

UNIVERSIDADE FEDERAL FLUMINENSE

IVAN CÉSAR MARTINS

**Algoritmos para o Problema de Formação de Células  
de Manufatura com Sequência de Operações e  
Processos Alternativos**

NITERÓI

2017

UNIVERSIDADE FEDERAL FLUMINENSE

IVAN CÉSAR MARTINS

Algoritmos para o Problema de Formação de  
Células de Manufatura com Sequência de  
Operações e Processos Alternativos

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

Orientadores:  
Luiz Satoru Ochi  
Fábio Protti

NITERÓI  
2017

# Algoritmos para o Problema de Formação de Células de Manufatura com Sequência de Operações e Processos Alternativos

Ivan César Martins

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

Aprovada por:

---

Prof. Luiz Satoru Ochi / IC-UFF (Orientador)

---

Prof. Fábio Protti / IC-UFF (Orientador)

---

Profª. Eliana Silva de Almeida / IC-UFAL

---

Prof. Virgílio José Martins Ferreira Filho / COPPE-UFRJ

---

Profª. Loana Tito Nogueira / IC-UFF

---

Profª. Simone de Lima Martins / IC-UFF

Niterói, 05 de maio de 2017.

# Resumo

O PROBLEMA DE FORMAÇÃO DE CÉLULAS DE MANUFATURA é um problema NP-difícil em que dado um conjunto de máquinas e de partes (de produtos) a serem processadas, deseja-se agrupar as máquinas em células e as partes em famílias de modo que cada família seja alocada a uma célula distinta, minimizando o número de deslocamento de partes entre as células ao mesmo tempo em que se maximiza o aproveitamento (número de operações) de cada célula. Este problema em si já é bem estudado e explorado na literatura, com centenas de artigos e heurísticas desenvolvidas nas últimas décadas. No entanto, a inclusão de diversos fatores práticos relevantes é uma preocupação mais recente que leva a um aumento na complexidade do problema, tornando difícil a inclusão de muitos desses fatores simultaneamente. Neste trabalho iremos tratar do Problema de Formação de Células de Manufatura com Sequência de Operações e Processos Alternativos, no qual são incluídas informações sobre a sequência de operações das partes juntamente com a possibilidade de escolha entre múltiplos planos de processamento. Para tal, propomos inicialmente um algoritmo baseado na meta-heurística ILS para a versão clássica do problema e em seguida desenvolvemos uma extensão deste algoritmo para a resolução do problema com a inclusão desses novos requisitos. O algoritmo proposto é comparado com os métodos existentes na literatura, apresentando resultados superiores em um tempo computacional baixo.

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

# Abstract

The CELL FORMATION PROBLEM is a 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. This problem is well studied and explored in the literature, with hundreds of papers and algorithms developed in the last decades. However, the inclusion of several relevant practical factors is a more recent concern that leads to an increase in the problem complexity, making it difficult to include many factors simultaneously. In this study, we solve the Cell Formation Problem considering Operation Sequence and Alternative Process Plans, where the sequence of operations are given and for each part one process plan must be chosen. We propose an algorithm based on ILS metaheuristic 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.

**Keywords:** Metaheuristics. Cellular Manufacturing. Group Technology.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	O Sistema de Manufatura Celular . . . . .	2
1.2	Contextualização do Problema . . . . .	3
1.3	Objetivos da Pesquisa . . . . .	5
1.4	Estrutura do Trabalho . . . . .	5
<b>2</b>	<b>Definições do Problema</b>	<b>7</b>
2.1	Tamanho Mínimo das Células . . . . .	8
2.2	Sequência de Operações e Rotas de Processamento . . . . .	10
2.3	Planos de Processamento . . . . .	11
2.4	Níveis de Alternatividade . . . . .	12
2.5	Qualidade da Solução . . . . .	15
<b>3</b>	<b>Revisão da Literatura</b>	<b>17</b>
3.1	Critérios de Otimalidade . . . . .	18
3.2	Técnicas Utilizadas . . . . .	20
3.2.1	Procedimentos Descritivos . . . . .	20
3.2.2	Clusterização . . . . .	21
3.2.3	Teoria dos Grafos . . . . .	23
3.2.4	Meta-heurísticas . . . . .	25
<b>4</b>	<b>Proposta</b>	<b>30</b>
4.1	Algoritmo ILS-RVND . . . . .	30
4.1.1	Formação das Sementes . . . . .	32
4.1.2	Construção de Soluções Iniciais . . . . .	32
4.1.3	Busca por Melhores Soluções . . . . .	34
4.1.4	Perturbação de Soluções . . . . .	36
4.2	Estendendo o ILS-RVND para a Resolução do PFCM-SOPA . . . . .	38
4.2.1	Representação do Problema . . . . .	38
4.2.2	Construção de Soluções Iniciais . . . . .	39
4.2.3	Estruturas de Vizinhança e Perturbações Adicionais . . . . .	41
<b>5</b>	<b>Resultados</b>	<b>43</b>
5.1	PFCM . . . . .	43
5.1.1	Instâncias . . . . .	43
5.1.2	Avaliação de Desempenho . . . . .	44
5.1.3	Comparação com a Literatura . . . . .	46
5.2	PFCM-SOPA . . . . .	48

---

5.2.1	Adaptação das Instâncias . . . . .	48
5.2.2	Comparação com a Literatura . . . . .	50
<b>6</b>	<b>Considerações Finais</b>	<b>54</b>
	<b>Bibliografia</b>	<b>63</b>

# Capítulo 1

## Introdução

Na indústria, a organização dos recursos de produção afeta diretamente os custos e a produtividade do sistema de manufatura. Uma alocação de recursos racional, definida na fase de planejamento desse sistema, contribui significativamente para o aumento na eficiência das operações envolvidas no processo de manufatura e conseqüentemente em uma redução dos custos relacionados à movimentação destes recursos. Em um mercado global cada vez mais competitivo, as indústrias de manufatura estão sob intensa pressão em responder às crescentes demandas do mercado, e desta forma precisam ser capazes de fornecer os produtos aos consumidores em tempo, enquanto mantém os baixos custos de produção e a alta qualidade da mercadoria.

Existem dois conceitos tradicionais usados na organização do sistema de manufatura de acordo com o fluxo do produto na linha de produção: a manufatura linear (*flow shop*) e a manufatura funcional (*job shop*). Cada uma apresenta um conjunto de vantagens e limitações referentes ao tempo de processamento, flexibilidade da produção, demanda do produto, utilização de maquinário, alocação de operadores e custos de manutenção.

Em geral, a manufatura funcional é designada quando há uma grande variedade de tipos de produtos a serem fabricados. Para tal, as máquinas são agrupadas funcionalmente, ou seja, de acordo com a semelhança entre os tipos de processos desempenhados: perfuração em um departamento, acabamento em outro, e assim por diante. Isto permite uma melhor utilização das máquinas, maior especialização no trabalho dos operadores e um controle de produção descentralizado. Em contrapartida, quando há alguma mudança na demanda de determinado produto ou um novo tipo de produto é designado, o alto tráfego de materiais entre os departamentos reduz consideravelmente a eficiência do sistema, dificultando o planejamento e agendamento dos novos processos.

Em contraste com a manufatura funcional, na manufatura linear a disposição dos equipamentos obedece a seqüência de processamento do produto, formando agrupamentos de equipamentos e atividades distintas entre si. Geralmente há um uso de máquinas especializadas em cada processo da seqüência, tornando-os mais caros e, portanto, tal

investimento somente é justificado quando há uma grande demanda por determinado produto. Além disso, a maior limitação deste conceito diz respeito à falta de flexibilidade da produção pela alta especialização das máquinas, o que dificulta a utilização desse sistema na elaboração de novos produtos que não foram previamente designados.

## 1.1 O Sistema de Manufatura Celular

A fim de evitar as desvantagens da manufatura linear e funcional, um sistema de manufatura alternativo foi proposto originalmente em Flanders (1924), mas somente a partir de Mitrofanov (1966) e Burbidge (1971) foi formalizado e amplamente difundido: o sistema de manufatura celular. A manufatura celular é uma aplicação do conceito de Tecnologia de Grupo, uma teoria baseada no princípio de que coisas similares devem ser produzidas de modo similar. Partindo deste princípio, as diversas peças que compõem um produto são agrupadas em famílias, de acordo com a similaridade entre as operações necessárias para sua fabricação. Desta forma, cada família que utiliza um mesmo conjunto de recursos é agrupada em um mesmo subsistema, as células de manufatura.

O principal objetivo da manufatura celular é o de agregar as vantagens dos conceitos da manufatura linear e funcional, e ainda reduzir a complexidade no controle do sistema. De acordo com Kusiak & Cheng (1990), cerca de 20% a 88% de redução nos esforços de movimentação de materiais pode ser alcançado ao se implantar células de manufatura, além de apresentar diversas vantagens e apontar algumas dificuldades e considerações. Máquinas são dedicadas a uma família de partes que são dispostas nas proximidades uma das outras. Como mostrado na Figura 1.1, a movimentação de materiais entre as máquinas é representada por setas. Deseja-se que o fluxo destes materiais esteja contido apenas nas máquinas designadas a uma mesma célula, ou seja, que não haja movimentação entre células distintas, promovendo um fluxo eficiente que se traduz em uma alta taxa de produção. O uso de máquinas e equipamentos de propósito geral nas células de manufatura permite que elas possam ser realocadas de modo a comportar mudanças no projeto e na demanda das partes com pouco esforço em termos de custo e tempo. Logo, provê uma grande flexibilidade na produção de uma variedade de partes. Em suma, a manufatura celular representa um sistema de produção híbrido, um misto de manufatura funcional, produzindo uma grande variedade de partes, com a manufatura linear, dedicada na produção em massa de um produto.

### **Vantagens:**

- redução do tempo necessário para preparar a máquina para a realização das operações;
- redução no inventário de materiais em produção;

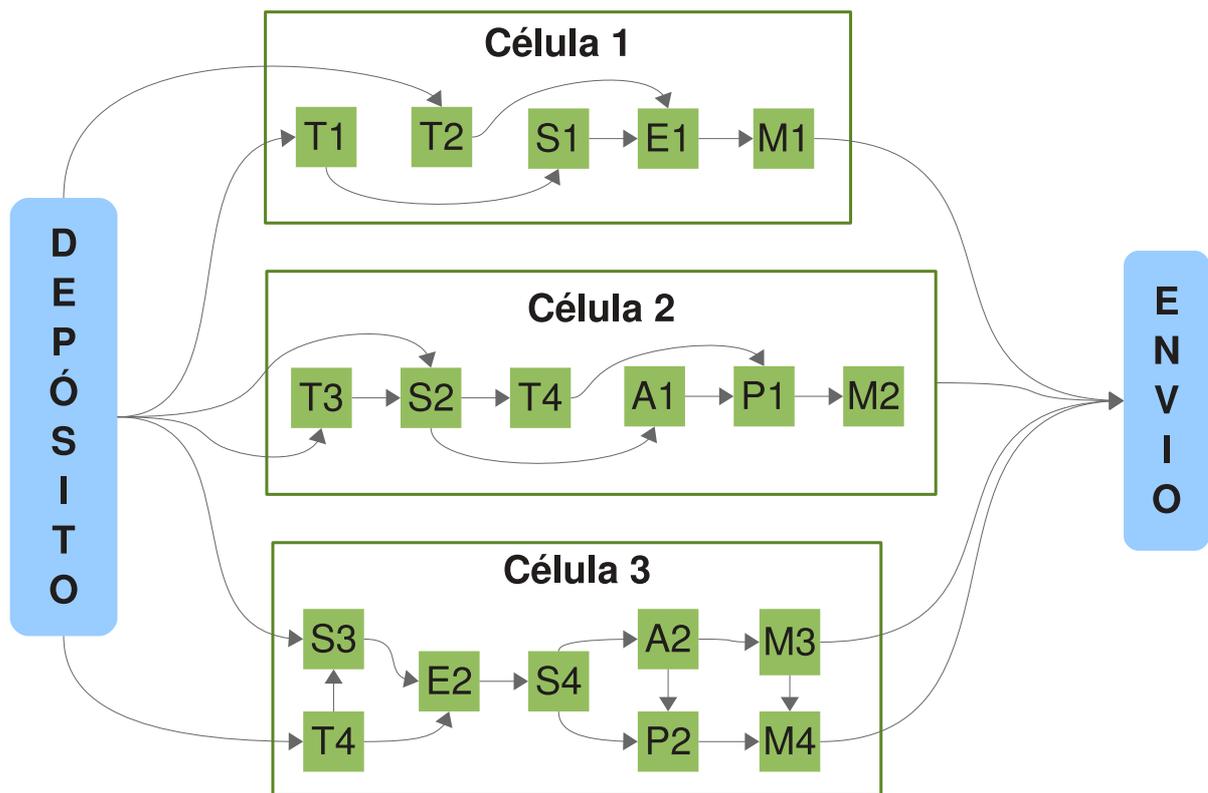


Figura 1.1: Sistema de Manufatura Celular. As letras (T, S, A, P, M e E) representam o tipo da máquina utilizado no processo e as setas o fluxo de material entre as máquinas.

- redução do tempo necessário no processamento de determinada parte;
- redução nos custos de transporte e manipulação (carga, descarga, paletização, despaletização) das matérias-primas;
- agenda de atividades simplificada;
- simplificação do fluxo de produção

#### Desvantagens:

- custo alto de implantação em sistemas já existentes;
- dificuldade em contornar casos de quebra, sobrecarga e balanceamento;
- dificuldade em gerenciar as operações intercelulares;
- requer sincronização das partes na subsequente montagem.

## 1.2 Contextualização do Problema

Uma das questões mais importantes na manufatura celular é o projeto das células de manufatura. Ao reagrupar a produção de diferentes partes em *clusters*, a gestão da manufatura é reduzida a gerenciar diferentes entidades menores. Sendo assim, o principal

desafio na manufatura celular é o projeto dessas entidades chamadas células. Estas células representam um conjunto de máquinas que podem ser dedicadas à produção de uma ou várias partes de um produto. A configuração ideal é alcançada ao tornar as células de manufatura totalmente independentes entre si, ou seja, cada produto é suficientemente fabricado em apenas uma célula.

No entanto, nem sempre é possível alcançar essa configuração ideal; uma vez criada as células, pode existir algum tráfego de material entre elas. Este tráfego corresponde à transferência de uma peça entre duas máquinas alocadas em células diferentes (tráfego intercelular). O objetivo final é o de minimizar o tráfego intercelular. Existem diferentes métodos para produzir essas células e dedicá-las a partes. Em geral, dois problemas interdependentes devem ser resolvidos.

- a atribuição de cada operação numa máquina: cada parte é definida por uma ou várias sequências de operações e cada uma delas pode ser alcançada por um conjunto de máquinas. Uma sequência final de máquinas deve ser escolhida para produzir cada parte.
- o agrupamento de cada máquina em células que produzem tráfego dentro e fora das células: em função da solução para o primeiro problema, diferentes *clusters* serão criados para minimizar o tráfego intercelular.

O problema clássico de agrupar um conjunto de máquinas em células, dedicadas à produção de uma família de partes, de modo a minimizar o tráfego intercelular e ao mesmo tempo maximizar o uso das máquinas das células (quantidade de partes processados pela máquina) é conhecido como Problema de Formação de Células de Manufatura (PFCM).

Outros fatores estratégicos podem ser considerados no planejamento de um sistema de manufatura celular. Na literatura, se alinha o termo Problema de Formação de Células de Manufatura Generalizado (PFCMG) para designar quaisquer variantes do PFCM em que se inclui um ou mais desses fatores práticos. Dependendo de quais forem considerados, o problema a ser tratado pode aumentar consideravelmente em complexidade.

Estes fatores podem ser subdivididos em três conjuntos. O primeiro diz respeito às questões estratégicas envolvidas na resolução do problema. A solução desejada pode ser orientada ao *layout* de cada célula; aos diversos tipos de máquinas disponíveis ou à flexibilidade quanto a substituição de máquinas, partes e equipamentos.

Em seguida, temos os fatores relacionados aos objetivos específicos das questões acima apresentadas: minimizar movimentos intercelulares, distâncias entre máquinas ou células, o número de partes excepcionais que gerem movimentos intercelulares, o número de máquinas conflitantes usadas por mais de uma célula ou custos em geral. Geralmente, todos esses objetivos estão em conflito uns com os outros. Por exemplo, a redução de custos está em conflito com a criação de células independentes, já que para diminuir a quantidade

de movimentos intercelulares, uma solução consiste em adicionar as máquinas necessárias em cada célula e, portanto, o custo para a aquisição de equipamentos aumenta neste caso.

Por fim, temos os fatores dados por restrições operacionais: utilização da máquina, o volume de produção, o número de células de fabricação, o tamanho das células, planos de processamento alternativos, sequência de operações, entre outros.

## 1.3 Objetivos da Pesquisa

Esta pesquisa tem por objetivo resolver de forma integrada o Problema de Formação de Células de Manufatura com a inclusão de dois fatores práticos de produção pertinentes no contexto das restrições operacionais do problema: a sequência de operações e os planos de processos alternativos. Há uma certa escassez de abordagens que tratam dessas informações na prática, deste modo, temos como objetivo contribuir com a literatura do problema com uma abordagem que resolva o problema de forma eficiente e flexível com a inclusão de outros fatores de produção. Mais informações sobre a sequência em que as operações das máquinas nas partes são realizadas são providas na Seção 2.2 e a presença de processos alternativos, isto é, quando se tem definido diversas formas para processar uma parte (plano de processamento) na Seção 2.3. A fim de alcançá-lo, os seguintes problemas deverão ser solucionados:

1. escolher um plano de processamento para cada parte;
2. alocar as operações que definem esse plano de processamento a uma máquina;
3. agrupar as máquinas em células de manufatura;
4. alocar as partes nas células.

Os três primeiros problemas precisam ser resolvidos simultaneamente, enquanto o último será uma consequência dos três primeiros. Para alcançarmos esse objetivo, primeiramente apresentamos um algoritmo baseado na meta-heurística ILS para o PFCM, resolvendo o item 3 e 4 (Seção 4.1). Em seguida, propomos uma adaptação desse algoritmo de modo a contemplar os itens 1 e 2, que será descrita em detalhes na Seção 4.2.

