

UNIVERSIDADE FEDERAL FLUMINENSE

IVAN CÉSAR MARTINS

Contribuições para o Problema de Formação de Células
de Manufatura

NITERÓI

2012

UNIVERSIDADE FEDERAL FLUMINENSE

IVAN CÉSAR MARTINS

Contribuições para o Problema de Formação de Células de Manufatura

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização.

Orientador:
Fábio Protti

NITERÓI
2012

Contribuições para o Problema de Formação de Células de Manufatura

Ivan César Martins

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização.

Aprovada por:

Prof. Fábio Protti / IC-UFF (Presidente)

Prof. Luiz Satoru Ochi / IC-UFF

Prof. Lucídio dos Anjos Formiga Cabral / DI-UFPB

Prof. Márcia Helena Costa Fampa / IM-UFRJ

Niterói, 31 de agosto de 2012 .

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

M386 Martins, Ivan César

Contribuições para o problema de formação de células de
manufatura / Ivan César Martins. – Niterói, RJ : [s.n.], 2012.
79 f.

Dissertação (Mestrado em Computação) - Universidade Federal
Fluminense, 2012.

Orientador: Fábio Protti.

1. Metaheurística. 2. Manufatura celular. 3. Tecnologia de grupo.
I. Título.

CDD 005.136

Agradecimentos

A minha família, pelas palavras de conforto e incentivo em todos os momentos da minha vida. Em especial minha querida mãe, Ruth, pelo amor incondicional e a Nice, pelas suas orações.

Aos meus grandes amigos, André, Bárbara e Elka, sempre cheios de otimismo e alegria.

A minha tia Amélia e vó Yara, sempre preocupados com os meus estudos e meu bem-estar.

Aos meus amigos: Rian, Paulo, Uéverton, Phillipe e Manu, pela convivência e apoio.

Aos professores Luiz Satoru e Fábio Protti, pelos ensinamentos, conselhos e pelas oportunidades durante o Mestrado.

Ao professor Alejandro Frery, por ser um exemplo de pesquisador e me iniciar no ambiente acadêmico.

Aos colegas do IC-UFF pela ideias e conversas descontraídas.

Aos amigos de Maceió.

A CAPES.

Ivan César Martins

Resumo

O PROBLEMA DE FORMAÇÃO DE CÉLULAS DE MANUFATURA (PFCM) é um problema de otimização \mathcal{NP} -difícil que consiste em agrupar um conjunto de máquinas em células, dedicadas à produção de uma família de partes de produtos, minimizando os deslocamentos de materiais e a sub-utilização de máquinas. Estes agrupamentos parte-máquina contribuem para uma redução nos custos e um aumento na eficiência do sistema de manufatura de uma indústria. Neste trabalho, o problema é tratado de forma heurística, considerando diferentes restrições sobre o tamanho mínimo das células. Para tal, são propostos algoritmos baseados nas meta-heurísticas GRASP e ILS, ambas fazendo uso do método VND com uma ordenação aleatória de vizinhança na fase de busca local. Os algoritmos propostos são comparados com os procedimentos existentes na literatura, apresentando resultados relevantes em um tempo computacional razoável. O valor ótimo é alcançado na maioria das instâncias, com gap médio abaixo de 0,25%. Os experimentos computacionais foram realizados em 35 instâncias clássicas da literatura com até 40 máquinas e 100 partes.

Palavras-chave: Meta-heurísticas. Manufatura Celular. Tecnologia de Grupo

Abstract

The CELL FORMATION PROBLEM (CFP) is a \mathcal{NP} -hard optimization problem which consists of grouping a set of machines into cells, dedicated to the production of a family of parts, minimizing intercell moves and cell machine subutilization. These machine-part groups aim at reducing costs and increase the efficiency of a cellular manufacturing system. In this work, a heuristic approach is used, treating different constraints about minimum size of cells. We propose some algorithms based on GRASP and ILS metaheuristics, both using a VND method with a random ordering of neighborhoods in local search phase. The proposed algorithms are compared with some available algorithms in the literature, presenting meaningful results in a reasonable computational time. Optimum is reached in most instances, with an average gap below 0.25%. Computational experiments were performed in 35 literature instances with up to 40 machines and 100 parts.

Keywords: Metaheuristics. Cellular Manufacturing. Group Technology.

Sumário

Acrônimos	vii
Lista de Figuras	viii
Lista de Tabelas	ix
1 Introdução	1
1.1 Sistemas de Manufatura	2
1.1.1 Posicional	2
1.1.2 Funcional	3
1.1.3 Linear	4
1.1.4 Celular	4
1.2 Meta-heurísticas	6
1.3 Motivação e Objetivos	6
1.4 Estrutura do Trabalho	7
2 O Problema de Formação de Células de Manufatura	8
2.1 Definição	8
2.2 Características	9
2.2.1 Número de Soluções	9
2.2.2 Tamanho da Célula	10
2.3 Formulação Matemática	12
2.3.1 Formação de Células de Manufatura Clássica	13
2.3.2 Formação de Células de Manufatura Generalizada	14
2.4 Medidas de Similaridade	17
2.4.1 Coeficiente de Jaccard	18
2.4.2 Distância Euclidiana e Manhattan	18
2.5 Medidas de Desempenho	20
2.5.1 Eficiência de Agrupamento	20
2.5.2 Eficácia de Agrupamento	21
2.5.3 Índice de Capacidade de Agrupamento	21
2.5.4 Medida Primária	22
2.5.5 Índice de Célula	22
2.6 Considerações sobre as Medidas	23
3 Abordagens Existentes para o PFCM	24
3.1 Procedimentos Descritivos	24
3.2 Clusterização	25
3.2.1 Clusterização Baseada em Matrizes	25

3.2.2	Clusterização Hierárquica	26
3.2.3	Clusterização Não-hierárquica	26
3.3	Teoria dos Grafos	27
3.4	Meta-heurísticas	28
3.4.1	Simulated Annealing	28
3.4.2	Algoritmos Genéticos	28
3.4.3	Busca Tabu	29
3.5	Outras Abordagens	29
3.5.1	Redes Neurais	30
3.5.2	Lógica Fuzzy	30
4	Proposta	31
4.1	Algoritmo GRASP-RVND	31
4.1.1	Método Construtivo	33
4.1.2	Busca Local	33
4.2	Algoritmo ILS-RVND	37
4.2.1	Perturbações	38
5	Experimentos Computacionais	41
5.1	Instâncias	41
5.2	Teste de Desempenho	43
5.3	Comparação com os Métodos da Literatura	46
6	Conclusões e Trabalhos Futuros	50
A	Instâncias	51
B	Erros encontrados na literatura do PFCM	60
B.1	Instâncias Erradas	60
B.2	Resultados Incorretos	60
	Bibliografia	68

Acrônimos

GRASP	:	Greedy Randomized Adaptive Search Procedure
ILS	:	Iterated Local Search
RVND	:	Random Variable Neighborhood Descent
VND	:	Variable Neighborhood Descent
CFP	:	CELL FORMATION PROBLEM
DM	:	Deslocamento de Máquinas
PFCM	:	PROBLEMA DE FORMAÇÃO DE CÉLULAS DE MANUFATURA
TM	:	Troca de Máquinas

Lista de Figuras

1.1	Sistema de Manufatura Posicional	2
1.2	Sistema de Manufatura Funcional	3
1.3	Sistema de Manufatura Linear	4
1.4	Sistema de Manufatura Celular	5
2.1	Representação matricial do problema	9
2.2	Solução do problema permitindo células <i>singleton</i>	11
2.3	Solução do problema não permitindo células <i>singleton</i>	12
4.1	Processamento inicial da entrada	32
4.2	Deslocamento da máquina 2 para a célula 1	35
4.3	Troca de células entre as máquinas 2 e 5	36
4.4	Remoção da célula 2, distribuindo-se suas máquinas	39
4.5	Divisão da célula 1, em que M_5 formou uma nova célula	40

Lista de Tabelas

2.1	Número de soluções	10
5.1	Instâncias da Literatura do PFCM	42
5.2	Teste de desempenho para o PFCM-CRS	44
5.3	Teste de desempenho para o PFCM-SRS	45
5.4	Resultado geral do PFCM-CRS	47
5.5	Resultado geral do PFCM-SRS	49

Capítulo 1

Introdução

Na indústria, a organização dos recursos de produção afeta diretamente os custos de produção e a produtividade. Uma alocação de recursos racional, definida na fase de planejamento do sistema de manufatura, contribui significativamente para o aumento na eficiência das operações envolvidas no processo de fabricação e na consequente redução dos custos referentes à movimentação destes recursos.

A forma como os recursos são organizados podem ser classificados de acordo com o fluxo do produto na linha de produção: posicional, linear, funcional e celular [71]. Cada tipo apresenta um conjunto de vantagens e limitações referentes ao tempo de processamento, flexibilidade do produto e da demanda, utilização de maquinário e de operadores, especialização dos operadores e custos de manutenção.

A *manufatura celular* é uma aplicação do conceito de *Tecnologia de Grupo* que propõe identificar e agrupar partes similares para tomar vantagem dessa similaridade no processo de manufatura. Este conceito foi proposto originalmente por Flanders [23], mas somente a partir do trabalho de Mitrofanov [46] foi formalizado e amplamente difundido.

Uma das primeiras técnicas para a criação de um sistema de manufatura celular foi proposta por Burbidge [9], denominada *Production Flow Analysis* (PFA). Desde então, o modelo de manufatura celular vem sendo utilizado como uma estratégia eficiente quando há quantidade e variedade média de produtos, unindo as vantagens da manufatura funcional e linear.

No entanto, o alto grau de complexidade do problema e a crescente necessidade em se obter melhores resultados em tempo computacional razoável tem levado os pesquisadores a concentrar os esforços na busca de métodos eficientes, mas que não necessariamente garantam a otimalidade da solução, incorporando mecanismos de escape de ótimos locais, conhecidos como meta-heurísticas.

1.1 Sistemas de Manufatura

Um sistema de manufatura é composto por métodos de organização da produção conhecidos como *layouts*. Cada *layout* possui uma característica específica que o permite ser aplicada em diferentes situações, a depender das características do produto e do dimensionamento da produção. A seguir, serão detalhados os principais tipos.

1.1.1 Posicional

Este era o principal método no sistema artesanal de trabalho. O produto era iniciado e acabado no local de trabalho do artífice, que o construía inteiramente. Com a evolução da indústria, este tipo se tornou menos importante, e atualmente a sua aplicação se restringe, principalmente, a casos onde o material, ou o componente principal, é difícil de ser movimentado, sendo mais fácil transportar equipamentos e operadores.

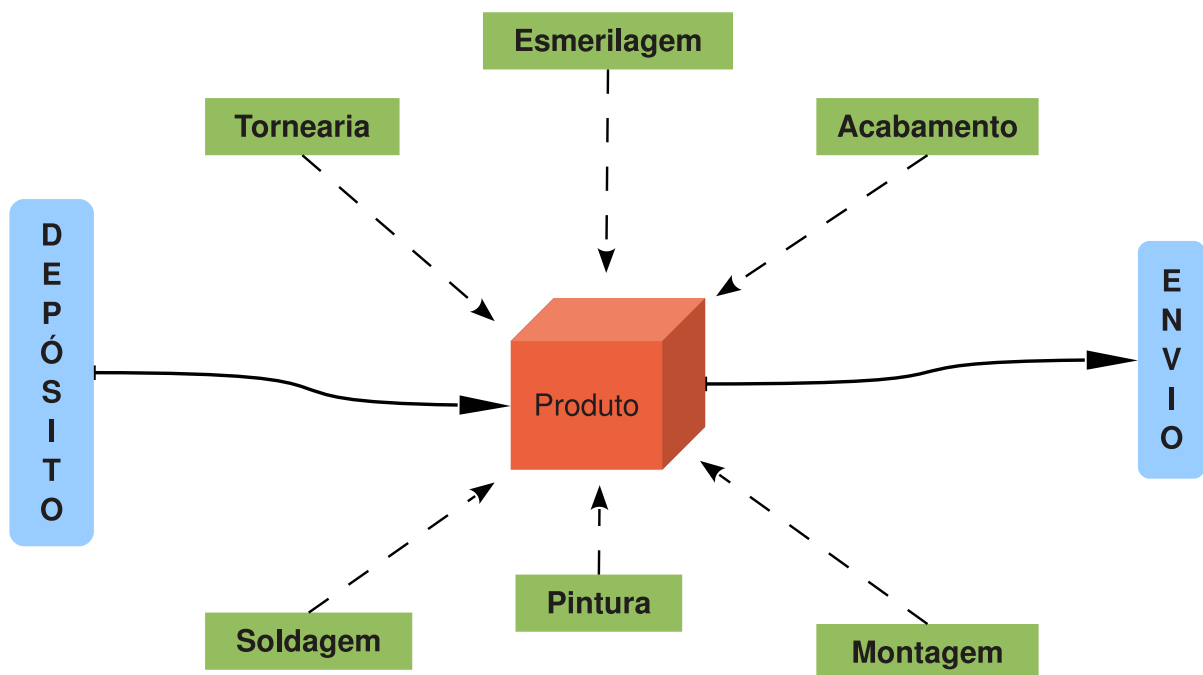


Figura 1.1: Sistema de Manufatura Posicional

descrição: o material permanece parado, enquanto que operadores, equipamentos e todas as outras partes se movimentam à sua volta.

uso: produção de produtos volumosos e em quantidades reduzidas, priorizando a redução na movimentação de material em relação à de operadores e equipamentos.

vantagem: flexibilidade quanto a mudanças no produto.

desvantagens: requer um alto nível de habilidade dos operadores, controle e sincronização da produção.

1.1.2 Funcional

De modo geral, tem por objetivo agrupar máquinas com funcionalidades similares, de acordo com o tipo de processo que executam. A manufatura funcional possui uma grande flexibilidade quanto a mudanças nos produtos processados, sendo usualmente aplicada em fábricas de processamento intermitente, produção por encomendas de produtos com pequeno volume de produção ou de baixa padronização.

Partes fabricadas neste modelo usualmente requerem diferentes operações e sequências, fazendo com que o tempo necessário para cada operação varie significativamente. Algo em torno de 95% do tempo é gasto em atividades de espera para processamento. Os 5% restantes são divididos entre preparação e processamento [2].

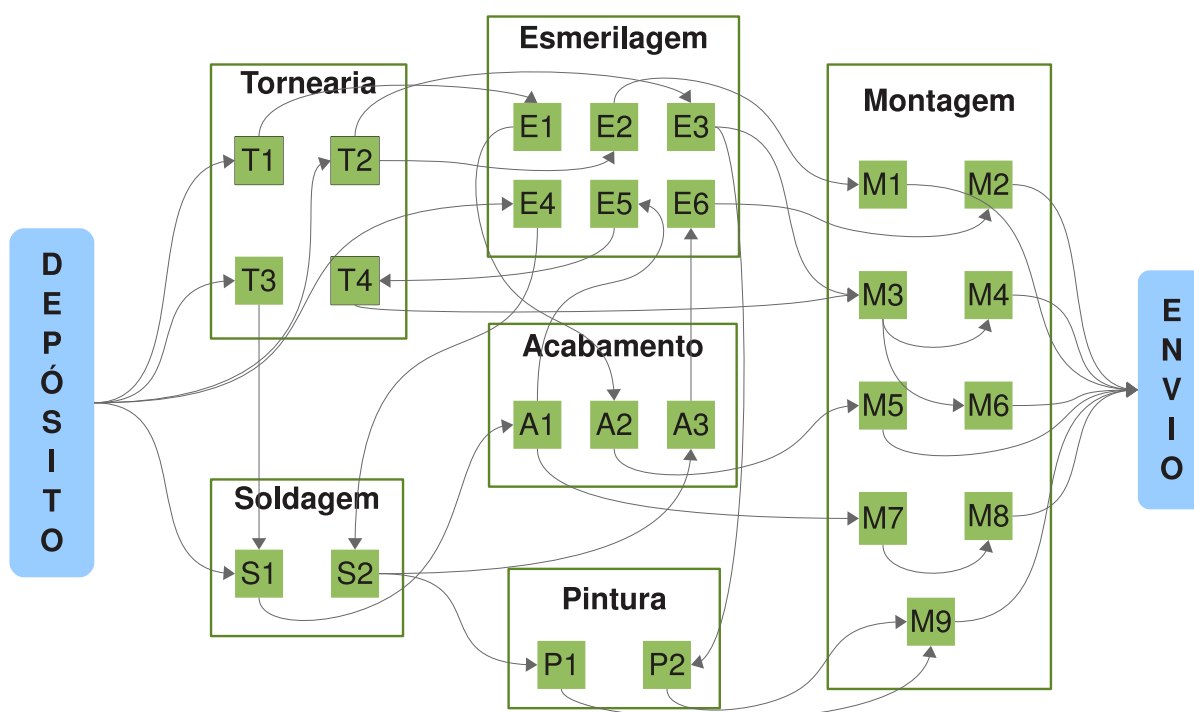


Figura 1.2: Sistema de Manufatura Funcional

descrição: as operações com os materiais são agrupadas de acordo com a semelhança no tipo de processo de produção, independentemente do produto processado.

uso: quando há uma grande variedade de produtos, priorizando a redução na movimentação de operadores e equipamentos em relação à de materiais.

vantagem: melhor utilização das máquinas, especialização no trabalho dos operadores e controle de produção descentralizado.

desvantagens: aumento no tempo de produção e na quantidade de material em processo de fabricação.

1.1.3 Linear

A disposição dos postos de trabalho obedece a sequência do processamento do produto, formando agrupamentos de equipamentos e atividades distintas entre si, mantendo em comum o processamento sequencial. Geralmente, há um uso de máquinas especializadas em cada processo da sequência, tornando-os mais caros. Tal investimento somente é justificado quando um grande volume de determinada parte precisa ser produzida. Entretanto, a maior limitação deste modelo é a falta de flexibilidade na manufatura de partes que não foram previamente designadas, uma vez que estas máquinas são projetadas para executarem operações limitadas e não permitem reconfiguração.

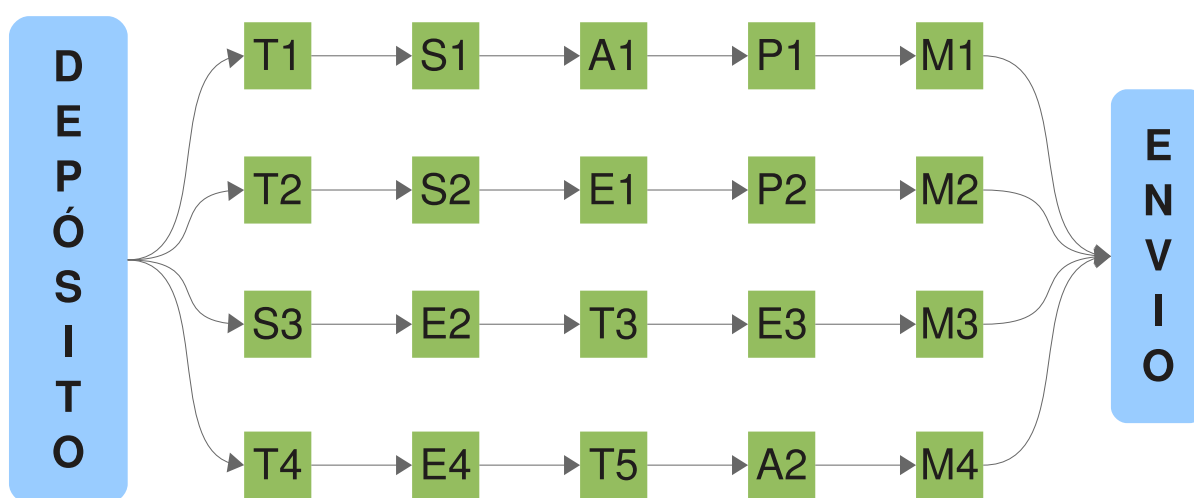


Figura 1.3: Sistema de Manufatura Linear

descrição: os equipamentos são dispostos de acordo com uma determinada sequência de operações, permanecendo fixos enquanto os materiais se movem pelos equipamentos.

uso: produção em grandes quantidades de um único tipo de produto.

vantagem: controle simples de produção e minimização no manuseio de material.

desvantagens: se uma máquina parar toda a linha de produção é interrompida e o posto de trabalho mais lento determina o ritmo da linha de produção.

