

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**RESOLUCIÓN DEL MANUFACTURING CELL  
DESIGN PROBLEM UTILIZANDO CUCKOO SEARCH**

**GUSTAVO ANDRÉS ZÚÑIGA MOYA**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

JULIO, 2014

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**RESOLUCIÓN DEL MANUFACTURING CELL  
DESIGN PROBLEM UTILIZANDO CUCKOO SEARCH**

**GUSTAVO ANDRÉS ZÚÑIGA MOYA**

Profesor Guía: **Ricardo Soto de Giorgis**

Profesor Co-referente: **Broderick Crawford Labrín**

Carrera: **Ingeniería de Ejecución en Informática**

Julio, 2014

Dedico esta memoria a mi familia,  
amigos y profesores, ya que gracias  
a ellos, se ha vuelto realidad este  
sueño presentado en mi vida.

GUSTAVO ANDRÉS ZÚÑIGA M.

# Índice

Resumen .....	iii
Abstract.....	iv
Lista de Figuras.....	v
Lista de Tablas .....	vi
<b>1 Introducción .....</b>	<b>1</b>
<b>2 Objetivos.....</b>	<b>2</b>
<b>2.1 Objetivo General .....</b>	<b>2</b>
<b>2.2 Objetivos Específicos.....</b>	<b>2</b>
<b>3 Estado del Arte.....</b>	<b>3</b>
<b>4 Manufacturing Cell Design Problem .....</b>	<b>4</b>
<b>4.1 Modelo Matemático.....</b>	<b>4</b>
<b>5 Cuckoo Search.....</b>	<b>6</b>
<b>5.1 Algoritmo Cuckoo Search .....</b>	<b>7</b>
<b>5.1.1 Lévy Flight .....</b>	<b>8</b>
<b>6 Prototipo.....</b>	<b>9</b>
<b>6.1 Características del equipo .....</b>	<b>9</b>
<b>6.2 Experimentación.....</b>	<b>9</b>
<b>6.3 Software.....</b>	<b>10</b>
<b>6.3.1 Función de Generación inicial de nidos .....</b>	<b>11</b>
<b>6.3.2 Función Reparadora.....</b>	<b>11</b>
<b>6.3.3 Funciones de Binarización.....</b>	<b>11</b>
<b>6.3.3.1 Funciones de la forma V .....</b>	<b>12</b>
<b>6.3.3.2 Función Sigmoide.....</b>	<b>13</b>
<b>6.3.4 Matriz de Similitud.....</b>	<b>14</b>
<b>6.3.4.1 Creación de las matrices de Similitud .....</b>	<b>14</b>
<b>6.3.4.1 Algoritmo de la Matriz de Similitud.....</b>	<b>16</b>
<b>7 Experimentos.....</b>	<b>17</b>
<b>7.1 Características .....</b>	<b>17</b>
<b>7.2 Resultados.....</b>	<b>17</b>

<b>7.3 Análisis de resultados</b> .....	21
<b>8 Conclusiones</b> .....	22
<b>9 Referencias</b> .....	23
<b>Anexos</b> .....	1
<b>A Problemas a Resolver</b> .....	2

## Resumen

El problema de diseño de celdas de manufactura o Manufacturing Cell Design Problem (MCDP), consiste en organizar una fábrica en celdas de manufactura con el objetivo de minimizar los traslados de partes que pertenecen a una misma familia entre las distintas celdas.

Este proyecto se centra en resolver el problema utilizando la metaheurística Cuckoo Search, la cual está inspirada en la capacidad parasitaria del ave cuckoo. La solución presentada además incluye el uso de funciones de transferencia y matrices de similitud con el objetivo de mejorar los resultados.

**Palabras clave:** Manufacturing Cell Design Problem, Cuckoo Search, metaheurística.

## **Abstract**

The manufacturing problem cell design problem (MCDP) consists on organizing a factory in manufacturing cells in order to minimize the flow of parts that belong to a same family among the different cells.

This project focuses on solving this problem by using the metaheuristic called Cuckoo Search, which is inspired by the parasitic behaviour of the cuckoo bird. The proposed solution also involves the use of transfer functions and similarity matrices in order to improve results.

**Keywords:** Manufacturing Cell Design Problem, Cuckoo Search, metaheuristic.

# Lista de Figuras

<b>Figura 1: Algoritmo Cuckoo Search .....</b>	<b>7</b>
<b>Figura 2: Gráfica de la Forma V .....</b>	<b>12</b>
<b>Figura 3: Gráfica de la función Sigmoide .....</b>	<b>13</b>
<b>Figura 4: Algoritmo de la Matriz de Similitud .....</b>	<b>16</b>

# Lista de Tablas

<b>Tabla 1: Ejemplo de Matriz máquinas x partes</b> .....	15
<b>Tabla 2: Ejemplo de Matriz de similitud respecto a máquinas</b> .....	15
<b>Tabla 3: Ejemplo de Matriz de similitud respecto a partes</b> .....	15
<b>Tabla 4: Resultados óptimos de los problemas</b> .....	18
<b>Tabla 5: Resultados del problema 1</b> .....	18
<b>Tabla 6: Resultados del problema 2</b> .....	18
<b>Tabla 7: Resultados del problema 3</b> .....	19
<b>Tabla 8: Resultados del problema 4</b> .....	19
<b>Tabla 9: Resultados del problema 5</b> .....	19
<b>Tabla 10: Resultados del problema 6</b> .....	19
<b>Tabla 11: Resultados del problema 7</b> .....	20
<b>Tabla 12: Resultados del problema 8</b> .....	20
<b>Tabla 13: Resultados del problema 9</b> .....	20
<b>Tabla 14: Resultados del problema 10</b> .....	20

# 1 Introducción

Este informe se basa en la resolución del Manufacturing Cell Design Problem (MCDP) con la utilización de la metaheurística Cuckoo Search (CS). Fundamentalmente se buscará un método con el cual se pueda adaptar el CS a la presentación matricial en que se desarrolla el MCDP.

