

**UNIVERSIDAD ESTATAL DE SONORA
UNIDAD ACADÉMICA SAN LUIS RÍO COLORADO**

**MAESTRÍA EN SISTEMAS DE PRODUCCIÓN
BIOSUSTENTABLES**



**“OPTIMIZACIÓN BI-OBJETIVO DE UN PROCESO PRODUCTIVO
PARA LA ELABORACIÓN DE TORTILLAS CONSIDERANDO EL
AHORRO DE ENERGÍA ELÉCTRICA”**

TESIS

**QUE COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRO EN SISTEMAS DE PRODUCCIÓN BIOSUSTENTABLES**

PRESENTA:

FRANCISCO VILLALOBOS RODRÍGUEZ

SAN LUIS RÍO COLORADO, SONORA.

AGOSTO, 2016

FORMATO F5

ACTA DE LIBERACIÓN DE TESIS
PROCEDIMIENTO PARA LA OBTENCIÓN DEL GRADO DE MAESTRO,
OPCIÓN TESIS

Los miembros del Comité de Tesis del alumno Francisco Villalobos Rodríguez recomendamos que el documento titulado: "Optimización bi-objetivo de un proceso productivo para la elaboración de tortillas considerando el ahorro de energía eléctrica" sea aceptado como Requisito Parcial para obtener el grado de Maestro en Sistemas de Producción Biosustentables.

COMITÉ DE TESIS



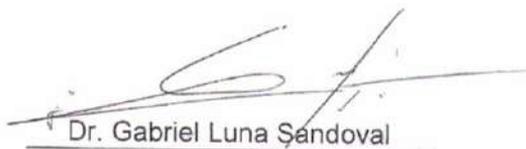
Dr. Victor Hugo Yaurima Basaldúa

Nombre y firma
Director



Dr. Andrei Tchernykh

Nombre y firma
Sinodal



Dr. Gabriel Luna Sandoval

Nombre y firma
Sinodal

AGRADECIMIENTOS

A mi familia por contar siempre con su apoyo en mis estudios.

A mi comité de tesis, por seguir dándome la oportunidad de trabajar con ellos y enriquecer mi trabajo.

Al Dr. Rafael Villa Angulo, por su valiosa aportación en este trabajo.

A la Universidad Estatal de Sonora por darme siempre el apoyo de seguir en mi superación.

A mis maestros de la UES, de licenciatura y posgrado, de los cuales siempre puedo contar con ellos.

DEDICATORIA

A mi familia

ÍNDICE DE CONTENIDO

AGRADECIMIENTOS	ii
DEDICATORIA	iii
ÍNDICE DE TABLAS	vi
RESUMEN	x
INTRODUCCIÓN	2
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA	6
1.1. Objetivos	9
1.1.1. Objetivo general	9
1.1.2. Objetivos específicos	9
1.2. Justificación	9
CAPÍTULO II. MARCO TEÓRICO	12
2.1. Proceso productivo de la industria de estudio	12
2.2. Programación de la producción	15
2.3. Clasificación de problemas en la programación de la producción	16
2.4. Notación general	17
2.5. Taller de flujo híbrido	22
2.6. Métodos de la programación de la producción	24
2.6.1. Métodos exactos	24
2.6.2. Métodos heurísticos	25
2.7. Computación Evolutiva	28
2.8. Algoritmos Genéticos	30

2.9. Optimización multiobjetivo	32
2.10. Algoritmo Genético de Ordenamiento No-dominado II (NSGA-II).....	34
2.11. Funciones de deseabilidad.....	34
2.12. Optimización multiobjetivo en las industrias.....	35
2.13. Calendarización consciente del consumo de energía en un Taller de Flujo	37
CAPÍTULO III. METODOLOGÍA	39
3.1. Desarrollo de la metodología.....	39
3.2. Diseño conceptual del algoritmo	40
3.2.1. Modelo	40
3.2.2. Representación del cromosoma	41
3.2.3. Inicialización y evaluación.....	46
3.2.4. Selección, cruzamiento y mutación.....	47
3.3. Calibración del algoritmo.....	51
3.3.1. Diseño del experimento	51
3.3.2. Especificaciones computacionales	52
3.3.3. Configuración de parámetros de entrada.....	53
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	57
4.1. Análisis de varianza.....	57
4.3. Comparación de resultados con la industria de estudio	66
CONCLUSIONES Y APORTACIONES	71
BIBLIOGRAFÍA.....	74

ÍNDICE DE TABLAS

1. Enumeración de lotes por trabajo.....	42
2. Parámetros usados para la calibración.	51
3. Capacidad de procesamiento de máquinas (Kilogramos x minuto) en la etapa 2.	53
4. Capacidad de procesamiento de máquinas (Kilogramos x minuto) en las etapas 3 a la 7.....	53
5. Características de la carga de trabajo.....	54
6. Tiempo de procesamiento en minutos.....	55
7. Tiempos de configuración de las máquinas por etapa.	56
8. Consumo de energía de las máquinas por etapa (kW x minuto).....	56
9. Análisis de varianza del IRSM para el caso de 2 máquinas por etapa.....	57
10. Análisis de varianza del IRSM para el caso de 4 máquinas por etapa.....	58
11. Análisis de varianza del IRSM para el caso de 6 máquinas por etapa.....	58
12. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 2 máquinas por etapa.	61
13. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 4 máquinas por etapa.	62
14. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 6 máquinas por etapa.	62
15. Óptima combinación de parámetros para el AG bi-objetivo en cada caso de máquinas por etapa.....	62
16. Resultados de los datos de producción de la industria y el AG bi-objetivo.	66
17. Porcentaje de rendimiento en la comparación de los resultados del AG bi-objetivo con los datos de producción de la industria.	68

18. Porcentaje de rendimiento respecto al CO ₂ del AG bi-objetivo con los obtenidos por la industria.	70
--	----

ÍNDICE DE FIGURAS

1. Consumo mensual de energía eléctrica, Enero 2013 a Octubre 2014.....	7
2. Emisiones de CO ₂ producidas en relación a la producción de los pedidos. Enero, 2013 a Octubre, 2014.....	8
3. Línea de producción de tortillas.....	13
4. Taller de flujo híbrido de 6 etapas.	14
5. Esquema general de un algoritmo genético.	31
6. Frente de Pareto.....	33
7. Ejemplo de representación del cromosoma.	42
8. Ejemplo de obtención de tareas.	43
9. Ejemplo de calendario.	44
10. Mutación Insert.	48
11. Mutación Desplacement.	48
12. Mutación Switch.....	49
13. Operador de cruzamiento OX.....	50
14. Operador de cruzamiento PMX.	50
15. Frente de Pareto para el caso de 2 máquinas por etapa.	60
16. Frente de Pareto para el caso de 4 máquinas por etapa.	60
17. Frente de Pareto para el caso de 6 máquinas por etapa.	61
18. Operador de cruzamiento del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.	63
19. Operador de mutación del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.	64
20. Probabilidad de cruzamiento del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.	64
21. Probabilidad de mutación del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.	65

23. Población del caso de 4 máquinas por etapa. Media y 95% de	
24. nivel de confianza	65
25. Comparación del Cmax de los datos de producción de la industria y el AG bi-objetivo.....	67
26. Comparación de la energía de los datos de producción de la industria y el AG bi-objetivo.....	68
27. Comparación del CO ₂ producido por la energía utilizada en la industria y el AG bi-objetivo.....	69
28. Comparación del CO ₂ producido por la energía utilizada en la industria y el AG bi-objetivo.....	69
29. Comparación del CO ₂ producido por la energía utilizada en la industria y el AG bi-objetivo.....	69

RESUMEN

La reducción de la demanda de energía eléctrica en las industrias de producción es indispensable para el desarrollo sostenible. El consumo de energía y su suministro causan efectos ambientales negativos (por ejemplo, las emisiones de gases de efecto invernadero y la acidificación).

Este trabajo se enfoca en la industria de producción de tortillas, dónde la planificación de la producción se basa casi por completo en la experiencia práctica del empleado responsable y por métodos manuales, por lo que el desempeño de la industria es a menudo sub-óptimo. Sobre este marco general, se considera que el uso de métodos matemáticos y la teoría de la programación de la producción podrían dar mejores resultados.

El aumento de la eficiencia en relación con el consumo de energía eléctrica, la productividad de las máquinas y la asignación de horarios en el personal, comprenden un alto potencial para disminuir los costos de producción, mediante el cálculo de un plan de producción óptimo. Además, las mejoras en la programación de la producción también tendrán un efecto sobre el consumo de recursos (eficiencia energética) y emisiones (GEI, residuos, desechos, etc.).

Generalmente la programación de la producción puede ser definida como la distribución de órdenes de producción (trabajos) a unidades de producción (máquinas) y la asociada secuenciación y sincronización en las máquinas. Al modelo de un proceso productivo, con un conjunto de trabajos que se procesan sucesivamente en varias etapas, y dónde cada etapa tiene un conjunto de máquinas capaces de ejecutar los trabajos, se le conoce como taller de flujo híbrido. Se ha demostrado en publicaciones relacionadas que el problema del taller de flujo híbrido es NP-Difícil. En los últimos años, modernas técnicas de optimización heurística, han ganado popularidad en la solución de este tipo de problemas.

En la presente tesis, se propone un algoritmo genético bi-objetivo basado en el modelo de un entorno de producción real, específicamente de una industria de

producción de tortillas. Se considera la optimización simultánea de dos objetivos: 1) Minimizar el tiempo de completar los pedidos y 2) minimizar los consumos de energía eléctrica.

El modelo representa un taller de flujo híbrido, con máquinas no relacionadas e idénticas, tiempos de ajuste en las máquinas que son dependientes e independientes de la secuencia y un búfer para el trabajo en proceso.

Se considera un taller de flujo híbrido con seis etapas, con casos de 2, 4 y 6 máquinas por etapa. A partir del análisis del algoritmo propuesto se concluye que posee una alta eficiencia al hacer la comparación con la situación real de producción de la industria. Los resultados muestran que el algoritmo, considerando los 3 casos de máquinas por etapa, es en promedio 52.7% mejor en el tiempo de terminación de los trabajos y se disminuye en un 51.8% el consumo de energía eléctrica y por consiguiente las emisiones de CO₂ producidas.