1.1.4 Celular

O principal objetivo da manufatura celular é a de unir as vantagens da manufatura funcional e linear de modo a reduzir a complexidade no controle de produção. De acordo com Kusiak [38], uma redução de 20% a 88% nos esforços relacionados à manipulação de materiais podem ser alcançadas. As máquinas são dispostas em proximidade umas com as outras e se dedicam à manufatura de uma família de partes, promovendo um fluxo eficiente

aliado a alta taxa de produção, de modo similar à manufatura linear. Além disso, o uso de máquinas de propósito geral e equipamentos na manufatura celular permite grande flexibilidade e produção de uma variedade de partes.

Em resumo, uma célula representa um sistema de produção híbrido, um misto de um departamento responsável pela produção de uma grande variedade de partes e para criação em massa de um produto. Um sistema de manufatura celular pode então produzir partes em média quantidade e variedade mais economicamente que qualquer outro tipo. Esta abordagem tem sido aplicada em uma variedade de indústrias nas últimas três décadas, tais como indústrias de maquinário e ferramentas, aeroespacial e defesa, automotiva e elétrica.

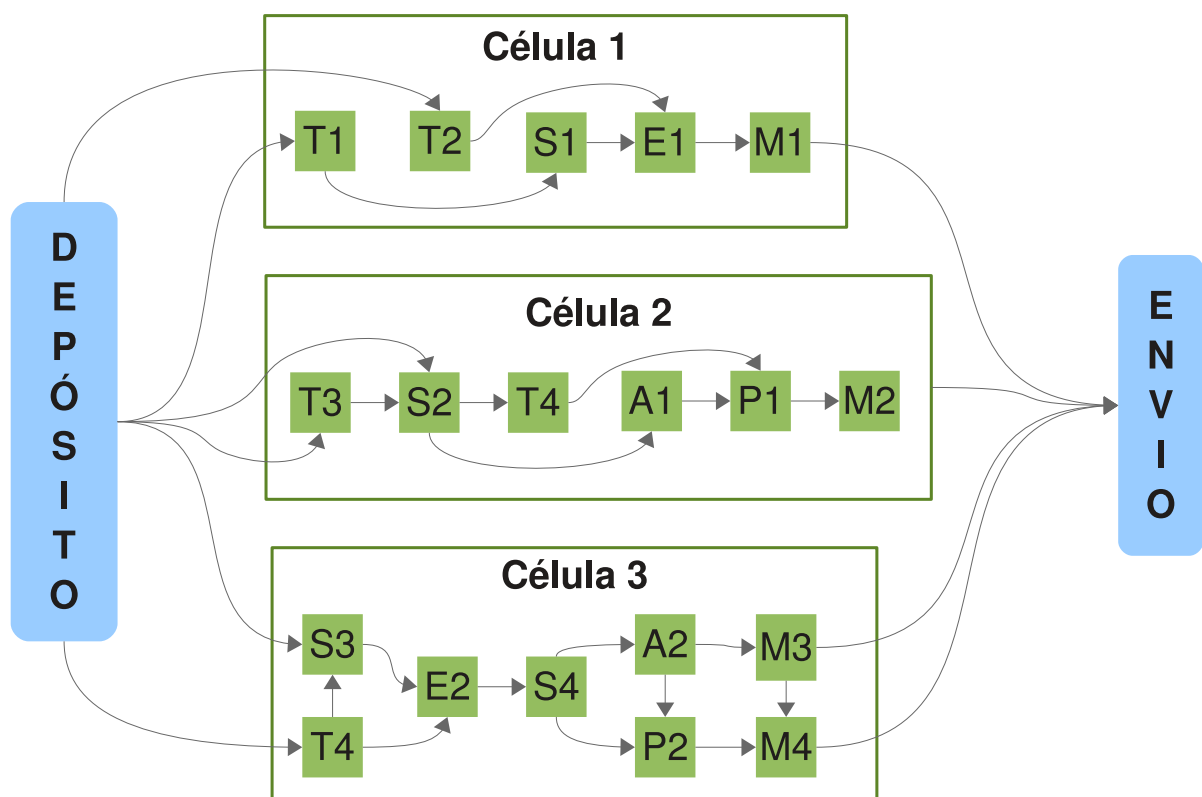


Figura 1.4: Sistema de Manufatura Celular

Vantagens:

- redução do tempo necessário para preparar a máquina para funcionamento
- redução no inventário de materiais em produção
- redução do tempo necessário para a parte ser fabricada
- redução nos custos de transporte e manipulação (carga, descarga, paletização, despaletização) das matérias-primas
- agenda de atividades simplificada

- simplificação do fluxo de produção
- melhora na qualidade dos produtos

Desvantagens:

- alto custo de implementação
- dificuldade em contornar casos de quebra, sobrecarga e balanceamento
- dificuldade em gerenciar as operações inter-células
- requer sincronização das partes na subsequente montagem

1.2 Meta-heurísticas

As meta-heurísticas são procedimentos empregados na busca de boas soluções, eventualmente a ótima, a partir da aplicação de uma heurística subordinada, modelada para cada problema específico. Ao contrário das heurísticas convencionais, as meta-heurísticas são de caráter geral e providas de mecanismos de escape de ótimos locais ainda distantes dos ótimos globais, diferenciando-se basicamente em como este escape é feito.

Elas podem ser divididas em duas categorias, de acordo com o princípio usado para explorar o espaço de soluções: busca local e busca populacional. Nas meta-heurísticas baseadas em busca local, a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança. Busca Tabu, *Simulated Annealing*, *Variable Neighborhood Search*, GRASP e ILS são exemplos de métodos que se enquadram nesta categoria. Os métodos baseados em busca populacional, por sua vez, consistem em manter um conjunto de boas soluções e combiná-las de forma a tentar produzir soluções ainda melhores. Exemplos clássicos de procedimentos desta categoria são os Algoritmos Genéticos, Algoritmos Meméticos e o Algoritmo Colônia de Formigas.

O PFCM é um problema de otimização \mathcal{NP} -difícil. Portanto, algoritmos que garantam a otimalidade da solução possuem um tempo computacional proibitivo [27]. As meta-heurísticas têm surgido como uma alternativa eficiente na resolução de problemas de otimização, produzindo soluções ótimas ou quase-ótimas em um tempo computacional razoável.

1.3 Motivação e Objetivos

- revisar o PFCM, apresentando o problema e descrevendo as metodologias existentes na literatura;

- desenvolver métodos eficientes para a resolução do problema, com considerações sobre a presença de *singleton*;
- avaliar o desempenho das meta-heurísticas GRASP e ILS empregando o método RVND aplicado ao PFCM;
- padronizar a comparação entre os resultados computacionais da literatura quanto ao uso da restrição de *singleton* na resolução do PFCM.

1.4 Estrutura do Trabalho

Este trabalho está dividido em seis capítulos, organizados da seguinte maneira:

Capítulo 2: define e explica o PFCM;

Capítulo 3: apresenta uma revisão dos métodos empregados na literatura para a resolução do problema;

Capítulo 4: proposta dos algoritmos desenvolvidos, apresentando as técnicas empregadas;

Capítulo 5: fornece uma comparação entre os algoritmos propostos e os métodos da literatura;

Capítulo 6: contém as conclusões deste trabalho e projeções para trabalhos futuros.

Capítulo 2

O Problema de Formação de Células de Manufatura

Neste capítulo, descrevemos o PFCM em detalhes, apresentando os conceitos básicos necessários para sua compreensão. Primeiramente, definimos o problema e mostramos suas características, justificando as abordagens empregadas na resolução. Mostramos uma formulação matemática para o problema clássico, apresentando suas limitações, e uma formulação generalizada que supre essas limitações. Em seguida, mostramos os métodos de avaliação da solução através das medidas de similaridade e de desempenho. Por fim, fazemos uma avaliação desses métodos, e justificamos o uso da medida eficácia de agrupamento.

2.1 Definição

Um sistema de manufatura pode ser representado como uma matriz binária parte–máquina $A_{|P|\times|M|}$, em que P é o conjunto de partes e M o de máquinas que compõem este sistema. Um elemento a_{ij} que vale 1 indica que é realizada uma operação na parte $i \in P$ pela máquina $j \in M$; $a_{ij} = 0$ indica o caso contrário.

O PFCM tem por objetivo formar células, nas quais são agrupadas máquinas utilizadas na produção de uma família de partes, de forma a minimizar o número de movimentação de partes entre as células e a subutilização das máquinas nas células. Cada família é composta por partes, que de preferência possuam um alto nível de similaridade entre si, ou seja, um grande número de máquinas em comum que realizam operações nestas partes.

As células podem ser formadas através de um rearranjo (permutação) de linhas e colunas da matriz, de modo a formar blocos célula–máquina e família–parte ao longo da diagonal da matriz. Para caracterizar uma solução, definem-se algumas estruturas presentes na matriz parte–máquina:

Bloco: é uma submatriz da matriz parte–máquina de entrada que representa uma célula

de manufatura.

Operação: elemento com valor 1 que aparece dentro de um bloco. Este elemento representa a operação de uma parte por uma determinada máquina.

Lacuna: elemento com valor 0 que aparece dentro de um bloco. Este elemento corresponde a um ponto de subutilização da máquina na célula por não operar a parte designada.

Exceção: elemento com valor 1 que aparece fora dos blocos. Este elemento representa a operação de uma parte em máquinas presentes em células distintas, e por isso precisa ser movida para processamento.

Na Figura 2.1, temos um exemplo da representação matricial de um sistema de manufatura, com $|P| = 7$ e $|M| = 5$. Nas Figuras 2.2 e 2.3, temos exemplos de formações para o sistema da Figura 2.1 com diferentes restrições para o tamanho mínimo da célula, explicado em detalhes na Seção 2.2.2.

	M_1	M_2	M_3	M_4	M_5
P_1	0	1	1	0	1
P_2	1	0	0	1	0
P_3	0	1	1	0	0
P_4	1	0	0	1	0
P_5	1	0	0	0	1
P_6	1	0	1	1	0
P_7	0	0	1	0	1

Figura 2.1: Representação matricial do problema

Em um sistema de manufatura celular perfeito, cada uma das máquinas destinada a uma célula opera todas as partes designadas à célula, e apenas estas. Com esta formação, as peças não precisam ser deslocadas para outras células, e as máquinas não são subutilizadas. Apesar de esta ser a formação ideal, uma solução desta forma é difícil de ser encontrada na prática.

2.2 Características

2.2.1 Número de Soluções

Podemos visualizar o PFCM como um problema de clusterização, no qual precisamos agrupar as máquinas em células e as partes em famílias. De acordo com Wang [75], o número de maneiras de se agrupar n dados (máquinas e partes) em c clusters mutuamente exclusivos e não vazios é determinado pelo número de Stirling, dado pela equação:

$$\sum_{i=1}^c (-1)^{c-i} \cdot \frac{i^n}{[i! \cdot (c-i)!]} \quad (2.1)$$

Desta forma, o número de soluções possíveis para o PFCM, ou seja, o número de maneiras nas quais podemos agrupar p partes e m máquinas em c clusters mutuamente exclusivos e não vazios, considerando que cada célula pode ser associada com qualquer família, é:

$$\sum_{i=1}^c (-1)^{c-i} \cdot \frac{i^p}{[i! \cdot (c-i)!]} \cdot \sum_{i=1}^c (-1)^{c-i} \cdot \frac{i^m}{[i! \cdot (c-i)!]} \quad (2.2)$$

A Tabela 2.1 ilustra a explosão do número de soluções em função do número m de máquinas e p de partes, para 3, 4 e 5 clusters.

Tabela 2.1: Número de soluções

m	p	$c = 3$	$c = 4$	$c = 5$
10	10	$5.417058 \cdot 10^{12}$	$1.542641 \cdot 10^{15}$	$3.185999 \cdot 10^{16}$
20	35	$4.841430 \cdot 10^{24}$	$2.224650 \cdot 10^{30}$	$1.813379 \cdot 10^{34}$
30	90	$4.991617 \cdot 10^{55}$	$3.065252 \cdot 10^{69}$	$5.192093 \cdot 10^{79}$
40	100	$1.740496 \cdot 10^{65}$	$3.372553 \cdot 10^{81}$	$4.979083 \cdot 10^{93}$

2.2.2 Tamanho da Célula

O tamanho da célula é normalmente expresso pela quantidade de máquinas que a compõem. Um valor máximo é definido quando se está preocupado com o espaço disponível para a disposição das máquinas ou por questões de controle dos operadores e operações. Logo, além de saber quais máquinas processam quais partes, mais informações são necessárias para a tomada de decisão.

Na literatura do problema não existe um consenso na determinação de um valor mínimo para o tamanho da célula, já que não consiste de um aspecto operacional, e sim um problema de definição. O termo *singleton* é definido como uma célula que possua menos que duas máquinas ou partes [27, 77]. Existem abordagens que consideram ou não a presença de *singleton*. Dentre os que permitem, há trabalhos como o de Pailla et al. [52], que permite a existência de células vazias na solução, ou seja, máquinas ou partes que não pertençam a nenhuma célula, e os que permitem apenas células unitárias [30].

Restringir o tamanho da célula torna o problema mais realista, porém diminui a qualidade da solução. Trabalhos em que *singletons* não são permitidos possuem soluções piores do que aqueles que a permitem. Por este motivo, a comparação entre esses algoritmos

não é justa. Como em Wu et al. [77], neste trabalho faremos uma distinção entre as abordagens que permitem e as que não permitem a presença de *singletons*. Quando a presença de *singleton* é permitida, aceitamos soluções com células vazias, uma vez que não contraria a definição.

A Figura 2.2 consiste de uma solução para a matriz parte-máquina da Figura 2.1 em que *singleton* são permitidos. Observe a presença dos blocos diagonais representando a formação das células do sistema de manufatura. Nesta figura teremos 3 células, dispostas da seguinte maneira:

Célula 1: máquinas M_2 e M_3 , que operam a família de partes $\{P_1, P_3\}$. A máquina M_2 está perfeitamente alocada nesta célula pois opera todas as partes associadas e não há nenhuma operação em partes associadas a outras células. Já a máquina M_3 , apesar de operar todas as partes associadas, opera a parte P_6 que está na célula 2 e a parte P_7 que está na célula 3; logo, estas partes terão que ser movidas para serem operadas nesta célula.

Célula 2: máquinas M_1 e M_4 , que operam a família de partes $\{P_2, P_4, P_6\}$. A máquina M_4 está perfeitamente alocada na célula; no entanto, a máquina M_1 , apesar de operar todas as partes associadas, opera a parte P_5 que está na célula 3.

Célula 3: máquina M_5 , que opera a família de partes $\{P_5, P_7\}$. A máquina M_5 , apesar de operar todas as partes associadas, opera a parte P_1 que está na célula 1.

	M_2	M_3	M_1	M_4	M_5
P_1	1	1	0	0	1
P_3	1	1	0	0	0
P_2	0	0	1	1	0
P_4	0	0	1	1	0
P_6	0	1	1	1	0
P_5	0	0	1	0	1
P_7	0	1	0	0	1

Figura 2.2: Solução do problema permitindo células *singleton*.

Observe que, nesta formação, a célula 3 possui apenas 1 máquina (a máquina M_5); portanto, dizemos que esta é uma célula *singleton*. A Figura 2.3 consiste de uma solução para a matriz parte-máquina da Figura 2.1, em que células *singleton* não são permitidas. Nesta figura teremos 2 células, dispostas da seguinte maneira:

Célula 1: máquinas M_2 , M_3 e M_5 , que operam a família de partes $\{P_1, P_3, P_7\}$. A máquina M_2 não opera nenhuma parte associada à outra célula, porém está subutilizada

por não operar a parte P_7 associada. Por outro lado, a máquina M_3 opera todas as partes associadas, mas opera a parte P_6 que está na célula 2, e terá que ser movida para ser operada nesta célula. Por fim, temos a máquina M_5 , que não opera todas as parte associadas (já que não opera a parte P_3), e opera a parte P_5 que está na célula 2.

Célula 2: máquinas M_1 e M_4 , que operam a família de partes $\{P_2, P_4, P_5, P_6\}$. Tanto a máquinas M_1 quanto a M_4 não operam partes associadas a outra célula; no entanto, a máquina M_4 está subutilizada na célula por não operar a parte P_5 associada.

	M_2	M_3	M_5	M_1	M_4
P_1	1	1	1	0	0
P_3	1	1	0	0	0
P_7	0	1	1	0	0
P_2	0	0	0	1	1
P_4	0	0	0	1	1
P_5	0	0	1	1	0
P_6	0	1	0	1	1

Figura 2.3: Solução do problema não permitindo células *singleton*

Vale ressaltar que temos uma formação ideal quando nenhuma máquina é subutilizada na célula e não é necessário deslocar partes entre células. No entanto, na primeira formação (Figura 2.2) temos 4 movimentações de partes entre células. Como veremos na Seção 2.5, podemos medir o desempenho de uma solução através do índice de Eficácia de Agrupamento. Nesta formação, este índice é de 75,00%. Já na segunda formação (figura 2.3) temos 3 subutilizações de máquinas e 2 movimentações de partes entre células, com um índice de eficácia de 73,68%.

Como foi dito anteriormente, apesar de fazer mais sentido uma formação sem células *singleton* na prática, esta possui uma medida de eficácia pior. No entanto, se penalizarmos mais a movimentação de partes entre células, neste exemplo, a formação sem células *singleton* pode ficar pior, pois houve 2 movimentações a mais de partes entre células. Por este e outros motivos, definir uma boa métrica para a avaliação da qualidade da solução é importante. Na Seção 2.5 serão apresentadas e discutidas com mais detalhes diversas medidas utilizadas na literatura.

2.3 Formulação Matemática

Diversas formulações matemáticas têm sido propostas nas ultimas décadas por um grande número de pesquisadores. A maioria destas são identificadas e discutidas em Selim

et al. [63] e Papaioannou & Wilson [53]. A seguir, discutiremos a formulação clássica do problema, que considera apenas as operações parte-máquina, e uma generalização que inclui diversos outros fatores de grande importância prática.

2.3.1 Formação de Células de Manufatura Clássica

Em um nível conceitual, a modelagem das células de manufatura considera apenas as operações das máquinas sobre as partes. A princípio, diversos fatores práticos do processo de manufatura não são levados em consideração. Boctor [7] apresenta a primeira formulação matemática para o PFCM que permite o controle sobre o tamanho da célula:

$$\min \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^P a_{ij} \cdot \sum_{k=1}^N |x_{ik} - y_{jk}| \quad (2.3)$$

$$s.a \quad \sum_{k=1}^N x_{ik} = 1 \quad \forall i \quad (2.4)$$

$$\sum_{j=1}^P y_{jk} = 1 \quad \forall j \quad (2.5)$$

$$\sum_{i=1}^M x_{ik} \leq m \quad \forall k \quad (2.6)$$

onde:

i - índice de máquinas

j - índice de partes

k - índice de células

m - número máximo permitido de máquinas em cada célula

N - número de células de manufatura

M - número de máquinas

P - número total de partes

$x_{i,j}$ - variável binária que indica se a máquina i está associada à célula j

$y_{j,k}$ - variável binária que indica se a parte j está associada à célula k

A função objetivo minimiza o número exceções, ou seja, o número de operações $a_{ij} = 1$ em que a máquina i e a parte j não estão associadas à mesma célula k ; com isso, $|x_{ik} - y_{jk}| = 1$ e uma unidade é contabilizada. Observe que, neste caso, o custo será

contabilizado duas vezes, uma quando $x_{ik} = 1$ e $y_{jk} = 0$, e outra quando $x_{ik} = 0$ e $y_{jk} = 1$. As restrições 2.4 e 2.5, garantem que cada máquina ou parte está associada a apenas uma célula. Já a restrição 2.6 previne que em cada célula mais máquinas que o permitido m sejam associadas.