En la actualidad del ámbito empresarial, específicamente sobre términos de desarrollo y creación, ha existido una constante búsqueda del mayor rendimiento posible, con el fin de reducir los costos al mínimo y aumentar los beneficios. A causa de lo anterior y con la visión extendida al caso planteado, se origina el MCDP. Esta sigla se divide en dos términos, en primer lugar, está el “Manufacturing Cell Design”, el cual se define como una metodología que representa una máquina manufacturera en matrices, es decir, organiza las partes pertenecientes a cada máquina y su ubicación en celdas, otorgando una visión más amplia de cómo está organizada la máquina. En segundo lugar, se halla el “Problem” relacionándose directamente con el problema del Manufacturing Cell Design, cuyo objetivo es minimizar los traslados de partes que pertenecen a una misma familia entre las distintas celdas, obteniendo de esta manera el camino óptimo en que se crea el producto.

En este proyecto se verá el estudio particular de la metaheurística Cuckoo Search. La expresión “metaheurística” deriva del griego “meta”, que se refiere a ir más allá, además “heurística”, que significa encontrar, y por definición, es un método de búsqueda para resolver un problema computacional. Se basan en casos poco convencionales, por ejemplo, el comportamiento de un enjambre. Por otro lado, el Cuckoo Search es una técnica inspirada en la capacidad parasitaria que posee una especie particular de ave Cuckoo, donde ésta pone huevos “parásitos” en nidos de otras aves, con el fin de ser criados por el ave anfitriona, aunque a veces los huevos corren el riesgo de ser detectados por el huésped, siendo desechados o abandonados.

Para comprobar la adaptación del Cuckoo Search en el MCDP, se realizó una serie de experimentos obtenidos del trabajo de Boctor [12], con el propósito de verificar la eficiencia que posee esta metaheurística en cuanto a la resolución del problema.

Para apoyar la adaptación del CS en el MCDP, se utilizan algoritmos de binarización y además se incluye el uso de matrices de similitud con el fin de encontrar mejores resultados sobre el problema.

## **2 Objetivos**

### **2.1 Objetivo General**

- Resolución del Manufacturing Cell Design Problem (MCDP) utilizando Cuckoo Search.

### **2.2 Objetivos Específicos**

- Comprender la metaheurística Cuckoo Search.
- Comprender el MCDP.
- Implementar el algoritmo para resolver el MCDP mediante el Cuckoo Search.
- Realizar experimentos con la implementación desarrollada.

### 3 Estado del Arte

Este trabajo se encuentra enfocado en el estudio de la resolución del Manufacturing Cell Design Problem. Las investigaciones sobre este tema se puede dividir en dos grupos; métodos de aproximación y de optimización global. Los métodos de aproximación contienen a las metaheurísticas y la gran mayoría tiende a explotar la diversificación de éstas, es decir, intentar recorrer una mayor cantidad de áreas con el fin de encontrar soluciones, las cuales no siempre logran el óptimo. En contrario, las técnicas de optimización global, se dedican a analizar detenidamente todo el espacio con el fin de encontrar el óptimo del problema, sin embargo, gastan una gran cantidad de trabajo, memoria y tiempo.

En el primer grupo de las metaheurísticas usadas para resolver el MCDP, se encuentran: Aljaber, Baek, Chen [1] y Lozano, Díaz, Eguía, Onieva [2] con el uso del Tabu Search. Wu, Chang, Chung [4] utilizando el Simulated Annealing (SA), Durán, Rodríguez y Consalter [8] combinan el Particle Swarm Optimization con el método del Data Mining, y Venugopal, Narendran [14] proponen usar Genetics Algorithms(GA), más adelante Gupta, Gupta, Kumar, Sundaram [15] deciden utilizar la misma solución representada con los GA pero con una optimización diferente multi-objetivo la cual se dirige a la simultanea minimización del total del número entre células con variación de carga celular.

En el segundo grupo se encuentra el uso de programación lineal por parte de Purcheck [16] Olivia-Lopez y Purcheck [17], modelos cuadráticos propuestos por Boctor [12] y Kusiak, Chow[18]. La programación por metas “Goal Programming” (GP) es otro paradigma de este grupo cuya función es la toma de decisiones con objetivos multiples, Sankaran [19], Shafter y Roger [9] propusieron modelos sobre este paradigma.

En este caso particular, se utilizó la metaheurística Cuckoo Search, creada por Xin-She Yang [5], sin embargo, el CS, como la mayoría de las metaheurísticas, se propone a diversificar, buscando un gran número de soluciones, pero a la vez, se dedica a intensificar en las áreas que puedan tener el óptimo del problema.

## 4 Manufacturing Cell Design Problem

El MCDP es una metodología popular que se está utilizando en la actualidad, en los sistemas de producción en masa. Este consiste principalmente en dividir la máquina manufacturera en celdas, cada celda se vuelve independiente con respecto a las otras, y cada una de estas se encuentra encargada de hacer su parte del producto, cuando éste pase por ellas.

La ventaja de distribuir de esta manera la máquina, es que se puede ver por cada lugar donde debe pasar el producto, mientras se buscan los mejores lugares donde distribuir las celdas, para así obtener un menor costo al crear el producto, mayor eficacia en el ámbito de la velocidad y reducción de los movimientos que debe realizar el producto al ir de un lugar a otro.

Tomando lo anterior, al dividir las máquinas en celdas, el algoritmo busca dejar éstas lo más compactas posible. También hay casos en los que existen más de una maquina principal, de modo que cada celda de las partes de la máquina deban estar asociadas a una máquina en particular; esto se reconoce como pertenecer a la misma familia.

Este problema es modelado mediante matrices. En total hay tres, una con respecto a las celdas, otra con respecto a las máquinas y una última con respecto a las partes de cada máquina.

### 4.1 Modelo Matemático

Se presentará el modelo del MCDP utilizado en [3].

Entrada:

Mmax: El número máximo de máquinas por celda.