## 1.4 Estrutura do Trabalho

Os capítulos que se seguem estão organizados da seguinte maneira:

**Capítulo 2:** define o PFCM e explica os conceitos referentes à inclusão de roteamentos e processos alternativos ao problema;

**Capítulo 3:** apresenta as abordagens utilizadas na literatura para a resolução do problema e os critérios utilizados para a extensão do PFCM;

**Capítulo 4:** propõe um algoritmo para a resolução do PFCM e a extensão deste algoritmo para a inclusão dos novos fatores de produção;

**Capítulo 5:** apresenta os resultados computacionais;

**Capítulo 6:** apresenta as considerações finais.

# Capítulo 2

## Definições do Problema

Um sistema de manufatura pode ser representado de forma simplificada como uma matriz binária parte-máquina  $A_{|P| \times |M|}$ , em que  $P$  é o conjunto de partes de produtos e  $M$  o conjunto de máquinas que participam da fabricação das partes. Um elemento  $a_{ij}$  que vale 1 indica a realização de uma operação na parte  $i \in P$  pela máquina  $j \in M$ ;  $a_{ij} = 0$  indica o caso contrário.

O Problema de Formação de Células de Manufatura (PFCM) consiste em agrupar as máquinas em células e as partes em famílias, que serão alocadas nestas células, de forma a minimizar o número de movimentações das partes entre as células para a realização de uma operação e ao mesmo tempo maximizar o uso das máquinas das células. Uma família de partes é composta por aquelas partes que possuam um alto nível de similaridade entre si, isto é, há um grande número de máquinas em comum que realizam operações nelas. Para caracterizar uma solução, definem-se algumas estruturas presentes na matriz parte-máquina:

**Bloco:** é um subconjunto de partes e máquinas da matriz. Representa uma célula de manufatura.

**Operação:** elemento com valor 1. 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 uma subutilização da máquina na célula por não operar a parte designada.

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

Uma solução pode ser formada através de um rearranjo (permutação) de linhas e colunas da matriz, de modo a formar blocos de operações dispostos diagonalmente. Na Figura 2.1a, temos um exemplo da representação matricial de um sistema de manufatura, com

$|P| = 7$  e  $|M| = 5$ , dado de entrada do problema. Já na Figura 2.1b, temos um exemplo de solução com as células formadas pelo subconjunto de máquinas  $\{M_2, M_3, M_5\}$  e  $\{M_1, M_4\}$  associadas respectivamente com as famílias de partes  $\{P_1, P_3, P_7\}$  e  $\{P_2, P_4, P_5, P_6\}$ .

<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th></th> <th><math>M_1</math></th> <th><math>M_2</math></th> <th><math>M_3</math></th> <th><math>M_4</math></th> <th><math>M_5</math></th> </tr> </thead> <tbody> <tr> <th><math>P_1</math></th> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <th><math>P_2</math></th> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <th><math>P_3</math></th> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th><math>P_4</math></th> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <th><math>P_5</math></th> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th><math>P_6</math></th> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <th><math>P_7</math></th> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table> <p>a) representação matricial</p>		$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$P_1$	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	$\implies$	<table border="1" style="border-collapse: collapse; margin: auto;"> <thead> <tr> <th></th> <th><math>M_2</math></th> <th><math>M_3</math></th> <th><math>M_5</math></th> <th><math>M_1</math></th> <th><math>M_4</math></th> </tr> </thead> <tbody> <tr> <th><math>P_1</math></th> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th><math>P_3</math></th> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th><math>P_7</math></th> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th><math>P_2</math></th> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <th><math>P_4</math></th> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <th><math>P_5</math></th> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <th><math>P_6</math></th> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p>b) exemplo de solução</p>		$M_2$	$M_3$	$M_5$	$M_1$	$M_4$	$P_1$	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
	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$																																																																																													
$P_1$	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																																																																																													
	$M_2$	$M_3$	$M_5$	$M_1$	$M_4$																																																																																													
$P_1$	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 2.1: Representação matricial da entrada e uma solução.

## 2.1 Tamanho Mínimo das Células

O tamanho da célula é normalmente expresso pela quantidade de máquinas que a compõe. 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. Ou seja, além de conhecer quais máquinas processam quais partes, informações sobre a capacidade das células podem ser 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 do problema. O termo *singleton* é definido como uma célula que possua menos que duas máquinas ou partes (Gonçalves & Resende, 2004; Wu et al., 2009). Existem abordagens que consideram ou não a presença de *singletons*. Dentre os que permitem, há trabalhos como o de Pailla et al. (2010), que permite a existência de células vazias na solução, ou seja, máquinas ou partes que não pertençam a nenhuma célula, e os que permitem no mínimo células unitárias (James et al., 2007).

Restringir o tamanho mínimo da célula torna o problema mais realista, porém pode diminuir a qualidade da solução, ao forçar que uma máquina esteja em uma determinada célula para que esse requisito seja cumprido. Por este motivo, trabalhos em que *singletons* não são permitidos normalmente possuem soluções de menor qualidade do que aqueles que a permitem e, portanto, a comparação direta entre as soluções produzidas com e sem essa restrição não é justa. Como em Wu et al. (2009), 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 2.1b consiste de uma solução para a matriz parte-máquina da Figura 2.1, em que células *singleton* não são permitidas. Observe a presença dos blocos diagonais representando a formação das células do sistema de manufatura. 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 partes 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 à outra célula; no entanto, a máquina  $M_4$  está subutilizada na célula por não operar a parte  $P_5$  associada.

	$M_2$	$M_3$	$M_1$	$M_4$	$M_5$
$P_1$	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*.

Já a Figura 2.2 consiste de uma solução para a matriz parte-máquina da Figura 2.1 em que *singleton* são permitidos. 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.

Observe que, nesta formação, a célula 3 possui apenas 1 máquina (a máquina  $M_5$ ); portanto, dizemos que esta é uma célula *singleton*.

## 2.2 Sequência de Operações e Rotas de Processamento

A sequência de operações é definida como uma ordenação dos processos (ou operações) na qual a parte é submetida durante sua fabricação (Vakharia & Wemmerlov, 1990). Considerar a sequência das operações complica o PFCM e muitos algoritmos não tratam desta questão. No entanto, isto permite um cálculo mais realista sobre a movimentação de partes entre as células e o custo das operações.

Métodos para a formação de células que utilizem estes dados calculam a quantidade de movimentos baseados no número de células que aquela parte visita no processo de fabricação, ao invés de apenas contabilizarem o número de exceções. Contudo, o número de movimentos pode ser maior que o número de células visitadas, uma vez que, dependendo da sequência de operações, uma parte pode ter que ir e voltar por entre as células para realizar as operações necessárias. Sem utilizar a informação sobre a sequência de operações, a contagem desses movimentos não será precisa e realista.

Quando atribuímos um conjunto de máquinas para realização das operações em determinada parte, temos um roteamento ou rota de processamento. A vasta maioria dos métodos da literatura que tratam as sequências de operações assume que cada parte possui apenas um roteamento. As vantagens e possibilidades em se ter roteamentos alternativos são discutidos em (Nagi et al., 1990). Seu uso é típico em ambientes de produção discretos com tamanhos de lotes reduzidos em que a produção de uma parte é realizada de forma paralela (Defersha & Chen, 2006) (em mais de uma máquina ao mesmo tempo). Esta flexibilidade no roteamento aumenta o número de formas em que se podem projetar as células de manufatura.

	$P_1$	$P_3$	$P_2$	$P_4$	$P_5$
$M_2$	1	1	0	1	0
$M_3$	3	2	0	0	0
$M_1$	2	0	1	3	2
$M_4$	0	0	2	2	1

Figura 2.3: PFCM com Sequência de Operações.

Na Figura 2.3, temos um exemplo de solução do PFCM com informações sobre a sequência das operações, em que para cada parte temos uma sequência de máquinas que

devem ser visitadas para que as operações sejam executadas. Observe que temos duas células de máquinas:  $C_1 = \{M_2, M_3\}$  e  $C_2 = \{M_1, M_4\}$ . Se para processar a parte  $P_1$  devemos visitar as máquinas na ordem  $M_2 \rightarrow M_1 \rightarrow M_3$ , temos que a ordem de visitação das células será  $C_1 \rightarrow C_2 \rightarrow C_1$  e, portanto, dois deslocamentos de materiais entre células. Se por outro lado, a ordem de visitação das máquinas tivesse sido  $M_2 \rightarrow M_3 \rightarrow M_1$ , como as máquinas  $M_2$  e  $M_3$  fazem parte da mesma célula  $C_1$ , temos apenas um deslocamento de material entre células e, portanto, mais eficiente.

## 2.3 Planos de Processamento

Máquinas idênticas ou que possuem uma mesma finalidade podem ser agrupadas em tipos. Neste contexto, um plano de processamento é caracterizado por um conjunto ou sequência de tipos de máquinas, responsáveis pela execução das operações numa parte. Podemos então definir que o plano de processamento é uma generalização do conceito de rota de processamento, uma vez que nesta consideramos a sequência de operações e não agrupamos as máquinas em tipos, já no plano de processamento esses dois fatores são opcionais.

De fato, quando temos um plano de processamento caracterizado por uma sequência de tipos de máquinas, obteremos uma rota de processamento ao se definir quais as máquinas específicas de cada tipo de máquina realizará a operação. Levando em consideração que dentre os tipos de máquinas estabelecidos haverá várias possibilidades de escolha de máquinas para a execução dos processos, teremos assim vários roteamentos possíveis para este plano de processamento. Ter várias possibilidades de roteamentos para o processamento de uma parte oferece diversos benefícios, como a redução na interdependência das células (redução de exceções) e aumento no uso das máquinas na célula (redução de lacunas) (Kusiak, 1987; Kang & Wemmerlov, 1993).

O problema de formação de células que incorpora o uso de planos de processamentos alternativos, mas sem considerar a sequência das operações, como no PFCM clássico, e o agrupamento de máquinas em tipos, é conhecido como Problema de Tecnologia de Agrupamento Generalizado (PTAG). Kusiak (1987) foi um dos primeiros a considerar seu uso. Se desconsiderarmos ainda esses planos de processamento alternativos, reduziremos o problema de volta ao PFCM clássico.

Na Figura 2.4, temos um exemplo de entrada para o PFCM com múltiplos planos de processamento  $R$  para cada parte  $P$ . A solução do problema consistirá em estabelecer o plano de processamento que deve ser utilizado em cada parte, ao mesmo tempo em que agrupa as máquinas em células, de forma que, respeitando a sequência de operações, minimize o custo pela movimentação das partes (entre as células) para a realização dessas operações. Tomando essa figura como exemplo, vamos supor que nessa entrada, para cada

PN	P1			P2			P3		P4	
RN	R1	R2	R3	R1	R2	R3	R1	R2	R1	R2
M1		1	1		1	1				
M2		2								
M3										
M4	1			1			1	1	1	1
M5							2	2	2	2
M6	2			2				3	3	3
M7	3		2	3	2	2	3			4
M8		3		4		3	4	4	4	5
M9	4		3		3					

Figura 2.4: Representação dos Planos de Processamento: M = Máquinas; P = Partes; R = Plano de Processamento

parte P, escolhemos o plano de processamento R sombreado. Teremos então o apresentado na Figura 2.5, no qual a solução final se dará em seguida, ao se estabelecer o agrupamento de máquinas e partes ideal considerando as sequências de operações, como o realizado na Figura 2.3.

	$P_1$	$P_2$	$P_3$	$P_4$
$M_1$		1		
$M_4$	1		1	1
$M_5$			2	2
$M_6$	2			3
$M_7$	3	2	3	
$M_8$			4	4
$M_9$	4	3		

Figura 2.5: Entrada após escolha dos Planos de Processamento para cada parte.

Se em outro momento, a escolha dos planos de processamento for diferente, teremos uma outra entrada para o posterior agrupamento das máquinas e partes, sendo possível a formação de uma solução final com menos movimentação de materiais entre as células.

## 2.4 Níveis de Alternatividade

Várias abordagens na literatura permitem levar em consideração os planos de processamentos alternativos, mas não há um consenso entre os autores sobre os pontos em que essa flexibilidade se apresenta. A maneira em que podemos levar em consideração essa questão pode ser resumida em seis casos:

1. plano único fixo (caso base, sem alternatividade);
2. plano único variável por máquinas replicadas;

3. plano único variável por máquinas alternativas;
4. plano único variável por tipos de máquinas;
5. planos múltiplos fixos;
6. planos múltiplos variáveis por tipos de máquinas (tratado neste trabalho).

Cada caso é ilustrado por um exemplo de processamento de uma parte  $p_i$  que requer cinco operações  $\{o_1 - o_2 - o_3 - o_4 - o_5\}$ . O processamento de uma parte é definido por um ou mais planos  $R$ , representados por uma sequência de máquinas (rota)  $m_j$ , ou sequência de tipos de máquinas  $t_j$ , que executam as operações.

**Caso 1:** o processamento de cada parte é representado por uma sequência fixa de máquinas (Chandrasekharan & Rajagopalan, 1986a; Stawowy, 2006).

Exemplo:  $R_1(p_i) = \{m_1 - m_2 - m_3 - m_4 - m_5\}$ . Há apenas uma maneira de fabricar a parte  $p_i$ .

**Caso 2:** o processamento de cada parte é representado por uma sequência de máquinas com possibilidade de replicação. Em certos estudos, os autores não definem explicitamente tipos de máquinas, mas propõem simplesmente a duplicação de máquinas existentes. Este caso é semelhante ao caso 4, mas aqui não é obrigatório que cada tipo de máquina possua mais que um exemplar (Nagi et al., 1990; Ramabhatta & Nagi, 1998).

Exemplo:  $R_1(p_i) = \{m_1 - (m_2 \text{ ou } m_3) - m_4 - (m_2 \text{ ou } m_3) - m_5\}$ . As operações  $o_2$  e  $o_4$  podem ser realizadas pelas máquinas  $m_2$  ou  $m_3$ . Neste caso, podemos agrupar as máquinas em quatro tipos:  $t_1 = \{m_1\}$ ,  $t_2 = \{m_2; m_3\}$ ,  $t_3 = \{m_4\}$  e  $t_4 = \{m_5\}$ . Assim, temos quatro possibilidades de processamento desta parte ( $1 \times 2 \times 1 \times 2 \times 1 = 4$ ).

**Caso 3:** o processamento de cada parte é representado por uma sequência de máquinas com alternativas. Neste caso, não podemos agrupar todas as máquinas em tipos, uma vez que podemos não ter uma correspondência entre as máquinas alocadas com as operações. Aqui todas as combinações são possíveis, seguindo a escolha de máquinas para cada operação. Este caso é semelhante ao caso 5, mas aqui cada máquina é responsável por uma operação (Vivekanand & Narendran, 1998; Mosier & Taube, 1985).

Exemplo:  $R_1(p_i) = \{(m_1 \text{ ou } m_3) - m_2 - (m_2 \text{ ou } m_3) - m_4 - (m_5 \text{ ou } m_3)\}$ . As máquinas  $m_1$  e  $m_3$  não podem ser definidas como pertencente ao mesmo tipo já que não aparecem conjuntamente como alternativas em outras operações. Neste exemplo há oito combinações possíveis para fabricar esta parte ( $2 \times 1 \times 2 \times 1 \times 2 = 8$ ).

**Caso 4:** o processamento de cada parte é representado por uma sequência de tipos de máquinas. Aqui a operação é definida pelos requisitos das ferramentas. Se uma ferramenta é disponível em mais de uma máquina, essas máquinas são consideradas como pertencentes ao mesmo tipo, ou seja, cada tipo de máquina é capaz de realizar uma operação específica. Neste caso, o problema consiste em encontrar para cada operação a melhor máquina dentre as pertencentes ao tipo requisitado (Askin et al., 1997; Baykasoglu & Gindy, 2000; Diallo et al., 2001).

Exemplo: cada parte  $p_i$  possui um plano  $R_1(p_i) = \{t_1 - t_2 - t_3 - t_4 - t_5\}$ . Aqui o problema consiste em escolher a melhor máquina para cada operação. Se cada tipo de máquina  $t_i$  contém  $n_i$  máquinas, o número de combinações será igual a  $n_1 \times n_2 \times n_3 \times n_4 \times n_5$ .

**Caso 5:** o processamento de cada parte é representado por múltiplas sequências de máquinas (ver Figura 2.4 como exemplo). Neste caso, o número de máquinas usadas na sequência pode variar, ou seja, uma máquina pode realizar mais de uma operação. O problema se resume em buscar boas sequências dentre todas as possíveis (Gupta, 1993; Logendran et al., 1994; Sofianopoulou, 1999).

Exemplo:  $p_i$  possui diferentes planos possíveis:  $R_1(p_i) = \{m_3 - m_2 - m_3 - m_4 - m_5\}$ ,  $R_2(p_i) = \{m_3 - m_2 - m_6 - m_4 - m_5\}$  e  $R_3(p_i) = \{m_6 - m_5 - m_3 - m_4\}$ . A parte  $p_i$  pode ter duas sequências diferentes com o mesmo número de operações como  $R_1(p_i)$  e  $R_2(p_i)$  com cinco operações, mas não necessariamente, já que temos  $R_3(p_i)$  com quatro. Neste exemplo, o problema pode ser resumido como a escolha de uma dentre as três sequências possíveis, de modo a otimizar o problema como um todo, ou seja, em relação às demais partes.

**Caso 6:** pode ser considerado como a combinação dos casos 4 e 5, em que o processamento de cada parte é representada por várias possíveis sequências de tipos de máquinas, fazendo com que o número de possíveis combinações para cada parte seja ainda maior (Sankaran & Kasilingam, 1990; Uddin & Shanker, 2002).