Partindo desta formulação, diversas outras começaram a considerar mais critérios sobre a qualidade da solução. De imediato, poderíamos considerar a minimização do número de lacunas, acrescentando uma parcela $\frac{1}{N} \sum_{i=1}^M \sum_{j=1}^P (1 - a_{ij}) \cdot \sum_{k=1}^N (1 - |x_{ik} - y_{jk}|)$ à função objetivo, de forma a contabilizar a não operação de uma parte j por uma máquina i ($a_{ij} = 0$) quando estão associadas à mesma célula. Observe que, quando isto ocorre, uma unidade de custo é contabilizada N vezes, uma para cada $1 \leq k \leq N$, já que quando $x_{ik} = y_{jk} = 1$ para um determinado k , teremos $x_{ik} = y_{jk} = 0$ para os demais, e portanto $\sum_{k=1}^N (1 - |x_{ik} - y_{jk}|) = N$.

2.3.2 Formação de Células de Manufatura Generalizada

Nas últimas décadas, muitos pesquisadores tem reconhecido a necessidade de se considerar os diversos fatores práticos relacionados ao problema. Dado um conjunto de partes e máquinas, para cada parte diferentes tipos de operações são requisitados, assim como o tempo de preparo e processamento de cada operação. Cada máquina tem uma determinada capacidade de produção e há diversos custos envolvidos nas operações executadas por estas máquinas. As ferramentas estão intimamente relacionadas com os diferentes tipos de operações. A sequência de operações pode ou não ter restrições de precedência.

Em Selim et al. [63] é apresentada uma formulação matemática considerando todos esses fatores apresentados. Devido à grande complexidade combinatória do problema e sua natureza multiobjetivo, o modelo apresentado está longe de ser resolvido para quaisquer problemas reais. No entanto, a partir deste, formulações menores (que incorporem apenas alguns dos objetivos e restrições) tem sido propostas por diversos pesquisadores [22, 59, 17, 73, 24]. Segue a formulação em [63]:

$$\min \sum_{m=1}^M \sum_{c=1}^C F_m \cdot y_{mc} + \sum_{m=1}^M C_m \sum_{c=1}^C y_{mc} \sum_{p=1}^P \sum_{j=1}^J E_{j_p m} \cdot D_p \cdot x_{j_p c} \quad (2.7)$$

$$+ H \sum_{p=1}^P \sum_{j=1}^J \sum_{c=1}^C D_p \cdot v_{j_p c} + \sum_{m=1}^M \sum_{c=1}^C R_m \cdot o_{mc}$$

$$s.a \sum_{c=1}^C x_{j_p c} = 1 \quad \forall j, p \quad (2.8)$$

$$\sum_{p=1}^P \sum_{j=1}^J E_{j_p m} \cdot D_p \cdot x_{j_p c} \leq B_m \cdot I \cdot y_{mc} \quad \forall m, c \quad (2.9)$$

$$x_{jpc} - x_{j_{p-1},c} = v_{jpc} \quad \forall j_p > 1, p, c \quad (2.10)$$

$$\sum_{m=1}^M y_{mc} \leq S_c \quad \forall c \quad (2.11)$$

$$o_{mc} + \sum_{w=1}^W A_{mw} \cdot u_{wc} \geq \theta y_{mc} \quad \forall m, c \quad (2.12)$$

$$\sum_{c=1}^C u_{wc} = 1 \quad \forall w \quad (2.13)$$

$$\sum_{w=1}^W Q_w \cdot u_{wc} \geq \sum_{m=1}^M G_m \sum_{p=1}^P \sum_{j=1}^J D_p \cdot E_{jpm} \cdot x_{jpc} \quad \forall c \quad (2.14)$$

$$y_{mc} \in \mathbb{Z}, u_{wc}, x_{jpc} \in \{0, 1\} \quad (2.15)$$

índices:

- c - células ($c = 1, \dots, C$)
- j - tipos de operações ($j = 1, \dots, J$)
- t - tipos de ferramentas ($t = 1, \dots, T$)
- m - tipos de máquinas ($m = 1, \dots, M$)
- p - partes ($p = 1, \dots, P$)
- w - trabalhadores ($w = 1, \dots, W$)

parâmetros:

- A_{mw} - 1 se o trabalhador w pode operar a máquina m ; 0 caso contrário
- B_m - proporção de tempo que uma máquina do tipo m está disponível
- C_m - custo variável por unidade de tempo para operar uma máquina do tipo m
- D_p - demanda média da parte p (em unidades) por período
- F_m - custo amortizado por período para obter uma máquina do tipo m
- G_m - proporção de atenção requerida pelo operador enquanto a máquina m está em operação
- H - custo de manipulação de matéria-prima inter-células
- R_m - custo para treinar um trabalhador para operar uma máquina do tipo m
- E_{jpm} - unidade de tempo de processamento para execução de uma operação j_p em uma máquina do tipo m
- I - tempo total de trabalho por período
- S_c - número máximo de máquinas permitido na célula c

Q_w - disponibilidade do trabalhador w em cada período

variáveis:

o_{mc} - número de trabalhadores treinados para operar uma máquina do tipo m na célula c

y_{mc} - número de máquinas do tipo m na célula c

u_{wc} - vale 1 no caso do trabalhador w estar associado à célula c ; 0 caso contrário

x_{jpc} - vale 1 no caso da operação j na parte p estar associada à célula c ; 0 caso contrário

v_{jpc} - vale 1 no caso da j -ésima operação da parte p for executada em uma célula diferente que a operação anterior; 0 caso contrário

k_{mwc} - quantidade de tempo que o operador w está associado a máquinas do tipo m na célula c

A função objetivo minimiza a soma dos custos de compra das máquinas, custos variáveis de uso das máquinas, custo de manipulação de materiais e treinamento dos trabalhadores por período. No entanto, o custo sobre as ferramentas somente precisa ser considerada se está associada com as operações e cada ferramenta deverá estar presente na célula que executa a operação.

Segue abaixo uma breve explicação sobre as restrições desta formulação:

Restrição 2.8: força a associação de cada operação e de cada parte a uma única célula.

Restrição 2.9: garante que cada célula tenha uma alocação adequada de máquinas de cada tipo para a execução da carga de trabalho destinada. O lado esquerdo da equação acumula o carga de trabalho total para as máquinas do tipo m na célula, enquanto que o lado direito determina o tempo disponível nessas máquinas em cada período, baseado nas máquinas associadas.

Restrição 2.10: determina as movimentações intra-células. As variáveis v_{jpc} serão 1 se a j -ésima operação da parte p é executada em uma célula diferente que a operação anterior. Se isso ocorrer, é acrescentado um custo h na função objetivo para cada carregamento da parte p por período. Se a sequência de operações não está fixada, este conjunto de restrições pode ser substituído por um que conte o número de células em que partes do tipo p estão associadas em uma ou mais operações.

Restrição 2.11: limita o número de máquinas em cada célula.

Restrição 2.12: garante que pelo menos θ trabalhadores estão treinados para operarem as máquinas em cada célula.

Restrição 2.13: associa cada trabalhador a exatamente um grupo.

Restrição 2.14: garante que em cada célula haja força de trabalho suficiente para suprir a carga de trabalho.

A formulação acima assume o conhecimento prévio dos tipos de máquinas a serem usadas em cada operação. Se as máquinas são flexíveis, ou seja, podem assumir mais de um tipo, as variáveis x_{jpc} podem ser expandidas para incluir um índice da máquina, e o modelo será capaz de fazer a seleção das máquinas correspondentes. Neste caso, os parâmetros E_{jpc} representarão tempos de processamento condicionais.

Algumas questões, no entanto, ainda precisam ser consideradas, a fim de prover um modelo final para o problema. A disposição dos processos são insensíveis às múltiplas demandas de determinado produto e mudanças na concepção do produto geralmente não causam maiores problemas, desde que a demanda total por um processo não varie. Contudo, na manufatura celular, cada produto representa uma grande proporção da capacidade da célula, e assim máquinas podem estar sobrecarregadas em uma célula e paradas em outra.

O efeito é similar ao apresentado na teoria das filas, onde é bem conhecido que o agrupamento de servidores em uma fila em comum é mais efetivo do que em filas separadas. A ideia por trás da manufatura celular, no entanto, consiste em separar a fila de tal forma que as vantagens em processamento, planejamento e transporte compensem a perda de eficiência. O modelo final precisa considerar a flexibilidade de mudanças a nível de demanda, roteamento, projeto e variedade dos produtos.

Além disso, o modelo acima não considera explicitamente algumas vantagens na associação de partes em células. Frequentemente, uma família de partes possui uma certa semelhança no preparo da operação a ser efetuada. Por exemplo, um torno pode conter todas as ferramentas de corte necessárias para fazer todas as partes em uma família. Partes com formas similares podem conter um componente genérico que necessite apenas de alguns ajustes individuais na preparação. Na usinagem de peças, o tempo para a troca de ferramentas de corte pode ser eliminado se a próxima parte apresenta dimensões e características em comum com a parte anterior.

Para tal, uma possível adição seria definir um tempo de preparação por período para cada operação associada a uma célula. Um fator para o tempo de preparação também pode ser adicionado nas restrições 2.9 e 2.14. No entanto, se em uma célula todas as partes utilizam uma determinada operação, este custo só precisa ser considerado uma única vez.

2.4 Medidas de Similaridade

As medidas de similaridade têm por objetivo calcular o quanto as máquinas ou partes são similares. Duas partes são similares se possuem um mesmo conjunto de máquinas

que participem de sua fabricação. De forma análoga acontece com máquinas similares, ou seja, máquinas que participam na fabricação do mesmo conjunto de partes. Yin & Yasuda [81] fornecem uma ampla revisão sobre as medidas de similaridade aplicadas ao PFCM. A seguir, comentaremos sobre as três medidas mais utilizadas: o coeficiente de Jaccard, a distância Euclidiana e a distância Manhattan.

2.4.1 Coeficiente de Jaccard

Em geral, o **Coeficiente de Jaccard** mede a similaridade entre conjuntos. É definido como a razão entre o tamanho da interseção e o tamanho da união desses conjuntos. Quando aplicado entre o conjunto de operações de duas máquinas, fornece um índice baseado na razão entre o número de partes atendidas por **ambas** as máquinas e o número de partes atendidas por **pelo menos** uma delas.

McAuley [43] foi o primeiro autor a aplicar o Coeficiente de Jaccard em seu método (*Single Linkage Cluster Analysis*). Neste contexto, o Coeficiente de Jaccard J_{ij} entre as máquinas i e j é definido como:

$$J_{ij} = \frac{P_{ij}}{P_i + P_j - P_{ij}} \quad (2.16)$$

onde:

P_{ij} - número de partes processadas por ambas as máquinas

P_i - número de partes processadas pela máquina i

P_j - número de partes processadas pela máquina j

A similaridade entre duas máquinas será máxima ($J_{ij} = 1$) quando todas as partes atendidas por elas forem as mesmas ($P_{ij} = P_i = P_j$), e a similaridade será igual a 0 quando as duas máquinas não atenderem nenhuma parte em comum ($P_{ij} = 0$).

Por exemplo, considere matriz da Figura 2.1. Calculemos o coeficiente de Jaccard para as máquinas 1 e 4 e para as máquinas 1 e 3. Para as máquinas 1 e 4, teremos $P_{1,4} = 3$, $P_1 = 4$ e $P_4 = 3$, então $J_{1,4} = 0,75$. Para as máquinas 1 e 3, teremos $P_{1,3} = 1$, $P_1 = 4$ e $P_3 = 3$, então $J_{1,3} = 0,14$. Logo, as máquinas 1 e 4 são mais similares entre si do que as máquinas 1 e 3.

2.4.2 Distância Euclidiana e Manhattan

Enquanto que o Coeficiente de Jaccard é uma medida de similaridade, a **Distância Euclidiana** e a **Distância Manhattan** são medidas de dissimilaridade. Medidas de dissimilaridade expressam um sentido oposto quando aplicadas ao PFCM, já que as máquinas (ou partes) serão mais similares quanto menor for a distância. Vemos a aplicação

da Distância Manhattan em diversos trabalhos, tais como Chandrasekharan & Rajagopalan [14], Srinivasan [67], Cheng et al. [16]. Quanto à Distância Euclidiana, podemos citar Kulkarni & Kiang [34], Masnata & Settineri [42] e Ben-Arieh & Sreenivasan [6].

Neste contexto, a Distância Euclidiana entre duas máquinas i e j é definida como:

$$E_{ij} = \sqrt{\sum_{k=1}^P |x_{ik} - x_{jk}|^2} \quad (2.17)$$

Já a Distância Manhattan entre duas máquinas i e j é definida como:

$$M_{ij} = \sum_{k=1}^P |x_{ik} - x_{jk}| \quad (2.18)$$

onde:

P - número de partes

x_{ik} - 1 se a máquina i atende a parte k ; 0 caso contrário

x_{jk} - 1 se a máquina j atende a parte k ; 0 caso contrário

Vemos que $|x_{ik} - x_{jk}| = 0$ caso ambas as máquinas i e j atendam, ou não, a parte k , e $|x_{ik} - x_{jk}| = 1$ quando apenas uma das máquinas atende a parte k . A distância será mínima quando o valor de $|x_{ik} - x_{jk}|$ for 0 para todas as partes $k \in P$, ou seja:

- todas as partes atendidas pela máquina i também são pela máquina j , ou vice-versa; com isso $x_{ik} = x_{jk} = 1$
- todas as partes que não são atendidas pela máquina i também não são pela máquina j , ou vice-versa; com isso $x_{ik} = x_{jk} = 0$

De forma análoga podemos aplicar essa ideia para calcular a distância entre partes. Observe também que, como estamos trabalhando com variáveis binárias, $E_{ij} = \sqrt{\sum_{k=1}^P |x_{ik} - x_{jk}|^2} = \sqrt{\sum_{k=1}^P |x_{ik} - x_{jk}|}$, e portanto $M_{ij} = E_{ij}^2$.

Como no exemplo da seção anterior, calculemos a Distância Euclidiana e Manhattan entre as máquinas 1 e 4 e entre as máquinas 1 e 3. Para as máquinas 1 e 3, teremos que $M_{1,3} = |0-1|+|1-0|+|0-1|+|1-0|+|1-0|+|1-1|+|0-1| = 6$ e $E_{1,3} = 6^2 = 36$. Para as máquinas 1 e 4, teremos que $M_{1,4} = |0-0|+|1-1|+|0-0|+|1-1|+|1-0|+|1-1|+|0-0| = 1$ e $E_{1,4} = 1^2 = 1$. Logo, a dissimilaridade entre as máquinas 1 e 3 é maior que entre as máquinas 1 e 4, uma vez que existem 5 partes a mais em que ocorre o fato de apenas uma das máquinas realizar uma operação nela.

2.5 Medidas de Desempenho

Uma solução do PFCM é composta por células de máquinas e famílias de partes, agrupadas de modo a permitir o melhor desempenho do sistema de manufatura. As medidas de desempenho têm por objetivo avaliar a qualidade da solução obtida. Com base nisso, definimos as seguintes notações, que serão utilizadas nas medidas de desempenho:

	Número de 1's	Número de 0's
Dentro dos blocos	e_1^{in}	e_0^{in}
Fora dos blocos	e_1^{out}	e_0^{out}
Total na matriz	e_1	e_0

Uma revisão das medidas sobre diferentes parâmetros de avaliação é apresentada em Sarker & Khan [60]. Nas seções a seguir, as medidas mais conhecidas são apresentadas e utilizadas no cálculo da qualidade das soluções apresentadas nas Figuras 2.2 e 2.3. A tabela abaixo contém a contabilização dos termos utilizados pelas medidas.

	Figura 2.2	Figura 2.3
e_1^{in}	12	14
e_0^{in}	0	3
e_1^{out}	4	2
e_0^{out}	19	16
e_1	16	16
e_0	19	19

2.5.1 Eficiência de Agrupamento

Em Chandrasekharan & Rajagopalan [13] foi proposta pela primeira vez uma medida para avaliar a qualidade de uma solução. A **Eficiência de Agrupamento** η é definido por:

$$\eta = q\eta_1 + (1 - q)\eta_2 \quad (2.19)$$

onde:

- η_1 é a razão entre o número de operações e o total de elementos nos blocos:

$$\eta_1 = \frac{e_1^{in}}{e_1^{in} + e_0^{in}}$$

- η_2 é a razão entre o número de lacunas e o total de elementos fora dos blocos:

$$\eta_2 = \frac{e_0^{out}}{e_1^{out} + e_0^{out}}$$

- q é um fator de peso ($0 \leq q \leq 1$)
- $0 \leq \eta \leq 1$

Podemos observar que o fator de peso q é utilizado para dar mais importância à minimização das lacunas ou das exceções. Logo, se $q = 0.5$, ambas possuem o mesmo impacto na qualidade da solução. Quando não houver nenhuma operação dentro dos blocos e todos os elementos com valor 0 estiverem dentro dos blocos, η_1 e η_2 serão iguais a 0. Por outro lado, quando todas as operações estiverem dentro dos blocos e todos os elementos com valor 0 estiverem fora dos blocos, η_1 e η_2 serão iguais a 1. Nestes casos, η será respectivamente 0 e 1 independentemente do valor de q .

Calculando-se a Eficiência de Agrupamento da solução apresentada na Figura 2.2, considerando $q = 0.5$, temos que $\eta_1 = \frac{12}{12} = 1$ e $\eta_2 = \frac{19}{23} = 0,826$, logo $\eta = 0,5 \cdot 1 + 0,5 \cdot 0,826 = 0,913$. Para a solução apresentada na Figura 2.3, também considerando $q = 0.5$, temos que $\eta_1 = \frac{14}{17} = 0,82$ e $\eta_2 = \frac{16}{18} = 0,89$; logo, $\eta = 0,5 \cdot 0,82 + 0,5 \cdot 0,89 = 0,855$.

2.5.2 Eficácia de Agrupamento

O índice de **Eficácia de Agrupamento** foi proposto por Kumar & Chandrasekharan [35] com o objetivo de evitar as dificuldades em estabelecer o parâmetro q da Eficiência de Agrupamento. A partir de Srinivasan & Narendran [68], diversas abordagens são desenvolvidas considerando esta medida como padrão na avaliação de soluções. Este índice é definido como:

$$\mu = \frac{e_1 - e_1^{out}}{e_1 + e_0^{in}} \quad (2.20)$$

Nesta equação, o numerador contém o número de operações nas células, e o denominador aumenta conforme o número de exceções. Observe que, quanto mais próximo de 1 for este índice, melhor será o agrupamento.

Calculando-se a Eficácia de Agrupamento da solução apresentada na Figura 2.2, temos que $\mu = \frac{16-4}{16+0} = 0,75$. Para a solução apresentada na Figura 2.3, temos $\mu = \frac{16-2}{16+3} = 0,7368$.