M: El número de máquinas

P: El número de partes

C: El número de celdas

i: El índice de máquinas ( $i = 1, \dots, M$ )

j: El índice de partes ( $j = 1, \dots, P$ )

k: El índice de celdas ( $k = 1, \dots, C$ )

A =  $[a_{ij}]$  M x P La matriz de incidencia de la maquina-partes

MxC Vector de máquinas x celdas (adaptado a este problema en particular)

PxC Matriz de partes x celdas

$$a_{ij} = \begin{cases} 1 & \text{si } j^{\text{th}} \text{ visita la } i^{\text{th}} \text{ máquina} \\ 0 & \text{para el resto de los casos} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{si la máquina } i \text{ pertenece a la celda } k \\ 0 & \text{para el resto de los casos} \end{cases}$$

$$z_{jk} = \begin{cases} 1 & \text{si la parte } j \text{ pertenece a familia } k \\ 0 & \text{para el resto de los casos} \end{cases}$$

Finalmente el problema se representa de la siguiente forma:

$$\min \sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P a_{ij} z_{jk} (1 - y_{ik})$$

Sujeto a

$$\sum_{k=1}^C y_{ik} = 1 \quad \forall i, \quad \sum_{k=1}^C z_{jk} = 1 \quad \forall j, \quad \sum_{i=1}^M y_{ik} = M_{max} \quad \forall k$$

## 5 Cuckoo Search

El algoritmo usado en la investigación, se encuentra inspirado en las aves llamadas Cuckoo. Algunas especies de Cuckoo tienen una manera bastante particular de reproducirse. Ésta consiste en dejar huevos “parásitos” dentro de nidos pertenecientes a otra ave, de modo que sean criados por la anfitriona.

Este método pertenece a los nuevos algoritmos de metaheurística inspirada en la naturaleza, siguiendo la inteligencia del enjambre. El tipo de método es adaptable, ya que dependiendo del ambiente en que se encuentren, se irá adecuando para poder sobrevivir. Además, existen dos características cruciales dentro del Cuckoo Search, las cuales son intensificación y diversificación. La intensificación busca las mejores soluciones posibles explotando un área en particular, por otro lado, tenemos la diversificación, la cual permite al algoritmo explorar otras áreas con el fin de encontrar alguna con un buen potencial.

Ahora bien, en cuanto a la manera de reproducirse de los Cuckoo, primero se ponen en búsqueda de un huésped potencial, el cual sería un nido perteneciente a otra ave. Cuando es encontrado, el Cuckoo pone un huevo en dicho nido, para luego desplazarse nuevamente hacia algún otro nido donde incubar más crías. Sin embargo, no todo sale correctamente, varias veces ocurre que el ave huésped descubre el huevo ajeno en su nido, destruyéndolo, o abandonando el nido. También existen ciertos Cuckoo que son incluso capaces de producir huevos que se mimetizan en color y la forma de los huevos del huésped, con, así reduciendo la posibilidad de que sean descubiertos.

En la implementación de este algoritmo, se seguirán 3 pasos.

- 1- Cada Cuckoo pondrá un huevo a la vez, y será en un nido que se encontrará de manera aleatoria.
- 2- Los mejores nidos (soluciones), se mantendrán para las siguientes generaciones.
- 3- El número de nidos huéspedes será actualizado, además el huevo del Cuckoo será descubierto con una probabilidad de  $P_a$  perteneciente a  $\{0,1\}$ , en caso que ocurra, el ave botará el huevo o abandonará su nido.

También se debe tomar en cuenta que los Cuckoos para encontrar nuevos nidos, utilizan un algoritmo especial llamado Lévy Flight, el cual simula el vuelo de varios animales e insectos. La mayor característica del Lévy Flight es su vuelo aleatorio, con cierta restricción de lo que recorre, sin embargo, gracias a esta característica se logra que el conjunto pueda ser recorrido de una manera mucho más amplia para hallar una solución óptima, o cercana a ésta.

## 5.1 Algoritmo Cuckoo Search

El algoritmo se representará en pseudo lenguaje de su forma nativa, ya que es algo cargado ponerlo en pseudocódigo, la versión que se utilizará es una pequeña mejora de la que se encuentra en [10], el cual es el código puesto por Xin-She Yang en la dirección web [11].

```
Variables:

Pa: porcentaje que existe que un huésped encuentre el huevo del
Cuckoo en su nido
n: la cantidad de nidos huéspedes disponibles a la vez.
F: calidad de los nidos.

Función Cuckoo Search
Generar la población de n nidos huésped (los nidos son el
vector MxC)

While (N_Iter < F_Iter) o (F_iter = N de iteraciones a
realizar)
    Obtener n cuckoos aleatoriamente vía Lévy Flights /*ver en
[5]*/
    Aplicar función reparadora
    N_iter = N_Iter + 1
    Evaluar su calidad Fi
    For (int i, i<n,i++)
        Ver la calidad de cada nido encontrado
        Elegir un nido aleatorio entre n (dígase j)
        If (Fi<Fj)
            Reemplazar j por la nueva solución
        End if
    End For
    Ordenar las soluciones
    Una fracción (Pa) de los malos nidos son abandonados
    Generar la población de los nidos abandonados
    Ordenar las soluciones y guardar la mejor actual
End while
Procesar los resultados y visualizarlos
End
```

Figura 1: Algoritmo Cuckoo Search

### 5.1.1 Lévy Flight

Para la formula de los saltos aleatorios se utilizó el algoritmo de Mantegna[10].

$$1 \leq \beta \leq 2$$

$$u = \text{randn}(\text{size}(s)) * \sigma$$

$$v = \text{randn}(\text{size}(s))$$

$$\Gamma(z) = \text{gamma}(z)$$

$$\sigma = \frac{[\Gamma(1+\beta)\sin(\pi\beta/2)]^{1/\beta}}{[\Gamma[(1+\beta)/2] * \beta 2^{(\beta-1)/2}]^{1/\beta}}$$

$$\text{step} = u / (|v|)^{1/\beta}$$

El step es el valor que se obtiene finalmente y mediante Lévy Flight, usando el algoritmo de Mantegna, el cual se debe aplicar a la solución.

## 6 Prototipo

El IDE que se utilizó para realizar el prototipo fue Netbeans, bajo el lenguaje Java. Las principales razones de esta decisión, son debido a la comodidad que presenta el lenguaje y, también, al ser desarrollado orientado a objetos, se ve claramente y encapsulado, lo cual hace bastante legible y versátil de código, además de las bibliotecas que ayudan bastante en la tarea de la programación y de visibilidad.

