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

RESOLUCIÓN DEL MANUFACTURING CELL DESIGN PROBLEM UTILIZANDO HARMONY SEARCH

FELIPE EDUARDO ESCÁRATE QUINTEROS

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

SEPTIEMBRE 2016

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

RESOLUCIÓN DEL MANUFACTURING CELL DESIGN PROBLEM UTILIZANDO HARMONY SEARCH

FELIPE EDUARDO ESCÁRATE QUINTEROS

Profesor Guía: **Ricardo Soto De Giorgis**

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

Carrera: **Ingeniería de Ejecución en Informática**

Septiembre 2016

Dedicado a mis padres y familiares, por apoyarme desde que tengo memoria. A mis amigos y cercanos, por haberme brindado gratos momentos a través de estos años. Y a mis compañeros, que me ayudaron a sobrepasar las dificultades tanto en lo emocional como en lo académico.

Índice

Resumen iii

Abstract iii

Lista de Figuras iv

Lista de Tablas v

1. Introducción 1

2. Definición de Objetivos 2

2.1 Objetivo General 2
2.2 Objetivos Específicos 2

3. Estado del Arte 3

4. Manufacturing Cell Design Problem 4

4.1 Definición del problema 4
4.2 Estado del problema 4
4.3 Representación del problema: 5

5. Harmony Search 7

5.1 Harmony Memory 7
5.2 Comportamiento del HS 8

6. Resultados 11

7. Conclusión 16

8. Referencias 17

Resumen

El presente proyecto tiene por objetivo modelar y resolver el Manufacturing Cell Design Problem (MCDP), utilizando la metaheurística llamada Harmony Search, (HS). El MDCP consiste en la agrupación de máquinas, y las piezas que éstas procesan, en conjuntos llamados celdas. La idea entonces es lograr identificar una organización de celdas de tal manera que el número de veces que una pieza es transportada entre éstas sea el mínimo posible. Para esto se utiliza el algoritmo de optimización HS, el cual se basa en el proceso de improvisación que realizan los músicos para encontrar un estado perfecto de armonía musical. Los resultados obtenidos demuestran la eficiencia de la implementación desarrollada al lograr llegar a todos los óptimos globales de un conjunto de 90 instancias de MCDP estudiadas.

Palabras claves: Manufacturing Cell Design Problem, Metaheurística, Harmony Search.

Abstract

This project aims to model and solve the Manufacturing Cell Design Problem (MCDP) using the metaheuristic called Harmony Search (HS). MDCP consists in grouping machines and parts that they process, into groups called cells. So, the idea then is to identify an organization of cells such the number of times that a piece is transported between these is minimized. To this end we use the optimization algorithm HS, which is based on the process of improvisation performed by musicians to find a perfect musical harmony. The experimental results demonstrate the eficiency of the proposed approach which is able to reach all global optimums for a set of 90 well-known MDCP instances.

Keywords: Manufacturing Cell Design Problem, Metaheuristic, Harmony Search.

Lista de Figuras

Figura 5.1 Analogía entre la improvisación musical y la optimización ingenieril 7
Figura 5.2 Estructura de la Harmony Memory 8
Figura 6.1 Gráfico Problema 2, MMax=10, C=2, HMS=5, 10.000 iteraciones. 13
Figura 6.2 Gráfico Problema 7, MMax=7, C=3, HMS=5, 10.000 iteraciones. 14
Figura 6.3 Gráfico Problema 1, MMax=9, C=2, HMS=5, 10.000 iteraciones. 14
Figura 6.4 Gráfico Problema 9, MMax=6, C=3, HMS=5, 10.000 iteraciones. 15

Lista de Tablas

Tabla 4-1 Ejemplo matriz máquina-pieza 5
Tabla 4-2 Ejemplo matriz máquina-pieza agrupada 5
Tabla 6-1 Resultados al MCDP implementando HS con C=2 y Mmax 8 hasta 10. 11
Tabla 6-2 Resultados al MCDP implementando HS con C=2 y Mmax 11 y 12. 12
Tabla 6-3 Resultados al MCDP implementando HS con C=3 y Mmax 6 y 7. 12
Tabla 6-4 Resultados al MCDP implementando HS con C=3 y Mmax 8 y 9. 13

1. Introducción

El proyecto documentado en este informe, busca encontrar una solución al Manufacturing Cell Design Problem, o MCDP, mediante el uso de la metaheurística Harmony Search. El MCDP es una estrategia de manufacturación que consiste en la creación de un diseño óptimo de plantas de producción, buscando una asignación de máquinas y piezas de manera que se minimice el movimiento e intercambio de éstas entre familias llamadas celdas. Esto nace a través de la constante búsqueda de empresas y organizaciones de encontrar soluciones que generen una mayor productividad, reduciendo los costes de producción. Dado esto, el MCDP ha sido objeto de diversas investigaciones, en donde Burbidge, con su análisis de flujo de producción en el año 1963, convierte su procedimiento en uno de los primeros para resolver esta problemática [1].

El MCDP ha tenido dos líneas de investigación complementarias: métodos aproximados y optimización global [2]. Este proyecto se centra en el uso de los métodos aproximados, los cuales sacrifican la garantía de obtener un óptimo global a cambio de poder encontrar una solución satisfactoria en menor tiempo y con un menor consumo de recursos [2].