2.5.3 Índice de Capacidade de Agrupamento

Proposta por Hsu [29], o **Índice de Capacidade de Agrupamento** (GCI - *Grouping Capability Index*) é definido por:

$$\text{GCI} = 1 - \frac{e_1^{out}}{e_1} \quad (2.21)$$

O GCI tem seu valor igual a 0 quando os valores de e_1^{out} e e_1 forem iguais, ou seja, todas as operações da matriz estiverem fora dos blocos. Teremos valor 1 quando não

existirem exceções ($e_1^{out} = 0$). Esta medida não leva em conta o número de lacunas, o que pode fazer com que soluções contendo células grandes e com um número considerável de lacunas sejam consideradas boas soluções.

Calculando-se o Índice de Capacidade de Agrupamento da solução apresentada na Figura 2.2, temos que $GCI = 1 - \frac{4}{16} = 0,75$. Para a solução apresentada na Figura 2.3, temos $GCI = 1 - \frac{2}{16} = 0,875$. Observe que, diferentemente das medidas apresentadas até agora, a solução da Figura 2.3 foi considerada pior que a da Figura 2.2.

2.5.4 Medida Primária

Proposta por Miltenburg & Zhang [45], a **Medida Primária** é definida por:

$$n_g = \frac{e_1^{in}}{e_1^{in} + e_0^{in}} - \frac{e_1^{out}}{e_1} \quad (2.22)$$

Quando não houver operações executadas dentro das células, o valor de e_1^{in} será igual a 0 e o número de operações executadas fora das células será igual ao número de operações da matriz ($e_1^{out} = e_1$). Desta forma, o valor de n_g será igual a -1 . Por outro lado, quando todas as operações da matriz forem executadas dentro das células e não houver lacunas nas mesmas ($e_1^{out} = e_0^{in} = 0$), o valor de n_g será igual a 1.

Calculando-se a Medida Primária da solução apresentada na Figura 2.2, temos que $n_g = \frac{12}{12+0} - \frac{4}{16} = 0,75$. Para a solução apresentada na Figura 2.3, temos $n_g = \frac{14}{14+3} - \frac{2}{16} = 0,6985$. Vale ressaltar que nesta medida, apesar do número de lacunas ter sido levado em consideração, a solução da Figura 2.3 também foi considerada pior que a da Figura 2.2.

2.5.5 Índice de Célula

Proposta por Mukattash et al. [50], o **Índice de Célula** (CI - *Cell Index*) é dado por:

$$CI = \frac{1}{n} \cdot \sum_{j=1}^P \frac{k_j \cdot n_j}{k_j + v_j + e_j} \quad (2.23)$$

onde:

n : número total de máquinas;

n_j : número total de máquinas na j -ésima célula;

v_j : número de lacunas na j -ésima célula;

e_j : número de exceções relacionadas com a j -ésima célula;

k_j : número de operações na j -ésima célula;

p : número de células;

$0 \leq \text{CI} \leq 1$.

O valor de **CI** será igual a 0 quando o valor de k_j for igual a 0 para todas as células, ou seja, na ausência de operações nas células. Quando não houver exceções nem lacunas, os valores de v_j e e_j serão iguais a 0; desta forma:

$$\text{CI} = \frac{1}{n} \cdot \sum_{j=1}^P \frac{k_j \cdot n_j}{k_j + v_j + e_j} = \frac{1}{n} \cdot \sum_{j=1}^P \frac{k_j \cdot n_j}{e_j} = \frac{1}{n} \cdot \sum_{j=1}^P n_j = 1 \quad (2.24)$$

Um aspecto interessante desta medida é que ela permite a avaliação do impacto individual de cada célula na qualidade da solução, possibilitando ao projetista do sistema de manufatura identificar e ajustar as células ruins.

2.6 Considerações sobre as Medidas

Escolhemos a eficácia de agrupamento como medida de desempenho para os algoritmos propostos neste trabalho por diversas razões, apresentadas a seguir.

- na literatura é considerada uma medida padrão para representar a qualidade das soluções.
- é considerada uma medida melhor que a eficiência de agrupamento.
- incorpora na avaliação tanto a subutilização de máquinas na célula quanto a movimentação de partes entre células.
- não requer um fator de peso para priorizar um dos dois aspectos acima.
- possui um alto poder de discriminação entre matrizes bem e mal estruturadas (matrizes com um grande número de exceções).

Capítulo 3

Abordagens Existentes para o PFCM

Nas últimas três décadas, um grande número de artigos científicos e relatórios técnicos tem sido publicados sobre a formação de células de manufatura. Revisões da literatura podem ser encontradas em Selim et al. [63] e Papaioannou & Wilson [53]. Nas seções a seguir, apresentaremos uma classificação das principais técnicas empregadas ao longo dos anos.

3.1 Procedimentos Descritivos

Uma visão geral dos primeiros métodos desta categoria é apresentado em Askin & Vakharia [4]. São métodos que se utilizam de bancos de dados relacionais, classificação e sistema de codificação para a criação dos agrupamentos. Em geral, podem ser classificados em três subcategorias:

Identificação de família de partes: inicia o processo de formação das células identificando as famílias de partes e em seguida faz a alocação das máquinas às famílias

Identificação de células de máquinas: procede com a identificação das células de máquinas baseada nas informações disponíveis no roteamento das partes antes de identificar as famílias de partes, e então alocando-as nas células de máquinas

Agrupamento família–parte/célula–máquina: determina a família de partes e a célula de máquinas simultaneamente.

A principal abordagem é a *Part Coding Analysis (PCA)*, que consiste em utilizar um sistema de codificação para atribuir pesos às características das partes em relação a sua forma geométrica, dimensões, características de produção, tipos de materiais, entre outros. Com base nesses códigos, apenas as famílias de partes são criadas, e a formação das células fica como um processo secundário.

3.2 Clusterização

Os métodos de clusterização são caracterizados por reconhecer determinadas estruturas em um conjunto complexo de dados. O principal objetivo é de agrupar objetos, entidades ou seus atributos de tal maneira que eles possuam um alto grau de “associação natural”. As técnicas de clusterização podem ser classificadas em: clusterização baseada em matrizes, clusterização hierárquica e clusterização não-hierárquica.

3.2.1 Clusterização Baseada em Matrizes

Esta é uma das técnicas mais simples para a resolução do PFCM. Utilizam como descrição do problema uma matriz binária, conhecida como matriz parte-máquina, representando as operações de partes nas máquinas. O algoritmo realiza várias permutações nas linhas e colunas, agrupando as máquinas em células e as partes em famílias, formando blocos agrupados (submatrizes) de operações ao longo da diagonal principal.

Revisões sobre os métodos de clusterização baseada em matrizes são apresentados em Chu [18], Singh [66], Shafer [64]. Os métodos mais conhecidos desta categoria são:

- *Bond Energy Analysis* (BEA) [44]
- *Rank Order Clustering* (ROC) [33]
- *Improved Rank Order Clustering* (ROC2) [32]
- *Modified Rank Order Clustering* (MODROC) [13]
- *Direct Clustering Analysis* (DCA) [11]

Estes métodos são eficientes e de fácil aplicação na matriz parte-máquina. No entanto, são dependentes da configuração inicial da matriz e possuem dificuldades em prover bons agrupamentos para matrizes mal estruturadas, ou seja, com uma quantidade excessiva de elementos excepcionais (operações inter-células).

Em geral, uma importante crítica contra a resolução baseada em matrizes de incidência binárias, quando se trata da generalização do problema, é a de que não levam em consideração outras informações, tais como o volume de produção, o custo das máquinas, o tamanho máximo das células, entre outros, que podem influenciar significativamente na formação das células. Isto ocorre pois não é possível representar em uma matriz de incidência a rota das partes ou a sequência de operações, uma vez que tais matrizes não expressam o número de vezes que uma máquina é necessária e em que ordem.

3.2.2 Clusterização Hierárquica

Nos métodos de clusterização hierárquica, o agrupamento de máquinas ou partes é realizado por etapas. Com base em dados de entrada sob a forma de coeficientes de similaridade, determina-se uma hierarquia de agrupamentos das máquinas ou partes. Inicialmente obtêm-se um agrupamento inicial, e a cada iteração um novo agrupamento, com base no anterior, é construído, até que não seja mais possível realizar novas iterações ou um critério de parada especificado seja alcançado. Os algoritmos desta classe podem ser divididos em duas categorias:

Divisivo: iniciam colocando todas as máquinas (partes) em um mesmo grupo e iterativamente particionam-se estes grupos, até que cada máquina (parte) esteja em um único grupo.

Aglomerativo: iniciam com cada máquina (parte) em um grupo diferente e iterativamente são unidos, até que um número desejado de células seja formado ou até que um único grupo tenha sido formado.

Estes métodos, diferentemente da clusterização baseada em matrizes, possuem a flexibilidade de incorporar outras informações além das que a matriz parte-máquina fornece, e levam em conta fatores como: capacidade de produção de uma máquina, tempo da produção, volume de produção de uma peça, entre outros. Em compensação, não conseguem criar as células de máquinas e as famílias de partes simultaneamente. Como exemplos de algoritmos que se enquadram nesta categoria, podemos citar o algoritmo divisivo proposto por Harhalakis et al. [28] e a segunda etapa do algoritmo MODROC [13].

3.2.3 Clusterização Não-hierárquica

Os métodos de clusterização não-hierárquica são procedimentos iterativos que realizam o particionamento de um conjunto de dados por meio de uma série de agrupamentos, sem que haja uma hierarquia entre si. Inicialmente, define-se alguns dados de forma arbitrária, artificial ou natural, como sementes para a clusterização. Em seguida, os dados restantes são agrupados ao cluster mais promissor, com base numa função de similaridade ou distância entre os dados e os clusters.

Em todos os casos, o número de clusters precisa ser definido *a priori*, o que é sem dúvida uma desvantagem, pois esta escolha potencialmente força alguns clusters naturais a serem unidos ou particionados. No entanto, permite que diferentes números de clusters possam ser testados.

Muitas heurísticas que adotam esta abordagem permitem a identificação de boas partições, mas possivelmente sub-ótimas. Ao contrário da clusterização hierárquica, esta abordagem não precisa calcular e armazenar matrizes de similaridade. Os algoritmos ZODIAC

[14], baseado em um critério de avaliação da eficiência de agrupamento, e GRAPHICS [68], método que gera sementes iniciais a partir de um problema de alocação que maximiza a similaridade entre as máquinas, são os exemplos mais conhecidos desta classe. No geral, os algoritmos dessa classe podem ser classificados em três categorias:

Métodos de Passo Simples: produzem clusters de forma dependente da ordem em que as máquinas e as partes são tratadas.

Métodos de Relocação: aloca máquinas ou partes em um número definido de clusters, então os realoca iterativamente de forma a otimizar o resultado. Tais métodos são inclinados a alcançarem ótimos locais ao invés de ótimos globais, e geralmente não é possível saber quando um ótimo global foi atingindo.

Métodos de Vizinhança Mais Próxima: aloca as máquinas ou partes no mesmo cluster que seus vizinhos mais próximos. Uma lista com estes vizinhos e o nível de similaridade entre eles precisa ser previamente definido.

3.3 Teoria dos Grafos

Esta abordagem trata as máquinas e/ou partes como vértices e o atendimento das partes por uma máquina como arestas. Estes modelos visam encontrar subgrafos desconexos de um grafo máquina-máquina ou parte-máquina para identificar as células de manufatura e alocar as partes às células.

O trabalho de Rajagopalan & Batra [55] foi o primeiro desenvolvido com base na Teoria dos Grafos. Este utiliza um grafo em que cada nó representa uma máquina, e as arestas especificam a similaridade entre elas (utilizando o coeficiente de JACCARD, visto em detalhes na Seção 2.4.1). Uma aresta conecta dois vértices no grafo apenas se a similaridade entre as máquinas for maior que um valor limiar previamente especificado. Esta similaridade é então avaliada, dependendo do número de partes usadas em ambas as máquinas. Depois que todas as arestas foram calculadas, subgrafos (cliques) são formados. Estes pequenos cliques são então unidos, de forma a criar células de máquinas que minimizem a movimentação inter-células.

Em Vannelli & Kumar [72] um modelo de grafo é usado para obter uma estrutura de blocos diagonal perfeita utilizando duplicação de máquinas. Em Askin et al. [1] é proposto um algoritmo que encontra um caminho hamiltoniano que reorganiza as linhas da matriz de forma a gerar uma estrutura diagonal de blocos perfeita na matriz parte-máquina.

A limitação desta categoria de métodos é que as células de máquinas e as famílias de partes não são formadas simultaneamente. No entanto, com esta abordagem, clusters podem ser naturalmente formados, desde que não haja uma especificação *a priori* para o número de clusters.

3.4 Meta-heurísticas

Duas desvantagens básicas das heurísticas são que as soluções encontradas frequentemente são de baixa qualidade e fortemente dependentes das instâncias e do problema. Para contornar essas desvantagens, as meta-heurísticas são providas de mecanismos de escape de ótimos locais, permitindo que o espaço de busca seja melhor explorado. As meta-heurísticas diferenciam-se uma das outras pelo mecanismo empregado para escapar dos ótimos locais. Elas podem ser divididas em duas categorias, de acordo com o princípio usado para explorar o espaço de soluções:

busca local: a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança;

busca populacional: consistem em manter um conjunto de boas soluções e combiná-las de forma a tentar produzir soluções ainda melhores.

O grande número de trabalhos encontrados na literatura sugere que estes métodos são promissores para a aplicação no PFCM. Nas seções a seguir, veremos algumas das meta-heurísticas mais empregadas e os trabalhos mais relevantes de cada uma.

3.4.1 Simulated Annealing

O *Simulated Annealing* é uma meta-heurística de busca local probabilística, que se fundamenta em uma analogia com a termodinâmica, ao simular o esfriamento de um conjunto de átomos aquecidos, operação conhecida como recozimento. Esta técnica imita o processo de resfriamento de um sistema físico de modo a encontrar um estado de energia potencial mínima global.

A natureza estocástica do algoritmo permite escapar de mínimos locais, explorar o espaço de estados e encontrar soluções ótimas ou próximas da ótima. Metodologias para a resolução do PFCM utilizando esta técnica são apresentadas em Harhalakis et al. [28] e Xambre & Vilarindo [80]. A qualidade dos resultados depende da temperatura inicial e da velocidade de resfriamento. Quanto maior a temperatura inicial e quanto mais lenta a velocidade de resfriamento, melhor será o resultado final. No entanto, o tempo requerido para alcançar a solução ótima aumenta com esses dois parâmetros.

3.4.2 Algoritmos Genéticos

Trata-se de uma meta-heurística que se fundamenta em uma analogia com processos naturais de evolução, nos quais, dada uma população, os indivíduos com características genéticas melhores têm maiores chances de sobrevivência e de produzirem filhos cada

vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer. Nos Algoritmos Genéticos (AGs), cada cromossomo (indivíduo da população) está associado a uma solução do problema e cada gene está associado a uma componente da solução. Um mecanismo de reprodução, baseado em processos evolutivos, é aplicado sobre a população com o objetivo de explorar o espaço de busca e encontrar melhores soluções para o problema.

Existem vários algoritmos na literatura que se utilizam desta técnica para resolver o PFCM. Em Joines et al. [31] é apresentado um algoritmo genético com programação inteira, permitindo funções-objetivo com múltiplos critérios e restrições. O algoritmo possui 4 operadores de mutação e 3 operadores de *crossover*. Em Gonçalves & Resende [27] é apresentado uma técnica híbrida combinando algoritmo genético com uma heurística de busca local. O algoritmo genético é utilizado para construir células de máquinas e a busca local para agrupar as peças às máquinas.

3.4.3 Busca Tabu

A Busca Tabu é uma técnica de otimização baseada na noção de vizinhança, originada nos trabalhos de Glover [25, 26]. Sendo S o espaço de busca de um problema de otimização, f a função objetivo a minimizar ou maximizar e s uma solução viável do problema, denota-se *vizinhança* de s por $N(s) \subset S$. Cada solução $s' \in N(s)$ é chamada de *vizinho* de s e é obtida por uma modificação em sua estrutura denominada *movimento*, dependente do problema em questão. Para evitar que a Busca Tabu retorne a soluções já geradas anteriormente, existe uma **Lista Tabu**, que é uma lista contendo movimentos proibidos. Geralmente a Lista Tabu contém movimentos reversos aos últimos T movimentos realizados (onde T é um parâmetro do método).

Este método foi proposto para o PFCM em Wu et al. [79], e se apresenta em três fases: primeiramente determina-se o melhor sequenciamento de máquinas e partes, de maneira a formar dois caminhos mínimos (um para partes e outro para máquinas), onde as distâncias estão diretamente relacionadas com o grau de similaridade. Numa segunda fase, é montada a matriz de clusterização referente à melhor solução encontrada na primeira fase. Na última fase determina-se o número de clusters através da identificação das fronteiras de cada cluster e da informação por parte do projetista de um limite superior para o número de clusters a serem formados.

3.5 Outras Abordagens

Além dos métodos mostrados até agora, existem diversos outros que são menos explorados na literatura do PFCM. Alguns desses métodos são baseados em conceitos de inteligência artificial e apresentam bons resultados.

3.5.1 Redes Neurais

A primeira aplicação de Redes Neurais Artificiais (RNA) para o PFCM foi proposta por Rao & Gu [56]. As RNAs imitam o funcionamento do cérebro humano. São compostas por elementos de processamento (neurônios) interconectados, que processam, recebem e retransmitem informações. Estes neurônios mudam sua estrutura baseados nestas informações (internas e externas) que fluem na rede durante a fase de aprendizado.

Em Ateme-Nghema & Dao [5] é apresentado um modelo de RNA baseado no modelo de *Hopfield* que tem como principal vantagem a capacidade de resolver instâncias grandes. Shankar et al. [65] apresentam um modelo transiente de RNA e comparam os resultados com quatro outros modelos que também utilizam redes neurais.

3.5.2 Lógica Fuzzy

A Lógica Fuzzy é uma extensão da Lógica Booleana que admite valores lógicos intermediários entre o FALSO(0) e o VERDADEIRO(1); por exemplo o valor médio ‘TAL-VEZ’(0,5). Isto significa que um valor lógico fuzzy é um valor qualquer no intervalo de valores entre 0 e 1. Um algoritmo que utiliza Lógica Fuzzy é encontrado em Ravichandran & Rao [57], onde a entrada do algoritmo é uma matriz parte-máquina de relação fuzzy em que cada elemento representa o grau de relacionamento entre a máquina e a parte, representado por um coeficiente que varia entre 0 e 1.

Capítulo 4

Proposta

Neste capítulo são apresentadas as contribuições em termos de algoritmos. Essas contribuições consistem nas meta-heurísticas Greedy Randomized Adaptive Search Procedure (GRASP) e Iterated Local Search (ILS), ambas fazendo uso do método Variable Neighborhood Descent (VND) com uma ordenação aleatória da vizinhança de busca na fase de busca local para a resolução do PFCM. Em todos os algoritmos, considera-se trivial a inclusão das restrições para o tamanho mínimo da célula, e portanto, não será feita distinção nas explicações.

Nas meta-heurísticas propostas, uma heurística baseada em árvore geradora mínima para a fase de construção é implementada. Os detalhes desta heurística construtiva, são cobertos na Seção 4.1.1. A fase de busca local, por sua vez, será explicada em detalhes na Seção 4.1.2, em que três vizinhanças de busca são propostas e incluídas numa estratégia de mudança sistemática destas vizinhanças.

Estes métodos são aplicados tanto no GRASP quanto no ILS. No entanto, a principal diferença entre as meta-heurísticas propostas consiste no uso de procedimentos de perturbação da solução. Na Seção 4.2.1, são apresentados três destes procedimentos, que são escolhidos de forma aleatória durante a execução do ILS. Além disso, na fase de construção deste, uma versão gulosa da heurística apresentada é utilizada, como será explicado na Seção 4.2.

