Index

[**Introducción** 2](#_Toc94264384)

[**Cellular Manufacturing System (CMS)** 2](#_Toc94264385)

[**Cell Formation Problem** (Machine-Part Cell Formation) 2](#_Toc94264386)

[Introducción 2](#_Toc94264387)

[Solving methods 3](#_Toc94264388)

[Exact methods 4](#_Toc94264389)

[Metaheuristics 4](#_Toc94264390)

[Funciones objetivo 4](#_Toc94264391)

[Group Efﬁciency 4](#_Toc94264392)

[Group Efficacy 4](#_Toc94264393)

[Intercellular movements 4](#_Toc94264394)

[Intracellular movements 4](#_Toc94264395)

[Multiobjetivo 5](#_Toc94264396)

[Encodings 5](#_Toc94264397)

[Machine-part assignation 5](#_Toc94264398)

[Number of cells 6](#_Toc94264399)

[Static 6](#_Toc94264400)

[Dynamic (DCFP) 6](#_Toc94264401)

[Constraints 6](#_Toc94264402)

[Alternative Routing 6](#_Toc94264403)

[Alternative Machines 6](#_Toc94264404)

[Human Resources Allocation 6](#_Toc94264405)

[**Ulutas (2019)** - Clonal Selection Algorithm 7](#_Toc94264406)

[**Tunnukij & Hicks (2009)** - Enhanced Grouping Genetic Algorithm (EnGGA) 9](#_Toc94264407)

[**Ghosh (2020)** – Optimal design of manuf. cells considering machine usage % 10](#_Toc94264408)

[**Eliguzel & Ozceylan (2019)** - Comparison of Different Clustering Methods for CMS 11](#_Toc94264409)

[**Pichandi et al. (2021)** – CARIMO 11](#_Toc94264410)

[**Conclusiones y trabajos futuros** 12](#_Toc94264411)

[Continuación de la revisión del estado del arte y taxonomía 12](#_Toc94264412)

[Propuesta propia para la resolución del problema de formación de células 13](#_Toc94264413)

[Reflexiones y notas propias 13](#_Toc94264414)

# **Introducción**

El objetivo de este documento es resumir parte del conocimiento adquirido, además de reflejar pensamientos y reflexiones del autor, durante la revisión del estado del arte sobre la optimización de sistemas de fabricación celular.

Junto a este documento descriptivo se encuentran todos los PDFs de los artículos relevantes recopilados hasta el momento. En el archivo **02 REFERENCIAS.pdf** hay una bibliografía donde quedan reflejados la mayoría de ellos. Por otra parte, los artículos mencionados están resumidos esquemáticamente en las hojas de cálculo del fichero **Analisis\_Bibliografia.xlsx**. Las categorías utilizadas para describir cada una de las propuestas responden a los primeros bosquejos de una taxonomía del problema a tratar. Entre ellas encontramos: el tipo de método de resolución, la función objetivo del algoritmo de optimización o las restricciones del problema. Estas categorías son desarrolladas en el apartado Cell Formation Problem de este documento. Para acabar, hay que mencionar que se ha comenzado a realizar un mapa mental, con el objetivo de dar una visualización a la taxonomía, y que se encuentra en el archivo **01 TAXONOMÍA.drawio**.

# **Cellular Manufacturing System (CMS)**

Hay múltiples problemas a la hora de enfrentarse a la optimización de un sistema de fabricación celular (Chaudhuri, 2020).

* **Cell Formation Problem:** Creación de las células de fabricación. Asignación de trabajos (parts) y máquinas a un número indeterminado de células.
* **Layout**: Disposición en la planta de la fábrica
  + Cell layout: Disposición de las distintas células en la planta de la fábrica.
  + Intracellular machine layout: Disposición de las distintas máquinas

dentro de cada célula.

* **Group Scheduling:** Programación de la producción. Con sus correspondientes particularidades en sistemas de fabricación celular.
* **Resource allocation problem:** Asignación de recursos a cada una de las células. Mención especial a la asignación de recursos humanos (Azadeh, 2017).
* **Cell-related cost:** Optimización de costes particularmente asociados a sistemas de fabricación celular.

