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

RESOLUCIÓN DEL MANUFACTURING CELL DESIGN PROBLEM CON AFRICAN BUFFALO OPTIMIZATION

ALBERTO FELIPE CABRERA MORALES

Profesor Guía : Ricardo Soto de Giorgis
Co-referente : Boris Almonacid Gutiérrez

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

JUNIO, 2017

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

RESOLUCIÓN DEL MANUFACTURING CELL DESIGN PROBLEM CON AFRICAN BUFFALO OPTIMIZATION

ALBERTO FELIPE CABRERA MORALES

Profesor Guía **: Ricardo Soto de Giorgis**
Co-referente **: Boris Almonacid Gutiérrez**

JUNIO, 2017

Dedicatoria

A mi familia, por su apoyo incondicional
y motivación para alcanzar mis objetivos.

A mi profesor guía y co-referente, por confiar
en mis capacidades y por toda la ayuda brindada.

A mis amigos y compañeros por
toda su confianza y cercanía.

A todos ellos muchas gracias.

Resumen

El propósito de esta investigación consiste en resolver el problema de diseño de celdas de manufactura (MCDP). Este problema se enfoca en la división de una planta de producción formada por celdas para agrupar máquinas que procesan distintas piezas. La idea es crear un diseño óptimo de producción, en el cual se minimice el traslado de piezas entre celdas, con el objetivo de reducir costos y aumentar la productividad. Para completar esta tarea, se utilizará la metaheurística African Buffalo Optimization, la cual se basa en el comportamiento del búfalo africano a la hora de buscar comida en bosques y sabanas. Se realizaron pruebas experimentales en los noventa problemas de Boctor, en los cuales se alcanzaron todos los óptimos globales. Adicionalmente se ejecutaron setenta nuevas instancias para comparar nuevos resultados.

Palabras Claves: Problema de diseño para celdas de manufactura, Problema de optimización, Metaheurística, African Buffalo Optimization, Boctor.

Abstract

The purpose of this research is to solve the Manufacturing Cell Design Problem (MCDP). This problem focuses on the division of a production plant formed by cells to group machines that process different pieces. The idea is to create an optimal production design, in which the transfer of pieces between cells is minimized, with the aim of reducing costs and increase productivity. To complete this task, the African Buffalo Optimization metaheuristic will be used, which is based on the behavior of the African Buffalo when they are looking for food in forests and savannas. Experimental tests were performed on the ninety problems of Boctor, in which all the global optimums were reached. Also, seventy new instances were executed to compare new results.

Key Words: Manufacturing Cell Design Problem, Optimization Problem, Metaheuristic, African Buffalo Optimization, Boctor.

Índice

Lista de Figuras

Lista de Tablas

1 Introducción

Actualmente, las empresas de manufactura se encuentran en una constante búsqueda de nuevas formas o procesos para el desarrollo o creación de sus productos. Por lo tanto surge la necesidad de optimizar sus métodos. De tal manera que se pueda lograr llegar al mayor rendimiento posible, con el fin de reducir los costos al mínimo y poder aumentar los beneficios.

Una de las metodologías para poder resolver este problema de optimización es el Manufacturing Cell Design Problem (MCDP). La cual es una metodología que representa las máquinas y las partes en una fábrica de manufactura, en una matriz de incidencia. En esta matriz se indican las partes que serán procesadas por las distintas máquinas. Sin embargo, esta matriz de incidencia no considera separar las máquinas y las partes en conjuntos de celdas de manufactura. Por lo que se deben de agrupar y asignar en una matriz que sí considera esta organización.

Existen diferentes combinaciones que pueden solucionar este problema de optimización, sin embargo, mientras aumenta el número de máquinas, de partes y de celdas, encontrar una solución óptima conlleva un alto costo computacional. Por lo tanto, para esta investigación se estudiará la metaheurística African Buffalo Optimization (ABO). La expresión "metaheurística" deriva del griego "meta", que significa ir más allá, ir a un nivel superior, y "heurística", se refiere a encontrar. Por definición, es un método de búsqueda para resolver un problema computacional.

La metaheurística ABO, como herramienta de optimización proporciona un procedimiento de búsqueda perteneciente a la inteligencia de enjambres, basado en el comportamiento social de los animales. La metaheurística ABO se inspira en los búfalos africanos que tienen como objetivo el lograr una explotación y exploración del espacio de búsqueda, donde estos espacios de soluciones son llamados lugares de comida.

Esta investigación se basa en la resolución del MCDP con la utilización de la metaheurística ABO. Primordialmente, se buscará un diseño con el cual se pueda adaptar la metaheurística ABO desde un dominio real a un dominio discreto. Esto, debido a que ABO es una metaheurística diseñada para problemas reales, en cambio, el MCDP es un problema discreto, específicamente binario.

Las secciones de este documento están divididas de la siguiente forma: en el capítulo 2, se especifican los objetivos generales y específicos del proyecto. En el capítulo 3 se mostrarán una serie de investigaciones previas al MCDP mediante un Estado del Arte. Posteriormente, en el capítulo 4 se realiza una descripción detallada de la problemática mencionada en un principio, definiendo su modelo matemático y restricciones que afectarán la búsqueda de soluciones. En los capítulos 5 y 6 se presenta la metaheurística escogida con todos los pasos que se realizaron para solucionar el MCDP. En el capítulo 7 se exponen los resultados obtenidos y finalmente en el capítulo 8, se presentan las conclusiones de la investigación realizada.

2 Definición de Objetivos

2.1 Objetivo General

- El objetivo principal de esta investigación es resolver el MCDP mediante la utilización del algoritmo metaheurístico African Buffalo Optimization.

2.2 Objetivos Específicos

- Comprender el MCDP y la metaheurística African Buffalo Optimization.

- Integrar el ABO para resolver el MCDP.

- Obtener soluciones para el problema planteado y evaluarlos en las distintas instancias de MCDP.

