# Group Technology and Facility Layout

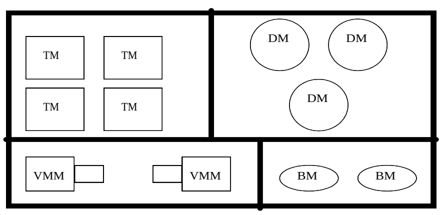
Chapter 6

## Benefits of GT and Cellular Manufacturing (CM)

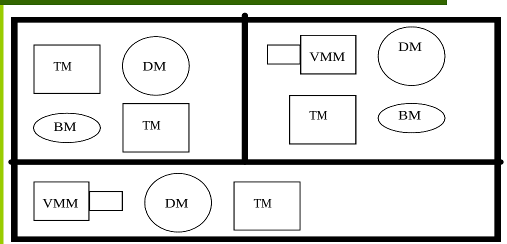
REDUCTIONS
- Setup time
- Inventory
- Material handling cost
- Direct and indirect labor cost

IMPROVEMENTS
- Quality
- Material Flow
- Machine and operator Utilization
- Space Utilization
- Employee Morale

## Process layout




## Group technology layout




## Sample part-machine processing indicator matrix

$[a_{ij}] =$ (Part × Machine)

| Part | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | | | 1 | | 1 | |
| $P_2$ | | 1 | 1 | | 1 | | |
| $P_3$ | | | | 1 | | 1 | |
| $P_4$ | | 1 | 1 | | | | |
| $P_5$ | | | 1 | | | | 1 |
| $P_6$ | | 1 | | | 1 | | 1 |

## Rearranged part-machine processing indicator matrix

$[a_{ij}] =$ (Part × Machine)

| Part | $M_1$ | $M_4$ | $M_6$ | $M_2$ | $M_3$ | $M_5$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | |
| $P_3$ | | 1 | 1 | | | | |
| $P_2$ | | | | 1 | 1 | 1 | |
| $P_4$ | | | | 1 | 1 | | |
| $P_5$ | | | | | 1 | | 1 |
| $P_6$ | | | | 1 | | 1 | 1 |

## Rearranged part-machine processing indicator matrix

$[a_{ij}] =$ (Part)

| | $M_1$ | $M_4$ | $M_6$ | $M_2$ | $M_3$ | $M_5$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | |
| $P_3$ | | 1 | 1 | | | | |
| $P_2$ | 1 | | | 1 | 1 | 1 | |
| $P_4$ | | | | 1 | 1 | | |
| $P_5$ | | | | | | 1 | 1 |
| $P_6$ | | | | 1 | | 1 | 1 |

## Rearranged part-machine processing indicator matrix

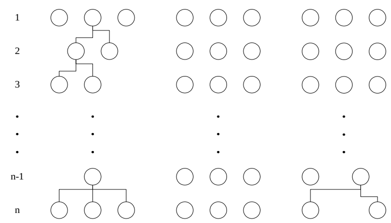
$[a_{ij}] =$ (Part)

| | Machine | | | | | | |
|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_4$ | $M_6$ | $M_2$ | $M_3$ | $M_5$ | $M_7$ |
| $P_1$ | 2 | 3 | 1 | | | | |
| $P_3$ | | 1 | 2 | | | | |
| $P_2$ | 3 | | | 1 | 4 | 2 | |
| $P_4$ | | | | 2 | 1 | | |
| $P_5$ | | | | | | 1 | 2 |
| $P_6$ | | | | 1 | | 2 | 3 |

## Classification and Coding Schemes

- Hierarchical
- Non-hierarchical
- Hybrid

## Classification and Coding Schemes




## Classification and Coding Schemes

| Name of system | Country Developed | Characteristics |
|---|---|---|
| TOYODA | Japan | Ten digt code |
| MICLASS | The Netherlands | Thirty digit code |
| TEKLA | Norway | Twelve digit code |
| BRISCH | United Kingdom | Based on four to six digit primary code and a number of secondary digits |
| DCLASS | USA | Software-based system without any fixed code structure |
| NITMASH | USSR | A hierarchical code of ten to fifteen digits and a serial number |
| OPITZ | West Germany | Based on a five digit primary code with a four digit secondary code |