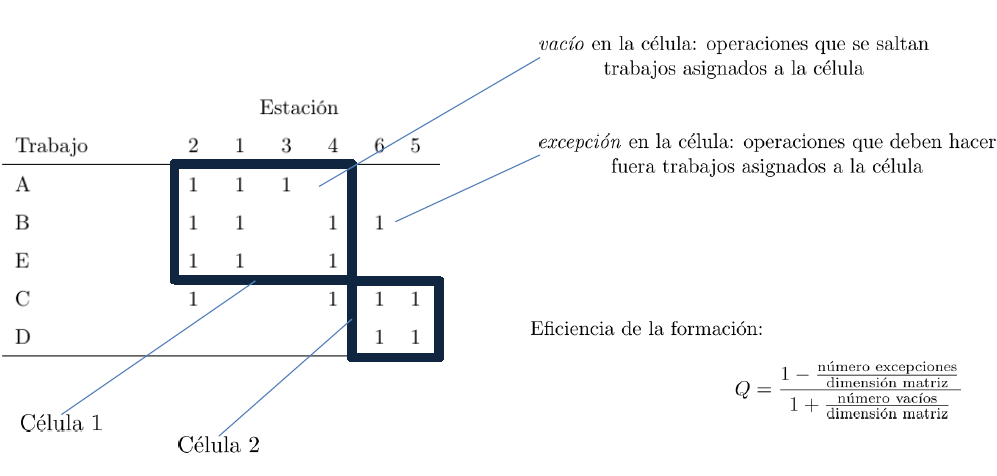
En este documento nos centraremos fundamentalmente en el primero de los problemas comentados, la creación de las células de fabricación (cell formation problem)

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

## Introducción

**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.

En (Papaioannu, 2010), la única review que he encontrado que merece la pena, se hace la siguiente clasificación de los métodos basados en datos de producción:

* Cluster analysis.
* Graph partitioning approaches.
* Mathematical programming methods.
* Heuristic and Metaheuristic algorithms.
* Artificial intelligence methodologies.

### Métodos exactos

A través de la optimización de modelos matemáticos del problema, habitualmente MILP (Mixed-Integer Linear Programming), se pretende obtener la solución óptima de una forma exacta. Funciona bien para problemas sencillos, pero los tiempos de resolución se disparan cuando la complejidad del problema aumenta (NP-problem).

(Brusco, 2015); (Azadeh, 2017); (Bychkov & Batsyn, 2018)

### Metaheurísticas

Aplicación de metaheurísticas para la obtención de soluciones aproximadas. Funcionan mucho mejor con problemas de gran tamaño. Muchas de ellas son algoritmos bioinspirados muy famosos como los algoritmos genéticos, colonias de hormigas, etc…

## Funciones objetivo

Son las funciones objetivo del algoritmo que pretenden ser optimizadas. Las podemos entender también como KPIs (Key Performance Indicators) del sistema de fabricación celular.

### *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

Es la medida más utilizada en algoritmos de optimización del CFP. No confundir con group efficiency.

(Suresh Kumar and Chandrasekharan 1990).

Imagen que contiene Gráfico de cajas y bigotes

Descripción generada automáticamente

### Intercellular movements

Número de movimientos de los trabajos entre distintas células. Minimizar estos movimientos tiene lógicamente beneficios porque implica minimizar los tiempos de transporte en la planta de la fábrica.

### Intracellular movements

Maximizar el número de movimientos dentro de la célula (intracelulares) equivaldría a minimizar los movimientos entre distintas células (intercelulares).

### Multiobjetivo

Algoritmos que no tratan de optimizar un único

Muchos incluyen también objetivos más específicos como el VAWIP (value added work in progress) o el Group Capability Index.

## Encodings

Codificación de la matriz de incidencia parte-máquina de tal forma que se incluya la definición de las celdas. Es decir, transformar la matriz en un vector donde quede claro qué máquinas y qué trabajos pertenecen a cada una de las celdas.