### 6.1 Características del equipo

Para realizar la tarea, se utilizó un computador con las siguientes características listadas, las cuales son las más importantes con respecto al trabajo de cómputo.

- ❖ 8 Gb de memoria ram.
- ❖ Procesador AMD Phenom II X2 555 Black Edition, sin embargo en este momento funciona con 3 núcleos de 3.2 GHz cada uno.
- ❖ GPU, ATI Radeon HD 5670 1Gb DDR5.
- ❖ Placa Madre Asus M4A79XTD EVO.
- ❖ Sistema operativo, Windows 7 Ultimate.
- ❖ Netbeans IDE 7.4.

### 6.2 Experimentación

Este prototipo experimentó los casos de MCDP que se encuentran en el trabajo de Boctor [12], en cual se realizaron una gran cantidad de pruebas para poder obtener resultados fidedignos. En esta investigación, para ver el mejor comportamiento en el problema, se estudió el cambio de ciertos parámetros, los cuales son  $P_a$  (fracción de nidos abandonados),  $\beta$  (valor entre 1 a 2) que se encuentra en el Lévy Flight y  $n$  (representa el números de nidos en uso).

Se realizó una gran cantidad de experimentos con la finalidad de ver la mejor forma de trabajo que logre Cuckoo Search y, a la vez, cómo se compara con los óptimos que se encuentran en [12], para así comprender qué tan eficiente puede ser esta metaheurística.

## 6.3 Software

Como se explicó anteriormente, el software se creó en Java, por lo que en esta sección se hablará de un par de funciones utilizadas en el prototipo, las cuales no se encuentran explicadas en los puntos de la metaheurística.

A nivel macro, el software para resolver el problema del MCDP trabaja del siguiente modo.

- En primer lugar, se debe indicar los valores de los parámetros iniciales, los cuales son: N número de nidos,  $\beta$  que pertenece al Lévy Flight en un rango (1-2) y  $P_a$ , que es la fracción de nidos abandonados. También se debe indicar el número de máquinas que soporta una celda y el número de celdas, los cuales dependen del ejercicio a estudiar.
- El programa genera “n” nidos llenados a base de una función aleatoria que permite crear un conjunto solución inicial.
- Luego, entra al primer bucle, en el cual se contabiliza la cantidad de veces que se utilizará el Cuckoo Search por cada caso. Aquí se utilizó el número de 100.000 iteraciones.
- El bucle del programa trabaja como está descrito en el punto 6.1.
- En el bucle del Cuckoo Search existen unas funciones que cabe destacar, pues se utilizan para adaptar la metaheurística al Software.
- El resultado obtenido en el Lévy Flight se debe transformar a número entero con el fin de indicar la celda, esto se logra mediante el uso de una función de Binarización que se indican en el punto 7.3.3. El Lévy Flight actúa individualmente en una celda del vector del conjunto solución.
- Al realizar el cambio en el vector solución a base del Lévy Flight, se debe aplicar la función reparadora que comprueba si el vector creado cumple la restricción del  $M_{max}$  y, en caso contrario, esta función repara el vector.
- Al finalizar el programa, se obtienen los resultados en orden.
- La solución se encuentra en el vector máquinas x celdas.

### **6.3.1 Función de Generación inicial de nidos**

Esta función se dedica a generar aleatoriamente conjuntos solución que cumplan con el Vector de máquinas x celdas, se encuentren bien definidos con respecto a sus restricciones del número máximo de máquinas por celdas y, además, que no puede existir una máquina en dos celdas. En términos simples, trabaja generando un valor en cada parte del vector, mientras revisa que se cumpla la restricción del número máximo de máquinas por celdas, hasta que se encuentre la correcta.

Cabe destacar que la función de Generación inicial de nidos posee una fuerte relevancia, ya que debe inicializar los nidos y, cada vez que se aplica el Pa (fracción de nidos abandonados), por cada nido abandonado genera una nueva solución.

### **6.3.2 Función Reparadora**

Como fue mencionado en el punto 7.3, la misión de esta función es, después de realizar el Lévy Flight, comprobar si se encuentra en correcto estado el vector máquinas x celdas según las restricciones del MCDP y, en caso de ser incorrecto, arregla la matriz para que sea viable.

En resumidas cuentas, la función Reparadora sólo se dedica a comprobar que el Mmax esté correcto y, en caso contrario, arregla el vector para que sea viable.

### **6.3.3 Funciones de Binarización**

Otro problema existente en utilizar el Cuckoo Search para resolver el MDCP, es el hecho que el Lévy Flight entrega un valor de carácter real, mientras que el MCDP solo es capaz de soportar valores de 1 y 0. Para solucionar este problema, una vez el Lévy Flight termina su aplicación, se utiliza una función de Binarización para convertir el número real en un valor del tipo 0 o 1.

### 6.3.3.1 Funciones de la forma V

Estas funciones son una modificación a la tangencial hiperbólica. En la presente experimentación, se utilizaron V2, V3 y V4, así escoger la que trabaje de mejor manera con el MCDP.

Dentro de las funciones de la forma V, tenemos una serie de características, pero las más destacables son, en primer lugar, que su dominio sea todos los reales y, segundo, que su imagen esté en el intervalo (-1,1). Por lo tanto, sirve para transformar un número real a 0 y 1, los que necesita el MCDP. Sin embargo, en este trabajo se utiliza la función en valor absoluto, así evitar tener como respuesta un número negativo. La gráfica se halla representada en Fig.2.