## MICLASS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 . . . 29 30

## Advantages of Classification and Coding Systems

- Maximize design efficiency
- Maximize process planning efficiency
- Simplify scheduling

## Clustering Approach

- Rank order clustering
- Bond energy
- Row and column masking
- Similarity coefficient
- Mathematical Programming

## Rank Order Clustering Algorithm

**Step 1**: Assign binary weight $BW_j = 2^{m-j}$ to each column *j* of the part-machine processing indicator matrix.

**Step 2**: Determine the decimal equivalent *DE* of the binary value of each row *i* using the formula

$$DE_i = \sum_{j=1}^{m} 2^{m-j} a_{ij}$$

**Step 3**: Rank the rows in decreasing order of their *DE* values. Break ties arbitrarily. Rearrange the rows based on this ranking. If no rearrangement is necessary, stop; otherwise go to step 4.

## Rank Order Clustering Algorithm

**Step 4**: For each rearranged row of the matrix, assign binary weight $BW_i = 2^{n-i}$.

**Step 5**: Determine the decimal equivalent of the binary value of each column *j* using the formula

$$DE_j = \sum_{i=1}^{m} 2^{n-i} a_{ij}$$

**Step 6**: Rank the columns in decreasing order of their *DE* values. Break ties arbitrarily. Rearrange the columns based on this ranking. If no rearrangement is necessary, stop; otherwise go to step 1.

## Rank Order Clustering – Example 1

$[a_{ij}] =$

| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | |
|---|---|---|---|---|---|---|---|---|
| Binary weight | 64 | 32 | 16 | 8 | 4 | 2 | 1 | Binary value |
| $P_1$ | 1 | | | 1 | | 1 | | 74 |
| $P_2$ | | 1 | 1 | | 1 | | | 52 |
| $P_3$ | | | | 1 | | 1 | | 10 |
| $P_4$ | | 1 | 1 | | | | | 48 |
| $P_5$ | | | 1 | | | | 1 | 17 |
| $P_6$ | | 1 | | | 1 | | 1 | 37 |

## Rank Order Clustering – Example 1

$[a_{ij}] =$

| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ | |
|---|---|---|---|---|---|---|---|---|
| Binary value | 32 | 28 | 26 | 33 | 20 | 33 | 6 | Binary weight |
| $P_1$ | 1 | | | 1 | | 1 | | 32 |
| $P_2$ | | 1 | 1 | | 1 | | | 16 |
| $P_4$ | | 1 | 1 | | | | | 8 |
| $P_6$ | | 1 | | | 1 | | 1 | 4 |
| $P_5$ | | | 1 | | | | 1 | 2 |
| $P_3$ | | | | 1 | | 1 | | 1 |

## Rank Order Clustering – Example 1

$[a_{ij}] =$

| Binary weight | $M_4$ 64 | $M_6$ 32 | $M_1$ 16 | $M_2$ 8 | $M_3$ 4 | $M_5$ 2 | $M_7$ 1 | Binary value |
|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | | 112 |
| $P_2$ | | | | 1 | 1 | 1 | | 14 |
| $P_4$ | | | | 1 | 1 | | | 12 |
| $P_6$ | | | | 1 | | 1 | 1 | 11 |
| $P_5$ | | | | | 1 | | 1 | 5 |
| $P_3$ | 1 | 1 | | | | | | 96 |

## Rank Order Clustering – Example 1

$[a_{ij}] =$

| Binary value | $M_4$ 48 | $M_6$ 48 | $M_1$ 32 | $M_2$ 14 | $M_3$ 12 | $M_5$ 10 | $M_7$ 3 | Binary weight |
|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | | 32 |
| $P_3$ | 1 | 1 | | | | | | 16 |
| $P_2$ | | | | 1 | 1 | 1 | | 8 |
| $P_4$ | | | | 1 | 1 | | | 4 |
| $P_6$ | | | | 1 | | 1 | 1 | 2 |
| $P_5$ | | | | | | | 1 | 1 |