3 Estado del Arte

Dentro de las investigaciones que se han enfocado en resolver el MCDP se pueden dividir en dos grupos: métodos de aproximación y de optimización global. Los métodos de aproximación contienen a las metaheurísticas, en las cuales se aplican inteligentemente diversas técnicas para explorar el espacio de soluciones. Existen muchas metaheurísticas, pero todas ellas tratan en mayor o menor medida la intensificación en la búsqueda (explotación del área actual para mejores soluciones) y la diversificación (explorar nuevas áreas, con el fin de encontrar nuevas posibles soluciones, las cuales no siempre alcanzan el óptimo). Por el contrario, las técnicas de optimización global, se dedican a analizar detenidamente todo el espacio de búsqueda con el fin de encontrar el óptimo al problema, lo que provoca un gasto mayor en recursos (tiempo, trabajo, memoria, dinero, entre otros).

Respecto al primer grupo que refiere a las metaheurísticas, se nombrarán a continuación aquellas usadas para resolver el MCDP: Aljaber, Baek, Chen [1] y Lozano, Díaz Eguía, Onieva [2] con el uso del Tabu Search. Wu, Chang, Chung [3] utilizando el Simulated Anneling (SA), Soto, Crawford, Almonacid, Paredes [4] usando el Migrating Bird Optimization Algorithm. Soto R., Crawford B., Vega E., Johnson F., Paredes F. [5], utilizan el Shuffled Frog Leaping Algorithm, Venusgopal, Narendran [6] proponen usar Genetics Algorithms (GA), luego Gupta, Gupta, Kumar, Sundaram [7] utilizan la misma solución representada con los GA, pero con una optimización diferente multi-objetivo la cual se dirige a la simultánea minimización del total del número entre células con variación de carga celular. Soto, Crawford, Vega, Paredes [8] usan el Artificial Fish Swarm Algorithm.

En segundo lugar se encuentra el uso de programación lineal por parte de Purcheck [9], Olivia-Lopez y Purcheck [10], modelos cuadráticos propuestos por Boctor [11] y Kusiak, Chow [12]. La Programación por metas denominadas Goal Programming (GP) es otro paradigma de este grupo cuya función es la toma de decisiones con objetivos múltiples, Sankaran [13], Shafter y Roger [14] propusieron modelos sobre este paradigma.

Para el desarrollo de este proyecto en particular, se utilizó la metaheurística African Buffalo Optimization creada por Odili, Kahar, Anwar [15]. Como la gran mayoría de metaheurísticas, ABO propone diversificar, buscando un gran número de soluciones, así como también intensificar el área en explotación actual para encontrar el óptimo del problema.

4 Manufacturing Cell Design Problem

La estrategia de producción del MCDP consiste en la división de una planta de manufactura en diversas celdas de producción. En donde cada celda posee un grupo de máquinas, las cuales se encargan de procesar un conjunto de piezas o partes similares de un producto. Estas partes son organizadas por su similitud (características, volumen, forma, entre otras características).

El MCDP busca posicionar las máquinas de tal forma que las piezas o partes del producto se muevan dentro de una celda, minimizando el flujo de estas entre celdas. De esta forma se organizaría la planta de producción para reducir costos de producción, tiempos del traslado del producto y consecuentemente, aumentar la productividad.

Para lograr una organización efectiva del sistema de producción es necesario conocer los caminos que recorren todas las piezas para la fabricación de un producto específico. De esta forma, se podrá determinar cuáles máquinas son utilizadas por cada una de las partes del producto. Para desarrollar lo antes mencionado, se hará uso de una matriz binaria, donde las máquinas representan a las filas y las partes a las columnas. Cada intersección de máquina-pieza será un 1 y por el contrario un 0 si no hay relación.

En la figura 1 se muestran esas posiciones *(i,j)* dentro de la matriz ejemplificando la explicación anterior.

Pieza / **Máquina**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	0	1	0	0	1	0	0
2	1	0	0	0	0	0	1
3	0	0	1	1	0	1	0
4	0	0	1	1	0	1	0
5	1	0	0	0	0	0	1
6	0	0	1	1	0	1	0
7	0	1	0	0	1	0	0

Figura 1. Ejemplo de Matriz Máquina-Parte

A partir de la figura 1, se puede diferenciar que piezas hacen utilización de cierta máquina. Como el propósito del MCDP es organizar estas piezas de manera que como se aprecia en la figura 2, el movimiento de estas entre celdas sea el mínimo posible.

Pieza / Máquina	1	7	3	4	6	2	5
2	**1**	**1**	0	0	0	0	0
5	**1**	**1**	0	0	0	0	0
3	0	0	**1**	**1**	**1**	0	0
4	0	0	**1**	**1**	**1**	0	0
6	0	0	**1**	**1**	**1**	0	0
1	0	0	0	0	0	**1**	**1**
7	0	0	0	0	0	**1**	**1**

Figura 2. Ejemplo Matriz Máquina-Parte Ordenada.

Para esto, es necesario generar un procedimiento lógico, en el cual se permita transformar la matriz de la figura 1 a la matriz de la figura 2. En esta última se puede observar una diagonal formada meramente por unos y otros campos formados por ceros. De esta manera, se conforman las celdas. Los números marcados en negrita nos muestran las 3 celdas independientes existentes para este problema.

4.1 El Modelo Matemático MCDP

El MCDP es considerado un problema de restricciones que debe ser optimizada. El MCDP se puede modelar matemáticamente como un modelo que posee, parámetros, variables y dominios.

A continuación se describe el modelo matemático:

- Parámetros:
  - M, representa el número de máquinas.
  - P, representa el número de partes o piezas.
  - C, representa el número de celdas.
  - $A = [a_{ij}]$ M x P, Matriz de las máquinas x piezas.
  - *Mmax*, es el número máximo de máquinas por celda.

- Índices:
  - i, es el índice de las máquinas ($i = 1,. . . ., M$).
  - j, es el índice de las partes ($j = 1,. . . ., P$).
  - k, es el índice de las celdas ($k = 1,. . . .,C$).