En este problema se utilizará la metaheurística desarrollada por Geem et al. [2], Harmony Search (HS), el cual es un algoritmo inspirado en la manera que los músicos buscan la armonía perfecta u óptima, mediante la improvisación musical. La armonía en la música interpretada es análoga al vector solución de la optimización, y la improvisación de los músicos es análoga a la búsqueda de óptimos locales y globales. El HS no requiere de valores iniciales en las variables de decisión, ya que éstas las genera aleatoriamente al comienzo la ejecución del algoritmo. Además, comparado con otras metaheurísticas, el HS compromete pocos cálculos matemáticos, lo que hace que sea más fácil adaptarlo a diversos problemas de optimización.

Para comenzar, el informe realiza la especificación de los objetivos generales y específicos del proyecto. Se detallan luego investigaciones previas del MCDP mediante el Estado del Arte. Posteriormente, se realiza una descripción detallada de la problemática, en donde se definen las fórmulas matemáticas que serán las restricciones para la búsqueda de soluciones. Luego, se presenta el algoritmo de optimización escogido, y se definen sus pasos para llevar a cabo la búsqueda de soluciones. Después, se muestran los resultados obtenidos a través de la implementación del HS en MCDP. Finalmente, se presentan las conclusiones del trabajo realizado.

2. Definición de Objetivos

La investigación aplicada que se desarrollará y será explicada mediante este documento, comprende un conjunto de metas a cumplir, las cuales también se pueden resumir en un único objetivo que abarca a todo el proyecto.

2.1 Objetivo General

El objetivo principal es modelar y resolver el MCDP, utilizando el algoritmo de optimización Harmony Search (HS).

2.2 Objetivos Específicos

La solución planteada satisface los siguientes objetivos específicos, desprendidos del objetivo general, los cuales consisten en:

- Investigar las variables y restricciones referentes al MCDP.
- Estudiar las variables y condiciones referentes a la metaheurística Harmony Search.
- Integrar el algoritmo de metaheurística Harmony Search con la implementación del MCDP.
- Estudiar y experimentar el comportamiento de la implementación desarrollada, aplicada en problemas complejos de MCDP.

3. Estado del Arte

En la actualidad la mayoría de las empresas de todo el mundo están en una constante mejora de procesos de manufactura con el fin de aumentar su productividad, buscando principalmente disminuir los costes de operación en términos de tiempo y dinero. El diseño de celdas de manufactura busca desarrollar una estrategia de productividad que entregue lineamientos sobre técnicas y sistemas de producción en empresas para tener la suficiente competitividad y rentabilidad en el mercado.

En años recientes y en la actualidad diferentes métodos metaheurísticos han sido usados para resolver problemas de agrupación de máquinas; Boctor [3]; Chen & Srivastava [5] y más recientemente Wu, Chang, y Chung, presentaron un algoritmo denominado SACF (Simulated Annealing Cell Formation), técnica fundada en Simulated Annealing (SA) [6], que buscaba esencialmente minimizar el número de elementos excepcionales, entendiéndose estos últimos como máquinas asignadas a una celda mientras al mismo tiempo son requeridas por otras celdas. El comportamiento eficiente de este método es esperable principalmente sólo en experimentos a gran escala.

Venugopal y Narendran [7] presentaron el intento de resolver un problema de diseño de celdas con la ayuda de un algoritmo de computación evolutiva, el cual considera tres funciones medulares; minimizar los movimientos totales interceldas así como los movimientos intraceldas, minimizar la variación de carga en una celda y por último minimizar las funciones objetivo de manera simultánea. Gupta, Kumar, y Sundaram [8] emplean la misma representación genética de soluciones que Venugopal y Nerendran [7] sobre la base de un algoritmo evolutivo, pero siguiendo un multi-objetivo diferente de optimización, donde el número total de movimientos y la variación de carga de la celda es aproximado.

Aljaber, Baek y Chen [9] y Lozano, Adenso, Salinas, y Gimenez [9] usan el algoritmo Tabu Search (TS). Aljaber et al. [9] entregaba un diseño del MCDP basado en la teoría de grafos y en un par de problemas del camino más corto, este modelo en general producía mejores soluciones en términos de calidad pero operaba con un alto tiempo computacional. Lozano et al. [9] presentó un enfoque gradual al problema de la agrupación de máquinas, él asume ciertos límites en el tamaño de celdas y familias de componentes para máquinas. Él implementa un algoritmo basado en TS el cual fue un punto de referencia contra varias técnicas SA, heurísticas y otros métodos TS, además de entregar la propuesta de un modelo de programación cuadrático entero, basado en la suma ponderada de espacios intraceldas y movimientos interceldas, este método propuesto supera a otros procedimientos de la época teniendo menores tiempos de computación.

4. Manufacturing Cell Design Problem

4.1 Definición del problema

El Manufacturing Cell Design Problem es una estrategia de manufactura que divide un sistema de producción en pequeños grupos o células cada una de ellas enfocada a la producción de un conjunto de partes o componentes. Esta estrategia de producción es una clara consecuencia de la aplicación de la filosofía de la tecnología de grupos, en inglés Group Technology (GT), la cual plantea que "cosas parecidas deberían de ser fabricadas de manera parecida". Las celdas de manufactura buscan distribuir la totalidad de piezas fabricadas en máquinas necesarias para su fabricación en un conjunto de grupos, concentrado cada uno a la fabricación de un subconjunto de estas partes que forman familias y que son determinadas de acuerdo a las similitudes entre ellas, similitudes que pueden ser de tipo geométrico, por peso, materiales de fabricación, volúmenes de fabricación, operaciones necesarias, entre otras.