## ROC Algorithm Solution – Example 1

$[a_{ij}] =$

| Binary value | $M_4$ 48 | $M_6$ 48 | $M_1$ 32 | $M_2$ 14 | $M_3$ 12 | $M_5$ 10 | $M_7$ 3 | Binary weight |
|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | | 32 |
| $P_3$ | 1 | 1 | | | | | | 16 |
| $P_2$ | | | | 1 | 1 | 1 | | 8 |
| $P_4$ | | | | 1 | 1 | | | 4 |
| $P_6$ | | | | 1 | | 1 | 1 | 2 |
| $P_5$ | | | | | | | 1 | 1 |

## Bond Energy Algorithm

**Step 1**: Set $i$=1. Arbitrarily select any row and place it.

**Step 2**: Place each of the remaining $n$-$i$ rows in each of the $i$+1 positions (i.e. above and below the previously placed $i$ rows) and determine the row bond energy for each placement using the formula

$$\sum_{i=1}^{i+1}\sum_{j=1}^{m} a_{ij}\left(a_{i-1,j}+a_{i+1},j\right)$$

Select the row that increases the bond energy the most and place it in the corresponding position.

## Bond Energy Algorithm

**Step 3**: Set $i$=$i$+1. If $i$ < $n$, go to step 2; otherwise go to step 4.

**Step 4**: Set $j$=1. Arbitrarily select any column and place it.

**Step 5**: Place each of the remaining $m$-$j$ rows in each of the $j$+1 positions (i.e. to the left and right of the previously placed $j$ columns) and determine the column bond energy for each placement using the formula

$$\sum_{i=1}^{n}\sum_{j=1}^{j+1} a_{ij}\left(a_{i,j-1}+a_{i,j+1}\right)$$

**Step 6**: Set $j$=$j$+1. If $j$ < $m$, go to step 5; otherwise stop.

## BEA – Example 2

| Row \ Column | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 |
| 4 | 1 | 0 | 1 | 0 |

## BEA – Example 2

| Row Selected | Where Placed | Row Arrangement | Row Bond Energy | Maximize Energy |
|---|---|---|---|---|
| 1 | Above Row 2 | 1 0 1 0<br>0 1 0 1 | 0 | No |
| 1 | Below Row 2 | 0 1 0 1<br>1 0 1 0 | 0 | No |
| 3 | Above Row 2 | 0 1 0 1<br>0 1 0 1 | 4 | Yes |
| 3 | Below Row 2 | 0 1 0 1<br>0 1 0 1 | 4 | Yes |
| 4 | Above Row 2 | 1 0 1 0<br>0 1 0 1 | 0 | No |
| 4 | Below Row 2 | 0 1 0 1<br>1 0 1 0 | 0 | No |

## BEA – Example 2

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |

## BEA – Example 2

| Column Selected | Where Placed | Column Arrangement | Column Bond Energy | Maximize Energy |
|---|---|---|---|---|
| 2 | Left of Column 1 | 0 1<br>0 1<br>1 0<br>1 0 | 0 | No |
| 2 | Right of Column 1 | 1 0<br>1 0<br>0 1<br>0 1 | 0 | No |
| 3 | Left of Column 1 | 1 1<br>1 1<br>0 0<br>0 0 | 4 | Yes |
| 3 | Right of Column 1 | 1 1<br>1 1<br>0 0<br>0 0 | 4 | Yes |
| 4 | Left of Column 1 | 0 1<br>0 1<br>1 0<br>1 0 | 0 | No |
| 4 | Right of Column 1 | 1 0<br>1 0<br>0 1<br>0 1 | 0 | No |

## BEA Solution – Example 2

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |

## Row and Column Masking Algorithm

**Step 1**: Draw a horizontal line through the first row. Select any 1 entry in the matrix through which there is only one line.

**Step 2**: If the entry has a horizontal line, go to step 2a. If the entry has a vertical line, go to step 2b.

**Step 2a**: Draw a vertical line through the column in which this 1 entry appears. Go to step 3.

**Step 2b**: Draw a horizontal line through the row in which this 1 entry appears. Go to step 3.

**Step 3**:If there are any 1 entries with only one line through them, select any one and go to step 2. Repeat until there are no such entries left. Identify the corresponding machine cell and part family. Go to step 4.

