilizando Algoritmos Genéticos.

Algoritmos genéticos (AG) foram utilizados por Dikos (DIKOS *et al.*, 1997) para solução do problema de posicionamento dos componentes no alimentador em uma máquina do tipo *turret* de alta velocidade em um ambiente com *mix* de produção altamente variado. Os resultados obtidos, apesar de não quantizados pelos autores, foram descritos como melhores que a solução proposta pelo fabricante da máquina.

Também adotando algoritmos genéticos, Sun (SUN *et al.*, 2005) resolveu o problema de localização dos componentes no alimentador, visando maximizar o número de componentes carregados simultaneamente pela cabeça insersora. A estratégia tinha como objetivo reduzir a quantidade total de carregamentos efetuados pela cabeça, ao montar uma placa de circuito impresso, aumentando com isso a eficiência da máquina. Foi observado que o algoritmo deixava *slots* vazios no alimentador para atingir o objetivo de carregar o maior número possível de componentes de forma simultânea e posicionava os componentes o mais próximo do centro do alimentador, de modo a minimizar o deslocamento da cabeça insersora até a placa de circuito impresso.

3.4 Abordagens a Outros Tipos de Máquina Utilizando Busca Tabu

O autor Peter (PETER *et al.*, 2000) apresentou um trabalho onde era utilizada a metaheurística Busca Tabu para solução do problema de alocação dos componentes no alimentador e seqüência de inserção em máquinas do tipo *multi-station*. Em seu trabalho, o problema original foi particionado em duas fases e resolvido através da metaheurística Busca-Tabu e outras heurísticas mais simples para geração da solução inicial. Na implementação da BT, foram utilizadas duas estratégias, uma onde cada módulo de inserção era otimizado separadamente, visando realizar a intensificação, e outra chamada pelo autor de *diversification*

cluster rebuilding, onde as tarefas especificadas para cada módulo eram novamente redistribuídas entre os módulos de inserção. Em experimentos que comparam busca local randomizada com a metaheurística Busca Tabu, o autor demonstrou que a abordagem utilizando BT (quando as duas estratégias eram adotadas simultaneamente), conseguia resultados de maior qualidade, consumindo menos tempo computacional para uma ampla variedade de placas montadas pela máquina analisada no problema. Em média, a melhora usando BT foi de 15%. O autor destaca que o método de otimização apresentado pode ser utilizado com grande possibilidade de sucesso em outros modelos de máquinas insersoras, o que realmente é esperado, já que a BT é uma metaheurística contemporânea, voltada a solução de problemas de difícil tratabilidade.

A literatura possui uma vasta quantidade de trabalhos voltados ao problema da máquina insersora de componentes. A grande maioria aborda o problema do sequenciamento como um PCV, ou PCCD. Também a vasta maioria trata os problemas utilizando metaheurísticas hierárquicas, onde cada problema é resolvido separadamente e a solução de um problema é condição inicial do problema seguinte.

3.5 Considerações desta Pesquisa em Relação aos Trabalhos Apresentados

O diferencial do trabalho proposto por esta dissertação é abordar o Problema da Alocação de Componentes no Alimentador e o Problema do Sequenciamento de Inserção de forma conjunta, sem ignorar, contudo, o problema do Escalonamento do Conjunto de Ferramentas, que também é tratado utilizando a técnica adequada. Outro ponto importante diz respeito ao problema da sequência de inserção: diferente da grande maioria dos trabalhos disponíveis na literatura, o problema será tratado com um problema de escalonamento, onde tarefas são ordenadas em função do tempo. Através do uso de uma abordagem diferenciada, utilizando metaheurísticas atuais, bons resultados são apresentados ao longo do trabalho.

4 APRESENTAÇÃO DO PROBLEMA

Neste capítulo será apresentada uma descrição detalhada do problema, as dificuldades envolvidas na tarefa de encontrar uma boa solução e possíveis reduções dos problemas apresentados a problemas conhecidos na área de otimização.

4.1 Projeto no CAD de Placas de Circuito Impresso

No capítulo 1 foi apresentado o conceito de instruções de montagem. Nesta fase da dissertação será explicado o processo de gerar as mesmas e a contextualização destas instruções no problema.

Para geração das instruções de montagem, geralmente é utilizado um programa de CAD específico para desenvolvimento de placas de circuito impresso. Nesse programa o operador cria uma representação virtual da placa completamente montada com todas as conexões entre os componentes. Esse processo pode ser manual ou automatizado e é chamado de roteamento.

Estando a placa finalizada, o programa permite exportar diversos dados, dentre eles, os necessários para geração da instrução de montagem.

Na Figura 17, dentre as várias opções disponíveis, é ilustrada a interface gráfica de um CAD típico para o desenvolvimento de placas de circuito impresso.

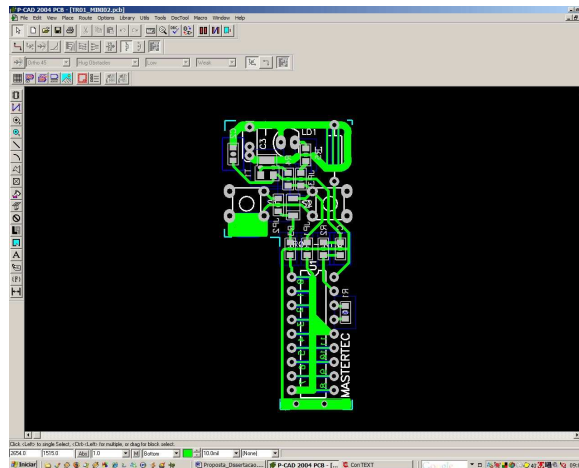


FIGURA 17– CAD PARA DESENVOLVIMENTO DE PLACAS DE CIRCUITO IMPRESSO.

4.2 Integração Entre o Projeto do CAD e a *Pick-and-Place* (CAM)

Além das listas de dados fornecidas pelo CAD, para geração da instrução de montagem, uma terceira lista é fornecida pelo operador da máquina, que indica em qual posição do alimentador de componentes da máquina cada tipo de componente foi colocado. Na Figura 18 pode-se observar a foto de uma máquina comercial com os alimentadores de componentes (*magazines*) nela destacados. Cada magazine da Figura 18 possui oito posições para colocação de componentes.



FIGURA 18– MÁQUINA COMERCIAL E POSIÇÃO DOS ALIMENTADORES.

Apesar de normalmente fornecido pelo operador, essa informação é passível de otimização, ou seja, o algoritmo de otimização pode resolver o problema da alocação dos

componentes no alimentador, uma vez que a localização dos componentes afeta o tempo total de montagem da placa.

4.3 O Problema de Otimização a ser Resolvido – Contexto Separado

Para melhor compreensão, o problema da otimização do ciclo de montagem de uma máquina insersora automática de componentes será dividido em três problemas distintos, que no momento da solução não podem ser integralmente separados.

4.3.1 Determinação do Conjunto Ideal de Ferramentas e Escalonamento das Mesmas

É necessário determinar o conjunto ideal de ferramentas na cabeça insersora a cada instante de tempo, de modo a minimizar o número de trocas e maximizar a quantidade de componentes pinçados. Desconsiderando o problema de otimização de trajetória, a associação entre componentes e ferramentas pode ser abordada como um problema de Família de Partes, mais especificamente identificação de *Clusters*. Já a seqüência de carregamento e troca de ferramentas pode ser abordada como um problema de escalonamento.

4.3.2 Problema da Alocação de Componentes no Alimentador de Componentes

É necessário determinar a posição ótima de cada componente no alimentador de componentes, de modo a minimizar a trajetória da cabeça de inserção, o que afeta diretamente o tempo total de montagem.

4.3.3 Ordem de Execução das Inserções de Componentes

Determinar a ordem de execução das inserções, baseada na alocação dos componentes no alimentador de componentes e baseado nos grupos de ferramentas, de modo a minimizar o deslocamento linear total e também a quantidade de trocas de ferramentas. Este problema pode ser modelado como um problema de escalonamento. Neste caso, o problema de escalonamento utiliza como condições iniciais uma possível solução parcial dos problemas anteriores.

4.4 O Problema de Otimização a ser Resolvido – Contexto Global

Conforme análise separada dos problemas, os três estão interligados em um objetivo comum, que é reduzir o tempo total de montagem de uma placa. A visão separada do problema completo é importante, pois permite compreender de forma mais clara a dimensão do mesmo, porém utilizá-la de forma direta para solucionar o problema é inviável, já que o problema de escalonamento, conforme Viana (VIANA, 1998), é classificado como um problema NP-Difícil, sendo utilizado para modelagem de dois sub-problemas. Neste contexto, a solução do problema geral, resolvendo os problemas separadamente, de maneira alguma garante a qualidade da solução. Apenas pode-se garantir que uma solução viável foi encontrada. Ainda é possível que, ao otimizar um determinado problema separadamente, a solução do problema seguinte seja de má qualidade devido a condições iniciais desfavoráveis. Utilizando a resolução de forma individual, não é nem mesmo possível garantir que a solução esteja convergindo para um ótimo local.

Com base nos três problemas descritos nos itens anteriores, é possível formalizar o enunciado do problema geral como sendo:

“Otimizar o conjunto de ferramentas e a seqüência de trocas das mesmas na cabeça insersora, a alocação dos componentes no alimentador e a seqüência de inserção dos componentes, de modo a reduzir o tempo total de montagem de uma PCB.”

Fica implícito que a otimização da seqüência de inserção deve ter como objetivo otimizar a trajetória.

O enunciado anterior valida a afirmação: “Enunciar problemas de otimização combinatória costuma ser simples, porém devido a quantidade de soluções, resolvê-los é uma tarefa complicada.” (OSMAN; LAPORTE, 1996)

5 MODELO PROPOSTO

O objetivo deste capítulo é apresentar o modelo desenvolvido neste trabalho. O item 5.1 apresenta uma descrição sucinta do modelo. O item 5.2 apresenta de forma detalhada a formulação matemática adotada. Finalmente, o item 5.3 apresenta a arquitetura utilizada para implementação do modelo proposto.

5.1 Problema Estudado para Definição do Modelo

A Figura 19 ilustra, em forma de diagrama, a vista superior da máquina de inserção automática de componentes do tipo *multi-head* que será utilizada para a modelagem. Essa máquina possui a mesa de fixação da PCB e alimentador de componentes fixos.

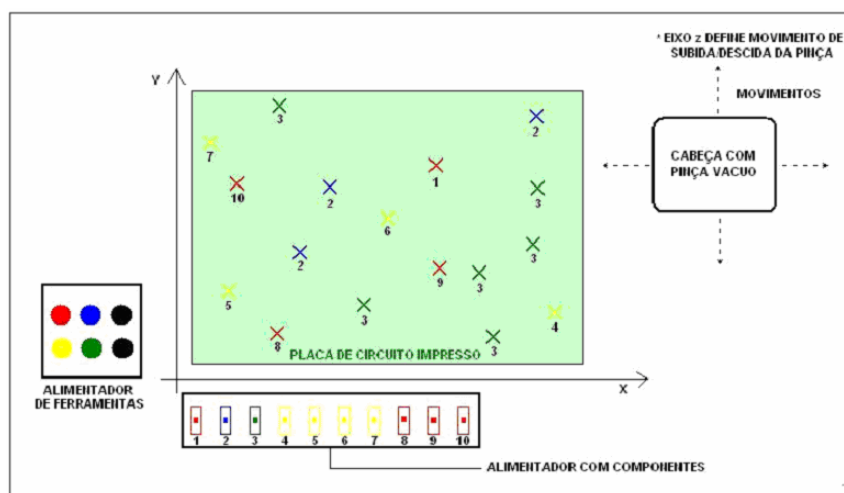


FIGURA 19- VISTA SUPERIOR DA ESTRUTURA DO SISTEMA MODELADO.

No diagrama da Figura 19 pode-se observar, à esquerda da Figura, o alimentador de ferramentas. A máquina modelada é capaz de efetuar trocas automáticas de ferramentas neste alimentador sempre que necessário. Na parte inferior do diagrama está localizado o alimentador de componentes com apenas dez componentes ilustrados. Com o objetivo de

modelar uma máquina real, o alimentador do modelo terá capacidade para até cem componentes. No lado direito do diagrama está localizada a cabeça inseridora com uma ou mais pinças a vácuo. No modelo proposto o número de pinças é um parâmetro que pode variar de 1 a 20. A cabeça de inserção tem liberdade para movimentar-se por toda superfície da máquina (eixos X e Y), atingindo o alimentador de ferramentas, o alimentador de componentes e a PCB. As pinças são independentes e tem liberdade de movimento no eixo Z , com o objetivo de carregar e descarregar componentes. O quadro com marcações em “X” na Figura 19, ilustra uma possível placa de circuito impresso onde os componentes serão inseridos. As marcações dentro da placa indicam a posição onde os componentes disponíveis no alimentador devem ser inseridos.

Para a máquina exemplificada no diagrama da Figura 19, considerando que a seqüência de movimentos já foi definida, a placa está posicionada no local correto e os componentes e ferramentas já estão disponíveis em seus respectivos alimentadores. O fluxograma do ciclo de montagem de uma placa de circuito impresso é apresentado na Figura 20.

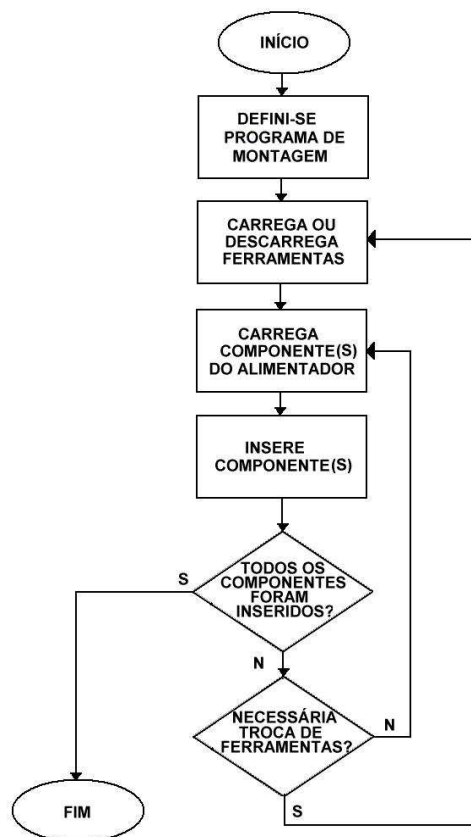


FIGURA 20- CICLO DE MONTAGEM DA MÁQUINA MODELADA.

Inicialmente, o programa de montagem específico para uma determinada placa e conjunto de componentes é carregado na máquina. O programa de montagem, conforme descrito anteriormente, é composto por uma lista que descreve todas as operações que a máquina irá realizar. As operações possíveis são: carregar ferramentas, descarregar ferramentas, carregar componente do alimentador e inserir um componente em determinada posição. No passo seguinte, conforme as operações descritas no programa de montagem, a máquina carrega ou descarrega ferramentas de modo a assumir a condição inicial imposta pelo programa de montagem. Com as ferramentas devidamente carregadas na cabeça de inserção, a máquina busca componentes no alimentador e os insere na placa conforme o programa de montagem. O passo seguinte consiste em verificar se todos os componentes foram inseridos (programa de montagem finalizado). Em casos negativos, quando for necessária a inserção de novos componentes, a máquina troca ferramentas da cabeça de inserção (se necessário) e inicia um novo ciclo de inserção. O processo se repete até que a condição de parada tenha sido atingida.

O programa de montagem é o resultado da metaheurística híbrida proposta por essa dissertação. A Figura 21 ilustra, em forma de blocos, o fluxo de dados necessários e também gerados pela metaheurística proposta.

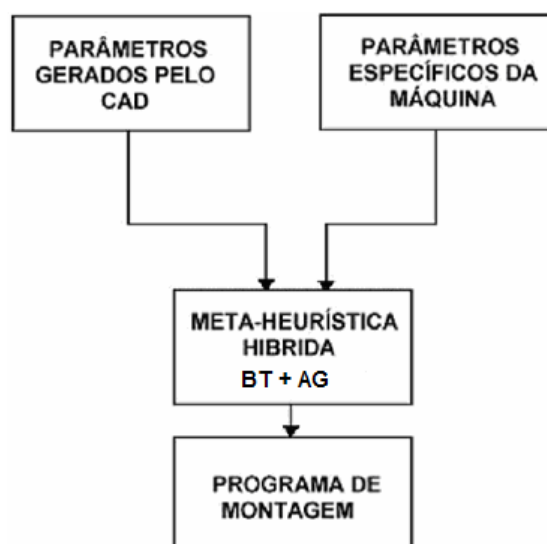


FIGURA 21– INFORMAÇÕES NECESSÁRIAS PARA IMPLEMENTAÇÃO DA METAHEURÍSTICA.

Conforme citado no capítulo anterior, parte da informação utilizada para resolver o problema é oriunda do CAD, onde parâmetros como o tipo de componente, e sua localização na placa de circuito impresso são definidos. Restam ainda parâmetros específicos da máquina modelada, são eles:

- a) Associação do tipo de componente e ferramenta(s) capaz(es) de manuseá-lo;
- b) Quantidade máxima de ferramentas carregadas pela cabeça de inserção;
- c) Coordenadas cartesianas da posição das ferramentas nos alimentadores, da localização específica de cada carregador nos alimentadores de componentes e da posição de referência da PCB. Estas coordenadas serão utilizadas para o cálculo da distância percorrida pela cabeça de inserção em seus movimentos. Todas elas devem ser definidas em relação a um mesmo ponto de origem.

5.2 Formulação do Modelo

Para definição da função objetivo foi utilizada uma formulação própria, porém usando uma notação, onde possível, semelhante à utilizada por Ayob (AYOB; KENDALL, 2002). O modelo por ele proposto não considerava os tempos de troca de ferramenta, além de ter sido desenvolvido para um modelo de insersora diferente da adotada neste trabalho, o que altera completamente as características do modelo. Baseado nisso, apenas a notação foi mantida quando possível. Basicamente, é necessário minimizar o tempo total de montagem de uma placa de circuito impresso, daqui para frente denominado “tempo de ciclo”. O tempo de ciclo é uma função linear, dependente diretamente da distância percorrida pela cabeça de inserção, da quantidade de componentes inseridos e do número de trocas de ferramentas realizadas. Baseado nestas informações, a função objetivo é apresentada de forma compacta na equação (10):

Minimizar:

$$TC = \sum_{n=1}^I \left[\frac{D_n}{V_r} \right] + I \times Tu + I \times Tp + P \times Tf \quad (10)$$

onde,

n	Indexa as inserções, onde $n = 1..I$
TC	Representa o tempo total de montagem (tempo de ciclo)
I	Número total de componentes inseridos
P	Número de troca de ferramentas efetuadas
D_n	Representa a função que descreve a distância percorrida pela cabeça de inserção na inserção n
Tu	Tempo para inserir (<i>unload</i>) um componente (igual para todos tipos)
Tp	Tempo para carregar (<i>pick</i>) um componente (igual para todos tipos)
Tf	Tempo para troca de uma ferramenta (igual para todos tipos)

V_r Velocidade média da cabeça de inserção.

A função D_n é apresentada na equação (11):

$$D_n = A \times D_{C1} + B \times D_{C2} + C \times D_{C3} \quad (11)$$

onde,

D_{C1} , D_{C2} e D_{C3} representam as distâncias percorridas pela cabeça de inserção em três casos distintos, apresentados posteriormente na Figura 22. Na equação (11), os coeficientes A , B e C representam variáveis condicionais que anulam termos da equação geral da distância, aplicáveis somente a casos específicos:

$$A = \begin{cases} 1, \text{ caso o próximo passo seja uma inserção simples (figura 22-a)} \\ 0, \text{ caso contrário} \end{cases} \quad (12)$$

$$B = \begin{cases} 1, \text{ caso ocorra apenas carregamento de componentes} \\ \text{antes da inserção (figura 22-b)} \\ 0, \text{ caso contrário} \end{cases} \quad (13)$$

$$C = \begin{cases} 1, \text{ caso ocorra troca de ferramentas e carregamento de componentes} \\ \text{antes da inserção (figura 22-c)} \\ 0, \text{ caso contrário} \end{cases} \quad (14)$$

$$A, B, C \in \{ 0, 1 \}. \quad (15)$$

A Figura 22 está dividida conforme enumeração abaixo:

- a) É apresentado o caso de inserção simples;
- b) É apresentado o caso de inserção com recarga de componente(s);
- c) É apresentado o caso da inserção com troca de ferramentas e recarga de componentes.

Todas as definições de variáveis e constantes da Figura 22 são descritas após o equacionamento dos três casos apresentados.

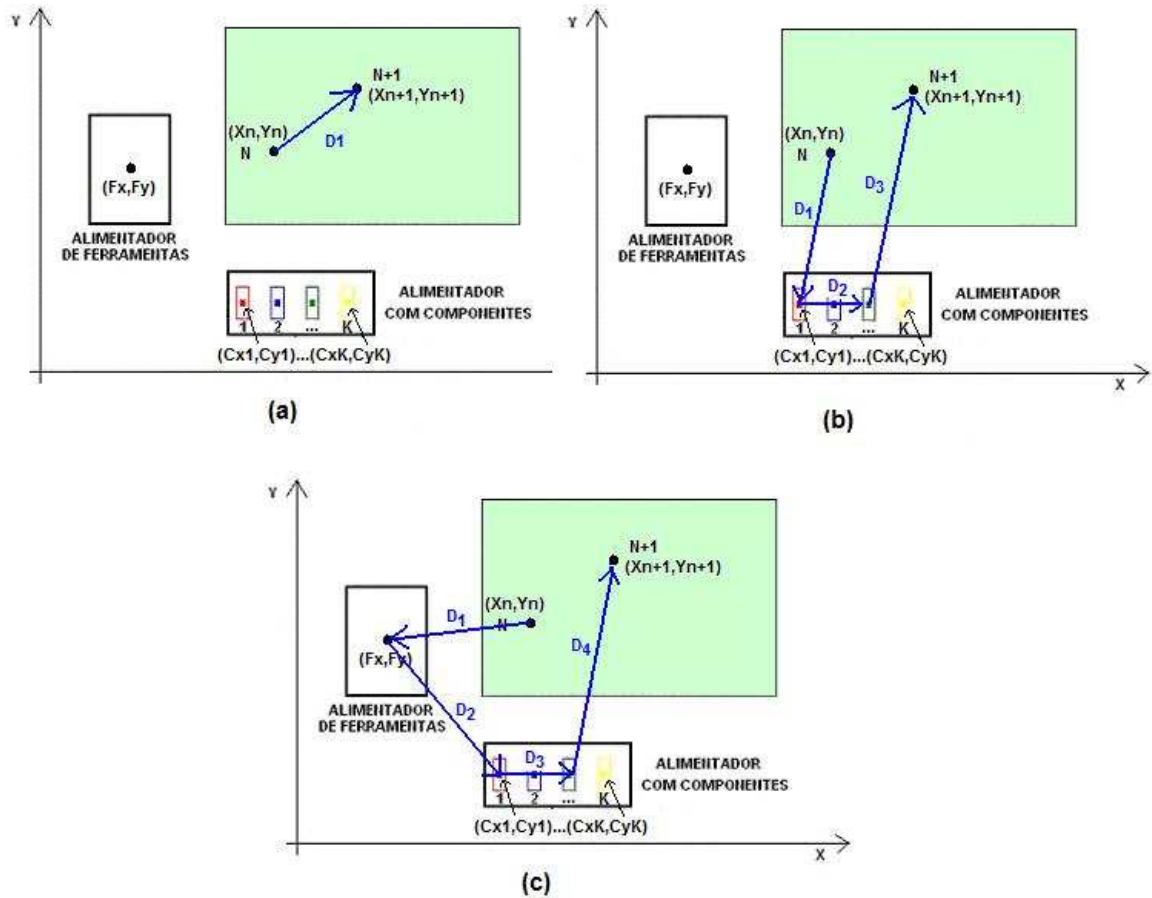


FIGURA 22– CASOS POSSÍVEIS PARA O MODELO DA DISTÂNCIA PERCORRIDA PELA CABEÇA DE INSERÇÃO.

Na inserção simples, exemplificada na Figura 22 (a), a distância percorrida pela cabeça inseridora é composta apenas do movimento da cabeça inseridora da posição do componente indexada pela variável n para a posição do componente indexado por $n+1$. Essa distância é representada por D_1 . Esse tipo de movimento somente ocorre em máquinas com mais de uma pinça a vácuo na cabeça inseridora (*multi-head*). Máquinas com apenas uma pinça sempre realizam uma nova recarga após uma inserção.

Para o caso descrito anteriormente (Figura 22-a), DCI é expressa por:

$$D_{C1} = D_1 \equiv \sqrt{(X_n - X_{n+1})^2 + (Y_n - Y_{n+1})^2}. \quad (16)$$

Para o caso da inserção com recarga de componentes, exemplificado na Figura 22 (b), a distância percorrida é composta por três movimentos distintos. A cabeça inseridora sai da posição N e vai até o primeiro componente a ser carregado (distância representada por D_1), a seguir, a cabeça executa um número de deslocamentos igual a quantidade de componentes carregados. A soma desses deslocamentos é representada por D_2 . Após o último componente

ser carregado, a cabeça insersora se desloca até a posição do componente indexado por $n+1$. Essa distância é representada por D_3 . Cabe salientar um caso particular, onde em máquinas com apenas uma pinça na cabeça insersora ($CR = 1$) a distância D_2 é sempre nula, uma vez que apenas um componente é carregado.

Para o caso ilustrado na Figura 22-b D_{C2} é expressa por:

$$D_{C2} = D_1 + D_2 + D_3 \quad , \quad (17)$$

onde,

$$D_1 = \sqrt{(X_n - Cx_{n+1})^2 + (Y_n - Cy_{n+1})^2} \quad , \quad (18)$$

$$D_2 = \sum_{j=n+1}^{n+CR} \sqrt{(Cx_j - Cx_{j+1})^2 + (Cy_j - Cy_{j+1})^2} \quad , \quad (19)$$

$$D_3 = \sqrt{(Cx_{n+CR} - X_{n+1})^2 + (Cy_{n+CR} - Y_{n+1})^2} \quad . \quad (20)$$

Finalmente, para o caso da inserção com troca de ferramentas e recarga de componentes, exemplificado na Figura 22 (c), a distância percorrida é composta por quatro movimentos distintos. Inicialmente, a cabeça de inserção sai da posição N e se desloca até o alimentador de ferramentas para executar a sua troca. Essa distância é representada por D_1 . Especificamente para este caso, o modelo sofreu uma simplificação em relação ao caso real. Em função da proximidade das ferramentas, foi considerado que todas as ferramentas estão localizadas em um mesmo ponto (Fx, Fy) , sendo esse ponto o centro geométrico do alimentador de ferramentas. No passo seguinte, a cabeça de inserção se desloca do alimentador de ferramentas para o primeiro componente a ser carregado, e essa distância é representada por D_2 . A seguir, a cabeça de inserção executa um número de deslocamentos igual a quantidade de componentes carregados. A soma destes deslocamentos é representada por D_3 . Após o último componente ser carregado, a cabeça insersora se desloca até a posição do componente indexado por $n+1$, cuja distância é representada por D_4 . Conforme o caso apresentado em 22 (b), uma simplificação devido a um caso particular também ocorre. Em máquinas com apenas uma pinça na cabeça insersora (*sequential pick-and-place*), a distância D_3 também é sempre nula.