4.1 Algoritmo GRASP-RVND

O GRASP é uma meta-heurística em que cada iteração consiste de duas fases: uma para a construção de uma solução inicial e outra que busca melhores soluções a partir desta [21]. A solução geral é atualizada sempre que a fase de busca encontrar uma solução melhor que todas as outras já encontradas. O processo é repetido até um certo número de iterações ou até um critério de parada estabelecido ser satisfeito. Nos últimos anos, o GRASP tem sido aplicado com sucesso em uma grande variedade de problemas de otimização [58].

A entrada do problema consiste de uma matriz parte-máquina A e um inteiro k . Na matriz A , cada célula a_{pm} representa a utilização da parte p pela máquina m . Uma vez encontrada uma solução para A , os conjuntos M e P são, respectivamente, as células de máquinas e as famílias de partes que representam os agrupamentos desta solução.

Com base na matriz A , cria-se uma clique de máquinas C , um grafo completo em que os vértices correspondem às máquinas e os pesos são dados pela expressão $d_{ij} = \sum_{k=1}^n |a_{ki} - a_{kj}|$, que é a distância entre as máquinas i e j , ou seja, o número de partes que são operadas em apenas uma das máquinas. A partir da clique C obtemos a árvore geradora mínima T por meio do algoritmo de *Kruskal* [19]. Nesta árvore, cada aresta conterá as máquinas mais similares entre si quanto à sua utilização na operação de partes, e portanto, possuem uma grande chance de estarem na mesma célula.

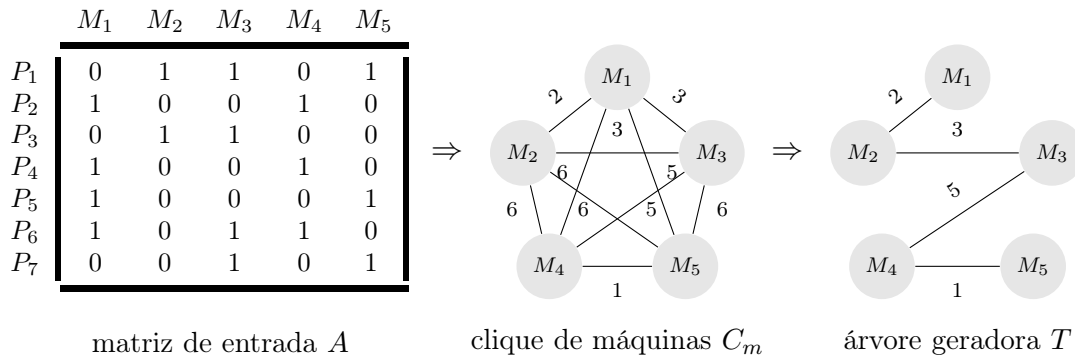


Figura 4.1: Processamento inicial da entrada

Esta árvore T fornecerá sementes para a criação das soluções iniciais no GRASP, explicado com detalhes na Seção 4.1.1. Em seguida, inicia-se uma busca por melhores soluções a partir desta solução inicial encontrada, através do processo apresentado na Seção 4.1.2. No Algoritmo 4.1 apresentamos a visão geral do GRASP.

Algoritmo 4.1 GRASP-RVND

- 1: **procedimento** GRASP-RVND(A_0, k)
 - 2: $it \leftarrow 1$
 - 3: $\mu^* \leftarrow 0$
 - 4: dado A_0 , crie uma clique de máquinas C com pesos $d_{ij} \quad \forall i, j \in M$
 - 5: calcule a árvore geradora de custo mínimo T de C
 - 6: **repita**
 - 7: $A_{it} \leftarrow \text{CONSTRUTIVO}(T, k)$ $\triangleright k = 1$, guloso; $k = m$, aleatório
 - 8: $A'_{it} \leftarrow \text{RVND}(A_{it})$
 - 9: **se** $\mu(A'_{it}) > \mu^*$ **então**
 - 10: $A^* \leftarrow A'_{it}$
 - 11: $\mu^* \leftarrow \mu(A'_{it})$
 - 12: **fim se**
 - 13: $it \leftarrow it + 1$
 - 14: **até** $it = \text{MaxIt}$
 - 15: **retorne** A^*
 - 16: **fim procedimento**
-

4.1.1 Método Construtivo

Dada como entrada a árvore geradora mínima T , as soluções iniciais de cada iteração do GRASP são construídas com base em remoções aleatórias das arestas $E(T)$ desta árvore. Na floresta geradora resultante, cada árvore definirá um das células que compõem o conjunto de células de máquinas M . Com base nisso, verifica-se a qual célula M cada parte deve se associar, de maneira a conter o menor número de lacunas e exceções, formando um conjunto de famílias P . Em seguida, combinamos M e P para formarmos a matriz resultante A' . Sumarizamos no Algoritmo 4.2 o método proposto.

Algoritmo 4.2 Método Construtivo

```

1: procedimento CONSTRUTIVO( $T, k$ )
2:   enquanto  $E(T) \neq \emptyset$  faça
3:      $\mu \leftarrow \{\infty, \dots, \infty\}_{1 \times |E(T)|}$ 
4:     para todo  $i \in E(T)$  faça
5:        $T_i \leftarrow T - \{i\}$  ▷ Remoção da aresta  $i$  de  $T$ 
6:       construa  $M_i$  a partir de  $T_i$  ▷  $M_i$  célula de máquinas baseada em  $T_i$ 
7:       construa  $P_i$  ideal dado  $M_i$  ▷  $P_i$  família de produtos baseada em  $M_i$ 
8:       gere matriz  $A_i$  dado  $M_i$  e  $P_i$ 
9:     fim para
10:    escolha  $\tilde{A}$  aleatoriamente dentre os  $k$  melhores  $A_i$  baseado nos valores em  $\mu$ 
11:    atualize  $T$  pelo  $\tilde{T}$  referente ao  $\tilde{A}$  escolhido
12:    se  $\tilde{\mu} > \mu'$  então
13:       $A' \leftarrow \tilde{A}$ 
14:    fim se
15:  fim enquanto
16:  retorne  $A'$ 
17: fim procedimento

```

4.1.2 Busca Local

Ao término da fase de construção, uma solução inicial para o problema é dada. Em seguida, é executada uma fase de busca local com o propósito de melhorar a solução apresentada. Nesta fase, pelo menos uma vizinhança de busca é necessária, mas é preferível que mais vizinhanças estejam disponíveis; geralmente, quanto maior o número de vizinhanças distintas, melhores são os resultados. No entanto, vizinhanças demais podem se sobrepor, ou seja, realizar o mesmo trabalho, reduzindo a eficiência do algoritmo e ocasionando um tempo computacional excessivo.

Apesar dos benefícios em se definir vizinhanças diferentes para a qualidade da solução final, é também de suma importância mudá-las à medida que a busca é conduzida. O VND proposto por Mladenovic & Hansen [47] é um método que consiste em explorar o espaço de soluções através de trocas sistemáticas das estruturas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor for encontrada. Dependendo do problema abordado, a busca pelo melhor vizinho pode ser cara computacionalmente. Nesta situação é comum fazer a busca pela

primeira solução de melhora. Outra alternativa é considerar a exploração em apenas um certo percentual da vizinhança.

Uma variante do VND conhecida como Random Variable Neighborhood Descent (RVND) utiliza uma ordenação aleatória das vizinhanças na fase de busca local. Seja $V = V^1, \dots, V^r$ o conjunto de estruturas de vizinhança. Sempre que uma dada vizinhança do conjunto V falha em melhorar a solução atual, o RVND (ver Algoritmo 4.3) toma a próxima vizinhança. A ordem de visita é pré-estabelecida aleatoriamente, e uma próxima chamada a esse procedimento possivelmente trará uma outra ordem de visitação. Esta variante é empregada com sucesso em Munhoz [51].

Algoritmo 4.3 RVND

```

1: procedimento RVND( $A$ )
2:   inicialize  $V$  aleatoriamente
3:    $k \leftarrow 1$ 
4:    $\mu^* \leftarrow \mu(A)$ 
5:   repita
6:      $A' \leftarrow V^k(A)$ 
7:     se  $\mu(A') > \mu^*$  então
8:        $A^* \leftarrow A'$ 
9:        $k \leftarrow 1$ 
10:    senão
11:       $k \leftarrow k + 1$ 
12:    fim se
13:  até  $k = k_{max}$ 
14:  retorne  $A^*$ 
15: fim procedimento

```

Para a construção do RVND, três vizinhanças de busca são empregadas: Reagrupar, Deslocamento de Máquinas (DM) e Troca de Máquinas (TM). Adicionalmente, para as vizinhanças DM e TM, um procedimento BuscaCandidatos é executado com o objetivo de reduzir o espaço de busca. Essas vizinhanças são apresentada a seguir em detalhes, seguidas pelo procedimento BuscaCandidatos.

Reagrupar

Esta busca consiste em inverter alternadamente a entrada ao processo da linha 7 do Algoritmo 4.2. Como visto na Seção 4.1.1, esta construção toma por base as células de máquinas M para criar a família de partes P . No processo inverso, considera-se como dada a família de partes P , e para cada parte $p \in P$ verifica-se em qual agrupamento de M a parte p deve ser associada, de modo a minimizar o número de lacunas e exceções. Este procedimento é realizado até que as células de máquinas M' encontradas pela busca sejam iguais às células M inicial, ou seja, a partir do momento em que a busca entrar em *loop*. Ao longo do procedimento, atualiza-se A^* caso uma melhor solução seja encontrada.

Algoritmo 4.4 Reagrupar

```

1: procedimento REAGRUPAR( $A_0, M_0$ )
2:    $A^* \leftarrow A_0$ 
3:    $\mu_0 \leftarrow \mu(A_0)$ 
4:    $M'_0 \leftarrow M_0$ 
5:    $it \leftarrow 1$ 
6:   repita
7:      $M_{it} \leftarrow M'_{it-1}$ 
8:     procure pelo melhor  $P_{it}$  dado  $M_{it}$ 
9:     combine  $M_{it}$  e  $P_{it}$  para criar  $A_{it}$ 
10:    se  $\mu(A_{it}) < \mu(A_{it-1})$  então
11:       $A^* \leftarrow A_{it}$ 
12:    fim se
13:    procure pelo melhor  $M'_{it}$  dado  $P_{it}$ 
14:    combine  $M'_{it}$  e  $P_{it}$  para criar  $A'_{it}$ 
15:    se  $\mu(A'_{it}) < \min\{\mu(A_{it}), \mu(A_{it-1})\}$  então
16:       $A^* \leftarrow A'_{it}$ 
17:    fim se
18:     $it \leftarrow it + 1$ 
19:  até  $M'_i = M_i$ 
20:  retorne  $A^*$ 
21: fim procedimento

```

Deslocamento de Máquina

Esta busca tem o objetivo de fazer pequenos ajustes através da movimentação de máquinas de uma célula para outra. Este movimento de máquinas é também a menor modificação possível numa solução e, portanto, a operação ideal numa tentativa de refino de soluções. O Algoritmo 4.5 ilustra o funcionamento desta busca. Inicialmente é realizada uma busca por máquinas candidatas à mudança de célula W , por meio do procedimento dado pelo Algoritmo 4.7. Essas máquinas são consideradas as que mais influenciam no mau desempenho de uma determinada célula, e portanto possuem uma grande probabilidade de que pertençam a outra célula. Em seguida, estas máquinas candidatas são deslocadas para todas as demais células existentes. Sempre quando uma solução melhor for encontrada, A^* é atualizada.

	M_3	M_1	M_4	M_5	M_2		M_2	M_3	M_1	M_4	M_5	
P_3	1	0	0	0	1		1	1	0	0	1	
P_7	1	0	0	1	0		1	1	0	0	0	
P_2	0	1	1	0	0	\Rightarrow	P_2	0	0	1	1	0
P_4	0	1	1	0	0		P_4	0	0	1	1	0
P_6	1	1	1	0	0		P_6	0	1	1	1	0
P_5	0	1	0	1	0		P_5	0	0	1	0	1
P_1	1	0	0	1	1		P_7	0	1	0	0	1

Figura 4.2: Deslocamento da máquina 2 para a célula 1

Algoritmo 4.5 Deslocamento de Máquinas

```

1: procedimento DM( $A, M$ )
2:    $A^* \leftarrow A$ 
3:    $W \leftarrow \text{BUSCACANDIDATOS}(A, M)$ 
4:   para todo  $w \in W$  faça
5:      $\phi \leftarrow$  célula em que  $w$  pertence
6:     para todo  $\varphi \in M \setminus \phi$  faça
7:        $M' \leftarrow M$  com  $w$  movido da célula  $\phi$  para a  $\varphi$ 
8:       procure pelo melhor  $P'$  dado  $M'$ 
9:       combine  $M'$  e  $P'$  para criar  $A'$ 
10:      se  $\mu(A') > \mu(A^*)$  então
11:         $A^* \leftarrow A'$ 
12:      fim se
13:    fim para
14:  fim para
15:  retorne  $A^*$ 
16: fim procedimento

```

Troca de Máquina

Esta busca realiza permutações entre as máquinas candidatas W . Para cada máquina candidata, trocas são realizadas com todas as outras máquinas candidatas, exceto pelas máquinas da própria célula. Sempre quando uma solução melhor for encontrada, A^* é atualizada. Diferentemente da busca por Deslocamento de Máquinas, não possui a capacidade de esvaziar uma célula, e portanto, diminuir sua quantidade. Além disso, realiza uma mudança mais significativa, já que esta busca pode corresponder a duas execuções da busca SHIFT.

	M_5	M_3	M_1	M_4	M_2		M_2	M_3	M_1	M_4	M_5	
P_1	1	1	0	0	1		1	1	0	0	1	
P_7	1	1	0	0	0		1	1	0	0	0	
P_2	0	0	1	1	0	\Rightarrow	0	0	1	1	0	
P_4	0	0	1	1	0		0	0	1	1	0	
P_5	1	0	1	0	0		P_6	0	1	1	1	0
P_6	0	1	1	1	0		P_5	0	0	1	0	1
P_3	0	1	0	0	1		P_7	0	1	0	0	1

Figura 4.3: Troca de células entre as máquinas 2 e 5

Busca de Candidatos

No Algoritmo 4.7, temos o procedimento que busca as máquinas candidatas à mudança de células. Este procedimento tem por objetivo diminuir o número de vizinhos das vizinhanças TM e DM, limitando a verificação apenas entre as máquinas mais prováveis de ocorrer uma melhora na solução.

Algoritmo 4.6 Troca de Máquinas

```

1: procedimento TM( $A, M$ )
2:    $W \leftarrow \text{BUSCACANDIDATOS}(A, M)$ 
3:   para todo  $u \in W$  faça
4:     para todo  $v \in W - \{u\}$  faça
5:        $M' \leftarrow M$  com  $u$  e  $v$  trocado de células
6:       procure pelo melhor  $P'$  dado  $M'$ 
7:       combine  $M'$  e  $P'$  para criar  $A'$ 
8:       se  $\mu(A') > \mu(A^*)$  então
9:          $A^* \leftarrow A'$ 
10:      fim se
11:    fim para
12:  fim para
13:  retorne  $A^*$ 
14: fim procedimento

```

Inicialmente temos que o conjunto de máquinas candidatas W é vazio. Então, para cada célula de máquinas, uma clique é formada com base nas máquinas presentes nesta célula, de forma análoga ao que foi descrito na Seção 4.1, em que os vértices representam as máquinas da célula. Em cada clique, considero como parte do conjunto de máquinas candidatas os vértices nos quais incidem as arestas de maior peso. Como critério de desempate temos que, destes vértices, o que possuir a maior soma dos pesos das arestas incidentes a ele será considerado como um vértice candidato.

Algoritmo 4.7 Busca de Candidatos

```

1: procedimento BUSCACANDIDATOS( $A, M$ )
2:    $W \leftarrow \emptyset$  ▷  $W$ , vértices candidatos à troca
3:   para todo  $\varphi \in M$  faça ▷  $\varphi$ , célula de máquina
4:     crie uma clique  $C_\varphi$  com pesos  $d_{ij} \quad \forall (i, j) \in \varphi$ 
5:      $\theta \leftarrow$  arestas de maior peso de  $C_\varphi$ 
6:     para todo  $v \in V(C_\varphi[\theta])$  faça ▷ critério de desempate
7:        $S_v \leftarrow$  soma dos pesos das arestas em  $C_\varphi$  incidentes a  $v$ 
8:     fim para
9:      $W_\varphi \leftarrow$  vértices de  $\varphi$  que possuem o maior  $S_v$ 
10:  fim para
11:  retorne  $W$ 
12: fim procedimento

```

4.2 Algoritmo ILS-RVND

Considere um ótimo local encontrado por um algoritmo de busca local. Em vez de reiniciar o mesmo procedimento a partir de uma solução completamente nova, o ILS realiza uma perturbação na solução, levando a um estado intermediário no qual a busca local será novamente realizada. Este procedimento é realizado iterativamente até que um critério de parada seja satisfeito.

Esta abordagem gera boas soluções se as perturbações não forem pequenas nem gran-

des demais. Perturbações muito pequenas podem ser desfeitas após a aplicação da busca local, enquanto perturbações excessivamente grandes levam a soluções aleatórias que eliminam a vantagem da amostragem realizada pelo método construtivo, tornando o procedimento um reinício aleatório.

Para implementar o ILS, três métodos são especificados: (i) CONSTRUTIVO, onde uma solução inicial é construída, baseado no Algoritmo 4.2; (ii) RVND, que tenta melhorar a solução obtida, baseado no Algoritmo 4.3; (iii) PERTURBAR, onde um novo ponto de início é gerado através da perturbação da solução encontrada no RVND, como será apresentado na Seção 4.2.1. O Algoritmo 4.8 descreve como esses componentes são combinados.

Algoritmo 4.8 ILS-RVND

```

1: procedimento ILS-RVND( $A$ )
2:    $itILS \leftarrow 0$ 
3:   transforme a matriz  $A$  numa clique de máquinas  $C_m$  com pesos  $d_{ij} \quad \forall i, j \in M$ 
4:   calcule a árvore geradora de custo mínimo  $T_m$  de  $C_m$ 
5:    $A \leftarrow \text{CONSTRUTIVO}(T_m, 1)$ 
6:    $A^* \leftarrow A$ 
7:   repita
8:      $A \leftarrow \text{RVND}(A)$ 
9:     se  $\mu(A) > \mu(A^*)$  então
10:       $A^* \leftarrow A$ 
11:       $itILS \leftarrow 0$ 
12:     senão
13:       $A' \leftarrow \text{CONSTRUTIVO}(T_m, k)$ 
14:       $A' \leftarrow \text{RVND}(A')$ 
15:      se  $\mu(A') > \mu(A^*)$  então
16:         $A^* \leftarrow A'$ 
17:      fim se
18:     fim se
19:      $A \leftarrow \text{PERTURBAR}(A^*)$ 
20:      $itILS \leftarrow itILS + 1$ 
21:   até  $itILS = maxItILS$ 
22:   retorne  $A^*$ 
23: fim procedimento

```

4.2.1 Perturbações

Como descrito na Seção 4.2, o ILS consiste em percorrer um caminho de exploração do conjunto de ótimos locais. Dada uma solução, a próxima solução a ser visitada é obtida através da aplicação de um procedimento de perturbação seguido de um procedimento de busca local. A seguir, são apresentados os algoritmos de perturbação de soluções propostos neste trabalho para integrar o ILS.

Deslocamentos Múltiplos de Máquinas

Realiza movimentações sucessivas de máquinas entre as células. Tanto as máquinas escolhidas quanto as células para onde as máquinas serão deslocadas são escolhidas alea-

toriamente.

