Index

[**Cellular Manufacturing System (CMS)** 2](#_Toc93414149)

[**Cell Formation Problem** (Machine-Part Cell Formation) 2](#_Toc93414150)

[Solving methods 3](#_Toc93414151)

[Funciones objetivo 3](#_Toc93414152)

[Group Efﬁciency 3](#_Toc93414153)

[Group Efficacy 3](#_Toc93414154)

[Intercellular movements 4](#_Toc93414155)

[Intracellular movements 4](#_Toc93414156)

[Multiobjetivo 4](#_Toc93414157)

[Encodings 4](#_Toc93414158)

[Machine-part assignation 4](#_Toc93414159)

[Definition of the number of cells 5](#_Toc93414160)

[Static 5](#_Toc93414161)

[Dynamic (DCFP) 5](#_Toc93414162)

[Constraints 5](#_Toc93414163)

[Alternative Routing 5](#_Toc93414164)

[Alternative Machines 5](#_Toc93414165)

[Li 2021. 5](#_Toc93414166)

[Human Resources Allocation 5](#_Toc93414167)

[**Ulutas (2019)** - Clonal Selection Algorithm 6](#_Toc93414168)

[Artificial immune systems (McCall, 2005) 7](#_Toc93414169)

[**Tunnukij & Hicks (2009)** - Enhanced Grouping Genetic Algorithm (EnGGA) 8](#_Toc93414170)

[**Ghosh (2020)** – Optimal design of manuf. cells considering **machine usage percentage** 9](#_Toc93414171)

[Conclusiones 9](#_Toc93414172)

[**Eliguzel & Ozceylan (2019)** - Comparison of Different Clustering Methods for Cellular Manufacturing: A Case of Gym Centre 10](#_Toc93414173)

[Conformación celdas 10](#_Toc93414174)

[**Pichandi et al. (2021)** – CARIMO 10](#_Toc93414175)

[**REFLEXIONES PROPIAS** 10](#_Toc93414176)

# **Cellular Manufacturing System (CMS)**

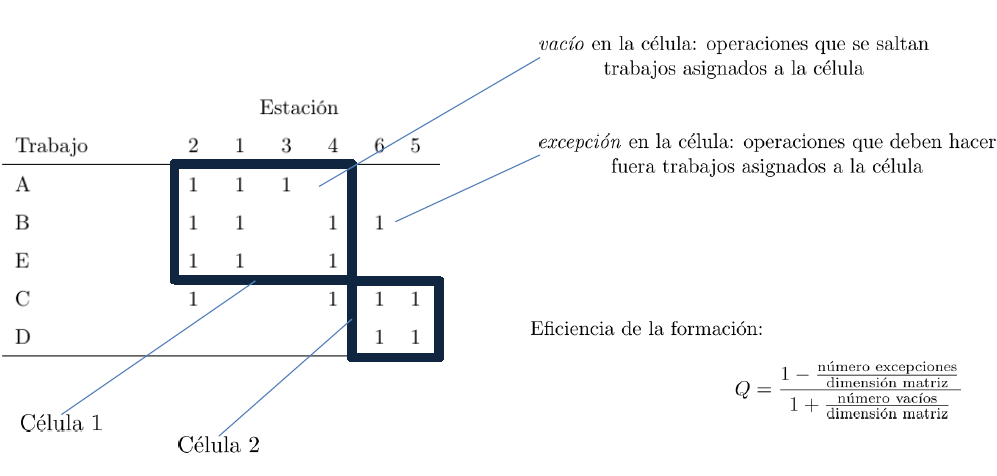
There are many different problems in designing effective CMS (chaudhuri, 2020)

* Cell formation problem
* Layout
  + Cell layout
  + Intracellular machine layout
* Group Scheduling
* Resource allocation problem
* Cell-related cost

# **Cell Formation Problem** (Machine-Part Cell Formation)

**Cell formation problem** (CFP) that is one of the critical CMS design problems is the assignment of parts and machines to specific cells based on their similarity.

A cellular manufacturing system considers set of part types, part type demand and machines (resources) and basically design problems deal with forming part families according to their processing requirements, grouping machines into manufacturing cells and assigning part families to cells.



La dimensión de la matriz es el número total de operaciones

Tunnukij and Hicks (2009) state that each cell must contain at least one part and one machine.