Para o caso ilustrado na Figura 22-c, D_{C3} é expressa por:

$$D_{C3} = D_1 + D_2 + D_3 + D_4, \quad (21)$$

onde,

$$D_1 = \sqrt{(X_n - Fx)^2 + (Y_n - Fy)^2}, \quad (22)$$

$$D_2 = \sqrt{(Fx - Cx_{n+1})^2 + (Fy - Cy_{n+1})^2}, \quad (23)$$

$$D_3 = \sum_{j=n+1}^{n+CR} \sqrt{(Cx_j - Cx_{j+1})^2 + (Cy_j - Cy_{j+1})^2}, \quad (24)$$

$$D_4 = \sqrt{(Cx_{n+CR} - X_{n+1})^2 + (Cy_{n+CR} - CY_{n+1})^2}. \quad (25)$$

Para todos os casos as restrições são:

$$\begin{aligned} D_{C1}, D_{C2}, D_{C3} &\geq 0, \\ D_1, D_2, D_3, D_4 &\geq 0, \\ 1 \leq CR &\leq F_{MAX}, \\ CR &\in \{1, 2, \dots, F_{MAX}\}. \end{aligned} \quad (26)$$

Para as distâncias expressas pelas equações D_{C1} eq. (16), D_{C2} eq. (17) e D_{C3} eq. (21), a nomenclatura utilizada é:

D_{C1}	Distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 22 (a).
D_{C2}	Distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 22 (b).
D_{C3}	Distância percorrida pela cabeça de inserção para o caso ilustrado na Figura 22 (c).
D_1, D_2, D_3, D_4	Distâncias percorridas pela cabeça de inserção para os casos expressos na Figura 22.
X_n, Y_n	Coordenadas cartesianas da inserção indexada por n .
Cx_n, Cy_n	Coordenadas cartesianas do componente no alimentador, inserido na inserção n .
Fx, Fy	Coordenadas cartesianas do centro médio do alimentador de ferramentas.
CR	Número de componentes carregados do alimentador em uma operação de carregamento.
$Fmax$	Número máximo de ferramentas simultâneas na cabeça de inserção.
j	Indexador de carregamentos.
n	Indexador de inserções.

5.2.1 Considerações Sobre a Função Objetivo

A função o objetivo (FO) proposta na equação (10) é apresentada de forma simplificada na equação (27), e modela os parâmetros que são necessários ao processo de otimização.

$$TC = \sum_{n=1}^I \left[\frac{D_n}{Vr} \right] + I \times Tu + I \times Tp + P \times Tf \quad (27)$$

O número de componentes inseridos I , é um parâmetro fornecido pelo CAD. Ele não pode ser alterado e é uma constante na FO. O mesmo vale para Tu e Tp , onde estes parâmetros descrevem tempos que a máquina leva para executar determinadas operações. Com base nisto os termos $I \times Tu$ e $I \times Tp$ são constantes que em nada afetam o processo de otimização. A vantagem desses termos estarem presentes na FO está no fato de que o valor fornecido pela mesma, deixa de ser comparativo e passa a ser quantitativo, ou seja, permite comparar diretamente o resultado obtido com um caso real.

5.2.2 Caracterização dos Parâmetros da Máquina Inserora

Para que a função objetivo e todo algoritmo consigam representar de forma realista uma máquina inserora de componentes, alguns parâmetros que definem as características mecânicas da máquina precisam ser definidos. Estes parâmetros são apresentados na Tabela 2.

TABELA 2 – PARÂMETROS DA MODELAGEM DE UMA MÁQUINA INSERSORA REAL.

PARÂMETRO	DESCRIÇÃO
F_{MAX}	Número máximo de ferramentas simultâneas na cabeça de inserção.
Vr	Velocidade média da cabeça de inserção.
Tu	Tempo para inserir um componente
Tp	Tempo para carregar um componente
Tf	Tempo para troca de uma ferramenta
Fx, Fy	Coordenadas cartesianas do centro médio do alimentador de ferramentas.
Cx_m, Cy_n	Coordenadas cartesianas da posição dos componentes no alimentador
Bx, By	Coordenadas cartesianas do ponto (0,0) da PCB em relação a origem da máquina

Na Tabela 2, a coluna da esquerda apresenta o nome do parâmetro utilizado no modelo. Estes nomes, juntamente com todas as outras variáveis foram apresentados na seção 5.2 quando a função objetivo foi apresentada. Na coluna a direita é apresentada a descrição física do parâmetro. Cabe salientar que todos os parâmetros apresentados na Tabela 2 são dimensionais, definindo quantidades, velocidades e posições físicas.

5.3 Arquitetura do Modelo

A arquitetura do modelo para seleção de famílias de partes, definição da posição dos componentes no alimentador e geração dos escalonamentos de ferramentas e inserções de componentes é ilustrada na Figura 23.

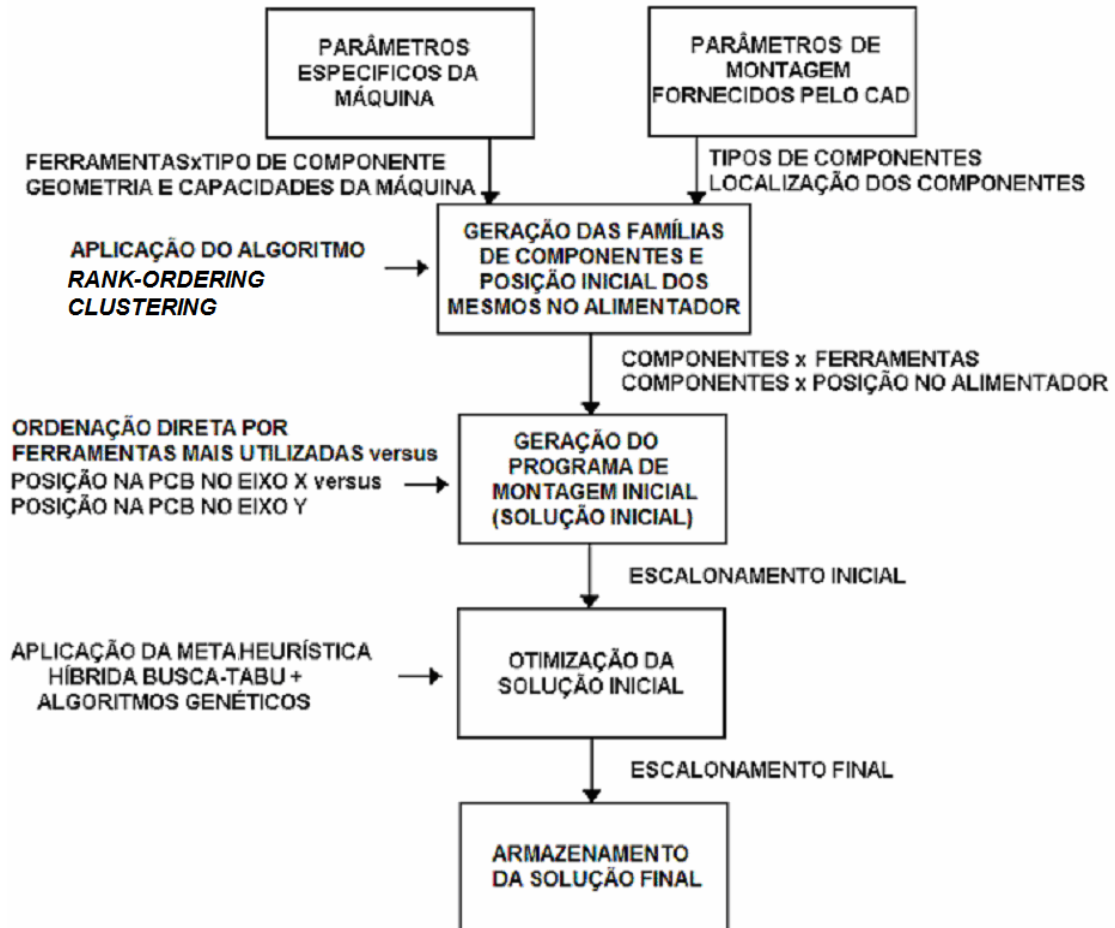


FIGURA 23– ARQUITETURA DO MODELO PROPOSTO.

A arquitetura foi dividida em 5 fases. Na primeira, são reunidos os dados fornecidos pelo CAD de circuitos eletrônicos e parâmetros da máquina, relativos as suas capacidades e geometria.

Na segunda fase, utilizando a TG e aplicando um algoritmo de identificação de *clusters*, especificamente o algoritmo *rank-order clustering*, são definidas as famílias de partes. Este algoritmo determinístico baseado em ordenação associa em grupos os componentes e pinças a vácuo disponíveis, levando em consideração os componentes utilizados na montagem e as capacidades e quantidades de pinças disponíveis. Essa

classificação não sofre alterações posteriores. O passo seguinte consiste na definição da posição inicial dos componentes no alimentador, essa definição apenas precisa ser válida, assim os componentes são ordenados em função da utilização, a partir do centro do alimentador.

Na terceira fase é gerada a solução inicial de escalonamento necessária para implementação da metaheurística Busca Tabu. A solução inicial é gerada através da ordenação direta da posição geométrica dos componentes na placa de circuito impresso *versus* grupos de ferramentas mais utilizadas. Essa aproximação garante o menor número possível de troca de ferramentas e também que a inserção seguinte à atual seja vizinha não muito distante da mesma, evitando grandes deslocamentos por parte da cabeça de inserção. A política utilizada para determinação do escalonamento do *setup* visa maximizar o número de componentes transportados de forma simultânea pela cabeça de inserção no alimentador de componentes e é realizada por um algoritmo determinístico.

A Figura 24 ilustra a solução inicial gerada pelo algoritmo descrito anteriormente, para o escalonamento completo em um problema hipotético.

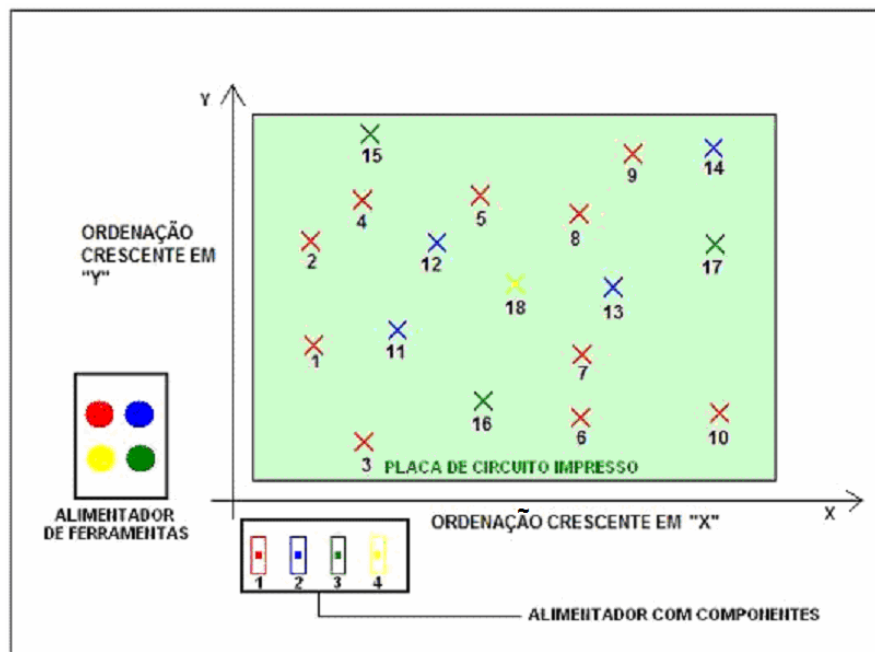


FIGURA 24– SOLUÇÃO INICIAL GERADA PELO ALGORITMO PROPOSTO.

No exemplo da Figura 24, a seqüência de inserção é definida pelos números apresentados abaixo das marcações na placa de circuito impresso. Em relação ao problema exemplo, a cabeça inserora da máquina tem apenas uma pinça a vácuo, o que permite à

mesma carregar um componente por vez, sendo que existem apenas quatro componentes diferentes, que utilizam ferramentas diferentes para sua inserção. A Tabela 3 apresenta para o problema hipotético a associação entre o tipo de componente, quantidade de inserções e modelo da pinça a vácuo utilizada para carregar o componente.

TABELA 3 – ASSOCIAÇÃO ENTRE COMPONENTES, INSERÇÕES E FERRAMENTAS

COMPONENTE	QUANTIDADE DE INSERÇÕES	MODELO DE FERRAMENTA
Tipo 1	10	1
Tipo 2	4	2
Tipo 3	3	3
Tipo 4	1	4

Seguindo o que foi proposto pelo algoritmo, o grupo de componentes do tipo 1 é o primeiro a ser escalonado, pois apresenta a ferramenta com maior utilização. A sequência de inserção é primeiramente ordenada em relação ao eixo *X* e posteriormente em relação ao eixo *Y*. A seguir são escalonados os demais componentes seguindo o mesmo processo.

Na quarta fase, depois de gerado o escalonamento inicial, é possível aplicar a metaheurística híbrida BT+AG para obtenção do escalonamento final e da posição final dos componentes no alimentador de componentes. Na implementação da metaheurística híbrida, cada metaheurística possui uma função bem definida. A BT é responsável pela busca da solução principal. Os algoritmos genéticos serão utilizados com o objetivo de diversificar a busca, quando a BT perder eficiência e não conseguir mais melhorar a qualidade da solução. Na Figura 25 é apresentado o diagrama em blocos do encadeamento realizado entre a Busca Tabu e o Algoritmo Genético.

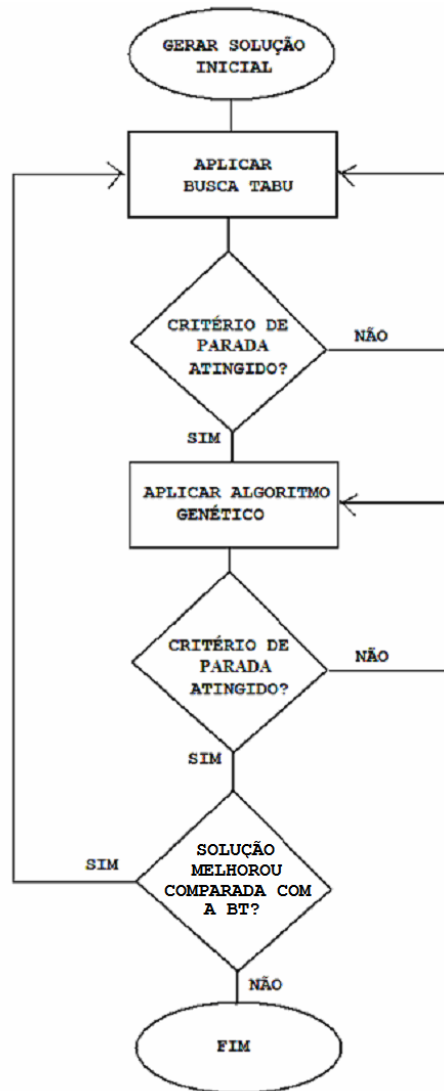


FIGURA 25– ENCADEAMENTO ENTRE A BUSCA TABU E O ALGORITMO GENÉTICO.

No diagrama da Figura 25, a partir da solução inicial, a BT é executada até que o critério de parada, a ser definido posteriormente, seja atingido. Durante a execução uma lista com as melhores soluções encontradas é armazenada. Essa lista serve como população inicial para o AG, que é executado até que o critério de parada, a ser definido posteriormente, tenha sido atingido. Posteriormente a solução gerada pelo AG é comparada com a solução encontrada anteriormente pela BT, em caso de melhora, a BT é novamente executada, caso a solução não melhore, a metaheurística híbrida é finalizada.

Na quinta fase, do diagrama da Figura 23, a partir da solução final gerada, um programa de montagem é gerado adequando o mesmo a detalhes específicos da máquina onde o mesmo será aplicado, também a definição da posição de cada componente nos alimentadores são armazenadas para uso posterior pela máquina insersora.

6 IMPLEMENTAÇÃO DO MODELO E ALGORITMO

Este capítulo descreve, em detalhes, a codificação e implementação da metaheurística híbrida. Inicialmente, a codificação foi realizada em linguagem “C”, porém para facilitar e acelerar a depuração e desenvolvimento, o código foi reescrito (quando o desenvolvimento estava em 10% do total) em *VisualBasic 6.0*. Essa alteração permitiu simplificar a implementação e a análise do funcionamento em forma gráfica, o que agilizou o desenvolvimento do algoritmo.

6.1 Processamento Inicial dos Arquivos do CAD

Com base no projeto do CAD (apresentado no capítulo 4), são geradas cinco estruturas de dados básicas que definem todos os parâmetros do problema. As estruturas armazenam as seguintes informações: posição dos componentes na PCB, alimentador onde determinado componente está localizado, posição do alimentador na máquina, ferramenta associada a cada componente e finalmente, uma estrutura secundária que faz a correspondência entre número real e número lógico de um alimentador.

6.2 Estruturas de Dados Básicas para Definição dos Parâmetros do Problema

Ao gerar as estruturas principais para definição do problema, cada componente inserido recebe um número único, que é utilizado como referência para a pesquisa nas estruturas. O mesmo ocorre com os alimentadores e ferramentas. Cada uma das estruturas é apresentada a seguir.

6.2.1 Posição dos Componentes na PCB

Essa estrutura armazena a posição de inserção de cada componente na PCB. A Figura 26 ilustra, de forma visual, a informação armazenada.

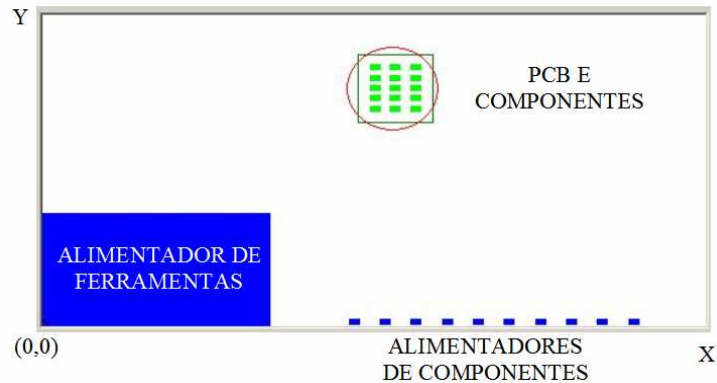


FIGURA 26– INFORMAÇÃO DA POSIÇÃO DOS COMPONENTES NA PCB.

Na Figura 26, o círculo delimita o espaço onde os componentes estão localizados na PCB. A estrutura é organizada da seguinte forma: número do componente *versus* posição (x,y) na PCB. Essa estrutura é usada especificamente no problema da seqüência de inserção.

6.2.2 Alimentador onde cada Componente Está Localizado

Essa estrutura armazena em qual alimentador cada componente está localizado. A Figura 27 ilustra, de forma visual, a informação armazenada.

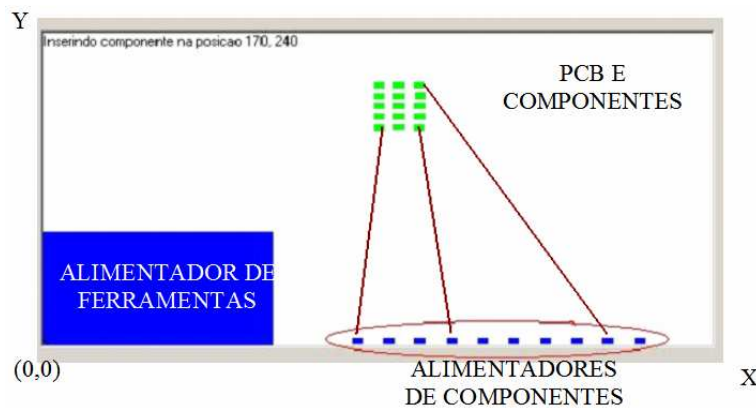


FIGURA 27– INFORMAÇÃO DO ALIMENTADOR USADO POR CADA COMPONENTE.

Na Figura 27, a elipse delimita a região utilizada pelos alimentadores, como exemplo, as três setas indicam onde o componente inserido está armazenado no momento da carga. A organização dos dados segue a forma descrita a seguir: número do componente *versus* o alimentador onde ele está localizado. Cada alimentador recebe, durante a geração das estruturas, um número específico e único, que permite a sua identificação.

6.2.3 Posição Física do Alimentador na Máquina

Essa estrutura armazena a posição física (x,y) dos alimentadores no modelo representado pela Figura 26. Os dados estão organizados no formato: número do alimentador *versus* posição (x,y) na máquina. O número do alimentador é o mesmo utilizado na estrutura apresentada no item anterior. Definida esta estrutura, sempre que o algoritmo necessitar saber onde a posição de carga de um componente está localizada, primeiramente deve ser acessada a estrutura que armazena o número do componente *versus* número do alimentador utilizado. Posteriormente, de posse do número do alimentador, um acesso à estrutura apresentada nesse tópico, permite localizar a posição física no modelo de um determinado alimentador.

6.2.4 Ferramenta Associada a cada Componente

Essa estrutura armazena que tipo de pinça a vácuo (ferramenta) cada componente necessita para ser inserido. A Figura 28 ilustra, de forma visual, a informação armazenada.

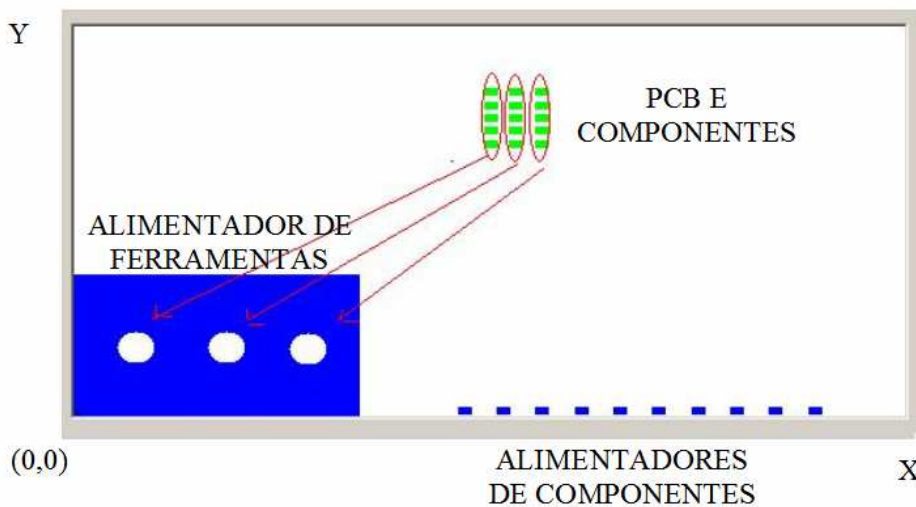


FIGURA 28– INFORMAÇÃO DA FERRAMENTA NECESSÁRIA PARA INSERÇÃO DE UM COMPONENTE.

A Figura 28 apresenta um caso onde quinze componentes são inseridos por três ferramentas. As ferramentas são representadas pelos círculos à esquerda. As famílias de componentes são delimitadas por três elipses. As setas indicam qual ferramenta será utilizada por cada família de componentes. A organização dos dados segue a forma: Número do componente *versus* ferramenta utilizada. Salienta-se que, apesar dos componentes estarem separados em famílias, o armazenamento e o tratamento das informações, assim como nas estruturas anteriores, é realizado de forma individual.

6.2.5 Correspondência entre Endereço lógico e Endereço Físico de um Alimentador

Essa estrutura é secundária, ou seja, não é necessária para definir o problema. Ela faz a correspondência entre o número físico do alimentador (definido durante a geração das estruturas) e o número lógico. Ela foi criada de modo a simplificar as movimentações de dados durante a otimização da alocação de componentes nos alimentadores. Ela permite que a posição de um alimentador seja alterada, sem modificar as estruturas que definem o problema. A Figura 29 ilustra como o mapeamento dos dados é realizado.

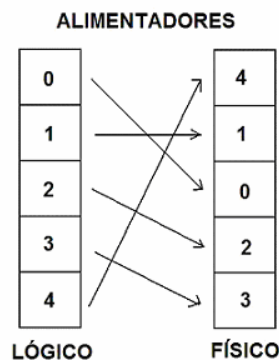


FIGURA 29– CORRESPONDÊNCIA ENTRE NÚMEROS LÓGICOS E FÍSICOS DE ALIMENTADORES.

Na Figura 29, a primeira coluna apresenta o número lógico do alimentador, enquanto a segunda, apresenta o número físico. Com isso, o acesso à posição dos alimentadores pode ser mapeado, simplesmente alterando a correspondência entre os valores desta lista. Essa operação, além de ser mais simples de ser efetuada que alterar as estruturas principais, evita que as mesmas sejam alteradas.

6.3 Estruturas de apoio à implementação do algoritmo.

Além das estruturas que definem o problema, outras, de pequeno e grande porte, foram criadas, visando implementar o algoritmo de otimização proposto. Neste caso específico, a quantidade e organização das mesmas é dependente de como o algoritmo foi codificado, porém as principais estruturas de apoio (algumas compartilhadas entre a Busca Tabu e Algoritmos Genéticos), que independem da codificação são:

- a) Lista tabu;
- b) Vizinhanças geradas (filhos dos cruzamentos da população para o AG);

c) Melhores soluções geradas pela BT (população inicial AG).

A lista tabu contém de forma completa, as últimas soluções visitadas pelo algoritmo de Busca Tabu. Os movimentos efetuados não são armazenados. Ela é utilizada apenas na implementação da Busca Tabu. As estruturas de dados com vizinhanças ou filhos de cruzamentos no caso do AG, são utilizadas para armazenar todas as novas soluções geradas a cada iteração de um determinado algoritmo. Por fim, uma estrutura contendo as melhores e mais diferentes soluções geradas pela BT (soluções de elite), são armazenadas e utilizadas como população inicial do algoritmo genético.

6.4 Formato das Soluções

Existem duas maneiras de representar uma solução. Uma consiste na solução reduzida ou lógica e outra, a solução completa. A solução reduzida não é aplicável diretamente à máquina insersora e ela pressupõe que a máquina possui inteligência para separar as várias operações agrupadas em um único comando. A solução completa, pressupõe que a máquina não dispõe de inteligência e todas as operações são separadas em operações menores. Essa solução pode ser utilizada diretamente em uma máquina insersora.

6.4.1 Solução Reduzida

A solução reduzida possui quatro operações básicas, que são apresentadas a seguir:

a) *IF - INSERIR_FERRAMENTA* <ferramenta> <cabeça>

Essa operação indica que a pinça a vácuo identificada pelo número <ferramenta>, deve ser inserido na posição <cabeça>, da cabeça insersora. A operação pressupõe que a posição na cabeça insersora onde a ferramenta será inserida está livre.

b) *TF - TROCAR_FERRAMENTA* <ferramenta> <cabeça>

Essa operação indica que a ferramenta localizada na cabeça insersora na posição <cabeça> deve ser substituída pela pinça a vácuo identificada pelo número <ferramenta>. A diferença básica desta operação para operação INSERIR_FERRAMENTA, é que neste caso, a posição na cabeça insersora onde a ferramenta será inserida, já está ocupada por uma ferramenta.

c) IC - INSERIR_COMPONENTE <família> <número do componente>

Essa operação indica que o componente identificado pelo <número do componente> deve ser carregado e inserido em sua posição. Essa é uma operação lógica que considera resolvidas as diversas sub-operações, como determinação da posição de inserção, determinação da posição de carga e determinação da pinça a vácuo utilizada. O parâmetro <família> é utilizado para identificar qual o tipo de ferramenta que será usado para carregar o componente.

d) FIM - FIM_SEQUÊNCIA

Uma vez que a quantidade de operações pode variar de solução para solução, esse parâmetro indica para as diversas funções que processam a solução, que a seqüência chegou ao fim.

Na Figura 30, é apresentada uma solução reduzida e a representação gráfica para a mesma.