- Variables:
  - $Y = [y_{ik}]$ M x C, Matriz de las máquinas x celdas.
  - $Z = [z_{jk}]$ P x C, Matriz de piezas x celdas.

Para la Matriz *A*, se tiene el siguiente modelo:

$$[a_{ij}] = \begin{cases} 1, & si\ la\ máquina\ i\ produce\ la\ parte\ j \\ 0, & en\ otro\ caso \end{cases}$$

Para la Matriz *Y*, se tiene:

$$[y_{ik}] = \begin{cases} 1, & si\ la\ máquina\ i\ pertenece\ a\ la\ celda\ k \\ 0, & en\ otro\ caso \end{cases}$$

Para la Matriz *Z*, se tiene:

$$[z_{jk}] = \begin{cases} 1, & si\ la\ parte\ j\ pertenece\ a\ la\ familia\ k \\ 0, & en\ otro\ caso \end{cases}$$

Luego tenemos la función objetivo del problema. Esta es una función de minimización, la cual busca reducir la cantidad de veces que se mueve una pieza entre celdas.

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

Este modelo también posee algunas restricciones, las cuales serán presentadas a continuación:

- Una máquina pertenece sólo a una celda:

$$\sum_{k=1}^{C} y_{ik} = 1 \quad \forall\, i$$

- Una parte pertenece sólo a una familia de componentes:

$$\sum_{k=1}^{C} z_{jk} = 1 \quad \forall\, j$$

- El número de máquinas no exceda el número máximo de máquinas por celda:

$$\sum_{i=1}^{M} y_{ik} \leq M_{max} \quad \forall\, k$$

5 African Buffalo Optimization

Esta metaheurística pertenece al grupo de los algoritmos inspirados en la naturaleza, específicamente al comportamiento de enjambres. El tipo de método utilizado en la metaheurística ABO es adaptable, ya que dependiendo del ambiente en que se encuentre, se irá adecuando a las circunstancias para sobrevivir. La metaheurística ABO fue creada por Odili, Kahar y Anwar en 2015, inspirado en el comportamiento del búfalo africano a la hora de buscar comida en bosques y sabanas de África. La migración de estos búfalos es gatillada por la búsqueda de pastos exuberantes, ellos tienden a seguir los movimientos de las estaciones de lluvia cuando pueden obtener estas pasturas. Ya que las estaciones varían de un lugar a otro, los búfalos siempre están en movimiento, buscando nuevas fuentes de comida.

Como en la mayoría de las metaheurísticas, las dos características más importantes dentro de ABO son la intensificación y diversificación. La intensificación en ABO, refiere a la explotación del área actual en que está situado el búfalo, en cambio, la diversificación, se refiere a la exploración de nuevas áreas, es decir, buscar nuevos lugares de comida.

En la metaheurística ABO, los búfalos se mueven en espacios de búsqueda multidimensionales. Cada búfalo busca el óptimo global y actualiza su posición a medida que sigue al mejor búfalo del rebaño. Este seguimiento de sus coordenadas se asocia a la mejor solución (Fitness) que ha conseguido hasta el momento. Este valor representa la mejor ubicación de los búfalos en particular, en relación con la solución óptima. Es por esto, que los búfalos hacen un seguimiento de la ubicación dinámica de cada búfalo hacia el Fitness.

Nuestro interés se centra en la capacidad de organización que poseen los búfalos a la hora de buscar espacios de soluciones, hay 3 factores que influyen en la forma de actuar de los búfalos:

- Su memoria de ubicaciones pasadas (no repetir iteraciones, búsqueda de soluciones evitando zonas con malos resultados).
- Cuidar y cooperar con los demás búfalos para poder intercambiar información.
- La inteligencia de los búfalos.

Respecto a la inteligencia de los búfalos podemos mencionar dos modos básicos de comunicación:

- El sonido de alerta "*Maaa*", instando a los búfalos para permanecer en explotar los recursos disponibles.

- El sonido de alarma "*Waaa*", utilizado por los búfalos para indicar la presencia de peligro o la falta de buenos campos de pastoreo, impulsando a la manada a explorar otros entornos o áreas ya explotadas o no.

5.1 El Algoritmo African Buffalo Optimization

El algoritmo correspondiente a African Buffalo Optimization modela las 3 principales características del búfalo africano, memoria, comunicación e inteligencia.

El inicio del algoritmo se representa de la siguiente forma:

1) Inicializar: Corresponde a colocar aleatoriamente los búfalos en nodos dentro del espacio de solución.

2) Escoger el mejor fitness de la manada.

3) Generar soluciones vecinas, utilizando la siguiente fórmula:

$$m.k + 1 = m.k + lp1(bgmax - w.k) + \ lp2(bpmax.k - w.k)$$

Dentro de la fórmula *w.k* hace referencia al sonido "*Waaa*", donde k representa a cada búfalo de la manada (k=1, 2, 3 … n), expresando peligro o incitando a la manada a explorar nuevas pasturas. Luego, *m.k* referencia al sonido "*Maaa*" el cual expresa la explotación del área local (Pastos en los cuales se encuentran parados los búfalos). Los factores de aprendizaje estarán representados por *lp1* y *lp2*, el búfalo con el mejor pasto de la manada es *bgmax* y por último, el mejor pasto encontrado por el búfalo k denotará a *bpmax.k*.

La ecuación posee 3 partes principales:

- La parte (*m.k* + 1) que indica que los animales están conscientes de que han sido reubicados desde su posición inicial a una nueva. (Indicando así, su gran capacidad para recordar posiciones anteriores).

- *lp1*(*bgmax* – *w.k*) indica la cooperación que poseen los búfalos. Con esto, son capaces de realizar un seguimiento de la ubicación del mejor búfalo en cada iteración.

- *lp2*(*bgmax.k* – *w.k*) indica la inteligencia de los búfalos. Así tienen la capacidad de decirle a los demás búfalos sobre sus mejores anteriores lugares de comida en comparación con los actuales.