Researchers have attempted to solve the MPCF in three different ways:

1. Part grouping followed by machine cell creation
2. Machine grouping followed by part family creation

3. Simultaneous grouping of parts and machines

## Solving methods

* One approach to the formation of cells is to permute the rows and columns of the binary incidence matrix with the goal of inducing a blockdiagonal form in the reordered matrix. An inherent limitation of permutation methods is that they do not explicitly place parts and machines into cells and, accordingly, this must be accomplished either visually or via some type of post-processing method.
* An alternative approach to cell formation is the direct clustering of the part–machine matrix. At its fundamental level, the process is one of two-mode partitioning (van Rosmalen et al., 2009), which is also known as co-clustering or biclustering.
* El tercer camino sería el empleado por (Ulutas, tunnukij, etc) que consiste en crear encodings sobre los que aplicar metaheurísticas.

## Funciones objetivo

### *Group Efﬁciency*

(Chandrasekharan and Rajagopalan 1986b), g, which can be expressed as follows:



where MU1 is the ratio between the number of 1’s in the diagonal blocks of the solution matrix and the total number of elements in the diagonal blocks and MU22 Is the ratio between 0’s in the off-diagonal blocks of the solution matrix and total number of elements in the off-diagonal blocks.

### Group Efficacy

(Suresh Kumar and Chandrasekharan 1990).

Imagen que contiene Gráfico de cajas y bigotes

Descripción generada automáticamente

### Intercellular movements

### Intracellular movements

### Multiobjetivo

## Encodings

Codificación de la matriz de incidencia parte-máquina de tal forma que se incluya

Ejemplos:

* Ulutas (2019)
* Elbenani (2012) (diría que es el mismo que se usa tmb en Tunnukij, 2009)
* Falkenauer E (1998) Genetic algorithm for grouping problems.

## Machine-part assignation

The design procedure of a manufacturing cell comprises the following three main phases: (Nalluri 2019)

* grouping of machines as machine cells, also known as the cell formation problem (CFP)
* Arrangement of parts into different part families depending

on the production similarities

* Mapping of these machine cells and product families (McAuley 1972; King 1980; Ham et al. 1985).

OTROS AUTORES (ULUTAS 2019, TUNNUKIJ 2009, ETC) SE SALEN DE ESTA DEFINICIÓN DE NALLURI AL SACAR LAS CELDAS COMPLETAS (MAQUINAS+PARTES) DE UNA VEZ A TRAVÉS DE LOS ENCODINGS.

Por lo tanto, distingo entre dos tipos de asignación

* Separately assignation (Nalluri, 2019)
* Simultaneously assignation, donde la creación de celdas y familias de partes, así como su asignación, tienen lugar de forma simultánea.

## Definition of the number of cells

### Static

El algoritmo recibe el número de celdas como parámetro.

### Dynamic (DCFP)

El algoritmo explora distintas soluciones con diferentes números de celdas y devuelve la mejor.

M. Rheult, J.R. Drole, G. Abdulnour, Dynamic cellular manufacturing (DCMS), Comput. Ind. Eng. 31 (1–2) (1996) 143–146.

## Constraints

### Alternative Routing

Li 2021:

Texto

Descripción generada automáticamente

### Alternative Machines

### Li 2021.

### Human Resources Allocation

Review en [20]

[20] B. Bidanda, P. Ariyawongrat, K.M. Needy, B.A. Norman, W. Tharmmaphornphilas, Human-related issues in manufacturing cell design, implementation,and operation: a review and survey, Comput. Ind. Eng. 48 (3) (2005) 507–523.

**Azadeh, 2017:**

Tabla

Descripción generada automáticamente

#### Required Skills

[11] N. Safaei, M. Saidi-Mehrabad, M.S. Jabal-Ameli, A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system, Eur. J. Oper. Res. 185 (2) (2008) 563–592.

# **Ulutas (2019)** - Clonal Selection Algorithm

* Artificial Immune System (AIS) algorithm.
* it can obtain the optimum number of cells to generate best efficacy value.

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza baja

PARÁMETROS: population size, receptor editing, termination criteria.

N%? 🡪 SELECTION ROULETTE WHEEL

R%? 🡪 10% (igual que B%)

Affinity? mejor función objetivo

**ENCODING**

Parts index **i** takes values from 1 to **p**

Machines index **j** takes values from 1 to **m**

Cells index **k** takes values from 1 to **c** = min (p,m) = **MaxCell** = min(parts,machines)

Cell number is represented as “0” in the encoding that is randomly generated. The number of “0” in an encoding is equal to MaxCell-1.