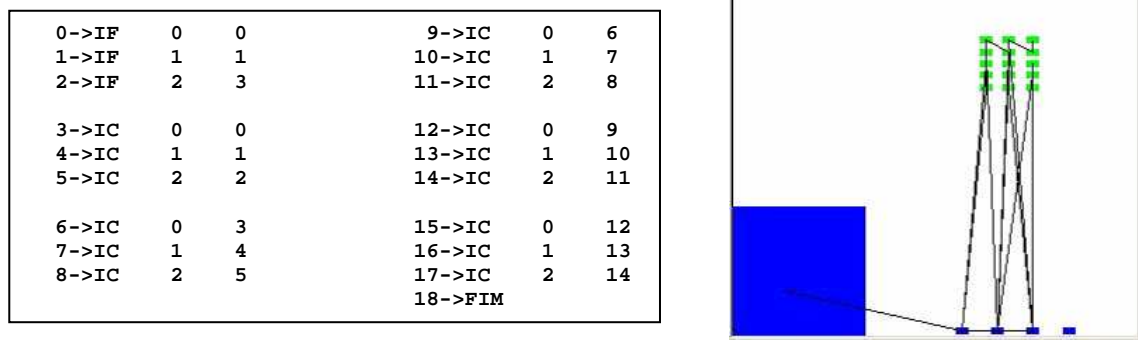


FIGURA 30- SOLUÇÃO REDUZIDA E REPRESENTAÇÃO GRÁFICA CORRESPONDENTE.

No exemplo de solução da Figura 30, a máquina possui cabeça inserora com capacidade para 3 ferramentas, 15 componentes para inserir (numerados de 0 a 14), sendo que os mesmos estão agrupados em três famílias de 5 itens cada. O agrupamento em famílias é realizado conforme a ferramenta utilizada, cujos números de identificação são {0,1,2}. Desconsiderando a indicação de final da seqüência, toda a solução é composta por 18 operações.

A solução reduzida não permite a aplicação direta na máquina, já que vários sub-problemas importantes na operação de inserção "IC" são desconsiderados. Em função disso, a

solução reduzida é usada apenas para gerar a solução inicial pelo algoritmo determinístico. A partir da expansão dela, uma solução completa é gerada. Cabe ressaltar que todas as operações da metaheurística híbrida fazem uso apenas da solução completa.

6.4.2 Solução Completa

A solução completa é uma variação da solução reduzida. Nesta nova solução, a operação `INSERIR_COMPONENTE` foi removida e separada em duas novas operações, com isso, esse formato de solução conta com cinco operações básicas ao invés dos quatro disponíveis na solução reduzida. As novas operações são:

a) `CARREGAR_ITEM <família> <número do componente>`

Essa operação indica que o componente identificado pelo `<número do componente>` deve ser carregado na cabeça insersora em uma ferramenta desocupada e compatível. O parâmetro `<família>` é utilizado para identificar qual o tipo de ferramenta que será usado para carregar o componente, mesma função que ocupava na solução reduzida. O alimentador e a posição onde o mesmo se encontra, são resolvidos pelo avaliador de soluções.

b) `INSERIR_ITEM <família> <número do componente>`

Essa operação indica que o componente identificado pelo `<número do componente>` deve ser inserido em sua posição final na placa de circuito impresso. A operação considera que o componente já foi previamente carregado na cabeça insersora. Para esta operação, o parâmetro `<família>` é utilizado apenas para checagem de consistência da solução, através de uma função específica, não sendo portanto, necessário para o processo de montagem.

A solução completa representa a seqüência de passos completa e ordenada que a máquina irá executar. A Figura 31 ilustra uma solução completa, gerada a partir da expansão de uma solução reduzida. Graficamente, o caso é o mesmo abordado na Figura 30.

SOLUÇÃO COMPLETA			SOLUÇÃO REDUZIDA		
0->IF	0	0	0->IF	0	0
1->IF	1	1	1->IF	1	1
2->IF	2	2	2->IF	2	3
3->CC	0	0	3->IC	0	0
4->CC	1	1	4->IC	1	1
5->CC	2	2	5->IC	2	2
6->IC	0	0			
7->IC	1	1			
8->IC	2	2			
9->CC	0	3	6->IC	0	3
10->CC	1	4	7->IC	1	4
11->CC	2	8	8->IC	2	5
12->IC	0	3			
13->IC	1	4			
14->IC	2	8			
15->CC	0	6	9->IC	0	6
16->CC	1	7	10->IC	1	7
17->CC	2	5	11->IC	2	8
18->IC	0	6			
19->IC	1	7			
20->IC	2	5			
21->CC	0	12	12->IC	0	12
22->CC	1	10	13->IC	1	10
23->CC	2	11	14->IC	2	11
24->IC	0	12			
25->IC	1	10			
26->IC	2	11			
27->CC	0	9	15->IC	0	9
28->CC	1	13	16->IC	1	13
29->CC	2	14	17->IC	2	14
30->IC	0	9			
31->IC	1	13			
32->IC	2	14			
33->FIM			18->FIM		

FIGURA 31– EXPANSÃO DE UMA SOLUÇÃO REDUZIDA PARA SOLUÇÃO COMPLETA.

Na Figura 31, a solução à esquerda, composta por 33 passos é a solução completa. Essa solução foi gerada a partir da expansão de uma solução reduzida apresentada à direita. Uma solução completa é composta por um número variável de linhas e sempre 3 colunas. É possível observar, no caso apresentado, que as operações de carregamento de componentes foram agrupadas, assim como as operações de inserção.

Todo o algoritmo de otimização opera apenas com soluções completas válidas, isto é, não pode haver incoerências que evitem a aplicação da solução em uma máquina real, como por exemplo, carregar um componente sem ferramenta disponível para isso na cabeça inserora, carregar mais itens que a cabeça suporta, inserir componentes não disponíveis na cabeça de inserção, dentre outros. A Figura 32 ilustra o caso de uma solução inválida.

0->IF	0	0	12->IC	0	3	24->IC	0	12
1->IF	1->0	1	13->IC	1	4	25->IC	1	10
2->IF	2	2	14->IC	2	8	26->IC	2	11
3->CC	0	0	15->CC	0	6	27->CC	0	9
4->CC	1	1	16->CC	1	7	28->CC	1	13
5->CC	2	2	17->CC	2	5	29->CC	2	14
6->IC	0	0	18->IC	0	6	30->IC	0	9
7->IC	1	1	19->IC	1	7	31->IC	1	13
8->IC	2	2	20->IC	2	5	32->IC	2	14
9->CC	0	3	21->CC	0	12	39->FIM	-1	-1
10->CC	1	4	22->CC	1	10			
11->CC	2	8	23->CC	2	11			

FIGURA 32– CASO DE UMA SOLUÇÃO COMPLETA INVÁLIDA.

No caso apresentado na Figura 32, na linha um, a operação “*IF 1 1*”, que indica que a ferramenta tipo 1 deve ser inserida na posição 1 da cabeça inserora, foi propositalmente substituída por “*IF 0 1*”, o que causa a inserção de uma ferramenta tipo 0 na posição 1 da cabeça inserora. Com isso, todas as operações que fazem uso da ferramenta tipo 1, passaram a ser inválidas, uma vez que a ferramenta não está disponível. Elas estão localizadas nas linhas : {4, 7, 10, 13, 16, 19, 22, 25, 28, 31} e grifadas em vermelho.

6.5 Avaliação da Solução

A avaliação tem por objetivo quantificar o tempo total que uma dada solução completa precisa para ser executada pela máquina modelada. A avaliação é realizada utilizando a equação (10) e todas as constantes e variáveis associadas a ela.

O avaliador implementado no algoritmo segue a seqüência de passos descrita por uma solução completa e, a cada passo, acumula os tempos de *setup*, carga e inserção envolvidos e a distância percorrida pela cabeça de inserção. Ao final do programa de montagem, a distância é dividida pela velocidade da cabeça de inserção, para determinar o tempo de movimentação. Esse tempo é somado ao tempo acumulado gasto com *setup*, carga e inserção. O resultado é o tempo total de execução do programa de montagem.

Durante a avaliação, todas as operações de *setup* onde não há carga ou inserção posterior de componentes são desprezadas, ou seja, não entram no cálculo que avalia o valor da FO. Isso permite que, durante o processo de otimização, operações desnecessárias de *setup* sejam movidas para o final e não removidas do programa de montagem, e permite também, que as operações necessárias sejam resgatadas do final da lista. Esse modo de funcionamento possibilita que os movimentos para otimização apenas realizem movimentações de operações, nunca remoções ou criações de novas operações.

6.6 Avaliação da Similaridade

Com o objetivo de garantir que boas soluções iniciais (bons valores para a FO e soluções com pouca semelhança) sejam utilizadas quando a metaheurística inicia o algoritmo genético, foi implementada uma medida de similaridade entre as soluções. Essa avaliação foi

criada para minimizar o possível problema de falta de diversidade genética na população inicial. A equação (28) apresenta o modo como o coeficiente é calculado.

$$COEFICIENTE = \frac{Dm}{Np} \times 100 \quad (28)$$

onde,

Dm = Distância média entre operações idênticas;

Np = Número total de passos da solução.

Essa medida é calculada, analisando duas soluções quaisquer, verificando a distância média percentual entre operações idênticas. A Figura 33 ilustra duas soluções diferentes, que são comparadas com o uso da equação proposta (28).

SOLUÇÃO 1			SOLUÇÃO 2		
0->IF	0	0	0->IF	0	0
1->IF	1	1	1->IF	1	1
2->IF	2	2	2->IF	2	2
3->CC	0	0	3->CC	0	0
4->CC	1	1	4->CC	1	1
5->CC	2	2	5->CC	2	2
6->IC	0	0	6->IC	2	2
7->IC	1	1	7->IC	1	1
8->IC	2	2	8->IC	0	0
9->FIM			9->FIM		

FIGURA 33- AVALIAÇÃO DA SIMILARIDADE.

Para o caso apresentado na Figura 33, apenas duas operações estão em posições diferentes da solução completa. A distância entre a operação “IC 0 0” é de duas linhas, assim como ocorre também com a operação “IC 2 2”, com isso, como existem duas linhas diferentes e a distância em linhas até a próxima operação idêntica é 2, $dm = (0 + 0 + 0 + 0 + 0 + 0 + 2 + 0 + 2)/9 = 0,44$, com isso, para o caso ilustrado, o coeficiente de similaridade é 4,9% ($0,44/9$). Quanto mais próximo de 0%, mais semelhante é solução. Essa medida permite quantificar, não apenas quantas linhas mudaram, mas também a distância entre elas, com isso ela se torna pouco sensível a deslocamentos em bloco de partes da solução, causadas pela introdução ou movimentação de operações.

Esse coeficiente é utilizado sempre que a solução deve ser inserida na lista de soluções de elite, porém existe outra solução com valor idêntico. A solução que prevalece é a que apresentar o maior coeficiente, em relação às demais soluções armazenadas.

6.7 Avaliação da Validade de uma Solução

Em função da complexidade na representação da solução, movimentos aleatórios na grande maioria das situações tornam a solução inválida. Essa dificuldade de gerar soluções viáveis motivou a implementação de um validador da solução gerada, cuja função é semelhante ao que softwares para verificação e correção de problemas em sistema de arquivos realizam.

Esse algoritmo procura por inconsistências na solução gerada e tenta resolvê-las, utilizando uma filosofia de mínimo impacto no *setup* de ferramentas. O validador pode ser visto como uma busca local, que visa encontrar uma solução válida. O uso primário do validador ocorre quando a metaheurística está sendo executada no modo genético e realiza operações de mutação. Na geração de vizinhanças da Busca Tabu e nos cruzamentos do algoritmo genético, os movimentos foram projetados visando evitar a geração de soluções inválidas.

O validador é capaz de corrigir problemas na seqüência de inserção e carga de componentes, e também problemas no *setup* de ferramentas. O problema da alocação de alimentadores não é contemplado pelo validador, pois todos os movimentos realizados geram apenas soluções válidas.

Para corrigir problemas na seqüência de inserção, a idéia básica consiste em empurrar operações que não podem ser efetuadas no passo analisado para o final e adiantar operações que poderiam ser realizadas no passo atual. Cada alteração na estrutura da solução gera um novo reinício no processo de busca por erros. O ciclo termina quando a estrutura for completamente varrida sem sofrer nenhuma alteração. A Figura 34 ilustra o primeiro passo do processo de correção de uma solução completa inválida devido a problemas de inserção.

0->IF	1	0
1->CC	0	0
2->IC	0	0
3->CC	0	1
4->IC	0	1
5->TF	1	0
6->CC	1	2
7->IC	1	2
8->CC	1	3
9->IC	1	3
10->TF	2	0
11->CC	2	5
12->IC	2	5
13->CC	2	4
14->IC	2	4
21->FIM		

FIGURA 34– CORREÇÃO DE PROBLEMAS NA SEQÜÊNCIA DE INSERÇÃO.

Na Figura 34, a solução apresentada foi gerada para uso em uma máquina com capacidade para apenas 1 ferramenta na cabeça insersora. Na primeira linha, a ferramenta tipo 1 é inserida na cabeça, porém as duas operações seguintes necessitam da ferramenta tipo 0. Como as mesmas não podem ser executadas naquela posição, são movidas para o final. As posições das linhas 1 e 2 ficaram disponíveis, assim a primeira operação possível de ser realizada é deslocada para este espaço. Como ocorreram movimentações, a análise é novamente iniciada a partir da posição zero. Esse processo terminará quando uma análise completa for executada sem alterar a solução.

Para corrigir problemas no *setup* de ferramentas, operações redundantes ou desnecessárias são movidas para o fim da lista, aproveitando o mecanismo apresentado no item 6.5, onde operações de *setup* no final da lista não são contabilizadas.

Problemas de falta de ferramenta para uma dada operação são corrigidos movendo operações sobressalentes de troca de ferramentas no final da lista para a posição onde a mesma se faz necessária, para que torne assim a solução válida. Em função do mecanismo adianta e atrasa na correção da seqüência de inserção, essa operação é executada com muito pouca freqüência. A Figura 35 ilustra um caso composto por dois passos para correção de uma solução inválida.

PRIMEIRA ANÁLISE	SEGUNDA ANÁLISE	TERCEIRA ANÁLISE
0->IF 1 0	0->IF 1 0	0->TF 1 0
1->CC 0 0	1->CC 1 2	1->CC 1 2
2->IC 0 0	2->IC 1 2	2->IC 1 2
3->CC 0 1	3->CC 1 3	3->CC 1 3
4->IC 0 1	4->IC 1 3	4->TF 2 0
5->TF 1 0	5->TF 1 0	6->CC 2 5
6->CC 1 2	6->TF 2 0	7->IC 2 5
7->IC 1 2	7->CC 2 5	8->CC 2 4
8->CC 1 3	8->IC 2 5	9->IC 2 4
9->IC 1 3	9->CC 2 4	10->TF 0 0
10->TF 2 0	10->IC 2 4	11->CC 0 0
11->CC 2 5	11->TF 0 0	12->IC 0 0
12->IC 2 5	12->CC 0 0	13->CC 0 1
13->CC 2 4	13->IC 0 0	14->IC 0 1
14->IC 2 4	14->CC 0 1	* 15->TF 1 0
15->TF 0 0	15->IC 0 1	21->FIM
21->FIM	21->FIM	

FIGURA 35- CORREÇÃO DE PROBLEMAS DE *SETUP*.

Na Figura 35, o primeiro passo, localizado à esquerda, executa uma correção na seqüência de inserção, movendo as operações que não podem ser executadas nos passos analisados, para o final da lista e adiantando operações possíveis. No segundo passo, localizado no centro da figura e foco principal desta análise, uma operação redundante de *setup* (criada em função da primeira correção) foi movida para o final da lista, assim a mesma é ignorada pelo avaliador de soluções. A direita da Figura 35, é apresentada a solução final. Foi destacada a operação da linha 15, que como foi dito, será desconsiderada pelo avaliador. Apesar de não exibido, nas linhas entre a posição 16 e 20, estão localizadas várias operações de *setup* não utilizadas pelo programa de montagem.

O processo de validação é complexo e por ser iterativo, e em casos extremos pode precisar de $(n^2-n)/2$ iterações para validar uma solução com um erro muito grave, onde n é o número de linhas da solução. Atualmente, nos problemas testados, e do modo como as operações de mutação são realizadas, o algoritmo tem corrigido instâncias inválidas em $n/2$ passos em média.

6.8 Implementação da Metaheurística Híbrida

A implementação da metaheurística híbrida tem como base a Busca Tabu, que é capaz de executar movimentos para otimização de todos os sub-problemas, além de selecionar a população inicial usada para os cruzamentos do Algoritmo Genético. Na arquitetura proposta, o Algoritmo Genético desempenha um papel secundário. Ele não atua na otimização das

operações de *setup*. Seu objetivo principal é diversificar a busca em situações onde a diversificação realizada pela BT não é eficiente.

A implementação da metaheurística híbrida segue exatamente a arquitetura proposta no diagrama das figuras 23 e 25. Considerando que o problema já foi definido nas estruturas de dados, o encadeamento entre as metaheurísticas é executado em 4 passos que são:

- a) Inicialmente, é gerada uma solução inicial reduzida;
- b) A solução reduzida é expandida para uma solução completa;
- c) Aplica-se a Busca Tabu, guardando as melhores e mais diferentes soluções, enquanto o critério de parada não for satisfeito. A implementação permite armazenar entre 100 e 10000 soluções.
- d) Posteriormente, aplica-se o Algoritmo Genético, usando como população inicial, as melhores soluções encontradas e separadas pela Busca Tabu no passo anterior. Cruzamentos e mutações são efetuados, até que o critério de parada tenha sido atingido. Caso a solução encontrada pelo Algoritmo Genético tenha qualidade superior a melhor solução da população inicial, essa passa a ser a nova solução inicial da BT, que é executada novamente (passo c). Esse laço continua indefinidamente, enquanto a execução do AG gerar melhora na solução. Caso a solução não melhore com a execução do AG, o algoritmo encerra a execução e apresenta a melhor solução encontrada.

6.9 Busca Tabu – Visão Geral

Atualmente, utiliza-se o algoritmo clássico de busca (GLOVER; LAGUNA, 1997), usufruindo do refinamento das soluções de elite. Sempre que o critério de aspiração for atingido a busca é reiniciada, utilizando como base as soluções presentes na lista de melhores soluções (que posteriormente irão se tornar a população inicial do AG).

A BT opera nos três sub-problemas apresentados (escalonamento das ferramentas, escalonamento da seqüência de inserção e alocação dos alimentadores). Para cada sub-problema, movimentos específicos em determinadas estruturas de dados que contemplam o mesmo, são executados para geração das vizinhanças.

6.9.1 *Lista Tabu*

A lista Tabu implementada neste trabalho guarda o valor das soluções presentes na mesma e também uma cópia completa da solução que gerou o valor. O valor é guardado visando agilizar o processamento. Durante a verificação de soluções proibidas sempre é feita a análise da solução armazenada e não do valor presente na lista. A lista implementada não guarda os últimos movimentos efetuados.

6.9.2 *Geração de Vizinhanças*

Para cada um dos problemas propostos, a implementação atual contempla dois movimentos, amplamente utilizados na bibliografia pesquisada e considerados eficientes pelos autores. Considerando três sub-problemas, com dois movimentos associados a cada um, no total são executados seis movimentos distintos. Os dois movimentos base para cada sub-problema são listados abaixo:

- a) Troca de pares ou grupos de operações;

Nesse movimento, uma ou mais operações da solução são escolhidas de forma aleatória, e posteriormente são trocadas com todas as outras operações que permitam que a solução continue sendo válida. O número de trocas e o tamanho do bloco movimentado (uma ou mais operações) dependem basicamente do parâmetro que define a quantidade de vizinhos gerados.

- b) Retirada e Inserção.

Nesse movimento, novamente, uma ou mais operações são escolhidas de forma aleatória. A operação escolhida é removida de sua posição e inserida em todas as posições possíveis que mantenham a solução válida. A quantidade de vezes que essa operação é executada dentro de uma mesma solução é novamente dependente do parâmetro que define a quantidade de vizinhos gerados.

O parâmetro que define a quantidade de vizinhos é estático (definido pelo usuário). Entretanto, dependendo do movimento efetuado, a quantidade de vizinhos pode ser inferior ao

valor estipulado quando a instância for pequena ou o número de movimentos inválidos for muito grande.

6.9.3 *Sub-problema da Seqüência de Execução dos Carregamentos.*

Para a devida solução do problema do escalonamento da inserção é necessária ainda a solução do sub-problema da seqüência de carregamento da cabeça insersora. Na Figura 36 é apresentada parte de uma solução onde este sub-problema está destacado. Esse sub-problema aparece, pois a seqüência de carregamento ideal não é, necessariamente, a mesma seqüência de inserção dos componentes. Considerando que o número de pinças na cabeça insersora para o tipo de máquina abordada dificilmente é superior a dez, esse sub-problema sempre é resolvido utilizando força bruta. Todas as combinações possíveis da seqüência de carga são geradas para um determinado movimento de geração de vizinhanças e a melhor solução é escolhida.

0->IF	0	0
1->IF	0	1
2->IF	2	2
3->IF	2	3
4->IF	1	4
5->IF	1	5
6->CC	0	0
7->CC	0	1
8->CC	1	2
9->CC	1	3
10->CC	2	4
11->CC	2	5
12->IC	0	0
13->IC	0	1
14->IC	1	2
15->IC	1	3
16->IC	2	4
17->IC	2	5

FIGURA 36- SUB-PROBLEMA DA SEQÜÊNCIA DE CARREGAMENTO.

Na Figura 36, são apresentadas as 18 primeiras linhas de uma solução completa, para um problema onde a máquina possui seis ferramentas na cabeça insersora. Nas linhas 6 a 11, destacadas na Figura, está localizado o sub-problema da seqüência de carregamento. A resolução deste problema dentro das funções geradoras de vizinhança atua como uma busca local, facilitando a convergência da metaheurística.

6.9.4 *Critério de Aspiração*

Foi utilizado um critério de aspiração baseado em frequência e também por valor percentual em relação a melhor solução encontrada na busca. Na aspiração por frequência, sempre que uma seqüência de N soluções tabu são geradas de forma consecutiva, onde N é

definido pelo usuário, a metaheurística realiza novas buscas em torno da melhor solução encontrada até o momento. Na aspiração por valor percentual em relação a melhor solução, sempre que o valor atual da solução pesquisada ultrapassar um valor especificado em um parâmetro pré-definido. (Passa-se a realizar novas buscas em torno da melhor solução encontrada até o momento). Ao longo dos testes, verificou-se que o critério de aspiração raramente é utilizado.

6.9.5 Critério de Parada

O critério de parada adotado é o mesmo proposto por Glover e Laguna em sua implementação (GLOVER; LAGUNA, 1997). Nesse critério, um parâmetro chamado *nbmax*, determina a quantidade de iterações sem melhora na F.O, que causa a parada no algoritmo.

6.10 Algoritmo Genético – Visão Geral

O Algoritmo Genético tem como função diversificar a procura por novas soluções, quando a Busca Tabu perde a eficiência. Conforme já mencionado, durante a busca, a BT guarda as melhores e mais diferentes soluções encontradas. Essas soluções são utilizadas como população base para o AG. O algoritmo consiste na implementação do modelo clássico de AG descrito por Viana (VIANA, 1998), utilizando as operações de cruzamento e mutação com um critério de parada.

Para simplificar a implementação do algoritmo genético, uma particularidade é o fato que a população inicial de soluções obrigatoriamente possui a mesma seqüência de *setups* de ferramentas. Isso permite codificar o cromossomo no formato da solução completa, apenas especificando que parte da solução representa uma dada característica.

Uma vez que o escalonamento de ferramentas não é alterado, o Algoritmo Genético não atua nos três sub-problemas e sim, em dois. O escalonamento das ferramentas, por questões de complexidade e natureza do problema, será abordado apenas pela BT, restando para o AG o escalonamento da seqüência de inserção e alocação dos alimentadores. Essa abordagem foi adotada, pois um novo escalonamento de ferramentas implica em um “indivíduo” completamente diferente, o que dificulta muito a execução das operações de cruzamento. Se o escalonamento de ferramentas fosse contemplado pelo AG, a complexidade

da implementação seria superior a implementação da BT, que foi projetada para ser o metaheurística dominante na hibridização.

Ao longo dos testes, verificou-se que o escalonamento de ferramentas, após 30% do total de iterações para otimizar um problema, não era mais alterado. As demais iterações apenas atuavam nos sub-problemas do escalonamento da seqüência de inserção e alocação dos alimentadores. Devido à natureza do problema, a simplificação adotada no AG não trouxe prejuízo significativo à qualidade da solução.

6.10.1 Codificação do Cromossomo.

Para a codificação das informações, a solução completa é considerada um cromossomo e cada linha da solução é considerada um gene. A codificação interna do cromossomo foi realizada separando a solução em grupos de genes semelhantes. A Figura 37 apresenta uma solução completa codificada como um cromossomo.

0->IF	1	0	*	
- 1->CC	1	2		
2->IC	1	2		GRUPO 1
3->CC	1	3		
- 4->IC	1	3		
4->TF	2	0	*	
- 6->CC	2	5		
7->IC	2	5		GRUPO 2
----- LINHA DE CRUZAMENTO				
8->CC	2	4		
- 9->IC	2	4		
10->TF	0	0	*	
- 11->CC	0	0		
12->IC	0	0		GRUPO 3
13->CC	0	1		
- 14->IC	0	1		
15->TF	1	0	*	
16->FIM				
ALIMENTADORES: 0 1 2 3 4 5 6 7 8 9				

FIGURA 37- CODIFICAÇÃO DO CROMOSSOMO.

Na Figura 37, uma solução composta por 17 linhas (genes) é apresentada. As linhas marcadas com asterisco representam os genes que tratam do *setup* de ferramentas e não são alterados pelo AG. A linha marcada como linha de cruzamento indica um ponto de corte aleatório onde posteriormente será realizado um cruzamento. Os genes são agrupados em famílias que utilizam um mesmo tipo de ferramentas (diferenciados na figura por cores diferentes). Dentro deste grupo, as operações de cruzamento podem ser efetuadas livremente.

6.10.2 Operação de Cruzamento

As operações de cruzamento são realizadas em blocos entre genes de um mesmo grupo. Para efetuar o cruzamento, uma linha é escolhida de forma aleatória. A partir desta linha, todos os genes de um mesmo grupo antes e após a mesma, são substituídos pelos genes dos pais, gerando dois filhos. No exemplo da figura 37, uma linha de cruzamento foi definida com estando entre as linhas 7 e 8. Para este caso, assumindo que o cromossomo da figura é o pai e o cromossomo a cruzar com este é chamado de doador, um dos filhos (cópia do pai) receberá novas linhas 6 e 7 do cromossomo doador e o outro filho receberá as linhas 9 e 10 do mesmo cromossomo doador. A substituição é feita de forma sequencial dentro do grupo. Visando evitar soluções inválidas, todas as operações duplicadas, que podem ser duas inserções de um mesmo componente ou um alimentador duplicado na solução final, tem a segunda ocorrência encontrada removida, deixando a solução com linhas em branco. A seguir, é gerada uma lista das operações que deveriam estar presentes na solução mas que não foram encontradas. Essa lista então é aleatoriamente disposta nos espaços vazios, criados pelas remoções efetuadas anteriormente. É importante salientar que visando corrigir os filhos gerados, uma pequena parte dos genes alterados assume posições randômicas. A técnica de cruzamento proposta, sempre gera dois filhos a partir de dois pais. Isso mantém o número de indivíduos da população entre as diversas gerações, constante.

6.10.3 Operação de Seleção

De geração para geração, 10% das melhores soluções da geração anterior (pais) são mantidas, substituindo 10% das piores soluções da nova geração. Essa elitização visa evitar o AG diversifique demais e assuma um caráter primariamente randômico.

6.10.4 Operação de Mutação

As operações de mutação visam diversificar a busca. A cada geração, dentre todos os cruzamentos, foi definido - baseado na pesquisa bibliográfica - que 0,5% dos genes (linhas) sofreriam mutação (Durante os testes outros valores percentuais para a mutação foram testados). Nessa implementação, a mutação é efetuada realizando uma troca de pares de operações (duas linhas da solução) de forma aleatória. Como exemplo pode-se considerar a troca das linhas 7 e 9 da Figura 37. Após a mutação, a solução é validada e corrigida pelo algoritmo apresentado no item 6.7. Após a validação, caso o escalonamento de ferramentas não tenha sido alterado, a solução é aceita e inserida na população. Caso o escalonamento

tenha sido alterado, a mutação é desprezada e uma nova mutação é realizada. O processo continua de forma indefinida, enquanto uma mutação válida não for gerada. Ao longo dos testes, foi implementado um gene oculto, indicando que um determinado indivíduo havia sofrido mutação, esse gene era passado adiante a todos os novos indivíduos gerados a partir de um pai com mutação. Com isso era possível determinar se o filho analisado foi gerado a partir de um pai normal ou a partir de um pai que sofreu mutação. Durante o desenvolvimento de todo trabalho, em apenas um caso a solução final estava com o gene ativo, indicando que a mesma havia sido gerada através de uma mutação. A mutação tem a função principal no AG de diversificar a busca, entretanto, pelo exposto anteriormente, a hibridização fez com que o papel deste operador tivesse pouca importância na busca da solução final.