INTRODUCCIÓN

La demanda mundial de casi cualquier tipo de mercancías está creciendo continuamente a medida que las poblaciones y los niveles de vida están aumentando. Sin embargo, los recursos necesarios para cumplir estas exigencias son inherentemente escasos. Por lo tanto, el uso sostenible de los recursos es esencial, y "la satisfacción de las necesidades del presente sin comprometer la capacidad de las generaciones futuras para satisfacer sus propias necesidades, debe convertirse en un principio central de las empresas" (Naciones Unidas, 1987).

Los procesos de fabricación e industriales también son conocidos por ser las principales fuentes de gases de efecto invernadero (GEI). Las estadísticas han demostrado que los GEI emitidos por el uso de fuentes de energía en la industria, como la electricidad, el carbón, el petróleo y el gas durante la fabricación, representan el 50% del total de gases de efecto invernadero del mundo. Por lo tanto, las empresas han comenzado a tomar medidas para reducir las emisiones de gases de efecto invernadero de sus productos y servicios bajo la creciente presión derivada de la aplicación del protocolo de Kyoto y el Protocolo de Copenhagen (Newman et al., 2012).

En cuanto a los procesos industriales, las principales iniciativas de las naciones se centraron en la restricción de los impactos ambientales adversos del consumo de energía. El consumo de energía tiene un impacto directo en el progreso económico, por lo que el dilema de la energía se enfrenta a dos retos principales: 1) los países deben reducir el impacto ambiental negativo del consumo de energía, y 2) necesitan consumir energía para la industrialización (Nishant et al., 2014).

Norteamérica representa el 25.6% del consumo de energía eléctrica mundial, dónde a México le pertenece un 5.0%. El país tiene una contribución del 1.67%, del total de las emisiones anuales de gases de efecto invernadero a nivel mundial. El sector industrial encargado de la generación de energía eléctrica es el que produce mayor cantidad de emisiones de CO₂ al ambiente sobrepasando los 75 millones de toneladas de CO₂ anuales (MTCO₂) (Beltrán et al., 2012).

En la región noroeste de México, se encuentra el estado de Sonora, el cual se posicionó en el duodécimo lugar de las entidades federativas que reportan mayores emisiones estacionarias de MTCO_2 con alrededor de 26.4 (Beltrán et al., 2012). Se considera que el sector industrial es el que tiene mayor porcentaje de consumo del servicio de energía eléctrica al disponer de un 51%, de entre el total de usuarios que cuentan con el servicio (Comisión de Energía del Estado de Sonora, 2011).

El municipio de San Luis Río Colorado, al noroeste del estado de Sonora, se considera que la industria es la segunda actividad más importante en la economía. Los principales ramos del sector son la construcción, tortillerías, agroindustrias y la industria maquiladora (Programa de Desarrollo Urbano, 2013). Considerando a estos tipos de industrias, los altos consumos de energía eléctrica de la producción y las emisiones de CO_2 producidos por los procesos industriales, así como los riesgos químicos y residuos peligrosos, son frecuentes en el municipio. El sector de tortillerías, es uno de los mayores consumidores de energía eléctrica, debido a su proceso de producción. El municipio cuenta con 9 productoras.

El objeto de estudio en este trabajo representa una de estas empresas. El consumo de electricidad de la empresa en estudio ha llegado a alcanzar los 22,844 kWh durante el periodo del 2013 a 2014 (Comisión Federal de Electricidad [CFE], 2014). Entre ese periodo, su producción promedio de paquetes de tortillas se ha ido incrementando mensualmente de 50,052.17 a 86,659.7. Esto ocasiona que la demanda de energía eléctrica para la producción, esté incrementado de la misma forma. Debido a esto la industria ha llegado a tener una producción de emisiones de CO_2 , de 14.93 toneladas durante el periodo mencionado.

En la empresa surge la necesidad de reducir su consumo de energía eléctrica. La adopción de capacidades de gestión de la energía se limita generalmente a control de la energía mediante análisis y generación de informes.

La gestión de operaciones en la fabricación, y en especial la programación de la producción, es uno de los problemas más estudiados por las comunidades de investigación y de control de las operaciones. La programación es la asignación de

recursos (humanos y técnicos) a las tareas durante períodos de tiempo determinados, y su objetivo es optimizar uno o más criterios (Pusavec et al., 2010).

Se considera que en la producción de línea de flujo, hay un equilibrio entre la optimización de los tiempos de terminación de los trabajos (que dependen del procesamiento y los tiempos de preparación) y el consumo de energía. Por lo tanto, el análisis de las ventajas y desventajas es una manera eficiente de apoyar la toma de decisiones al programar las operaciones de producción en este entorno. Esta investigación tiene como objetivo promover la noción de producción consciente de la energía.

Desde el punto de vista de la programación de la producción en una tortillería, ésta puede ser descrita como un taller de flujo híbrido de acuerdo con las definiciones comunes, por ejemplo, en Pinedo, (2012) y Ruiz y Vázquez-Rodríguez (2010). En una planta de producción en que la materia prima, tiene que someterse a una serie de operaciones o etapas de procesamiento relacionadas con máquinas específicas establecidas en serie y ser procesada en cada una de las máquinas de forma consecutiva; a este entorno de producción se conoce como un taller de flujo. Si una etapa de producción se compone de varias máquinas paralelas, a este paso de producción en particular se refiere como una etapa. Si los trabajos tienen que ser procesados en una de las máquinas en una etapa, como es el caso de una tortillería, el entorno de máquinas representa un taller de flujo híbrido. (Hecker et al., 2013).

La mayoría de los problemas de este tipo se incluyen en la categoría de los problemas NP-duro (Garey & Johnson, 1979). En consecuencia, los métodos convencionales podrían necesitar tiempo de cálculo exponencial en el peor de los casos para obtener el óptimo, lo que conduce a que el tiempo de cálculo es demasiado alto para fines prácticos (Baños et al., 2011). En los últimos años, modernas técnicas de optimización heurística, que son métodos de búsqueda estocásticos inspirados en los conceptos y principios de la inteligencia artificial, han ganado popularidad en la optimización de energía (Zheng et al., 2013).

En el presente trabajo, se considera la utilización de algoritmos de optimización bioinspirados (Algoritmos Evolutivos), específicamente algoritmos genéticos (AG) debido a sus ventajas tales como la relativa simplicidad de implementación y escalabilidad en los espacios de soluciones de alta dimensión (Goldberg, 1989). Se propone optimizar el proceso productivo de la empresa, considerando 1) satisfacer los pedidos en un tiempo razonable y 2) minimizar los consumos de energía eléctrica. Se considera la optimización simultánea de estos dos objetivos, que pueden a menudo estar en conflicto.

Para la mayoría de estos casos en aplicaciones del mundo real, no existe una solución óptima única y por lo tanto se requiere de un método de solución que deba buscar un conjunto de soluciones no dominadas (óptimo de Pareto), es decir, que busque todas las soluciones de tal manera que no existe ninguna otra solución mejor en todos los objetivos. Por lo que los AG son capaces de encontrar varios miembros del óptimo de Pareto (Lagunas, Moo & Ortiz, 2013; Peñuela, 2007).

CAPÍTULO I.

PLANTEAMIENTO DEL PROBLEMA

En la industria de estudio, se requiere tomar decisiones encaminadas a cubrir los pedidos con la mejor utilización de los recursos disponibles, pero las características y restricciones del ambiente productivo hacen compleja la toma de decisiones, lo cual representa un problema para la empresa al querer aumentar la productividad. Otras condiciones como la diversificación en los productos, cambios frecuentes en los planes y dinámica en la disponibilidad de las máquinas implican cambios frecuentes en la programación de la producción.

Actualmente, en muchos casos, la programación de la producción se realiza por un grupo de personas llamados *planners*, basándose en la experiencia, por métodos manuales, y por lo tanto no garantiza buenas y rápidas soluciones. Aumentar la productividad requiere mejorar la toma de decisiones en la programación de la producción.

El proceso productivo consta de 8 etapas, en las cuales los pedidos pasan por diferentes procesos. En cada etapa existen máquinas con capacidades diferentes que realizan el mismo proceso, pero al momento de asignarles trabajo, no se aprovechan en su totalidad, por lo que existen máquinas que permanecen ociosas. Estos factores mencionados le producen a la empresa un costo, principalmente en el consumo de energía eléctrica y horas de trabajo hombre.

La industria tiende a incrementar su producción por atender el aumento de los pedidos, por lo que se debe de considerar un aprovisionamiento de máquinas para aumentar la capacidad de la planta con la idea de influir en la disminución del tiempo de realización de los pedidos.

Durante el año 2013 la producción de paquetes de tortillas de harina, fue de 50,052; en el año 2014 su producción llegó a 86,660 paquetes. En el período 2013-2014 la producción aumentó mensualmente. Esta tendencia de producción, ocasionó que la

energía eléctrica utilizada también se incrementará (Figura 1).

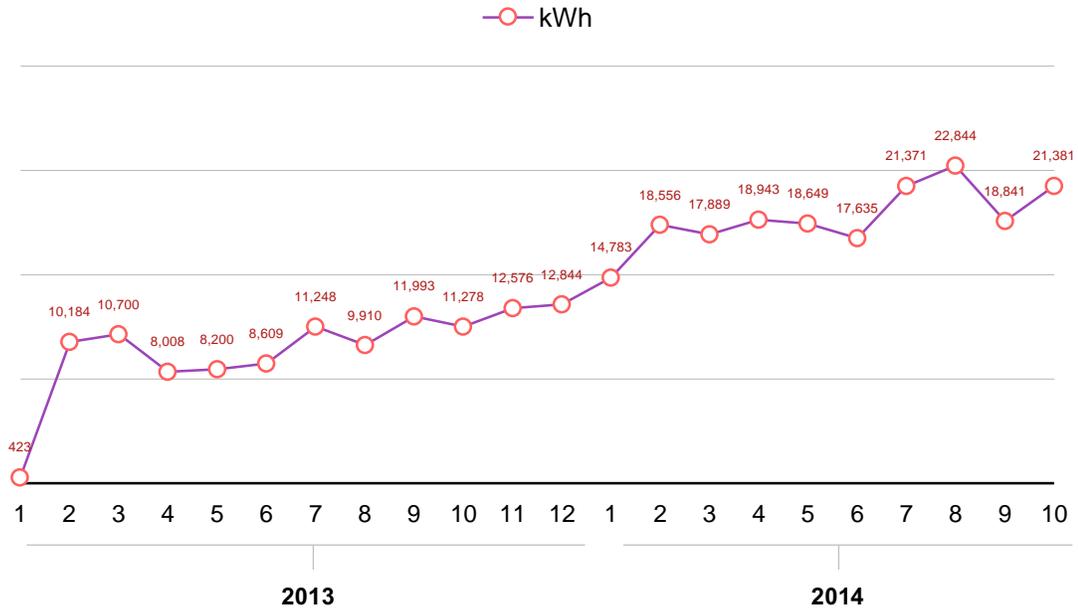


Figura 1. Consumo mensual de energía eléctrica, Enero 2013 a Octubre 2014 (CFE, 2014).