Exemplo: cada parte  $p_i$  possui três planos de processamento:  $R_1(p_i) = \{t_1 - t_2 - t_3 - t_4 - t_5\}$ ,  $R_2(p_i) = \{t_1 - t_5 - t_6 - t_4 - t_2\}$  e  $R_3(p_i) = \{t_1 - t_3 - t_6 - t_5\}$ . Neste caso, o problema se resume em escolher a melhor máquina para cada operação, considerando todos os tipos de máquinas possíveis dentre todas as alternativas de processamento. Logo, o número de possíveis combinações será de  $(n_1 \times n_2 \times n_3 \times n_4 \times n_5) + (n_1 \times n_5 \times n_6 \times n_4 \times n_2) + (n_1 \times n_3 \times n_6 \times n_5)$ .

No caso 1 não temos nível algum de alternatividade: há somente uma sequência de máquinas disponível. Nos casos 2 a 5, temos níveis crescentes de alternatividade, ou seja, o problema vai se tornando cada vez mais complexo e com mais escolhas a serem

feitas. Apenas no caso 6 que podemos considerar processamentos alternativos em termos de sequências de tipos de máquinas. No entanto, seu uso é limitado, como observado por (Suresh & Slomp, 2001), em que a maioria das indústrias não possuem mais que um plano de processamento ou existe apenas um conjunto de ferramentas para cada operação. Considerando que o caso 6 é o mais geral e pode ser reduzido aos casos anteriores, o algoritmo proposto nesse trabalho o aborda para que dependendo de qual caso o método da literatura considere seremos capazes de resolvê-lo.

## 2.5 Qualidade da Solução

Na literatura, existem diversas medidas que avaliam a qualidade de uma solução do PFCM. Uma revisão destas medidas, sobre diferentes parâmetros de avaliação, é apresentada em Sarker & Khan (2001). Na Tabela 2.1, definimos algumas notações utilizadas pelas 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	$e_1$	$e_0$

Tabela 2.1: Principais parâmetros utilizados pelas medidas de desempenho.

Uma das métricas mais utilizadas na literatura do PFCM é a Eficácia de Agrupamento proposta por Kumar & Chandrasekharan (1990). Foi a partir de Srinivasan & Narendran (1991) que diversas abordagens começaram a ser desenvolvidas considerando-a como padrão na avaliação de qualidade das soluções. A Eficácia de Agrupamento tem por objetivo evitar as dificuldades em se estabelecer um parâmetro presente em outra métrica conhecida como Eficiência de Agrupamento, utilizada para dar mais importância à minimização do número de lacunas em relação ao de exceções, ou vice-versa. Na Eficácia de Agrupamento, essa compensação é feita de forma implícita, dando uma maior ênfase à minimização das exceções, mas atuando de forma escalável em relação ao tamanho da solução. Este índice é definido como:

$$\mu = \frac{e_1 - e_1^{out}}{e_1 + e_0^{in}} \quad (2.1)$$

Nesta equação, o numerador representa o número de operações dentro dos blocos, em que quanto mais próximo for do total de operações melhor. Enquanto isso o denominador aumenta conforme o número de exceções, diminuindo o valor final da medida. Em outras palavras, quanto menor o número de lacunas maior será o valor do índice. Como temos uma formação ideal quando não há lacunas nem exceções, observe então que quanto mais próximo de 1 for este índice, melhor será o agrupamento.

Na solução apresentada pela Figura 2.1b temos 3 lacunas e 2 exceções. Medindo a qualidade da solução através deste índice temos que  $\mu = \frac{16-2}{16+3} = 0,7368$ . Já na solução apresentada pela Figura 2.2 temos 4 exceções, e com isso  $\mu = \frac{16-4}{16+0} = 0,75$ . Dito isto, partindo destes dois valores obtidos, reforça-se a ideia de que, como dito anteriormente, apesar de fazer mais sentido uma formação sem células *singleton* na prática, geralmente esta possui uma pior qualidade.

Quando tratamos de sequência de operações, a eficácia de agrupamento perde o sentido, pois a definição de exceção não será mais a mesma (número de operações fora dos blocos). Aqui, uma exceção será toda operação em que a operação anterior foi executada em uma célula diferente. Na Figura 2.3, a operação da máquina  $M_3$  na parte  $P_1$  será uma exceção, já que a operação anterior desta parte foi executada na máquina  $M_2$  pertence à outra célula. Neste caso, a qualidade da solução será dada em termos de custos de movimentação das partes, calculado pelo somatório do número de operações executadas em células diferentes em relação à operação anterior.

# Capítulo 3

## Revisão da Literatura

Nas últimas três décadas, um grande número de artigos científicos e relatórios técnicos têm sido publicados sobre a formação de células de manufatura. Revisões da literatura podem ser encontradas em Selim et al. (1998) e Papaioannou & Wilson (2010). Mais recentemente, muitos pesquisadores tem reconhecido a necessidade em se considerar os diversos fatores práticos relacionados ao problema. Defersha & Chen (2006) faz um levantamento desses fatores de produção, utilizado como base em diversos trabalhos da literatura recente.

Tabela 3.1: Principais fatores práticos de produção.

Número	Fatores de Produção
1	Processos alternativos
2	Demanda variável
3	Balanceamento da carga de trabalho
4	Reconfiguração dinâmica das células
5	Separação do lote das partes
6	Tipos de ferramentas requisitadas pelas partes
7	Tipos de ferramentas disponíveis nas máquinas
8	Restrições de proximidade das máquinas
9	Sequência nas operações
10	Custos/Tempo de preparação
11	Restrições no tamanho da célula
12	Disposição física do sistema de manufatura
13	Alocação de operadores
14	Capacidade das máquinas
15	Custos de aquisição das máquinas
16	Custos de consumo das ferramentas
17	Custos de operação

Uma vez que é muito difícil incorporar todos esses fatores em um único modelo, a principal abordagem utilizada é a consideração de apenas alguns poucos fatores-chave simultaneamente. Neste trabalho, tratamos o PFCM levando em consideração os processos

alternativos (1) e sequência de operações (9). De acordo com a classificação proposta na Seção 2.4, tratamos o caso (6) que representa o maior nível de flexibilidade do sistema de manufatura, e por consequência todos os outros casos mais simples podem ser facilmente adaptados. De certa forma, também tratamos parcialmente Restrições no tamanho da célula (11) ao considerar quando a célula é ou não *singleton*.

### 3.1 Critérios de Otimalidade

Mansouri et al. (2000) apresenta uma revisão completa sobre os critérios de otimalidade empregados no tratamento dos fatores de produção na formação de células de manufatura. Este analisa um grande painel de critérios utilizados para a tomada de decisão, comparação entre as entradas do problema, abordagens de solução e saídas entre os modelos selecionados. Abaixo, temos uma classificação dos principais critérios abordados na literatura e seus respectivos trabalhos.

**Custos:** todos os custos devem ser minimizados: custos fixos de máquinas, custos de manuseio de materiais, custos de inventário de processos de fabricação, custos de inventário de ciclo de produção, custos de produção variáveis e custos de instalação, custo associado à duplicação de máquinas (Askin & Subramaniam, 1987; Defersha & Chen, 2006) ou investimento em novos equipamentos (Adil et al., 1993; Shafer & Rogers, 1991), custo associado à subcontratação de partes de produtos (Kumar & Vannelli, 1987), custos de movimentação de materiais (movimentos intercelular e intracelular) (Defersha & Chen, 2006), custo de operação ou processamento (Adil et al., 1993), custo de gargalo na produção (Kumar & Vannelli, 1987; Wei & Gaither, 1990), custo de uso espacial, custo de amortização anual (Logendran et al., 1994), custo de penalidade para operações que precisam ser realizadas em células que não sejam as designadas, etc.

**Similaridade:** diferentes autores trabalham em maximizar a similaridade de máquinas (Grefenstette, 1985), similaridade de partes com base nas características das máquinas, nas ferramentas necessárias ou na eficiência do sistema (medida baseada no grau de similaridade de partes) (Sarker & Xu, 1998; Vakharia & Wemmerlov, 1990). Por outro lado, outros autores minimizam a dissimilaridade das peças com base nos atributos de projeto e fabricação ou na dissimilaridade com base nas sequências de operações (Yasuda & Yin, 2001).

**Movimentação:** na movimentação de material entre máquinas e células, o objetivo difere caso a sequência de operações esteja sendo empregada no problema. Neste caso, a maximização da independência das células será descrito como a minimização dos movimentos intercelulares e/ou maximização dos movimentos intracelulares.

res (Logendran, 1990; Dahel & Smith, 1993). Caso a sequência de operações não seja levada em consideração (com matriz de incidência, por exemplo), a maximização da independência de células será obtida minimizando o número de elementos excepcionais e/ou minimizando o número de elementos vazios (Adil et al., 1996; Kumar & Chandrasekharan, 1990).

**Balanceamento de Carga:** o balanceamento da carga de trabalho contribui para o bom funcionamento do sistema e melhores desempenhos em termos de rendimento, tempo total, fluxo de tempo e atraso (Kim, 1993). Balancear a carga reduz o inventário de trabalho em processo, melhora o fluxo de material através do sistema e impede o uso intenso de algumas células ou o uso mais baixo de outras (Baykasoglu et al., 2001). Nas extensões do PFCM, vários autores tentam minimizar os desequilíbrios de carga intracelular e/ou intercelular (Logendran, 1990). Este critério permite maximizar o uso médio das células, minimizando a soma da variabilidade da carga de trabalho em cada célula e/ou entre diferentes células (Wei & Gaither, 1990).

**Uso das Máquinas:** quando os autores trabalham com os roteamentos alternativos, onde a escolha das máquinas deve ser feita para cada operação, vários critérios podem ser minimizados: os tempos de preparação ou a variação entre os tempos de preparação, a variação entre os custos operacionais dentro das células, a variação entre os usos da máquina. O objetivo também pode ser manter um nível aceitável de utilização da máquina, minimizando o não cumprimento do limite inferior de uso ou o excedente do limite superior (Sofianopoulou, 1999; Defersha & Chen, 2006).

**Flexibilidade:** nas extensões do PFCM, os autores lidam com a maximização de quatro tipos de flexibilidade:

- a flexibilidade do tipo de máquina, que representa a capacidade das máquinas agrupadas em células de processar um grande número de tipos de operações distintas (Selim et al., 1995);
- a flexibilidade de roteamento, que representa a capacidade do sistema em processar partes completamente em múltiplas células (Selim et al., 1995; Sankaran & Kasilingam, 1990);
- a flexibilidade no volume de partes, que representa a capacidade do sistema em lidar com alterações no volume do conjunto de partes (calculada como o aumento percentual máximo no volume para todas as partes que podem ser trocadas sem alterar a configuração do sistema) (Selim et al., 1995);
- a flexibilidade no conjunto de partes, que representa a capacidade do sistema em lidar com diferentes conjuntos de produtos com o mínimo de interrupção (Sarin & Dar-El, 1986).

Existe uma grande diversidade de objetivos e critérios, e seus usos dependem do método de resolução do problema. Os coeficientes de similaridade permitem o agrupamento de elementos semelhantes. Todos os custos podem ser combinados com uma soma ponderada em métodos de programação matemática. No entanto, estes dados nem sempre são conhecidos. A avaliação de movimento e balanceamento de carga de trabalho é frequentemente usada ao mesmo tempo com um método auxiliar de decisão multicritério. Se o objetivo é a concepção de uma nova fábrica com o menor tráfego possível entre as células, o uso de movimento e carga de trabalho é realmente pertinente. Esses diferentes critérios podem ser melhorados pelo uso de planos de processo alternativos.

A flexibilidade de fabricação está se tornando significativamente importante na manufatura moderna. É um parâmetro altamente desejado, mas ter um máximo de flexibilidade significa um máximo de máquinas em cada célula ou dobrar as mesmas máquinas em cada célula, o que não é realista para os custos. No entanto, no ponto de vista do negócio, desejam-se resultados em termos de custos. Assim, é interessante incluir os custos ou uma parte dos custos na resolução. É importante também produzir novas fábricas fáceis de gerenciar e ser capaz de quantificar o custo total e o custo economizado de novas instalações, além de maximizar o uso de máquinas para maior eficiência considerando o custo de operação das máquinas.

## 3.2 Técnicas Utilizadas

Nesta seção organizamos as principais abordagens empregadas na literatura para a resolução do problema. Exemplificamos cada uma com os trabalhos mais relevantes e uma breve descrição.

### 3.2.1 Procedimentos Descritivos

É apresentado em Askin & Vakharia (1990) uma visão geral dos primeiros métodos desta categoria. 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 as alocando 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) desenvolvida inicialmente por Burbidge (1971), 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.

### 3.2.2 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.

#### 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 (1989), Singh (1993) e Shafer (1998). Os métodos mais conhecidos desta categoria são:

- *Bond Energy Analysis* (BEA) (McCormick et al., 1972)
- *Rank Order Clustering* (ROC) (King, 1980)
- *Improved Rank Order Clustering* (ROC2) (King & Nakornchai, 1982)
- *Modified Rank Order Clustering* (MODROC) (Chandrasekharan & Rajagopalan, 1986b)
- *Direct Clustering Analysis* (DCA) (Chan & Milner, 1982)

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 intercelulares).

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.

### 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 preliminar, 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 num único grupo e iterativamente particionando estas máquinas em grupos menores. O critério de parada é alcançado enquanto não houver melhora após certo número de iterações ou até que cada máquina/parte fique isolada, ou seja, cada máquina em um grupo distinto.

**Aglomerativo:** iniciam colocando cada máquina (parte) em um grupo distinto e iterativamente estes grupos são conjugados, até que um número desejado de células seja formado ou até que haja apenas um único grupo com todas as máquinas.

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. (1990) e a segunda etapa do algoritmo MODROC (Chandrasekharan & Rajagopalan, 1986b).

### 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, definem-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, apesar de que 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 (Chandrasekharan & Rajagopalan, 1987), baseado em um critério de avaliação da eficiência de agrupamento, e GRAPHICS (Srinivasan & Narendran, 1991), 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 (Yu & V.V., 1977).

**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 (Lloyd, 1982).

**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 (Jarvis & Patrick, 1973).

### 3.2.3 Teoria dos Grafos

Esta abordagem trata as máquinas e 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 (1975) 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. 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 intercelular.

Em Vannelli & Kumar (1986) um modelo de grafo é usado para obter uma estrutura de blocos diagonal perfeita utilizando duplicação de máquinas. Em Askin et al. (1991) 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.

Um dos métodos mais recentes e promissores utilizando essa abordagem é a apresentada em Pinheiro et al. (2016). Este investiga o Problema de Biclusterização por Edição de Aresta (PBEA) e traça um paralelo com o PFCM, desenvolvendo um algoritmo que encontra soluções ótimas para o PFCM baseado em formulações matemáticas para o PBEA. Além de apresentar uma ideia geral de como adaptar alguns dos fatores práticos de produção utilizando este modelo. Neste sentido, classifica cada um deles em dois grupos: os que permitem uma adaptação direta no modelo do PBEA (resolvendo o problema com pequenos ajustes no algoritmo), e as que admitem uma adaptação parcial (resolvendo inicialmente o subproblema relativo ao novo fator de produção, e em seguida utilizando o resultado como entrada do algoritmo baseado no PBEA).

### **1. Permitem a adaptação direta do modelo PBEA:**

#### **a - custo/tempo de operação:**

acrescente custos nas arestas.

#### **b - capacidade nas células ou famílias de partes/máquinas:**

acrescente restrições limitando o tamanho das bicliques.

#### **c - capacidade das máquinas e demanda por produto:**

acrescente custos nos vértices.

#### **d - restrições de proximidade das máquinas:**

dada a distância entre cada par vértices, estabeleça restrições que proíbam a presença de pares de vértices na mesma biclique.

### **2. Permitem uma adaptação parcial:**

#### **a - fatores além do contexto de máquinas e partes:**

temos nessa categoria a alocação de operadores, custos de ferramentas, tipos de ferramentas, etc. Primeiramente, resolva o problema associado com os operadores/ferramentas, isto é, decida quais operadores/ferramentas serão alocados em cada máquina; em seguida, resolva o PFCM utilizando o resultado da alocação como entrada do algoritmo PBEA.

#### **b - fatores que não possuem como o objetivo final a formação de bicliques perfeitas:**

é o caso do PTAG apresentado na Seção 2.3. Para tal, resolva o problema da escolha do plano de processamento de cada parte, e então utilize o algoritmo para o PBEA a fim de minimizar os movimentos das partes.

### 3. Não permitem uma adaptação:

São os fatores que necessitam de estruturas extras: disposição das máquinas (problema espacial), balanceamento de carga e divisão dos lotes de produção (fluxo de operações entre as máquinas). Nestes casos, é necessário que a resolução seja integrada (não pode ser separada em dois subproblemas), o que impossibilita o uso do PBEA.

#### 3.2.4 Meta-heurísticas

Em problemas de otimização  $\mathcal{NP}$ -difícil muitas vezes é impraticável aplicar métodos exatos para instâncias grandes do problema, de modo a obter soluções ótimas em tempo aceitável. Abordagens de busca heurística têm sido utilizadas como alternativas quando o tempo computacional é proibitivo e/ou funções objetivos lineares não podem ser formuladas. No entanto, duas desvantagens básicas das heurísticas são que as soluções encontradas frequentemente são de baixa qualidade e fortemente dependentes das instâncias e do problema.

As meta-heurísticas podem ser definidas como procedimentos de alto nível que orientam uma heurística subordinada para encontrar soluções de alta qualidade e de forma eficiente, providas de mecanismos de escape de ótimos locais que permitem que o espaço de busca seja mais bem explorado e diferenciam-se entre si pelo mecanismo empregado para tal. Abaixo, temos uma breve descrição de algumas das meta-heurísticas mais conhecidas:

***Simulated Annealing:*** 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. 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 necessário para alcançar a solução ótima aumenta com esses dois parâmetros.

**Algoritmos Genéticos:** fundamenta-se 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. Algoritmos genéticos utilizam uma espécie de busca aleatória que impede o algoritmo de ficar preso em mínimos locais indesejados e deixa com que o todo o domínio de busca seja explorado.

**Busca Tabu:** técnica de otimização baseada na noção de vizinhança, originada nos trabalhos de Glover (Glover, 1989, 1990). 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 por *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).