4.2 Estado del problema

Uno de los objetivos principales del MCDP es el de minimizar los movimientos e intercambio de material entre los grupos, objetivo que es logrado si se generan celdas que garanticen la fabricación completa de los productos asignados. Si esto no es posible entonces habría piezas que deberán visitar diferentes grupos en su fabricación.

Para lograr la reorganización del sistema de producción es necesario conocer las rutas de fabricación de cada una de las piezas, lo cual permite determinar las máquinas visitadas por cada una de las partes producidas en su ruta de fabricación. Una herramienta que permite de manera ordenada resumir esta información es la matriz pieza-máquina en la cual se puede determinar con unos o ceros las máquinas que son necesarias para la fabricación de cada una de las piezas. En esta matriz de manera general se representan las máquinas en las filas y las piezas en las columnas, esta representación no presenta en ningún momento información referente a la secuencia de las operaciones o el número de máquinas existentes de cada tipo en las instalaciones de manufactura.

En la Tabla 4.1 se muestra un ejemplo de una matriz máquina-pieza, en la cual se observa que los únicos componentes son 1 o 0 (que en este caso se representa con una celda vacía) teniendo los siguientes significados: si la componente a_{ij}= 0, la máquina *i* no procesa la pieza *j*, de lo contrario si la componente a_{ij}= 1 la máquina *i* sí procesa la pieza *j*. A partir de la matriz máquina-pieza se podría determinar una agrupación natural de piezas y partes en celdas. Para ello se hace necesario generar un procedimiento lógico que permita transformar una matriz de la forma observada en la Tabla 4.1 en una matriz máquina-pieza como la mostrada en la Tabla 4.2 0. En la Tabla 4.2 se observa una matriz máquina-pieza en la que la diagonal principal está formada preferentemente por números 1, mientras que las componentes en otro sector de la matriz son 0. En esta tabla se observa que los 7 tipos de piezas y los 7 tipos de máquinas se agrupan principalmente en tres grupos totalmente independientes entre sí.

Máquina \ Pieza	1	2	3	4	5	6	7
1		1			1		
2	1						1
3			1	1		1	
4			1	1		1	
5	1						1
6			1	1		1	
7		1			1		

Tabla 4-1 Ejemplo matriz máquina-pieza

Máquina \ Pieza	1	7	3	4	6	2	5
2	1	1					
5	1	1					
3			1	1	1		
4			1	1	1		
6			1	1	1		
1						1	1
7						1	1

Tabla 4-2 Ejemplo matriz máquina-pieza agrupada

4.3 Representación del problema:

Manufacturing Cell Design Problem se concibe como un modelo con variables, dominios y restricciones, que deben satisfacerse para encontrar una organización de celdas que sea óptima en cuanto a costos de producción. Dicha estructura a buscar debe tener como característica principal reducir al mínimo los movimientos de piezas entre grupos. Es así como MCDP se considera un problema con restricciones que deben ser optimizadas, el cual posee parámetros, variables y dominios como los descritos a continuación 0:

- *M*, representa el número de máquinas.
- *P*, representa el número de piezas (o partes) a ser producidas.
- *C*, representa el número de celdas del modelo.
- M_{max}, corresponde a la cantidad máxima de máquinas que permite una celda.
- A = $[a_{ij}]$ matriz binaria Máquina-Pieza con dominio [0,1] y dimensión *M* por *P*.
  - Donde $[a_{ij}] = \begin{cases} 1, & si\ la\ máquina\ i\ produce\ la\ pieza\ j \\ 0, & en\ otro\ caso \end{cases}$
- B = $[y_{ik}]$ matriz binaria Máquina-Celda con dominio [0,1] y dimensión *M* por *C*.
  - Donde $[y_{ik}] = \begin{cases} 1, & si\ la\ máquina\ i\ pertenece\ a\ la\ celda\ k \\ 0, & en\ otro\ caso \end{cases}$

- C = $[z_{jk}]$ matriz binaria Pieza-Celda con dominio [0,1] y dimensión *P* por *C*.
  - Donde $[z_{jk}] = \begin{cases} 1, & si\ la\ pieza\ j\ pertenece\ a\ la\ celda\ k \\ 0, & en\ otro\ caso \end{cases}$
- *i*, corresponde al índice de las máquinas (*i* = 1, … , *M*).
- *j*, corresponde al índice de las piezas (*j* = 1, … , *P*).
- *k, corresponde al índice de las celdas (k = 1,… ,C).*

La función objetivo de un MCDP está definida por la combinación de las variables y constantes declaradas con anterioridad; la formulación matemática de ésta se presenta a continuación, en donde se expresa que el objetivo es minimizar el movimiento de piezas entre las celdas:

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

Para finalizar, la función objetivo presentada de manera previa, está sujeta a las siguientes restricciones, en donde cada una de ellas asegura si:

- Una máquina pertenece sólo a una celda:

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

- Una pieza pertenece sólo a una celda:

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

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

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

5. Harmony Search

El Harmony Search (HS) es un algoritmo metaheurístico basado en el proceso musical de búsqueda de un perfecto estado de armonía. Los interpretes musicales buscan encontrar una armonía agradable (un estado perfecto) determinado por un estándar estético, al igual como un proceso de optimización busca encontrar una solución global (un estado perfecto) determinada por una función objetivo. La nota de cada instrumento musical determina la calidad estética, al igual como el valor de una función objetivo es determinado por un conjunto de valores asignados a cada variable de decisión. El HS se basa en el proceso que ocurre cuando un músico busca un mejor estado de armonía durante alguna performance musical, como por ejemplo durante una improvisación de jazz [3].