Problem with 4 parts and 8 machines

The encoding can be represented as

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

(Three ‘0’ = MaxCell -1)

where first 4 numbers correspond to part number and the rest defines the machines.

An antibody can randomly be generated as:

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

### Artificial immune systems (McCall, 2005)

<https://www.sciencedirect.com/science/article/pii/S0377042705000774>

AIS are a problem-solving heuristic technique inspired by the human immune system in the same way that GAs are inspired by evolution. AIS are a computational technique that use biological immune systems as a problem-solving metaphor in the same way that genetic algorithms use biological evolution. Key properties of immune systems are translated into the data structures and workflow of AIS. Early collaborations in this area between researchers in genetic algorithms, immunology and complexity theory include Kauffman et al. [[18]](https://www.sciencedirect.com/science/article/pii/S0377042705000774" \l "bib18) and Forrest et al. [[14]](https://www.sciencedirect.com/science/article/pii/S0377042705000774" \l "bib14).

De Castro [[10]](https://www.sciencedirect.com/science/article/pii/S0377042705000774" \l "bib10) identifies several ways in which AIS are applied. For example, in biological immune systems, clonal selection creates immune cells that match a particular antigen by a process of proliferation and differentiation biased towards producing cells that match the antigen. This process is mimicked by some AIS used for learning and optimisation tasks. The analogy is that naive immune cells represent solutions to a learning problem represented by the antigens. Clonal selection is then an evolutionary process that results in highly fit solutions to the learning problem. A typical algorithm would work along the following lines. Candidate solutions (to a learning problem say) are encoded as (bit) strings. Each solution is presented with a set of antigens (representing a training set for the learning problem, also encoded as strings) and an affinity measure is calculated. Those solutions with the highest affinity to the antigens are preferentially selected for cloning. Once cloned, the copies of selected solutions are then “differentiated” using one of a set of possible mutation operators. The process then iterates until some stopping condition is reached. This approach bears many similarities to a GA. However, the particular selection, reproduction and mutation processes are quite distinctive to AIS.

Another feature of immune systems that can be reflected in an AIS is self–nonself discrimination. The property that T-cells that can discriminate only nonself can be evolved through a negative selection process is an important one for distributed computer systems where viruses, spam emails and other attacks are a large and growing problem. Algorithms that can monitor computer systems to detect anomalous programs and files promise to be of great use in combatting this problem. De Castro [[10]](https://www.sciencedirect.com/science/article/pii/S0377042705000774#bib10) provides a top-level workflow for a typical AIS implementing negative selection to solve some anomaly detection problem. He also identifies areas of commonality and complementarity between AIS and other “Soft Computing” paradigms such as GAs and Neural Networks and provides some useful suggestions for the development of more sophisticated hybrid algorithms.

# **Tunnukij & Hicks (2009)** - Enhanced Grouping Genetic Algorithm (EnGGA)

* Chromosome representation

Tabla

Descripción generada automáticamente

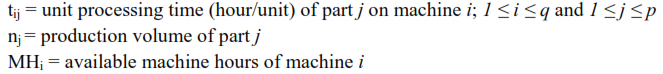
* Initial population
* Crossover
  + **Two parents are randomly chosen from the populatio**n (si son iguales sustituyen uno aleatoriamente de entre la poblacion, en Falkenauer 1998 lo mantienen. Sin embargo, cuando se acerca la convergencia esta búsqueda se dificulta y se rastrea hasta un máximo del 30% de la pop.)
  + **Two crossover points are randomly selected** from the group section of each parent
  + **Genes** from the 1st parent **are** **copied** to the 1st child  
    Genes from the 2nd parent are copied to the 2nd child
  + The section within the crossover points of 2nd parent is appended to the 1st child.   
    Section from 1st parent appended to the 2nd child.
  + All the parts and machines that belong to the cells within the appended section are inherited by the child.
* Mutation
  + Parent is chosen from the population randomly
  + Number of cells is checked
    - **Number of cells more than 2**: STANDARD ELIMINATION MUTAT.  
      one of the cells in the group sections is randomly selected and all of its elements are eliminated. Remaining elements are inherited by child.
    - **Number of cells less than 2**: MODIFIED DIVISION MUTAT.  
      A cell that contains at least two parts and two machines is randomly selected and divided into two new cells.
* Repair Process

Process to rectify infeasible chromosomes produced by genetic operations.