6.10.5 Critério de Parada

O critério de parada adotado é baseado em duas regras. A primeira delas, define o número máximo de gerações, independente de melhora ou não da solução. A segunda regra define o número máximo de gerações consecutivas sem melhora na solução. Baseado em testes, empiricamente foi determinado o número máximo de gerações como sendo 1000 e o número de gerações consecutivas sem melhora na solução como sendo 100 gerações. Independente de ordem, a primeira regra que for verdadeira causa a parada na execução do AG.

7 EXPERIMENTOS E VALIDAÇÃO DO MODELO

Com o objetivo de validar a implementação do modelo proposto, que engloba as estruturas de dados, modelos, metaheurísticas e, conseqüentemente, a qualidade da solução, nesse capítulo o modelo será testado com três instâncias diferentes do problema proposto. Os problemas testes utilizados podem ser classificados em dois grandes grupos:

a) Instâncias teóricas de teste disponíveis na literatura, onde dados, geralmente fictícios, são utilizados por diversos autores para confrontar a qualidade da solução de um algoritmo;

b) Instâncias abordando um caso real, onde uma máquina insersora comercial será modelada e a qualidade da solução será confrontada com a solução comercial gerada pelo software do fabricante da máquina.

7.1 Validação do Modelo

Visando testar a eficiência da técnica computacional apresentada neste trabalho, algumas instâncias de teste disponíveis e já analisadas por outros autores e por outras técnicas, foram adaptadas para se adequarem ao modelo proposto e testadas na técnica implementada.

Em função do problema estudado ser composto por 3 sub-problemas, a eficiência do algoritmo foi testada para cada um deles separadamente, quando isso foi possível. A separação facilita o confronto dos resultados de outros autores, bem como evita alterações profundas na instância de teste, o que eventualmente pode descaracterizar a mesma.

A análise individual de cada sub-problema foi dividida da seguinte forma: a determinação do escalonamento do conjunto ideal de ferramentas foi tratada como um Problema de Família de Partes associado a um algoritmo determinístico para a geração da solução inicial. Posteriormente, durante a otimização, o problema de Família de Partes é considerado resolvido e O problema do *setup* de ferramentas passa a ser tratado como um Problema de Escalonamento; o Problema da Alocação de itens no alimentador de componentes foi abordado com um PCV; a ordem de execução das inserções também foi tratada como um PCV.

7.1.1 Ordem de Execução das Inserções

Para avaliação deste problema especificamente, a função objetivo apresentada na equação (10) sofreu algumas modificações. Todos os parâmetros de tempo relativos a *setup* assumiram o valor zero e, alterando a codificação do avaliador da função objetivo, todas as operações de carga e movimentação, relacionadas com a posição dos componentes nos alimentadores, foram anuladas. Com isso, pode-se dizer que criou-se um modelo onde a máquina possui todas as ferramentas necessárias, sempre disponíveis e prontas para uso nas cabeças de inserção (ausência de *setups*) e que a cabeça de inserção tem todos os componentes necessários carregados de forma simultânea, o que evita as operações e movimentações de carregamento. Devido às alterações, o problema fica simplificado à otimização da trajetória da cabeça de inserção, de modo a reduzir a distância total percorrida. Essa simplificação permite analisar o problema como um Problema do Caixeiro Viajante, que possui um grande número de instâncias disponíveis para teste.

Nesta abordagem foram realizados três experimentos, classificados de forma a se encaixarem na realidade do problema estudado: problema de pequena escala (até 60 componentes), problema de média escala (até 500 componentes) e problema de grande escala (mais de 2000 componentes). Cabe salientar que o percentual de placas de circuito impresso com mais de 2000 componentes inseridos automaticamente é extremamente baixo. Em média, baseado na experiência profissional do autor, as placas possuem entre 100 e 300 componentes inseridos de forma automatizada.

Para realizar estes testes foram escolhidas 3 instâncias disponíveis na literatura e amplamente testadas. Essas instâncias são parte da biblioteca TSPLIB95, mantidas pela universidade de Heidelberg, localizada na Alemanha. Estas instâncias podem ser acessadas

livremente no endereço apresentado no anexo A. As instâncias selecionadas foram: eil51 (contendo 51 pontos de visitação), d493 (esta contendo 493 pontos visitação) e finalmente, a d2103 (com 2103 pontos de visitação). Anexado às informações que definem as instâncias sempre é incluído o melhor resultado conhecido até o momento para um determinado problema. Quando este resultado é considerado o ótimo global, essa informação também está presente. A instituição não informa quem, nem quando, nem qual metaheurística foi utilizada para gerar a solução apresentada. A informação é anônima, porém respaldada pela universidade de Heidelberg e amplamente utilizada por outros autores, baseado nisso, essa pesquisa considerou as informações confiáveis. As três instâncias são apresentadas em sua forma integral no anexo A.

Na Tabela 4 são apresentados os resultados obtidos com a implementação proposta no trabalho.

TABELA 4 – ANÁLISE COMPARATIVA COM INSTÂNCIAS CONHECIDAS APLICADA AO PROBLEMA DA SEQÜÊNCIA DE INSERÇÃO.

INSTÂNCIA TESTADA	MELHOR RESULTADO DA LITERATURA	MELHOR RESULTADO DO ALGORITMO PROPOSTO	DIFERENÇA PERCENTUAL
eil51	426	426,25	+0,059%
d493	35002	35002,36	+0,001%
d2103	80450	80448,19	-0,0022%

Pode-se observar na Tabela 4 que os valores são idênticos, caso a parte fracionária seja desconsiderada. Por definição da entidade que administra as instâncias, as soluções sempre são fornecidas com resultados inteiros. Segundo os dados de teste, as soluções das instâncias eil51 (426) e d493 (35002) são consideradas ótimos globais. Caso a definição adotada pelos mantenedores das instâncias seja usada, é possível dizer que o algoritmo é capaz de atingir os mesmos resultados de outros trabalhos, para instâncias de até médio porte, e até mesmo apresentar desempenho superior para as instâncias aqui consideradas de grande porte.

Cabe salientar que na Tabela 4 foi apresentada a melhor resposta de 200 execuções para cada uma das instâncias testadas. Para a instância Eil51, em toda série de experimentos, 121 das 200 execuções apresentaram o valor 426,25. Em termos percentuais, 60,5% dos experimentos atingiram o valor apresentado na tabela. Na Figura 38 é apresentado o histograma das soluções encontradas.

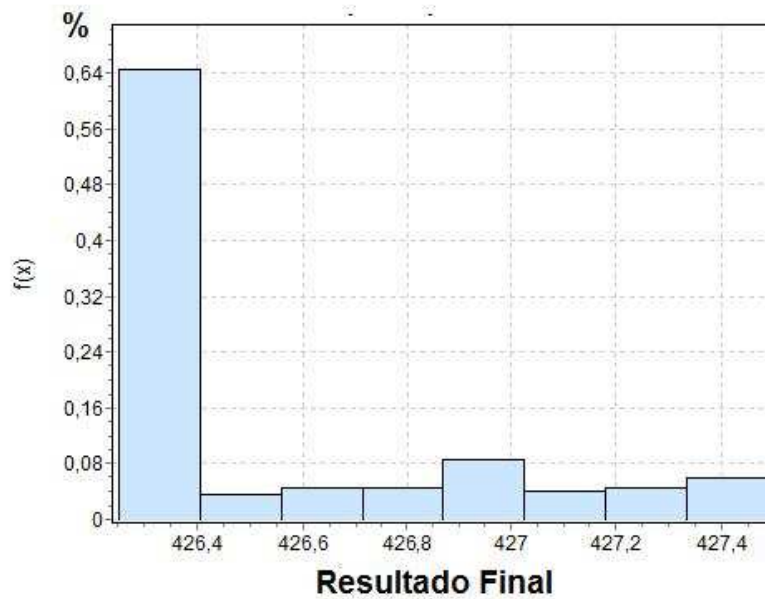


FIGURA 38– HISTOGRAMA DAS SOLUÇÕES ENCONTRADAS PARA UM TOTAL DE 200 EXECUÇÕES DA METAHEURÍSTICA PARA A INSTÂNCIA EIL51 .

Na Figura 38 pode-se observar a baixa variabilidade dos resultados, bem como o desempenho da técnica. O pior valor encontrado foi de 427,49, apenas 0,3% maior que a melhor solução encontrada. Para cada um dos 200 experimentos, a solução inicial era gerada de forma randômica, fator que contribuiu muito para o resultado apresentado no histograma.

Realizando a mesma análise para a instância d493, observou-se que 83 dos 200 experimentos atingiram o valor apresentado na tabela. Esse valor corresponde a 41,5% do total. Na Figura 39 é apresentado o histograma das soluções encontradas.

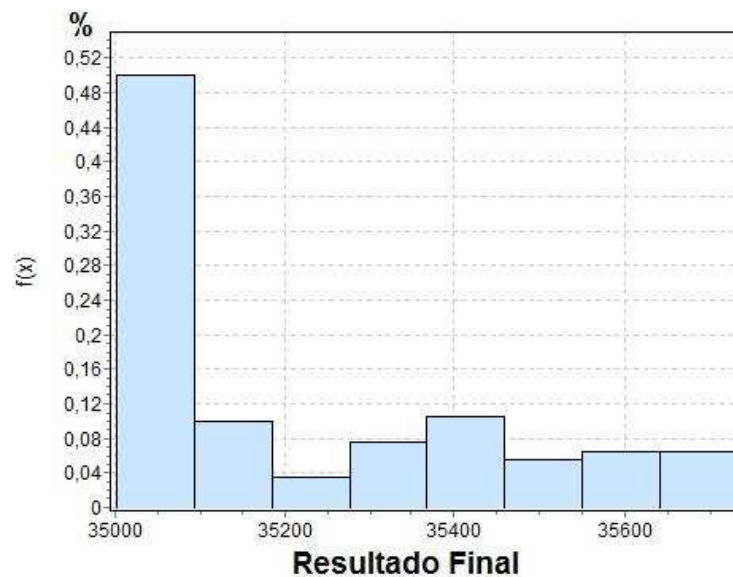


FIGURA 39– HISTOGRAMA DAS SOLUÇÕES ENCONTRADAS PARA UM TOTAL DE 200 EXECUÇÕES DA METAHEURÍSTICA PARA A INSTÂNCIA D493.

Na Figura 39, pode-se observar que, além do percentual de experimentos que encontraram o melhor valor do teste ter reduzido em relação à instância de pequeno porte, observa-se que o pior valor (35733,10) agora é 2,1% maior que a melhor solução encontrada. Isso é esperado, uma vez que a instância d493 possui quase 10 vezes mais elementos que a instância Eil51.

Finalmente, analisando os resultados apresentados pela técnica de otimização para a instância d2103, pode-se observar pelo histograma apresentado na Figura 40, que 12 dos 200 experimentos feitos chegaram ao valor apresentado, o que corresponde a 6% do total de experimentos.

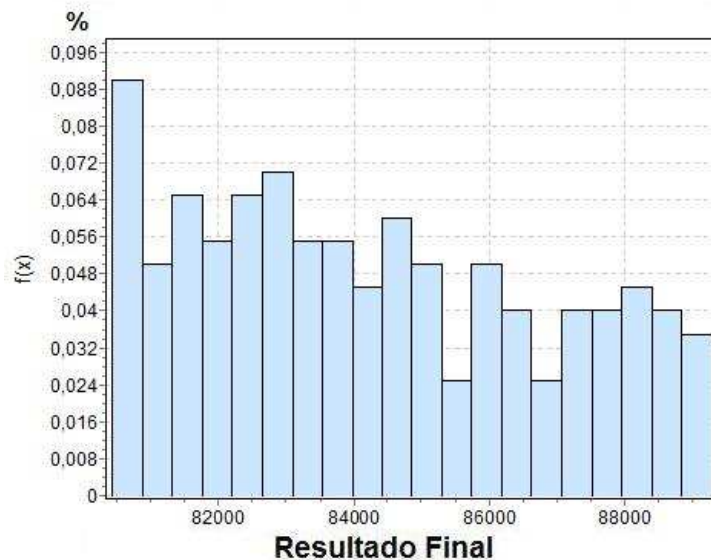


FIGURA 40– HISTOGRAMA DAS SOLUÇÕES ENCONTRADAS PARA UM TOTAL DE 200 EXECUÇÕES DA METAHEURÍSTICA PARA A INSTÂNCIA D2103.

Na Figura 40 pode-se observar uma maior variabilidade nos resultados. A diferença entre o melhor e o pior valor (89274,37) passou para 11%. Basicamente, isso ocorre devido ao modo como a solução inicial é gerada. Nesse teste, novamente visando testar a eficácia da busca, a solução inicial foi gerada de forma aleatória, o que efetivamente dificulta a busca por boas soluções.

7.1.2 Problema da Alocação de Componentes no Alimentador de Componentes

Para validar o problema da alocação de componentes nos alimentadores, a mesma metodologia do teste anterior foi adotada. Entretanto o enfoque da aplicação da instância foi diferente. No problema anterior, cada item da instância representava uma posição de inserção de componente na PCB. No problema atual, cada item da instância representa a posição física

de um alimentador. Especificamente para este teste, todos os tempos gastos com *setup* e movimentações de inserção de componentes foram anulados. O avaliador da FO foi alterado, de modo a considerar apenas os tempos gastos com movimentação, visando a carga de componentes. Após a carga do item ter sido efetuada, todos os movimentos gastos com inserção não eram contabilizados. Com isso, novamente é possível modelar o problema como um PCV, onde o resultado final para a ordem dos itens no alimentador de componentes é o caminho percorrido e não a distância total percorrida. Com este teste é possível verificar se os movimentos realizados pela metaheurística na busca da solução são efetivos. Apesar dos movimentos codificados serem os mesmos do teste anterior, eles são aplicados a partes diferentes (matrizes) da solução em análise, o que justifica o teste. Cabe salientar que até o presente momento não existem máquinas com 493 ou mesmo 2103 alimentadores. Essas instâncias são utilizadas apenas para validar a eficiência desta parte do algoritmo em particular.

Neste teste, foram encontrados os mesmos resultados apresentados na Tabela 4. Entretanto os movimentos foram realizados de modo a alterar a alocação dos itens no alimentador de componentes que, conforme apresentado no seção 6.2, que descreve a implementação, utiliza matrizes de dados distintas da matriz de dados utilizada para a seqüência de inserção (objetivo do teste anterior). A solução idêntica mostra que os movimentos efetuados pela metaheurística híbrida são efetivos para solução deste problema, como também foram no problema anterior. Novamente neste teste foram efetuados 200 experimentos para cada instância testada. A dispersão das soluções encontradas foi muito semelhante às apresentadas nos histogramas do teste anterior, o que torna a sua reapresentação redundante, considerando que os movimentos realizados são os mesmos de forma geral, porém aplicados a matrizes diferentes.

7.1.3 Escalonamento das Ferramentas

A determinação do escalonamento de ferramentas é o único problema que não pode ser analisado separadamente. Isso ocorre pois todos os *setups* de ferramentas são dependentes de forma direta dos componentes inseridos em momentos posteriores ao *setup*. Em função disso, esse problema é sempre dependente da ordem de execução das inserções. Essa característica torna complicada a comparação direta com instâncias públicas para o problema sem pesadas alterações no algoritmo.

A metodologia adotada para o teste assumiu, baseada nos testes efetuados na seção 7.1.1, que a ordem de execução das inserções era determinada de forma eficiente pela técnica proposta. Baseado nisso, a função objetivo foi alterada de forma que somente os tempos relativos a *setup* de ferramentas fossem computados. Para padronizar os testes, as mesmas instâncias do item 7.1.1 foram utilizadas. Em todos os experimentos, o total de componentes inseridos foi dividido em três grupos, sendo que cada grupo necessita de um tipo diferente de ferramenta para sua inserção. Essa é uma condição realista, baseada na máquina utilizada para o teste real apresentado na seção 7.4. Nessa seção, a máquina apresentada possui seis ferramentas distintas, porém três delas são capazes de transportar 80% dos tipos de componentes que ela é capaz de inserir.

Na Tabela 5 são apresentados os resultados dos testes efetuados. Nela é realizada uma análise comparativa entre a solução inicial do *setup* gerada pelo algoritmo determinístico e o tempo final gasto com *setup*, após a execução da técnica proposta que otimiza o *setup* de forma conjunta com a seqüência de inserção.

TABELA 5 – ANÁLISE COMPARATIVA ENTRE AS SOLUÇÕES INICIAIS E FINAIS.

INSTÂNCIA	CABEÇAS INSERSORAS DISPONÍVEIS P/ USO NA MÁQUINA	SOLUÇÃO INICIAL		SOLUÇÃO FINAL		REDUÇÃO PERCENTUAL DA SOLUÇÃO FINAL RELATIVO A SOLUÇÃO INICIAL
		TEMPO (S)	% DE USO DAS CABEÇAS	TEMPO(S)	% DE USO DAS CABEÇAS	
Ei151	1	5	100%	5	100% (1/1)	0 %
Ei151	4	12	100%	3	75% (3/4)	-76,9 %
Ei151	10	20	100%	3	33% (3/10)	-85,0 %
d493	1	5	100%	5	100% (1/1)	0 %
d493	4	15	100%	3	75% (3/4)	-80,0 %
d493	10	40	100%	3	33% (3/10)	-92,5 %
d2103	1	5	100%	5	100% (1/1)	0 %
d2103	4	15	100%	3	75% (3/4)	-80,0 %
d2103	10	45	100%	3	33% (3/10)	-93,3 %

Na Tabela 5 é possível verificar que, independente da instância utilizada, a solução final encontrada para cada um dos três casos propostos (1, 4 e 10 pinças na cabeça insersora) é a mesma, quando a comparação é realizada entre instâncias. Os resultados encontrados são esperados por dois fatores:

a) O primeiro deles, deve-se ao fato de que para todas as instâncias, são usadas três ferramentas para inserção de todos os componentes;

b) No segundo, a nova função objetivo teve todas as variáveis não relacionadas ao *setup* anuladas.

Com isso, independente do número de componentes inseridos, as inserções são agrupadas pela metaheurística em três grandes grupos, onde, no pior dos casos, três *setups* são necessários. Nos problemas da Tabela 5, estipulou-se o tempo para inserção ou remoção de ferramenta, como sendo 1 segundo e o tempo para troca de ferramenta igual a 2 segundos.

Observa-se, nos casos onde a máquina possui apenas uma cabeça de inserção, que as três operações de *setup* realizadas foram, respectivamente: inserir uma ferramenta, seguido de duas trocas posteriores de ferramentas. Nos demais problemas, a metaheurística optou por deixar ferramentas na cabeça de inserção ociosas, de modo a minimizar o *setup*. Em todos os casos, as três operações de *setup* foram a inserção de todas as ferramentas necessárias no início do programa de montagem.

Outra observação de muita importância, que demonstra a eficiência da técnica proposta, é baseada na solução inicial. O *setup* inicial é gerado por um algoritmo determinístico, onde a regra básica é manter as ferramentas na cabeça de inserção o menos ociosas possível, maximizando o número de itens transportados. Para os casos apresentados, devido à simplificação na FO, em máquinas com grande número de ferramentas na cabeça insersora, a solução inicial gerada, apesar de válida, é de baixa qualidade. Percebe-se com isso que, mesmo partindo de uma solução desfavorável, a metaheurística é capaz de alterar todo o escalonamento de inserções, de modo a unificar os componentes em três grupos contínuos para reduzir a quantidade de *setups*. A técnica converge, mesmo com soluções iniciais altamente desfavoráveis, como é o caso da instância d2103, aplicada a uma máquina com 10 ferramentas na cabeças de inserção, onde o tempo de *setup* baixou de 45 segundos (início de componentes desagrupados no programa de montagem) para 3 segundos.

7.2 Experimentos

Com o objetivo de testar o comportamento dinâmico do modelo e a contribuição de cada sub-problema na composição da solução final, alguns experimentos utilizando testes reais foram propostos. Posteriormente, visando analisar os ganhos relativos a tempo e custo em um caso real, o resultado de um programa de montagem, gerado pela metaheurística, foi confrontado com a solução gerada pelo software da máquina em teste.

7.2.1 *Análise de Significância dos Sub-problemas*

Para realização deste experimento foram utilizados três problemas teste. Todos eles correspondem a itens montados na fábrica onde uma insersora foi disponibilizada para os testes.

Em cada problema teste, para avaliar o peso individual dos sub-problemas, a seguinte metodologia foi adotada:

a) Inicialmente, a metaheurística foi executada sem restrição alguma aos três problemas. O melhor resultado de 200 experimentos (para cada sub-problema) foi considerado como base para comparação com resultados posteriores;

b) Para cada problema teste, visando a análise da significância de cada sub-problema, a solução inicial adotada foi a mesma;

c) Para cada um dos sub-problemas, os movimentos visando otimização dos mesmos, foram desativados de forma independente dos demais movimentos. Assim, nos testes que analisaram a contribuição da otimização do *setup*, os movimentos que buscavam melhores *setups*, deixaram de ser feitos. O mesmo ocorreu nos testes que analisaram a significância da otimização da seqüência de inserção e alocação de alimentadores. Os movimentos foram desativados na Busca Tabu e Algoritmo Genético (com exceção dos movimentos para otimizar *setup*, que não são contemplados pelo Algoritmo Genético e foram desativados apenas na BT).

Na Tabela 6 são apresentadas as características principais dos três problemas teste. Os parâmetros da máquina que definem em parte o comportamento da FO são, com exceção de um parâmetro, iguais aos da máquina utilizada na seção 7.4. A característica principal deste modelo é possuir uma cabeça insersora com capacidade para até quatro pinças para inserção de componentes, enquanto que a máquina da seção 7.4 possui uma cabeça insersora com capacidade para até duas pinças. Optou-se modelar a máquina com quatro pinças de inserção nos testes apresentados, pois essa condição permite analisar de forma efetiva a capacidade da metaheurística híbrida em otimizar a solução. Ao longo dos testes constatou-se que, em função das limitações mecânicas impostas pelas máquinas e geométricas impostas pelo problema, a otimização de problemas em máquinas com uma ou duas pinças na cabeça

insersora é consideravelmente mais simples de ser feita, mascarando de forma severa o problema do *setup*, o que inviabilizaria a análise de significância proposta.

TABELA 6 – CARACTERÍSTICAS DOS PROBLEMAS TESTE DOS EXPERIMENTOS.

NOME DO PROBLEMA	TOTAL DE ITENS INSERIDOS	TOTAL DE FERRAMENTAS UTILIZADAS	NÚMERO DE ITENS DIFERENTES
RECEPTOR_23	23	3	7
ALARME_57	57	3	12
CENTRAL_315	315	3	22

Na Tabela 6, os três problemas apresentados foram escolhidos baseados nos critérios de quantidade de itens inseridos, diversidade de itens inseridos e diversidade no uso das ferramentas. O problema CENTRAL_315 é o maior problema real em uso pela empresa que forneceu os modelos e alocou horas da máquina para realização dos experimentos do item 7.4. A primeira coluna da tabela apresenta o nome do problema, a segunda, a quantidade de componentes inseridos, a terceira, a quantidade de ferramentas distintas utilizadas para inserção de todos os componentes e, finalmente, a quarta coluna informa a quantidade de componentes diferentes, o que se traduz na quantidade de alimentadores utilizados (a máquina apresentada tem capacidade para até 100 alimentadores).

Na Tabela 7 são apresentados os resultados encontrados para cada problema teste, desativando na metaheurística a otimização de cada um dos sub-problemas separadamente.

TABELA 7 – SOLUÇÕES GERADAS FORÇANDO LIMITAÇÕES NA METAHEURÍSTICA.

PROBLEMA	SOLUÇÃO INICIAL	MELHOR SOLUÇÃO FINAL	SOLUÇÃO FINAL SEM RESTRIÇÕES APLICADAS A BUSCA		
			SETUP	INSERÇÃO	ALIMENTADORES
RECEPTOR_23	29,59	12,67	28,03	13,96	12,98
ALARME_57	46,10	27,93	42,05	28,405	28,86
CENTRAL_315	148,58	133,73	136,49	136,29	135,01

* OS DADOS APRESENTADOS NA TABELA REPRESENTAM SEGUNDOS.

Na Tabela 7, a primeira coluna representa o nome do problema teste, a segunda, apresenta a solução inicial gerada pelo algoritmo determinístico. Esse algoritmo não possui nenhum tipo de randomização, logo, para um dado problema, a solução inicial é sempre a mesma. A terceira coluna apresenta a melhor solução encontrada pela metaheurística proposta sem limitar os movimentos de busca. As colunas quatro, cinco e seis, apresentam a solução encontrada pela metaheurística, aplicando limitações aos movimentos da busca. A limitação torna o peso da otimização do sub-problema analisado nulo (sem otimização), enquanto os demais sub-problemas permanecem inalterados. As limitações inseridas são citadas a seguir:

a) Na quarta coluna, intitulada “*SETUP*”, não foram aplicadas limitações aos movimentos que buscavam melhorar o escalonamento de ferramentas. Com isso o escalonamento final de ferramentas é o mesmo da solução inicial apresentada na segunda coluna. Neste caso, somente a seqüência de inserção de componentes e a alocação dos alimentadores foi otimizada.

b) Na quinta coluna, intitulada “*INSERÇÃO*”, não foram aplicadas limitações a todos os movimentos que buscavam melhorar o escalonamento da seqüência de inserção. Neste caso, apenas o *setup* de ferramentas e a alocação dos alimentadores foi otimizada.

c) Na sexta e última coluna, intitulada “*ALIMENTADORES*”, todos os movimentos que buscavam otimizar a alocação de componentes nos alimentadores foram desativados. Neste caso a alocação de alimentadores gerada pela solução inicial é a mesma adotada na solução final. Neste problema foram otimizados o escalonamento de ferramentas e o escalonamento da seqüência de inserção de componentes.

Pelos resultados, pode-se observar para os problemas onde há uma quantidade relativamente pequena de componentes, no caso *RECEPTOR_23* e *ALARME_57*, que a regra básica utilizada para gerar a solução inicial (minimizar a ociosidade das ferramentas na cabeça de inserção, maximizando o número de componentes transportados) é ineficiente, fazendo com que a otimização do *setup* seja o sub-problema predominante na busca por uma boa solução. Paralelamente, os sub-problemas da seqüência de inserção e alocação de alimentadores, tem significância semelhante na composição da solução, sendo que as características do problema teste definem qual dos dois é predominante. No problema *RECEPTOR_23*, a seqüência de inserção predomina sobre alocação dos componentes no alimentador, o que é demonstrado pela resposta de pior qualidade quando esse sub-problema é ignorado. No problema *ALARME_57*, alocação dos componentes no alimentador predomina sobre a seqüência de inserção.

Especificamente no problema *ALARME_315*, observa-se que não há dominância de nenhum sub-problema na composição da resposta final. Este caso teste permite visualizar, de forma clara, como os sub-problemas estão correlacionados. Caso apenas um deles seja ignorado, a nova solução ótima local é, no mínimo 1% maior que no caso completo.