A função objetivo pode possuir diversos ótimos globais, isto é, vários pontos no qual os valores da função objetivo são iguais ao valor ótimo global. Além disso, devem possuir alguns ótimos locais nos quais os valores da função objetivo são próximos ao valor ótimo global. As regras que regem a população de indivíduos devem orientar a busca a um ótimo global. Essas meta-heurísticas podem ser facilmente paralelizadas, o que frequentemente torna possível a criação de algoritmos de otimização altamente eficientes para diversas áreas de aplicação.

Um inconveniente em comum a essas meta-heurísticas reside no fato de que não há garantias que a melhor solução seja até mesmo um ótimo local. Para tanto, muitas meta-heurísticas destinadas a problemas de otimização complexos se utilizam de hibridizações, objetivando o aprimoramento da qualidade da solução ou dos tempos de execução. As meta-heurísticas são especialmente eficientes quando são combinadas com heurísticas específicas ao problema, técnicas de busca local, métodos de aproximação ou métodos exatos.

Há uma extensa literatura devotada às heurísticas para a formação de células de manufatura, mas a maioria delas não garante a otimalidade ou trabalham exclusivamente com a matriz de incidência parte-máquina. Enquanto alguns desses algoritmos oferecem resultados superiores para determinadas aplicações, nenhuma técnica se mostrou capaz de prover a melhor solução para uma ampla gama de aplicações (Joines et al., 1997).

Nas sessões a seguir, descrevemos algum dos métodos da literatura do problema que são baseados em meta-heurísticas, separados em três categorias de acordo com o uso de processos e roteamentos alternativos. Vale salientar que a maioria dos trabalhos mencionados abaixo trata também de outras questões além das que serão destacadas.

### Roteamentos Alternativos

Os autores apresentados nesta seção trabalham com a flexibilidade de roteamento ao se utilizar da duplicação de máquinas, uso de máquinas alternativas para as operações ou a inclusão de tipos de máquinas com vários exemplares.

Logendran et al. (1994) apresentou duas heurísticas diferentes baseadas na busca tabu. Eles consideraram máquinas alternativas para executarem a operação de uma parte em duas fases sequenciais. A primeira trata do problema de determinar o número de máquinas de cada tipo e um único plano de processamento para cada parte. Na segunda fase, temos a formação das células de manufatura, minimizando a soma do custo de aquisição das máquinas e o custo anual de operação para todas as partes processadas.

Kazerooni et al. (1997) definiu uma abordagem em dois estágios, aplicando algoritmos genéticos para o projeto de formação de células de manufatura considerando roteamentos alternativos. Nesta pesquisa, a melhor rota alternativa para cada parte é determinada com base no coeficiente de similaridade em um algoritmo genético. Em seguida, as famílias de partes e as células de máquinas são agrupadas numa segunda etapa por outro algoritmo genético, minimizando o custo final da solução: custo total da soma dos custos de manipulação dos materiais, custos de operação e não operação (carregamento e descarregamento das partes), custos fixos de máquinas e custos e duplicação das máquinas.

Wu et al. (2009) desenvolveu um algoritmo *simulated annealing* híbrido com um operador de mutação para a resolução do problema com o objetivo de minimizar os movimentos intercelulares. Com o uso de uma matriz de incidência, este critério é expresso por maximizar a eficácia de agrupamento. Eles utilizam o coeficiente de similaridade definido por Won & Kim (1997) para inicializar a população com escolhas de rotas específicas. Neste algoritmo, o número de células que resultam nos melhores valores da função objetivo é gerado automaticamente. Para preservar a flexibilidade aos usuários, é permitido especificar o número de células desejada.

### Processos Alternativos

Os métodos discutidos nessa seção tratam da flexibilidade de processamento ao se utilizar de diversas rotas ou sequência de máquinas.

Zhao & Wu (2000) introduziu um algoritmo genético usando operadores especiais para agrupar máquinas em células com múltiplos objetivos: minimizar o custo pela movimentação intracelular e intercelular; minimizar o total de variação de carregamento da célula

e o número de exceções. Pesos definidos pelo usuário são usados para converter os múltiplos objetivos em um único objetivo. Roteamentos alternativos são considerados a cada iteração ao selecionar as rotas que minimizem o custo. Forma-se um grafo acíclico para lidar com as rotas alternativas e adota-se programação dinâmica para encontrar a melhor rota.

Em seu modelo multiobjetivo, Gravel et al. (1998) propôs um procedimento com dois laços de algoritmos genéticos. O primeiro deles, o laço externo, utiliza cromossomos representando a alocação de máquinas nas células (cromossomo máquina-célula). Cada cromossomo é avaliado por um segundo, o laço interno, no qual os cromossomos representam a alocação das partes nas rotas (cromossomo parte-rota). A cada iteração do primeiro, cada célula é produzida no laço externo e então avaliada de acordo com a melhor parte-rota encontrada no laço interno.

Adenso-Diaz et al. (2001) propôs uma busca tabu eficiente para simultaneamente selecionar as rotas de processamento para as partes, agrupar essas partes em famílias e máquinas em células, e tem como função objetivo a minimização dos custos de transporte. O problema tratado possui restrições quanto aos tamanhos das células (máximos e mínimos) e agrupamento de máquinas (separação e posicionamento).

### Roteamentos e Processos Alternativos

Esta seção apresenta os métodos que tratam da flexibilidade de roteamento e processamento simultaneamente.

Sofianopoulou (1999) utiliza sequencialmente um *simulated annealing* e um modelo de programação linear para selecionar as rotas das partes e construir as células de máquinas. Motivado por este trabalho, Spiliopoulos & Sofianopoulou (2007) apresentam um algoritmo baseado em *bounding scheme* que examina todas as combinações de roteamentos alternativos e resolve apenas algumas poucas formações de células, reduzindo drasticamente a complexidade do espaço de solução.

Uddin & Shanker (2002) desenvolvem um procedimento baseado em algoritmo genético para resolver o problema quando cada parte tem mais que uma rota de processamento. O problema de simultaneamente atribuir máquinas e rotas de processamento (partes) às células é formulado por um problema de programação inteira. Na inicialização, o conjunto de rotas de processamento é escolhido aleatoriamente. A rota de processamento é definida por um conjunto de máquinas visitadas com um tempo de processamento associada a cada máquina, sem que leve em consideração as sequências das operações. Assim sendo, o objetivo de minimizar a movimentação intercelular é alcançada ao minimizar o número de visitas em cada célula. O tamanho da célula também é restrito por limites máximos e mínimos.

### Tabela Comparativa

Na literatura citada anteriormente são apresentadas diversas abordagens meta-heurísticas para o problema de formação de células de manufatura com planos de processamento alternativos. O conjunto dos artigos publicados estão sumarizados na Tabela 3.2 com suas respectivas características. Primeiramente, destaca-se quando o método empregado trata as sequências de operações (SO), flexibilidade de roteamento (FR) e de processamento (FP). Além disso, quando os autores enumeram todas as rotas de processamento a fim de lidar com todas alternativas, marcamos esta situação em ENUM. A enumeração de todas as soluções implica que o método tende a consumir muitos recursos computacionais, levando a hipótese de que não seja aplicável em problemas de larga escala.

Tabela 3.2: Características da amostra de trabalhos publicados.

Artigos	SO	FR	FP	ENUM
Logendran et al. (1994)	✓	✓		
Kazerooni et al. (1997)	✓	✓		
Gravel et al. (1998)			✓	✓
Zhao & Wu (2000)	✓		✓	
Adenso-Diaz et al. (2001)	✓		✓	
Uddin & Shanker (2002)		✓	✓	
Spiliopoulos & Sofianopoulou (2007)	✓	✓	✓	✓
Wu et al. (2009)		✓		
Esta tese	✓	✓	✓	

Existem diversos outros artigos que tratam do problema de formação de células de manufatura com a adoção de múltiplos critérios de avaliação e que sejam adaptáveis em larga escala. No entanto, artigos que tratam da flexibilidade de roteamento e processamento e que sejam eficientes em larga escala são escassos. A razão para isto é que a complexidade do problema aumenta rapidamente com o uso de máquinas e processos alternativos. Como será demonstrado nos capítulos seguintes, esse estudo supre essa necessidade ao apresentar um algoritmo eficiente e flexível para o problema, ou seja, que trate todos esses fatores em um baixo tempo computacional e que encontre soluções de boa qualidade, além de possibilitar a inclusão futura de outros fatores de produção.

# Capítulo 4

## Proposta

Este trabalho tem por objetivo tratar do PFCM com Sequencia de Operações e Processos Alternativos (PFCM-SOPA), em que se inclui a possibilidade de escolha entre múltiplos planos de processamento, cada um com uma sequência específica de operações a serem realizadas. Nas seções a seguir, apresentamos os algoritmos propostos para a resolução desse problema. Por meio de um algoritmo baseado na meta-heurística ILS, primeiramente trata-se o problema clássico, tomando como entrada uma matriz de entrada binária (ver Figura 2.1). O algoritmo é projetado visando um certo grau de flexibilidade, ao ponto de que ao se adaptar a construção das soluções iniciais e adicionar algumas estruturas de vizinhança específicas para o problema, seja possível obter soluções válidas, de qualidade e em baixo tempo computacional. A Seção 4.2 mostra a extensão realizada no algoritmo de modo a resolver o PFCM-SOPA.

### 4.1 Algoritmo ILS-RVND

Nesta seção descrevemos o método empregado para a resolução do PFCM, baseado na meta-heurística ILS. A essência do ILS consiste em focar a busca em subconjuntos menores do espaço de solução, definido pelos ótimos locais de uma dada heurística (Lourenço et al., 2002). O ILS é executado iterativamente até que um critério de parada seja cumprido.

No entanto, ao invés de reiniciar o mesmo procedimento de uma solução completamente nova a cada iteração, realiza certo grau de perturbação na solução atual, retroalimentando o algoritmo com novas soluções. Este procedimento tem o objetivo de direcionar a busca de forma a escapar de ótimos locais no espaço de soluções, mas que mantenha alguma das características presentes nesta solução. Para tal, deve-se considerar que esta nova solução seja suficientemente diferente da anterior para que essas modificações não sejam desfeitas na próxima iteração após a aplicação de uma busca local. Ao mesmo tempo também não se deseja que essas mudanças levem a uma outra solução completamente diferente, em que se elimina completamente as vantagens da amostragem realizada pelo método

construtivo, mas sim de forma que seja possível manter algumas das características que tornaram a solução a ser perturbada a melhor até o momento.

Contudo, métodos híbridos baseados em meta-heurísticas populares tem se mostrado bastante promissoras quanto à qualidade das soluções apresentadas (Benlic & Hao, 2013). Neste trabalho, empregamos em conjunto com o ILS, uma variante do método VND como procedimento de busca local, baseado numa ordenação aleatória das estruturas de vizinhança, conhecida como RVND. Esta variante tem sido utilizada com sucesso por Penna et al. (2013); Subramanian et al. (2013); Silva et al. (2012).

Para implementar o ILS-RVND, apresentado no Algorithm 4.1, quatro métodos principais são especificados: (i) GENERATESEED, que cria as sementes utilizadas como guia para o método construtivo; (ii) CONSTRUCT, onde uma solução inicial é construída (Seção 4.1.2); (iii) RVND, que realiza o procedimento de busca local baseado no RVND (Seção 4.1.3); (iv) PERTURB, onde uma nova solução inicial é gerada através de uma perturbação na solução atual (Seção 4.1.4)

---

**Algoritmo 4.1** ILS-RVND
 

---

```

1: procedimento ILS-RVND( $A, k$ )
2:    $it, it_{ILS} \leftarrow 0$ 
3:    $T_m \leftarrow \text{GENERATESEED}(A)$ 
4:    $S \leftarrow \text{CONSTRUCT}(T_m, 1)$ 
5:    $S^* \leftarrow S$ 
6:   repita
7:      $S' \leftarrow \text{RVND}(S)$ 
8:     se  $\mu(S') > \mu(S^*)$  então
9:        $S^* \leftarrow S'$ 
10:     $it \leftarrow 0$ 
11:    fim se
12:    se  $it = |M|$  então
13:       $S \leftarrow \text{CONSTRUCT}(T_m, k)$ 
14:    senão
15:       $S \leftarrow \text{PERTURB}(S')$ 
16:    fim se
17:     $it \leftarrow it + 1$ 
18:  até  $it = k \times |M|$ 
19:  retorne  $S^*$ 
20: fim procedimento

```

---

A entrada consiste da matriz parte-máquina  $A$  e um fator de aleatoriedade  $k$  a ser usado no método construtivo. Na matriz  $A$ , cada elemento  $a_{ij}$  representa o uso da parte  $i \in P$  por uma máquina  $j \in M$ . Primeiro, geramos uma semente (passo i), que é utilizada pelo método construtivo (passo ii) para criar boas soluções iniciais. Uma solução  $S$  é composta por máquinas agrupadas em células e partes em famílias, definidas como dois conjuntos denominados  $MC$  and  $PF$ , respectivamente. Estes conjuntos são então utilizados na fase de busca local (passo iii) e uma fase de perturbação (passo iv) no intuito de melhorar a solução atual.

No Algoritmo 4.1, os passos (iii) e (iv) são executados iterativamente até que um

número fixo de iterações seja alcançado; o número de iterações é baseado no tamanho da entrada. Uma característica particular desse algoritmo é que o método construtivo é executado em duas situações distintas. Primeiro, antes das iterações do ILS começarem de fato, executamos um método construtivo guloso ( $k=1$ ); testes preliminares mostraram que este método frequentemente gera soluções que possuem um valor de eficácia de agrupamento próximo ou igual ao das melhores soluções apresentadas na literatura, provendo um bom ponto de partida para as iterações do ILS e conseqüentemente reduzindo o número de iterações estabelecidas como critério de parada. Segundo, após certo número de iterações, o método construtivo é executado novamente utilizando o fator de aleatoriedade  $k$ . A seguir, cada um desses procedimentos será descrito em detalhes.

### 4.1.1 Formação das Sementes

Primeiramente, cria-se uma clique de máquinas  $Cm$  baseada na matriz de entrada  $A$ .  $Cm$  é um grafo completo no qual cada vértice corresponde às máquinas e o peso nas arestas são dados pela expressão  $d_{ij} = \sum_{k=1}^{|P|} |a_{ki} - a_{kj}|$ , que representa a “distância” entre as máquinas  $i$  e  $j$ , isto é, o número de partes operadas por apenas uma das máquinas. Em seguida, obtém-se uma árvore geradora mínima  $Tm$ , dado  $Cm$  como entrada. Em tal árvore, as arestas representam um conjunto de pares de máquinas com a maior similaridade possível considerando a formação de um grafo acíclico e conexo e, portanto, possuem uma grande probabilidade de pertencerem a uma mesma célula.

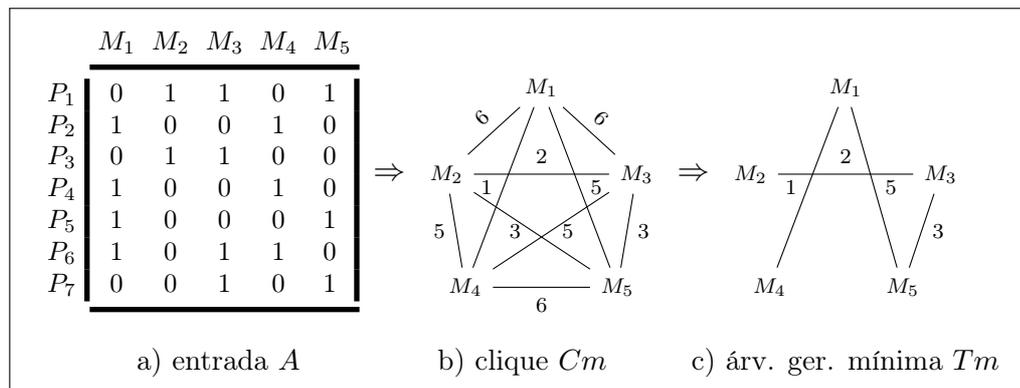


Figura 4.1: Geração das sementes.

### 4.1.2 Construção de Soluções Iniciais

Na fase construtiva, a semente  $Tm$  é usada na criação de soluções iniciais para o problema. No Algoritmo 4.2 temos um resumo do algoritmo proposto. Dado como entrada a árvore geradora mínima  $Tm$  e um parâmetro  $k$ , soluções intermediárias  $A^i$  serão construídas (linha 7) a partir da remoção de cada uma das arestas  $i \in E(Tm)$  (linha 4). O objetivo inicial será o de buscar uma aresta na qual sua remoção gere uma solução  $A'$  com

um melhor valor de eficácia, utilizando o parâmetro  $k$  de entrada para escolher  $A'$  aleatoriamente dentre as  $k$  melhores soluções intermediárias  $A^i$  (linha 9). Por fim, atualiza-se a semente  $Tm$  (linha 10) e repete-se esse procedimento até que não haja mais arestas a remover, retornando a melhor dentre todas as soluções obtidas.

---

**Algoritmo 4.2** Método Construtivo
 

---

```

1: procedimento CONSTRUCT( $Tm, k$ )
2:   enquanto  $E(Tm) \neq \emptyset$  faça
3:     para todo  $i \in E(Tm)$  faça           ▷ Obtêm todas as soluções intermediárias  $A^i$  dado  $Tm$ 
4:        $Tm^i \leftarrow Tm - \{i\}$ 
5:       gera  $MC^i$  de  $Tm^i$ 
6:       gera  $PF^i$  de  $MC^i$ 
7:       combina  $MC^i$  e  $PF^i$  para obter a solução intermediária  $A^i$ 
8:     fim para
9:     escolhe aleatoriamente uma solução  $A'$  dentre as  $k$  melhores soluções  $A^i$ 
10:     $Tm \leftarrow Tm'$                        ▷ Atualiza a semente  $Tm$ 
11:    se  $\mu' > \mu^*$  então
12:       $A^* \leftarrow A'$                        ▷ Salva se solução melhor for encontrada
13:    fim se
14:  fim enquanto
15:  retorne  $A^*$ 
16: fim procedimento
  
```

---

A construção de uma solução intermediária  $A^i$ , mencionada no parágrafo anterior, é realizada da seguinte forma: primeiro remove-se a aresta  $i$ , obtêm-se uma floresta resultante  $Tm^i$  que define um conjunto de células  $MC^i$  (linha 5), como mostrado na Figura 4.2. Então, dado  $MC^i$ , cada parte  $p \in P$  é associada a uma das células em  $MC^i$  de modo a produzir o menor número de lacunas e exceções, formando assim um conjunto de famílias de partes. Por fim,  $MC^i$ , e  $PF^i$  são combinados para gerar a solução intermediária  $A^i$ . Os exemplos a seguir ilustram essa etapa do Algoritmo 4.2. Na Figura 4.2, temos que  $MC^i = \{\{M_1, M_4\}; \{M_2, M_3, M_5\}\}$ , é obtido através da remoção da aresta  $(M_1, M_5)$  de  $Tm$  e tomando cada componente conexa como uma célula em  $MC^i$ .