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).

Sin embargo, leyendo el trabajo de otros autores (ULUTAS 2019, TUNNUKIJ 2009, ETC) vemos que estos se salen de la definición dada en (Nalluri, 2019) puesto que obtienen las celdas completas (grupos de máquinas + familias de trabajos) simultáneamente. Para esto vemos que sus algoritmos se sirven de los ENCODINGS.

Por lo tanto, distinguimos entre dos tipos de asignación:

* **Separately assignation** correspondería a los descrito en (Nalluri, 2019)

En este caso, se crean los grupos de máquinas y las familias de trabajos de forma secuencial. Distinguimos dos casos:

* + Crear primero los grupos de máquinas y luego las familias de trabajos
  + Crear primero las familias de trabajos y luego los grupos de máquinas
* **Simultaneously assignation**  
  Donde la creación de celdas y familias de partes, así como la asignación entre celdas y familias, tienen lugar de forma simultánea.

## 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

Texto

Descripción generada automáticamente

### Alternative Machines

Li 2021.

### Human Resources Allocation

Review:

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

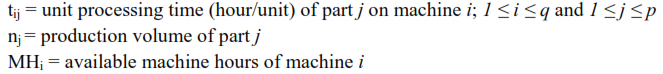
* Initial population
* Crossover
  + **Two parents are randomly chosen from the populatio**n (si son iguales sustituyen uno aleatoriamente de entre la poblacion, en Falkenauer 1998 lo mantienen. Sin embargo, cuando se acerca la convergencia esta búsqueda se dificulta y se rastrea hasta un máximo del 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 %

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 CMS

ESTUDIAR BIEN LAS HEURÍSTICAS DE 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.

# **Conclusiones y trabajos futuros**

En cuanto a los trabajos futuros, quedan dos grandes líneas abiertas. Por un lado, continuar con la revisión de la bibliografía (intentando llevar a cabo de forma paralela una taxonomía de los problemas asociados al CMS) y, por otro, llevar a cabo una propuesta propia para la resolución del CFP.

### Continuación de la revisión del estado del arte y taxonomía

Han quedado varios estados del arte por revisar en artículos no demasiado antiguos y publicados en revistas de prestigio:

**Revisión del estado del arte en (Imram, 2017)**

Se centra en aproximaciones al problema que optimizan KPIs (Key Performance Indicators) muy variados. Su propia solución se centra en la optimización del VAWIP (Value Added Work in Progress).

**Revisión del estado del arte en (Danilovic, 2019)**

Se centra en aproximaciones al problema más habitual, la optimización del grouping efficacy.

**Revisión del estado del arte de artículos que emplean REDES NEURONALES:**

La tabla a continuación indica artículos antiguos que introdujeron el uso de redes neuronales para el problema de formación de células. En la actualidad hay bastantes artículos modernos que las utilizan y convendría revisarlos.

|  |  |
| --- | --- |
| Chu | 1997 |
| Chattopadhyay et al. | 2012 |
| Chen and Cheng | 1995 |
| Guerrero et al. | 2002 |
| Pandian and Mahapatra | 2009 |
| Yang and Yang | 2008 |

### Propuesta propia para la resolución del problema de formación de células

Identificamos dos objetivos fundamentales de la investigación, el primero a medio y el segundo a largo plazo.

El primero consistiría en **la implementación de un algoritmo propio** para la resolución del problema CFP para optimizar algún KPI (o un grupo de KPIs) que se haya encontrado más interesante en la revisión del estado del arte. La propia definición de qué KPIs pueden ser más interesante en cada situación supone ya una investigación en sí misma.

El segundo trataría de llevar a cabo un cuadro de mandos, que haciendo uso del algoritmo implementado y/o otros algoritmos del estado del arte, propusiera múltiples soluciones a un problema de CFP evaluando múltiples KPIs. Este cuadro de mando tendría una utilidad industrial muy clara a la hora de la toma de decisiones relativas a la reestructuración de la planta en sistemas de fabricación celular.

### Reflexiones y notas 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