A Tabela 7, apesar de permitir a visualização da importância de cada sub-problema na composição de uma boa solução final, não permite analisar de forma quantitativa, o peso individual de cada sub-problema. Para que essa análise fosse possível, o avaliador de soluções da metaheurística híbrida foi alterado, de modo a quantificar de forma separada os tempos gastos com *setup* e movimentações relacionadas à carga e inserção. A divisão foi feita deste modo, devido à forte dependência natural existente entre os problemas da seqüência de inserção e alocação de componentes (analisando os problemas como um PCV, a seqüência de inserção equivale a ordem de visitação das cidades, enquanto o problema de alocação define a posição geométrica de algumas cidades envolvidas no percurso).

Na Figura 41 é apresentado um gráfico onde o peso de cada sub-problema foi quantificado de forma percentual em relação à solução final encontrada, aplicado a 4 casos: a metaheurística híbrida trabalhando sem limitações nos movimentos de busca, com limitações na otimização de *setup*, com limitações na otimização da seqüência de inserção de componentes e finalmente, com limitações na otimização da alocação de alimentadores. Os casos e limitações são os mesmos que foram utilizados na Tabela 7, porém agora com um maior nível de detalhamento.

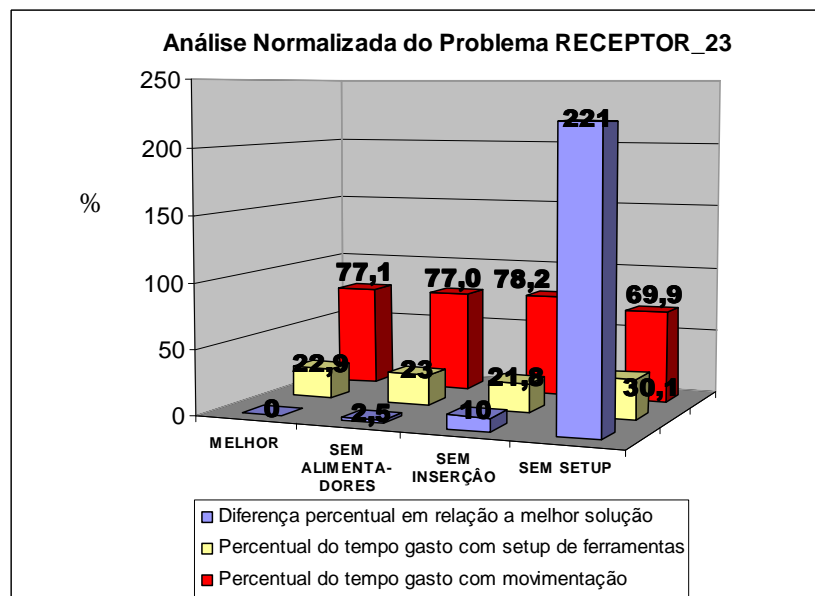


FIGURA 41– PERCENTUAL DE TEMPO DA SOLUÇÃO FINAL GASTO COM CADA SUB-PROBLEMA PARA O CASO TESTE RECEPTOR_23.

Na Figura 41 os gráfico de barra estão organizados da seguinte forma: a linha composta pelas colunas ao fundo (em vermelho), com valores entre 69% e 77%, representam o percentual de tempo gasto com a movimentação da cabeça de inserção, relativos ao problema em questão. A linha composta pelas colunas do meio (em amarelo), representa o

percentual de tempo total(novamente relativo ao problema em questão) gasto com *setup*. A linha descrita pelas colunas frontais (em azul) representa o quão pior foi a solução, em relação a melhor solução (com isso, para o caso teste descrito como “MELHOR”, a coluna sempre será nula, uma vez que a comparação é realizada tendo ele mesmo como referência). O problema teste rotulado como “MELHOR” representa o caso onde a metaheurística foi executada sem limitações ao movimentos de busca. O rótulo “SEM ALIMENTADORES”, representa o caso onde o problema da alocação de componentes no alimentador não foi otimizado. O rótulo “SEM INSERÇÃO” representa o caso onde o problema da seqüência de inserção não foi otimizado e finalmente, o rótulo “SEM SETUP” representa o caso onde o problema do *setup* de ferramentas não foi otimizado.

Para o problema-teste RECEPTOR_23, percebe-se claramente a importância do *setup* na busca de uma solução de qualidade. Percebe-se também, que na composição da solução final, o tempo gasto com movimentação sempre representou a maior parcela da solução final, o que justificou o aumento em 2,5% do tempo total de montagem, em relação a melhor solução para o caso onde a otimização da alocação de alimentadores estava desativada e 10% para o caso onde a seqüência de inserção estava desativada. Observou-se também que, não somente para esse caso, mas para vários outros casos de pequeno porte testados no decorrer da dissertação, a otimização da seqüência de inserção foi mais decisiva do que a alocação de componentes nos alimentadores. Em função do problema ser NP-Difícil, não é possível afirmar com certeza o motivo, mas aparentemente esse comportamento ocorre devido à baixa diversidade de componentes desse tipo de problema. Isso implica em uma quantidade menor de alimentadores utilizados, o que acaba agrupando em uma pequena área as movimentações de carga de componentes. Esse agrupamento faz com que a posição de cada item não tenha um papel tão importante na otimização da trajetória. Extrapolando, é possível imaginar um caso hipotético onde todos os componentes estão localizados em um mesmo ponto, neste caso, a posição de cada item não afeta de maneira alguma a solução final.

No caso onde a otimização do *setup* foi desativada, o *setup* final de ferramentas é idêntico ao *setup* inicial gerado pelo algoritmo determinístico. Como os problemas são correlacionados, um *setup* inicial desfavorável implica em limitações na otimização da seqüência de inserção. A estratégia adotada pelo algoritmo determinístico, que implica em manter as ferramentas na cabeça de inserção o menos ociosas possível, é desfavorável para este tipo de caso. Independente desta constatação, em função do tamanho do problema-teste, a

metaheurística proposta não tem dificuldades em convergir para uma boa solução, uma vez que a solução inicial apenas precisa ser válida. Pode-se observar que o *setup* desfavorável implicou também em um aumento significativo da movimentação. Analisando a resposta gerada, observou-se que o maior número de trocas de ferramentas, aliado à distância significativa do alimentador de ferramentas em relação ao ponto de montagem, implicou no aumento do tempo gasto com a movimentação proporcionalmente ao problema analisado.

Na Figura 42 é apresentado o mesmo tipo de gráfico utilizado na Figura 41, porém aplicado ao problema teste ALARME_57.

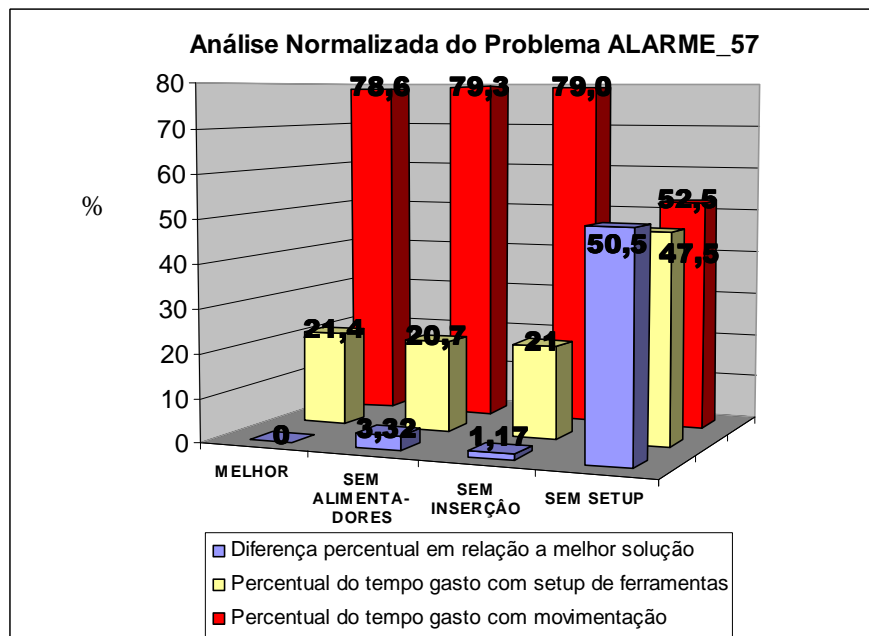


FIGURA 42– PERCENTUAL DE TEMPO DA SOLUÇÃO FINAL GASTO COM CADA SUB-PROBLEMA PARA O CASO TESTE ALARME_57.

Para o gráfico apresentado na Figura 42, a organização dos dados e colunas seguiu o formato apresentado na Figura 41.

No gráfico da Figura 42, é possível observar um comportamento diferente do problema ALARME_57 em relação ao problema anterior, RECEPTOR_23. Nesse novo caso é possível observar que a otimização da alocação dos alimentadores é mais significativa que a seqüência de inserção. Isso ocorre, principalmente, devido a maior quantidade de alimentadores utilizados e quantidade de itens inseridos. Devido a distância entre a área média de inserção dos itens e os alimentadores ser significativa, o grande número de movimentações entre a área de inserção e o alimentador de componentes, impacta na solução final. No caso onde a otimização do *setup* de ferramentas foi desativada, houve um aumento significativo no

tempo gasto com *setup*, mas não com a movimentação (quando comparada a melhor solução). Os 52,5% de tempo gasto com a movimentação, mudam para 78,6% quando o valor de referência do percentual é alterado para o tempo do melhor caso. Com isso, pode-se observar claramente que o sub-problema dominante para o caso ALARME_57, passou a ser a movimentação, que independente do caso, se manteve entre 78% e 80%. Porém, novamente devido à dependência entre os problemas, o *setup* precisa ser otimizado para uma solução de boa qualidade. Observou-se, nesse caso, que a estratégia de maximizar o uso de ferramentas na cabeça de inserção adotada pelo algoritmo determinístico foi mantida pela metaheurística, porém a seqüência inicial de ferramentas foi alterada de modo a minimizar o número de *setups* de ferramentas ao longo do programa de montagem.

Para o caso teste final, na Figura 43 é apresentado o mesmo tipo de gráfico utilizado na Figura 41, porém aplicado ao problema teste CENTRAL_315.

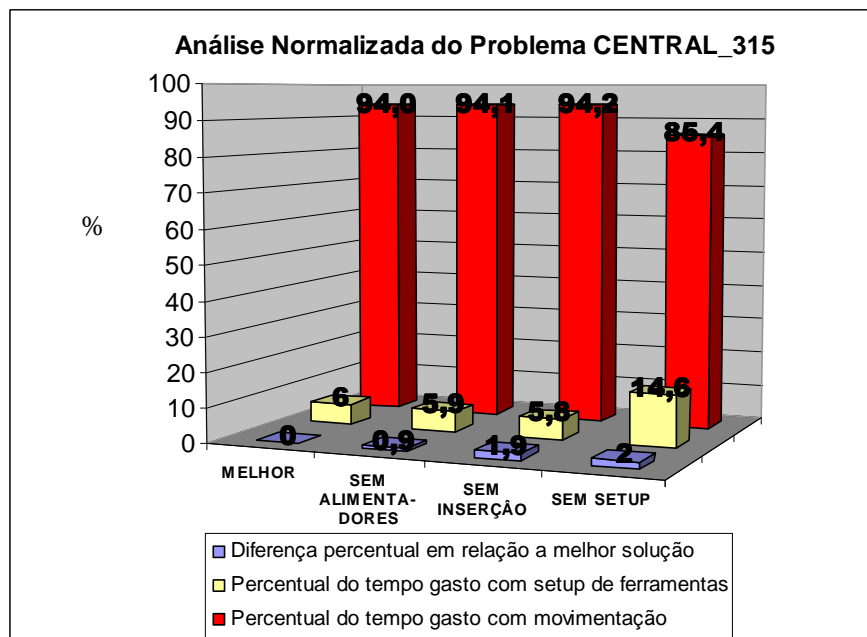


FIGURA 43– PERCENTUAL DE TEMPO DA SOLUÇÃO FINAL GASTO COM CADA SUB-PROBLEMA PARA O CASO TESTE CENTRAL_315.

Assim como na análise anterior, para o gráfico apresentado na Figura 43, a organização dos dados e colunas seguiu o formato apresentado na Figura 41.

Para os resultados apresentados na Figura 43 aplicados ao problema CENTRAL_315, pode-se observar um comportamento completamente diferente dos dois casos apresentados anteriormente. Nesta situação, devido à complexidade do caso analisado, a diferença entre a pior e a melhor solução foi de apenas 2%. Analisando os dados, também não é possível

afirmar se houve um sub-problema dominante. Pode-se observar que a estratégia adotada pelo algoritmo gerador da solução inicial gerou um resultado bom, porém completamente diferente dos apresentados pelo programa com *setup* otimizado. A maximização do uso das ferramentas na cabeça insersora implicou em uma sensível redução no tempo gasto com a movimentação (que desde o projeto inicial do algoritmo foi o objetivo), porém atacar apenas o problema do *setup* com resultados satisfatórios não é o suficiente para obter um resultado de alta qualidade.

Instâncias deste porte, em casos especiais, permitem privilegiar algumas operações, como minimizar o *setup* de ferramentas ou a movimentação da cabeça insersora, sem afetar de forma significativa a qualidade final da resposta encontrada.

Conclui-se com estes experimentos que a análise de problemas de pequeno porte é importante por ser, para algumas linhas de produção, o regime principal de uso de uma máquina insersora. Comparando os três tipos de problema teste é possível verificar que o *setup* tende a ter a importância reduzida, quando o tamanho da placa montada aumenta. Também é possível observar que a dinâmica do modelo muda completamente para os casos estudados. Isso inviabiliza o uso de um algoritmo determinístico geral na busca de uma boa solução. Como exemplo, é possível citar o gerador de soluções iniciais. Quando codificado no início da implementação, tentou-se abranger o maior número possível de casos e condições; mesmo assim, apenas para casos médios o resultado foi satisfatório quando consideramos a qualidade da solução.

Apesar de não ser o objetivo do trabalho, pelos dados expostos, pode-se verificar, dependendo do tamanho da PCB montada (isso inclui a quantidade de componentes e diversidade dos mesmos), que máquinas insersoras com capacidade para um grande número de ferramentas na cabeça insersora podem não apresentar um resultado satisfatório no aumento da produtividade de uma linha de produção, quando esta operar com *mix* elevado de produtos, associados a placas com poucos componentes.

Fica claro também que essa análise é qualitativa e não quantitativa, já que os problemas não podem ser separados, conforme explicado ao longo do trabalho.

7.3 Análise da Interação entre a BT e o AG durante a Busca

Para visualizar a interação entre a Busca Tabu e o Algoritmo Genético, o problema ilustrado na parte superior da Figura 44 foi testado, e a evolução da FO encontrada foi analisada. Este problema artificial define a máquina com tendo uma cabeça com capacidade para 4 ferramentas, 100 componentes dispostos em 10 grupos de itens diferentes que necessitam 4 ferramentas distintas. Ainda na Figura 44, na parte inferior é possível verificar um gráfico com a evolução da busca.

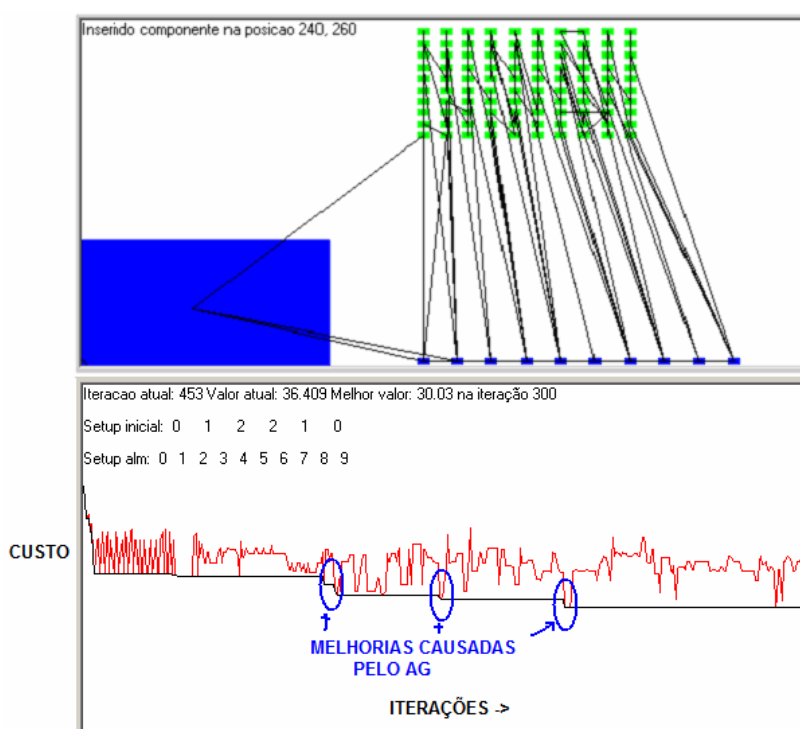


FIGURA 44– INTERAÇÃO ENTRE A BT E O AG DURANTE A BUSCA DA SOLUÇÃO

Na Figura 44, o eixo X representa as iterações e o eixo Y o melhor valor da FO na iteração correspondente. Na função que representa o custo é possível observar três pequenos degraus (destacados) causados pela execução do AG. Neste gráfico, é apresentado em preto o melhor valor na iteração correspondente e em vermelho o valor atual da busca na iteração. Isso permite visualizar o degrau da solução na troca das metaheurísticas. No problema analisado a BT tem dificuldades em efetuar melhorias que necessitem movimentações de um grande bloco de dados, situação que é natural para um AG. Combinando os dois pontos fortes de cada algoritmo, a metaheurística híbrida é capaz de sair de ótimos locais de difícil movimentação através do AG e intensificar a busca por uma boa solução através da BT. Na grande maioria dos casos testados, durante o desenvolvimento da pesquisa, as metaheurísticas

se alternaram apenas duas vezes. Após isso o AG não consegue mais melhorar a solução, o que satisfaz o critério de parada, encerrando o algoritmo.

Visando analisar o tempo de processamento gasto com cada metaheurística, os problemas apresentados na seção 7.2 tabela 6, tiveram os tempos de execução de cada metaheurística mensurados. Os resultados são apresentados na Tabela 8.

TABELA 8 – GASTO PERCENTUAL DE PROCESSAMENTO NA BUSCA DA SOLUÇÃO.

NOME DO PROBLEMA	PERCENTUAL GASTO COM A BT	PERCENTUAL GASTO COM O AG
RECEPTOR_23	41 %	59 %
ALARME_57	55 %	45 %
CENTRAL_315	79 %	21 %

Na Tabela 8, a primeira coluna representa o nome do problema, a segunda coluna o percentual do tempo usado para encontrar a solução gasto pela BT e na coluna mais a direita o percentual de tempo usado pelo AG para encontrar uma solução. Os percentuais são sempre relativos ao problema analisado. É possível observar que a medida que o problema cresce, o tempo de processamento consumido pela BT tende a aumentar. Isso ocorre primariamente devido ao modo como o algoritmo foi implementado e as intensas movimentações de dados realizadas pela BT.

7.4 Caso Real

Neste tópico, uma máquina real é apresentada, e seus parâmetros são introduzidos no modelo, utilizando a metodologia proposta em 7.4.2. Com a máquina já caracterizada, dois casos reais confrontarão os resultados obtidos pelo software da máquina com os resultados gerados pela metaheurística híbrida.

7.4.1 Máquina Utilizada

A máquina utilizada para testes será uma insersora *multi-head*, denominada *Quadra Basic*. Esta máquina é produzida pela *TWS Automation* e a foto da mesma é apresentada na Figura 45. O equipamento possui como características principais:

- a) Cabeça insersora com até 2 pinças;
- b) Capacidade para até 120 alimentadores;
- c) Inserção média nominal de 3600 componentes por hora.

A máquina apresentada na Figura 45 possui um software muito simples, que não

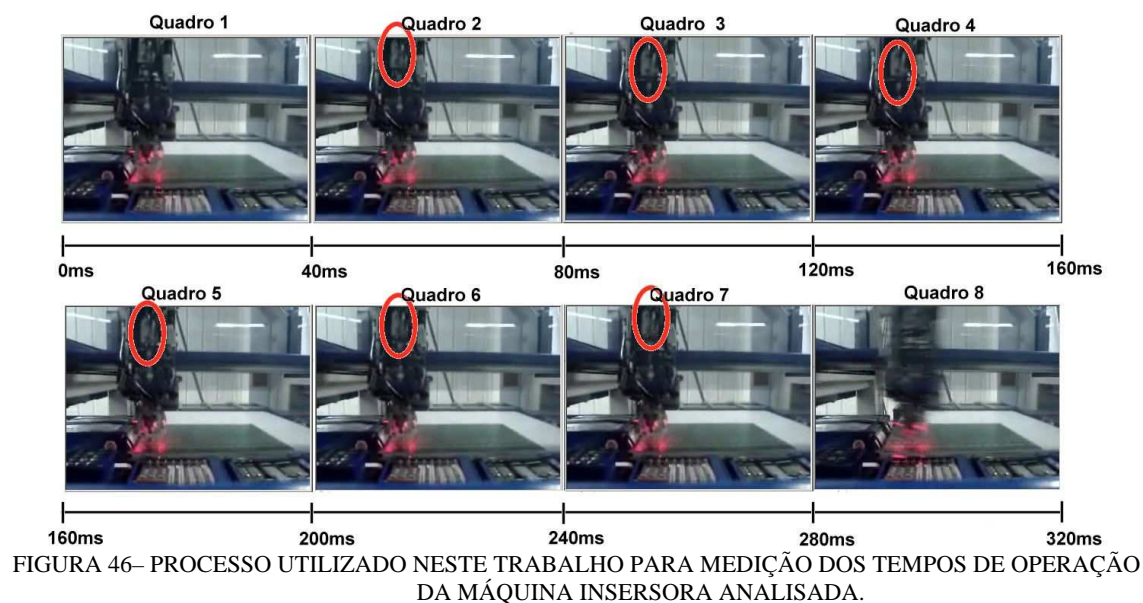


FIGURA 45– FOTO DA INSERSORA UTILIZADA PARA TESTES NO MODELO (FONTE: SITE TWS AUTOMATION [HTTP://WWW.TWS-AUTOMATION.COM](http://www.tws-automation.com)).

realiza qualquer tipo de otimização no programa de montagem visando extrair desempenho do conjunto mecânico. A ordem dos componentes nos alimentadores é especificada pelo usuário. A seqüência de inserção é ordenada, primeiramente em função da ferramenta mais utilizada e, posteriormente, pode ser efetuada uma ordenação em função das coordenadas de cada componente em X e Y na placa de circuito impresso. Existe ainda a possibilidade da seqüência de inserção não ser ordenada, neste caso a lista oriunda do CAD, com a posição de cada componente na placa de circuito impresso, será considerada como uma seqüência de montagem.

7.4.2 Obtenção dos Parâmetros Utilizados no Modelo Real

Com exceção do parâmetro F_{MAX} , que indica a quantidade de pinças a vácuo disponíveis na cabeça de inserção, nenhum outro parâmetro está disponível no manual da máquina, sendo assim, todos os parâmetros relativos a coordenadas serão obtidos através de medições na própria máquina. Os parâmetros que definem os tempos de operação da máquina apresentam valores inferiores a 1 segundo, em função disso, dentre os vários métodos existentes, a técnica utilizada para medição dos tempos foi feita através da análise de vídeo. Utilizando uma câmera de vídeo que captura imagens a uma taxa de 60 quadros por segundo, sabe-se que ocorre um quadro a cada $1/60s$ ou $16,67ms$. Com base nisso, fazendo a análise quadro a quadro do vídeo gravado, é possível medir os tempos de operação com uma resolução de $16,67ms$. A Figura 46 ilustra este processo de medição.



Na Figura 46 pode-se observar uma seqüência de oito quadros de uma filmagem realizada à uma taxa de 25 quadros por segundos (um quadro a cada 40ms). Nessa seqüência é analisado o tempo que a máquina gasta para carregar um componente do alimentador. No quadro 1 a cabeça insersora pára na posição de carregamento. No intervalo entre o quadro 2 e quadro 7 é possível ver o mecanismo de carregamento em movimento. No quadro 8, com a seqüência de carregamento completa, é possível ver claramente a cabeça de inserção iniciando o movimento. Fica claro que a exatidão deste modo de medição está ligada à velocidade com que a câmera captura os quadros.

7.4.3 Parâmetros Utilizados no Modelo Real

Na Tabela 9 são apresentados os parâmetros da máquina, levantados seguindo o método proposto em 7.4.2.

TABELA 9 – CARACTERIZAÇÃO DA MÁQUINA INSERSORA TESTADA.

PARÂMETRO	DESCRIÇÃO	VALOR
F_{MAX}	Número máximo de ferramentas simultâneas na cabeça de inserção.	2
V_r	Velocidade média da cabeça de inserção.	1200 mm/s
T_u	Tempo para inserir um componente	0,31s
T_p	Tempo para carregar um componente	0,31s
T_f	Tempo para troca de uma ferramenta	1 s
F_x, F_y	Coordenadas cartesianas do centro médio do alimentador de ferramentas.	(50,50) mm
Cx_n, Cy_n	Coordenadas cartesianas da posição dos componente no alimentador	(200,30)-(710,30) mm espaçados a cada 10mm em x
B_x, B_y	Coordenadas cartesianas do ponto (0,0) da PCB em relação a origem da máquina	(400,200) mm

Na Tabela 9, a coluna da esquerda apresenta os nomes das variáveis usadas no modelo. Na coluna central são descritos o significado físico de cada parâmetro e na coluna direita são apresentados os valores medidos.

A fim de tornar a análise o mais realista possível, o programa de montagem gerado pela metaheurística híbrida foi executado pela máquina e o tempo total de montagem foi comparado ao tempo de montagem, utilizando a solução gerada pelo software da máquina.

Foram abordados 2 casos, os quais são apresentados na Tabela 10.

TABELA 10 – OTIMIZAÇÃO DE UM CASO REAL.

DESCRIÇÃO	SOLUÇÃO MÁQUINA (SEGUNDOS)	MELHOR SOLUÇÃO METAHEURÍSTICA (SEGUNDOS)	DIFERENÇA PERCENTUAL	AUMENTO DO TEMPO MENSAL DISPONÍVEL PARA MONTAGEM	TEMPO DE PROCESSAMENTO (SEGUNDOS)
CASO 1	37,14 s	33,57 s	- 10,6 %	3,18%	348 s
CASO 2	75,08 s	72,16 s	-4,0%	0,84%	731 s

Na Tabela 10, a coluna a esquerda descreve o nome do caso testado. Foram escolhidos dois casos, o caso 1 corresponde a uma placa pequena (apenas 30 itens), o que corresponde a 30% do tempo total de produção no mês, porém representa apenas 17% no faturamento da empresa. O caso 2, corresponde a uma placa média com 72 itens montados. Neste caso, o produto consome 21% do tempo de produção, porém representa 31% do faturamento mensal. Na segunda coluna é apresentado o tempo de montagem, utilizando o programa gerado pelo software da máquina. Na terceira coluna é mostrado o tempo de montagem utilizando o programa de montagem gerado pela metaheurística. A quarta coluna apresenta a diferença percentual entre os tempos de montagem das duas colunas anteriores. A quinta coluna, apresenta quanto tempo disponível no mês foi poupado, utilizando o novo programa de montagem. A sexta coluna apresenta o tempo de processamento gasto pela metaheurística na busca da solução.

Pode-se observar, em ambos os casos, um ganho pequeno em relação ao programa apresentado pela máquina (máximo 10%). Isso ocorre primariamente, devido às características da máquina insersora modelada que tem capacidade para duas ferramentas na cabeça insersora. Foi constatado que quanto menor o número de ferramentas disponíveis na cabeça de inserção, menor é o percentual de otimização. Isso ocorre pois a solução determinística apresenta qualidade satisfatória devido a menor quantidade de configurações mecânicas possíveis em máquinas simples.