La figura 5.1 muestra en detalle la analogía entre la improvisaci1ón musical y la optimización ingenieril. En la improvisación musical, cada músico toca una nota en el rango posible, produciendo en conjunto un vector de armonía. Si todos las notas producen una buena armonía, la experiencia es almacenada en la memoria de cada músico, y así se incrementa la posibilidad de crear una buena armonía la próxima vez. Esto ocurre de una manera similar en un proceso de optimización ingenieril, cada variable de decisión escoge inicialmente un valor cualquiera dentro del rango posible, haciendo juntos un vector solución. Si todos los valores de las variables de decisión hacen una buena solución, la experiencia es almacenada en la memoria de cada variable, y las posibilidades de crear una buena solución la próxima vez aumentan.

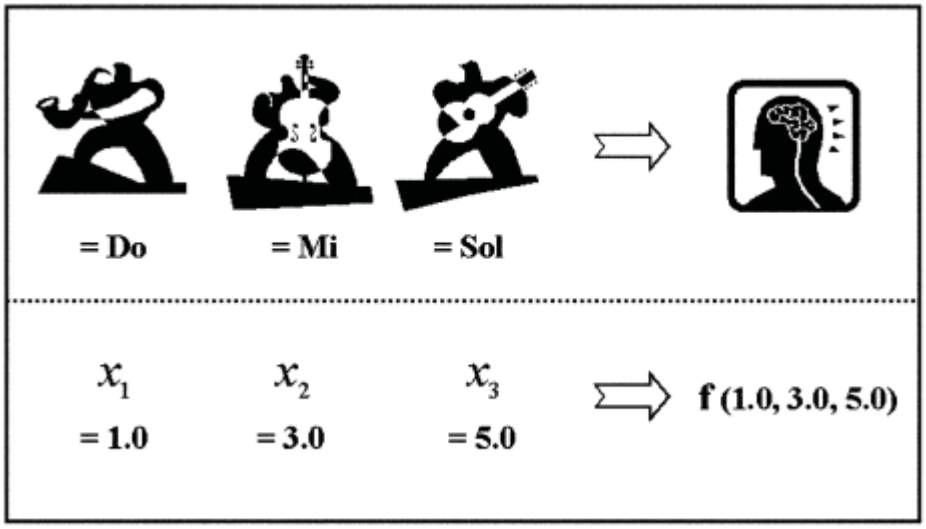

Figura 5.1 Analogía entre la improvisación musical y la optimización ingenieril

5.1 Harmony Memory

Harmony Memory (HM) o en español Memoria de Armonía, es un elemento fundamental de la metaheurística HS. En la figura 5.2 se muestra su estructura. Imaginemos un trío de jazz compuesto por un saxofonista, un contrabajista y un guitarrista. Existe una cierta cantidad de notas preferibles en la memoria de cada instrumentista, por ejemplo: saxofonista, {Do, Mi, Sol};

contrabajista, {Si, Sol, Re}, y guitarrista, {La, Fa, Do}. Si el saxofonista toca aleatoriamente la nota {Sol} sacado de su memoria {Do, Mi, Sol}, el contrabajista toca la nota {Si} de su memoria {Si, Sol, Re}, y el guitarrista toca la nota {Do} de su memoria {La, Fa, Do}, en conjunto crean la armonía (Sol, Si, Do) la cual crea otra armonía llamada acorde de Do mayor séptima menor. Si esta armonía es mejor que la peor almacenada en la HM, se guarda reemplazando a la peor. Este proceso se repite hasta que una armonía fantástica es encontrada. En un proceso de optimización real, los músicos son reemplazados por cada variable de decisión, y sus notas preferibles son reemplazadas por el valor preferible de cada variable. Supongamos que tenemos 3 variables, las cuáles eligen los valores: {100} de {100, 300, 500}, {500} de {700, 500, 200}, y {400} de {100, 500, 400}, esos valores escogidos (100, 500, 400) crean otro vector solución. Si este vector es mejor que el peor de la HM, este incluido reemplazando al peor vector almacenado. Este proceso se repite hasta que un criterio de parada es alcanzado.

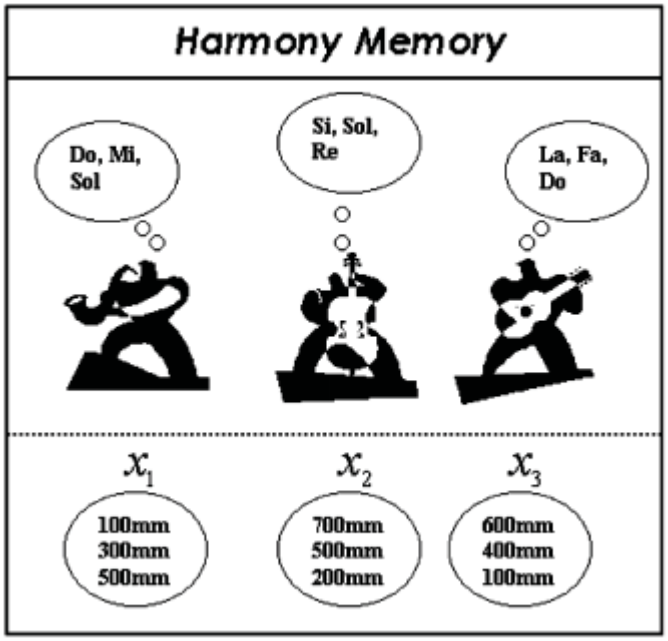

Figura 5.2 Estructura de la Harmony Memory

5.2 Comportamiento del HS

Cuando un músico improvisa una nota, usualmente él sigue una de estas tres reglas:

1) Toca una nota de su memoria.
2) Toca una nota adyacente a una de su memoria.
3) Toca una nota aleatoria dentro del posible rango de notas.

