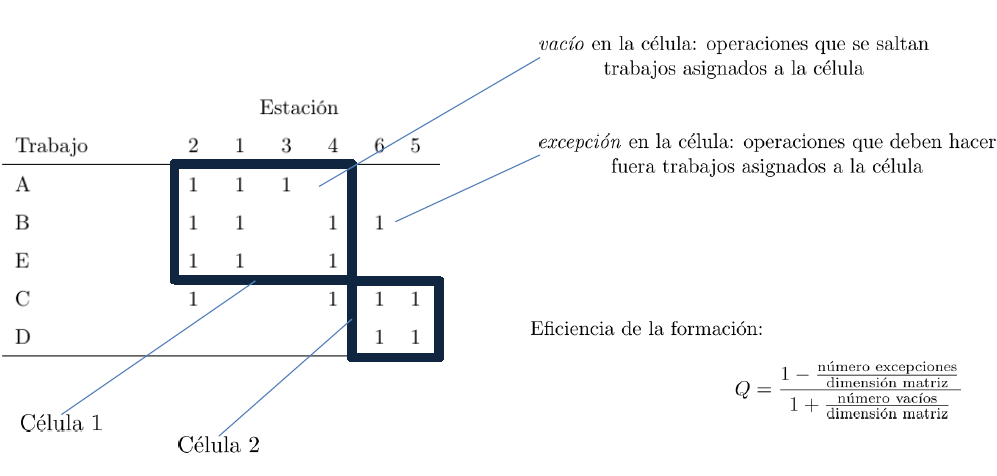
**Cell formation problem** (CFP) that is one of the critical CMS design problems is the assignment of parts and machines to specific cells based on their similarity.

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.

**5.4. Artificial immune systems (McCall, 2005)**

**https://www.sciencedirect.com/science/article/pii/S0377042705000774**

To complete this section, we briefly review the area of artificial immune systems (AIS). 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 a number of 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

**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%)

Affinitie? 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}