Para a fábrica estudada os produtos pequenos, além de possuírem um baixo valor agregado, correspondem a mais de 60% dos itens montados pela máquina. Para esses itens, especificamente, é muito interessante reduzir o tempo montagem, o quanto for possível. Isso diminui o custo de produção do item, além de liberar mais tempo para montagem de produtos de maior valor agregado. Considerando que atualmente o faturamento está limitado pela capacidade de produção da máquina, somente estas duas otimizações permitem elevar o faturamento médio da empresa em 4%, o que corresponde a aproximadamente R\$ 8000,00 por mês no ano da publicação desta dissertação.

No caso 1, a diversificação do Algoritmo Genético não trouxe melhora à solução. Devido ao tamanho do problema, a BT consegue sozinha otimizar o problema. No caso 2, como na grande maioria dos casos estudados, o Algoritmo Genético executou duas diversificações com melhora na solução.

8 CONSIDERAÇÕES FINAIS

Este trabalho abordou um tema que está associado ao problema do ciclo de montagem de uma máquina insersora, que em linhas gerais, é constituído por três sub-problemas complexos na área de otimização combinatória, onde o objetivo é reduzir o tempo total de montagem, elevando a produtividade da máquina.

O problema é definido pela literatura como NP-Difícil, o que motiva o uso de heurísticas e metaheurísticas para a abordagem do mesmo. A revisão bibliográfica apresenta soluções propostas por diversos autores, aplicadas a diversos tipos de máquina, onde modelos específicos foram desenvolvidos e metaheurísticas foram aplicadas. Grande parte dos trabalhos pesquisados não contemplam todos os sub-problemas associados, concentrando-se em sub-problemas específicos.

Neste trabalho foi desenvolvido um modelo teórico que aborda os três sub-problemas associados ao problema principal do ciclo de montagem de uma máquina insersora automática de componentes *multi-head*, sendo esse modelo utilizado no desenvolvimento da metaheurística híbrida, formada pelo uso das técnicas Busca Tabu e Algoritmos Genéticos.

No estudo do modelo, o trabalho aborda em detalhes todo o funcionamento do modelo proposto, apresentando de forma separada os três sub-problemas associados ao problema principal e como os mesmos foram inseridos no modelo proposto. A construção do modelo visou abrangência, assim, o mesmo é genérico e contempla um grande número de configurações mecânicas de máquinas insersoras *multi-head*.

A arquitetura do modelo foi apresentada visando elucidar como ocorreu o encadeamento entre a Busca Tabu e o Algoritmo Genético. O diagrama da arquitetura foi dividido em cinco fases, onde o objetivo de cada etapa foi apresentado.

A implementação da técnica computacional foi apresentada em detalhes. Estruturas para modelar o problema são apresentadas e soluções para problemas gerados pela forma de codificação são descritas. Técnicas para solução de problemas locais foram apresentadas, onde possível, com um exemplo associado.

A validação do modelo e do algoritmo foi feita separando, quando possível, o problema principal em problemas menores. Nos problemas menores, foram testadas instâncias com resultados conhecidos, utilizadas por outros autores, o que permitiu verificar a eficiência da implementação.

Nos experimentos, máquinas reais foram modeladas e problemas reais foram testados no modelo, utilizando a técnica computacional apresentada. Quando possível, os resultados foram comparados com os resultados reais obtidos testando as soluções geradas em uma máquina insersora real. Soluções compatíveis com resultados de outros autores foram encontradas e a técnica proposta foi capaz de reduzir custos e elevar a produtividade da máquina, quando comparada ao algoritmo apresentado pelo fabricante da máquina.

A hibridização permitiu que a implementação dos algoritmos em sua forma clássica, apresentasse resultado igual ou superior a trabalhos semelhantes, notadamente os problemas de teste da TSPLIB95. Apesar de não ser um fator importante no contexto onde os problemas estão inseridos (o algoritmo pode ser executado apenas uma vez), o tempo de processamento da metaheurística ficou dentro de condições realistas (menos de 60 minutos) os problemas apresentados, considerando os computadores existentes na data da publicação desta dissertação.

A abordagem conjunta dos três sub-problemas que compõem o problema principal constitui a grande contribuição deste trabalho para o meio científico. A grande maioria das publicações e trabalhos relacionados ou ignora algum sub-problema ou trata o mesmo separado dos demais.

8.1 Trabalhos Futuros

O objetivo do trabalho, que consistiu em desenvolver uma técnica computacional eficiente para o Problema do Ciclo de Operação de uma Máquina insersora, foi atingido. Entretanto, sempre há espaço para refinamentos. Em uma abordagem futura, a solução

completa pode conter uma nova operação que contemple especificamente a repetição do processo de montagem, em um determinado número de vezes. Esse parâmetro possibilitará ao modelo saber a quantidade de ciclos executados em seqüência, o que permitirá ao algoritmo de otimização optar em algumas instâncias por um programa de montagem com apenas um *setup* inicial de ferramentas, ou múltiplos *setups* de ferramentas durante o ciclo de montagem.

A abordagem adotada para o AG pode ser melhorada, permitindo que o mesmo opere nos três sub-problemas associados, assim como ocorre na BT. Essa modificação não afeta instâncias de pequeno e médio porte, porém pode ter um efeito muito positivo em instâncias de grande porte.

Um caso complexo, porém presente na indústria, pode ser abordado. Ele consiste na montagem automática de placas diferentes, utilizando uma configuração fixa de alimentadores para todas as placas. Atualmente, nesse caso, a configuração dos alimentadores é determinada por um operador humano. A implementação de múltiplas otimizações encadeadas pode permitir a determinação de uma configuração, que na média é boa para a maioria dos casos montados.

A base para as futuras implementações citadas consiste no modelo apresentado por este trabalho e provou ser bastante funcional. Especificamente a melhoria do AG e o Problema da Configuração Fixa de Alimentadores são casos que podem ser implementados reescrevendo o AG e partes específicas da codificação, aliadas a pequenas modificações no modelo. As poucas alterações no modelo, demonstram a versatilidade da solução apresentada.

REFERÊNCIAS BIBLIOGRÁFICAS

AHMADI, R.H; MAMER, J. W. *Routing heuristics for automated pick and place machines*. European Journal of Operational Research, p.117, 533-552, 1999.

ALTINKEMER, K.; KAZAZ, B.; KÖKSALAN, M.; MOSKOWITZ H. *Optimization of printed circuit board manufacturing: Integrated modeling and algorithms*. European Journal of Operational Research, p. 124, 409-421, 2000.

AYOB, M. *Optimisation of Surface Mount Device Placement Machine in Printed Circuit Board Assembly*, The School of Computer Science and Information Technology - University of Nottingham, 2005.

AYOB, M.; KENDALL, G. *A New Dynamic Point Specification Approach To Optimise Surface Mount Placement Machine in Printed Circuit Board Assembly*, The School of Computer Science and Information Technology - University of Nottingham, 2002.

BALL, M. O.; MAGAZINE, M.J. *Sequencing of insertions in printed circuit board assembly*, Operations Research, vol. 36(2), p. 192–201, 1988.

BENTZEN, B. *SMD placement in the SMT in FOCUS, 2000*. (Disponível em: http://www.labelectron.org.br/artigos/SMD_placement.pdf, acessado em 01/06/2008).

BIGGS, N. L.; LLOYD, E. K; WILSON, R. J. *Graph Theory 1736-1936* , Clarendon Press, Oxford, 1976.

BLAZEWICZ, J.; DOMSCHKE, W.; PESCH, E. *The job shop scheduling problem: Conventional and new solution techniques*. European Journal of Operational Research, vol. 93, p. 1-33, 1996.

BLAZEWICZ, J.; KOVALYOV, M. Y. *The complexity of two group scheduling problems*. Journal of Scheduling, vol. 5, p. 477 - 485, 2002.

BURKE, E.; COWLING, P.; KEUTHEN R. *The printed circuit board assembly problem: heuristic approaches for multi-headed placement machinery*. Proceedings of the IC-AI2001, p. 1456-1462, 2001.

COOK, S. A. *The complexity of theorem-proving procedures*, *Proceedings of the third annual ACM symposium on Theory of computing*, p.151-158, 1971.

CORMEN, T. H. *Introduction to algorithms*. The MIT Press, 2001.

CRAMA, Y.; . *Throughput rate optimization in the automated assembly of printed circuit boards*, *Annals of Operations Research*, vol. 26, p. 455–480, 1990.

CRAMA, Y.; KLUNDERT, J. van de; SPIEKSMAN, F. C. R. *Production planning problems in printed circuit board assembly*. *Discrete Applied Mathematics*, vol. 123, p. 339-361, 2002.

DIKOS, A.; NELSON, P. C.; TIRPAK, T. M.; WANG, W. *Optimization of high-mix printed circuit card assembly using genetic algorithms*, *Annals of Operation Research*, vol. 75, p. 303–324, 1997.

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. *The complexity of flowshop and jobshop scheduling*, *Mathematics of Operations Research*, vol. 1, p. 117-129, 1976.

GLOVER, F. *Future paths for integer programming and links to artificial intelligence*, *Computers and Operation Research*, vol.13, p 533-549, 1986.

GLOVER, F. *Tabu search methods in artificial intelligence and operation Research*. ORSA. Artificial intelligence Newsletter, n.1, 1987.

GLOVER, F. *Tabu search a tutorial*. USA: Center of Applied Artificial Intelligence-University of Colorado, 1990.

GLOVER, F.; LAGUNA, M. *Tabu Search*. [S.l.]: Kluwer Academic Publishers, 382 ISBN 0-7923-9965-X, 1997.

HOFFART, J. G.; WILEY, J. P. *High performance circuit boards*, United States Patent 4868350, New York, 1989.

JAIN, A. S.; MEERAN, S. *Deterministic Job-Shop Scheduling: Past, Present and Future*, European Journal of Operational Research, vol. 113, p. 390-434, 1998.

JEEVAN, K.; PARTHIBAN, A.; SEETHARAMU, K.; AZID, I.; QUADIR, I. *Optimization of PCB Component Placement using Genetic Algorithms*. Journal of Electronics Manufacturing, vol. 11, p. 69-79, 2002.

KING, J. R. *Machine-component grouping in production low analysis: an approach using a rank order clustering algorithm*. International Journal of Production Research, v. 18, p. 213-232, 1980.

KUMAR, R.; LI, H. *Integer programming approach to printed circuit board assembly time optimization*, IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advance Packaging , vol. 18, p. 720–727, 1995.

LEE, S. H.; LEE, B. H.; PARK, T. H. *A hierarchical method to improve the productivity of multi-head surface mounting machines*. Intelligent Automation and Soft Computing, Vol. 6, p. 291-301, 1999.

LEE, W.; HONG, J.; LEE, S.; LEE, B.; LEE, Y. *An Efficient Planning Algorithm for Multi-head Surface Mounting Machines using a Genetic Algorithm*. Journal of Universal Computer Science, vol. 5, 1999.

LEIPÄLÄ, T. AND NEVALAINEN, O. *Optimization of the movements of a component placement machine*. European Journal of Operational Research, 38, 167-177, 1989.

LORINI, F. J. *Tecnologia de Grupo e Organização da Manufatura*. [S.l.]: Universidade Federal de Santa Catarina, p. 105, 1993.

MAGYAR, G.; JOHNSON, M.; NEVALAINEN, O. *On solving single machine optimization problems in electronics assembly*. Journal of Electronics Manufacturing, vol. 9, p. 249-267, 1999.

MOYER, L. K.; GUPTA, S. M. *An Efficient Assembly Sequencing Heuristic for Printed Circuit Boards Configurations*, Journal of Electronics Manufacturing, vol. 7, p. 143-160, 1997.

MÜLLER, G. I.; GÓMEZ, A. T. *Utilização da Busca Tabu para a Geração de um Modelo Aplicado ao Job-Shop Scheduling Problem Considerando um Sistema de Manufatura Flexível*. Anais do V WORCAP, INPE, São José dos Campos, 2005.

OSMAN, I.H., LAPORTE, G. *Metaheuristics: A bibliography*. Annals of Operations Research, vol. 63, p. 511-628, 1996.

PETER, C.; TIRPAK T. M.; NELSON, P. C. *Optimization of a high-speed placement machine using tabu search algorithms*. Annals of Operations Research, vol. 96, p. 125–147, 2000.

PHAM, D.T.; KARABOGA, D. *Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks*. Springer-Verlag, 2000.

PRASAD, R. P. *Surface Mount Technology: Principles and Practice*, 2^a edição, 1989.

RAYWARD-SMITH, V.J.; OSMAN, I.H.; REEVES, C.R.; SMITH, G.D. *Modern Heuristic Search Methods*, John Wiley & Sons, p. 1-25, 1996.

SUN, D. S.; LEE, T. E.; KIM, K. H. *Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool*. International Journal of Production Economics, in press, vol. 95, edição 2, pages 245-264, 2005.

TIRPAK, T.M.; MOHAPATRA, P.K.; NELSON, P.C.; RAJBHANDARI, R.R. *A Generic Classification and Object-Oriented Simulation Toolkit for SMT Assembly Equipment*. IEEE Systems, Man, and Cybernetics Society, vol. 32, p. 104-122, 2002.

VIANA, G. V. R. *Metaheurísticas e Programação Paralela em Otimização Combinatória*. UFC Edições, 1998.

ANEXO A – INSTÂNCIAS EXTERNAS UTILIZADAS NA VALIDAÇÃO

ARQUIVO DE COORDENADAS DA INSTÂNCIA EIL51

FONTE: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> (acessado em 11/10/2008)

NAME : eil51

COMMENT : 51-city problem (Christofides/Eilon)

COMMENT : Optimal tour for eil51.tsp (426)

TYPE : TSP

DIMENSION : 51

EDGE_WEIGHT_TYPE : EUC_2D

NODE_COORD_SECTION

1 37 52	27 30 48
2 49 49	28 43 67
3 52 64	29 58 48
4 20 26	30 58 27
5 40 30	31 37 69
6 21 47	32 38 46
7 17 63	33 46 10
8 31 62	34 61 33
9 52 33	35 62 63
10 51 21	36 63 69
11 42 41	37 32 22
12 31 32	38 45 35
13 5 25	39 59 15
14 12 42	40 5 6
15 36 16	41 10 17
16 52 41	42 21 10
17 27 23	43 5 64
18 17 33	44 30 15
19 13 13	45 39 10
20 57 58	46 32 39
21 62 42	47 25 32
22 42 57	48 25 55
23 16 57	49 48 28
24 8 52	50 56 37
25 7 38	51 30 40
26 27 68	EOF

* Todas as instâncias são apresentadas em formato integral, visando facilitar a comparação deste trabalho com trabalhos futuros, a falta de referências as instâncias de teste utilizada em outros trabalhos é um item comum em grande parte dos textos pesquisados na revisão bibliográfica.

ARQUIVO DE COORDENADAS DA INSTÂNCIA D493

FONTE: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> (acessado em 11/10/2008)

NAME : D493

COMMENT : DRILLING PROBLEM (REINELT)

TYPE : TSP

DIMENSION : 493

EDGE_WEIGHT_TYPE : EUC_2D

NODE_COORD_SECTION

1 0.0000e+00 0.0000e+00	76 2.19580e+03 2.93950e+03	151 3.43410e+03 2.10130e+03	226 3.59920e+03 1.63140e+03
2 1.11630e+03 1.55520e+03	77 2.24660e+03 2.88870e+03	152 2.61490e+03 2.08860e+03	227 3.13560e+03 1.62510e+03
3 1.35760e+03 1.47900e+03	78 1.89100e+03 2.86970e+03	153 3.07850e+03 2.08230e+03	228 3.07850e+03 1.62510e+03
4 1.14810e+03 1.77110e+03	79 2.20220e+03 2.84430e+03	154 2.58950e+03 2.07590e+03	229 3.05940e+03 1.62510e+03
5 1.18620e+03 1.79650e+03	80 1.90370e+03 2.81890e+03	155 2.48790e+03 2.06320e+03	230 2.77370e+03 1.62510e+03
6 1.20520e+03 1.88540e+03	81 1.87200e+03 2.81890e+03	156 2.73560e+03 2.06320e+03	231 2.91340e+03 1.61870e+03
7 1.23700e+03 1.99340e+03	82 2.01170e+03 2.76170e+03	157 3.44680e+03 2.04420e+03	232 3.30710e+03 1.61870e+03
8 1.30050e+03 2.00610e+03	83 1.90370e+03 2.69820e+03	158 3.35150e+03 2.03150e+03	233 3.13560e+03 1.60600e+03
9 1.16080e+03 2.02510e+03	84 1.95450e+03 2.58390e+03	159 2.62760e+03 2.01880e+03	234 3.01500e+03 1.60600e+03
10 1.17350e+03 2.03780e+03	85 2.05610e+03 2.57760e+03	160 2.69750e+03 2.00610e+03	235 2.42440e+03 1.59970e+03
11 1.22430e+03 2.05050e+03	86 2.09420e+03 2.56490e+03	161 2.48790e+03 1.99340e+03	236 2.83080e+03 1.59330e+03
12 1.35130e+03 2.24100e+03	87 2.19580e+03 2.47600e+03	162 2.68480e+03 1.97430e+03	237 3.29440e+03 1.59330e+03
13 1.73230e+03 2.13310e+03	88 2.58950e+03 2.99030e+03	163 2.46250e+03 1.96800e+03	238 3.12930e+03 1.58060e+03
14 1.74500e+03 2.19020e+03	89 2.60860e+03 2.99030e+03	164 3.13560e+03 1.96800e+03	239 3.37690e+03 1.56790e+03
15 1.76400e+03 2.29180e+03	90 2.50700e+03 3.00300e+03	165 3.39600e+03 1.96800e+03	240 2.61490e+03 1.56790e+03
16 1.76400e+03 2.34260e+03	91 2.89430e+03 3.04750e+03	166 2.74830e+03 1.96160e+03	241 2.93240e+03 1.54890e+03
17 1.73230e+03 2.39340e+03	92 2.89430e+03 3.00300e+03	167 2.72290e+03 1.96160e+03	242 3.04620e+03 1.54730e+03
18 1.74500e+03 2.41880e+03	93 2.90700e+03 2.97130e+03	168 3.23090e+03 1.95530e+03	243 3.07220e+03 1.54730e+03
19 1.64970e+03 2.43150e+03	94 3.02770e+03 3.04750e+03	169 2.97050e+03 1.94260e+03	244 3.09820e+03 1.54730e+03
20 1.75770e+03 2.52040e+03	95 2.79270e+03 2.81250e+03	170 2.47520e+03 1.93620e+03	245 3.12420e+03 1.54730e+03
21 2.29740e+03 1.77110e+03	96 2.22760e+03 2.81250e+03	171 2.51330e+03 1.92350e+03	246 3.15020e+03 1.54730e+03
22 2.21490e+03 1.42820e+03	97 2.58320e+03 2.79980e+03	172 2.92610e+03 1.91720e+03	247 3.17620e+03 1.54730e+03
23 2.22760e+03 1.33930e+03	98 2.59590e+03 2.77440e+03	173 2.49430e+03 1.91080e+03	248 3.20220e+03 1.54730e+03
24 2.18310e+03 1.32030e+03	99 2.50700e+03 2.77440e+03	174 2.61490e+03 1.90450e+03	249 3.22820e+03 1.54730e+03
25 2.61490e+03 1.11070e+03	100 2.72290e+03 2.75540e+03	175 2.51970e+03 1.89180e+03	250 3.25420e+03 1.54730e+03
26 2.50700e+03 1.16150e+03	101 2.61490e+03 2.74900e+03	176 2.58950e+03 1.89180e+03	251 3.20020e+03 1.54730e+03
27 2.34820e+03 9.52000e+02	102 2.88800e+03 2.73000e+03	177 3.40870e+03 1.89180e+03	252 3.36260e+03 1.54730e+03
28 2.22120e+03 1.00910e+03	103 3.21180e+03 2.73000e+03	178 3.64360e+03 1.89180e+03	253 3.33220e+03 1.54730e+03
29 2.20850e+03 1.08530e+03	104 3.14200e+03 2.71730e+03	179 2.78640e+03 1.88540e+03	254 3.35820e+03 1.54730e+03
30 2.20220e+03 1.13610e+03	105 2.75460e+03 2.69820e+03	180 2.65940e+03 1.88540e+03	255 2.72290e+03 1.53620e+03
31 2.21490e+03 1.17420e+03	106 2.45620e+03 2.69820e+03	181 2.49430e+03 1.87910e+03	256 3.06570e+03 1.52460e+03
32 2.02440e+03 1.12340e+03	107 2.57050e+03 2.62200e+03	182 2.95780e+03 1.87910e+03	257 3.11120e+03 1.52460e+03
33 2.04340e+03 1.30120e+03	108 2.72290e+03 2.62200e+03	183 3.45950e+03 1.87910e+03	258 3.13720e+03 1.52460e+03
34 1.94820e+03 1.30760e+03	109 3.36420e+03 2.62200e+03	184 3.05310e+03 1.86640e+03	259 3.16320e+03 1.52460e+03
35 1.94180e+03 1.39010e+03	110 2.90070e+03 2.61570e+03	185 2.77370e+03 1.86640e+03	260 3.18920e+03 1.52460e+03
36 1.84660e+03 1.41550e+03	111 3.13560e+03 2.60300e+03	186 2.71020e+03 1.86640e+03	261 3.21520e+03 1.52460e+03
37 1.75770e+03 1.36470e+03	112 2.71020e+03 2.59660e+03	187 3.23720e+03 1.85370e+03	262 3.24120e+03 1.52460e+03
38 1.80210e+03 1.35200e+03	113 2.71650e+03 2.54580e+03	188 3.32610e+03 1.85370e+03	263 3.26720e+03 1.52460e+03
39 1.83390e+03 1.26310e+03	114 2.88800e+03 2.53310e+03	189 2.73560e+03 1.84730e+03	264 3.29320e+03 1.52460e+03
40 1.73230e+03 1.25040e+03	115 2.84990e+03 2.51410e+03	190 2.49430e+03 1.83460e+03	265 3.34520e+03 1.52460e+03
41 2.04340e+03 1.58060e+03	116 2.29740e+03 2.51410e+03	191 3.26900e+03 1.83460e+03	266 2.43710e+03 1.52350e+03
42 2.04980e+03 1.74570e+03	117 2.60860e+03 2.49500e+03	192 3.30710e+03 1.83460e+03	267 2.96420e+03 1.51080e+03
43 1.99900e+03 1.80290e+03	118 3.22450e+03 2.48870e+03	193 3.36420e+03 1.83460e+03	268 2.74190e+03 1.50440e+03
44 1.96090e+03 1.81560e+03	119 2.69110e+03 2.47600e+03	194 2.67210e+03 1.82830e+03	269 2.43080e+03 1.50440e+03
45 2.05610e+03 1.82190e+03	120 2.87530e+03 2.43790e+03	195 3.35150e+03 1.82190e+03	270 3.41350e+03 1.49860e+03
46 2.01170e+03 1.82830e+03	121 2.50060e+03 2.41880e+03	196 3.47220e+03 1.82190e+03	271 2.99750e+03 1.49210e+03
47 2.04340e+03 1.85370e+03	122 2.43080e+03 2.39980e+03	197 2.86890e+03 1.81560e+03	272 2.82450e+03 1.48540e+03
48 1.84660e+03 1.78380e+03	123 2.87530e+03 2.39980e+03	198 2.81180e+03 1.80920e+03	273 3.02020e+03 1.47910e+03
49 1.82120e+03 1.82830e+03	124 3.12930e+03 2.36800e+03	199 3.35150e+03 1.80290e+03	274 3.39070e+03 1.47910e+03
50 1.84660e+03 1.86640e+03	125 2.58320e+03 2.36800e+03	200 3.29440e+03 1.79650e+03	275 3.41350e+03 1.47260e+03
51 1.81480e+03 1.65050e+03	126 2.49430e+03 2.36800e+03	201 2.54510e+03 1.79650e+03	276 2.83720e+03 1.46630e+03
52 1.66880e+03 1.80920e+03	127 2.78000e+03 2.36170e+03	202 3.11020e+03 1.77750e+03	277 2.60860e+03 1.46630e+03
53 1.85930e+03 2.16480e+03	128 2.33550e+03 2.35530e+03	203 3.16100e+03 1.77750e+03	278 2.99750e+03 1.46610e+03
54 1.98630e+03 2.13310e+03	129 2.61490e+03 2.31720e+03	204 3.28170e+03 1.77750e+03	279 3.10470e+03 1.46610e+03
55 2.01170e+03 2.17750e+03	130 2.92610e+03 2.30450e+03	205 3.33880e+03 1.75840e+03	280 3.13070e+03 1.46610e+03
56 1.94820e+03 2.19660e+03	131 2.80540e+03 2.29180e+03	206 2.99590e+03 1.75840e+03	281 3.15670e+03 1.46610e+03
57 1.89100e+03 2.24740e+03	132 2.58320e+03 2.29180e+03	207 3.19910e+03 1.72030e+03	282 3.18270e+03 1.46610e+03
58 2.01170e+03 2.27280e+03	133 2.82450e+03 2.28550e+03	208 3.00230e+03 1.72030e+03	283 3.20870e+03 1.46610e+03
59 2.03710e+03 2.30450e+03	134 2.86260e+03 2.27910e+03	209 2.63400e+03 1.72030e+03	284 3.23470e+03 1.46610e+03
60 2.05610e+03 2.35530e+03	135 2.79910e+03 2.26010e+03	210 2.50700e+03 1.70760e+03	285 3.26070e+03 1.46610e+03
61 1.89740e+03 2.41250e+03	136 2.69110e+03 2.26010e+03	211 2.96420e+03 1.70760e+03	286 3.28670e+03 1.46610e+03
62 1.89740e+03 2.45690e+03	137 2.75460e+03 2.25370e+03	212 3.05940e+03 1.70760e+03	287 3.31270e+03 1.46610e+03
63 1.91640e+03 2.46960e+03	138 2.91340e+03 2.25370e+03	213 3.09750e+03 1.70760e+03	288 3.41350e+03 1.44660e+03
64 2.05610e+03 2.46960e+03	139 2.34820e+03 2.24100e+03	214 3.64360e+03 1.70760e+03	289 3.07870e+03 1.44660e+03
65 2.15770e+03 2.13940e+03	140 2.90700e+03 2.23470e+03	215 2.80540e+03 1.70130e+03	290 3.11770e+03 1.44330e+03
66 1.63700e+03 2.69820e+03	141 3.13560e+03 2.23470e+03	216 2.58320e+03 1.69490e+03	291 3.14370e+03 1.44330e+03
67 1.70690e+03 2.70460e+03	142 2.92610e+03 2.22830e+03	217 3.07850e+03 1.69490e+03	292 3.16970e+03 1.44330e+03
68 1.76400e+03 2.74900e+03	143 3.43410e+03 2.20290e+03	218 2.89430e+03 1.68860e+03	293 3.19570e+03 1.44330e+03
69 1.70690e+03 2.77440e+03	144 2.88160e+03 2.19020e+03	219 3.14830e+03 1.68220e+03	294 3.22170e+03 1.44330e+03
70 1.76400e+03 2.79980e+03	145 2.60220e+03 2.19020e+03	220 3.21820e+03 1.66320e+03	295 3.24770e+03 1.44330e+03
71 1.87200e+03 3.06650e+03	146 2.71020e+03 2.16480e+03	221 3.19910e+03 1.66320e+03	296 3.27370e+03 1.44330e+03
72 1.85290e+03 3.00300e+03	147 2.79270e+03 2.14580e+03	222 2.72920e+03 1.66320e+03	297 2.90070e+03 1.44090e+03
73 1.87200e+03 2.99670e+03	148 2.82450e+03 2.13310e+03	223 2.95150e+03 1.65050e+03	298 2.69110e+03 1.44090e+03
74 1.89740e+03 2.99670e+03	149 2.42440e+03 2.10770e+03	224 2.43080e+03 1.64410e+03	299 2.99750e+03 1.44010e+03
75 2.21490e+03 2.97130e+03	150 2.86890e+03 2.10130e+03	225 2.50700e+03 1.63140e+03	300 3.33220e+03 1.44010e+03