Algoritmo 4.9 deslocamento aleatório de máquinas

```

1: procedimento RANDOMSHIFT( $A, M, \alpha$ )
2:    $n_\alpha \leftarrow |M| \cdot \alpha$ 
3:    $i \leftarrow 0$ 
4:    $MC \leftarrow$  todas as máquinas em  $M$ 
5:   repita
6:      $m \leftarrow$  retire aleatoriamente uma máquina de  $MC$ 
7:      $k \leftarrow$  selecione aleatoriamente uma célula de  $M$ 
8:      $M' \leftarrow$  mova máquina  $m$  para a célula  $k$ 
9:      $i \leftarrow i + 1$ 
10:  até  $i = n_\alpha$ 
11:  procure pelo melhor  $P'$  dado  $M'$ 
12:  combine  $M'$  e  $P'$  para criar  $A'$ 
13:  retorne  $A'$ 
14: fim procedimento
  
```

Remoção de Célula

Esta perturbação, seleciona uma célula aleatoriamente e a remove, distribuindo todas as máquinas pertencentes a ela nas demais células, escolhidas também de forma aleatória.

Algoritmo 4.10 remoção aleatória de uma célula

```

1: procedimento REMOVE( $A, M$ )
2:    $k \leftarrow$  selecione aleatoriamente uma célula de  $M$ 
3:    $LM \leftarrow$  todas as máquinas de  $M_k$ 
4:   enquanto  $LM \neq \emptyset$  faça
5:      $l \leftarrow$  selecione aleatoriamente uma célula destino de  $M \mid l \neq k$ 
6:      $LM \leftarrow$  retire uma máquina  $m$  de  $LM$ 
7:      $M' \leftarrow$  mova  $m$  da célula  $k$  para  $l$  em  $M$ 
8:   fim enquanto
9:   procure pelo melhor  $P'$  dado  $M'$ 
10:  combine  $M'$  e  $P'$  para criar  $A'$ 
11:  retorne  $A'$ 
12: fim procedimento
  
```

	M_2	M_3	M_1	M_4	M_5		M_2	M_3	M_1	M_4	M_5
P_1	1	1	0	0	1		1	1	0	0	1
P_3	1	1	0	0	0		1	1	0	0	0
P_2	0	0	1	1	0	\Rightarrow	0	1	1	1	0
P_4	0	0	1	1	0		0	0	1	1	0
P_6	0	1	1	1	0		0	0	1	0	1
P_5	0	0	1	0	1		0	0	1	0	1
P_7	0	1	0	0	1		0	1	0	0	1

Figura 4.4: Remoção da célula 2, distribuindo-se suas máquinas

Divisão de Célula

Esta perturbação seleciona uma célula aleatoriamente e a divide em duas células. Para tal, cada máquina desta célula tem 50% de chance de ser deslocada para uma nova célula.

Algoritmo 4.11 divisão aleatória de uma célula

- 1: **procedimento** SPLIT(A, M)
 - 2: $k \leftarrow$ selecione aleatoriamente uma célula de M
 - 3: $M' \leftarrow$ crie uma nova célula w
 - 4: $LM \leftarrow$ todas as máquinas de M_k
 - 5: **enquanto** $|LM| \neq \emptyset$ **faça**
 - 6: $LM \leftarrow$ retire uma máquina m de LM
 - 7: $M' \leftarrow$ mova m da célula k para w em M , com 50% de probabilidade
 - 8: **fim enquanto**
 - 9: procure pelo melhor P' dado M'
 - 10: combine M' e P' para criar A'
 - 11: **retorne** A'
 - 12: **fim procedimento**
-

	M_2	M_3	M_5	M_1	M_4		M_2	M_3	M_1	M_4	M_5	
P_1	1	1	1	0	0	\Rightarrow	P_1	1	1	0	0	1
P_3	1	1	0	0	0		P_3	1	1	0	0	0
P_7	0	1	1	0	0		P_2	0	0	1	1	0
P_2	0	0	0	1	1		P_4	0	0	1	1	0
P_4	0	0	0	1	1		P_6	0	1	1	1	0
P_5	0	0	1	1	0		P_5	0	0	1	0	1
P_6	0	1	0	1	1		P_7	0	1	0	0	1

Figura 4.5: Divisão da célula 1, em que M_5 formou uma nova célula

Capítulo 5

Experimentos Computacionais

Neste capítulo serão apresentados os resultados de testes computacionais realizados sobre os métodos propostos no Capítulo 4. Primeiramente, na Seção 5.1, são apresentadas as instâncias empregadas nos testes, a fonte em que foi definida e o valor de eficácia da melhor solução existente. Na Seção 5.2, são apresentados os resultados dos testes sobre essas instâncias, avaliando a performance e o desempenho quanto a qualidade da solução. E por fim, na Seção 5.3, são comparadas as melhores soluções encontradas com as soluções obtidas pelos demais métodos da literatura.

Estudaremos o PFCM com e sem restrição de *singleton*, que chamaremos respectivamente de PROBLEMA DE FORMAÇÃO DE CÉLULAS DE MANUFATURA COM RESTRIÇÃO DE SINGLETON (PFCM-CRS) e PROBLEMA DE FORMAÇÃO DE CÉLULAS DE MANUFATURA SEM RESTRIÇÃO DE SINGLETON (PFCM-SRS). Vale ressaltar que no PFCM-CRS proíbe-se soluções com a presença de células *singleton*, enquanto que o PFCM-SRS as admite.

Para avaliar a qualidade da solução, a medida de desempenho usada pelos algoritmos foi a Eficácia de Agrupamento, como explicado na seção 2.6. Nas tabelas apresentadas, os valores de eficácia serão multiplicados por 100, de maneira a apresentar os resultados em termos percentuais. Na avaliação sobre a performance, os tempos computacionais serão apresentados em segundos. Todos os métodos desenvolvidos foram executados numa máquina Intel Core i7-2600 com 3.40GHz e 32 GB de memória RAM no sistema operacional Arch Linux 3.3.4.

5.1 Instâncias

Foram utilizadas 35 instâncias da literatura, utilizadas em Gonçalves & Resende [27] e disponíveis no apêndice A deste trabalho, amplamente empregadas por diversos autores para avaliarem seus métodos. As matrizes selecionadas possuem tamanhos que variam de 5×7 a 40×100 , com diferentes graus de dificuldade, admitindo soluções com células

perfeitas até soluções com células mal estruturadas, ou seja, com um grande número de exceções e lacunas.

Na Tabela 5.1, para cada instância indicamos o tamanho da matriz de entrada $m \times p$, em que m é o número de máquinas e p o número de partes, a referência onde foi especificada (fonte) e a melhor eficácia obtida para o PFCM-CRS e o PFCM-SRS. Ressaltamos que, para cada valor de solução apresentada, cita-se em qual trabalho aparece pela primeira vez, considerando apenas os trabalhos comparados. Quando temos uma solução com o ótimo comprovado, citamos o trabalho de Pinheiro [54], no qual prova-se a otimalidade da maioria das instâncias, inclusive obtendo soluções ainda desconhecidas.

Tabela 5.1: Instâncias da Literatura do PFCM

Instância	Dimensão	Fonte	MSol. CRS	MSol. SRS
PFCM01	05 × 07	King & Nakornchai [32]	73.68 [54]	75.00 [54]
PFCM02	05 × 07	Waghodekar & Sahu [74]	62.50 [54]	69.57 [54]
PFCM03	05 × 18	Seifoddini [62]	79.59 [54]	80.85 [54]
PFCM04	06 × 08	Kusiak & Cho [39]	76.92 [54]	79.17 [54]
PFCM05	07 × 11	Kusiak & Chow [40]	53.13 [54]	60.87 [54]
PFCM06	07 × 11	Boctor [7]	70.37 [54]	70.83 [54]
PFCM07	08 × 12	Seifoddini & Wolfe [61]	68.29 [54]	69.44 [54]
PFCM08	08 × 20	Chandrasekharan & Rajagopalan [13]	85.25 [54]	85.25 [54]
PFCM09	08 × 20	Chandrasekharan & Rajagopalan [12]	58.72 [54]	58.72 [54]
PFCM10	10 × 10	Mosier & Taube [48]	70.59 [54]	75.00 [54]
PFCM11	10 × 15	Chan & Milner [11]	92.00 [54]	92.00 [54]
PFCM12	14 × 23	Askin & Subramanian [3]	69.86 [54]	74.24 [54]
PFCM13	14 × 24	Stanfel [70]	69.33 [54]	72.86 [54]
PFCM14	16 × 24	McCormick et al. [44]	51.96 [54]	53.33 [54]
PFCM15	16 × 30	Srinivasan et al. [69]	67.83 [54]	69.92 [54]
PFCM16	16 × 43	King [33]	56.52 [54]	58.04 [54]
PFCM17	18 × 24	Carrie [10]	54.46 [54]	57.73 [54]
PFCM18	20 × 20	Mosier & Taube [49]	42.96 [27]	43.97 [52]
PFCM19	20 × 23	Kumar et al. [36]	49.65 [54]	50.81 [54]
PFCM20	20 × 35	Carrie [10]	76.54 [54]	79.38 [54]
PFCM21	20 × 35	Boe & Cheng [8]	58.15 [54]	58.79 [54]
PFCM22	24 × 40	Chandrasekharan & Rajagopalan [15]	100.00 [54]	100.00 [54]
PFCM23	24 × 40	Chandrasekharan & Rajagopalan [15]	85.11 [54]	85.11 [54]
PFCM24	24 × 40	Chandrasekharan & Rajagopalan [15]	73.51 [54]	73.51 [54]
PFCM25	24 × 40	Chandrasekharan & Rajagopalan [15]	51.97 [54]	53.29 [77]
PFCM26	24 × 40	Chandrasekharan & Rajagopalan [15]	47.37 [78]	48.95 [20]
PFCM27	24 × 40	Chandrasekharan & Rajagopalan [15]	44.87 [27]	46.58 [20]
PFCM28	27 × 27	McCormick et al. [44]	54.27 [27]	54.82 [52]
PFCM29	28 × 46	Carrie [10]	46.06 [78]	47.68 [52]
PFCM30	30 × 51	Kumar & Vannelli [37]	58.94 [54]	63.04 [54]
PFCM31	30 × 50	Stanfel [70]	59.66 [54]	59.77 [54]
PFCM32	30 × 50	Stanfel [70]	50.51 [76]	50.83 [77]
PFCM33	30 × 90	King & Nakornchai [32]	42.64 [27]	47.93 [52]
PFCM34	37 × 53	McCormick et al. [44]	59.85 [78]	61.16 [52]
PFCM35	40 × 100	Chandrasekharan & Rajagopalan [14]	84.03 [54]	84.03 [54]

5.2 Teste de Desempenho

Nesta seção, avaliaremos o desempenho dos métodos desenvolvidos quanto à qualidade da solução obtida e o tempo computacional. Para tal, serão realizadas duas baterias de testes, uma para o PFCM-CRS e outra para o PFCM-SRS. Em cada uma, executamos os algoritmos GRASP-RVND e ILS-RVND dez vezes para cada instância com um número fixo de iterações.

Estabelecemos um número de iterações $MaxIt$ proporcional à variedade de soluções construídas pelo método construtivo, pois foi observado que quanto maior essa variedade, mais iterações são necessárias para a convergência da solução. Esta variedade é definida pelo número de máquinas m da instância e um fator $k = 3$ de aleatoriedade utilizado. Para o GRASP-RVND utilizamos um total de $MaxIt = 1 + m \cdot (k - 1)$ iterações e para o ILS-RVND utilizamos $MaxIt = k \cdot \lceil m^{1/2} \rceil$ iterações sem melhora na solução.

Em cada execução, computamos a eficácia de agrupamento da solução e o tempo computacional gasto. Para cada instância, verificamos a solução de melhor qualidade, computamos a média da eficácia de agrupamento das soluções e o desvio padrão observado. Além disso, apresentamos o tempo computacional correspondente à média dos tempos.

No entanto, isso ainda não é suficiente para atestar o desempenho dos métodos. Podemos ter uma média próxima à melhor solução onde em apenas uma das dez execuções foi encontrado um valor distante, ou podemos ter várias execuções em que não foi encontrada a melhor solução, mas que a diferença entre essas soluções e a melhor foi pouca. Para tal, apresentamos ainda o desvio padrão para quantificar a dispersão dos resultados e melhorar a avaliação. Na Tabela 5.2 é apresentado os resultados da primeira bateria de testes, onde analisou-se o desempenho do GRASP-RVND e do ILS-RVND na resolução do PFCM-CRS.

Observando os resultados desta tabela podemos constatar que, em termos de tempo computacional, o GRASP-RVND obteve um desempenho razoável frente a complexidade do problema, com um tempo máximo em torno de 23 segundos na maior instância. Já o ILS-RVND obteve um desempenho computacional consideravelmente melhor, com um tempo máximo em torno de 6 segundos na maior instância.

Pode-se observar também que a qualidade das soluções no ILS-RVND também é melhor que no GRASP-RVND. Um dos aspectos observados é que em mais instâncias foi obtido o mesmo valor de eficácia de agrupamento nas dez execuções, 22 casos para o GRASP-RVND contra 27 casos para o ILS-RVND. Outro aspecto observado é que, nas instâncias em que isso não ocorre, o desvio padrão foi ligeiramente menor, indicando uma maior robustez. Na Tabela 5.3 são apresentados os resultados da segunda bateria de testes, onde analisou-se o desempenho do GRASP-RVND e do ILS-RVND na resolução do PFCM-SRS.

Tabela 5.2: Teste de desempenho para o PFCM-CRS

Instância	GRASP-RVND				ILS-RVND			
	melhor	média	desvio	tempo (s)	melhor	média	desvio	tempo (s)
PFCM1	73,68	73,68	0	0,035	73,68	73,68	0	0,024
PFCM2	62,50	62,50	0	0,014	62,50	62,50	0	0,007
PFCM3	79,59	79,59	0	0,037	79,59	79,59	0	0,011
PFCM4	76,92	76,92	0	0,026	76,92	76,92	0	0,008
PFCM5	53,13	53,13	0	0,014	53,13	53,13	0	0,041
PFCM6	70,37	70,37	0	0,091	70,37	70,37	0	0,023
PFCM7	68,29	68,29	0	0,190	68,29	68,29	0	0,038
PFCM8	85,25	85,25	0	0,199	85,25	85,25	0	0,043
PFCM9	58,72	58,72	0	0,082	58,72	58,72	0	0,020
PFCM10	70,59	70,59	0	0,317	70,59	70,59	0	0,052
PFCM11	92,00	92,00	0	0,435	92,00	92,00	0	0,089
PFCM12	67,65	67,65	0	0,745	69,86	69,86	0	0,149
PFCM13	69,33	69,33	0	0,736	69,33	69,33	0	0,151
PFCM14	50,53	50,34	0,53	1,204	51,96	51,96	0	0,264
PFCM15	67,83	67,83	0	1,213	67,83	67,83	0	0,305
PFCM16	55,90	55,23	0,42	2,250	56,52	56,37	0,29	0,678
PFCM17	54,05	54,05	0	0,793	54,46	54,46	0	0,237
PFCM18	42,62	42,40	0,39	1,528	42,96	42,93	0,05	0,353
PFCM19	49,22	49,11	0,16	1,720	49,65	49,65	0	0,690
PFCM20	76,54	76,47	0,16	1,499	76,54	76,54	0	0,392
PFCM21	58,15	58,15	0	1,427	58,15	58,15	0	0,406
PFCM22	100,00	100,00	0	0,106	100,00	100,00	0	0,365
PFCM23	85,11	85,11	0	3,854	85,11	85,11	0	0,960
PFCM24	73,51	73,51	0	5,296	73,51	73,51	0	1,682
PFCM25	51,88	51,73	0,26	4,733	51,97	51,95	0,04	1,347
PFCM26	46,98	46,95	0,06	5,216	47,06	47,04	0,03	1,547
PFCM27	44,59	44,57	0,03	6,864	44,87	44,84	0,09	2,139
PFCM28	54,01	53,93	0,13	2,466	54,27	54,27	0	0,694
PFCM29	45,57	45,53	0,08	9,208	45,57	45,57	0	2,756
PFCM30	58,90	58,77	0,20	7,672	58,90	58,86	0,13	2,296
PFCM31	59,66	59,66	0	12,339	59,66	59,66	0	3,556
PFCM32	50,51	50,43	0,13	10,254	50,51	50,51	0	3,354
PFCM33	42,64	42,64	0	17,266	43,37	43,22	0,31	4,735
PFCM34	59,39	59,39	0	5,059	59,39	59,39	0	1,322
PFCM35	84,03	84,03	0	23,240	84,03	84,03	0	5,810

Analisando os resultados desta bateria de testes, nota-se um leve aumento no tempo computacional em comparação com a anterior. Nesta, enquanto o GRASP-RVND apresenta um tempo máximo em torno 26 segundos na maior instância, o ILS-RVND apresenta um tempo máximo em torno de 7 segundos.

Este aumento pode ser atribuído a dois fatores relacionados com a restrição de *singleton*. Primeiramente, há um aumento no número de arestas a serem verificadas no método construtivo, já que quando há a restrição, arestas que ligam as folhas da árvore geradora T não serão verificadas, pois sua remoção iria construir soluções com *singleton*. Além disso, há um aumento no número de células formadas nas instâncias, pois com a possibilidade de termos células *singleton*, temos um maior número de vértices candidatos na fase de busca local.

Tabela 5.3: Teste de desempenho para o PFCM-SRS

Instância	GRASP-RVND				ILS-RVND			
	melhor	média	desvio	tempo (s)	melhor	média	desvio	tempo (s)
PFCM1	75,00	75,00	0	0,032	75,00	75,00	0	0,026
PFCM2	69,57	69,57	0	0,015	69,57	69,57	0	0,007
PFCM3	80,85	80,85	0	0,034	80,85	80,85	0	0,012
PFCM4	79,17	79,17	0	0,030	79,17	79,17	0	0,007
PFCM5	60,87	60,87	0	0,013	60,87	60,87	0	0,044
PFCM6	70,83	70,83	0	0,086	70,83	70,83	0	0,024
PFCM7	69,44	69,44	0	0,208	69,44	69,44	0	0,034
PFCM8	85,25	85,25	0	0,202	85,25	85,25	0	0,044
PFCM9	58,72	58,72	0	0,080	58,72	58,72	0	0,018
PFCM10	75,00	75,00	0	0,290	75,00	75,00	0	0,053
PFCM11	92,00	92,00	0	0,433	92,00	92,00	0	0,088
PFCM12	74,24	74,16	0,27	0,855	74,24	74,24	0	0,150
PFCM13	72,86	72,86	0	0,817	72,86	72,86	0	0,145
PFCM14	53,33	53,33	0	1,357	53,33	53,33	0	0,286
PFCM15	69,92	69,92	0	1,362	69,92	69,92	0	0,302
PFCM16	57,96	57,87	0,18	2,072	57,96	57,96	0	0,664
PFCM17	57,73	57,73	0	0,753	57,73	57,73	0	0,243
PFCM18	42,96	42,96	0	1,663	43,26	43,20	0,13	0,319
PFCM19	50,41	50,41	0	1,872	50,41	50,41	0	0,769
PFCM20	79,38	79,38	0	1,681	79,38	79,38	0	0,457
PFCM21	58,70	58,70	0	1,532	58,70	58,70	0	0,430
PFCM22	100,00	100,00	0	0,108	100,00	100,00	0	0,437
PFCM23	85,11	85,11	0	4,459	85,11	85,11	0	0,970
PFCM24	73,51	73,51	0	6,290	73,51	73,51	0	1,897
PFCM25	53,29	53,29	0	4,281	53,29	53,29	0	1,274
PFCM26	48,61	48,59	0,02	5,401	48,63	48,62	0,01	1,716
PFCM27	46,26	46,22	0,06	7,303	46,26	46,26	0	2,186
PFCM28	54,55	54,55	0	2,886	54,55	54,55	0	0,800
PFCM29	47,41	47,30	0,17	9,888	47,41	47,37	0,11	3,302
PFCM30	62,94	62,94	0	7,769	62,94	62,94	0	2,142
PFCM31	59,77	59,77	0	12,979	59,77	59,77	0	3,777
PFCM32	50,83	50,83	0	10,356	50,83	50,83	0,14	3,148
PFCM33	47,38	47,38	0	16,649	47,38	47,38	0	5,606
PFCM34	61,16	61,16	0,25	5,794	61,16	61,16	0	1,441
PFCM35	84,03	84,03	0	26,588	84,03	84,03	0	6,955

Outro aspecto observado é quanto à qualidade da solução, e em contraste com a bateria de testes do PFCM-CRS onde o desempenho computacional foi superior, esta apresentou resultados mais robustos. Nesta bateria de testes, em apenas 4 instâncias o ILS-RVND apresentou variações nas soluções, enquanto que no GRASP-RVND isso ocorre em 6 instâncias. Por outro lado, observa-se ainda a superioridade do ILS-RVND em relação ao GRASP-RVND neste quesito, apresentando um menor número de instâncias com variações.