De forma proporcional al incremento de los paquetes producidos y los consumos de energía eléctrica, se genera un incremento en las emisiones de CO₂, (Figura 2).

Si esta tendencia de producción continúa, el aumento en las demandas de consumo eléctrico, provocarán el incremento en la producción de emisiones de CO₂.

Para minimizar el consumo de energía eléctrica, en este trabajo de investigación se pretende abordar el problema de la planeación de la producción, aplicando un algoritmo genético bi-objetivo. Dada la naturaleza de búsqueda que usan los algoritmos genéticos, es viable aplicarlo al problema para obtener resultados comparables a los logrados por la industria. La idea es tener una estrategia de planeación capaz de lograr la mejor utilización de los recursos, considerando 1) la reducción del tiempo de terminación de los trabajos y 2) la reducción del consumo de energía eléctrica. Para ello es necesario desarrollar un algoritmo capaz de lograr

los resultados esperados y después realizar una comparación de rendimiento con una situación real de producción de la industria.

Dado que el espacio de búsqueda para los mejores resultados del algoritmo es extenso y su búsqueda requiere de una gran cantidad de tiempo de cómputo, se espera poder encontrar un conjunto de soluciones lo suficientemente buenas con un desempeño aceptable dentro de un tiempo razonable.

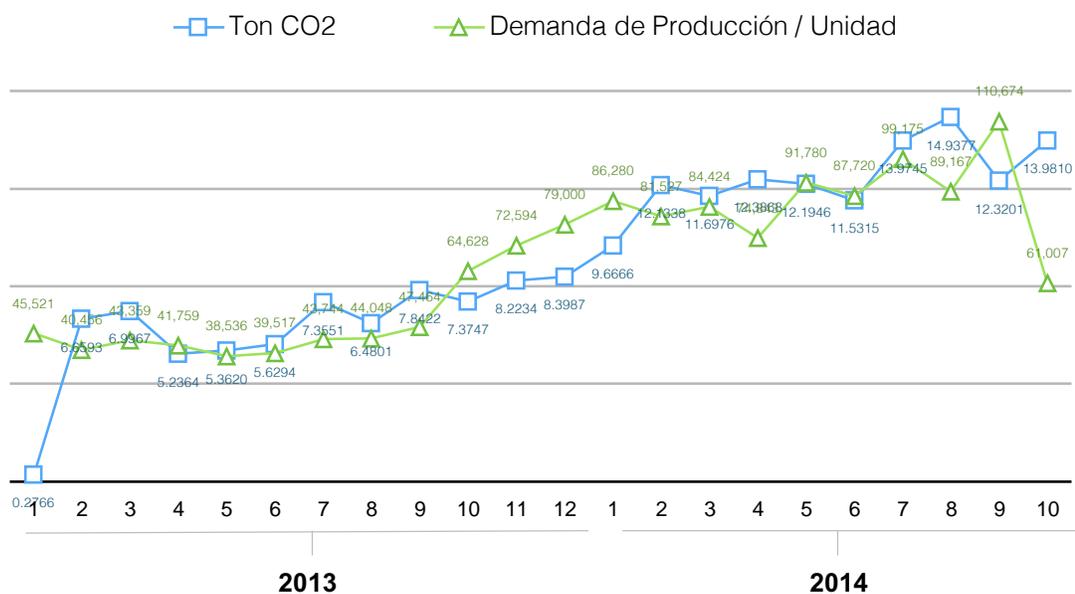


Figura 2. Emisiones de CO₂ producidas en relación a la producción de los pedidos. Enero, 2013 a Octubre, 2014.

1.1. Objetivos

1.1.1. Objetivo general

Desarrollar un algoritmo genético bi-objetivo que permita optimizar un taller de flujo híbrido, considerando un entorno de producción real, donde se minimice el tiempo de terminación de los pedidos y el consumo de energía eléctrica.

1.1.2. Objetivos específicos

1. Analizar el proceso productivo de la industria, considerando características de las máquinas y consumos de energía eléctrica.
2. Modelar el problema del taller de flujo híbrido.
3. Desarrollar un algoritmo genético para resolver el problema.
4. Analizar los resultados de experimentos.
5. Proponer escenarios de optimización del proceso productivo.

1.2. Justificación

Debido a que el sector industrial está actuando para cumplir con la creciente demanda de bienes y siendo uno de los principales consumidores de energía, es esencial considerar la sustentabilidad en el sector de la producción. La reducción de la demanda de energía en las empresas de producción es indispensable para el desarrollo sostenible, porque el consumo de energía y su suministro causan efectos ambientales negativos (por ejemplo, las emisiones de gases de efecto invernadero y la acidificación) (Gahm et al., 2016).

Centrándose en la industria de estudio de producción de tortillas, la planificación de la producción se basa casi por completo en la experiencia práctica del empleado responsable en lugar del uso de métodos matemáticos como la teoría de la

programación de la producción. La combinación de la diversidad de productos y la complejidad de las tareas de programación, hace que el desempeño de las industrias sea a menudo sub-óptimo. El aumento de la eficiencia en relación con el consumo de energía o la asignación de personal y horas de trabajo de origen humano, comprende un alto potencial para disminuir los costos de producción en esta industria. Se hace necesario entonces, analizar si la programación de la producción es un método apropiado para mejorar la eficiencia de la energía en la compañía y si de este modo se puede hacer una contribución sustancial hacia la producción de bienes más sostenibles.

En los procesos reales de producción, la programación es uno de los factores clave en el nivel operativo que influyen en la eficiencia de la producción, la calidad y el costo. Además, las diferencias en la programación también tendrán un efecto sobre el consumo de recursos (eficiencia energética) y emisiones (GEI, residuos, desechos, etc.).

La programación de la producción se define como la distribución de órdenes de producción (trabajos) a unidades de producción (máquinas) y la asociada secuenciación y sincronización en las máquinas. Las condiciones para que la programación de la producción pueda calificarse como sostenible, es cuando considera, además de los objetivos de producción habituales (tiempos de terminación, tiempos de flujo, tardanzas, etc.), la optimización de los medios de entrada (energía, máquinas, materias primas, etc.) y/o medios de salida (residuos, desechos, contaminación) (Giret et al., 2015).

El uso de algoritmos de optimización evolutiva para resolver la tarea de programación podría aumentar la eficiencia de la empresa mediante el cálculo de un plan de producción óptimo y, por tanto, determinar el calendario adecuado y la capacidad de los dispositivos necesarios para alcanzar la meta de producción. Los tiempos de inactividad de las máquinas pueden ser reducidos, lo que conduce a una reducción del consumo de energía o la disminución en el tiempo de terminación de los pedidos que conduce a una reducción de horas necesarias de trabajo hombre.

En un contexto de desarrollo sostenible, servirá para que las operaciones en la programación de la producción, disminuyan el impacto ambiental de la industria, reduciendo la huella de carbono y haciendo más eficiente la producción.

CAPÍTULO II.

MARCO TEÓRICO

2.1. Proceso productivo de la industria de estudio

Esta tesis enfoca su atención en la programación de la producción para una industria de producción de tortillas. En este apartado se describe el ambiente de producción, para lo cual se realizó un seguimiento a una planta productiva.

En los procesos para la preparación de tortillas, los ingredientes son transferidos a una mezcladora donde son combinados para crear la masa. Posteriormente la masa es dividida en fragmentos de forma esférica. Los fragmentos son puestos a reposar para obtener cierta consistencia en la temperatura y humedad. Posteriormente son presionados para tomar la forma de tortilla. Las tortillas son calentadas en estufas. Luego son enfriadas, recolectadas y empaquetadas.

La Figura 3 muestra el proceso de producción dividido en 8 etapas: 1) Preparación de ingredientes, 2) Mezclado, 3) División y Redondeo, 4) Reposo, 5) Presionado, 6) Calentado, 7) Enfriado y 8) Recuento y empaquetado.

Mensualmente son calculadas las necesidades de producción, estimando los kilogramos de tortilla a producir y determinando el orden de su ejecución en las máquinas. En total se procesan mensualmente un promedio de 62,500 paquetes de tortillas de diferentes tipos. La planeación mensual está sujeta a cambios durante el mes en curso por variaciones en los pedidos y circunstancias externas que se gestionan diariamente durante la etapa de “recuento y empaquetado”.

La presente investigación se enfoca en las etapas 2 a la 7, siendo que en estas etapas los tiempos de procesamiento en cada máquina son estadísticamente establecidos. Cada etapa cuenta con una serie de máquinas con iguales o diferentes capacidades.