**Step 4**: Select any row through which there is no line. If there are no such rows, STOP. Otherwise draw a horizontal line through this row, select any 1 entry in the matrix through which there is only one line and go to Step 2.

## R&CM Algorithm – Example 3

Machine

$[a_{ij}] =$ Part

| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | | | | | | |
| $P_2$ | | 1 | 1 | | 1 | | |
| $P_3$ | | | | | | | |
| $P_4$ | | 1 | 1 | | | | |
| $P_5$ | | | 1 | | | | 1 |
| $P_6$ | | 1 | | | 1 | | 1 |

## R&CM Algorithm – Example 3

Machine

| $[a_{ij}] =$ Part | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | | | 1 | | 1 | |
| $P_2$ | | | 1 | | 1 | | |
| $P_3$ | | | | 1 | | 1 | |
| $P_4$ | | | 1 | | | | |
| $P_5$ | | | 1 | | | | 1 |
| $P_6$ | | | | | 1 | | 1 |

## R&CM Algorithm - Solution

Machine

| $[a_{ij}] =$ Part | $M_1$ | $M_4$ | $M_6$ | $M_2$ | $M_3$ | $M_5$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 1 | 1 | | | | |
| $P_3$ | | 1 | 1 | | | | |
| $P_2$ | | | | 1 | 1 | 1 | |
| $P_4$ | | | | 1 | 1 | | |
| $P_5$ | | | | | 1 | | 1 |
| $P_6$ | | | | 1 | | 1 | 1 |

## Similarity Coefficient (SC) Algorithm

$$s_{ij} = \frac{\sum_{k=1}^{n} a_{ki} a_{kj}}{\sum_{k=1}^{n} \left( a_{ki} + a_{kj} - a_{ki} a_{kj} \right)},$$

$$\text{where } a_{ki} = \begin{cases} 1 \text{ if part } k \text{ requires machine } i \\ 0 \text{ otherwise} \end{cases}$$

## SC Algorithm – Example 4

Machine

| $[a_{ij}] =$ Part | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ | $M_7$ |
|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | | | 1 | | 1 | |
| $P_2$ | | 1 | 1 | | 1 | | |
| $P_3$ | | | | 1 | | 1 | |
| $P_4$ | | 1 | 1 | | | | |
| $P_5$ | | | 1 | | | | 1 |
| $P_6$ | | 1 | | | 1 | | 1 |

## SC Algorithm – Example 4

| Machine Pair | SC Value | Combine into one cell? |
|---|---|---|
| {1,2} | 0/4=0 | No |
| {1,3} | 0/4=0 | No |
| {1,4} | 1/2 | No |
| {1,5} | 0/3=0 | No |
| {1,6} | 1/2 | No |
| {1,7} | 0/3=0 | No |
| {2,3} | 1/2 | No |
| {2,4} | 0/5=0 | No |
| {2,5} | 2/3 | Yes |
| {2,6} | 0/5=0 | No |
| {2,7} | 1/4 | No |
| {3,4} | 0/5=0 | No |
| {3,5} | 1/4 | No |
| {3,6} | 0/5=0 | No |
| {3,7} | 1/4 | No |
| {4,5} | 0/4=0 | No |
| {4,6} | 2/2=1 | Yes |
| {4,7} | 0/4=0 | No |
| {5,6} | 0/4=0 | No |
| {5,7} | 1/3 | No |
| {6,7} | 0/4=0 | No |

## SC Algorithm – Example 4

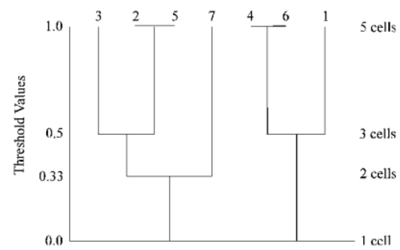
| Machine/Cell Pair | SC Value | Combine into one cell? |
|---|---|---|
| {1, (2,5)} | 0 | No |
| {1, (4,6)} | 1/2 | Yes |
| {1,3} | 0 | No |
| {1,7} | 0 | No |
| {(2,5), (4,6)} | 0 | No |
| {(2,5), 3} | 1/2 | Yes |
| {(2,5), 7} | 1/3 | No |
| {(4,6), 3} | 0 | No |
| {(4,6), 7} | 0 | No |
| {3,7} | 1/4 | No |