Continuando con el algoritmo, ahora se debe actualizar la localización.

4) Actualizar la localización del búfalo k (*bpmax* y *bgmax*) usando la siguiente fórmula:

$$w.k + 1 = \frac{(w.k + m.k)}{\lambda}$$

Esta ecuación posee 3 partes principales:

- La parte (*w.k* + 1), que indica que los búfalos desde su posición (de exploración), serán localizados en una nueva.

- La parte (*w.k* + *m.k*), que es una suma entre la posición actual de exploración, con la posición anterior que poseía el búfalo.

- Y por último tenemos el parámetro Lambda (λ), que es utilizado para valores dentro del intervalo ± 0.5.

Siguiendo con el algoritmo, ahora se debe evaluar *bgmax*.

5) Escoger el mejor Fitness de las soluciones vecinas y comparar con *bgmax*.

6) Se actualizo *bgmax*? Si → avanzar a 7, No → retroceder a 3.

7) Si no se cumplen los criterios al parar la iteración, volver a 4, sino avanzar a 8.

8) Mostrar la mejor solución.

5.2 Pseudocódigo ABO

A continuación se describe el pseudocódigo que representa el comportamiento de la metaheurística.

Función Objetivo f(x), x= $(x_{1,}\ x_{2,}.., x_d)T$

1. Inicializar: Colocar aleatoriamente los búfalos en nodos dentro del espacio de solución.

2. Actualizar el valor del Fitness de los búfalos utilizando;

$$m.k + 1 = m.k + lp1(bgmax - w.k) + \ lp2(bpmax.k - w.k)$$

3. Actualizar la localización del búfalo *k*(*bpmax and bgmax*) usando;

$$w.k + 1 = \frac{(w.k + m.k)}{\lambda}$$

4. *bgmax* ha sido actualizado? Sí, avanzar a 5. No, retroceder a 2.

5. Si no se cumplen los criterios al parar la iteración, volver a 3, sino avanzar a 6.

6. Mostrar la mejor solución.

6 Integración de ABO para resolver MCDP

Para esta implementación de la metaheurística ABO para resolver el MCDP, lo primero será generar una población inicial de búfalos. Básicamente, se crea un arreglo que representa el conjunto inicial de soluciones de la metaheurística; el arreglo en sí, será la manada de búfalos.

Para generar cada solución inicial, se obtiene la matriz *A*, *M*, *P*, *C* y *Mmax* (constantes del problema MCDP). Los elementos que componen una solución son, la matriz *Y*, la matriz *Z* y su Fitness respectivo. Se comienza llenando la matriz binaria *Y* (Máquina-Celda) aleatoriamente. Luego corroborando que la matriz *Y* cumpla con las restricciones del MCDP, se hace un procedimiento lógico (mover filas y columnas), para luego obtener un nuevo orden de las partes y así representar la matriz Z (Parte-Celda). Luego para su Fitness, se verá cuántos unos quedaron fuera de las celdas (llamadas excepciones), se sumarán y se sabrá en números reales, cuál es su valor. Es así como se obtienen los elementos necesarios para representar las soluciones iniciales.

Luego se comparará cada Fitness de la manada, para saber cuál es mejor. Este búfalo con el mejor Fitness será almacenado en una matriz denominada *bgmax*.

Para generar nuevas soluciones, el algoritmo nos dice que los búfalos deben moverse. Este movimiento lo haremos respecto a las matrices que posee cada búfalo (*Y* y *Z*). Tomaremos cada fila de la matriz Máquina-Celda y la llenaremos con ceros, luego aplicaremos la fórmula que actualiza el fitness de los búfalos, en cada índice de estas filas, teniendo como resultado distintas filas de números reales. Se continuará con la matriz Parte-Celda, haciendo exactamente lo mismo. Posteriormente, se escogerá el mayor valor de la fila y se le asignará un 1, para el resto un 0. Luego se revisará el *Mmax* (el máximo de máquinas por celda), para corroborar que se cumplan las restricciones. En caso de que no se cumplan, se procederá a reparar la matriz, seleccionando la celda con menos máquinas y haciendo el cambio de valores en sus índices.

Con esto habremos generado soluciones vecinas, con nuevos fitness. Al terminar este proceso, actualizaremos la localización de los búfalos con la fórmula que refiere al movimiento de estos, en cada iteración la matriz de solución *w.k* que hace referencia a la exploración, será cambiada según el *m.k* (matriz del búfalo_k, que hace referencia a la explotación) y el valor de Lamba.

Siguiendo la metodología del ABO, escogeremos el mejor Fitness de las soluciones vecinas y lo compararemos con *bgmax*.

Si es que *bgmax* se actualizó, significa que encontramos una mejor solución global y avanzaremos dentro del algoritmo. En caso contrario, deberemos volver a generar nuevas soluciones vecinas, siguiendo con la búsqueda del óptimo. Si aún no se cumplen el número de iteraciones definidas, se deberá actualizar nuevamente la localización de los búfalos y en caso contrario de que el número de iteraciones definidas ya se cumplió, se deberá mostrar cual es el mejor Fitness que se pudo conseguir realizando estos ciclos.

6.1 Pasos para integrar el problema MCDP en la metaheurística ABO

A continuación se describe el pseudocódigo con los pasos realizados para resolver una instancia del MCDP utilizando ABO.

Función Objetivo f(x), x= $(x_{1,}\, x_{2,}.., x_d)T$

1. Inicializar: Colocar aleatoriamente los búfalos en nodos dentro del espacio de búsqueda

2. Escoger el mejor Fitness de la población

3. Generar soluciones vecinas

$$\begin{array}{ccc} & C1 & C2 \\ M1 & 0 & 0 \\ M2 & 0 & 0 \\ M3 & 0 & 0 \end{array} \rightarrow m.k+1 = m.k + lp1(bgmax - w.k) + \; lp2(bpmax.k - w.k) \rightarrow$$

$$\begin{array}{ccc} & C1 & C2 \\ M1 & 8 & -3 \\ M2 & 4 & 5 \\ M3 & 6 & 7 \end{array}$$

Se aplica la fórmula que actualiza el Fitness de los búfalos, en cada índice de estas filas, teniendo como resultado una matriz de números reales.