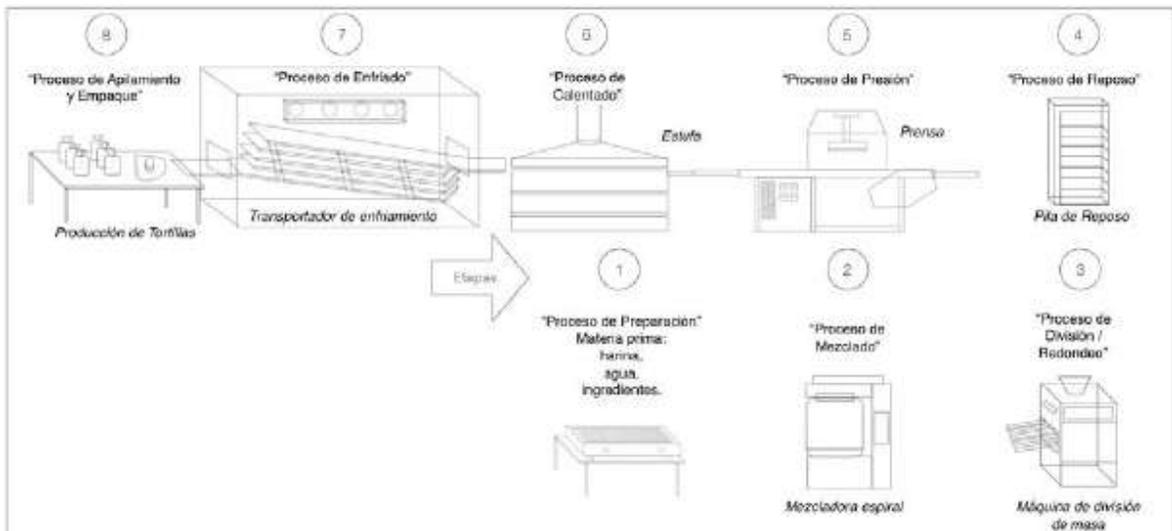


Figura 3. Línea de producción de tortillas.

La masa de tortilla debe de pasar por un proceso en las máquinas de cada etapa. El tiempo de procesamiento para una determinada cantidad de kilogramos de tortilla varía de acuerdo a la máquina que lo procesa y se calcula a través de estadísticas de la capacidad de la máquina, específicamente de acuerdo a los kilogramos por minuto. En las Tablas 3 y 4 se presentan las capacidades de procesamiento de las máquinas.

El plan actual de la empresa incluye 5 tipos de tortillas. Cada tipo de tortilla se distingue principalmente por el tipo de masa, de tal manera que los kilogramos de tortilla de tipo similar pueden procesarse en una sola máquina con la suficiente capacidad. En la Tabla 5 se muestran las características de los trabajos para cada tipo de tortilla.

Las características del ambiente productivo, son las siguientes:

- 1) Los trabajos deben pasar sucesivamente por varios procesos, uno por cada etapa.
- 2) En las etapas 2 y 5 existen grupos de máquinas de distintas velocidades y capacidades. En el resto de etapas las máquinas son idénticas. El tiempo de

procesamiento de cada trabajo depende de la máquina en particular y del tipo de trabajo.

- 3) En la etapa 2, el tiempo de preparación de una máquina y su ajuste correspondiente requerido por el cambio de trabajo, depende esencialmente del trabajo que pasó previamente por esa máquina, lo que significa que existe un tiempo de ajuste dependiente de la secuencia (*Sequence Dependent Setup Time - SDST*).
- 4) Cada máquina dispone de un espacio físico para el almacenamiento temporal (búfer) de los trabajos procesados. Estos trabajos en el búfer son llamados “inventario en proceso” (*Work in Process - WIP*).
- 5) Se requiere minimizar el tiempo total de procesar todos los trabajos y el consumo de energía utilizado en cada proceso.

Este entorno productivo representa un taller de flujo híbrido (TFH) de ocho etapas, de las cuales se consideran para optimizar las etapas 2 a la 7, estimando casos de 2, 4 y 6 máquinas por etapa (Figura 4). Se consideran máquinas paralelas no relacionadas en las etapas 2 y 5 e idénticas en las 3, 4, 6 y 7, tiempos de ajuste dependientes de la secuencia en la etapa 2 e independientes en las demás, estimando un búfer ilimitado.

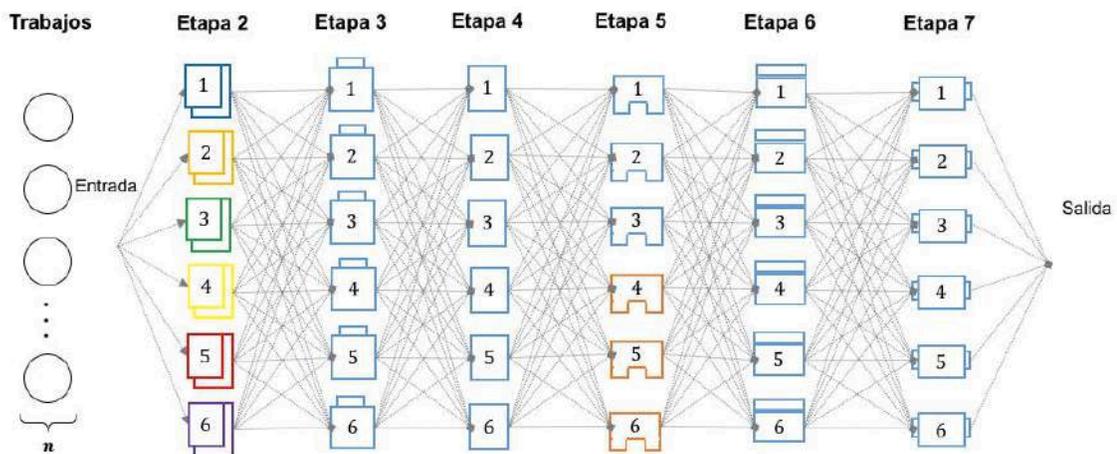


Figura 4. Taller de flujo híbrido de 6 etapas.

Se consideran estas 6 etapas debido a que el proceso es realizado por máquinas, en las etapas 1 y 8, el proceso es realizado por el operador.

Aumentar la productividad, implica establecer un orden en el flujo de los trabajos que permita un mayor uso de las máquinas de tal forma que se minimice el tiempo de completar los trabajos y el consumo de energía.

2.2. Programación de la producción

La elección del orden, en el cual deben entrar los trabajos a la línea de producción y a cuál máquina entre varias idénticas deben asignarse, representa un problema para muchas industrias; a esto se le llama “programación de la producción” que influye entre otras cosas en la productividad de las máquinas, oportunidad del servicio y la rentabilidad.

Independientemente de los tipos de sistemas de producción, una empresa de manufactura debe hacer una planeación de la producción incluyendo en éste un “plan maestro de producción” (*master plan schedule*), con el cual define ciertos parámetros como el personal fijo a laborar, turnos, recursos financieros, productos a fabricar, cantidades, recursos disponibles, etc.

La programación de la producción (*scheduling*) es una etapa importante que debe tomarse en cuenta explícitamente para el logro de los objetivos en la producción (Thomas, 1993). El criterio de optimización más utilizado para este problema es la minimización del tiempo máximo de finalización (*makespan*). Este objetivo consigue una eficiente utilización de las máquinas y permite una finalización más rápida posible del producto.

Una de las primeras investigaciones dedicadas a problemas de la programación de la producción fue realizada por Johnson (1954), quien encuentra una solución óptima para ejecutar un conjunto determinado de trabajos en dos máquinas sucesivas. En los años 80 aparecen varios algoritmos de carácter aproximado para

resolver problemas de programación de la producción. Desde los años 90 hasta la actualidad se ha incrementado la variedad de técnicas avanzadas como algoritmos genéticos (*genetic algorithms*), recocido simulado (*simulated annealing*), búsqueda tabú (*tabu search*), búsqueda local iterativa (*iterated local search*), GRASP (*Greedy Randomized Adaptive Search*) (Allaoui & Artiba, 2006).

2.3. Clasificación de problemas en la programación de la producción

Los problemas en la programación de la producción se clasifican de la siguiente manera (Portman, 1997):

1. Una máquina (*single machine*)

Cada trabajo tiene una sola operación. No existe el problema de asignación pero si el de secuenciación.

2. Máquinas paralelas (*parallel machines*)

Más de una máquina. Cada trabajo tiene una sola operación. Si el número de máquinas es superior al número de trabajos no existe problema de secuenciación pero si el de asignación. En el caso general se tienen que asignar y secuenciar las máquinas.

Las máquinas paralelas se dividen en:

- a) (*Pm*) Máquinas idénticas (*identical parallel machines*). Procesan los trabajos a la misma velocidad.
- b) (*Qm*) Máquinas Proporcionales o Uniformes (*Uniform parallel machines*). Tienen velocidades diferentes y se miden con un factor de proporción.
- c) (*Rm*) Máquinas no relacionadas (*unrelated parallel machine*). La velocidad de procesamiento depende de la máquina y la tarea que se está procesando. Es un caso general que engloba a los dos anteriores.

3. Problemas de tipo taller (Shop).

Se dispone de varias máquinas; cada trabajo tiene un número de operaciones, hay un número de máquinas. Existe una relación de precedencia entre las tareas para cada trabajo, a esta relación se le denomina “ruta” y de acuerdo a las características de las rutas, son:

- a) (*Jm*) Taller de Trabajo (*Job Shop*). Cada trabajo tiene su propia ruta a seguir en las m máquinas.
- b) (*Om*) Taller Abierto (*Open Shop*). No existe ninguna restricción en la ruta. Las operaciones se realizan en cualquier orden.
- c) (*Fm*) Taller de flujo (simple) (*Flow Shop*). Todos los trabajos tienen la misma ruta en las m máquinas. Existe una ordenación predeterminada de las tareas que es la misma en todos los trabajos.

El taller de flujo se extiende en:

- d) (*FFm*) Taller de flujo flexible (*Flexible flowshop*). Es una generalización del taller de flujo simple. Consiste en m etapas en serie. Las etapas contienen máquinas paralelas idénticas. Un trabajo requiere pasar por todas las etapas sucesivamente y por cualquier máquina (una sola) de cada etapa.
- e) (*HFm*) Taller de flujo híbrido (*Hybrid flowshop*). Es una generalización del taller de flujo flexible (TFF). Las máquinas paralelas en al menos una etapa no son idénticas.

2.4. Notación general

Los problemas de programación de la producción se describen a través de:

- El número de trabajos a procesar.
- La manera en que llegan los órdenes de fabricación a la planta productiva.
- El orden en que se utilizan las máquinas para realizar las operaciones.
- Las restricciones tecnológicas sobre los trabajos.

- los objetivos de secuenciación.

Se utiliza la notación de tres campos $\alpha | \beta | \gamma$, conocida como tripleta de Graham para definir cualquier problema de programación de la producción en toda la variedad de problemas, especialmente aquellos que se encuentran en la realidad (Graham et al., 1979).

- El primer campo α define el modelo y el número de máquinas.
- El segundo campo β representa características y restricciones de los trabajos.
- El tercer campo γ define el criterio de optimización.