Análogamente, cuando cada variable de decisión escoge un valor en el algoritmo HS, sigue una de estas tres reglas:

1) Escoge cualquier valor dentro de la HM (definido como consideración de memoria).
2) Escoge un valor adyacente a uno almacenado en la HM (definido como ajuste de nota).
3) Escoge aleatoriamente cualquier valor dentro del rango posible (definido como randomización)

Estas 3 reglas del HS son seguidas efectivamente utilizando dos parámetros, Harmony memory consideration rate (HMCR) y Pitch adjusting rate (PAR), las cuales se tratarán más adelante.

El HS considera 5 pasos en su algoritmo, los cuales se detallan a continuación:

1) Inicializar el problema de optimización y los parámetros del algoritmo: en este paso se especifican los parámetros propios del problema a optimizar, además de los parámetros usados en el HS: el Harmony memory size (la cantidad de vectores solución almacenables en la HM, HMS), el Harmony memory consideration rate (HMCR), el Pitch adjusting rate (PAR), y el criterio de parada (la cantidad máxima de búsquedas o iteraciones). Los parámetros HMCR y PAR son usados para mejorar los vectores de solución, éstos se definirán en el paso 3.

2) Inicializar la Harmony Memory (HM): en este paso se llena la matriz HM con vectores solución generados aleatoriamente.

3) Improvisar una nueva armonía a partir de la HM: en este paso, se genera un nuevo vector de armonía a partir de la HM, basado en la consideración de memoria, el ajuste de nota y la randomización. El valor de cada variable de decisión del nuevo vector a generar puede ser escogido desde cualquier valor especificado en el rango de la HM. El parámetro HMCR, que puede tomar valores entre 0 y 1, es la probabilidad de escoger un valor almacenado en la HM, por lo tanto la probabilidad de escoger un valor dentro del rango posible no limitado a la HM es de 1 – HMCR. Por ejemplo, un HMCR con valor 0.95, indica que el algoritmo HS escogerá un valor guardado en la HM para la variable de decisión con una probabilidad del 95%, en consecuencia hay una probabilidad de un 5% de escoger un valor aleatorio dentro del rango posible, no limitándose a la HM. Escoger un valor de 1 para la HMCR no es recomendable, ya que hay posibilidades de que la solución pueda ser mejorada gracias a valores no almacenados en la HM. Luego cada componente del nuevo vector de armonía generado es examinado para determinar si debería ser ajustado. Este procedimiento usa el parámetro PAR que determina la probabilidad de que la variable escogida cambie su valor a uno adyacente. El valor 1- PAR determina entonces la posibilidad de que la variable de decisión se quede como está. Si la variable de decisión es ajustada por este parámetro, a esta variable se le suma $\propto$, el cual tiene un valor de bw x u(-1,1). El parámetro bw es una distancia arbitraria, y u(-1,1) es un número comprendido entre -1 y 1. Los parámetros HMCR y PAR son utilizados en el algoritmo Harmony Search para facilitar el descubrimiento de mejores soluciones globales y locales, respectivamente.

4) Actualizar la HM: en este paso, si el nuevo vector de armonía es mejor que el peor existente en la HM en términos del valor de la función objetivo, la nueva armonía se almacena en la HM reemplazando a la peor.

5) Repetir los pasos 3 y 4 hasta que se cumpla el criterio de parada: en este paso, el algoritmo finaliza si se cumple el criterio de parada. En caso contrario se repiten los pasos 3 y 4.

6. Resultados

Para el algoritmo Harmony Search (HS), al igual que el MCDP, se codificó en lenguaje Java en el entorno de desarrollo NetBeans 8.0.2, y se ejecutó en un notebook con procesador Intel® Pentium® CPU N2840 @ 2.16GHz, una RAM de 4GB y un Sistema Operativo de Windows 10. Los resultados obtenidos hasta el momento se ven resumidos en las Tabla 6.1, 6.2 y 6.3. Cada instancia fue puesta a prueba 31 veces, con los siguientes parámetros de HS: HMCR = 0.9, PAR = 0.2, HMS = 5 y cantidad máxima de iteraciones igual a 10.000. Como referencia se utilizó el paper de Fayez F. Boctor, A linear formulation of the machine-part cell formation problem, cuya tabla de comparación observada es "Table 1 Optimal Solution for the 10 test problems" [3]. A continuación se muestra en la Tabla 6.1 los resultados obtenidos por la implementación de HS, en comparación con los óptimos entregados por el paper ya especificado.

B= Óptimo alcanzado por Boctor; M= Mejor óptimo alcanzado en HS; P= Peor óptimo alcanzado en HS; X= Promedio de óptimos alcanzados en HS; RPD= Diferencia alcanzada con el óptimo global.