Escoger el mayor valor de la fila y asignarle un 1, para el resto 0

$$\begin{array}{ccc} & C1 & C2 \\ M1 & 8 & -3 \\ M2 & 4 & 5 \\ M3 & 6 & 7 \end{array} \longrightarrow \begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 0 \\ M2 & 0 & 1 \\ M3 & 0 & 1 \end{array} \longrightarrow \text{Revisar Restricciones (Mmax)}$$

Si no cumple con las restricciones se reparará la matriz, escogiendo la celda con menos máquinas, haciendo un cambio de valores en esa fila

$$\begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 0 \\ \mathbf{M2} & \mathbf{0} & \mathbf{1} \\ M3 & 0 & 1 \end{array} \longrightarrow \begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 0 \\ \mathbf{M2} & \mathbf{1} & \mathbf{0} \\ M3 & 0 & 1 \end{array}$$

4. Actualizar la localización de los búfalos $\begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 0 \\ M2 & 1 & 0 \\ M3 & 0 & 1 \end{array} \longrightarrow w.k+1 = \frac{(w.k+m.k)}{\lambda} \longrightarrow \begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 2 \\ M2 & -3 & 4 \\ M3 & 8 & 7 \end{array}$
Escoger el mayor valor de la fila y asignarle un 1, para el resto 0 $\begin{array}{ccc} & C1 & C2 \\ M1 & 1 & 2 \\ M2 & -3 & 4 \\ M3 & 8 & 7 \end{array} \longrightarrow \begin{array}{ccc} & C1 & C2 \\ M1 & 0 & 1 \\ M2 & 0 & 1 \\ M3 & 1 & 0 \end{array} \longrightarrow$ Revisar Restricciones (*Mmax*) Si no cumple con las restricciones se reparará la matriz, escogiendo la celda con menos máquinas, haciendo un cambio de valores en esa fila $\begin{array}{ccc} & C1 & C2 \\ M1 & 0 & 1 \\ \boldsymbol{M2} & \mathbf{0} & \mathbf{1} \\ M3 & 1 & 0 \end{array} \longrightarrow \begin{array}{ccc} & C1 & C2 \\ M1 & 0 & 1 \\ \boldsymbol{M2} & \mathbf{1} & \mathbf{0} \\ M3 & 1 & 0 \end{array}$
5. Escoger el mejor Fitness de las soluciones vecinas y comparar con *bgmax*
6. Se actualizo bgmax? Si, avanzar a 7. No, volver a 3.
7. Si no se cumplen los criterios al parar la iteración, volver a 4. Sino, avanzar a 8.
8. Mostrar la mejor solución.

7 Experimentos

El proceso de implementación de la metaheurística ABO aplicado al MCDP, ha permitido obtener resultados, los cuales se presentarán a continuación. Para llevar a cabo las pruebas de este proyecto, se utilizaron los problemas propuestos por Boctor [11]. Éstos son 10 problemas propuestos para el MCDP, y en cada uno varía la cantidad de celdas (*C*) y el número máximo de máquinas por celda (*Mmax).*

7.1 Parámetros utilizados en ABO

Para todas las pruebas se han utilizado los siguientes parámetros de la metaheurística ABO:

- Número de iteraciones = 1000.
- Cantidad de población = 100.

7.2 Matrices de Boctor

Las matrices utilizadas para la implementación de la solución, son extraídas de un grupo de problemas estudiados por F. Boctor [11]. Dichas matrices son de dimensión 16x30 y en total se presentan 10 de éstas. Para los problemas de cantidad de celdas igual a 2, varía la cantidad máxima de máquinas (*Mmax*) entre 8 y 12, y para los problemas de cantidad de celdas igual a 3, varía la cantidad máxima de máquinas (*Mmax*) entre 6 y 9. Los valores óptimos que entrega la investigación de Boctor, para cada uno de los 10 distintos problemas, están adjuntados dentro de las tablas de resultados.

7.3 Referencias nuevas instancias

Tabla 7.3. Referencias de las nuevas 35 instancias.

Problem	Source	M	P	Mmax
CFP01	King-Nakornchai[16]	5	7	3
CFP02	Waghodekar-Sahu[17]	5	7	3
CFP03	Seifoddini[18]	5	18	3
CFP04	Kusiak-Cho[19]	6	8	3
CFP05	Kusiak-Chow[12]	7	11	4
CFP06	Boctor[11]	7	11	4
CFP07	Seifoddini-Wolfe[20]	8	12	4
CFP08	Chandrasekharan-Rajagopalan[21]	8	20	4
CFP09	Chandrasekharan-Rajagopalan[22]	8	20	4
CFP10	Mosier-Taube[23]	10	10	5
CFP11	Chan-and-Milner[24]	10	15	5
CFP12	Askin-Subramanian[25]	14	24	7
CFP13	Stanfel[26]	14	24	7
CFP14	McCormick[27]	16	24	8
CFP15	Srinisavan[28]	16	30	8
CFP16	King[29]	16	43	8
CFP17	Carrie[30]	18	24	9
CFP18	Mosier-Taube[31]	20	20	10
CFP19	Kumar[32]	20	23	10
CFP20	Carrie[30]	20	35	10
CFP21	Boe-Cheng[33]	20	35	10
CFP22	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP23	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP24	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP25	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP26	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP27	Chandrasekharan-Rajagopalan[34]	24	40	12
CFP28	McCormick[27]	27	27	14
CFP29	Carrie[30]	28	46	14
CFP30	Kumar-Vannelli[35]	30	41	15
CFP31	Stanfel[26]	30	50	15
CFP32	Stanfel[26]	30	50	15
CFP33	King-Nakornchai[16]	36	90	18
CFP34	McCormick[27]	37	53	19
CFP35	Chandrasekharan-Rajagopalan[36]	40	100	20

7.4 Resultados

La metaheurística African Buffalo Optimization aplicado al MCDP, ha sido implementando en lenguaje de programación Java en el entorno de desarrollo Eclipse Committers Mars 2.0, y ha sido ejecutado en un notebook con procesador AMD A6-7310 con 4GB de RAM y Sistema Operativo Microsoft Windows 8.1. En promedio, el proceso de ejecución del algoritmo toma 13 horas 20 minutos aproximadamente, poniendo a prueba todos los problemas presentados por Boctor [11].