5.3 Comparação com os Métodos da Literatura

Nesta seção, apresentaremos os resultados computacionais do GRASP-RVND e ILS-RVND em comparação com alguns métodos da literatura do PFCM sobre as instâncias apresentadas na Seção 5.1. Para tal, utilizamos a medida *gap*, para verificar o quão distante a solução encontrada por cada algoritmo está da melhor solução conhecida de cada instância. Esta medida é calculada por base na seguinte equação:

$$\%gap = \frac{\mu(S^*) - \mu(S)}{\mu(S^*)} \quad (5.1)$$

onde, $\mu(S)$ corresponde à eficácia de agrupamento da solução S encontrada e $\mu(S^*)$ da melhor solução conhecida, podendo ser a ótima caso a instância tenha ótimo comprovado pelo trabalho de Pinheiro [54], ou a melhor dentre as demais soluções existentes na literatura (ver as colunas **MSol. CRS** e **MSol. SRS** da Tabela 5.1).

Nas tabelas apresentadas nessa seção, cada instância identificada com um asterisco possui ótimo comprovado, e portanto os métodos avaliados que alcançaram *gap* igual a 0% nessas instâncias encontraram uma solução ótima para esta. Nas demais, esses *gaps* significam que o método alcançou o melhor resultado já conhecido. Conseqüentemente, a presença de um *gap* negativo indica que um resultado melhor que o conhecido foi alcançado.

Primeiramente temos a Tabela 5.4, que apresenta um comparativo entre os métodos GRASP-RVND, ILS-RVND e alguns métodos da literatura do PFCM-CRS. Neste comparativo, é apresentado para cada instância o *gap* entre o melhor valor de solução obtido pelos métodos GRASP-RVND e ILS-RVND, baseado nos testes descritos na Seção 5.2, e o melhor valor de solução existente, dado na coluna **MSol. CRS** da Tabela 5.1. É também apresentado para cada instância, o *gap* entre o melhor valor de solução existente e os melhores valores de solução dos seguintes métodos da literatura:

SACF: *Simulated Annealing for Cell Formation* [76]

WFACF: *Water Flow-like Algorithm for Cell Formation* [78]

EA: *Evolutionary Algorithm* [27]

Pelos resultados da Tabela 5.4, observa-se que o ILS-RVND alcança os melhores resultados existentes em 31 das 35 instâncias. Das que possuem um valor ótimo conhecido, apenas na instância PFCM30 esse valor não foi alcançado, com um *gap* de 0,04%. Em contrapartida, na instância PFCM16 o valor ótimo foi encontrado quando nenhum outro método heurístico o encontrou. O maior *gap* apresentado foi alcançado na instância PFCM29 com 1,06%, e a média dos *gaps* ficou em 0,024%, em grande parte devido à

Tabela 5.4: Resultado geral do PFCM-CRS

Instância	SACF	WFACF	EA	GRASP-RVND	ILS-RVND
PFCM01*	0,00	0,00	0,00	0,00	0,00
PFCM02*	0,00	0,00	0,00	0,00	0,00
PFCM03*	0,00	0,00	0,00	0,00	0,00
PFCM04*	0,00	0,00	0,00	0,00	0,00
PFCM05*	0,00	0,00	0,00	0,00	0,00
PFCM06*	0,00	0,00	0,00	0,00	0,00
PFCM07*	0,00	0,00	0,00	0,00	0,00
PFCM08*	0,00	0,00	0,00	0,00	0,00
PFCM09*	0,53	0,00	0,00	0,00	0,00
PFCM10*	0,00	0,00	0,00	0,00	0,00
PFCM11*	0,00	0,00	0,00	0,00	0,00
PFCM12*	3,22	0,00	0,00	3,16	0,00
PFCM13*	0,00	0,00	0,00	0,00	0,00
PFCM14*	0,00	0,00	² 0,00	2,77	0,00
PFCM15*	0,00	0,00	0,00	0,00	0,00
PFCM16*	1,34	1,11	2,94	1,11	0,00
PFCM17*	1,51	0,00	0,00	0,75	0,00
PFCM18	1,44	0,00	0,00	0,79	0,00
PFCM19*	0,00	0,08	0,00	0,87	0,00
PFCM20*	0,52	0,00	0,42	0,00	0,00
PFCM21*	3,63	0,00	0,14	0,00	0,00
PFCM22*	0,00	0,00	0,00	0,00	0,00
PFCM23*	0,00	0,00	0,00	0,00	0,00
PFCM24*	0,00	0,00	0,00	0,00	0,00
PFCM25*	0,17	0,00	0,00	0,17	0,00
PFCM26	6,19	0,00	0,65	0,82	0,65
PFCM27	1,40	0,00	0,00	0,62	0,00
PFCM28	1,99	0,00	0,00	0,48	0,00
PFCM29	11,95	0,00	3,13	1,06	1,06
PFCM30*	0,61	¹ -0,98	0,78	0,07	0,07
PFCM31*	¹ -0,57	¹ -0,57	0,00	0,00	0,00
PFCM32	0,00	0,00	0,00	0,00	0,00
PFCM33	6,31	0,00	0,00	0,00	-1,71
PFCM34	2,71	0,00	5,73	0,77	0,77
PFCM35*	0,00	0,00	0,00	0,00	0,00

Notas:

1 - valor maior que o ótimo: a instância empregada pelo método está errada

2 - valor corrigido em relação ao apresentado no artigo original: a instância empregada está correta mas o cálculo da eficácia está errado

instância PFCM33, em que foi obtido um *gap* negativo. Entre as 4 instâncias que não alcançaram o melhor resultado existente, em duas houve alguma variação nos valores encontrados. Nas demais, houveram 4 instâncias em que ocorreu alguma variação nos valores.

No GRASP-RVND os resultados foram ligeiramente menos expressivos. Apenas em 21 instâncias os melhores resultados conhecidos são alcançados e em nenhum momento consegue-se um resultado melhor que o ILS-RVND. Na instância PFCM16, o método não alcança o valor ótimo, mas alcança o melhor valor encontrado na literatura. Isto pode

ser um indício de que no ILS-RVND as perturbações aplicadas foram determinantes para que a solução ótima fosse encontrada. Quanto à qualidade da solução, o método obteve um *gap* máximo de 3,16%, considerado relativamente alto, mas em geral a maioria dos *gaps* ficou abaixo de 1%, e a média dos *gaps* em 0,384%. Entre as 14 instâncias onde não foi alcançado o melhor resultado existente, em 10 houve alguma variação nos valores encontrados.

Dos métodos avaliados, o que obteve os melhores resultados foi o WFACF. Neste, apenas nas instâncias PFCM16 e PFCM19 o valor ótimo conhecido não foi alcançado. No entanto, o WFACF obteve um valor maior que o ótimo nas instâncias PFCM30 e PFCM31. Esta incoerência foi atribuída a alguns erros que foram observados nas instâncias utilizadas pelos autores do artigo (demonstrados logo mais no apêndice B), e portanto os resultados dessas duas instâncias para o WFACF foram desconsiderados. Com isso, em comparação com o ILS-RVND, o WFACF obtém melhores resultados em 3 instâncias, e piores em 2.

Analisaremos agora os resultados apresentados na Tabela 5.5, um comparativo entre os métodos GRASP-RVND, ILS-RVND e alguns métodos da literatura do PFCM-SRS. Esta tabela é apresentada de forma semelhante à anterior. No entanto, os métodos comparados são:

SCM-BMCF: *Similarity Coefficients Method whit Boltzmann function and Mutation operator for Cell Formation* [77]

SA: *Simulated Annealing* [52]

HMGA: *Hybrid Method Genetic Algorithm* [20]

Pelos resultados da Tabela 5.5 podemos observar que em 25 das 35 instâncias o ILS-RVND alcança os melhores resultados da literatura, sendo que em apenas 4 o valor ótimo conhecido não foi alcançado. Para o GRASP-RVND o comportamento foi semelhante, onde nos mesmos conjuntos de instâncias os melhores resultados foram alcançados.

Entre o GRASP-RVND e o ILS-RVND ocorreram diferenças em apenas duas instâncias. Nas instâncias PFCM18 e PFCM26, o ILS-RVND obteve um resultado melhor que o GRASP-RVND. Isto é um indício de que quando a restrição de *singleton* não é aplicada, o ILS-RVND passa a ter um comportamento mais similar ao GRASP-RVND. Contudo, há uma melhora no que diz respeito aos *gaps*. No GRASP-RVND, o maior *gap* foi apresentado na instância PFCM18, com um valor de 2,30%; a média foi de 0,204%. Já no ILS-RVND, o maior *gap* foi apresentado na instância PFCM18, com um valor de 1,61%; a média foi de 0,183%.

Dos métodos avaliados, o que obteve os melhores resultados foi o SA, que obteve o melhor resultado em 26 instâncias, sendo que em apenas 4 o valor ótimo conhecido não foi alcançado. Nos melhores resultados do SA, os métodos propostos obtiveram resultado melhor apenas nas instâncias PFCM5 e PFCM31, enquanto o SA obteve resultado melhor

Tabela 5.5: Resultado geral do PFCM-SRS

Instância	SCM-BMCF	SA	HMGA	GRASP-RVND	ILS-RVND
PFCM01*	-	0,00	¹ -9,80	0,00	0,00
PFCM02*	0,00	0,00	0,00	0,00	0,00
PFCM03*	1,56	0,00	1,56	0,00	0,00
PFCM04*	2,84	0,00	2,84	0,00	0,00
PFCM05*	0,00	1,43	0,00	0,00	0,00
PFCM06*	0,00	0,00	0,00	0,00	0,00
PFCM07*	-	0,00	0,00	0,00	0,00
PFCM08*	0,00	0,00	0,00	0,00	0,00
PFCM09*	0,00	0,00	0,00	0,00	0,00
PFCM10*	0,00	0,00	0,00	0,00	0,00
PFCM11*	0,00	0,00	0,00	0,00	0,00
PFCM12*	-	0,00	2,94	0,00	0,00
PFCM13*	1,41	0,00	1,41	0,00	0,00
PFCM14*	-	0,00	0,13	0,00	0,00
PFCM15*	-	0,00	0,56	0,00	0,00
PFCM16*	2,86	0,14	0,88	0,14	0,14
PFCM17*	-	0,00	0,00	0,00	0,00
PFCM18	1,61	0,00	1,18	2,30	1,61
PFCM19*	0,00	0,00	0,00	0,79	0,79
PFCM20*	1,23	0,00	1,85	0,00	0,00
PFCM21*	2,01	0,00	1,38	0,15	0,15
PFCM22*	0,00	0,00	0,00	0,00	0,00
PFCM23*	0,00	0,00	0,00	0,00	0,00
PFCM24*	0,00	0,00	0,00	0,00	0,00
PFCM25	0,00	0,00	0,00	0,00	0,00
PFCM26	0,65	0,78	0,00	0,69	0,65
PFCM27	0,92	1,25	0,00	0,69	0,69
PFCM28	-	0,00	0,00	0,49	0,49
PFCM29	-	0,00	1,30	0,57	0,57
PFCM30*	0,71	0,29	¹ -0,13	0,16	0,16
PFCM31*	¹ -0,59	0,18	¹ -0,59	0,00	0,00
PFCM32	0,00	0,55	0,00	0,00	0,00
PFCM33	-	0,00	0,38	1,15	1,15
PFCM34	-	0,00	0,87	0,00	0,00
PFCM35*	0,00	0,00	0,00	0,00	0,00

Nota:

1 - valor maior que o ótimo: a instância empregada pelo método está errada

em 6 outras instâncias. Assim como no PFCM-CRS, houve incoerências quanto aos valores encontrados na literatura, atribuídos pelos mesmos motivos e explicados com maiores detalhes no apêndice B. No HMGA estes problemas ocorreram nas instâncias PFCM1, PFCM30 e PFCM31, enquanto que no SCM-BMCF apenas foi identificado problema na instância PFCM31.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho, foi apresentado um estudo sobre o PFCM e foram propostos dois algoritmos baseados nas meta-heurísticas GRASP e ILS em composição com o método de busca local RVND. Para tal, foi definido um método construtivo aplicado tanto de forma aleatória quanto de forma gulosa; além disso, foram definidas três estruturas de vizinhança para compor o método RVND e 3 perturbações exclusivas ao ILS proposto. Em ambos os métodos, foram realizados estudos quanto à influência que a restrição de *singleton* causa na qualidade da solução.

Para a validação desses algoritmos, um conjunto de 35 instâncias foi utilizada, e os resultados obtidos foram comparados entre si e entre as abordagens da literatura. Os experimentos computacionais mostram que as soluções do ILS-RVND apresentaram a melhor qualidade média e o menor tempo médio nos testes, quando comparadas às do GRASP-RVND, tanto para o PFCM-CRS quanto para o PFCM-SRS.

Como trabalhos futuros, almeja-se aprimorar o ILS, investindo em novos métodos de busca local, estruturas de vizinhança e perturbações. Deseja-se também aprimorar o método construtivo, incorporando uma estrutura de dados eficiente que indexe as remoções de arestas da árvore de entrada T , de forma que quando um conjunto de remoções já tiver sido apresentado, possa-se evitar o recálculo da solução associada ao grafo resultante.

Além disso, pretende-se tratar outras variantes do problema, considerando os diversos aspectos práticos envolvidos na formação das células. Neste trabalho, limitamos o tamanho mínimo da célula, mas também poderia-se limitar o tamanho máximo da célula. Outro aspecto que também poderia ser tratado é quando se tem um número fixo de células a serem formadas. Na prática, isso representaria um número fixo de galpões, em que se quer a melhor associação entre as células e as máquinas. Nesta variante é possível ainda limitar de forma individual as dimensões de cada uma das células.

Apêndice A

Instâncias

							1 2 3 4 5											1 2 3 4 5 6 7													
							1	1	1	0	1	0	1	0							1	1	0	1	0	0	0	0	0		
							2	1	0	0	1	0	0	1	0							2	1	1	0	0	0	0	0	0	0
							3	1	1	0	1	0	1	0	0							3	0	0	1	1	1	0	0	0	0
							4	0	1	1	0	1	0	1	0							4	0	0	0	0	1	1	0	0	0
							5	1	0	0	1	0	1	0	0							5	0	0	0	0	0	0	1	1	0
							6	1	1	0	1	0	1	0	0							6	1	1	0	0	0	0	0	0	0
							7	0	1	1	0	0	1	0	0							7	0	0	1	1	0	0	0	0	0
							8	1	1	0	1	0	1	0	0							8	0	0	0	0	0	0	0	0	1
							9	0	0	0	0	0	1	0	0							9	0	1	0	0	0	0	0	0	0
							10	0	1	1	0	1	0	1	0							10	0	0	0	0	0	1	1	0	0
							11	1	1	0	1	0	1	0	0							11	0	0	1	0	1	0	0	0	0
							12	1	1	0	1	0	1	0	0							12	1	1	0	0	0	0	0	0	1
							13	1	1	0	1	0	1	0	0							13	1	1	0	1	0	1	0	1	0
							14	1	0	0	1	0	1	0	0							14	1	0	0	1	0	1	0	1	0
							15	0	1	1	0	1	0	1	0							15	0	1	1	0	1	0	1	0	0
							16	1	0	0	1	0	1	0	0							16	1	0	0	1	0	1	0	0	0
							17	1	0	0	1	0	1	0	0							17	1	0	0	1	0	1	0	1	0
							18	0	1	1	0	1	0	0	1							18	0	1	1	0	1	0	1	0	0

PFCM01

PFCM02

PFCM03

PFCM04

PFCM05

											1 2 3 4 5 6 7 8																			1 2 3 4 5 6 7 8															
											1	0	0	0	0	1	1	0	0	0												1	1	0	0	0	1	1	0	1	0	1			
											2	1	0	1	0	0	0	0	0	0												2	0	1	0	0	0	1	1	0	1	0	1		
											3	1	1	0	1	0	0	1	1	0												3	1	1	0	1	1	0	0	1	1	0	0	1	
											4	0	1	0	1	0	0	1	1	0												4	1	1	0	1	0	0	0	0	1	1	0	0	1
											5	0	0	0	0	1	1	0	0	0												5	0	0	1	0	0	1	1	1	0	1	1	0	0
											6	0	1	0	1	1	0	1	0	1												6	0	1	1	0	1	0	1	0	1	0	1	0	0
											7	0	0	0	1	0	0	1	0	0												7	0	1	1	1	0	1	1	0	1	1	0	1	0
											8	1	0	0	0	0	0	1	1	0												8	0	0	1	1	1	1	1	0	0	0	0	1	1
											9	0	0	1	1	1	1	1	0	0												9	1	1	0	1	0	0	0	0	1	0	0	0	1
											10	0	0	0	0	1	1	0	0	1												10	1	0	0	1	1	1	0	1	0	1	1	0	1
											11	0	1	0	0	0	0	1	1	0												11	0	1	1	0	0	1	1	0	1	1	0	1	0
											12	0	0	0	0	0	0	0	1	1												12	0	0	0	0	0	0	0	1	1	0	0	1	1

PFCM06

PFCM07

PFCM08

PFCM09

```

1 1001010000
2 0000000011
3 0100001011
4 0100000011
5 0010000000
6 0010000100
7 0000110000
8 0100001010
9 0000000100
10 1000110000

```

PFCM10

```

1 0011010000
2 1000001001
3 0100100100
4 0001010010
5 0100100100
6 0010010010
7 0000001001
8 0100100100
9 0011010010
10 1000001001
11 1000001001
12 1000001001
13 0100100100
14 0011010010
15 0100100100

```

PFCM11

```

1 00000101000000
2 00011010000000
3 00011010000000
4 01100000011000
5 01100000001000
6 00000001100000
7 10000000000010
8 10000010000110
9 00000000000110
10 00000101100001
11 00000100000001
12 00000101100000
13 00000001000001
14 00000101000000
15 00000101100001
16 00000101000000
17 00011010000000
18 00000000000010
19 00010000000000
20 00011010000000
21 00100000000100
22 00011000000010
23 00000000010000

```

PFCM12

```

1 00011010000000
2 00011010000000
3 01100000011000
4 01100000001000
5 00000001100000
6 10000000000010
7 10000010000110
8 00000000000110
9 00000101100001
10 00000101000000
11 00000100000001
12 00000101100000
13 00000000100001
14 00000101000000
15 00000101100001
16 00000101000000
17 00011010000000
18 00000000000010
19 00010000000000
20 00011010000000
21 00100000000100
22 00000101000000
23 00011000000010
24 00000000011000