301 2.43710e+03 1.43460e+03	350 2.99750e+03 1.33610e+03	399 2.51970e+03 1.24410e+03	448 3.15670e+03 1.13130e+03
302 3.39070e+03 1.43360e+03	351 3.33220e+03 1.33610e+03	400 3.07870e+03 1.23860e+03	449 3.18270e+03 1.13130e+03
303 3.30950e+03 1.42710e+03	352 2.91340e+03 1.33300e+03	401 3.41350e+03 1.23860e+03	450 3.20870e+03 1.13130e+03
304 3.02020e+03 1.42710e+03	353 3.10150e+03 1.32960e+03	402 3.29320e+03 1.23530e+03	451 3.23470e+03 1.13130e+03
305 2.58320e+03 1.42190e+03	354 3.39070e+03 1.32960e+03	403 3.26720e+03 1.23530e+03	452 3.26070e+03 1.13130e+03
306 3.07870e+03 1.42060e+03	355 3.30950e+03 1.32310e+03	404 3.24120e+03 1.23530e+03	453 3.28670e+03 1.13130e+03
307 3.41350e+03 1.42060e+03	356 3.24360e+03 1.32310e+03	405 3.21520e+03 1.23530e+03	454 3.31270e+03 1.13130e+03
308 3.28350e+03 1.41730e+03	357 3.02020e+03 1.32310e+03	406 3.18920e+03 1.23530e+03	455 3.33870e+03 1.13130e+03
309 3.12750e+03 1.41730e+03	358 2.95780e+03 1.32030e+03	407 3.16320e+03 1.23530e+03	456 3.36470e+03 1.13130e+03
310 2.99750e+03 1.41410e+03	359 3.15470e+03 1.32030e+03	408 3.13720e+03 1.23530e+03	457 2.74190e+03 1.12980e+03
311 3.33220e+03 1.41410e+03	360 3.07870e+03 1.31660e+03	409 2.99750e+03 1.23210e+03	458 2.86260e+03 1.12340e+03
312 3.23720e+03 1.40920e+03	361 3.41350e+03 1.31660e+03	410 3.33220e+03 1.23210e+03	459 3.06580e+03 1.09800e+03
313 3.19910e+03 1.40920e+03	362 3.61190e+03 1.31390e+03	411 2.88160e+03 1.22500e+03	460 3.03400e+03 1.09170e+03
314 3.10150e+03 1.40760e+03	363 3.33220e+03 1.31010e+03	412 3.09820e+03 1.21260e+03	461 2.87530e+03 1.07900e+03
315 3.39070e+03 1.40760e+03	364 2.99750e+03 1.31010e+03	413 3.12420e+03 1.21260e+03	462 2.79270e+03 1.04090e+03
316 3.44040e+03 1.40280e+03	365 3.10150e+03 1.30360e+03	414 3.15020e+03 1.21260e+03	463 2.90070e+03 1.02820e+03
317 2.95150e+03 1.40280e+03	366 3.39070e+03 1.30360e+03	415 3.17620e+03 1.21260e+03	464 2.83720e+03 1.02180e+03
318 3.02020e+03 1.40110e+03	367 3.30950e+03 1.29710e+03	416 3.20220e+03 1.21260e+03	465 2.94510e+03 1.01550e+03
319 3.30950e+03 1.40110e+03	368 3.02020e+03 1.29710e+03	417 3.22820e+03 1.21260e+03	466 2.86260e+03 1.00910e+03
320 3.41350e+03 1.39460e+03	369 2.84990e+03 1.29490e+03	418 3.25420e+03 1.21260e+03	467 2.93880e+03 8.31300e+02
321 3.07870e+03 1.39460e+03	370 2.87530e+03 1.29490e+03	419 3.28020e+03 1.21260e+03	468 3.07210e+03 9.32900e+02
322 3.16100e+03 1.39010e+03	371 2.90070e+03 1.29490e+03	420 3.30620e+03 1.21260e+03	469 3.05940e+03 9.52000e+02
323 2.99750e+03 1.38810e+03	372 3.54200e+03 1.29490e+03	421 3.41350e+03 1.21260e+03	470 3.03400e+03 9.64700e+02
324 3.33220e+03 1.38810e+03	373 3.41350e+03 1.29060e+03	422 2.58950e+03 1.21230e+03	471 3.06580e+03 9.71000e+02
325 2.93880e+03 1.38380e+03	374 3.07870e+03 1.29060e+03	423 2.57050e+03 1.21230e+03	472 3.09120e+03 9.90100e+02
326 2.87530e+03 1.38380e+03	375 3.17370e+03 1.28850e+03	424 2.99750e+03 1.20610e+03	473 3.20550e+03 9.83700e+02
327 3.10150e+03 1.38160e+03	376 3.33220e+03 1.28410e+03	425 3.02020e+03 1.19960e+03	474 3.40870e+03 8.75800e+02
328 3.39070e+03 1.38160e+03	377 2.99750e+03 1.28410e+03	426 3.39070e+03 1.19960e+03	475 3.45950e+03 9.20200e+02
329 3.13990e+03 1.37540e+03	378 3.10150e+03 1.27760e+03	427 3.45950e+03 1.19960e+03	476 3.51030e+03 1.01550e+03
330 3.02020e+03 1.37510e+03	379 3.39070e+03 1.27760e+03	428 3.48490e+03 1.19960e+03	477 3.36420e+03 1.01550e+03
331 3.30950e+03 1.37510e+03	380 3.30950e+03 1.27110e+03	429 2.65940e+03 1.19330e+03	478 3.38330e+03 1.02820e+03
332 3.41350e+03 1.36860e+03	381 3.02020e+03 1.27110e+03	430 3.41350e+03 1.18660e+03	479 3.49760e+03 1.02820e+03
333 3.07870e+03 1.36860e+03	382 2.76730e+03 1.26950e+03	431 2.93880e+03 1.18040e+03	480 3.39600e+03 1.05360e+03
334 2.79910e+03 1.36470e+03	383 3.07870e+03 1.26460e+03	432 2.99750e+03 1.18010e+03	481 3.29440e+03 1.05360e+03
335 3.59920e+03 1.36470e+03	384 3.41350e+03 1.26460e+03	433 3.06570e+03 1.15410e+03	482 3.51660e+03 1.07260e+03
336 3.33220e+03 1.36210e+03	385 2.60220e+03 1.26310e+03	434 3.11770e+03 1.15410e+03	483 3.21820e+03 1.09800e+03
337 2.99750e+03 1.36210e+03	386 3.12750e+03 1.26130e+03	435 3.14370e+03 1.15410e+03	484 3.56740e+03 1.10440e+03
338 3.10150e+03 1.35560e+03	387 3.28350e+03 1.26130e+03	436 3.16970e+03 1.15410e+03	485 3.55470e+03 1.12340e+03
339 3.39070e+03 1.35560e+03	388 3.33220e+03 1.25810e+03	437 3.19570e+03 1.15410e+03	486 3.39600e+03 1.12340e+03
340 3.26900e+03 1.35200e+03	389 2.99750e+03 1.25810e+03	438 3.22170e+03 1.15410e+03	487 3.54840e+03 1.14250e+03
341 2.91970e+03 1.35200e+03	390 2.88160e+03 1.25680e+03	439 3.24770e+03 1.15410e+03	488 3.52930e+03 1.15520e+03
342 2.83720e+03 1.35200e+03	391 2.79270e+03 1.25680e+03	440 3.27370e+03 1.15410e+03	489 3.40870e+03 1.16150e+03
343 3.02020e+03 1.34910e+03	392 2.55780e+03 1.25680e+03	441 3.29970e+03 1.15410e+03	490 3.47220e+03 1.17420e+03
344 3.30950e+03 1.34910e+03	393 3.10150e+03 1.25160e+03	442 3.34520e+03 1.15410e+03	491 3.69440e+03 1.30120e+03
345 2.87530e+03 1.34570e+03	394 3.39070e+03 1.25160e+03	443 2.92610e+03 1.14250e+03	492 3.74520e+03 1.27580e+03
346 3.07870e+03 1.34260e+03	395 3.47850e+03 1.25040e+03	444 3.05270e+03 1.13130e+03	493 3.74520e+03 1.02180e+03
347 3.41350e+03 1.34260e+03	396 3.02020e+03 1.24510e+03	445 3.07870e+03 1.13130e+03	EOF
348 3.48490e+03 1.33930e+03	397 2.91970e+03 1.24410e+03	446 3.10470e+03 1.13130e+03	
349 2.97050e+03 1.33930e+03	398 2.81810e+03 1.24410e+03	447 3.13070e+03 1.13130e+03	

ARQUIVO DE COORDENADAS DA INSTÂNCIA D2103

FONTE: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/> (acessado em 11/10/2008)

NAME : D2103

COMMENT : DRILLING PROBLEM (REINELT)

TYPE : TSP

DIMENSION : 2103

EDGE_WEIGHT_TYPE : EUC_2D

NODE_COORD_SECTION

1 0.0000e+00 0.0000e+00	74 9.8320e+02 2.2855e+03	147 2.0135e+03 2.1331e+03	220 1.1417e+03 1.9299e+03
2 6.9110e+02 8.5670e+02	75 1.0229e+03 2.2855e+03	148 2.0531e+03 2.1331e+03	221 1.1814e+03 1.9299e+03
3 7.4190e+02 8.5670e+02	76 1.0625e+03 2.2855e+03	149 2.0927e+03 2.1331e+03	222 1.2210e+03 1.9299e+03
4 7.9270e+02 8.5670e+02	77 1.1021e+03 2.2855e+03	150 2.1323e+03 2.1331e+03	223 1.2606e+03 1.9299e+03
5 8.6890e+02 8.6310e+02	78 1.1417e+03 2.2855e+03	151 2.1720e+03 2.1331e+03	224 1.3002e+03 1.9299e+03
6 7.9270e+02 9.0750e+02	79 1.1814e+03 2.2855e+03	152 2.2116e+03 2.1331e+03	225 1.3399e+03 1.9299e+03
7 6.9110e+02 9.0750e+02	80 1.2210e+03 2.2855e+03	153 2.2512e+03 2.1331e+03	226 1.3795e+03 1.9299e+03
8 8.6890e+02 9.1390e+02	81 1.2606e+03 2.2855e+03	154 2.2908e+03 2.1331e+03	227 1.4191e+03 1.9299e+03
9 7.9270e+02 9.5830e+02	82 1.3002e+03 2.2855e+03	155 2.3305e+03 2.1331e+03	228 1.4587e+03 1.9299e+03
10 7.4190e+02 9.5830e+02	83 1.3399e+03 2.2855e+03	156 2.3701e+03 2.1331e+03	229 1.4984e+03 1.9299e+03
11 6.9110e+02 9.5830e+02	84 1.3795e+03 2.2855e+03	157 2.4097e+03 2.1331e+03	230 1.5380e+03 1.9299e+03
12 6.9110e+02 1.6187e+03	85 1.4191e+03 2.2855e+03	158 2.4493e+03 2.1331e+03	231 1.5776e+03 1.9299e+03
13 7.4190e+02 1.6187e+03	86 1.4587e+03 2.2855e+03	159 2.4890e+03 2.1331e+03	232 1.6172e+03 1.9299e+03
14 7.9270e+02 1.6187e+03	87 1.4984e+03 2.2855e+03	160 2.5286e+03 2.1331e+03	233 1.6569e+03 1.9299e+03
15 7.9270e+02 1.6695e+03	88 1.5380e+03 2.2855e+03	161 2.5682e+03 2.1331e+03	234 1.6965e+03 1.9299e+03
16 6.9110e+02 1.6695e+03	89 1.5776e+03 2.2855e+03	162 2.6078e+03 2.1331e+03	235 1.7361e+03 1.9299e+03
17 6.9110e+02 1.7203e+03	90 1.6172e+03 2.2855e+03	163 2.6475e+03 2.1331e+03	236 1.7757e+03 1.9299e+03
18 7.4190e+02 1.7203e+03	91 1.6569e+03 2.2855e+03	164 2.6475e+03 2.0823e+03	237 1.8154e+03 1.9299e+03
19 7.9270e+02 1.7203e+03	92 1.6965e+03 2.2855e+03	165 2.6078e+03 2.0823e+03	238 1.8550e+03 1.9299e+03
20 1.0975e+03 2.4252e+03	93 1.7361e+03 2.2855e+03	166 2.5682e+03 2.0823e+03	239 1.8946e+03 1.9299e+03
21 9.0700e+02 2.4125e+03	94 1.7757e+03 2.2855e+03	167 2.5286e+03 2.0823e+03	240 1.9342e+03 1.9299e+03
22 8.3080e+02 2.4125e+03	95 1.8154e+03 2.2855e+03	168 2.4890e+03 2.0823e+03	241 1.9738e+03 1.9299e+03
23 1.0340e+03 2.3871e+03	96 1.8550e+03 2.2855e+03	169 2.4493e+03 2.0823e+03	242 2.0135e+03 1.9299e+03
24 1.0594e+03 2.3871e+03	97 1.8946e+03 2.2855e+03	170 2.4097e+03 2.0823e+03	243 2.0531e+03 1.9299e+03
25 8.5620e+02 2.3744e+03	98 1.9342e+03 2.2855e+03	171 2.3701e+03 2.0823e+03	244 2.0927e+03 1.9299e+03
26 8.3080e+02 2.3617e+03	99 1.9738e+03 2.2855e+03	172 2.3305e+03 2.0823e+03	245 2.1323e+03 1.9299e+03
27 8.5620e+02 2.3490e+03	100 2.0135e+03 2.2855e+03	173 2.2908e+03 2.0823e+03	246 2.1720e+03 1.9299e+03
28 8.3080e+02 2.3363e+03	101 2.0531e+03 2.2855e+03	174 2.2512e+03 2.0823e+03	247 2.2116e+03 1.9299e+03
29 9.8320e+02 2.3363e+03	102 2.0927e+03 2.2855e+03	175 2.2116e+03 2.0823e+03	248 2.2512e+03 1.9299e+03
30 1.0229e+03 2.3363e+03	103 2.1323e+03 2.2855e+03	176 2.1720e+03 2.0823e+03	249 2.2908e+03 1.9299e+03
31 1.0625e+03 2.3363e+03	104 2.1720e+03 2.2855e+03	177 2.1323e+03 2.0823e+03	250 2.3305e+03 1.9299e+03
32 1.1021e+03 2.3363e+03	105 2.2116e+03 2.2855e+03	178 2.0927e+03 2.0823e+03	251 2.3701e+03 1.9299e+03
33 1.1417e+03 2.3363e+03	106 2.2512e+03 2.2855e+03	179 2.0531e+03 2.0823e+03	252 2.4097e+03 1.9299e+03
34 1.1814e+03 2.3363e+03	107 2.2908e+03 2.2855e+03	180 2.0135e+03 2.0823e+03	253 2.4493e+03 1.9299e+03
35 1.2210e+03 2.3363e+03	108 2.3305e+03 2.2855e+03	181 1.9738e+03 2.0823e+03	254 2.4890e+03 1.9299e+03
36 1.2606e+03 2.3363e+03	109 2.3701e+03 2.2855e+03	182 1.9342e+03 2.0823e+03	255 2.5286e+03 1.9299e+03
37 1.3002e+03 2.3363e+03	110 2.4097e+03 2.2855e+03	183 1.8946e+03 2.0823e+03	256 2.5682e+03 1.9299e+03
38 1.3399e+03 2.3363e+03	111 2.4493e+03 2.2855e+03	184 1.8550e+03 2.0823e+03	257 2.6078e+03 1.9299e+03
39 1.3795e+03 2.3363e+03	112 2.4890e+03 2.2855e+03	185 1.8154e+03 2.0823e+03	258 2.6475e+03 1.9299e+03
40 1.4191e+03 2.3363e+03	113 2.5286e+03 2.2855e+03	186 1.7757e+03 2.0823e+03	259 8.8160e+02 1.9172e+03
41 1.4587e+03 2.3363e+03	114 2.5682e+03 2.2855e+03	187 1.7361e+03 2.0823e+03	260 8.3080e+02 1.9720e+03
42 1.4984e+03 2.3363e+03	115 2.6078e+03 2.2855e+03	188 1.6965e+03 2.0823e+03	261 8.1810e+02 1.8918e+03
43 1.5380e+03 2.3363e+03	116 2.6475e+03 2.2855e+03	189 1.6569e+03 2.0823e+03	262 8.8160e+02 1.8918e+03
44 1.5776e+03 2.3363e+03	117 1.0594e+03 2.2347e+03	190 1.6172e+03 2.0823e+03	263 9.8320e+02 1.8791e+03
45 1.6172e+03 2.3363e+03	118 1.0340e+03 2.2347e+03	191 1.5776e+03 2.0823e+03	264 1.0229e+03 1.8791e+03
46 1.6569e+03 2.3363e+03	119 1.0340e+03 2.1839e+03	192 1.5380e+03 2.0823e+03	265 1.0625e+03 1.8791e+03
47 1.6965e+03 2.3363e+03	120 1.0594e+03 2.1839e+03	193 1.4984e+03 2.0823e+03	266 1.1021e+03 1.8791e+03
48 1.7361e+03 2.3363e+03	121 9.8320e+02 2.1331e+03	194 1.4587e+03 2.0823e+03	267 1.1417e+03 1.8791e+03
49 1.7757e+03 2.3363e+03	122 1.0229e+03 2.1331e+03	195 1.4191e+03 2.0823e+03	268 1.1814e+03 1.8791e+03
50 1.8154e+03 2.3363e+03	123 1.0625e+03 2.1331e+03	196 1.3795e+03 2.0823e+03	269 1.2210e+03 1.8791e+03
51 1.8550e+03 2.3363e+03	124 1.1021e+03 2.1331e+03	197 1.3399e+03 2.0823e+03	270 1.2606e+03 1.8791e+03
52 1.8946e+03 2.3363e+03	125 1.1417e+03 2.1331e+03	198 1.3002e+03 2.0823e+03	271 1.3002e+03 1.8791e+03
53 1.9342e+03 2.3363e+03	126 1.1814e+03 2.1331e+03	199 1.2606e+03 2.0823e+03	272 1.3399e+03 1.8791e+03
54 1.9738e+03 2.3363e+03	127 1.2210e+03 2.1331e+03	200 1.2210e+03 2.0823e+03	273 1.3795e+03 1.8791e+03
55 2.0135e+03 2.3363e+03	128 1.2606e+03 2.1331e+03	201 1.1814e+03 2.0823e+03	274 1.4191e+03 1.8791e+03
56 2.0531e+03 2.3363e+03	129 1.3002e+03 2.1331e+03	202 1.1417e+03 2.0823e+03	275 1.4587e+03 1.8791e+03
57 2.0927e+03 2.3363e+03	130 1.3399e+03 2.1331e+03	203 1.1021e+03 2.0823e+03	276 1.4984e+03 1.8791e+03
58 2.1323e+03 2.3363e+03	131 1.3795e+03 2.1331e+03	204 1.0625e+03 2.0823e+03	277 1.5380e+03 1.8791e+03
59 2.1720e+03 2.3363e+03	132 1.4191e+03 2.1331e+03	205 1.0229e+03 2.0823e+03	278 1.5776e+03 1.8791e+03
60 2.2116e+03 2.3363e+03	133 1.4587e+03 2.1331e+03	206 9.8320e+02 2.0823e+03	279 1.6172e+03 1.8791e+03
61 2.2512e+03 2.3363e+03	134 1.4984e+03 2.1331e+03	207 1.0340e+03 2.0315e+03	280 1.6569e+03 1.8791e+03
62 2.2908e+03 2.3363e+03	135 1.5380e+03 2.1331e+03	208 1.0594e+03 2.0315e+03	281 1.6965e+03 1.8791e+03
63 2.3305e+03 2.3363e+03	136 1.5776e+03 2.1331e+03	209 8.8160e+02 1.9934e+03	282 1.7361e+03 1.8791e+03
64 2.3701e+03 2.3363e+03	137 1.6172e+03 2.1331e+03	210 1.0340e+03 1.9807e+03	283 1.7757e+03 1.8791e+03
65 2.4097e+03 2.3363e+03	138 1.6569e+03 2.1331e+03	211 1.0594e+03 1.9807e+03	284 1.8154e+03 1.8791e+03
66 2.4493e+03 2.3363e+03	139 1.6965e+03 2.1331e+03	212 8.8160e+02 1.9680e+03	285 1.8550e+03 1.8791e+03
67 2.4890e+03 2.3363e+03	140 1.7361e+03 2.1331e+03	213 8.5620e+02 1.9680e+03	286 1.8946e+03 1.8791e+03
68 2.5286e+03 2.3363e+03	141 1.7757e+03 2.1331e+03	214 8.4350e+02 1.9426e+03	287 1.9342e+03 1.8791e+03
69 2.5682e+03 2.3363e+03	142 1.8154e+03 2.1331e+03	215 8.8160e+02 1.9426e+03	288 1.9738e+03 1.8791e+03
70 2.6078e+03 2.3363e+03	143 1.8550e+03 2.1331e+03	216 9.8320e+02 1.9299e+03	289 2.0135e+03 1.8791e+03
71 2.6475e+03 2.3363e+03	144 1.8946e+03 2.1331e+03	217 1.0229e+03 1.9299e+03	290 2.0531e+03 1.8791e+03
72 8.5620e+02 2.3236e+03	145 1.9342e+03 2.1331e+03	218 1.0625e+03 1.9299e+03	291 2.0927e+03 1.8791e+03
73 8.5620e+02 2.2982e+03	146 1.9738e+03 2.1331e+03	219 1.1021e+03 1.9299e+03	292 2.1323e+03 1.8791e+03