## SC Algorithm – Example 4

| Machine/Cell Pair | SC Value | Combine into one cell? |
|---|---|---|
| {(1,4,6) (2,3,5)} | 0 | No |
| {(1,4,6), 7} | 0 | No |
| {(2,3,5), 7} | 1/3 | Yes |

| Machine/Cell Pair | SC Value | Combine into one cell? |
|---|---|---|
| {(1,4,6) (2,3,5,7)} | 0 | No |

## SC Algorithm Solution – Example 4




## Mathematical Programming Approach

$$s_{ij} = \frac{\sum_{k=1}^{m} a_{ik} a_{jk}}{\sum_{k=1}^{m} \left( a_{ik} + a_{jk} - a_{ik} a_{jk} \right)},$$

$$\text{where } a_{ik} = \begin{cases} 1 \text{ if part } i \text{ requires machine } k \\ 0 \text{ otherwise} \end{cases}$$

## Weighted Minkowski metric

$$d_{ij} = \left[ \sum_{k=1}^{n} w_k \left| a_{ki} - a_{kj} \right|^r \right]^{1/r}$$

- $r$ is a positive integer
- $w_k$ is the weight for part $k$
- $d_{ij}$ instead of $s_{ij}$ to indicate that this is a dissimilarity coefficient
- Special case where $w_k$=1, for $k$=1,2,...,$n$, is called the Minkowski metric
- Easy to see that for the Minkowski metric, when $r$=1, above equation yields an absolute Minkowski metric, and when $r$=2, it yields the Euclidean metric
- The absolute Minkowski metric measures the dissimilarity between part pairs

## *P*-Median Model

- Minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}$
- Subject to $\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,...,n$

$$\sum_{j=1}^{n} x_{jj} = P$$

$$x_{ij} \le x_{jj} \qquad i, j = 1,2,...,n$$

$$x_{ij} = 0 \text{ or } 1 \qquad i, j = 1,2,...,n$$

## *P*- Median Model – Example 5

- Setup LINGO model for this example

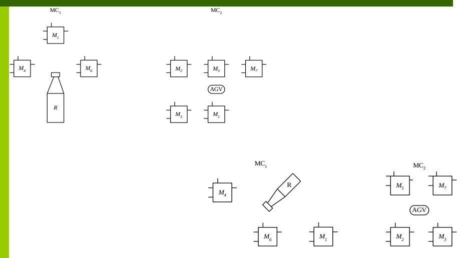
$[d_{ij}]$ =

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 0 | 6 | 1 | 5 | 5 | 6 |
| 2 | 6 | 0 | 5 | 1 | 3 | 2 |
| 3 | 1 | 5 | 0 | 4 | 4 | 5 |
| 4 | 5 | 1 | 4 | 0 | 2 | 3 |
| 5 | 5 | 3 | 4 | 2 | 0 | 3 |
| 6 | 6 | 2 | 5 | 3 | 3 | 0 |

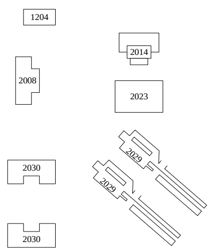
## Design & Planning in CMSs

- Machine Capacity
- Safety and Technological Constraints
- Upper bound on number of cells
- Upper bound on cell size
- Inter-cell and intra-cell material handling cost minimization
- Machine Utilization
- Machine Cost minimization
- Job scheduling in cells
- Throughput rate maximization

## Design & Planning in CMSs



## Grouping and Layout Project




- See input data file for GTLAYPC program
- Run GTLAYPC program
- See output data file for GTLAYPC program

## Grouping and Layout Project - Solution