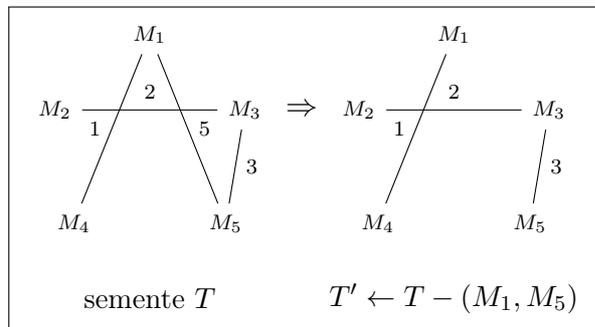


Figura 4.2: Linhas 4 e 5 do Algoritmo 4.2.

A seguir, na linha 6, cada parte será associada com exatamente uma célula de  $MC^i$  de modo a produzir o menor número de lacunas e exceções, formando assim um conjunto de famílias de partes  $PF^i$ . Ilustramos esse processo na Tabela 4.3.

	$M_1$	$M_4$	$M_2$	$M_3$	$M_5$
$P_1$	0	0	<b>1</b>	<b>1</b>	<b>1</b>
$P_2$	<b>1</b>	<b>1</b>	0	0	0
$P_3$	0	1	<b>1</b>	<b>1</b>	<b>0</b>
$P_4$	<b>1</b>	<b>1</b>	0	0	0
$P_5$	<b>1</b>	<b>0</b>	0	0	1
$P_6$	<b>1</b>	<b>1</b>	0	1	0
$P_7$	0	0	<b>0</b>	<b>1</b>	<b>1</b>

Figura 4.3: Linha 6 do Algoritmo 4.2.

Cada parte  $p \in P$  pode ser associada tanto à célula  $\{M_1, M_4\}$  quanto à  $\{M_2, M_3, M_5\}$ , dependendo de qual nos dá um custo menor, considerando o número de lacunas e exceções formada. Por exemplo, associar  $P_5$  com a célula  $\{M_1, M_4\}$  produz uma lacuna, mas fazer essa associação com a célula  $\{M_2, M_3, M_5\}$  produz duas lacunas; portanto  $P_5$  deve ser associada com a célula  $\{M_1, M_4\}$  (marcado em negrito na Figura 4.3). Após associar todas as partes em suas respectivas células, obtemos o conjunto de família de partes  $PF^i = \{\{P_2, P_4, P_5, P_6\}; \{P_1, P_3, P_7\}\}$ .

### 4.1.3 Busca por Melhores Soluções

Ao fim da fase de construção, obtem-se uma solução inicial para o problema. Então, uma fase de busca local baseada no método VND (Mladenovic & Hansen, 1997) é executada com o propósito de melhorar tal solução. O método VND consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, aceitando apenas soluções que melhorem a solução atual e retornando à primeira estrutura quando uma melhor solução for encontrada. Usamos uma variante que consiste em usar uma ordenação aleatória dessas estruturas, como mostrado no Algoritmo 4.3.

Seja  $V = \{V^1, \dots, V^r\}$  uma conjunto de estruturas de vizinhança. Sempre que uma busca em uma dada vizinhança falhar em melhorar uma solução atual, RVND executa a busca na seguinte. A ordem de visitação é pré-definida aleatoriamente, e chamadas subsequentes ao Algoritmo 4.3 irá possivelmente definir uma outra ordem de visitação.

Neste trabalho, três estruturas de vizinhança foram empregadas: *Reagrupamento*, *Deslocamento de Máquinas* e *Troca de Máquinas*. Buscas nessas vizinhanças baseiam-se na realização de movimentos de máquinas entre as células e encontrar a família de partes correspondente a essa novo agrupamento, esta etapa é realizada de forma similar a forma empregada na linha 6 do Algoritmo 4.2, em que para parte verifica-se qual célula deve pertencer de modo a minimizar o número de lacunas e exceções. Tendo em vista que no pior caso, o número de células é igual ao número de máquinas, a complexidade computacional da busca em cada uma dessas vizinhanças é  $O(n^2)$ .

**Algoritmo 4.3** RVND

---

```

1: procedimento RVND( $A$ )
2:   inicialize aleatoriamente  $V$ 
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

```

---

**Reagrupamento:** Dado um conjunto de famílias de partes  $PF$  de uma solução atual, reagrupa as máquinas em um novo conjunto de células de máquinas, associando cada máquina a uma família de partes que resulte em um menor custo, considerando o número de lacunas e exceções.

	$M_2$	$M_3$	$M_5$	$M_1$	$M_4$		$M_2$	$M_3$	$M_5$	$M_1$	$M_4$		$M_2$	$M_3$	$M_5$	$M_1$	$M_4$
$P_1$	1	1	1	0	0	$\Rightarrow$	1	1	1	0	0	$\Rightarrow$	1	1	1	0	0
$P_3$	1	1	0	0	0		1	1	0	0	0		1	1	0	0	0
$P_7$	0	1	1	0	0		0	1	1	0	0		0	1	1	0	0
$P_2$	0	0	1	1	1		0	0	1	1	1		0	0	0	1	1
$P_4$	0	0	0	1	1		0	0	0	1	1		0	0	0	1	1
$P_5$	0	0	1	1	0		0	0	1	1	0		0	0	1	1	0
$P_6$	0	1	0	1	1		0	1	0	1	1		0	1	0	1	1
	a) solução $S$						b) família de partes fixado						c) solução $S'$				

Figura 4.4: Reagrupando uma solução.

Na Figura 4.4 temos um exemplo em que se parte de uma solução inicial  $S$  fixando a família de partes dessa solução. Ao verificar para cada máquina qual família de partes gera menos lacunas e exceções, tem-se que a máquina  $M_5$  quando se associa à família  $\{P_1, P_3, P_7\}$  gera 1 lacuna a menos que quando está associada à família  $\{P_2, P_4, P_5, P_6\}$ , formando-se assim uma nova solução  $S'$  com um novo agrupamento de máquinas.

**Deslocamento de Máquinas:** Dado um conjunto de células de máquinas  $MC$  da solução atual, move máquinas de uma célula para outra. A Figura 4.5 ilustra esse movimento, em que a máquina  $M_2$  é movida da célula  $\{M_5, M_2\}$  para a  $\{M_3\}$ .

	$M_3$	$M_1$	$M_4$	$M_5$	$M_2$		$M_2$	$M_3$	$M_1$	$M_4$	$M_5$	
$P_3$	1	0	0	0	1	$\Rightarrow$	$P_1$	1	1	0	0	1
$P_7$	1	0	0	1	0		$P_3$	1	1	0	0	0
$P_2$	0	1	1	0	0		$P_2$	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
a) $\mu = \frac{16-5}{16+1} = 0.65$							b) $\mu = \frac{16-4}{16+0} = 0.75$					

Figura 4.5: Máquina  $M_2$  movida da célula  $\{M_5, M_2\}$  para a  $\{M_3\}$ .

**Troca de Máquinas:** Dado um conjunto de células de máquinas  $MC$  da solução atual, executa uma troca de máquinas provenientes de células diferentes. Ao contrário do Deslocamento de Máquinas, não possui a habilidade de esvaziar uma célula, ou seja, não há diminuição no número de células. A Figura 4.6 ilustra esse procedimento, em que uma troca é executada entre as máquinas  $M_2$  e  $M_5$ .

	$M_5$	$M_3$	$M_1$	$M_4$	$M_2$		$M_2$	$M_3$	$M_1$	$M_4$	$M_5$	
$P_1$	1	1	0	0	1	$\Rightarrow$	$P_1$	1	1	0	0	1
$P_7$	1	1	0	0	0		$P_3$	1	1	0	0	0
$P_2$	0	0	1	1	0		$P_2$	0	0	1	1	0
$P_4$	0	0	1	1	0		$P_4$	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
a) $\mu = \frac{16-4}{16+1} = 0.71$							b) $\mu = \frac{16-4}{16+0} = 0.75$					

Figura 4.6: Troca entre as máquinas  $M_2$  e  $M_5$ .

#### 4.1.4 Perturbação de Soluções

Como descrito no Algoritmo 4.1, o ILS consiste em explorar iterativamente a vizinhança das solução ao perturbar ótimos locais anteriormente visitadas. Apresentamos a seguir os três procedimentos de perturbação propostos para o PFCM. Uma vez que o procedimento PERTURB é chamado, uma destas perturbações será selecionada.

**Deslocamento Múltiplo de Máquinas:** executa um número aleatório (entre 1 e  $|C|$ ) de sucessivos Deslocamentos de Máquinas. Para tal, ambas, máquina e célula para

a qual essa máquina irá se deslocar, são escolhidas aleatoriamente. Um ponto a se notar aqui é quanto ao número de movimentos executados. Um número baixo de movimentos resulta em poucas mudanças na solução e, portanto, buscas locais subsequentes podem desfazer essas mudanças. Por outro lado um alto número de movimentos tornam as buscas praticamente aleatórias. Neste trabalho, utilizamos sempre um número de movimentos que varia entre um e o número de células presente na solução atual.

**Remoção de Célula:** seleciona aleatoriamente uma célula e a remove, distribuindo todas as máquinas pertencentes a esta célula para as demais, também escolhidas de forma aleatória. Esta perturbação pode ser vista como a Deslocamento Múltiplo de Máquinas em que todas as máquinas são escolhidas da mesma célula. O objetivo da Remoção de Células é a de diminuir o número de células das soluções. Na Figura 4.7 temos um exemplo em que a célula  $\{M_1, M_4\}$  é escolhida para remoção; as máquinas  $M_1$  e  $M_5$  são movidas aleatoriamente para as outras células.

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

Figura 4.7: Remoção da célula  $\{M_1, M_4\}$ .

**Divisão de Célula:** Seleciona aleatoriamente uma célula e a divide em duas novas células  $M'$  e  $M''$ . A probabilidade de cada máquina ser movida para  $M'$  ou  $M''$  é fixada em 50%. Ao contrário das demais perturbações, a Divisão de Célula nunca move uma máquina para células existentes. O objetivo da Divisão de Célula é a de aumentar o número de células. Na Figura 4.8 temos um exemplo em que a célula  $\{M_2, M_3, M_5\}$  é dividida em duas novas células:  $\{M_2, M_3\}$  e  $\{M_5\}$ .

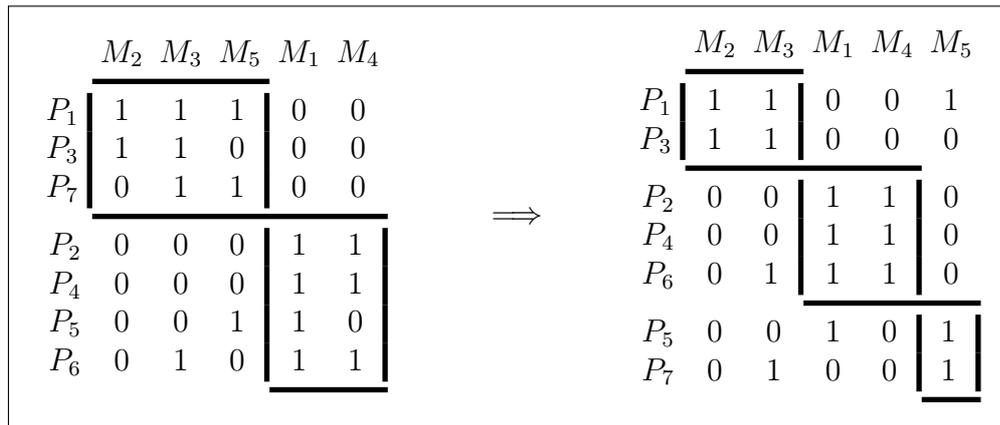


Figura 4.8: Célula  $\{M_2, M_3, M_5\}$  é dividida em duas novas células.

## 4.2 Estendendo o ILS-RVND para a Resolução do PFCM-SOPA

Nesta seção, apresentamos as adaptações realizadas no ILS-RVND proposto para a resolução do PFCM-SOPA. As adaptações foram realizadas apenas na composição dos procedimentos internos do ILS-RVND: método construtivo, estruturas de vizinhança e perturbações; a estrutura geral do ILS se manteve. Para tal, inicialmente definimos a nova formação das sementes para a criação de soluções iniciais, levando em consideração os roteamentos e os processos alternativos. Em seguida, propomos duas novas estruturas de vizinhança e perturbações, de forma a atender os novos critérios de solução. Como vimos na Seção 2.5, ao se utilizar sequência de operações, a eficácia de agrupamento perde o sentido para se avaliar a qualidade da solução. Neste caso, a avaliação se dará pelo custo total de movimentação que será minimizada.

### 4.2.1 Representação do Problema

Quando tratamos de processos alternativos, estamos incluindo ao problema de agrupar máquinas em células e partes em famílias, um problema adicional que trata da escolha do plano de processamento a ser utilizado na elaboração de cada parte. Ao mesmo tempo, quando temos especificado no plano de processamento o tipo de máquina que irá realizar as operações, temos a possibilidade de múltiplas máquinas de um mesmo tipo. Faz-se então necessário a definição da rota pela qual as operações deverão ser realizadas, isto é, qual conjunto específico de máquinas realizará a operação designada, uma vez que máquinas de um mesmo tipo podem ser alocadas em células diferentes. Temos na Figura 4.9 uma representação da entrada do problema considerando esses novos critérios, como explicado em detalhes na Seção 2.3.

Na primeira coluna, temos os planos de processamento  $P_i^j$ , em que  $i$  é a indicação

	$T_1^1$	$T_2^2$	$T_3^2$	$T_4^1$	$T_5^1$
$P_1^a$	3	0	2	1	4
$P_1^b$	3	4	2	1	0
$P_2^a$	1	0	2	4	3
$P_3^a$	0	0	1	0	0
$P_3^b$	0	1	0	0	2
$P_3^c$	2	0	0	1	0
$P_4^a$	2	1	4	0	3

Figura 4.9: Dado de entrada do PFCM-SOPA

de qual parte se trata e  $j$  o índice que representa alternativa do plano a ser escolhido. Enquanto isso, na primeira linha temos os tipos de máquinas  $T_m^n$ , em que  $m$  é o índice que identifica o tipo e  $n$  o número de máquinas deste tipo.

### 4.2.2 Construção de Soluções Iniciais

Dada a entrada  $S$ , primeiro criamos as sementes para a construção de boas soluções iniciais. Utilizamos uma metodologia semelhante à definida na Seção 4.1.1, aqui considerado uma clique de tipos de máquinas  $Ct$ .

Nessa etapa, vale ressaltar a importância que o uso de sementes representa para a criação das soluções iniciais. Quando criamos a clique de máquinas, usando como base a similaridade para definir o peso nas arestas, estamos aplicando um elemento central para qualquer possível extensão do problema ao apontar uma direção para a formação de boas soluções, focada no agrupamento das máquinas em células e partes em famílias.

Com o acréscimo de múltiplas rotas e planos de processamento, a similaridade continua exercendo um papel muito importante, e quaisquer outros dados extras (capacidade, custo, etc) podem ser atribuídos em forma de pesos aos vértices dessa clique para serem usados de alguma forma na geração das soluções, caso necessário.

No caso específico dessa extensão do problema, em que a criação das sementes se dará por meio de um clique de tipos de máquinas, ou seja, cada vértice representando um conjunto de máquinas, adotamos pesos nos vértices para representar a quantidade de máquinas do tipo em questão. Já o peso nas arestas é calculado com base na dissimilaridade entre dois tipos de máquinas no contexto de múltiplas rotas de processamento. Esse valor é obtido de modo semelhante a quando existe apenas um plano de processamento para cada parte, ou seja, contabilizando a quantidade de vezes em que uma determinada parte é processada apenas por uma das máquinas. Considerando múltiplos planos de processamento, basta realizar essa contagem para cada plano de processamento.

Com base na semente  $Ct$ , geramos uma solução inicial para o problema, por meio de uma adaptação do Algoritmo 4.2 com  $k = 1$  (guloso). Esta adaptação é realizada em

duas fases:

1. na etapa especificada na linha 6, como temos agora vários planos de processamento para cada parte, escolhe-se previamente de forma aleatória qual o plano a ser utilizado;
2. após a construção da solução com tipos das máquinas, definem-se as máquinas específicas de cada tipo de forma que produzam o menor número de exceções.