293 2.17200e+03 1.87910e+03 386 2.09270e+03 1.67590e+03 479 2.13230e+03 1.47270e+03 572 1.85500e+03 1.26950e+03
 294 2.21160e+03 1.87910e+03 387 2.13230e+03 1.67590e+03 480 2.17200e+03 1.47270e+03 573 1.89460e+03 1.26950e+03
 295 2.25120e+03 1.87910e+03 388 2.17200e+03 1.67590e+03 481 2.21160e+03 1.47270e+03 574 1.93420e+03 1.26950e+03
 296 2.29080e+03 1.87910e+03 389 2.21160e+03 1.67590e+03 482 2.25120e+03 1.47270e+03 575 1.97380e+03 1.26950e+03
 297 2.33050e+03 1.87910e+03 390 2.25120e+03 1.67590e+03 483 2.29080e+03 1.47270e+03 576 2.01350e+03 1.26950e+03
 298 2.37010e+03 1.87910e+03 391 2.29080e+03 1.67590e+03 484 2.33050e+03 1.47270e+03 577 2.05310e+03 1.26950e+03
 299 2.40970e+03 1.87910e+03 392 2.33050e+03 1.67590e+03 485 2.37010e+03 1.47270e+03 578 2.09270e+03 1.26950e+03
 300 2.44930e+03 1.87910e+03 393 2.37010e+03 1.67590e+03 486 2.40970e+03 1.47270e+03 579 2.13230e+03 1.26950e+03
 301 2.48900e+03 1.87910e+03 394 2.40970e+03 1.67590e+03 487 2.44930e+03 1.47270e+03 580 2.17200e+03 1.26950e+03
 302 2.52860e+03 1.87910e+03 395 2.44930e+03 1.67590e+03 488 2.48900e+03 1.47270e+03 581 2.21160e+03 1.26950e+03
 303 2.56820e+03 1.87910e+03 396 2.48900e+03 1.67590e+03 489 2.52860e+03 1.47270e+03 582 2.25120e+03 1.26950e+03
 304 2.60780e+03 1.87910e+03 397 2.52860e+03 1.67590e+03 490 2.56820e+03 1.47270e+03 583 2.29080e+03 1.26950e+03
 305 2.64750e+03 1.87910e+03 398 2.56820e+03 1.67590e+03 491 2.60780e+03 1.47270e+03 584 2.33050e+03 1.26950e+03
 306 8.81600e+02 1.86640e+03 399 2.60780e+03 1.67590e+03 492 2.64750e+03 1.47270e+03 585 2.37010e+03 1.26950e+03
 307 8.81600e+02 1.84100e+03 400 2.64750e+03 1.67590e+03 493 1.05940e+03 1.42190e+03 586 2.40970e+03 1.26950e+03
 308 1.03400e+03 1.82830e+03 401 8.68900e+02 1.66320e+03 494 1.03400e+03 1.42190e+03 587 2.44930e+03 1.26950e+03
 309 1.05940e+03 1.82830e+03 402 1.03400e+03 1.62510e+03 495 8.56200e+02 1.39650e+03 588 2.48900e+03 1.26950e+03
 310 8.81600e+02 1.81560e+03 403 1.05940e+03 1.62510e+03 496 8.30800e+02 1.39650e+03 589 2.52860e+03 1.26950e+03
 311 8.81600e+02 1.77750e+03 404 1.05940e+03 1.57430e+03 497 1.03400e+03 1.37110e+03 590 2.56820e+03 1.26950e+03
 312 1.03400e+03 1.77750e+03 405 1.03400e+03 1.57430e+03 498 1.05940e+03 1.37110e+03 591 2.60780e+03 1.26950e+03
 313 1.05940e+03 1.77750e+03 406 9.83200e+02 1.52350e+03 499 9.07000e+02 1.35840e+03 592 2.64750e+03 1.26950e+03
 314 9.83200e+02 1.72670e+03 407 1.02290e+03 1.52350e+03 500 8.30800e+02 1.35840e+03 593 9.07000e+02 1.25680e+03
 315 1.02290e+03 1.72670e+03 408 1.06250e+03 1.52350e+03 501 8.30800e+02 1.33300e+03 594 8.30800e+02 1.25680e+03
 316 1.06250e+03 1.72670e+03 409 1.10210e+03 1.52350e+03 502 9.07000e+02 1.33300e+03 595 8.30800e+02 1.23140e+03
 317 1.10210e+03 1.72670e+03 410 1.14170e+03 1.52350e+03 503 9.83200e+02 1.32030e+03 596 9.07000e+02 1.23140e+03
 318 1.14170e+03 1.72670e+03 411 1.18140e+03 1.52350e+03 504 1.02290e+03 1.32030e+03 597 1.03400e+03 1.21870e+03
 319 1.18140e+03 1.72670e+03 412 1.22100e+03 1.52350e+03 505 1.06250e+03 1.32030e+03 598 1.05940e+03 1.21870e+03
 320 1.22100e+03 1.72670e+03 413 1.26060e+03 1.52350e+03 506 1.10210e+03 1.32030e+03 599 9.07000e+02 1.20600e+03
 321 1.26060e+03 1.72670e+03 414 1.30020e+03 1.52350e+03 507 1.14170e+03 1.32030e+03 600 8.30800e+02 1.20600e+03
 322 1.30020e+03 1.72670e+03 415 1.33990e+03 1.52350e+03 508 1.18140e+03 1.32030e+03 601 8.30800e+02 1.18060e+03
 323 1.33990e+03 1.72670e+03 416 1.37950e+03 1.52350e+03 509 1.22100e+03 1.32030e+03 602 9.07000e+02 1.18060e+03
 324 1.37950e+03 1.72670e+03 417 1.41910e+03 1.52350e+03 510 1.26060e+03 1.32030e+03 603 1.03400e+03 1.16790e+03
 325 1.41910e+03 1.72670e+03 418 1.45870e+03 1.52350e+03 511 1.30020e+03 1.32030e+03 604 1.05940e+03 1.16790e+03
 326 1.45870e+03 1.72670e+03 419 1.49840e+03 1.52350e+03 512 1.33990e+03 1.32030e+03 605 8.56200e+02 1.14250e+03
 327 1.49840e+03 1.72670e+03 420 1.53800e+03 1.52350e+03 513 1.37950e+03 1.32030e+03 606 8.30800e+02 1.14250e+03
 328 1.53800e+03 1.72670e+03 421 1.57760e+03 1.52350e+03 514 1.41910e+03 1.32030e+03 607 9.83200e+02 1.11710e+03
 329 1.57760e+03 1.72670e+03 422 1.61720e+03 1.52350e+03 515 1.45870e+03 1.32030e+03 608 1.02290e+03 1.11710e+03
 330 1.61720e+03 1.72670e+03 423 1.65690e+03 1.52350e+03 516 1.49840e+03 1.32030e+03 609 1.06250e+03 1.11710e+03
 331 1.65690e+03 1.72670e+03 424 1.69650e+03 1.52350e+03 517 1.53800e+03 1.32030e+03 610 1.10210e+03 1.11710e+03
 332 1.69650e+03 1.72670e+03 425 1.73610e+03 1.52350e+03 518 1.57760e+03 1.32030e+03 611 1.14170e+03 1.11710e+03
 333 1.73610e+03 1.72670e+03 426 1.77570e+03 1.52350e+03 519 1.61720e+03 1.32030e+03 612 1.18140e+03 1.11710e+03
 334 1.77570e+03 1.72670e+03 427 1.81540e+03 1.52350e+03 520 1.65690e+03 1.32030e+03 613 1.22100e+03 1.11710e+03
 335 1.81540e+03 1.72670e+03 428 1.85500e+03 1.52350e+03 521 1.69650e+03 1.32030e+03 614 1.26060e+03 1.11710e+03
 336 1.85500e+03 1.72670e+03 429 1.89460e+03 1.52350e+03 522 1.73610e+03 1.32030e+03 615 1.30020e+03 1.11710e+03
 337 1.89460e+03 1.72670e+03 430 1.93420e+03 1.52350e+03 523 1.77570e+03 1.32030e+03 616 1.33990e+03 1.11710e+03
 338 1.93420e+03 1.72670e+03 431 1.97380e+03 1.52350e+03 524 1.81540e+03 1.32030e+03 617 1.37950e+03 1.11710e+03
 339 1.97380e+03 1.72670e+03 432 2.01350e+03 1.52350e+03 525 1.85500e+03 1.32030e+03 618 1.41910e+03 1.11710e+03
 340 2.01350e+03 1.72670e+03 433 2.05310e+03 1.52350e+03 526 1.89460e+03 1.32030e+03 619 1.45870e+03 1.11710e+03
 341 2.05310e+03 1.72670e+03 434 2.09270e+03 1.52350e+03 527 1.93420e+03 1.32030e+03 620 1.49840e+03 1.11710e+03
 342 2.09270e+03 1.72670e+03 435 2.13230e+03 1.52350e+03 528 1.97380e+03 1.32030e+03 621 1.53800e+03 1.11710e+03
 343 2.13230e+03 1.72670e+03 436 2.17200e+03 1.52350e+03 529 2.01350e+03 1.32030e+03 622 1.57760e+03 1.11710e+03
 344 2.17200e+03 1.72670e+03 437 2.21160e+03 1.52350e+03 530 2.05310e+03 1.32030e+03 623 1.61720e+03 1.11710e+03
 345 2.21160e+03 1.72670e+03 438 2.25120e+03 1.52350e+03 531 2.09270e+03 1.32030e+03 624 1.65690e+03 1.11710e+03
 346 2.25120e+03 1.72670e+03 439 2.29080e+03 1.52350e+03 532 2.13230e+03 1.32030e+03 625 1.69650e+03 1.11710e+03
 347 2.29080e+03 1.72670e+03 440 2.33050e+03 1.52350e+03 533 2.17200e+03 1.32030e+03 626 1.73610e+03 1.11710e+03
 348 2.33050e+03 1.72670e+03 441 2.37010e+03 1.52350e+03 534 2.21160e+03 1.32030e+03 627 1.77570e+03 1.11710e+03
 349 2.37010e+03 1.72670e+03 442 2.40970e+03 1.52350e+03 535 2.25120e+03 1.32030e+03 628 1.81540e+03 1.11710e+03
 350 2.40970e+03 1.72670e+03 443 2.44930e+03 1.52350e+03 536 2.29080e+03 1.32030e+03 629 1.85500e+03 1.11710e+03
 351 2.44930e+03 1.72670e+03 444 2.48900e+03 1.52350e+03 537 2.33050e+03 1.32030e+03 630 1.89460e+03 1.11710e+03
 352 2.48900e+03 1.72670e+03 445 2.52860e+03 1.52350e+03 538 2.37010e+03 1.32030e+03 631 1.93420e+03 1.11710e+03
 353 2.52860e+03 1.72670e+03 446 2.56820e+03 1.52350e+03 539 2.40970e+03 1.32030e+03 632 1.97380e+03 1.11710e+03
 354 2.56820e+03 1.72670e+03 447 2.60780e+03 1.52350e+03 540 2.44930e+03 1.32030e+03 633 2.01350e+03 1.11710e+03
 355 2.60780e+03 1.72670e+03 448 2.64750e+03 1.52350e+03 541 2.48900e+03 1.32030e+03 634 2.05310e+03 1.11710e+03
 356 2.64750e+03 1.72670e+03 449 8.68900e+02 1.48540e+03 542 2.52860e+03 1.32030e+03 635 2.09270e+03 1.11710e+03
 357 8.81600e+02 1.70130e+03 450 9.83200e+02 1.47270e+03 543 2.56820e+03 1.32030e+03 636 2.13230e+03 1.11710e+03
 358 9.83200e+02 1.67590e+03 451 1.02290e+03 1.47270e+03 544 2.60780e+03 1.32030e+03 637 2.17200e+03 1.11710e+03
 359 1.02290e+03 1.67590e+03 452 1.06250e+03 1.47270e+03 545 2.64750e+03 1.32030e+03 638 2.21160e+03 1.11710e+03
 360 1.06250e+03 1.67590e+03 453 1.10210e+03 1.47270e+03 546 9.07000e+02 1.30760e+03 639 2.25120e+03 1.11710e+03
 361 1.10210e+03 1.67590e+03 454 1.14170e+03 1.47270e+03 547 8.30800e+02 1.30760e+03 640 2.29080e+03 1.11710e+03
 362 1.14170e+03 1.67590e+03 455 1.18140e+03 1.47270e+03 548 8.30800e+02 1.28220e+03 641 2.33050e+03 1.11710e+03
 363 1.18140e+03 1.67590e+03 456 1.22100e+03 1.47270e+03 549 9.07000e+02 1.28220e+03 642 2.37010e+03 1.11710e+03
 364 1.22100e+03 1.67590e+03 457 1.26060e+03 1.47270e+03 550 9.83200e+02 1.26950e+03 643 2.40970e+03 1.11710e+03
 365 1.26060e+03 1.67590e+03 458 1.30020e+03 1.47270e+03 551 1.02290e+03 1.26950e+03 644 2.44930e+03 1.11710e+03
 366 1.30020e+03 1.67590e+03 459 1.33990e+03 1.47270e+03 552 1.06250e+03 1.26950e+03 645 2.48900e+03 1.11710e+03
 367 1.33990e+03 1.67590e+03 460 1.37950e+03 1.47270e+03 553 1.10210e+03 1.26950e+03 646 2.52860e+03 1.11710e+03
 368 1.37950e+03 1.67590e+03 461 1.41910e+03 1.47270e+03 554 1.14170e+03 1.26950e+03 647 2.56820e+03 1.11710e+03
 369 1.41910e+03 1.67590e+03 462 1.45870e+03 1.47270e+03 555 1.18140e+03 1.26950e+03 648 2.60780e+03 1.11710e+03
 370 1.45870e+03 1.67590e+03 463 1.49840e+03 1.47270e+03 556 1.22100e+03 1.26950e+03 649 2.64750e+03 1.11710e+03
 371 1.49840e+03 1.67590e+03 464 1.53800e+03 1.47270e+03 557 1.26060e+03 1.26950e+03 650 9.07000e+02 1.10440e+03
 372 1.53800e+03 1.67590e+03 465 1.57760e+03 1.47270e+03 558 1.30020e+03 1.26950e+03 651 8.30800e+02 1.10440e+03
 373 1.57760e+03 1.67590e+03 466 1.61720e+03 1.47270e+03 559 1.33990e+03 1.26950e+03 652 8.30800e+02 1.07900e+03
 374 1.61720e+03 1.67590e+03 467 1.65690e+03 1.47270e+03 560 1.37950e+03 1.26950e+03 653 8.68900e+02 1.07900e+03
 375 1.65690e+03 1.67590e+03 468 1.69650e+03 1.47270e+03 561 1.41910e+03 1.26950e+03 654 9.07000e+02 1.07900e+03
 376 1.69650e+03 1.67590e+03 469 1.73610e+03 1.47270e+03 562 1.45870e+03 1.26950e+03 655 9.83200e+02 1.06630e+03
 377 1.73610e+03 1.67590e+03 470 1.77570e+03 1.47270e+03 563 1.49840e+03 1.26950e+03 656 1.02290e+03 1.06630e+03
 378 1.77570e+03 1.67590e+03 471 1.81540e+03 1.47270e+03 564 1.53800e+03 1.26950e+03 657 1.06250e+03 1.06630e+03
 379 1.81540e+03 1.67590e+03 472 1.85500e+03 1.47270e+03 565 1.57760e+03 1.26950e+03 658 1.10210e+03 1.06630e+03
 380 1.85500e+03 1.67590e+03 473 1.89460e+03 1.47270e+03 566 1.61720e+03 1.26950e+03 659 1.14170e+03 1.06630e+03
 381 1.89460e+03 1.67590e+03 474 1.93420e+03 1.47270e+03 567 1.65690e+03 1.26950e+03 660 1.18140e+03 1.06630e+03
 382 1.93420e+03 1.67590e+03 475 1.97380e+03 1.47270e+03 568 1.69650e+03 1.26950e+03 661 1.22100e+03 1.06630e+03
 383 1.97380e+03 1.67590e+03 476 2.01350e+03 1.47270e+03 569 1.73610e+03 1.26950e+03 662 1.26060e+03 1.06630e+03
 384 2.01350e+03 1.67590e+03 477 2.05310e+03 1.47270e+03 570 1.77570e+03 1.26950e+03 663 1.30020e+03 1.06630e+03
 385 2.05310e+03 1.67590e+03 478 2.09270e+03 1.47270e+03 571 1.81540e+03 1.26950e+03 664 1.33990e+03 1.06630e+03

1037 8.30800e+02 2.84430e+03
 1038 9.07000e+02 2.84430e+03
 1039 9.07000e+02 2.88240e+03
 1040 8.56200e+02 2.88240e+03
 1041 8.30800e+02 2.88240e+03
 1042 9.83200e+02 2.89510e+03
 1043 1.02290e+03 2.89510e+03
 1044 1.06250e+03 2.89510e+03
 1045 1.10210e+03 2.89510e+03
 1046 1.14170e+03 2.89510e+03
 1047 1.18140e+03 2.89510e+03
 1048 1.22100e+03 2.89510e+03
 1049 1.26060e+03 2.89510e+03
 1050 1.30020e+03 2.89510e+03
 1051 1.33990e+03 2.89510e+03
 1052 1.37950e+03 2.89510e+03
 1053 1.41910e+03 2.89510e+03
 1054 1.45870e+03 2.89510e+03
 1055 1.49840e+03 2.89510e+03
 1056 1.53800e+03 2.89510e+03
 1057 1.57760e+03 2.89510e+03
 1058 1.61720e+03 2.89510e+03
 1059 1.65690e+03 2.89510e+03
 1060 1.69650e+03 2.89510e+03
 1061 1.73610e+03 2.89510e+03
 1062 1.77570e+03 2.89510e+03
 1063 1.81540e+03 2.89510e+03
 1064 1.85500e+03 2.89510e+03
 1065 1.89460e+03 2.89510e+03
 1066 1.93420e+03 2.89510e+03
 1067 1.97380e+03 2.89510e+03
 1068 2.01350e+03 2.89510e+03
 1069 2.05310e+03 2.89510e+03
 1070 2.09270e+03 2.89510e+03
 1071 2.13230e+03 2.89510e+03
 1072 2.17200e+03 2.89510e+03
 1073 2.21160e+03 2.89510e+03
 1074 2.25120e+03 2.89510e+03
 1075 2.29080e+03 2.89510e+03
 1076 2.33050e+03 2.89510e+03
 1077 2.37010e+03 2.89510e+03
 1078 2.40970e+03 2.89510e+03
 1079 2.44930e+03 2.89510e+03
 1080 2.48900e+03 2.89510e+03
 1081 2.52860e+03 2.89510e+03
 1082 2.56820e+03 2.89510e+03
 1083 2.60780e+03 2.89510e+03
 1084 2.64750e+03 2.89510e+03
 1085 8.43500e+02 2.92050e+03
 1086 8.43500e+02 2.94590e+03
 1087 9.83200e+02 2.94590e+03
 1088 1.02290e+03 2.94590e+03
 1089 1.06250e+03 2.94590e+03
 1090 1.10210e+03 2.94590e+03
 1091 1.14170e+03 2.94590e+03
 1092 1.18140e+03 2.94590e+03
 1093 1.22100e+03 2.94590e+03
 1094 1.26060e+03 2.94590e+03
 1095 1.30020e+03 2.94590e+03
 1096 1.33990e+03 2.94590e+03
 1097 1.37950e+03 2.94590e+03
 1098 1.41910e+03 2.94590e+03
 1099 1.45870e+03 2.94590e+03
 1100 1.49840e+03 2.94590e+03
 1101 1.53800e+03 2.94590e+03
 1102 1.57760e+03 2.94590e+03
 1103 1.61720e+03 2.94590e+03
 1104 1.65690e+03 2.94590e+03
 1105 1.69650e+03 2.94590e+03
 1106 1.73610e+03 2.94590e+03
 1107 1.77570e+03 2.94590e+03
 1108 1.81540e+03 2.94590e+03
 1109 1.85500e+03 2.94590e+03
 1110 1.89460e+03 2.94590e+03
 1111 1.93420e+03 2.94590e+03
 1112 1.97380e+03 2.94590e+03
 1113 2.01350e+03 2.94590e+03
 1114 2.05310e+03 2.94590e+03
 1115 2.09270e+03 2.94590e+03
 1116 2.13230e+03 2.94590e+03
 1117 2.17200e+03 2.94590e+03
 1118 2.21160e+03 2.94590e+03
 1119 2.25120e+03 2.94590e+03
 1120 2.29080e+03 2.94590e+03
 1121 2.33050e+03 2.94590e+03
 1122 2.37010e+03 2.94590e+03
 1123 2.40970e+03 2.94590e+03
 1124 2.44930e+03 2.94590e+03
 1125 2.48900e+03 2.94590e+03
 1126 2.52860e+03 2.94590e+03
 1127 2.56820e+03 2.94590e+03
 1128 2.60780e+03 2.94590e+03
 1129 2.64750e+03 2.94590e+03
 1130 9.07000e+02 2.95860e+03
 1131 8.43500e+02 2.97130e+03
 1132 9.07000e+02 2.98400e+03
 1133 9.32400e+02 2.98400e+03
 1134 8.43500e+02 2.99670e+03
 1135 1.03400e+03 2.99670e+03
 1136 1.05940e+03 2.99670e+03
 1137 9.32400e+02 3.02210e+03
 1138 9.07000e+02 3.02210e+03
 1139 8.43500e+02 3.02210e+03
 1140 8.43500e+02 3.04750e+03
 1141 8.43500e+02 3.07290e+03
 1142 8.94300e+02 3.07290e+03
 1143 9.19700e+02 3.07290e+03
 1144 8.43500e+02 3.09830e+03
 1145 9.83200e+02 3.09830e+03
 1146 1.02290e+03 3.09830e+03
 1147 1.06250e+03 3.09830e+03
 1148 1.10210e+03 3.09830e+03
 1149 1.14170e+03 3.09830e+03
 1150 1.18140e+03 3.09830e+03
 1151 1.22100e+03 3.09830e+03
 1152 1.26060e+03 3.09830e+03
 1153 1.30020e+03 3.09830e+03
 1154 1.33990e+03 3.09830e+03
 1155 1.37950e+03 3.09830e+03
 1156 1.41910e+03 3.09830e+03
 1157 1.45870e+03 3.09830e+03
 1158 1.49840e+03 3.09830e+03
 1159 1.53800e+03 3.09830e+03
 1160 1.57760e+03 3.09830e+03
 1161 1.61720e+03 3.09830e+03
 1162 1.65690e+03 3.09830e+03
 1163 1.69650e+03 3.09830e+03
 1164 1.73610e+03 3.09830e+03
 1165 1.77570e+03 3.09830e+03
 1166 1.81540e+03 3.09830e+03
 1167 1.85500e+03 3.09830e+03
 1168 1.89460e+03 3.09830e+03
 1169 1.93420e+03 3.09830e+03
 1170 1.97380e+03 3.09830e+03
 1171 2.01350e+03 3.09830e+03
 1172 2.05310e+03 3.09830e+03
 1173 2.09270e+03 3.09830e+03
 1174 2.13230e+03 3.09830e+03
 1175 2.17200e+03 3.09830e+03
 1176 2.21160e+03 3.09830e+03
 1177 2.25120e+03 3.09830e+03
 1178 2.29080e+03 3.09830e+03
 1179 2.33050e+03 3.09830e+03
 1180 2.37010e+03 3.09830e+03
 1181 2.40970e+03 3.09830e+03
 1182 2.44930e+03 3.09830e+03
 1183 2.48900e+03 3.09830e+03
 1184 2.52860e+03 3.09830e+03
 1185 2.56820e+03 3.09830e+03
 1186 2.60780e+03 3.09830e+03
 1187 2.64750e+03 3.09830e+03
 1188 7.92700e+02 3.14270e+03
 1189 7.41900e+02 3.14270e+03
 1190 6.91100e+02 3.14270e+03
 1191 8.94300e+02 3.14910e+03
 1192 9.19700e+02 3.14910e+03
 1193 9.83200e+02 3.14910e+03
 1194 1.02290e+03 3.14910e+03
 1195 1.06250e+03 3.14910e+03
 1196 1.10210e+03 3.14910e+03
 1197 1.14170e+03 3.14910e+03
 1198 1.18140e+03 3.14910e+03
 1199 1.22100e+03 3.14910e+03
 1200 1.26060e+03 3.14910e+03
 1201 1.30020e+03 3.14910e+03
 1202 1.33990e+03 3.14910e+03
 1203 1.37950e+03 3.14910e+03
 1204 1.41910e+03 3.14910e+03
 1205 1.45870e+03 3.14910e+03
 1206 1.49840e+03 3.14910e+03
 1207 1.53800e+03 3.14910e+03
 1208 1.57760e+03 3.14910e+03
 1209 1.61720e+03 3.14910e+03
 1210 1.65690e+03 3.14910e+03
 1211 1.69650e+03 3.14910e+03
 1212 1.73610e+03 3.14910e+03
 1213 1.77570e+03 3.14910e+03
 1214 1.81540e+03 3.14910e+03
 1215 1.85500e+03 3.14910e+03
 1216 1.89460e+03 3.14910e+03
 1217 1.93420e+03 3.14910e+03
 1218 1.97380e+03 3.14910e+03
 1219 2.01350e+03 3.14910e+03
 1220 2.05310e+03 3.14910e+03
 1221 2.09270e+03 3.14910e+03
 1222 2.13230e+03 3.14910e+03
 1223 2.17200e+03 3.14910e+03
 1224 2.21160e+03 3.14910e+03
 1225 2.25120e+03 3.14910e+03
 1226 2.29080e+03 3.14910e+03
 1227 2.33050e+03 3.14910e+03
 1228 2.37010e+03 3.14910e+03
 1229 2.40970e+03 3.14910e+03
 1230 2.44930e+03 3.14910e+03
 1231 2.48900e+03 3.14910e+03
 1232 2.52860e+03 3.14910e+03
 1233 2.56820e+03 3.14910e+03
 1234 2.60780e+03 3.14910e+03
 1235 2.64750e+03 3.14910e+03
 1236 1.44040e+03 3.18720e+03
 1237 1.40230e+03 3.18720e+03
 1238 1.35150e+03 3.18720e+03
 1239 1.31340e+03 3.18720e+03
 1240 9.19700e+02 3.18720e+03
 1241 8.94300e+02 3.18720e+03
 1242 7.92700e+02 3.19350e+03
 1243 6.91100e+02 3.19350e+03
 1244 1.50390e+03 3.19990e+03
 1245 1.57990e+03 3.19990e+03
 1246 1.14830e+03 3.22530e+03
 1247 1.12290e+03 3.22530e+03
 1248 1.09750e+03 3.22530e+03
 1249 1.07210e+03 3.22530e+03
 1250 1.04670e+03 3.22530e+03
 1251 1.02130e+03 3.22530e+03
 1252 9.95900e+02 3.22530e+03
 1253 9.70500e+02 3.22530e+03
 1254 9.72700e+02 3.24430e+03
 1255 7.41900e+02 3.24430e+03
 1256 6.91100e+02 3.24430e+03
 1257 1.07210e+03 3.25070e+03
 1258 1.09750e+03 3.25070e+03
 1259 1.12290e+03 3.25070e+03
 1260 1.14830e+03 3.25070e+03
 1261 1.17370e+03 3.25070e+03
 1262 1.19910e+03 3.25070e+03
 1263 1.22450e+03 3.25070e+03
 1264 1.24990e+03 3.25070e+03
 1265 1.27530e+03 3.25070e+03
 1266 1.50390e+03 3.25070e+03
 1267 1.52910e+03 3.25070e+03
 1268 1.55450e+03 3.25070e+03
 1269 1.57990e+03 3.25070e+03
 1270 1.60530e+03 3.25070e+03
 1271 1.63070e+03 3.25070e+03
 1272 1.65610e+03 3.25070e+03
 1273 1.68150e+03 3.25070e+03
 1274 1.71960e+03 3.25070e+03
 1275 1.74500e+03 3.25070e+03
 1276 1.77040e+03 3.25070e+03
 1277 1.79580e+03 3.25070e+03
 1278 1.82120e+03 3.25070e+03
 1279 1.84660e+03 3.25070e+03
 1280 1.87200e+03 3.25070e+03
 1281 1.89740e+03 3.25070e+03
 1282 1.92280e+03 3.25070e+03
 1283 1.96090e+03 3.25070e+03
 1284 1.98630e+03 3.25070e+03
 1285 2.01170e+03 3.25070e+03
 1286 2.03710e+03 3.25070e+03
 1287 2.06250e+03 3.25070e+03
 1288 2.08790e+03 3.25070e+03
 1289 2.11330e+03 3.25070e+03
 1290 2.13870e+03 3.25070e+03
 1291 2.16410e+03 3.25070e+03
 1292 2.22760e+03 3.25070e+03
 1293 2.25300e+03 3.25070e+03
 1294 2.27840e+03 3.25070e+03
 1295 2.30380e+03 3.25070e+03
 1296 2.32920e+03 3.25070e+03
 1297 2.35460e+03 3.25070e+03
 1298 2.38000e+03 3.25070e+03
 1299 2.40540e+03 3.25070e+03
 1300 2.43080e+03 3.25070e+03
 1301 2.46890e+03 3.25070e+03
 1302 2.49430e+03 3.25070e+03
 1303 2.51970e+03 3.25070e+03
 1304 2.54510e+03 3.25070e+03
 1305 2.57050e+03 3.25070e+03
 1306 2.59590e+03 3.25070e+03
 1307 2.62130e+03 3.25070e+03
 1308 2.64670e+03 3.25070e+03
 1309 2.67210e+03 3.25070e+03
 1310 1.70050e+03 3.25700e+03
 1311 9.45100e+02 3.26340e+03
 1312 1.31340e+03 3.26340e+03
 1313 1.35150e+03 3.26340e+03
 1314 1.40230e+03 3.26340e+03
 1315 1.44040e+03 3.26340e+03
 1316 2.69750e+03 3.26340e+03
 1317 2.43080e+03 8.37700e+02
 1318 2.97690e+03 8.63100e+02
 1319 3.00230e+03 8.63100e+02
 1320 3.02270e+03 8.63100e+02
 1321 3.05310e+03 8.63100e+02
 1322 3.07850e+03 8.63100e+02
 1323 3.10390e+03 8.63100e+02
 1324 3.12930e+03 8.63100e+02
 1325 3.15470e+03 8.63100e+02
 1326 3.18010e+03 8.63100e+02
 1327 3.20550e+03 8.63100e+02
 1328 3.23090e+03 8.63100e+02
 1329 3.25630e+03 8.63100e+02
 1330 3.28170e+03 8.63100e+02
 1331 3.30710e+03 8.63100e+02
 1332 3.33250e+03 8.63100e+02
 1333 3.35790e+03 8.63100e+02
 1334 3.38330e+03 8.63100e+02
 1335 3.40870e+03 8.63100e+02
 1336 3.43410e+03 8.63100e+02
 1337 3.45950e+03 8.63100e+02
 1338 3.48490e+03 8.63100e+02
 1339 3.51030e+03 8.63100e+02
 1340 3.53570e+03 8.63100e+02
 1341 3.56110e+03 8.63100e+02
 1342 3.58650e+03 8.63100e+02
 1343 3.61190e+03 8.63100e+02
 1344 3.63730e+03 8.63100e+02
 1345 3.66270e+03 8.63100e+02
 1346 3.68810e+03 8.63100e+02
 1347 3.71350e+03 8.63100e+02
 1348 3.73890e+03 9.13900e+02
 1349 3.76430e+03 9.13900e+02
 1350 3.66270e+03 9.13900e+02
 1351 3.63730e+03 9.13900e+02
 1352 3.61190e+03 9.13900e+02
 1353 3.58650e+03 9.13900e+02
 1354 3.56110e+03 9.13900e+02
 1355 3.53570e+03 9.13900e+02
 1356 3.51030e+03 9.13900e+02
 1357 3.48490e+03 9.13900e+02
 1358 3.45950e+03 9.13900e+02
 1359 3.43410e+03 9.13900e+02
 1360 3.40870e+03 9.13900e+02
 1361 3.38330e+03 9.13900e+02
 1362 3.35790e+03 9.13900e+02
 1363 3.33250e+03 9.13900e+02
 1364 3.30710e+03 9.13900e+02
 1365 3.28170e+03 9.13900e+02
 1366 3.25630e+03 9.13900e+02
 1367 3.23090e+03 9.13900e+02
 1368 3.20550e+03 9.13900e+02
 1369 3.18010e+03 9.13900e+02
 1370 3.15470e+03 9.13900e+02
 1371 3.12930e+03 9.13900e+02
 1372 3.10390e+03 9.13900e+02
 1373 3.07850e+03 9.13900e+02
 1374 3.05310e+03 9.13900e+02
 1375 3.02770e+03 9.13900e+02
 1376 3.00230e+03 9.13900e+02
 1377 2.97690e+03 9.13900e+02
 1378 2.97690e+03 1.06630e+03
 1379 3.00230e+03 1.06630e+03
 1380 3.02270e+03 1.06630e+03
 1381 3.05310e+03 1.06630e+03
 1382 3.07850e+03 1.06630e+03
 1383 3.10390e+03 1.06630e+03
 1384 3.12930e+03 1.06630e+03
 1385 3.15470e+03 1.06630e+03
 1386 3.18010e+03 1.06630e+03
 1387 3.20550e+03 1.06630e+03
 1388 3.23090e+03 1.06630e+03
 1389 3.25630e+03 1.06630e+03
 1390 3.28170e+03 1.06630e+03
 1391 3.30710e+03 1.06630e+03
 1392 3.33250e+03 1.06630e+03
 1393 3.35790e+03 1.06630e+03
 1394 3.38330e+03 1.06630e+03
 1395 3.40870e+03 1.06630e+03
 1396 3.43410e+03 1.06630e+03
 1397 3.45950e+03 1.06630e+03
 1398 3.48490e+03 1.06630e+03
 1399 3.51030e+03 1.06630e+03
 1400 3.53570e+03 1.06630e+03
 1401 3.56110e+03 1.06630e+03
 1402 3.58650e+03 1.06630e+03
 1403 3.61190e+03 1.06630e+03
 1404 3.63730e+03 1.06630e+03
 1405 3.66270e+03 1.06630e+03
 1406 3.68810e+03 1.06630e+03
 1407 3.71350e+03 1.06630e+03
 1408 3.73890e+03 1.11710e+03