De acuerdo con Baker (1974) y Pinedo (2002), en un problema de producción se tiene un conjunto N de trabajos, donde $N = \{1, 2, \dots, n\}$, los cuales se procesan en un conjunto m de máquinas, $M = \{1, 2, \dots, m\}$. Comúnmente se utilizan los subíndices j y k para referirse a los trabajos y el subíndice i para las máquinas.

En ciertas circunstancias se permite variar las notaciones de acuerdo a las necesidades específicas del problema. El campo α , define el entorno de producción o la configuración de las máquinas, y se divide en $\alpha_1 \alpha_2$, donde:

- $\alpha_1 = 1$: una sola máquina.
- = P : máquinas paralelas idénticas.
- = Q : máquinas paralelas proporcionales o uniformes. Se proporciona, para cada máquina i un coeficiente de velocidad v_i , donde $p_{ij} = p_j / v_i$.
- = R : máquinas no relacionadas. Además de tener velocidades diferentes, el tiempo de ejecución depende de la tarea.
- = O : taller abierto u “*Open Shop*”.
- = F : taller de flujo o “*Flow Shop*”.
- = J : taller de trabajo o “*Job Shop*”.
- $\alpha_2 = M = \{1, 2, \dots, m\}$: número de máquinas.

= \emptyset : el número de máquinas depende de la instancia a resolver.

El campo β indica las características atribuibles a los trabajos. Estas son numerosas y normalmente los autores dividen este campo en varios subcampos:

β_1 = $pmtn$: se permite la interrupción de operaciones o "*preemption*".

= \emptyset : no se permite la interrupción de operaciones.

β_2 = $prec$: relaciones de precedencia. Con este campo se especifica que existen relaciones de precedencia o "*precedence constraints*" entre los trabajos.

= \emptyset : no existen relaciones de precedencia.

β_3 = r_j : existen instantes o fechas de entrada (*release date*).

= \emptyset : no existen instantes o fechas de entrada.

β_4 = S_{nsd} : existen tiempos de ajuste de las máquinas (*no sequence dependent setup times*) por cambios de tipo de trabajo que no dependen de la secuencia de los trabajos.

= S_{sd} : existen tiempos de ajuste de las máquinas por cambios de tipo de trabajo, los cuales dependen de la secuencia (*sequence dependent setup times*).

= \emptyset : los tiempos de ajuste de las máquinas por cambios de tipo de trabajo no existen o son muy pequeños y se incluyen en los tiempos de procesamiento.

β_5 = d_j : existen instantes o fechas de finalización (*due dates*). Son independientes para cada trabajo.

= $d_j = d$: los instantes o fechas de finalización son homogéneas para todos los trabajos (*common due dates*).

- = Δ_j : los instantes o fechas de finalización son de cumplimiento obligado (*deadlines*).
 - = \emptyset : no existen fechas de entrega.
- β_6 = *prmu* : sólo aplicable cuando $\alpha_1 = F$. Indica que el orden de entrada de los trabajos es el mismo para todas las máquinas (la secuencia es idéntica para todas las máquinas del taller). En este caso existen $n!$ posibles secuencias.
- = \emptyset : indica que la secuencia de entrada en cada máquina varía.
- β_7 = *brkdown* : las máquinas están sujetas a averías o a períodos donde no procesan trabajos (*breakdowns*). Cuando después de una avería el trabajo se reanuda, se dice que es *resumable* (*r*), de lo contrario *non resumable* (*nr*).
- = \emptyset : las máquinas están disponibles en todo momento.
- β_8 = *nwt* : sin esperas o “*no-wait*”. Esta condición impone la restricción de que un trabajo no permite esperar entre dos máquinas.
- = \emptyset : se permite que los trabajos esperen por tiempo indefinido entre las máquinas.
- β_9 = *block* : bloqueo (*bloking*). Indica que existe una capacidad finita entre las máquinas para almacenar el producto en curso (*buffer*). El caso extremo se da cuando no hay buffer entre las máquinas (*zero buffers*).
- = \emptyset : existe un buffer ilimitado entre las máquinas y por tanto ningún trabajo o máquina queda bloqueado.
- β_{10} = *recrc* : recirculación o “*recirculation*”. Se da cuando $\alpha_1 = O, F, J$. Indica que un trabajo (o todos en general) necesita ser procesado más de una vez en alguna o todas las máquinas.

- = \emptyset : los trabajos sólo se procesan una vez en cada máquina.
- β_{11} = M_j : restricción en el uso de máquinas (*machine eligibility constraints*). No todas las m máquinas son capaces de procesar el trabajo j . Se presenta para $\alpha_1 = P$ o en problemas mixtos más complejos.
- = \emptyset : las máquinas siempre están disponibles para todos los trabajos.
- β_{12} = $p_{ij} = 1$: todos los tiempos de procesamiento de los trabajos son iguales a 1.
- = $p_{ij} = p$: todos los tiempos de procesamiento de los trabajos son iguales a una constante p .
- = \emptyset : los tiempos de procesamiento son valores no negativos arbitrarios.
- β_{13} = $size_{ij}$: cualquier tarea de un trabajo j requiere de un número $size_{ij}$ de máquinas para su procesamiento simultáneo en la etapa i (Oguz et al., 2004).
- = \emptyset : no existe esta característica.
- β_{14} = $wait_{ij}$: el tiempo de espera es limitado para cada trabajo j en una etapa i . (Ling-Huey, 2003)
- = \emptyset : no existe esta característica.

El campo γ indica el criterio de optimización:

- γ = C_{max} : minimización del tiempo máximo de finalización (*makespan*). Este criterio de optimización es el más utilizado en la literatura.
- = C : minimización del tiempo medio de finalización. Se calcula como $C = \frac{1}{n} \cdot \sum_{j=1}^n c_j$.

- = F_{max} : minimización del máximo tiempo de flujo, $F_{max} = \max \{F_1, F_2, \dots, F_n\}$. Si no existen instantes de entrada, esto es, si $r_j = 0, \forall j \in N$, entonces $F_{max} = C_{max}$.
- = F : minimización del tiempo medio de flujo $F = \frac{1}{n} \cdot \sum_{j=1}^n F_j$
- = L_{max} : minimización de la máxima holgura, $L_{max} = \max \{L_1, L_2, \dots, L_n\}$. Con esto se busca minimizar el máximo incumplimiento del instante de finalización.
- = L : Minimización de la holgura media. $L = \frac{1}{n} \cdot \sum_{j=1}^n L_j$
- = T_{max} : minimización del retraso máximo. $T_{max} = \max \{T_1, T_2, \dots, T_n\}$. Se busca minimizar el máximo retraso con respecto al instante de finalización.
- = T : Minimización del retraso medio. $T = \frac{1}{n} \cdot \sum_{j=1}^n T_j$
- = E_{max} : minimización del adelanto máximo. $E_{max} = \max \{E_1, E_2, \dots, E_n\}$
- = E : minimización del adelanto medio. $E = \frac{1}{n} \cdot \sum_{j=1}^n E_j$
- = N_T : minimización del número de trabajos retrasados. Se calcula a partir de la expresión: $N_T = \sum_{j=1}^n U_j$, donde $U_j = \begin{cases} 1, & \text{si } C_j > d_j \\ 0, & \text{en caso contrario.} \end{cases}$

Ejemplo: La expresión $HF3, \left((PM^{(i)})_{i=1}^{(3)} \right) | prec | C_{max}$ indica un TFH de tres etapas con máquinas paralelas en cada etapa, restricciones de precedencia y considera como criterio de optimización la minimización del tiempo máximo de finalización.

2.5. Taller de flujo híbrido

En un taller de flujo híbrido (TFH) se procesa un conjunto N de trabajos, $N = \{1, 2, \dots, n\}$, en un conjunto M de etapas, $M = \{1, 2, \dots, m\}$ y en cada etapa $i \in M$, un conjunto $M_i = \{1, \dots, m_i\}$ de máquinas paralelas (no relacionadas en al menos una etapa) procesan los trabajos. Un trabajo j se procesa en todas las m

etapas y dentro de cada etapa i en una de las m_i máquinas. El TFH es una generalización del taller de flujo simple.

El problema de TFH se estudió por primera vez en Arthanary y Ramaswamy (1971). Salvador (1973) y Brah & Hunsucker (1991) demostraron que el número de posibles rutas para los trabajos es muy elevado:

$$\prod_{i=1}^m \left[\binom{n-1}{m_i} \cdot \frac{n!}{m_i!} \right],$$

donde n es el número de trabajos, m el número de etapas y m_i es el número de máquinas en la etapa i , $i = (1, \dots, m)$.

Gourgand, Grangeon y Norre (1999) proponen otra evaluación para el número de posibles soluciones en un problema de tipo HFS:

$$n! \left(\prod_{i=1}^m m_i \right)^n$$

El taller de flujo simple se investiga desde los años 50 con la aportación de Johnson (1954) que proporciona un resultado óptimo para dos máquinas ($F2 \parallel C_{max}$). Cuando el número de máquinas es 3 ó más, el problema es de tipo **NP-Difícil**. Gupta (1988) demostró que este problema con solo dos etapas es **NP-Difícil** aun cuando una de las dos etapas contenga una máquina. De acuerdo a Garey y Johnson (1979) el problema con tres etapas ($F3 \parallel C_{max}$), es **NP-Difícil** en sentido estricto.

Siendo el TFH es una generalización del taller de flujo simple, podemos concluir que el TFH es también **NP-Difícil**. El TFH por el número de etapas, se clasifica en:

- 1) TFH con dos etapas (TFH2).
- 2) TFH con tres etapas (TFH3).
- 3) TFH con k etapas (TFHk).

2.6. Métodos de la programación de la producción

Los métodos que resuelven los problemas de la programación de la producción se dividen en: 1) métodos exactos que aseguran una solución óptima, y 2) métodos heurísticos que no garantizan el óptimo pero proporcionan una buena solución.

2.6.1. Métodos exactos

Los métodos que buscan una solución óptima, se clasifican en eficientes y de enumeración.

Los primeros son aplicables a un grupo de problemas relativamente reducido; principalmente problemas de una máquina, problemas de taller de flujo con un número pequeño de máquinas, o problemas con restricciones en las duraciones de las operaciones.