Na Figura 4.10 temos um exemplo de construção de uma solução inicial no momento em que os planos de processamento já foram escolhidos aleatoriamente para cada parte (fase 1), restando agora decidir em qual célula devem pertencer as máquinas de determinado tipo (fase 2). Por simplificação, suponha que apenas o tipo  $T_2$  possua duas máquinas:  $M_{2,1}$  e  $M_{2,2}$ . Como temos apenas duas células, atribuímos uma máquina a cada célula, eliminaremos a exceção formada com a parte  $P_4$ .

$$\begin{array}{c}
 \begin{array}{ccccc}
 & P_1 & P_3 & P_2 & P_4 & P_5 \\
 \hline
 T_2^2 & 1 & 1 & 0 & 1 & 0 \\
 T_3^1 & 3 & 2 & 0 & 0 & 0 \\
 \hline
 T_1^1 & 2 & 0 & 1 & 3 & 2 \\
 T_4^1 & 0 & 0 & 2 & 2 & 1 \\
 \hline
 \end{array}
 & \implies &
 \begin{array}{ccccc}
 & P_1 & P_3 & P_2 & P_4 & P_5 \\
 \hline
 M_{2,1} & 1 & 1 & 0 & 0 & 0 \\
 M_{3,1} & 3 & 2 & 0 & 0 & 0 \\
 \hline
 M_{2,2} & 0 & 0 & 0 & 1 & 0 \\
 M_{1,1} & 2 & 0 & 1 & 3 & 2 \\
 M_{4,1} & 0 & 0 & 2 & 2 & 1 \\
 \hline
 \end{array}
 \end{array}$$

Figura 4.10: Segunda fase da geração inicial de uma solução para o PFCM-SOPA

Com a execução da primeira fase em conjunto com a fixação do parâmetro  $k = 1$ , estamos basicamente transferindo a responsabilidade pela desejada aleatoriedade do algoritmo construtivo para a escolha aleatória do plano de processamento de cada parte, e a responsabilidade por direcionar o algoritmo a melhorar as soluções para as buscas locais. Uma vez formada a solução até esse ponto, é na segunda fase que resolvemos a questão dos tipos de máquinas. Isto é feito levando em consideração que, ao se ter várias máquinas de um mesmo tipo, podem-se atribuir máquinas (teoricamente idênticas) a mais de uma célula e isso sempre irá resultar em uma solução com menos movimentação de material entre as células.

De fato, se para cada tipo máquina tivermos um número de máquinas iguais ao número de células, poderíamos simplesmente atribuir uma máquina para cada célula e com isso eliminar toda e qualquer movimentação de materiais entre as células. O que limita essa questão é a necessidade de maximização do uso da máquina em cada célula, já que deixaria a célula mais ineficiente ao não considerar se há uso suficiente desta máquinas pelas partes associadas à célula que justifique essa duplicação. Além disso, se olharmos para a viabilidade prática, o custo de aquisição de tantas máquinas torna este cenário proibitivo.

### 4.2.3 Estruturas de Vizinhança e Perturbações Adicionais

Nas etapas de busca local e perturbação, todos os procedimentos definidos para o PFCM continuam sendo válidos para o PFCM-SOPA. Entretanto, eles apenas se limitam à etapa de formação das células, isto é, há uma necessidade em se definir novas estruturas de vizinhança e perturbações que levem em conta os roteamentos e planos de processamento alternativos. A estrutura geral da busca local é a mesma que aquela apresentada na Seção 4.1.3. Contudo, além das estruturas de vizinhanças já apresentadas, propomos duas outras: Troca de Planos de Processamento e Troca de Roteamento.

**Troca de Planos de Processamento:** uma vez que os planos de processamento das partes foram escolhidos de forma aleatória, esta estrutura de vizinhança é composta por todas as soluções formadas pela troca dos planos de processamento por algum outro plano que não foi atribuído à parte em questão. Em outras palavras, dada uma solução em que temos um plano de processamento alocado para cada parte, substitui-se um desses planos por outro que apresente uma solução de melhor qualidade. O tamanho da vizinhança é de  $O(|P|)$  soluções, considerando que cada parte possua no máximo uma constante  $k$  de planos de processamento.

$$\begin{array}{c}
 \begin{array}{c} P_1 \ P_3 \ P_2 \ P_4 \ P_5 \\
 \hline
 T_2^2 \left[ \begin{array}{cc|cc} 1 & 1 & 0 & 1 & 0 \\
 T_3^1 \left[ \begin{array}{cc|cc} 3 & 2 & 0 & 0 & 0 \\
 \hline
 T_1^1 \left[ \begin{array}{ccc|ccc} 2 & 0 & 1 & 3 & 2 \\
 T_4^1 \left[ \begin{array}{ccc|ccc} 0 & 0 & 2 & 2 & 1 \end{array} \right]
 \end{array} \right]
 \end{array} \right]
 \end{array}
 \implies
 \begin{array}{c}
 \begin{array}{c} P'_1 \ P_3 \ P_2 \ P_4 \ P_5 \\
 \hline
 T_2^2 \left[ \begin{array}{cc|cc} 1 & 1 & 0 & 1 & 0 \\
 T_3^1 \left[ \begin{array}{cc|cc} 2 & 2 & 0 & 0 & 0 \\
 \hline
 T_1^1 \left[ \begin{array}{ccc|ccc} 3 & 0 & 1 & 3 & 2 \\
 T_4^1 \left[ \begin{array}{ccc|ccc} 0 & 0 & 2 & 2 & 1 \end{array} \right]
 \end{array} \right]
 \end{array} \right]
 \end{array}
 \end{array}$$

Figura 4.11: Supondo que o plano  $P'_1$  seja uma alternativa ao  $P_1$ , obteve-se menos movimentação de material ao realizar essa troca.

**Troca de Roteamento:** quando o número de máquinas de um determinado tipo é menor que o número de células, ao se distribuir essas máquinas entre as células, haverá células que não irão receber uma máquina deste tipo. Sendo assim, esta estrutura de vizinhança é composta por todas as soluções formadas pela troca de uma máquina específica entre células que não possuem um representante daquele tipo de máquina. O tamanho da vizinhança é de  $O(|C| \cdot |M|)$  soluções, considerando que cada  $O(|M|)$  máquina possuem  $O(|C|)$  células para se deslocar.

Quanto às perturbações, além das propostas para o PFCM, definimos outras duas de forma análoga às buscas locais propostas: Troca Múltipla de Planos de Processamento e Troca Múltipla de Roteamentos.

**Troca Múltipla de Planos de Processamento:** executa um número aleatório (entre 1 e  $|C|$ ) de sucessivas Trocas de Planos de Processamento em que cada plano de processamento que vai substituir o atual também é escolhido de forma aleatória.

**Troca Múltipla de Roteamento:** executa um número aleatório (entre 1 e  $|C|$ ) de sucessivas Trocas de Roteamento em que a célula de destino da máquina também é escolhida de forma aleatória.

# Capítulo 5

## Resultados

### 5.1 PFCM

Apresentamos nesta seção os resultados computacionais para o Algoritmo 4.1 sobre duas variantes do PFCM: o Problema de Formação de Células de Manufatura Permitindo Singleton e o Problema de Formação de Células de Manufatura Rejeitando Singleton. Para avaliar a qualidade das soluções, utilizamos a Eficácia de Agrupamento, como explicado na Seção 2.5. Em todas as tabelas, a Eficácia de Agrupamento é apresentada como porcentagem composta por dois dígitos decimais e o tempo computacional dado em segundos (s). Todos os algoritmos foram executados numa máquina Intel Core i7-2600 com processador de 3.4 GHz e 32 GB of RAM, usando o sistema operacional Arch Linux 3.3.4.

#### 5.1.1 Instâncias

Usamos 35 instâncias da literatura, disponíveis em Gonçalves & Resende (2004), amplamente utilizadas por diversos autores em seus métodos. Essas instâncias são compostas por matrizes com dimensões variando de  $5 \times 7$  até  $40 \times 100$ , com diferentes graus de dificuldade, desde soluções que admitem células sem lacunas e exceções até mal estruturadas, ou seja, com grande número de lacunas e exceções.

As colunas na Tabela 5.1 mostram, da esquerda para a direita: nome, tamanho (devemos lembrar que uma instância é dada como uma matriz de entrada  $A_{m \times p}$ , em que  $m$  é o número de máquinas e  $p$  o número de partes), o valor e o artigo de referência que contém a melhor Eficácia de Agrupamento para o PFCM-RS e para o PFCM-PS, respectivamente. Nas últimas duas colunas, cada citação indica o artigo no qual o melhor valor de Eficácia de Agrupamento apareceu pela primeira vez; valores marcados com “\*” foram comprovados como valores ótimos através do trabalho desenvolvido por Pinheiro et al. (2016).

Tabela 5.1: Instâncias do PFCM usados nos experimentos.

Instância	Dimensão	Melhor PFCM-RS		Melhor PFCM-PS	
		Valor	Referência	Valor	Referência
PFCM01	05 × 07	73.68*	Gonçalves & Resende (2004)	75.00*	Pailla et al. (2010)
PFCM02	05 × 07	62.50*	Gonçalves & Resende (2004)	69.57*	Wu et al. (2009)
PFCM03	05 × 18	79.59*	Gonçalves & Resende (2004)	80.85*	Pailla et al. (2010)
PFCM04	06 × 08	76.92*	Gonçalves & Resende (2004)	79.17*	Pailla et al. (2010)
PFCM05	07 × 11	53.13*	Gonçalves & Resende (2004)	60.87*	Wu et al. (2009)
PFCM06	07 × 11	70.37*	Gonçalves & Resende (2004)	70.83*	Wu et al. (2009)
PFCM07	08 × 12	68.29*	Gonçalves & Resende (2004)	69.44*	Pailla et al. (2010)
PFCM08	08 × 20	85.25*	Gonçalves & Resende (2004)	85.25*	Wu et al. (2009)
PFCM09	08 × 20	58.72*	Gonçalves & Resende (2004)	58.72*	Wu et al. (2009)
PFCM10	10 × 10	70.59*	Gonçalves & Resende (2004)	75.00*	Wu et al. (2009)
PFCM11	10 × 15	92.00*	Gonçalves & Resende (2004)	92.00*	Wu et al. (2009)
PFCM12	14 × 23	69.86*	Gonçalves & Resende (2004)	74.24*	Pailla et al. (2010)
PFCM13	14 × 24	69.33*	Gonçalves & Resende (2004)	72.86*	Pailla et al. (2010)
PFCM14	16 × 24	51.96*	Gonçalves & Resende (2004)	53.33*	Pailla et al. (2010)
PFCM15	16 × 30	67.83*	Gonçalves & Resende (2004)	69.92*	Pailla et al. (2010)
PFCM16	16 × 43	56.52*	Pinheiro et al. (2016)	58.04*	Pinheiro et al. (2016)
PFCM17	18 × 24	54.46*	Gonçalves & Resende (2004)	57.73*	Pailla et al. (2010)
PFCM18	20 × 20	42.96	Gonçalves & Resende (2004)	43.97	Pailla et al. (2010)
PFCM19	20 × 23	49.65*	Gonçalves & Resende (2004)	50.81*	Wu et al. (2009)
PFCM20	20 × 35	76.54*	Wu et al. (2010)	79.38*	Pailla et al. (2010)
PFCM21	20 × 35	58.15*	Wu et al. (2010)	58.79*	Pailla et al. (2010)
PFCM22	24 × 40	100.00*	Gonçalves & Resende (2004)	100.00*	Wu et al. (2009)
PFCM23	24 × 40	85.11*	Gonçalves & Resende (2004)	85.11*	Wu et al. (2009)
PFCM24	24 × 40	73.51*	Gonçalves & Resende (2004)	73.51*	Wu et al. (2009)
PFCM25	24 × 40	51.97*	Gonçalves & Resende (2004)	53.29*	Wu et al. (2009)
PFCM26	24 × 40	47.37	Wu et al. (2010)	48.95	Elbenani et al. (2012)
PFCM27	24 × 40	44.87	Gonçalves & Resende (2004)	46.58	Elbenani et al. (2012)
PFCM28	27 × 27	54.27	Gonçalves & Resende (2004)	54.82	Pailla et al. (2010)
PFCM29	28 × 46	46.06	Wu et al. (2010)	47.68	Pailla et al. (2010)
PFCM30	30 × 51	58.94*	Pinheiro et al. (2016)	63.04*	Pinheiro et al. (2016)
PFCM31	30 × 50	59.66*	Gonçalves & Resende (2004)	59.77*	Pinheiro et al. (2016)
PFCM32	30 × 50	50.51	Gonçalves & Resende (2004)	50.83	Wu et al. (2009)
PFCM33	30 × 90	42.64	Gonçalves & Resende (2004)	47.93	Pailla et al. (2010)
PFCM34	37 × 53	59.85	Wu et al. (2010)	61.16	Pailla et al. (2010)
PFCM35	40 × 100	84.03*	Gonçalves & Resende (2004)	84.03*	Wu et al. (2009)

\* valores ótimos comprovados em Pinheiro et al. (2016).

## 5.1.2 Avaliação de Desempenho

Nesta seção, avaliamos o desempenho dos métodos propostos em relação a qualidade das soluções. Executamos duas baterias de testes, uma para o PFCM-PS e outra para o PFCM-RS. Em cada conjunto, executamos dez vezes o ILS-RVND para cada instância; o número de iterações em cada execução foi baseado no número  $|M|$  da instância e do fator de aleatoriedade  $k$ . Como mostrado no Algoritmo 4.1, a cada vez que uma solução é melhorada o contador de iterações é reiniciado; quando o contador chegar em  $k \times |M|$ , o algoritmo para. Após alguns experimentos preliminares, fixamos  $k = 2$ , para o qual ficou constatado que aumentar este valor representa um maior tempo de execução do algoritmo sem um ganho significativo na qualidade das soluções testadas.

A seguir serão apresentados os resultados dessa avaliação sobre o ILS-RVND proposto. Na Figura 5.1 mostramos o desvio padrão da qualidade da solução para cada instância. As barras vermelhas listradas representam o desvio padrão para a versão do problema em que células *singletons* são permitidos e as barras cinzas sólidas as que essas células não são permitidas.

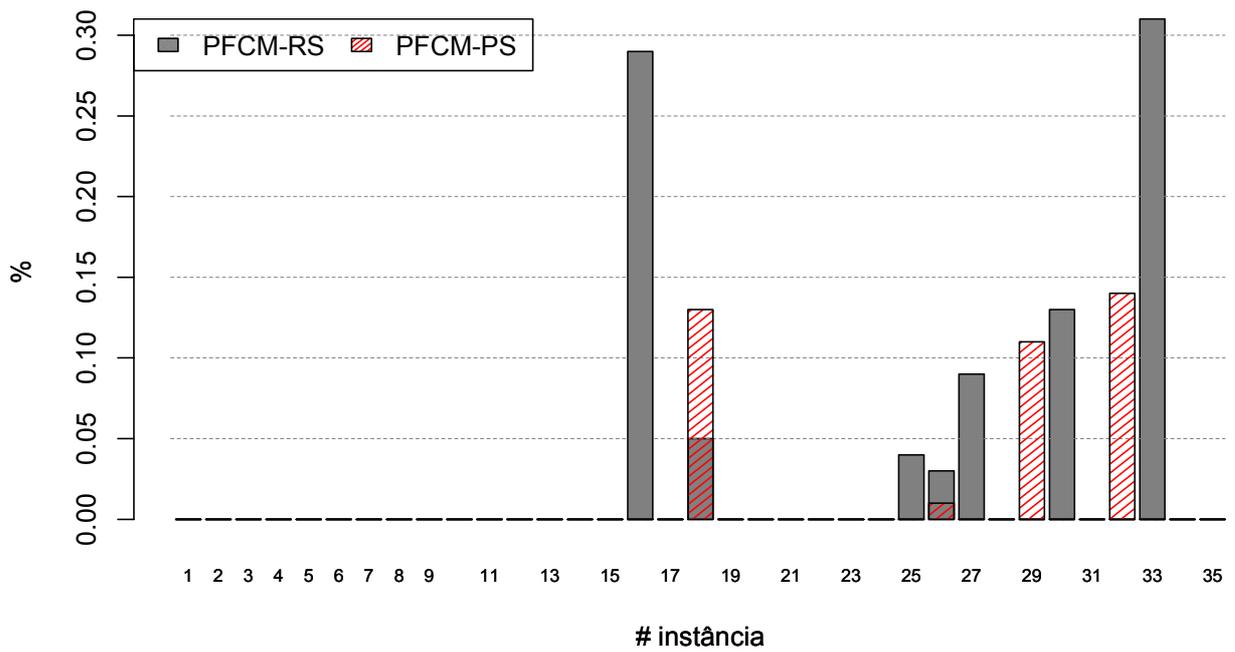


Figura 5.1: Valor de desvio padrão para o PFCM-RS e PFCM-PS.

Na Figura 5.2 temos o tempo médio de execução em segundos que o algoritmo proposto leva para encontrar uma solução em cada uma das variantes do problema para as 35 instâncias da literatura.



Figura 5.2: Tempo médio gasto para encontrar uma solução.

Os resultados comprovam a robustez do método proposto, como pode ser constatado pelos baixos valores de desvio padrão e tempo computacional. Note que o desvio padrão só não é zero para 4 casos do PFCM-PS e 7 no PFCM-RS. Pode ser observado também

que quando *singletons* são permitidos, o método necessita de um pouco mais de esforço computacional para encontrar a solução; isto ocorre pois o PFCM-PS tem um espaço de busca ligeiramente maior.

### 5.1.3 Comparação com a Literatura

Agora iremos comparar os resultados apresentados na Seção 5.1.2 com vários algoritmos da literatura, no mesmo conjunto de instâncias.

Para cada par  $(I, A)$  em que  $I$  é uma instância e  $A$  é um método, calcularemos o *gap* entre o valor da solução obtida por  $A$  quando aplicado em  $I$  e o valor da melhor solução conhecida para  $I$  (mostrado na Tabela 5.1), através da seguinte fórmula:

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

em que  $\mu$  é a Eficácia de Agrupamento da solução  $s$  obtida pelo método  $A$  quando aplicado a  $I$ , e  $\mu(s^*)$  é a Eficácia de Agrupamento para a melhor solução conhecida  $s^*$  para  $I$ .

Nas tabelas seguintes, teremos  $\%gap > 0$  quando a solução possuir uma qualidade pior que a melhor solução encontrada na literatura,  $gap = 0$  quando possuir a mesma qualidade e  $\%gap < 0$  quando possuir uma melhor qualidade. Instâncias marcadas com “\*” possuem ótimo conhecido. Portanto, nesses casos, se  $\%gap = 0$  então o método encontrou uma solução ótima. Nas instâncias em que as soluções apresentadas não possuem soluções ótimas comprovadas, um  $\%gap = 0$  significará que tal método somente foi capaz de encontrar uma solução de mesma qualidade que a melhor disponível na literatura.