Matemáticamente se describe de la siguiente forma.

$$V2 = |\text{Tan}(x)|$$

$$V3 = \left| \frac{x}{\sqrt{1+x^2}} \right|$$

$$V4 = \left| \frac{2}{\pi} \right| \arctan \left( \frac{\pi}{2} x \right)$$

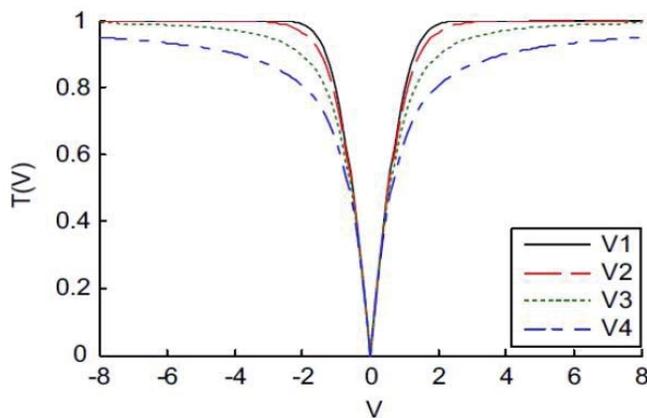


Figura 2: Gráfica de la Forma V [13]

### 6.3.3.2 Función Sigmoide

Esta función se define como una función de variable real diferenciable.

De sus características más relevantes, se desprende que su dominio sea todos los reales y su imagen esté en el intervalo  $[0,1]$ . A diferencia de la función tangente hiperbólica, en la función Sigmoide no es necesario hacer modificaciones, ya que entrega los valores necesarios. Su representación se encuentra en la gráfica *fig2*.

Matemáticamente se describe de la siguiente forma.

$$S(t) = \frac{1}{1 + e^{-t}}$$

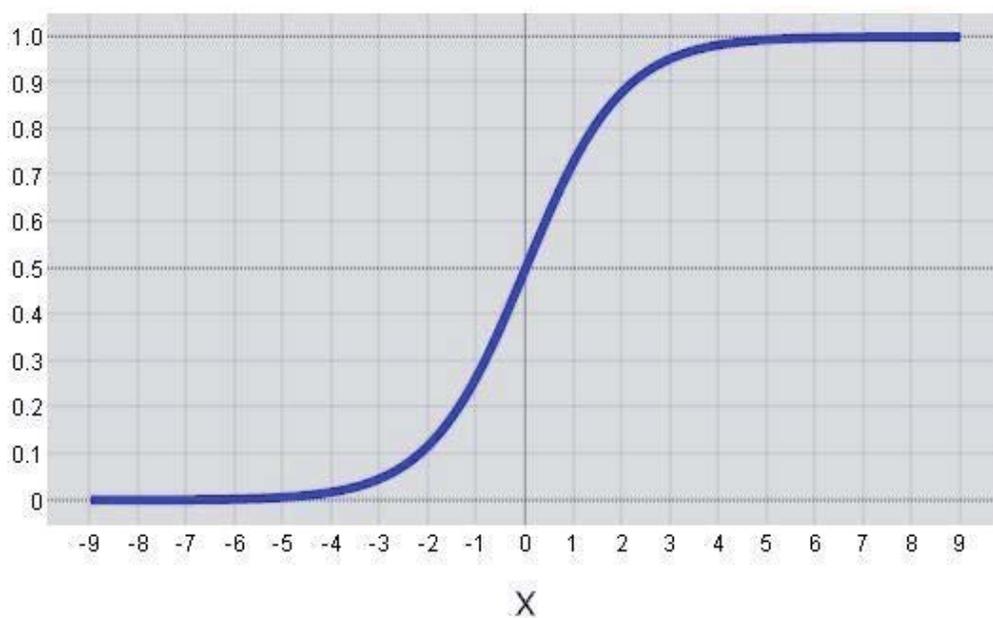


Figura 3: Gráfica de la función Sigmoide

### 6.3.4 Matriz de Similitud

Es única técnica del tipo “Clustering”, esta se refiere a la tarea de clasificar objetos, en el caso específico, como partes o máquinas, las cuales posean algo en común. La técnica de Matriz de Similitud se utiliza en áreas de análisis en general, tales como data mining o reconocimiento de patrones.

Por una medida de emergencia en la búsqueda del óptimo, se utilizó este método como último recurso, el cual reemplaza en un 100% al Lévy Flight, sin embargo, el resto del algoritmo del Cuckoo Search se mantiene intacto. El uso de esta técnica es un cambio bastante severo en el núcleo de este proyecto, por lo tanto puede que se encuentre como una alteración con el potencial de cambiar la naturaleza e, incluso, lograr que el actual Cuckoo Search deje de llamarse así.

Este método en el trabajo actual se describe de la siguiente forma:

#### 6.3.4.1 Creación de las matrices de Similitud

La creación de la matriz de similitud, se hace en base de la matriz inicial usada en el MCDP [máquinas x partes], véase en *tabla1*, con esta se crean las dos matrices de similitud. Utilizando la fórmula  $M * M^T$ , entrega el resultado de la matriz de similitud con respecto a las máquinas (máquinas x máquinas), véase en *tabla2*. Si se utiliza  $M^T * M$ , se obtiene el resultado de la matriz de similitud con respecto a las partes (partes x partes), véase en *tabla3*.

Se debe recalcar que el número en la matriz, se refiere al grado de similitud que hay entre máquinas o partes, mientras mayor sea el número, más partes poseen en común cada máquina.

Para el cálculo de los resultados sólo se utiliza la matriz de similitud con respecto a las máquinas.

Se utilizó un método más simple con respecto a los de [7], para la creación del vector solución, el cual se explicará en pseudo lenguaje.

Para obtener la solución, se utiliza la similitud, mientras mayor sea ésta, existe una mejor posibilidad que las máquinas posean partes en común y, por ende, permanezcan en la misma celda, mientras que a menor similitud, hay un mayor porcentaje que se encuentren en celdas distintas.