Los métodos exactos de enumeración realizan búsquedas exhaustivas para encontrar una solución óptima. A cada problema se asocia un árbol que representa el espacio de estados. Cada nodo de este árbol es un estado del problema. El camino desde la raíz hasta este nodo define una tupla que pertenece al conjunto de soluciones.

Los nodos de este árbol de estados se generan en dos formas distintas, lo que da lugar a dos técnicas: Búsqueda con retroceso (*backtracking*) y ramificación y acotación (*branch and bound*).

Otro método de búsqueda enumerativa es la programación dinámica. Sus fundamentos son elaborados por Richard Bellman (Bellman, 1957). Representa una optimización recursiva, con etapas múltiples. Se interpreta como un proceso multietapa de toma de decisiones.

La técnica llamada "Divide y vencerás" consiste en descomponer el problema a resolver en un cierto número de subproblemas más pequeños, resolverlos independientemente (utilizando en numerosas ocasiones técnicas recursivas) y

finalmente recoger los resultados obtenidos en dirección “de abajo hacia arriba” para construir la solución del problema inicial (Wirth, 1976).

Relajación lagrangiana (*lagrangian relaxation*) (Shapiro, 1979; Tang, Luh, Liu & Fang, 2002) se basa en modelos de Programación Lineal Mixta para problemas de programación de la producción. El objetivo es transformar una o varias restricciones en términos de la función objetivo utilizando coeficientes de penalización proporcionales al grado de violación de cada restricción. Una vez realizada la simplificación se intenta resolver el modelo entero mediante procedimientos de exploración dirigida.

El tiempo de ejecución de un algoritmo, el cual aplica un método de los descritos anteriormente, es exponencial, pero, si cada subproblema se resuelve una única vez almacenando el resultado en una tabla de forma que esté accesible siempre que sea necesario, el tiempo de ejecución resulta polinomial. Tales algoritmos resuelven el nombre de pseudopolinomiales (Rosen, 1999). Para algunos de estos problemas, es sencillo encontrar una secuencia de decisiones óptima sin cometer errores: serán los problemas que se resuelven mediante algoritmos voraces. En otras ocasiones, no es posible tomar una secuencia de decisiones que nos lleven al resultado óptimo.

2.6.2. Métodos heurísticos

a) Constructivos

Consisten en ir añadiendo componentes individuales a la solución, hasta que se obtiene una solución factible (Silver et al., 1980). Los algoritmos basados en métodos constructivos, se dividen en:

- 1) Voraces (*greedy*). Son procedimientos constructivos que utilizan una función de “vista corta” que añade un componente a la solución, de tal manera que se obtenga el máximo beneficio en cada paso (El-Zahar & Rival, 1985).

2) De despacho (*dispatching*). Simulan el comportamiento del sistema en tiempo real. Las decisiones se toman cada vez que una máquina finaliza un trabajo. Para ello se usan reglas de prioridad que proporcionan pautas para establecer la secuencia de los trabajos (Giffer & Thompson, 1960).

b) *De descomposición*

Se trata de dividir el problema en subproblemas más pequeños, siendo la solución de uno, la entrada del siguiente, de forma que al resolverlos todos, se obtenga una solución para el problema global.

c) *Basado en índices*

Se aplica un procedimiento que determina la secuencia de los trabajos en cada máquina del sistema, mediante un procedimiento sencillo que no exige una simulación de la secuencia (Palmer, 1965).

d) *Aplicación limitada de algoritmos eficientes*

Un ejemplo de este enfoque es la heurística CDS de Campbell et al. (1970), que extiende el algoritmo eficiente de Johnson (1954) a casos más generales aunque la solución óptima no esté asegurada.

e) *Redes neuronales*

Son estructuras de procesamiento distribuido, paralelo y adaptativo, los cuales se inspiran en la estructura cerebral humana (Martin & Sanz, 1997). La idea básica es que a partir del entrenamiento de la red con diferentes problemas de ejemplo basados en casos conocidos del área, la red sea capaz de extrapolar los resultados a otros problemas diferentes que se presentarían en el futuro.

f) *Metaheurísticas*

Incluyen todos aquellos procedimientos que en un proceso iterativo, guían a una heurística subordinada, combinando inteligentemente diferentes conceptos tomados de la analogía de la naturaleza y explorando el espacio de soluciones utilizando estrategias de aprendizaje para estructurar la información, con el objetivo de encontrar soluciones cercanas al óptimo (Osman, 1995). Estos métodos parten de una solución factible inicial y mediante iteraciones de esa solución, van generándose iterativamente otras soluciones, almacenando como

óptima la mejor de las soluciones generadas hasta que se cumple un criterio de paro. Dentro de las metaheurísticas se encuentran:

- 1) Algoritmos GRASP (*Greedy Randomized Adaptive Search Procedure*): Son procedimientos voraces aleatorizados y adaptativos en la búsqueda de soluciones, desarrollados por Feo y Resende (1989). Generan una solución al problema en dos fases.
- 2) Recocido simulado (*simulated annealing*): La característica de esta heurística es que no depende de la solución inicial. En el recocido simulado esto se consigue mediante el uso de una temperatura inicial alta que permite pasar por muchas soluciones muy alejadas entre sí en las primeras iteraciones del algoritmo (Cerny, 1985; Osman & Potts, 1989).
- 3) Búsqueda tabú (*tabu search*): Consiste en la búsqueda por entornos donde las posiciones visitadas se almacenan en una lista. La búsqueda tabú selecciona el mejor de los movimientos posibles en un entorno $N(x)$ de una solución x . Dentro del proceso de exploración, al alcanzar un óptimo local x_0 se repite la elección del mejor movimiento del vecindario $N(x_0)$ y se vuelve a la posición previa al óptimo local. A partir de esta posición, se vuelve a visitar x_0 , con lo que, el algoritmo entra en un ciclo sin fin (Glover, 1990).
- 4) Optimización basada en colonia de hormigas (*ant colony optimization*): Esta metaheurística fue desarrollada por Colorni et al. (1992). Se inspira en el comportamiento de las colonias de hormigas para solucionar problemas de optimización combinatoria. Las colonias de hormigas tienen la capacidad de encontrar la ruta más corta entre el hormiguero y las fuentes de alimento. Mientras hacen sus recorridos las hormigas depositan en la tierra una sustancia química llamada feromona, formando en el camino un rastro de dicha sustancia. Las hormigas perciben la feromona y cuando tienen que escoger una ruta se deciden por aquella que tenga el rastro más fuerte. Los algoritmos de ACO (*Ant Colony Optimization*) se basan en una colonia de hormigas artificiales, esto es, unos agentes computacionales simples

que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales. En Colorni et al. (1992) se aplica un algoritmo de hormigas a la resolución del problema del comerciante viajero.

- 5) Búsqueda local iterativa (*iterative local search*): Está basada en la búsqueda local que es la base de muchos de los métodos usados en problemas de optimización. La búsqueda local empieza con una solución inicial y busca en su vecindad una mejor solución. Si la encuentra, reemplaza su solución actual por la nueva y continua con el proceso, hasta que no se pueda mejorar la solución actual.
- 6) Algoritmos genéticos (*genetic algorithms*): Pertenecen al ramo de los algoritmos evolutivos. Un algoritmo genético es un procedimiento basado en una analogía con la evolución de los seres vivos. Se buscan soluciones aproximadas a problemas de gran complejidad computacional mediante un procedimiento de evolución simulada matemáticamente en una computadora (Holland, 1975).

2.7. Computación Evolutiva

El Cómputo Evolutivo (CE), también conocido como algoritmos evolutivos (AEs), se caracteriza por proveer de técnicas inspiradas en los principios de la teoría Neo-Darwiniana, de la evolución natural de las especies (Mejía, 2008), a través de un proceso de aprender a adaptarse al medio ambiente e ir evolucionando, imitando el punto de vista de la genética moderna, es decir, el principio de “la supervivencia del más apto” (Yu & Gen, 2010). A base de este principio, en el CE se desarrollan algoritmos capaces de adaptarse al problema que se pretende solucionar (Aguilar, 2014).

Eiben y Smith (2003) mencionan que los AEs mantienen las siguientes características principales: 1) Simula el proceso evolutivo de una población, a través de una búsqueda iterativa en el conjunto de soluciones factibles del problema a resolver mediante una medida de calidad de evaluación de cada solución,

determinando si pasará a la siguiente generación de la población. 2) Son orientados a la aptitud, donde cada individuo será evaluado mediante una función de fitness de acuerdo a su representación genética, esta evaluación retornará su valor fitness o de adaptabilidad en el cual se distinguirá si el individuo seguirá siendo una solución factible del problema, que estará en la siguiente generación. 3) Variación impulsada, donde los individuos son sometidos a una serie de operaciones de variación genética que imitan los cambios en los genes, para la búsqueda de soluciones.

Rodríguez (2005) enfatiza que para llegar a la simulación del proceso evolutivo a través de una computadora, se deben seguir los siguientes pasos:

- Traducir una solución o individuo en un contexto real del problema, denominado fenotipo, y al contexto de la computación evolutiva, denominado genotipo.
- Hacer las operaciones que afecten a los individuos, tomando partes de cada uno para producir individuos nuevos que heredan características de los padres, por medio del cruzamiento y la mutación.
- Utilizar una función de aptitud, que de acuerdo a una medida de calidad evaluará a los individuos.
- Utilizar un mecanismo de selección de los mejores individuos de acuerdo a su valor de aptitud para reproducirse o cruzarse, y generar una descendencia.

Yu y Gen (2010) mencionan que para poder relacionar los principios de la teoría darwiniana de la selección natural con los conceptos anteriores descritos, se consideran:

- El proceso de selección supervivencia del más apto.
- Dos padres generan descendencias cruzamiento y recombinación.
- Pequeños cambios hechos en la descendencia o mutación.