Logicamente, seguindo essa metodologia, um *gap* negativo deve ser impossível de ocorrer nas colunas referentes aos métodos da literatura, e somente poderá ocorrer em soluções apresentadas nas colunas do ILS-RVND em instâncias não marcadas com “\*”. No entanto, alguns trabalhos demonstraram este comportamento em algumas instâncias. Segundo Pinheiro et al. (2016), os erros provavelmente foram provocados por inconsistências nos arquivos de instâncias utilizadas pelos autores desses trabalhos em específico. Quando tal situação ocorre, marcamos com “×” e desconsideramos a solução apresentada por esses algoritmos para efeito de comparação. Como não tivemos acesso aos arquivos das instâncias inconsistentes, não pudemos aplicar nosso algoritmo para avaliação.

Vamos considerar primeiramente o PFCM-RS. Na Tabela 5.2 apresentamos uma comparação entre o método ILS-RVND proposto e as seguintes abordagens encontrados na literatura:

SACF: *Simulated Annealing for Cell Formation* (Wu et al., 2008)

WFACF: *Water Flow-like Algorithm for Cell Formation* (Wu et al., 2010)

EA: *Evolutionary Algorithm* (Gonçalves & Resende, 2004)

Nela mostramos o %gap para cada par  $(I, A)$ , além disso, para cada método  $A$ , a tabela mostra o número de vezes que  $A$  encontra uma solução ótima ou a melhor solução conhecida.

Tabela 5.2: PFCM-RS: comparação entre o ILS-RVNS e outras abordagens da literatura.

Instancia	SACF	WFACF	EA	ILS-RVND	Instance	SACF	WFACF	EA	ILS-RVND
PFCM01*	0,00	0,00	0,00	0,00	PFCM19*	0,00	0,08	0,00	0,00
PFCM02*	0,00	0,00	0,00	0,00	PFCM20*	0,52	0,00	0,42	0,00
PFCM03*	0,00	0,00	0,00	0,00	PFCM21*	3,63	0,00	0,14	0,00
PFCM04*	0,00	0,00	0,00	0,00	PFCM22*	0,00	0,00	0,00	0,00
PFCM05*	0,00	0,00	0,00	0,00	PFCM23*	0,00	0,00	0,00	0,00
PFCM06*	0,00	0,00	0,00	0,00	PFCM24*	0,00	0,00	0,00	0,00
PFCM07*	0,00	0,00	0,00	0,00	PFCM25*	0,17	0,00	0,00	0,00
PFCM08*	0,00	0,00	0,00	0,00	PFCM26	6,19	0,00	0,65	0,65
PFCM09*	0,53	0,00	0,00	0,00	PFCM27	1,40	0,00	0,00	0,00
PFCM10*	0,00	0,00	0,00	0,00	PFCM28	1,99	0,00	0,00	0,00
PFCM11*	0,00	0,00	0,00	0,00	PFCM29	11,95	0,00	3,13	1,06
PFCM12*	3,22	0,00	0,00	0,00	PFCM30*	0,61	×	0,78	0,07
PFCM13*	0,00	0,00	0,00	0,00	PFCM31*	×	×	0,00	0,00
PFCM14*	0,00	0,00	0,00	0,00	PFCM32	0,00	0,00	0,00	0,00
PFCM15*	0,00	0,00	0,00	0,00	PFCM33	6,31	0,00	0,00	<b>-1,71</b>
PFCM16*	1,34	1,11	2,94	<b>0,00</b>	PFCM34	2,71	0,00	5,73	0,77
PFCM17*	1,51	0,00	0,00	0,00	PFCM35*	0,00	0,00	0,00	0,00
PFCM18	1,44	0,00	0,00	0,00					
					# ótimos	18	23	23	26
					# melhores	19	30	27	31

Para o PFCM-RS, o ILS-RVND encontra a melhor solução em 31 das 35 instâncias; considerando apenas quando um ótimo é conhecido para essa instância, encontramos 26 das 28 instâncias. Em particular, apenas o ILS-RVND foi capaz de encontrar a solução ótima para o PFCM16; além disso, encontrou uma nova solução para o PFCM33.

Agora consideramos o PFCM-PS. Os resultados são apresentados na Tabela 5.3 da mesma maneira que na Tabela 5.2, em que o método ILS-RVND proposto é comparado com as seguintes abordagens da literatura:

SCM-BMCF: *Similarity Coefficients Method with Boltzmann function and Mutation operator for Cell Formation* (Wu et al., 2009)

SA: *Simulated Annealing* (Pailla et al., 2010)

HMGA: *Hybrid Method Genetic Algorithm* (Elbenani et al., 2012)

Para o PFCM-PS, o ILS-RVND encontra a melhor solução em 27 das 35 instâncias; considerando quando um ótimo é conhecido para essa instância, encontramos 26 das 28 instâncias. Para as instâncias PFCM30 e PFCM31, apenas o ILS-RVND encontrou uma solução ótima.

Tabela 5.3: PFCM-PS: comparação entre o ILS-RVND e outras abordagens da literatura.

Instancia	SCM-BMCF	SA	HMGA	ILS-RVND	Instance	SCM-BMCF	SA	HMGA	ILS-RVND
PFCM01*	-	0,00	×	0,00	PFCM19*	0,00	0,00	0,00	0,00
PFCM02*	0,00	0,00	0,00	0,00	PFCM20*	1,23	0,00	1,85	0,00
PFCM03*	1,56	0,00	1,56	0,00	PFCM21*	2,01	0,00	1,38	0,00
PFCM04*	2,84	0,00	2,84	0,00	PFCM22*	0,00	0,00	0,00	0,00
PFCM05*	0,00	1,43	0,00	0,00	PFCM23*	0,00	0,00	0,00	0,00
PFCM06*	0,00	0,00	0,00	0,00	PFCM24*	0,00	0,00	0,00	0,00
PFCM07*	-	0,00	0,00	0,00	PFCM25	0,00	0,00	0,00	0,00
PFCM08*	0,00	0,00	0,00	0,00	PFCM26	0,65	0,78	0,00	0,65
PFCM09*	0,00	0,00	0,00	0,00	PFCM27	0,92	1,25	0,00	0,69
PFCM10*	0,00	0,00	0,00	0,00	PFCM28	-	0,00	0,00	0,49
PFCM11*	0,00	0,00	0,00	0,00	PFCM29	-	0,00	1,30	0,57
PFCM12*	-	0,00	2,94	0,00	PFCM30*	0,71	0,29	×	<b>0,00</b>
PFCM13*	1,41	0,00	1,41	0,00	PFCM31*	0,59	0,18	×	<b>0,00</b>
PFCM14*	-	0,00	0,13	0,00	PFCM32	0,00	0,55	0,00	0,00
PFCM15*	-	0,00	0,56	0,00	PFCM33	-	0,00	0,38	1,15
PFCM16*	2,86	0,14	0,88	0,14	PFCM34	-	0,00	0,87	0,00
PFCM17*	-	0,00	0,00	0,00	PFCM35*	0,00	0,00	0,00	0,00
PFCM18	1.61	0.00	1.18	1.61					
					# ótimos	12	22	14	25
					# melhores	14	28	19	28

## 5.2 PFCM-SOPA

### 5.2.1 Adaptação das Instâncias

O tamanho de uma instância é denotada por  $m \times n$ , em que  $m$  é o número de máquinas e  $n$  é o número de partes, baseado na matriz de incidência de entrada. Na literatura, os autores definem o tamanho do problema com base no tamanho dessa matriz (número de linhas  $\times$  número de colunas). As instâncias de teste são classificadas em três categorias: pequenas, médias e grandes; dependendo do número de máquinas na matriz de incidência. Para mais que 30 máquinas, o problema é considerado como de larga-escala e menos que 12 máquinas como de baixa-escala.

A complexidade do problema também depende da quantidade de dados tratados. Na literatura, diversos tipos de estudos de caso podem ser encontrados:

- matrizes de incidência com a indicação do uso de cada máquina por todas as partes (1 define que a máquina é usada pela parte, 0 caso contrário);
- matrizes de incidência com a indicação da sequência de operações, um identificador  $i$  para  $i$ -ésima operação da parte;
- matrizes de incidência em que o valor 1 é substituído pelo tempo de operação na máquina associada;
- matrizes de incidência com a presença de roteamentos alternativos, em que máquinas de um mesmo tipo são definidas em linhas diferentes.

Para usar e comparar os dados encontrados na literatura, diversas dificuldades foram encontradas:

- as instâncias são pequenas e não contêm dados sobre a alternatividade;
- as instâncias são frequentemente definidas como uma matriz de incidência binária;
- nem sempre as sequências de operações são definidas
- há diversas definições existentes para roteamento e processos alternativos (como discutido em 2.4);
- os casos em larga-escala não são detalhados, exceto em alguns casos apenas com a matriz de incidência;
- casos industriais são complexos e difíceis de obter devido a questões de confidencialidade.

Para este estudo, utilizamos diversos casos de uso da literatura. Dependendo do conjunto de dados, eles foram adaptados de modo a serem usados pelos algoritmos propostos. No caso de quando os dados precisam ser complementados com a sequência de operações, a solução depende significativamente da sequência escolhida. Neste caso, o resultado não pode ser comparado com a melhor solução encontrada na literatura, pois o conjunto de dados foi alterado. Em todos esses casos estudados cada tipo de máquina contém de uma a três máquinas.

Na Tabela 5.4 listamos e apresentamos algumas características das instâncias. Das 27 instâncias utilizadas, 14 foram obtidas diretamente com os autores dos artigos Wu et al. (2009); PP3, PP4 e PP17 em Yin & Yasuda (2002); PP9 em Kazerooni et al. (1997); PPTM1 em Kusiak & Chow (1987) e TM6 em Nagi et al. (1990). Das 7 instâncias restantes, os artigos que as fornecem apenas as definem, mas não realizam testes nessas entradas. Estão nesses casos as instâncias: PP13 em Kazerooni et al. (1997); PP16 em Nagi et al. (1990); TM4, TM7 e PPTM3 em Suresh & Slomp (2001); TM1 em Harhalakis et al. (1990); TM2 em Heragu & Kakuturi (1997) e TM3 em Askin et al. (1997).

Estes problemas testes são organizados em três categorias: casos apenas com informações sobre Planos de Processamento, casos apenas com informações sobre Tipos de Máquinas e casos com informações sobre ambos. Na literatura, apenas três casos foram encontrados com essas informações presentes simultaneamente.

Tabela 5.4: Instâncias do PFCM-SOPA

Instância	# tipos	# máquinas	# partes	# operações	# células	tam. célula	# rotas possíveis
Planos de Processamento							
PP1	4	4	5	21	2	2	$4,8 \times 10^{01}$
PP2	4	4	4	16	2	2	$1,6 \times 10^{01}$
PP3	5	5	7	27	3	3	$1,6 \times 10^{01}$
PP4	6	6	8	44	2	3	$7,2 \times 10^{01}$
PP5	6	6	6	30	3	3	$9,6 \times 10^{01}$
PP6	6	6	10	56	4	2	$5,1 \times 10^{02}$
PP7	7	7	10	57	3	3	$3,1 \times 10^{03}$
PP8	7	7	14	78	3	3	$6,2 \times 10^{04}$
PP9	8	8	13	80	4	3	$8,2 \times 10^{03}$
PP10	10	10	10	75	5	3	$5,2 \times 10^{03}$
PP11	11	11	10	59	3	4	$2,3 \times 10^{03}$
PP12	12	12	20	109	4	3	$4,8 \times 10^{01}$
PP13	15	15	15	190	4	4	$4,1 \times 10^{03}$
PP14	15	15	15	190	5	3	$4,1 \times 10^{03}$
PP15	26	26	28	301	7	5	$7,0 \times 10^{10}$
PP16	30	30	40	344	7	5	$3,3 \times 10^{12}$
PP17	10	10	16	144	7	6	$3,3 \times 10^{12}$
Tipos de Máquinas							
TM1	5	7	6	20	3	3	$2,6 \times 10^{02}$
TM2	10	12	19	79	3	4	$2,7 \times 10^{01}$
TM3	10	17	19	79	4	5	$1,8 \times 10^{19}$
TM4	12	14	20	85	3	5	$2,1 \times 10^{06}$
TM5	16	18	30	154	7	3	$3,5 \times 10^{06}$
TM6	17	20	20	67	5	4	$1,0 \times 10^{07}$
TM7	30	47	10	50	3	16	$1,5 \times 10^{10}$
Planos de Processamento e Tipos de Máquinas							
PPTM1	6	8	12	41	2	3	$1,4 \times 10^{08}$
PPTM2	10	13	15	86	5	3	$7,4 \times 10^{17}$
PPTM3	10	16	16	144	4	5	$2,8 \times 10^{18}$

### 5.2.2 Comparação com a Literatura

Nesta seção, comparamos a abordagem proposta com os resultados da literatura, em especial com os descritos em Wu et al. (2009). Todos esses casos serão apresentados como tendo processos alternativos, descrito pela matriz de entrada em que cada parte contém de um a três roteamentos possíveis. Apresentamos na Tabela 5.5 uma relação dos métodos da literatura do PFCM-SOPA que serão usados como comparativo ao ILS-RVND proposto. É importante salientar que todos os autores apresentados nesta tabela usam diferentes métodos com diferentes critérios: em Yin & Yasuda (2002) minimiza-se o custo de investimentos e permite-se a adição de máquinas duplicadas; Askin et al. (1997) maximiza a flexibilidade das células, etc. Alguns desses casos são descritos através de sequência de operações, volume de produção e capacidade das máquinas como em Kazerooni et al. (1997) ou em Gupta (1993). Nos demais casos em que apenas a matriz

de incidência é utilizada, a fim de ser possível uma comparação, a sequência de operações será dada simplesmente pela ordem das máquinas na matriz.

Tabela 5.5: Relação dos métodos usados na avaliação

Método	Artigos	Ano	Abordagem Empregada
1	Kusiak & Chow (1987)	1987	Algoritmo de clusterização
2	Gupta (1993)	1993	Algoritmo baseado no coeficiente de similaridade
3	Kazerooni et al. (1997)	1997	Algoritmo Genético, minimizando o custo
4	Sofianopoulou (1999)	1999	<i>Simulated Annealing</i>
5	Won (2000)	2000	p-mediana, maximizando similaridade (v1)
6	Won (2000)	2000	p-mediana, maximizando similaridade (v2)
7	Adenso-Diaz et al. (2001)	2001	Busca Tabu
8	Yin & Yasuda (2002)	2002	Heurística em 2 estágios, minimizando o custo
9	Wu et al. (2004)	2004	Busca Tabu
10	Lei & Wu (2005)	2005	Busca Tabu
11	Spiliopoulos & Sofianopoulou (2007)	2007	<i>Bounding Scheme</i>
12	Wu et al. (2009)	2009	<i>Simulated Annealing, minimizando exceções</i>

Devido à grande variedade de métodos e estudos de caso, os resultados não são diretamente comparáveis; portanto, para testar o método proposto, os comparamos com base no número de exceções, um objetivo em comum em todos os métodos. Na Tabela 5.6, temos a relação do número de exceções para cada uma das abordagens relacionadas na Tabela 5.5 e o método proposto. Podemos observar que dentre as 20 instâncias que possuem algum resultado do total de 27, o algoritmo proposto encontrou 7 melhores resultados, 13 empates e nenhum pior resultado. Das 7 instâncias utilizadas que não foram testadas pelas demais abordagens da literatura, e introduzidas aqui no intuito de aumentar o número de casos de teste, obtivemos com sucesso uma solução e apresentamos aqui este resultado.

Dentre o artigos apresentados na Tabela 5.5, apenas Wu et al. (2009) permite uma comparação direta. Isto ocorre, pois é o único trabalho que trata exatamente do mesmo conjunto de restrições e do mesmo objetivo. Na Tabela 5.7 apresentamos os resultados da comparação com o nosso algoritmo, em termos de Eficácia de Agrupamento e tempo computacional gasto (em segundos). Vale ressaltar que das 27 instâncias reunidas da literatura, apenas 14 permitem uma comparação, pois as demais não foram solucionadas em Wu et al. (2009). Destas 14 instâncias, obteve-se 5 melhores resultados e empate nos demais.



Tabela 5.7: Tabela de comparação com o método de Wu et al. (2009)

Instância	Wu et al. (2009)		ILS-RVND	
	$\mu$	Tempo (s)	$\mu$	Tempo (s)
PP1	90	0,003	90	0,040
PP2	100	0,013	100	0,023
PP3	-	-	93,33	0,050
PP4	-	-	66,67	0,081
PP5	83,33	0,031	83,33	0,047
PP6	72,22	0,05	72,22	0,174
PP7	81,48	0,034	81,48	0,098
PP8	69,44	0,048	<b>70,27</b>	0,143
PP9	-	-	73,81	0,155
PP10	82,86	0,102	<b>83,33</b>	0,142
PP11	80,65	0,105	80,65	0,119
PP12	49,47	0,216	49,47	0,383
PP13	-	-	49,15	0,644
PP14	79,52	0,528	79,52	0,658
PP15	72,48	0,569	72,48	2,430
PP16	-	-	55,11	5,652
PP17	-	-	83,84	9,571
TM1	-	-	72,73	0,033
TM2	-	-	53,47	0,241
TM3	-	-	73,33	0,406
TM4	54,29	0,313	<b>54,85</b>	0,233
TM5	47,45	0,506	47,45	0,529
TM6	-	-	60,40	0,228
TM7	-	-	39,47	0,338
PPTM1	-	-	63,83	0,077
PPTM2	61,9	0,134	<b>64,52</b>	0,170
PPTM3	-	-	73,81	0,499

# Capítulo 6

## Considerações Finais

Neste trabalho, apresentou-se primeiramente um algoritmo baseado na meta-heurística ILS para a resolução do PFCM, projetado para permitir a inclusão de novos fatores de produção e tornar o problema mais realista. A exploração do espaço de soluções é realizada por três novas estruturas de vizinhança inseridas num módulo RVND e três algoritmos de perturbação. Para a validação desse algoritmo, um conjunto de 35 instâncias foi utilizado, e os resultados obtidos foram comparados com as abordagens da literatura. Os resultados demonstram o bom desempenho e a robustez da abordagem empregada em comparação com os resultados da literatura, encontrando soluções em um tempo computacional baixo e com boa qualidade. Com destaque para quatro soluções com melhor qualidade que qualquer outra na literatura (duas para o PFCM-RS e duas pro PFCM-PS), em especial pelo fato desse ser um problema amplamente explorado na literatura, mas sem haver um algoritmo exato que tenha sido capaz de encontrar uma solução ótima para todos os casos. Esta pesquisa resultou na publicação do artigo Martins et al. (2015) e contribuiu em co-autoria para Pinheiro et al. (2016).