C=2															
Pblm	Mmax=8					Mmax=9					Mmax=10				
	B	M	P	X	RPD (%)	B	M	P	X	RPD (%)	B	M	P	X	RPD (%)
1	11	11	11	11	0.00	11	11	11	11	0.00	11	11	32	11.87	0.00
2	7	7	14	8.06	0.00	6	6	17	7.42	0.00	4	4	12	7.28	0.00
3	4	4	9	6.06	0.00	4	4	8	4.81	0.00	4	4	12	6	0.00
4	14	14	30	15	0.00	13	13	17	13.61	0.00	13	13	28	14.55	0.00
5	9	9	13	10.45	0.00	6	6	13	7.61	0.00	6	6	14	7.9	0.00
6	5	5	17	7.19	0.00	3	3	10	5	0.00	3	3	9	4.71	0.00
7	7	7	10	7.51	0.00	4	4	11	4.84	0.00	4	4	8	4.77	0.00
8	13	13	24	14.52	0.00	10	10	23	13	0.00	8	8	19	10.06	0.00
9	8	8	27	11.32	0.00	8	8	17	10.52	0.00	8	8	18	9.94	0.00
10	8	8	12	9.1	0.00	5	5	9	7.16	0.00	5	5	17	8.03	0.00

Tabla 6-1 Resultados al MCDP implementando HS con C=2 y Mmax del 8 hasta el 10.

C=2										
Pblm	Mmax=11					Mmax=12				
	B	M	P	X	RPD (%)	B	M	P	X	RPD (%)
1	11	11	24	12.68	0.00	11	11	22	15.35	0.00
2	3	3	7	4.53	0.00	3	3	8	4.16	0.00
3	3	3	8	4.9	0.00	1	1	8	3.23	0.00
4	13	13	23	16.35	0.00	13	13	19	14.32	0.00
5	5	5	17	7	0.00	4	4	12	6.74	0.00
6	3	3	10	4.61	0.00	2	2	9	4.06	0.00
7	4	4	9	5.87	0.00	4	4	10	6.97	0.00
8	5	5	13	8.87	0.00	5	5	18	7.8	0.00
9	5	5	14	8.42	0.00	5	5	14	8.77	0.00
10	5	5	13	7.48	0.00	5	5	10	7.03	0.00

Tabla 6-2 Resultados al MCDP implementando HS con C=2 y Mmax de 11 y 12.

C=3										
Pblm	Mmax=6					Mmax=7				
	B	M	P	X	RPD (%)	B	M	P	X	RPD (%)
1	27	27	42	33.03	0.00	18	18	45	24.71	0.00
2	7	7	37	10	0.00	6	6	33	13.48	0.00
3	9	9	18	12.68	0.00	4	4	18	9.68	0.00
4	27	27	41	32.42	0.00	18	18	40	25.16	0.00
5	11	11	26	14.94	0.00	8	8	22	14.06	0.00
6	6	6	25	10.94	0.00	4	4	22	8.55	0.00
7	11	11	32	14.71	0.00	5	5	28	9.23	0.00
8	14	14	37	20.74	0.00	11	11	29	18.39	0.00
9	12	12	31	16.9	0.00	12	12	23	14.61	0.00
10	10	10	28	15.71	0.00	8	8	34	15.03	0.00

Tabla 6-3 Resultados al MCDP implementando HS con C=3 y Mmax de 6 y 7.

C=3										
Pblm	Mmax=8					Mmax=9				
	B	M	P	X	RPD (%)	B	M	P	X	RPD (%)
1	11	11	33	14.97	0.00	11	11	25	13.35	0.00
2	6	6	28	12.16	0.00	6	6	21	9.58	0.00
3	4	4	17	6.87	0.00	4	4	22	7.68	0.00
4	14	14	40	22.74	0.00	13	13	35	19.13	0.00
5	8	8	34	13.13	0.00	6	6	26	10	0.00
6	4	4	18	8.35	0.00	3	3	13	7.29	0.00
7	5	5	21	9.94	0.00	4	4	19	9.68	0.00
8	11	11	30	17.94	0.00	10	10	31	13.77	0.00
9	8	8	28	14.97	0.00	8	8	25	13.55	0.00
10	8	8	21	10.52	0.00	5	5	26	10.42	0.00

Tabla 6-4 Resultados al MCDP implementando HS con C=3 y Mmax de 8 y 9.

Se observa que en todas las instancias se logra llegar al óptimo global, lo que demuestra la eficiencia de la implementación del HS en el MCDP. Sin embargo, los promedios demuestran que la implementación de la metaheurística podría ser más robusta, ya que si bien estos no están tan alejados del óptimo, sigue habiendo una diferencia que podría disminuirse.

Otra forma de visualizar los resultados es a través de gráficos de convergencia. El objetivo de estos diagramas es determinar la convergencia del algoritmo hacia un óptimo, a medida que el número de iteraciones avanza durante la ejecución de éste. La Figura 6.1, Figura 6.2, Figura 6.3, Figura 6.4, expresa como el óptimo varía a medida que la metaheurística realiza una cantidad de iteraciones.