Durante la fase de implementación se intentó ver el comportamiento del problema bajo ciertos parámetros, (número de iteraciones de la metaheurística, cantidad de población, factores de aprendizaje y lambda), los cuales fueron definidos con la finalidad de mejorar la forma de trabajo de la metaheurística, siendo más cómodo para el implementador. Como se dijo anteriormente, los valores de número de iteraciones =1000 y cantidad de población = 100. Para factores de aprendizaje se utilizó *lp1*= 0.5 y *lp2*= 0.5. Y de igual forma, Lambda fue utilizado con valor 0.5. Al momento de realizar las pruebas, los resultados eran comparados con los óptimos para los problemas de Boctor [11], y así mejorar la eficiencia del programa para alcanzar el óptimo.

ID = Corresponde al número de problema de Boctor.

B = Óptimo alcanzado por Boctor.

PROM = Promedio de óptimos alcanzados en ABO.

O = Óptimo alcanzado en ABO.

RPD = Diferencia alcanzada con el óptimo global.

Tabla 7.4.1 Resultados MCDP implementando ABO con C=2 y Mmax del 8 al 10

C = 2												
ID	Mmax = 8				Mmax = 9				Mmax = 10			
	B	PROM	O	RPD%	B	PROM	O	RPD%	B	PROM	O	RPD%
1	11	11	11	0	11	13	11	0	11	14	11	0
2	7	11	7	0	6	6	6	0	4	4	4	0
3	4	4	4	0	4	4	4	0	4	4	4	0
4	14	17	14	0	13	13	13	0	13	13	13	0
5	9	9	9	0	6	6	6	0	6	6	6	0
6	5	5	5	0	3	6	3	0	3	5	3	0
7	7	5	5	0	4	7	4	0	4	4	4	0
8	13	14	13	0	10	10	10	0	8	12	8	0
9	8	12	8	0	8	8	8	0	8	8	8	0
10	8	8	8	0	5	5	5	0	5	9	5	0

Tabla 7.4.2 Resultados MCDP implementando ABO con C=2 y Mmax 11 y 12

C = 2								
ID	Mmax = 11				Mmax = 12			
	B	PROM	O	RPD%	B	PROM	O	RPD%
1	11	16	11	0	11	11	11	0
2	3	3	3	0	3	3	3	0
3	3	8	3	0	1	1	1	0
4	13	14	13	0	13	16	13	0
5	5	5	5	0	4	7	4	0
6	3	3	3	0	2	5	2	0
7	4	8	4	0	4	4	4	0
8	5	7	5	0	5	9	5	0
9	5	8	5	0	5	8	5	0
10	5	5	5	0	5	10	5	0

Tabla 7.4.3 Resultados MCDP implementando ABO con C=3 y Mmax 6 y 7

C = 3								
ID	Mmax = 6				Mmax = 7			
	B	PROM	O	RPD%	B	PROM	O	RPD%
1	27	34	27	0	18	26	18	0
2	7	12	7	0	6	11	6	0
3	9	14	9	0	4	10	4	0
4	27	34	27	0	18	25	18	0
5	11	22	11	0	8	16	8	0
6	6	13	6	0	4	10	4	0
7	11	18	11	0	5	15	5	0
8	14	22	14	0	11	20	11	0
9	12	21	12	0	12	20	12	0
10	10	17	10	0	8	14	8	0

Tabla 7.4.4 Resultados MCDP implementando ABO con C=3 y Mmax 8 y 9

C = 3								
ID	Mmax = 8				Mmax = 9			
	B	PROM	O	RPD%	B	PROM	O	RPD%
1	11	14	11	0	11	17	11	0
2	6	11	6	0	6	10	6	0
3	4	9	4	0	4	9	4	0
4	14	23	14	0	13	20	13	0
5	8	13	8	0	6	14	6	0
6	4	11	4	0	3	8	3	0
7	5	11	5	0	4	13	4	0
8	11	21	11	0	10	22	10	0
9	8	16	8	0	8	17	8	0
10	8	15	8	0	5	12	5	0

Se realizaron 31 ejecuciones para cada problema de MCDP, teniendo los resultados presentados en las tablas 7.4.1, 7.4.2, 7.4.3 y 7.4.4. De dichos datos, se puede observar que en general para los problemas de 3 celdas, la variación entre la solución encontrada y el promedio, puede llegar a ser una gran diferencia en algunos casos. Aun así, la metaheurística logro presentar soluciones óptimas al problema. Para los problemas de 2 celdas, también se alcanzó el óptimo en todos los casos, lográndose apreciar una variación menor, entre la solución encontrada y el promedio de las soluciones.

La metaheurística aplicada al MCDP, logró alcanzar el óptimo en los 90 problemas (100%) de los cuales 50 son de problemas de 2 celdas y 40 de 3 celdas respectivamente.

Para las nuevas 35 instancias, también se realizaron 31 ejecuciones del algoritmo, teniendo los resultados presentados en las tablas 7.4.5 y 7.4.6. En el análisis de dichos datos, se puede observar, que para los problemas de 2 celdas de Mmax 3 a Mmax 10, el algoritmo es robusto. De Mmax 10, específicamente el problema de Boe-Cheng[33] hasta Mmax 20, los resultados tienen ciertas variaciones entre el óptimo encontrado y el promedio de éstos, con un rango de diferencia de [1,8]. Para los problemas de 3 celdas, de Mmax 2 a Mmax 6, también se logra apreciar que el algoritmo es robusto. De Mmax 6, específicamente el problema de King [29] a Mmax 14, los resultados tienen ciertas variaciones entre el óptimo encontrado y el promedio de éstos, con un rango de diferencia de [2,9].