**Tabla 1: Ejemplo de Matriz máquinas x partes**

MxP	1	2	3	4	5	6	7
1	1	0	0	1	1	1	0
2	0	0	0	0	1	0	1
3	0	0	0	0	1	0	0
4	1	1	0	1	0	0	1
5	0	0	1	0	0	1	0
6	1	0	1	0	1	1	0

**Tabla 2: Ejemplo de Matriz de similitud respecto a máquinas**

MxM	1	2	3	4	5	6
1	4	1	1	2	1	3
2	1	2	1	1	0	1
3	1	1	1	0	0	1
4	2	1	0	4	0	1
5	1	0	0	0	2	2
6	3	1	1	1	2	4

**Tabla 3: Ejemplo de Matriz de similitud respecto a partes**

PxP	1	2	3	4	5	6	7
1	3	1	1	2	2	2	1
2	1	1	0	1	0	0	1
3	1	0	2	0	1	2	0
4	2	1	0	2	1	1	1
5	2	0	1	1	4	2	1
6	2	0	2	1	2	3	0
7	1	1	0	1	1	0	2

### 6.3.4.1 Algoritmo de la Matriz de Similitud

El algoritmo se representará en pseudo lenguaje en su forma nativa.

```
MxP:   Matriz máquinas x partes
MxM:   Matriz de similitud con respecto a máquinas
Porte: Vector que contiene la sumatoria de la similitud de cada
fila
Tope:  Parámetro de similitud /* su valor debe ser menor o
igual al mayor valor existente en MxM*/
Celdas: número de celdas

Función de Similitud
Se crea la matriz MxM a base de la MxP
Se calcula el porte para cada fila de MxM
For (int i, i<n,i++)
    if(i==0)
        Asignar aleatoriamente una celda a la maquina 1
        Contador++
    End if
    Else if( i>0 && contador < celdas)
        If( Similitud >= Tope) /*Similitud sacada de la
matriz MxM*/
            Se asigna la celda a la máquina
        End If
        Else
            Se asigna una celda no utilizada a la máquina
            Contador++
        End Else
    End if
    Else if(contador >= celdas && i>0) /* en este punto, cada
celda posee alguna máquina*/
        Se aplica aleatoriamente la elección de la siguiente celda
de la máquina, sin embargo este número aleatorio está
alterado por la similitud, mientras mayor sea ésta, es
probable que se asigne una celda con máquinas de una
similitud alta. Se utiliza el porte y la similitud en
MxM para realizar este cálculo.
    End Else
End For
Se obtiene un vector solución
End
```

Figura 4: Algoritmo de la Matriz de Similitud

## 7 Experimentos

Para la realización de este punto, como se mencionó anteriormente, se utilizan los problemas de Boctor [12], este documento posee 10 diferentes ejercicios pertenecientes a MCDP, de los cuales cada uno se harán con cambios de  $M_{max}$  (máximo número de máquinas por celda) y número de celdas.

### 7.1 Características

Parámetros utilizados:

- ❖ Se utilizo  $P_a$  de 0.25
- ❖ Se utilizo 100.000 iteraciones por prueba
- ❖ Se utilizaron  $n = 25$ , o sea 25 nidos.
- ❖ El  $\beta$  que se uso en el Lévy Flight corresponde al valor 1 en las tablas.

Las pruebas realizadas son:

- ❖ 2 celdas, con 8,9,10,11,12  $M_{max}$  respectivamente
- ❖ 3 celdas, con 6,7,8,9  $M_{max}$  respectivamente
- ❖ Cada prueba se realizo 10 veces
- ❖ Esto se repite para cada función de binarización, tanto la función sigmoide, tangencial hiperbólica, V3, V4 y matriz de similitud.

### 7.2 Resultados

Cabe mencionar que los resultados puestos en las siguientes tablas muestran el mejor obtenido de las 10 pruebas hechas en cada punto y sólo se pondrán estos resultados para no sobrecargar de tablas.

En primer lugar se presenta una tabla con los valores óptimos tabla4.

**Tabla 4: Resultados óptimos de los problemas**

P	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
<b>1</b>	11	11	11	11	11	27	18	11	11
<b>2</b>	7	6	4	3	3	7	6	6	6
<b>3</b>	4	4	4	3	1	9	4	4	4
<b>4</b>	14	13	13	13	13	27	18	14	13
<b>5</b>	9	6	6	5	4	11	8	8	6
<b>6</b>	5	3	3	3	2	6	4	4	3
<b>7</b>	7	4	4	4	4	11	5	5	4
<b>8</b>	13	10	10	5	5	14	11	11	10
<b>9</b>	8	8	8	5	5	12	12	8	8
<b>10</b>	8	5	5	5	5	10	8	8	5

Se expondrán las tablas con respecto a cada problema con todos los métodos empleados para resolverlo, mostrando los mejores resultados conseguidos.

**Tabla 5: Resultados del problema 1**

P1	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	39	39	39	39	39	99	74	56	39
TanH	39	39	39	39	39	99	74	56	43
V3	39	39	39	39	39	99	72	57	39
V4	39	39	39	39	39	99	78	39	39
Similarity	39	39	39	39	39	101	74	39	43
Boctor	11	11	11	11	11	27	18	11	11

**Tabla 6: Resultados del problema 2**

P2	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	20	20	20	16	14	30	25	24	20
TanH	20	20	20	16	14	30	25	24	20
V3	20	20	20	16	14	30	25	24	20
V4	20	20	20	16	14	30	25	24	20
Similarity	20	20	20	16	14	30	25	24	20
Boctor	7	6	4	4	3	7	6	6	6

**Tabla 7: Resultados del problema 3**

P3	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	18	14	14	9	5	29	14	18	14
TanH	18	14	14	9	5	29	14	14	14
V3	18	14	14	9	5	29	14	14	14
V4	18	14	14	9	5	29	23	19	18
Similarity	18	14	14	9	5	29	14	14	14
Boctor	4	4	4	3	1	9	4	4	4

**Tabla 8: Resultados del problema 4**

P4	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	58	52	52	43	38	58	69	58	52
TanH	58	52	52	43	38	88	69	58	52
V3	58	52	52	43	38	88	69	58	52
V4	58	52	52	43	38	88	69	58	52
Similarity	58	52	52	43	38	88	69	58	52
Boctor	14	13	13	13	13	27	18	14	13

**Tabla 9: Resultados del problema 5**

P5	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	34	25	24	21	19	43	37	34	24
TanH	34	25	24	21	19	43	37	34	31
V3	34	25	24	21	19	43	37	34	24
V4	34	25	24	21	19	43	37	34	33
Similarity	34	25	24	21	19	43	37	34	35
Boctor	9	6	6	5	4	11	8	8	6