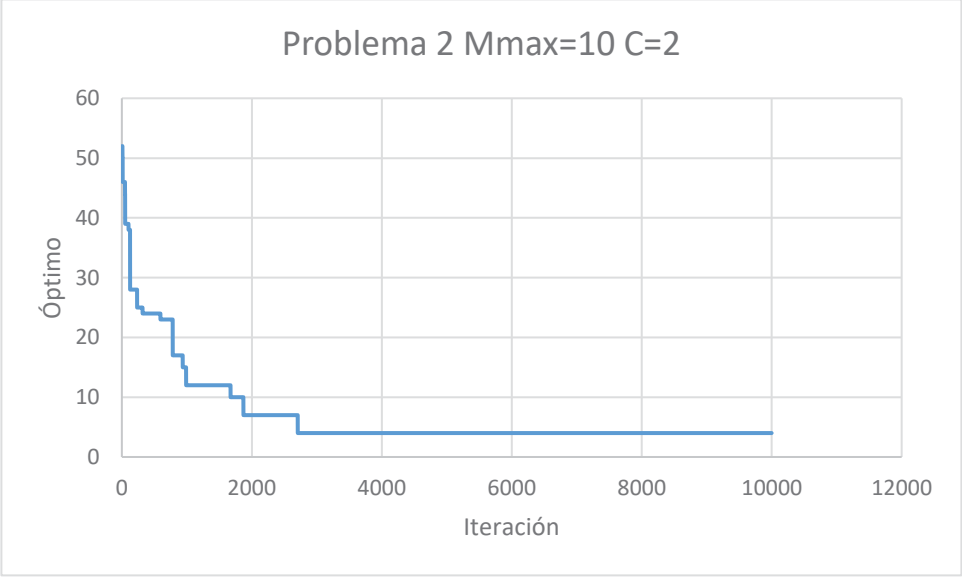

Figura 6.1 Gráfico Problema 2, MMax=10, C=2, HMS=5, 10.000 iteraciones.

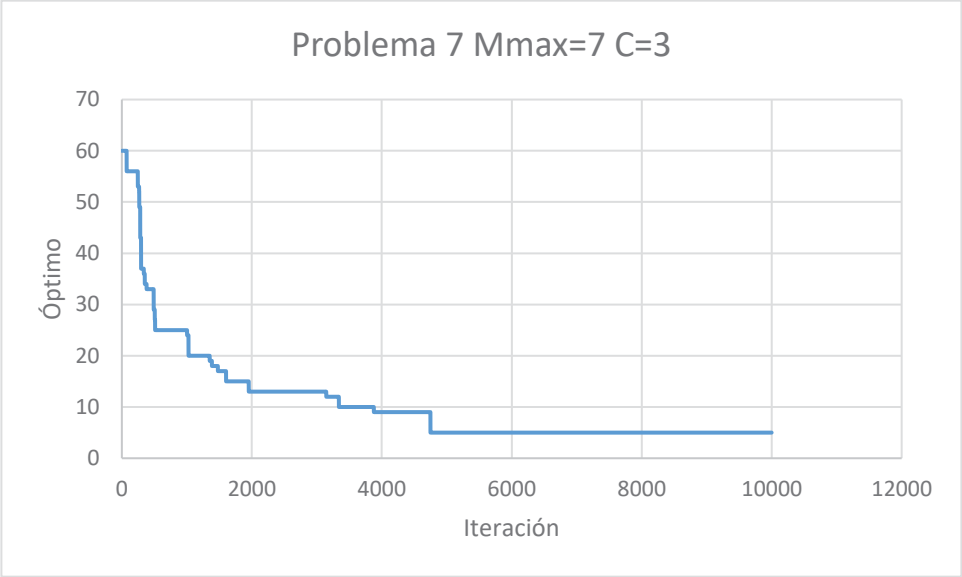

Figura 6.2 Gráfico Problema 7, MMax=7, C=3, HMS=5, 10.000 iteraciones.

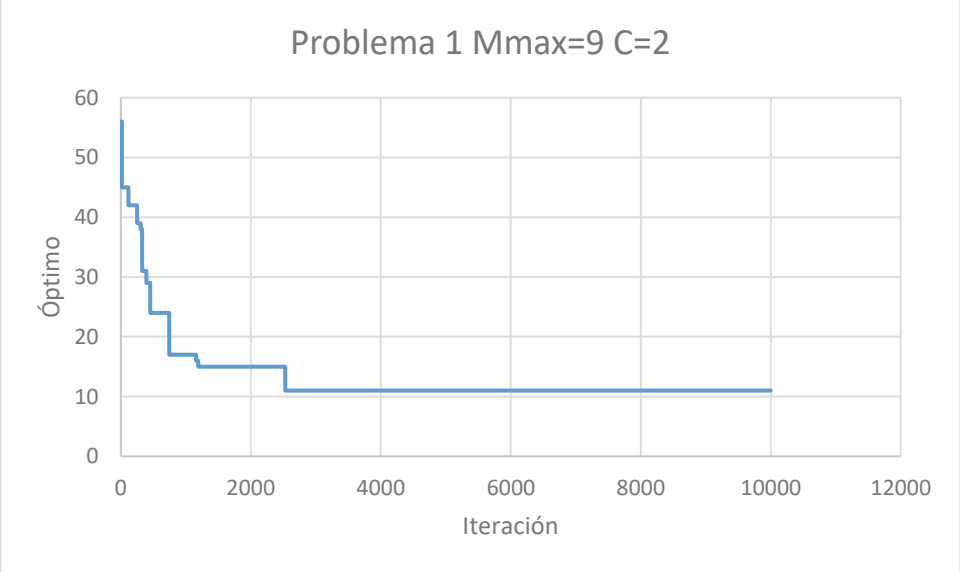

Figura 6.3 Gráfico Problema 1, MMax=9, C=2, HMS=5, 10.000 iteraciones.