Al no conocer los óptimos globales de ciertos problemas, no se puede afirmar haber llegado al óptimo en todos ellos, aun así, el algoritmo alcanzo el óptimo en 35 problemas (En los cuales se conocía el óptimo global y se podía hacer una comparación con el óptimo encontrado) de los cuales 20 son problemas de 2 celdas y 15 de problemas de 3 celdas respectivamente.

Tabla 7.4.5 Nuevas instancias de prueba para C=2

Autor del problema	Mmax	Óptimo Global	Óptimo Encontrado	Promedio	RPD %
King-Nakornchai	3	0	0	0	0
Waghodekar-Sahu	3	5	5	5	0
Seifoddini	3	5	5	5	0
Kusiak-Cho	3	2	2	2	0
Kusiak-Chow	4	3	3	3	0
Boctor	4	2	2	2	0
Seifoddini-Wolfe	4	6	6	6	0
Chandrasekharan-Rajagopalan	4	7	7	7	0
Chandrasekharan-Rajagopalan	5	28	28	28	0
Mosier-Taube	5	1	1	1	0
Chan-and-Milner	7	4	4	4	0
Askin-Subramanian	7	1	1	1	0
Stanfel	8	2	2	2	0
McCormick	8	16	16	16	0
Srinisavan	8	12	12	12	0
King	9	15	15	15	0
Carrie	9	13	13	13	0
Mosier-Taube	10	27	27	27	0
Kumar	10	25	25	25	0
Carrie	10	1	1	1	0
Boe-Cheng	10	-	21	24	-
Chandrasekharan-Rajagopalan	12	-	1	4	-
Chandrasekharan-Rajagopalan	12	-	3	5	-
Chandrasekharan-Rajagopalan	12	-	10	11	-
Chandrasekharan-Rajagopalan	12	-	19	22	-
Chandrasekharan-Rajagopalan	12	-	23	25	-
Chandrasekharan-Rajagopalan	12	-	22	23	-
McCormick	14	-	32	34	-
Carrie	14	-	38	40	-
Kumar-Vannelli	15	-	5	6	-
Stanfel	15	-	5	8	-
Stanfel	15	-	24	27	-
King-Nakornchai	15	-	36	37	-
McCormick	19	-	218	229	-
Chandrasekharan-Rajagopalan	20	-	21	28	-

Tabla 7.4.6 Nuevas instancias de prueba para C=3

Autor del problema	Mmax	Óptimo Global	Óptimo Encontrado	Promedio	RPD %
King-Nakornchai	2	2	2	2	0
Waghodekar-Sahu	2	8	8	8	0
Seifoddini	2	11	11	11	0
Kusiak-Cho	2	7	7	7	0
Kusiak-Chow	3	5	5	5	0
Boctor	3	2	2	2	0
Seifoddini-Wolfe	3	7	7	7	0
Chandrasekharan-Rajagopalan	3	14	14	14	0
Chandrasekharan-Rajagopalan	3	39	39	39	0
Mosier-Taube	4	0	0	0	0
Chan-and-Milner	4	0	0	0	0
Askin-Subramanian	5	2	2	2	0
Stanfel	5	2	2	2	0
McCormick	6	22	22	22	0
Srinisavan	6	17	17	17	0
King	6	-	21	24	-
Carrie	6	-	18	21	-
Mosier-Taube	7	-	42	45	-
Kumar	7	-	35	38	-
Carrie	7	-	13	18	-
Boe-Cheng	7	-	35	39	-
Chandrasekharan-Rajagopalan	8	-	4	10	-
Chandrasekharan-Rajagopalan	8	-	4	10	-
Chandrasekharan-Rajagopalan	8	-	16	18	-
Chandrasekharan-Rajagopalan	8	-	33	36	-
Chandrasekharan-Rajagopalan	8	-	35	37	-
Chandrasekharan-Rajagopalan	8	-	34	40	-
McCormick	9	-	74	79	-
Carrie	10	-	55	61	-
Kumar-Vannelli	10	-	10	18	-
Stanfel	10	-	13	21	-
Stanfel	10	-	44	47	-
King-Nakornchai	12	-	49	55	-
McCormick	13	-	324	333	-
Chandrasekharan-Rajagopalan	14	-	42	48	-

8 Conclusión

Hoy en día, el desarrollo y utilización de herramientas relacionadas con la informática, ha brindado muchas soluciones a problemas de manufactura y optimización de procesos. Es por esto que las empresas destinan de sus recursos a la investigación y estudio de nuevos métodos para el desarrollo de sus productos. Bajo esta premisa se enmarcan las metaheurísticas, dando la posibilidad de resolver problemas de la vida real, y obtener resultados beneficiosos para optimizar procesos.

La presente investigación es una muestra del proceso en el cual se llevaron a cabo actividades referentes a la investigación, estudio e implementación de una solución informática que cumple efectivamente con el objetivo de optimizar el MCDP, entregando resultados satisfactorios para los problemas de Boctor [11]. Como primera acción, se registró toda información necesaria con respecto al problema de diseño para celdas de manufactura. Luego, se estableció el modelo matemático con sus respectivas restricciones propias de este estudio. Posteriormente se comprendieron los conceptos fundamentales de la metaheurística African Buffalo Optimization, haciendo un análisis profundo de su pseudocódigo para llevarlo de la mejor forma al lenguaje de programación JAVA y lograr una implementación exitosa.

Dentro de los resultados experimentales entregados, se logra apreciar que la integración de la metaheurística en la solución del problema es eficiente, ya que encuentra los noventa óptimos globales de las instancias de Boctor [11], adicionalmente, el hecho de que sea muy robusto en los problemas de 2 celdas, muestra una característica bastante positiva de esta metaheurística. Respecto a los nuevos 70 problemas (Sólo en los problemas que si se poseía el óptimo global para compararlo con el óptimo encontrado), se logra encontrar 35 óptimos globales, 20 de 2 celdas y 15 de 3 celdas respectivamente.