Eiben y Smith (2003) mencionan que dentro del área del CE existen tres paradigmas caracterizados por la forma en que implementan la estrategia de evolución, éstas son:

- Programación evolutiva: Utilizada para resolver problemas de modelado, usando principalmente cruzamiento sobre sus poblaciones (Barbosa, 2014). Se basa en hacer evolucionar autómatas de estados finitos expuestos a una serie de símbolos de entrada. Se utiliza un operador de mutación para afectar los cambios de transición y una función de aptitud como un indicador para hacer los cambios (Rodríguez, 2005).
- Algoritmos genéticos: Usan cadenas basadas en un alfabeto finito como individuos de su población y son más utilizados para resolver problemas combinatorios de optimización (Eiben & Smith, 2003). Enfatizan la importancia del cruzamiento sobre la mutación (Rodríguez, 2005).
- Estrategias evolutivas: Usados para resolver problemas de optimización continua (Barbosa, 2014). Dada una solución, se realiza otra solución descendiente haciendo ajustes discretos a la primera, la segunda solución será evaluada para ver si es mejor que la primera, si es así la primera será sustituida por la segunda, repitiendo el proceso (Rodríguez, 2005).

2.8. Algoritmos Genéticos

El algoritmo genético (AG) fue originalmente concebido por Holland como un medio para estudiar el comportamiento adaptativo (Barbosa, 2014). Considerando la definición formal de Goldberg (1989) “los AG son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado” (p. 1).

Un AG es un método de búsqueda basada en la probabilidad. Bajo la condición de selección de individuos, el algoritmo converge en probabilidad al óptimo al aumentar el número de iteraciones (Eiben y Smith, 2003).

El funcionamiento de un algoritmo genético se ilustra en la Figura 5, donde se parte de la población inicial previamente generada, para proseguir a la selección de los padres.

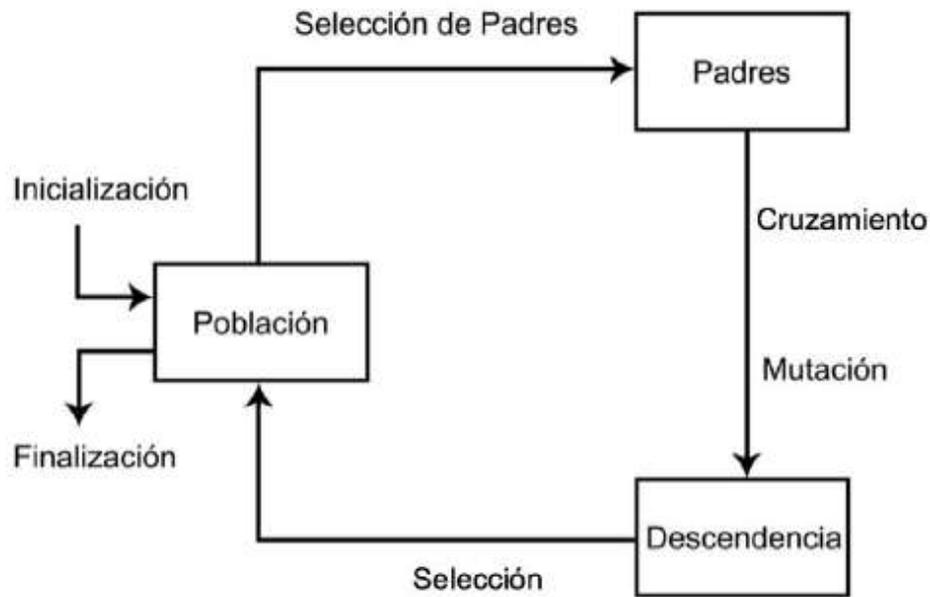


Figura 5. Esquema general de un algoritmo genético.

Se seleccionan al azar dos padres de la población de individuos; a los cuales posteriormente se les aplica el operador de cruzamiento. La función que desempeña este operador es mezclar las características de los individuos padres y producir hijos con aptitudes diferentes a ellos. Consecuentemente, a los individuos cruzados se les aplica otro operador genético: la mutación.

Existe una población de individuos que luchan por sobrevivir y reproducirse. Los individuos son evaluados en su ambiente para determinar una aptitud, la cual determina si sobreviven, y si se reproducen para obtener así una nueva generación de individuos. La población de individuos evoluciona al someterse a acciones aleatorias como cruce y mutación genética, así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados

(Chambers, 1998). Generalmente este criterio es una función que describe un problema de optimización.

La representación del individuo debe de estar relacionada con el problema a tratar. Los resultados obtenidos por el AG dependen del diseño de la representación del individuo, por ello su importancia. Para generar la población inicial, es importante que la aleatoriedad sea alta; entre más distintos sean los individuos en la población, mayor probabilidad habrá de encontrar un individuo que represente una solución óptima.

El problema por lo general es la búsqueda de soluciones, que puede ser demasiado difícil cuando el espacio de soluciones es grande y una búsqueda exhaustiva es prácticamente imposible (Chen y Yeh, 1997).

2.9. Optimización multiobjetivo

Muchos problemas de la vida real requieren de múltiples decisiones, pues son varios los criterios que deben de ser considerados para obtener el objetivo esperado, estos criterios pueden estar en conflicto, ya que se considera que optimizar un criterio puede convertir a otro en un resultado inaceptable (Zhou et al., 2011). Este tipo de problemas son llamados problemas de optimización multiobjetivo. Estos envuelven un número de funciones objetivos que deberán de ser maximizadas o minimizadas, donde varias condiciones se deben de considerar para obtener una solución satisfactoria (Deb, 2011).

Una razonable solución a los problemas multiobjetivo es investigar un conjunto de soluciones, cada una de las cuales, satisfagan los criterios en un nivel aceptable con respecto a los otros, o sea, sin que sea dominada por los demás (Konak, Coit & Smith, 2006).

Se dice que una solución x_1 domina a otra solución x_2 , si x_1 no es menor que x_2 en todos los objetivos, y que la solución x_1 es estrictamente mejor que x_2 en al menos

un objetivo. Al conjunto de soluciones no dominadas es llamado “Frente de Pareto” o “Conjunto de Pareto” (Deb, 2011).

En la Figura 6 se muestra un Frente de Pareto para el caso en que se intente minimizar y maximizar dos objetivos. Los elementos unidos con la línea representarán el Frente no-dominado, es decir, las soluciones no dominadas.

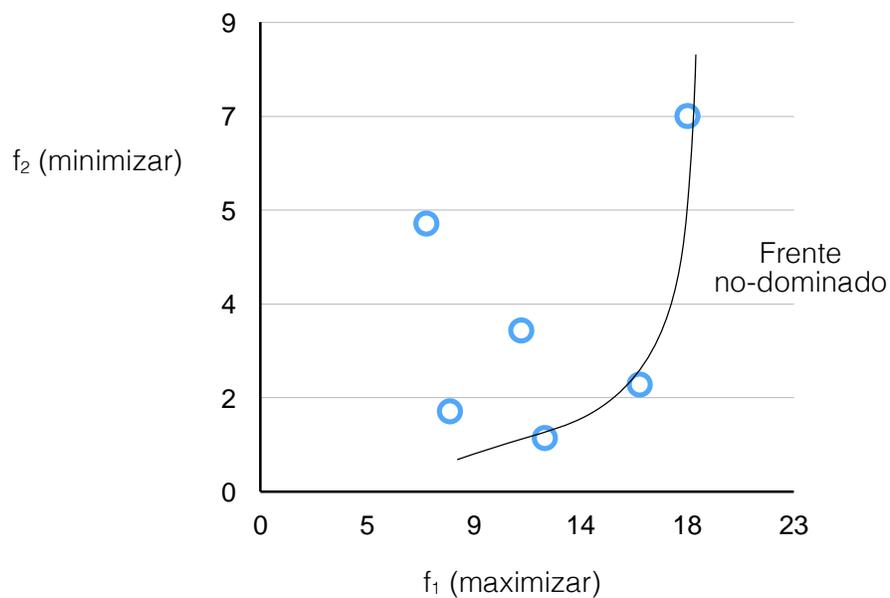


Figura 6. Frente de Pareto.

Los AEs son una selección ideal para resolver problemas multicriterio por su naturaleza basada en una población. Un número de óptimos de soluciones pueden, en principio, ser capturados de una población de un AE, permitiendo encontrar múltiples soluciones óptimas de Pareto en una corrida de simulación (Ghosh & Dehuri, 2005).

2.10. Algoritmo Genético de Ordenamiento No-dominado II (NSGA-II)

El algoritmo NSGA-II es utilizado para la asignación de tareas a las máquinas, de tal forma que el C_{max} y la energía consumida sean minimizadas. Este es un algoritmo genético multiobjetivo caracterizado por el elitismo y la distancia de apilamiento para mantener la diversidad de la población de tal manera que esto permita encontrar varias soluciones óptimas de Pareto como sea posible (Deb et al., 2002). Se encarga de generar una población de N individuos, donde cada uno representa una posible respuesta.

Los individuos evolucionan a través de los operadores genéticos para encontrar soluciones óptimas o casi óptimas. Son aplicados 3 operadores: Selección (usando un operador de torneo de apilamiento), cruzamiento y mutación para generar otros N individuos hijos. De la mezcla de estas dos poblaciones, surge una nueva población de medida $2N$.

Los mejores N individuos son tomados subsecuentemente de acuerdo a su valor de aptitud y después son ordenados para dividir la población en frentes no-dominados. Los individuos de los mejores frentes no-dominados son tomados, uno por uno, hasta completar N individuos. Es considerado un operador de apilamiento si no hay espacio para un frente completo en la nueva población.

De esta forma el algoritmo utiliza la base de los operadores genéticos y promueve al siguiente ciclo generacional las configuraciones que ocupan los mejores frentes, y más diversa, a través de distancias de apilamiento. Los individuos se desarrollan hasta que se cumpla el criterio de paro.

2.11. Funciones de deseabilidad

Obtener soluciones en el Frente de Pareto no resuelve el problema multiobjetivo, el tomador de decisiones aun tiene la tarea de seleccionar la solución más preferente de acuerdo a la información. Debido a esto hay, varias metodologías que incorporan

preferencias en el proceso de búsqueda (James & Coello, 2013), responsables de ofrecer un aviso acerca de seleccionar una justificable solución final del Frente de Pareto (Lu et al., 2014).

Una de estas metodologías son las funciones de deseabilidad (FD) que son responsables de mapear los valores de cada objetivo a deseabilidades, esto es, valores en una escala sin unidad en el rango $[0, 1]$. Consideremos que la imagen del objetivo f_i es $Z_i \subseteq \mathbb{R}$, entonces una FD es definida como una función $d_i: Z_i \rightarrow [0, 1]$ que especifica la deseabilidad de las diferentes regiones del dominio Z_i para el objetivo f_i (James & Coello, 2013).