**Tabla 10: Resultados del problema 6**

P6	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	16	13	13	9	8	27	15	15	13
TanH	16	13	13	9	8	27	15	15	13
V3	16	13	13	9	8	27	15	15	13
V4	16	13	13	9	8	27	15	15	13
Similarity	16	13	13	9	8	28	15	15	13
Boctor	5	3	3	3	2	6	4	4	3

**Tabla 11: Resultados del problema 7**

P7	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	28	24	22	22	17	60	27	27	24
TanH	28	24	22	22	17	60	27	27	27
V3	28	24	22	22	17	60	27	27	24
V4	28	24	22	22	17	60	27	28	27
Similarity	28	24	22	22	17	60	27	28	30
Boctor	7	4	4	4	4	11	5	5	4

**Tabla 12: Resultados del problema 8**

P8	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	41	38	30	23	23	57	48	41	38
TanH	41	38	30	23	23	57	48	41	38
V3	41	38	30	23	23	57	48	41	38
V4	41	38	30	23	23	57	48	41	38
Similarity	41	38	30	23	23	57	48	41	38
Boctor	13	10	8	5	5	14	11	11	10

**Tabla 13: Resultados del problema 9**

P9	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	36	36	36	29	23	61	52	36	36
TanH	36	36	36	29	23	61	52	36	36
V3	36	36	36	29	23	61	52	36	36
V4	36	36	36	29	23	61	52	36	36
Similarity	36	36	36	29	23	64	54	46	36
Boctor	8	8	8	5	5	12	12	8	8

**Tabla 14: Resultados del problema 10**

P10	Celda = 2					Celda = 3			
	$M_{\max}=8$	$M_{\max}=9$	$M_{\max}=10$	$M_{\max}=11$	$M_{\max}=12$	$M_{\max}=6$	$M_{\max}=7$	$M_{\max}=8$	$M_{\max}=9$
Sigmoide	29	23	23	23	23	44	34	29	23
TanH	29	23	23	23	23	44	34	29	23
V3	29	23	23	23	23	44	34	36	29
V4	29	23	23	23	23	44	34	29	29
Similarity	29	23	23	23	23	44	40	29	23
Boctor	8	5	5	5	5	10	8	8	5

### 7.3 Análisis de resultados

Al terminar estos experimentos, estudiados de forma macro, se puede ver que los resultados obtenidos son pésimos, en general bastante lejos de los óptimos, especialmente en los casos de tres celdas. Sin embargo en el proceso de dos celdas, los resultados son más cercanos a los ideales.

Al examinar de forma más precisa también existen otros detalles bastantes importantes, los cuales se pueden apreciar al ver las 10 pruebas por caso. Respecto a los casos con dos celdas, se comportaron casi todos de la misma manera, obteniendo resultados prácticamente iguales en cada método utilizado. Esto se debe principalmente a que al ocupar dos celdas, existen menos combinaciones, por lo cual logran converger a cierto punto, pero no es el óptimo.

En el caso de los experimentos empleando tres celdas, los resultados en general de cada problema fueron aleatorios, llegando tan solo unos pocos casos a un punto de convergencia, sin embargo si se creaban pruebas de mayor número de iteraciones, lograba llegar una mayor cantidad de casos al punto de convergencia, en general estos son los mejores valores obtenidos, se hicieron varias pruebas con 1.000.000 iteraciones y cambio de Pa, pero no se lograban mejores resultados.

Estudiando la cantidad de pasos, se puede obtener que el problema 3, entregó los resultados más cercanos al óptimo, el mejor caso encontrado en la celda=2 con Mmax = 12, solo existen 4 movimientos de diferencia al resultado ideal, su peor caso fue el correspondiente a la celda=3 y Mmax=6 con 20 movimientos de diferencia, lo cuales ven bastante favorables en comparación al resto de los problemas.

Otro comportamiento particular encontrado fue que el caso más cercano al óptimo lograba un resultado en relación 1: 2,92, que se encuentra en el problema 4, con celda =2 y Mmax =12, en general la relación del resto de los resultados obtenidos se encuentran entre 3 hasta 6 veces el óptimo, lo que vuelve a verificar los pésimos resultados, además podría confirmar que el mejor resultado (con respecto a cantidad de movimientos) obtenido en el problema 3, visto mediante relación es 1: 3, por ende se convertiría en un resultado ineficiente mirado bajo esta perspectiva.

## 8 Conclusiones

Verificando los objetivos de este proyecto, se logró lo principal, que es la adaptación del Cuckoo Search en el MCDP, sin embargo, los resultados obtenidos son bastantes lejanos del óptimo de cada problema utilizado. En los próximos párrafos se explicará de manera más profunda la problemática.

Se debe tomar en cuenta que para lograr adaptar esta metaheurística, se deben crear grandes cambios para ajustarlos a un problema con tantas restricciones como lo es el MCDP. En este caso, el punto más difícil de acondicionar fue el Lévy Flight. El MDCP es una función que solo acepta 0 y 1, por lo cual se adapta al Lévy Flight mediante una función de binarización. En un principio, con el nuevo cambio de la matriz  $M \times C$  a vector, Lévy Flight se debe pasar por intervalos en caso de que se utilicen tres celdas. Esto impide que el Lévy Flight trabaje libremente como suele hacerlo, pudiendo ser uno de los factores que afectó los resultados obtenidos.

En un principio del proyecto, se centro en la diversificación por parte del Lévy Flight, donde afectaba en toda la solución de la matriz  $M \times C$ , más adelante se hizo un énfasis en la intensificación por parte del Lévy Flight, el cual se aplico actuando sobre una columna de la matriz  $M \times C$ , en vez de toda la solución, sin embargo los resultados se mantuvieron muy parecidos entre ambos métodos, con la diferencia de la ejecución del programa logró una mejora en la velocidad alrededor de un 10-15%.

Al convertir la matriz de  $M \times C$  a vector, los tiempos mejoraron drásticamente, alrededor de una disminución entre 50-75%, sin embargo, los resultados se mantuvieron sin presentar mejora, es decir, fue netamente una optimización en el área de ejecución del programa.