No intuito de validar a flexibilidade do algoritmo quanto à inclusão de novos fatores de produção, consideramos um importante subconjunto de fatores de produção: a presença de **planos de processamentos alternativos** para operação das partes por meio de **sequências de operações**. Para tal, uma extensão deste algoritmo foi desenvolvida para o PCFM-SOPA, adaptando-se a representação da entrada do problema na geração de soluções iniciais pelo método construtivo, além de adicionar novas estruturas de vizinhança e perturbações que percorram este novo espaço de soluções. Experimentos computacionais foram realizados em 27 instâncias e comparados com os resultados obtidos por 12 algoritmos da literatura, apresentando sempre resultados iguais ou superiores. Desta etapa, temos um artigo em andamento para publicação em periódico.

Em termos de potencial para trabalhos futuros, este estudo possui diversos desdobramentos possíveis. Num primeiro momento, algumas melhorias podem ser realizadas no algoritmo através da inclusão de módulos exatos eficientes em duas situações distin-

tas: a primeira como aplicado em Spiliopoulos & Sofianopoulou (2007), com o objetivo de reduzir o espaço de busca ao fixar algumas variáveis da solução; ou como aplicado em Pinheiro et al. (2016), através de um método de refino de soluções baseado no *Set Partitioning*, tendo como entrada um conjunto de soluções intermediárias válidas geradas pelo algoritmo, e retornando uma solução formada pela melhor combinação do conjunto de entrada. Outra direção de pesquisa considera a inclusão de mais fatores práticos de produção em conjunto com os já apresentados. Fatores como capacidade das células, capacidade das máquinas, demanda das partes, divisão do fluxo de produção e restrições temporais de processamento, podem ser naturalmente acopláveis.

# Bibliografia

- Adenso-Diaz, B., Lozano, S., Racero, J. & Guerrero, F. (2001), 'Machine cell formation in generalized group technology', *Computers & Industrial Engineering* **41**(2), 227–240.
- Adil, G. K., Rajamani, D. & Strong, D. (1993), 'A mathematical model for cell formation considering investment and operational cost', *European Journal of Operational Research* **69**, 330–341.
- Adil, G. K., Rajamani, D. & Strong, D. (1996), 'Cell formation considering alternate routings', *International Journal of Production Research* **34**, 1361–1380.
- Askin, R. G. & Subramaniam, S. B. (1987), 'A cost-based heuristic for group technology configuration', *International Journal of Production Research* **25**(1), 101–113.
- Askin, R. G., Cresswell, S. H., Goldberg, J. B. & Vakharia, A. J. (1991), 'A hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing', *International Journal of Production Research* **29**(6), 1081–1100.
- Askin, R. G., Selim, H. M. & Vakharia, A. J. (1997), 'A methodology for designing flexible cellular manufacturing systems', *IIE Transactions* **29**(7), 599–610.
- Askin, R. J. & Vakharia, A. J. (1990), *Automated Factory Handbook: Technology and Management*, TAB Books, Inc., chapter Group technology - cell formation and operation, pp. 317–366.
- Baykasoglu, A. & Gindy, N. N. A. (2000), 'Mocacef 1.0: Multiple objective capability based approach to form part-machine groups for cellular manufacturing application', *International Journal of Production Research* **38**(5), 1133–1161.
- Baykasoglu, A., Gindy, N. & Cobb, R. (2001), 'Capability based formulation and solution of multiple objective cell formation problems using simulated annealing.', *Integrated Manufacturing System* **12**, 258–274.
- Benlic, U. & Hao, J. (2013), Hybrid metaheuristics for the graph partitioning problem, in E. Talbi, ed., 'Hybrid Metaheuristics', Vol. 434 of *Studies in Computational Intelligence*, Springer, pp. 157–185.

- Burbidge, J. L. (1971), 'Production flow analysis', *Production Engineer* **50**(4/5), 139–152.
- Chan, H. M. & Milner, D. A. (1982), 'Direct clustering algorithm for group formation in cellular manufacture', *Journal of Manufacturing Systems* **1**, 65–75.
- Chandrasekharan, M. P. & Rajagopalan, R. (1986a), 'An ideal seed non-hierarchical clustering algorithm for cellular manufacturing', *International Journal of Production Research* **24**(2), 451–464.
- Chandrasekharan, M. P. & Rajagopalan, R. (1986b), 'Modroc: An extension of rank order clustering for group technology', *International Journal of Production Research* **24**(5), 1221–1233.
- Chandrasekharan, M. P. & Rajagopalan, R. (1987), 'Zodiac - an algorithm for concurrent formation of part-families and machine-cells', *International Journal of Production Research* **25**(6), 835–850.
- Chu, C. H. (1989), 'Cluster analysis in manufacturing cellular formation', *Omega* **17**(3), 289–295.
- Dahel, N. E. & Smith, S. B. (1993), 'Designing flexibility into cellular manufacturing systems', *International Journal of Production Research* **31**, 933–945.
- Defersha, F. M. & Chen, M. (2006), 'A comprehensive mathematical model for the design of cellular manufacturing systems', *International Journal of Production Economics* **103**(2), 767–783.
- Diallo, M., Pierreval, H. & Quilliot, A. (2001), 'Manufacturing cells design with flexible routing capability in presence of unreliable machines', *International Journal of Production Economics* **74**(1), 175–182.
- Elbenani, B., Ferland, J. A. & Bellemare, J. (2012), 'Genetic algorithm and large neighbourhood search to solve the cell formation problem', *Expert Systems with Applications* **39**(3), 2408–2414.
- Flanders, R. E. (1924), 'Design, manufacture and production control of a standard machine', *Transactions of the American Society of Mechanical Engineers* **46**, 691–738.
- Glover, F. (1989), 'Tabu search - part i', *ORSA Journal on Computing* **1**(3), 190–206.
- Glover, F. (1990), 'Tabu search - part ii', *ORSA Journal on Computing* **2**(1), 4–32.
- Gonçalves, J. F. & Resende, M. G. C. (2004), 'An evolutionary algorithm for manufacturing cell formation', *Computers & Industrial Engineering* **47**(2-3), 247–273.

- Gravel, M., Nsakanda, A. L. & Price, W. (1998), 'Efficient solutions to the cell-formation problem with multiple routings via a double-loop genetic algorithm', *European Journal of Operational Research* **109**, 286–298.
- Grefenstette, J. (1985), Genetic algorithms and their applications, in 'Proceedings of International Conference on Genetic Algorithms and Their Applications', Lawrence Erlbaum Associates.
- Gupta, T. (1993), 'Design of manufacturing cells for flexible environment considering alternative routing', *International Journal of Production Research* **31**, 1259–1273.
- Harhalakis, G., Hilger, J. & Proth, J. M. (1990), Manufacturing cell formation in the case of multi-manufacturing processes, Technical report, University of Maryland.
- Heragu, S. S. & Kakuturi, S. R. (1997), 'Grouping and placement of machines cells', *IIE Transactions* **29**, 561–571.
- James, T. L., Brown, E. C. & Keeling, K. B. (2007), 'A hybrid grouping genetic algorithm for the cell formation problem', *Comput. Oper. Res.* **34**(7), 2059–2079.
- Jarvis, R. A. & Patrick, E. A. (1973), 'Clustering using a similarity measure based on shared nearest neighbors', *IEEE Transactions on Computers* **C-22**(11), 1025–1034.
- Joines, J. A., Kay, M. G. & King, R. E. (1997), A hybrid genetic algorithm for manufacturing cell design, Technical report, Department of Industrial Engineering, North Carolina State University.
- Kang, S. & Wemmerlov, E. (1993), 'A work load-oriented heuristic methodology for manufacturing cell formation allowing reallocation of operations', *European Journal of Operational Research* **69**(3), 292–311.
- Kazerooni, M., Luong, H. S. & Abhary, K. (1997), 'A genetic algorithm based cell design considering alternative routing', *Computer-Integrated Manufacturing Systems* **10**(2), 93–108.
- Kim, Y. (1993), 'A study on surrogate objectives for loading a certain type of flexible manufacturing system', *International Journal of Production Research* **31**, 381–392.
- King, J. R. (1980), 'Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm', *International Journal of Production Research* **18**(2), 213–232.
- King, J. R. & Nakornchai, V. (1982), 'Machine-component group formation in group technology: review and extension', *International Journal of Production Research* **20**(2), 117–133.

- Kumar, C. S. & Chandrasekharan, M. P. (1990), 'Group efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology', *International Journal of Production Research* **28**(2), 233–243.
- Kumar, K. R. & Vannelli, A. (1987), 'Strategic subcontracting for efficient disaggregated manufacturing', *International Journal of Production Research* **25**(12), 1715–1728.
- Kusiak, A. (1987), 'The generalized group technology concept', *International Journal of Production Research* **25**, 561–569.
- Kusiak, A. & Cheng, C. H. (1990), 'A branch-and-bound algorithm for solving the group technology problem', *Annals of Operations Research* **26**, 415–431.
- Kusiak, A. & Chow, W. S. (1987), 'Efficient solving of the group technology problem', *Journal of Manufacturing Systems* **6**(2), 117–124.
- Lei, D. & Wu, Z. (2005), 'Tabu search approach based on a similarity coefficient for cell formation in generalized group technology', *International Journal of Production Research* **43**, 4035–4047.
- Lloyd, S. (1982), 'Least squares quantization in PCM', *IEEE Transactions on Information Theory* **28**(2), 129–137.
- Logendran, R. (1990), 'A workload based model for minimizing total intercell and intracell moves in cellular manufacturing', *International Journal of Production Research* **28**(5), 913–925.
- Logendran, R., Ramakrishna, P. & Srikantharajah, C. (1994), 'Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans', *European Journal of Operational Research* **32**(2), 273–297.
- Lourenço, H. R., Martin, O. C. & Stützle, T. E. (2002), *Handbook of Metaheuristics*, Kluwer Academic Publishers, chapter Iterated Local Search, pp. 321–353.
- Mansouri, S. A., Moattar-Husseini, S. M. & Newman, S. T. (2000), 'A review of the modern approaches to multi-criteria cell design', *International Journal of Production Research* **38**(5), 1201–1218.
- Martins, I. C., Pinheiro, R. G. S., Protti, F. & Ochi, L. S. (2015), 'A hybrid iterated local search and variable neighborhood descent heuristic applied to the cell formation problem', *Expert Systems with Applications* **42**(22), 8947–8955.
- McCormick, W. T., Schweitzer, P. J. & White, T. W. (1972), 'Problem decomposition and data reorganization by a clustering technique', *Operations Research* **20**(5), 993–1009.

- Mitrofanov, S. P. (1966), *The Scientific Principles of Group Technology*, National Lending Library for Science and Technology.
- Mladenovic, N. & Hansen, P. (1997), 'Variable neighborhood search', *Computers and Operations Research* **24**(11), 1097–1100.
- Mosier, C. T. & Taube, L. (1985), 'The facets of group technology and their impacts on implementation – a state-of-the-art survey', *The International Journal of Management Science*.
- Nagi, R., Harhalakis, G. & Proth, J. (1990), 'Multiple routings and capacity considerations in group technology applications', *European Journal of Operational Research* **28**(12), 2243–2257.
- Pailla, A., Trindade, A. R., Parada, V. & Ochi, L. S. (2010), 'A numerical comparison between simulated annealing and evolutionary approaches to the cell formation problem', *Expert Systems with Applications* **37**(7), 5476–5483.
- Papaiouannou, G. & Wilson, J. M. (2010), '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**, 509–521.
- Penna, P. H. V., Subramanian, A. & Ochi, L. S. (2013), 'An iterated local search heuristic for the heterogeneous fleet vehicle routing problem', *Journal of Heuristics* **19**(2), 201–232.
- Pinheiro, R. G. S., Martins, I. C., Protti, F., Ochi, L. S., Simonetti, L. G. & Subramanian, A. (2016), 'On solving manufacturing cell formation via bicluster editing', *European Journal of Operational Research* **254**(3), 769–779.
- Rajagopalan, R. & Batra, J. L. (1975), 'Design of cellular production systems: a graph-theoretic approach', *International Journal of Production Research* **13**(6), 567–579.
- Ramabhatta, V. & Nagi, R. (1998), 'An integrated formulation of manufacturing cell formation with capacity planning and multiple routings', *Annals of Operations Research* **77**, 79–95.
- Sankaran, S. & Kasilingam, G. (1990), 'An integrated approach to cell formation and part routing in group technology manufacturing systems', *Engineering Optimization* **16**, 235–245.
- Sarin, S. & Dar-El, E. (1986), 'Scheduling parts in an fms', *Large Scale Systems* **11**, 83–94.
- Sarker, B. & Khan, M. (2001), 'A comparison of existing grouping efficiency measures and a new weighted grouping efficiency measure', *IIE Transactions* **33**, 11–27.

- Sarker, B. & Xu, Y. (1998), 'Operation sequences-based cell formation methods: a critical survey', *Production Planning & Control* **9**(8), 771–783.
- Selim, H., Askin, R. & Vakharia, A. (1995), Flexibility in cellular manufacturing: A framework and measures, Technical report, DIS Department, University of Florida.
- Selim, H., Askin, R. & Vakharia, A. (1998), 'Cell formation in group technology: Review, evaluation and directions for future research', *Computers & Industrial Engineering* **34**(1), 3–20.
- Shafer, S. M. (1998), *Group Technology and Cellular Manufacturing: A State of the Art Synthesis of Research and Practice*, Kluwer Academic Publishers, chapter Part-machine labour grouping: The problem and solution methods, pp. 131–152.
- Shafer, S. M. & Rogers, D. F. (1991), 'A goal programming approach to cell formation problems', *Journal of Operations Management* **10**, 28–43.
- Silva, G. C., Bahiense, L., Ochi, L. S. & Boaventura, P. O. (2012), 'The dynamic space allocation problem: applying a hybrid grasp and tabu search metaheuristics', *Computers & Operations Research* **39**, 671–677.
- Singh, N. (1993), 'Design of cellular manufacturing systems: an invited review', *European Journal of Operational Research* **69**(3), 284–291.
- Sofianopoulou, S. (1999), 'Manufacturing cells design with alternative process plans and/or replicate machines', *International Journal of Production Research* **37**(3), 707–720.
- Spiliopoulos, K. & Sofianopoulou, S. (2007), 'Manufacturing cell design with alternative routings in generalized group technology: Reducing the complexity of the solution space', *International Journal of Production Research* **45**(6), 1355–1367.
- Srinivasan, G. & Narendran, T. T. (1991), 'Grafics - a nonhierarchical clustering-algorithm for group technology', *International Journal of Production Research* **29**(3), 463–478.
- Stawowy, A. (2006), 'Evolutionary strategy for manufacturing cell design', *The International Journal of Management Science* **34**(1), 1–18.
- Subramanian, A., Uchoa, E. & Ochi, L. S. (2013), 'A hybrid algorithm for a class of vehicle routing problem', *Computers & Operations Research* **40**, 2519–2531.
- Suresh, N. C. & Slomp, J. (2001), 'A multi-objective procedure for labor assignments and grouping in capacitated cell formation problems', *International Journal of Production Research* **39**(18), 4103–4131.

- Uddin, M. K. & Shanker, K. (2002), 'Grouping of parts and machines in presence of alternative process routes by genetic algorithm', *International Journal of Production Economics* **76**(3), 219–228.
- Vakharia, A. J. & Wemmerlov, U. (1990), 'Designing a cellular manufacturing system: a materials flow based operations sequence', *IIE Transactions* **22**(1), 84–97.
- Vannelli, A. & Kumar, K. R. (1986), 'A method for finding minimal bottle-neck cells for grouping part-machine families', *International Journal of Production Research* **24**(2), 387–400.
- Vivekanand, P. & Narendran, T. (1998), 'Logical cell formation in fms, using flexibility-base criteria', *International Journal of Flexible Manufacturing Systems* **10**, 163–181.
- Wei, J. C. & Gaither, N. (1990), 'A capacity constrained multiobjective cell formation method', *Journal of Manufacturing Systems* **9**(3), 222–232.
- Won, Y. (2000), 'New p-median approach to cell formation with alternative process plans', *International Journal of Production Research* **38**(1), 229–240.
- Won, Y. K. & Kim, S. H. (1997), 'ultiple criteria clustering algorithm for solving the group technology problem with multiple process routings', *Industrial Engineering* **32**(1), 207–220.
- Wu, T. H., Chang, C. C. & Chung, S. H. (2008), 'A simulated annealing algorithm for manufacturing cell formation problems', *Expert Systems with Applications* **34**(3), 1609–1617.
- Wu, T. H., Chang, C. C. & Yeh, J. Y. (2009), 'A hybrid heuristic algorithm adopting both boltzmann function and mutation operator for manufacturing cell formation problems', *International Journal of Production Economics* **120**(2), 669–688.
- Wu, T. H., Chung, S. H. & Chang, C. C. (2010), 'A water flow-like algorithm for manufacturing cell formation problems', *European Journal of Operational Research* **205**(2), 346–360.
- Wu, T. H., Low, C. & Wu, W. T. (2004), 'A tabu search approach to the cell formation problem', *The International Journal of Advanced Manufacturing Technology* **23**(11), 916–924.
- Yasuda, K. & Yin, Y. (2001), 'A dissimilarity measure for solving the cell formation problem in cell manufacturing', *Computers & Industrial Engineering* **39**, 1–17.
- Yin, Y. & Yasuda, K. (2002), 'Manufacturing cells design in consideration of various production factors', *International Journal of Production Research* **40**(4), 885–906.

- 
- Yu, C. T. & V.V., R. (1977), 'A single pass method for determining the relationship between terms', *Journal of The American Society for Informations Sciences* **28**, 345–354.
- Zhao, C. & Wu, Z. (2000), 'A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives', *International Journal of Production Research* **38**(2), 385–395.