```

PFCM13

```

1 1010001100000100
2 0000010001011000
3 1000010100010000
4 1010000000000100
5 0000000000000010
6 1010000000000010
7 1000000110101000
8 1000000000001000
9 0000000001110011
10 1100000101000000
11 1000000000011001
12 0000000000000100
13 0000010100000000
14 0000000001110101
15 1000010000000000
16 0011000000000000
17 0110010001000001
18 1000010000000000
19 1011001000000000
20 0000000001110011
21 0000110000000000
22 1100000101000000
23 1100000101000000
24 0000001011110101

```

PFCM14

```

1 0110000000000000
2 0001001100110000
3 00001000000001000
4 1101001100010000
5 10000000010000010
6 00001000010000101
7 1001001100110001
8 0000100001000101
9 1001000100100000
10 01000000000001000
11 0000100001100100
12 10000001100110000
13 00100000000001000
14 0000100001000101
15 0000100001000101
16 0100000000000000
17 00000000000000100
18 1101001101110000
19 0000000010010000
20 0100000000011000
21 0000100001000100
22 1001001100010000
23 0010010010000010
24 0000100001000001
25 0010011010000010
26 0000100001000011
27 0010010010000111
28 0010000110000000
29 0010010000001010
30 10010010000101010

```

PFCM15


```

          1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0
2 0 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1
3 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0
4 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
5 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0
6 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0
7 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1
8 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0
9 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0
10 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
11 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
12 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0
13 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0
14 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0
15 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0
16 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0
18 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
19 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1 0
20 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
21 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0
22 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
23 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0
24 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0
25 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
26 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
27 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0
28 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
29 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
30 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
31 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
32 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
33 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0
34 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
35 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
36 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
37 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1
38 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1
39 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
40 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
41 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0
42 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
43 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0

```

PFCM16

```

          1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
2 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0
5 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
8 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0
9 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0
10 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
12 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0
13 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
15 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
16 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
17 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
18 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0
20 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0
21 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0
22 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
23 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0
24 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0

```

PFCM17

```

          1 1 1 1 1 1 1 1 1 1 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1
2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
4 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1
5 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
7 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
8 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
9 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
11 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
12 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
13 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0
14 0 0 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0
15 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
16 1 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1 0 0 0 0
17 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
18 0 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 1
20 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

PFCM18

```

          1 1 1 1 1 1 1 1 1 1 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 0 0 0 1 1 0
2 0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0
3 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1
4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
5 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0
6 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0
7 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0
8 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
10 1 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0
11 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
12 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1
13 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
14 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
15 1 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1
16 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1
17 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
18 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0
19 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0
20 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0
21 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
22 1 0 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 1
23 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0

```

PFCM19

```

          1 1 1 1 1 1 1 1 1 1 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
2 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0
3 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0
5 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0
7 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
8 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
9 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0
10 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0
11 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0
12 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0
13 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
14 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
15 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
16 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
18 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0
19 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
20 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0
22 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
23 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
24 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
25 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
26 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
27 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0
29 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0
31 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
33 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
34 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
35 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

PFCM20

```

          1 1 1 1 1 1 1 1 1 1 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
2 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0
3 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0
7 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
15 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
18 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
20 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
22 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
23 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
24 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
25 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
26 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
27 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
29 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
30 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
32 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
33 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
34 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
35 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0

```

PFCM21


```

1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
2 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0
4 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
5 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0
6 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
7 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0
11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0
12 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
13 0 0 1 0 0 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
15 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
16 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
17 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
18 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
19 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
20 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
22 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
23 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
24 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
25 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
26 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
27 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
28 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
29 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
30 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
32 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
33 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
34 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
35 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
36 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
37 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
38 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
39 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
41 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
42 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
43 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
44 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
45 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
46 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0

```

PFCM29

```

1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1
2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
5 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
9 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
10 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
11 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
14 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
15 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
16 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0
17 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
18 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
20 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
21 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1
22 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1
23 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
24 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
25 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
26 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
27 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0
28 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
29 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
31 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
32 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
33 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
34 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
35 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
36 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
37 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
38 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
39 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
40 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
41 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

PFCM30

Apêndice B

Erros encontrados na literatura do PFCM

B.1 Instâncias Erradas

James et al. [30]:

- PFCM01: 0 em (M_5, P_5) (Máquina 5, Parte 5) e (M_6, P_6) ;
- PFCM30: 0 em (M_5, P_5) , (M_{28}, P_{29}) , (M_{29}, P_9) , (M_{27}, P_8) e 1 em (M_{19}, P_9) , (M_{19}, P_{29}) , (M_{19}, P_8) .

Wu et al. [77]:

- PFCM31: 0 em (M_{15}, P_{27}) .

B.2 Resultados Incorretos

Artigos com resultados incorretos (sem restrição de *singleton*):

Wu et al. [76]: PFCM31

Mesmo não apresentado, a solução errada é apresentada no próximo artigo do mesmo autor [77].

Wu et al. [77]: PFCM31

A solução errada referente a esta instância foi apresentada neste artigo, na seção das soluções com restrição de *singleton*.

Mahdavi et al. [41]: PFCM31

Referencia Wu et al. [77], onde a solução errada foi apresentada.

James et al. [30]: PFCM01 e PFCM30

As soluções erradas referente a estas instâncias foram apresentadas neste artigo.

Elbenani et al. [20]: PFCM01, PFCM30 e PFCM31

Referencia James et al. [30] (erro na PFCM01 e PFCM30) e Mahdavi et al. [41] (erro na PFCM31).

Artigos com resultados incorretos (com restrição de *singleton*):

Wu et al. [77]: PFCM30

A solução errada referente a esta instância foi apresentada neste artigo, na seção das soluções com restrição de *singleton*.

Wu et al. [78]: PFCM30

Referencia Wu et al. [77], onde a solução errada foi apresentada.

Bibliografia

- [1] ASKIN, R. G. ; CRESSWELL, S. H. ; GOLDBERG, J. B. ; VAKHARIA, A. J. A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research* 29, 6 (1991), 1081-1100.
- [2] ASKIN, R. G. ; STANDRIDGE, C. R. *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, 1993.
- [3] ASKIN, R. G. ; SUBRAMANIAN, S. P. A cost-based heuristic for group technology configuration. *International Journal of Production Research* 25 (1987), 101.
- [4] ASKIN, R. J. ; VAKHARIA, A. J. *Automated Factory Handbook: Technology and Management*. TAB Books, Inc., 1990, *cap.* Group technology - cell formation and operation, pp. 317–366.
- [5] ATEME-NGHEMA, H. ; DAO, T. Optimum design of cellular manufacturing systems using hopfield neural network approach: the first step. / Department of Mechanical Engineering École de Technologie Supérieure, University of Quebec. 1997. – Forschungsbericht.
- [6] BEN-ARIEH, D. ; SREENIVASAN, R. Information analysis in a distributed dynamic group technology method. *International Journal of Production Economics* 60-61, 1 (1999), 427–432.
- [7] BOCTOR, F. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research* 29, 2 (1991), 343–356.
- [8] BOE, W. J. ; CHENG, C. H. A close neighbour algorithm for designing cellular manufacturing systems. *International Journal of Production Research* 29, 10 (1991), 2097–2116.
- [9] BURBIDGE, J. L. Production Flow Analysis. *Production Engineer* 50, 4/5 (1971), 139–152.
- [10] CARRIE, A. S. Numerical taxonomy applied to group technology and plant layout. *International Journal of Production Research* 11, 4 (1973), 399-416.

-
- [11] CHAN, H. M. ; MILNER, D. A. Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems* 1 (1982), 63.
- [12] CHANDRASEKHARAN, M. P. ; RAJAGOPALAN, R. An Ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Reseach* 24, 2 (1986), 451–464.
- [13] CHANDRASEKHARAN, M. P. ; RAJAGOPALAN, R. MODROC: An extension of rank order clustering for group technology. *International Journal of Production Reseach* 24, 5 (1986), 1221–1233.
- [14] CHANDRASEKHARAN, M. P. ; RAJAGOPALAN, R. ZODIAC - an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research* 25, 6 (1987), 835–850.
- [15] CHANDRASEKHARAN, M. P. ; RAJAGOPALAN, R. Groupability an analysis of the properties of binary data matrices for group technology. *International Journal of Production Research* 27 (1989), 1035.
- [16] CHENG, C. H. ; GUPTA, Y. P. ; LEE, W. H. ; WONG, K. F. A TSP-based Heuristic for Forming Machine Groups and Part Families. *International Journal of Production Research* 36, 5 (1998), 1325–1337.
- [17] CHOUBINEH, F. A framework for the design of cellular manufacturing systems. *International Journal of Production Research* 26, 7 (1988), 1161-1172.
- [18] CHU, C. H. Cluster analysis in manufacturing cellular formation. *Omega* 17, 3 (1989), 289–295.
- [19] CORMEM, T. H. ; LEISERSON, C. E. ; RIVEST, R. L. ; STEIN, C. *Introduction to Algorithms*. The MIT Press, 2001.
- [20] ELBENANI, B. ; FERLAND, J. A. ; BELLEMARE, J. Genetic algorithm and large neighbourhood search to solve the cell formation problem. *Expert Systems with Applications* 39, 3 (2012), 2408–2414.
- [21] FEO, T. ; RESENDE, M. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6 (1995), 109–133.
- [22] FERREIRA RIBEIRO, J. F. ; PRADIN, B. A methodology for cellular manufacturing design. *International Journal of Production Research* 31, 1 (1993), 235-250.
- [23] FLANDERS, R. E. Design, manufacture and production control of a standard machine. *Transactions of the American Society of Mechanical Enginneers* 46 (1924), 691–738.

-
- [24] FOULDS, L. R. ; FRENCH, A. P. ; WILSON, J. M. The sustainable cell formation problem: manufacturing cell creation with machine modification costs. *Computers & Operations Research* 33, 4 (2006), 1010–1032.
- [25] GLOVER, F. Tabu Search - Part I. *ORSA Journal on Computing* 1, 3 (1989), 190–206.
- [26] GLOVER, F. Tabu Search - Part II. *ORSA Journal on Computing* 2, 1 (1990), 4–32.
- [27] GONÇALVES, J. ; RESENDE, M. An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering* 47, 2-3 (2004), 247–273.
- [28] HARHALAKIS, G. ; HILGER, J. ; PROTH, J. M. Manufacturing cell formation in the case of multi-manufacturing processes / University of Maryland. 1990. – Forschungsbericht.
- [29] HSU, C. P. *Similarity Coefficient Approaches to Machine component Cell Formation in Cellular Manufacturing: A Comparative Study*, University of Wisconsin, Diss., 1990.
- [30] JAMES, T. L. ; BROWN, E. C. ; KEELING, K. B. A hybrid grouping genetic algorithm for the cell formation problem. *Comput. Oper. Res.* 34, 7 (2007), 2059–2079.
- [31] JOINES, J. A. ; CULBRETH, C. T. ; KING, R. E. Manufacturing Cell Design: An Integer Programming Model Employing Genetic Algorithms. *IIE Transactions* 28 (1996), 69–85.
- [32] KING, J. R. ; NAKORNCHAI, V. Machine-component group formation in group technology: review and extension. *International Journal of Production Research* 20, 2 (1982), 117–133.
- [33] KING, J. Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research* 18, 2 (1980), 213–232.
- [34] KULKARNI, U. R. ; KIANG, M. Y. Dynamic grouping of parts in flexible manufacturing systems – a self-organizing neural networks approach. *European Journal of Operational Research* 84, 1 (1995), 192–212.
- [35] KUMAR, C. S. ; CHANDRASEKHARAN, M. P. Group efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research* 28, 2 (1990), 233–243.

- [36] KUMAR, K. R. ; KUSIAK, A. ; VANNELLI, A. Grouping of parts and components in flexible manufacturing systems. *European Journal of Operational Research* 24, 3 (1986), 387–397.
- [37] KUMAR, K. R. ; VANNELLI, A. Strategic subcontracting for efficient disaggregated manufacturing. *International Journal of Production Research* 25, 12 (1987), 1715–1728.
- [38] KUSIAK, A. *Intelligent Manufacturing Systems*. Prentice Hall, 1992.
- [39] KUSIAK, A. ; CHO, M. Similarity coefficient algorithms for solving the group technology problem. *International Journal of Production Research* 30, 11 (1992), 2633–2646.
- [40] KUSIAK, A. ; CHOW, W. Efficient solving of the group technology problem. *Journal of Manufacturing Systems* 6, 2 (1987), 117–124.
- [41] MAHDAVI, I. ; PAYDAR, M. M. ; SOLIMANPUR, M. ; HEIDARZADE, A. Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Syst. Appl.* 36, 3 (2009), 6598–6604.
- [42] MASNATA, A. ; SETTINERI, L. An application of fuzzy clustering to cellular manufacturing. *International Journal of Production Research* 35, 4 (1997), 1077–1094.
- [43] MCAULEY, J. Machine grouping for efficient production. *Production Engineer* 51, 2 (1972), 53–57.
- [44] MCCORMICK, W. T. ; SCHWEITZER, P. J. ; WHITE, T. W. Problem Decomposition and Data Reorganization by a Clustering Technique. *Operations Research* 20, 5 (1972), 993–1009.
- [45] MILTENBURG, J. ; ZHANG, W. A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology. *Journal of Operations Management* 10, 1 (1991), 44–72.
- [46] MITROFANOV, S. P. *The Scientific Principles of Group Technology*. National Lending Library for Science and Technology, 1966.
- [47] MLADENOVIC, N. ; HANSEN, P. Variable Neighborhood Search. *Computers and Operations Research* 24, 11 (1997), 1097–1100.
- [48] MOSIER, C. ; TAUBE, L. The facets of group technology and their impacts on implementation — A state-of-the-art survey. *Omega* 13, 5 (1985), 381–391.

- [49] MOSIER, C. ; TAUBE, L. Weighted similarity measure heuristics for the group technology machine clustering problem. *Omega* 13, 6 (1985), 577-579.
- [50] MUKATTASH, A. M. ; ADIL, M. B. ; TAHBOUB, K. K. A modified cell formation group measure. In *International Conference on Computer and Information Engineering*, (2001), pp.
- [51] MUNHOZ, P. L. A. *Um algoritmo baseado em Iterated Local Search para o Problema de Roteamento de Veículos Periódico*, Universidade Federal Fluminense, Dissertação de Mestrado, 2012.
- [52] PAILLA, A. ; TRINDADE, A. R. ; PARADA, V. ; OCHI, L. S. A numerical comparison between simulated annealing and evolutionary approaches to the cell formation problem. *Expert Syst. Appl.* 37 (2010), 5476–5483.
- [53] PAPAIOANNOU, G. ; WILSON, J. M. The evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and direction for future search. *European Journal of Operational Research* 206 (2010), 509–521.
- [54] PINHEIRO, R. G. S. *Método Exato para Biclusterização por Edição de Arestas e Aplicação em Formação de Células de Manufatura*, Universidade Federal Fluminense, Dissertação de Mestrado, 2012.
- [55] RAJAGOPALAN, R. ; BATRA, J. L. Design of cellular production systems A graph-theoretic approach. *International Journal of Production Research* 13, 6 (1975), 567-579.
- [56] RAO, H. ; GU, P. Design of cellular manufacturing systems: A neural network approach. *International Journal of Systems Automation: Research and Applications* 2, 4 (1993), 407–424.
- [57] RAVICHANDRAN, K. S. ; RAO, K. C. S. A New Approach to Fuzzy Part-Family Formation in Cellular Manufacturing Systems. *The International Journal of Advanced Manufacturing Technology* 18 (2001), 591–597.
- [58] RESENDE, M. ; RIBEIRO, C. *Metaheuristics: Progress as real problem solvers*. Springer, 2005, cap. GRASP whit path-relinking: recent advances and applications, pp. 29–63.
- [59] SANKARAN, S. ; RODIN, E. Y. Multiple objective decision making approach to cell formation: A goal programming model. *Mathematical and Computer Modelling* 13, 9 (1990), 71 - 81.

- [60] SARKER, B. ; KHAN, M. A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure. *IIE Transactions* 33 (2001), 11-27.
- [61] SEIFODDINI, H. ; WOLFE, P. M. Application of the Similarity Coefficient Method in Group Technology. *IIE Transactions* 18, 3 (1986), 271–277.
- [62] SEIFODDINI, H. Single linkage versus average linkage clustering in machine cells formation applications. *Computers and Industrial Engineering* 16, 3 (1989), 419–426.
- [63] SELIM, H. M. ; ASKIN, R. G. ; VAKHARIA, A. J. Cell formation in group technology: Review, evaluation and directions for future research. *Computers & Industrial Engineering* 34, 1 (1998), 3 – 20. – Cellular manufacturing systems: Design, Analysis and Implementation.
- [64] SHAFER, S. M. *Group Technology and Cellular Manufacturing: A State of the Art Synthesis of Research and Practice*. Kluwer Academic Publishers, 1998, cap. Part-machine labour grouping: The problem and solution methods, pp. 131–152.
- [65] SHANKAR, R. ; VRAT, P. ; SOLEYMANPOUR, M. A transiently chaotic neural network approach to the design of cellular manufacturing. *International Journal of Production Research* 40, 10 (2002), 2225–2244.
- [66] SINGH, N. Design of cellular manufacturing systems: An invited review. *European Journal of Operational Research* 69, 3 (1993), 284 - 291. – Cellular Manufacturing Systems.
- [67] SRINIVASAN, G. A Clustering-Algorithm for Machine Cell-Formation in Group Technology Using Minimum Spanning-Trees. *International Journal of Production Research* 32, 9 (1994), 2149–2158.
- [68] SRINIVASAN, G. ; NARENDRAN, T. T. GRAFICS - a nonhierarchical clustering-algorithm for group technology. *International Journal of Production Research* 29, 3 (1991), 463–478.
- [69] SRINIVASAN, G. ; NARENDRAN, T. T. ; MAHADEVAN, B. Assignment model for the part-families problem in group technology. *International Journal of Production Research* 28, 1 (1990), 145–152.
- [70] STANFEL, L. E. Machine clustering for economic production. *Engineering Costs and Production Economics* 9, 1–3 (1985), 73–81.
- [71] TOMPKINS, J. A. *Facilities planning*. John Wiley & Sons, 1996.

- [72] VANNELLI, A. ; KUMAR, K. R. A method for finding minimal bottle-neck cells for grouping part-machine families. *International Journal of Production Research* 24, 2 (1986), 387-400.
- [73] VENUGOPAL, V. ; NARENDRAN, T. T. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers and Industrial Engineering* 22 (1992), 469-480.
- [74] WAGHODEKAR, P. H. ; SAHU, S. Machine-component cell formation in group technology: MACE. *International Journal of Production Research* 22, 6 (1984), 937-948.
- [75] WANG, J. A linear assignment clustering algorithm based on the least similar cluster representatives. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 29, 1 (1999), 100-104.
- [76] WU, T. H. ; CHANG, C. C. ; CHUNG, S. H. A simulated annealing algorithm for manufacturing cell formation problems. *Expert Systems with Applications* 34, 3 (2008), 1609-1617.
- [77] WU, T. H. ; CHANG, C. C. ; Y., Y. J. A hybrid heuristic algorithm adopting both Boltzmann function and mutation operator for manufacturing cell formation problems. *International Journal of Production Economics* 120, 2 (2009), 669-688.
- [78] WU, T. H. ; CHUNG, S. H. ; CHANG, C. C. A water flow-like algorithm for manufacturing cell formation problems. *European Journal of Operational Research* 205, 2 (2010), 346-360.
- [79] WU, T. H. ; LOW, C. ; WU, W. T. A tabu search approach to the cell formation problem. *The International Journal of Advanced Manufacturing Technology* 23 (2004), 916-924.
- [80] XAMBRE, A. R. ; VILARINDO, P. M. A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European Journal Of Operational Research* 151, 2 (2003), 434-446.
- [81] YIN, Y. ; YASUDA, K. Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *International Journal of Production Economics* 101, 2 (2006), 329-352.