La función Sigmoide casi solo arroja valores 1. En caso de utilizar tres celdas, habría un pequeño porcentaje de dar dos valores distintos y claramente el 0 no es uno. Por lo cual, la función no trabaja de la manera esperada, lo que sería equivalente a un caso donde se sobrecarga la función reparadora. Por esta razón, fueron implementadas una larga variedad de funciones de binarización, entre estas se utilizó variantes de la forma "V". Aun así no se lograron grandes cambios en los resultados.

Se utilizó la matriz de similitud principalmente por la necesidad de buscar nuevos métodos para lograr llegar a mejores resultados, sin embargo, este método no entregó mejores resultados e, incluso, en algunos casos se obtienen resultados inferiores al resto de las funciones. También se debe acentuar que los tiempos aumentaron drásticamente, alrededor de un 20-35% más lento que los métodos anteriores.

## 9 Referencias

- [1]Aljaber N., Baek W., Chen C. *A tabu search approach to the cell formation Problem*, Computers & Industrial Engineering, vol. 32(1),pp. 169–185, (1997).
- [2]Lozano S., Díaz A., Eguía I., Onieva L., *A one-step tabu search algorithm for manufacturing cell design*, Journal of the Operational Research Society, vol. 50(5), (1999).
- [3]Soto R., Paredes F., Kjellerstrand H., Durán O., Crawford B., Monfroy E., *Cell formation in group technology using constraint programming and Boolean satisfiability*, Expert Systems with Applications, vol. 39, pp. 11423–11427, (2012).
- [4]Wu T., Chang C., Chung S., *A simulated annealing algorithm for manufacturing cell formation problems*, Expert Systems with Applications, vol. 34(3), pp. 1609–1617, (2008).
- [5]Xin-She Yang. *Cuckoo Search via Lévy Flights*, December 2009, India. IEEE Publications, USA, pp. 210-214 (2009).
- [6]Barber F., Salido M., *Introducción a la programación de Restricciones*, Revista Iberoamericana de Inteligencia Artificial, vol. 20, pp. 13-30 (2003).
- [7]S. Oliviera, J.F.F. Ribeiro, S.C.Seok, *A spectral clustering algorithm for manufacturing cell formation*, Computers & Industrial Engineering, vol. 57, pp 1008-1014, (2009).
- [8]Durán O., Rodriguez N., Consalter L., *Collaborative particle swarm optimization with a data mining technique for manufacturing cell design*, Expert Systems with Applications, vol. 37(2), pp. 1563–1567, (2010).
- [9]Shafer S., Rogers D, *A goal programming approach to cell formation Problems*, Journal of Operations Management, vol. 10, pp. 28–34, (1991).
- [10]Yang X. S., *Nature-Inspired Metaheuristic Algorithms*, 2nd Edition, Luniver Press, pp 11-19, (2010).
- [11][http://www.mathworks.co.uk/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm/content/cuckoo\\_search\\_new.m](http://www.mathworks.co.uk/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm/content/cuckoo_search_new.m).
- [12]Fayez F. Boctor, *A linear formulation of the machine-part cell formation problem*, International Journal of Production Research, vol. 29, n° 2, pp. 343-356, (1991).
- [13]Crawford B., Soto R., Olivares-Suárez M., Palma W., Paredes F., Castro C., *Transfer Functions for a Binary Firey Algorithm that Solves the Set Covering Problem*, Preprint submitted to Mathematical Problems in Engineering, (2014).

- [14]Venugopal V., Narendran T. T., *A genetic algorithm approach to the machine-component grouping problem with multiple objectives*, Computers and Industrial Engineering, vol. 22(4), pp. 469–480, (1992).
- [15]Gupta Y., Gupta M., Kumar A., Sundaram C., *A genetic algorithm-based approach to cell composition and layout design problems*, International Journal of Production Research, vol. 34(2), pp. 447–482, (1996).
- [16]Purcheck G., *A linear-programming method for the combinatorial grouping of an incomplete set*, Journal of Cybernetics, vol. 5, pp. 51–58, (1975).
- [17]Olivia-Lopez E., Purcheck G., *Load balancing for group technology planning and control*, International Journal of MTDR, vol. 19, pp. 259–268, (1979).
- [18]Kusiak A., Chow W., *Efficient solving of the group technology problem*, Journal of Manufacturing Systems, vol. 6, pp. 117–124, (1987).
- [19]Sankaran S., *Multiple objective decision making approach to cell formation: A goal programming model*, Mathematical Computer Modeling, vol. 13, pp. 71–77, (1990).

# **Anexos**



**Tabla a.3: Matriz MxP del problema 3**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3					
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0					
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0		
2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1		
3	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0		
4	1	0	1	0	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0		
5	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0		
6	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0		
7	1	0	1	0	0	0	1	0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0		
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	
10	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
11	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
12	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	1	0	0

**Tabla a.4: Matriz MxP del problema 4**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3					
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0					
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	0	0	0		
2	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	
3	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	
6	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	
7	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	
10	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	
11	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
12	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	
13	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	0
15	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

**Tabla a.5: Matriz MxP del problema 5**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0
3	0	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
6	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	1	0	1	
8	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
9	0	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	1	
14	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	
16	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	1	0	1	0	

**Tabla a.6: Matriz MxP del problema 6**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
1	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0
5	1	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0
7	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
8	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0
10	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	1	0	0	1	0	0	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0
13	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
15	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0
16	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0

**Tabla a.7: Matriz MxP del problema 7**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1
3	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	
5	1	1	0	0	0	1	1	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	
10	1	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	
11	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	1
15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	
16	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	

**Tabla a.8: Matriz MxP del problema 8**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	1	1	1	
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	0	
3	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	
4	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	0	1	
6	1	0	0	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	
8	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	
10	0	0	0	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	
12	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
14	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	
15	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	

**Tabla a.9: Matriz MxP del problema 9**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3		
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0		
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0		
2	0	1	1	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	1	1	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	
4	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	
7	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	
8	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	
10	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	
11	1	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	
12	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	0	
14	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	
15	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	

**Tabla a.10: Matriz MxP del problema 10**

	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
1	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	0	0
5	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1
6	1	0	1	1	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
8	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
9	1	1	1	1	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	0	1	1	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
12	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	1	0	0	1	0	1	0	1	0	0	0
15	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0