* + Checking and removing empty cells
  + Checking the number of cells
  + Greedy Heuristic
  + Renumbering the groups

# **Ghosh (2020)** – Optimal design of manuf. cells considering **machine usage percentage**

Incluye una nueva variable en el problema: **el porcentaje de ocupación de cada trabajo en cada máquina**. De tal manera que cada valor (uij) de la matriz de incidencia máquina-trabajo oscile entre 0 y 1.



Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Con la restricción de que la suma sea menor o igual que 1 para cada máquina.

Un reloj de aguja

Descripción generada automáticamente con confianza baja

Por otra parte:

* Crea datos específicos según unos criterios que define para que sean realistas.
* Crea un modelo de minimización multiobjetivo en el que busca:
  + Minimizar: Total utilization of exceptional elements
  + Maximizar: Total in-cell utilization
  + Minimizar el número de vacíos.
* Define una nueva métrica del rendimiento: Utilization-based grouping efficacy

ESTUDIAR LA FUNCIÓN OBJETIVO

## Conclusiones

Es capaz de priorizar la agrupación de máquinas con los trabajos más importantes para las mismas.

Tampoco se adapta a la realidad, porque me puede llegar a interesar el escenario 1 frente al escenario 2 (que sería, en principio y a falta de estudiar bien FO, favorecido por Ghosh):

* Escenario 1: Un trabajo pertenece a una celda con máquinas que están poco tiempo ocupadas por el mismo y además ese trabajo tiene una única excepción con un porcentaje de ocupación altísimo.
* Escenario 2: El mismo trabajo está asignado a una celda con la máquina que ocupa con un porcentaje elevado y tiene como excepciones varias máquinas que ocupa en porcentajes pequeños. ESTO PROVOCARÍA MÁS TRASLADO DEL TRABAJO, ESPECIALMENTE SI LAS EXCEPCIONES ESTÁN EN CELDAS DISTINTAS.

# **Eliguzel & Ozceylan (2019)** - Comparison of Different Clustering Methods for Cellular Manufacturing: A Case of Gym Centre

ESTUDIAR BIEN ROC Y ALC

## Conformación celdas

Crea celdas únicamente asociando máquinas, no trabajos y máquinas

Aplicación, Tabla

Descripción generada automáticamente

Por lo tanto, un trabajo puede estar asignado a varias celdas.

Hay cosas que no me cuadran:

¿Qué se está realmente evaluando aquí? Porque, a efectos prácticos, el trabajo 1 está asignado a la celda 2 y el trabajo 5 no lo está. ¿Por qué selecciona unos si y otros no?

¿No deberían estar las celdas exclusivamente conformadas por líneas verticales que contemplen todos los trabajos para cada celda de máquinas?

# **Pichandi et al. (2021)** – CARIMO

The heuristic described in this paper ﬁnds non-singleton machine-part cells using:

* Correlation analysis (CA) 🡪 **part families**
* Relevance index-modiﬁed (RIMO) 🡪 most appropriate machine cell for each part family.

The CARIMO heuristic has achieved a higher GE than the GA2 algorithm and CARI heuristic for 14.3% and 31.4% of the standard test instances, respectively.

# **REFLEXIONES PROPIAS**

M1 M2 M3 M4 M5

P5 1 --N5-> 1 0 --N5-> 1 0

P6 1 1 1 0 0

Se podría minimizar volumen (N) que va desde una celda a otra (mejor que partes, porque así tenemos en cuenta que el mix de partes puede ser distinto. Aunque podemos usar este caso -todo el mix igual- para poder compararlo bien con los cálculos de efficacy que hemos hecho hasta ahora).

Idea: avanzar hacia un panel multiobjetivo con distintos KPI para tener en cuenta.

(efficacy, efficacy con volumen de transporte, etc.) (más a largo plazo también se podría incluir minimización de ct, inventarios medios, etc)

M1 M2 M3 | M4 M5

P1 1 0 1 | 0 0

P2 1 1 1 | 0 0

P3 0 0 0 | 0 1

P4 0 0 0 | 1 1