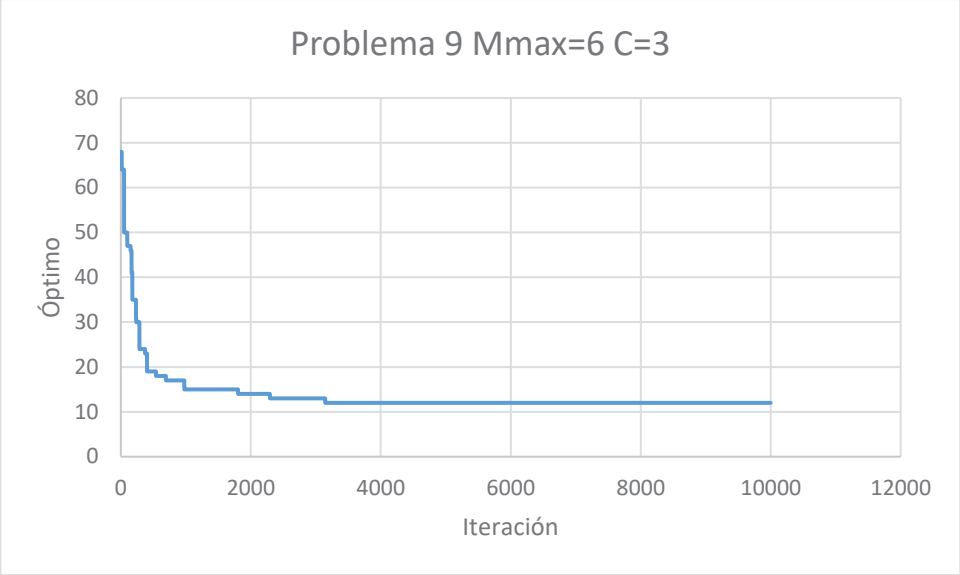

Figura 6.4 Gráfico Problema 9, MMax=6, C=3, HMS=5, 10.000 iteraciones.

Se puede concluir, a través de los gráficos de convergencia, que el valor óptimo converge de una manera adecuada para cada problema presentado, logrando llegar al valor óptimo de Boctor antes de alcanzar la mitad del máximo de iteraciones propuesto.

7. Conclusión

El desarrollo y utilización de herramientas relacionadas con la informática, ya sea para solucionar problemas u optimizar procesos, son eventos que cada día se observan con recurrencia. Debido a esto, se hace imprescindible que se destinen recursos humanos para dedicarse a la investigación y estudio de áreas ya existentes, o para la creación de nuevas que realicen la labor mencionada de forma previa. Bajo esta premisa se enmarcan los métodos aproximados o meta heurísticas, ya que al ser aplicada sobre problemas reales se pueden obtener resultados beneficiosos para optimizar procesos.

El presente informe es una muestra del proceso en el cual se llevaron a cabo actividades referentes a la investigación, estudio y factibilidad de implementación de una solución informática que cumpla con el objetivo de optimizar problemas como MCDP. En primer lugar, se ha registrado la información necesaria con respecto al problema que significa el diseño de celdas de manufactura, o MCDP; se establecieron los recursos que se requieren, así como las restricciones propias de este caso de estudio. Luego, se comprendieron los conceptos fundamentales de la meta heurística Harmony Search, además de entender paso a paso como opera este algoritmo. Finalmente, se mostraron los resultados la implementación desarrollada los cuales todos alcanzan el óptimo global, demostrando la capacidad que tiene el HS para solucionar problemas de optimización.

Como trabajo futuro se propone ajustar los parámetros de una manera en que se llegue más seguido a los óptimos globales, dejando así la implementación de la metaheurística más robusta. También se propone investigar y testear más algoritmos para lograr una mejor explotación de la solución.

Dicho lo anterior, se ha logrado con éxito cumplir con los cuatro objetivos específicos propuestos para este proyecto.

8. Referencias

[1] A. R. Xambre and P. M. Vilarinho. A simulated annealing approach for manufacturing cell formation with multiple identical machines. European Journal of Operational Research, 151(2):434{446, 2003.

[2] R. Soto, H. Kjellerstrand, O. Duran, B. Crawford, E. Monfroy, and F. Paredes. Cell formation in group technology using constraint programming and boolean satisfability. Expert Systems with Applications, 39(13):11423{11427, 2012.

[3] Z.W. Geem, J.-H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60– 68.

[4] Boctor, F. A linear formulation of the machine-part cell formation problem. *International Journal Production Research*, 29(2) (1991), 343-356.

[5] Chen, W. H., and Srivastava, B. (1994). Simulated annealing procedures of forming machine cells in group technology. *European Journal of Operational Research*, 75, 100-111.

[6] Wu, T-h., Chang, C.-C., Chung, S.-H (2008). A simulated annealing algorithm for manufacturing cell formation problems. *Expert Systems with Applications*, 34(3), 1609-1617.

[7] Venugopal, V., and Narendran, T. T. (1992). A genetic algorithm approach to the machine component grouping problem with multiple objectives. Computers and Industrial Engineering, 22.4, 469-480

[8] Gupta, Y., Gupta, M., Kumar, A., and Sundaram, C. (1996). A genetic algorithm-based approach to cell composition and layout design problems. International Journal of Production Research, 34(2), 447-482.

[9] Aljaber, N., Baek, W., and Chen, C, L, (1997). A Tabu Search approach to the cell formation problem. *Computers and Industrial Engineering*, 32(1), 169-185.

[10] Lozano, S., Adenso, B., Salinas, I., and Gimenez, L. (1999). A one-step Tabu search algorithm for manufacturing cell design. Journal of the Operational Research Society, 50(5), 509-516.

[11] P. Medina, E. Cruz, M. Pinzón (2010). *Generación de Celdas de Manufactura Usando un Algoritmo de Ordenamiento Binario*. Scientia et Technica Año XVI, No 44, Abril de 2010. Universidad Tecnológica de Pereira. ISSN 0122-1701. Páginas 106-108.

[12] J. Gutierrez, A. López. Modelado y Resolución del Manufacturing Cell Design Problem (MCDP) utilizando Programación con Restricciones. Informe Final de Proyecto para optar al título de Ingeniero de Ejecución en Informática, Octubre 2011.