Como trabajo futuro se propone mejorar el algoritmo implementado, ya sea, ajustando correcta y eficientemente los parámetros o incluyendo un algoritmo de mejoramiento, de manera que los óptimos globales sean encontrados con mayor velocidad o simplemente se encuentren mejores resultados, dejan la implementación de la metaheurística más robusta.

7 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] 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).

[4] Soto R., Crawford B., Almonacid B., Paredes F., A migrating birds optimization algorithm for Machine-Part Cell Formation Problems, Springer International Publishing Switzerland 2015, Part I, LNAI 9413, pp. 270- 281, (2015).

[5] Soto R., Crawford B., Vega E., Johnson F., Paredes F., Solving Manufacturing Cell Design Problems Using a Shuffled Frog Leaping Algorithm, The 1st International Conference on Advanced Intelligent Systems and Computing, vol. 407, pp. 253- 261, (2015).

[6] Venugopal V., Narendran T. T., A genetic algorithm approach to the machine-component grouping problem with multiple objetives, Computers and Industrial Engineering, vol. 22(4), pp. 469- 480, (1992).

[7] 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).

[8] Soto R., Crawford B., Vega E., Paredes F., An Artificial Fish Swarm Algorithm for Manufacturing Cell Design Problems, Springer International Publishing Switzerland, Part I, LNAI 9413, pp. 282- 290, (2015).

[9] Purcheck G., A linear-programming method for the combinatorial grouping of an incomplete set, Journal of Cybernetics, vol. 5, pp. 51- 58, (1975).

[10] Olivia-Lopez E., Purcheck G., Load balancing for group technology planning and control, International Journal of MTDR, vol. 19, pp. 259- 268, (1979).

[11] Fayez F. Boctor, A linear formulation of the machine-part cell formation problem, International Journal of Production Research, vol.29, nº 2, pp. 343- 356, (1991).

[12] Kusiak A., Chow W., Efficient solving of the group technology problem, Journal of Manufacturing Systems, vol 6, pp. 117- 124, (1987).

[13] Sankaran S., Multiple objective decision making approach to cell formation: A goal programming model, Mathematical Computer Modeling, vol. 13, pp. 71- 77, (1990).

[14] Shafter S., Rogers D., A goal programming approach to cell formation Problems, Journal of Operations Management, vol. 10, pp. 2834, (1991).

[15] Odili J. B., Kahar M. M., Anwar S., African Buffalo Optimization (ABO): A Swarm- Intelligence Technique, Procedia Computer Science 76, pp. 443- 448, (2015).

[16] J. King and V. Nakornchai. Machine-component Group formation in Group technology: review and extensión. *The International Journal of Production Research*, 20(2):117-133, 1982.

[17] P. Waghodekar and S. Sahu. Machine-component cell formation in group technology: Mace. *The International Journal of Production Research*, 22(6):937–948, 1984.

[18] H. Seifoddini. A note on the similarity coefficient method and the problem of improper machine assignment in group technology applications. *The International Journal of Production Research*, 27(7):1161–1165, 1989.

[19] A. Kusiak and M. Cho. Similarity coefficient algorithms for solving the group technology problem. *The International Journal Of ProductionResearch*, 30(11):2633–2646, 1992.

[20] H. Seifoddini and P. Wolfe. Application of the similarity coefficient method in group technology. *IIE transactions*, 18(3):271–277, 1986.

[21] M. Chandrasekharan and R. Rajagopalan. Modroc: an extension of rank order clustering for group technology. *International Journal of Production Research*, 24(5):1221–1233, 1986.

[22] M. Chandrasekharan and R. Rajagopalan. An ideal seed nonhierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research*, 24(2):451–463, 1986.

[23] C. Mosier and L. Taube. The facets of group technology and their impacts on implementation-a state-of-the-art survey. *Omega*, 13(5):381–391, 1985.

[24] H. Chan and D. Milner. Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing systems*, 1(1):65–75,1982.

[25] R. Asktn and P. Subramantan. A cost-based heuristic for group technology configuration. *International Journal of Production Research*, 25(1):101–113, 1987.

[26] L. Stanfel. Machine clustering for economic production. *Engineering costs and production economics*, 9(1):73–81, 1985.

[27] W. McCormick Jr, P. Schweitzer, and T. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20(5):993–1009, 1972.

[28] G. Srinlvasan, T. Narendran, and B. Mahadevan. An assignment model for the part-families problem in group technology. *The International Journal of Production Research*, 28(1):145–152, 1990.

[29] J. King. Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, 18(2):213–232, 1980.

[30] A. Carrie. Numerical taxonomy applied to group technology and plant layout. *The International Journal of Production Research*, 11(4):399–416, 1973.

[31] C. Mosier and L. Taube. Weighted similarity measure heuristics for the group technology machine clustering problem. *Omega*, 13(6):577–579, 1985.

[32] K. Kumar, A. Kusiak, and A. Vannelli. Grouping of parts and components in flexible manufacturing systems. *European Journal of Operational Research*, 24(3):387–397, 1986.

[33] W. Boe and C. Cheng. A close neighbour algorithm for designing celular manufacturing systems. *The International Journal of Production Research*, 29(10):2097–2116, 1991

[34] M. Chandrasekharan and R. Rajagopalan. Groupabil1ty: an analysis of the properties of binary data matrices for group technology. *The International Journal of Production Research*, 27(6):1035–1052, 1989.

[35] K. Kumar and A. Vannelli. Strategic subcontracting for efficient disaggregated manufacturing. *BEBR faculty working paper; no. 1252*, 1986.

[36] M. Chandrasekharan and R. Rajagopalan. Zodiac-an algorithm for concurrent formation of part-families and machine-cells. *International Journal of Production Research*, 25(6):835–850, 1987.