El índice de deseabilidad (ID) es la combinación de los valores individuales de las FD, en un valor preferencial en el rango $[0, 1]$. Entre más alto sean los valores de las FD, más grande será el ID. Los individuos en el frente final con mas alto ID deberán ser los mejores candidatos a ser seleccionados por el tomador de decisiones.

2.12. Optimización multiobjetivo en las industrias

Los métodos tradicionales de optimización han sido incapaces de hacer frente con las complejidades de muchos problemas del mundo real abordados por la simulación. Los AEs han sido utilizados en dominios tales como el diseño de ingeniería, planeación operativa y la planificación (Persson, Andersson, Grimm, & Ng, 2008).

Muchos problemas en ingeniería tienen múltiples objetivos, incluyendo sistemas de diseño y la optimización de la fiabilidad, aquí puede haber interesantes y exitosas implementaciones de AGs multiobjetivo para esta clase de problemas (Konak et al., 2006).

Los problemas del mundo real de optimización en la industria usualmente se concentran en encontrar óptimos niveles de eficiencia en la compleja línea de

producción tratando de conseguir simultáneamente minimizar los plazos de entrega del producto y maximizar el rendimiento del sistema de producción (Persson et al., 2008).

Para el sector industrial es de gran interés el desarrollo de herramientas que reflejen sus necesidades y que puedan entregar soluciones en un tiempo razonable (Baesler, Ceballos & Ramírez, 2006).

En el proceso de fabricar un producto se deben tomar decisiones basadas en una gran cantidad de características y comportamientos diferentes (Augusto & Granada, 2007).

Se puede ver a la toma de decisiones como un problema de optimización donde generalmente se involucran varios objetivos (Toscano, 2001). Por ejemplo al considerar el problema de producir un producto industrial, por un lado se desea maximizar el desempeño, por otro lado se desea minimizar el costo de manufacturarlo, en lo que respecta a los tiempos y energía consumida por la maquinaria.

Para tratar este tipo problemas, la literatura se ha apoyado en las técnicas del computo bioinspirado, debido al repertorio de algoritmos con características ideales para resolver los problemas multiobjetivos (Toscano, 2001), tales como los algoritmos evolutivos, tratándose de métodos robustos de búsqueda que permiten tratar los problemas de optimización (Estévez, 1997).

La mayoría de las investigaciones para aplicar los AEs en problemas de optimización del mundo real, se han enfocado en diseñar un mecanismo adecuado para el manejo de las restricciones adaptándolo a un algoritmo en particular (López & Mezura, 2007), tales como los algoritmos genéticos.

Los AG poseen un gran rendimiento para tratar con problemas de optimización (Eiben & Smith, 2003), al predisponer en encontrar los elementos del Conjunto de Pareto que serán las mejores soluciones considerando las funciones objetivos (Lagunas, Moo & Ortiz, 2013).

2.13. Calendarización consciente del consumo de energía en un Taller de Flujo

Investigaciones de nuevos equipos y hardware pueden ciertamente contribuir al ahorro de energía (Mori et al., 2011). El uso de técnicas “blandas” para lograr el mismo objetivo es siempre una opción efectiva (Blum et al., 2012). Los avances en calendarización (programación de la producción) pueden jugar un rol importante en reducir el consumo de energía en los procesos de manufactura. Metaheurísticas, tales como, los algoritmos genéticos (Liu et al., 2015), optimización por enjambre de partículas (Nilakantan et al., 2015), y recocido simulado (Wang et al., 2015) han tenido una gran popularidad para el diseño de los sistemas de producción.

Sin embargo, estudios de los problemas de calendarización en un taller de flujo para el ahorro de energía parecen ser limitados (Liu et al., 2014; May et al, 2015). De hecho, más sistemas de producción que permiten cambios en la velocidad de las máquinas pertenecen a la industria de la ingeniería mecánica y usualmente adquieren la configuración de taller de trabajo en vez de taller de flujo (Zhang & Chiong, 2016).

Uno de los primeros intentos de reducir el consumo de energía a través de la calendarización de la producción puede ser encontrado en el trabajo de Mouzon et al. (2007). Los autores recolectan estadísticas operacionales de cuatro máquinas en un taller de flujo, encontrando que las máquinas sin cuello de botella consumen un considerable aumento de energía cuando están ociosas. Como resultado, ellos proponen un marco de trabajo para la calendarización en los eventos de encendido y apagado que controlan las máquinas para lograr una reducción de la energía total consumida.

Dai et al. (2013) aplica esta estrategia de encendido/apagado en un taller de flujo flexible, obteniendo soluciones satisfactorias al minimizar la energía total consumida y el tiempo de terminación de los trabajos.

Zhang & Chiong (2016) analizan el balance entre la minimización del C_{max} , una medida del nivel de servicio y el consumo de energía en un taller de flujo por permutación con una secuencia de dos máquinas. Los autores desarrollan un modelo lineal entero mixto para la optimización multiobjetivo, para encontrar el Frente de Pareto compuesto por el consumo de energía y el C_{max} .

Hecker et al. (2013) usan algoritmos evolutivos para calendarizar en un taller de flujo híbrido de no-espera, la asignación de tareas en líneas de producción de pan, usando la optimización por enjambre de partículas y optimización por colonia de hormigas.

Hecker et al. (2014) usan un algoritmo genético modificado, optimización por colonia de hormigas, el procedimiento de búsqueda aleatoria, para estudiar el C_{max} y el tiempo de ocio total de las máquinas en un modelo de taller de flujo por permutación.

Liu and Huang (2014) estudian un problema de calendarización con máquinas de procesamiento por lotes en un taller de flujo híbrido con un criterio relacionado con la energía y la tardanza total. Los autores aplican el algoritmo NSGA-II.

CAPÍTULO III. METODOLOGÍA

3.1. Desarrollo de la metodología

Para cumplir con los objetivos propuestos y conseguir la respuesta a la pregunta de investigación se utilizó la siguiente metodología:

Primero se analizó el modelo del entorno de producción real, a través de una estancia de seis meses en la planta productiva dando seguimiento a las actividades de planeación y producción. Se recolectan datos de los pedidos, materia prima, características de las máquinas, entre otras.

Se formaliza el modelo bajo estudio y se identifican las características de los procesos, recursos y los criterios de optimización. Se realiza una revisión de la literatura acerca de la programación de la producción identificando las distintas contribuciones relacionadas con el caso real.

Se analizan casos especiales dentro del modelo teórico y su utilidad tomando en cuenta el número de etapas, máquinas y restricciones adicionales. Se aplican métodos heurísticos y metaheurísticos para desarrollar algoritmos que contribuyan a la solución del problema tomando en cuenta investigaciones previas relacionadas.

Se define el algoritmo genético de optimización multiobjetivo a utilizar, la representación de una solución que utilizará el algoritmo, los mecanismos de las funciones objetivo para evaluar las soluciones. Posteriormente se definen los operadores genéticos tomados de la literatura, utilizados para llevar a cabo el proceso evolutivo. Por último, se define el tamaño de la población y el número máximo de iteraciones en que se espera obtener soluciones satisfactorias.

Una vez diseñado el algoritmo, se prosiguió con la implementación. La fase de experimentación se realizó mediante simulaciones computacionales del algoritmo, donde se combinan los diferentes operadores de cruce y mutación para conseguir

una configuración de parámetros que obtuviera buenos resultados para cada caso.

Al terminar, se evalúa el desempeño del algoritmo desarrollado a través de un análisis estadístico. Primero es aplicado un análisis de varianza para conocer cuáles son los factores que más afectan el desempeño.

Posteriormente las soluciones son evaluadas mediante el Frente de Pareto, obteniendo un conjunto de mejores soluciones que satisfacen mejor ambos objetivos. Cada solución en el frente es evaluada mediante funciones de deseabilidad, de esta forma obtuvimos la mejor solución definitiva.

La solución obtenida indicará la mejor combinación de parámetros para cada caso, así como los valores de cada objetivo.

Finalmente, se comparan los mejores resultados del algoritmo con los resultados de la industria de estudio y se presentan las conclusiones.

3.2. Diseño conceptual del algoritmo

3.2.1. Modelo

Se considera el siguiente problema del TFH (Baker, 1974; Pinedo, 2002; Yaurima et al., 2008): Un conjunto de N de n trabajos $N = \{1, 2, \dots, n\}$, disponibles en el tiempo inicial 0, deberán de procesarse en un conjunto M de m etapas de producción consecutivas $M = \{1, 2, \dots, m\}$, con los objetivos: 1) Minimizar el tiempo de terminación de los trabajos (C_{max}) y 2) Minimizar el consumo de energía de las máquinas utilizadas (E_{op}). En cada etapa $i \in M$, hay un conjunto $M_i = \{1, 2, \dots, m_i\}$ de máquinas paralelas no relacionadas o idénticas, donde $|M_i| \geq 1$. Los trabajos son divididos en un conjunto de tareas $T_j = \{1, 2, \dots, m_j\}$ que deberán de procesarse exactamente en una máquina de cada etapa.

Cada tarea $t \in T_j$ consiste de un conjunto de lotes de 10 Kg cada uno. Cada tarea puede tener diferente cantidad de lotes.

Cada trabajo $j \in N$ pertenece a un determinado grupo G_i ; las tareas de los trabajos en el mismo grupo pueden ser procesados al mismo tiempo en la etapa $2 \in M$.

Sea $p_{i,l,q}$ el tiempo de procesamiento de la tarea $q \in T_j$, en la máquina $l \in M_i$ dentro de la etapa i . Sea $S_{i,l,qp}$ el tiempo de configuración (ajuste) en la máquina l de la etapa i para procesar la tarea $p \in T_j$, después de haber procesado la tarea q .

Para cada máquina $l \in M_i$ se considera un buffer ilimitado para almacenar temporalmente las tareas en espera, denominado “trabajo en proceso” (*work in process*).

Usando la notación de tres campos mejor conocida $\alpha | \beta | \gamma$ para los problemas de calendarización, introducida en Graham et al. (1979), el problema es denotado como sigue:

$$FH6, ((RM^{(2)}), (PM^{(i)})_{i=3^4}, RM^{(5)}, (PM^{(i)})_{i=6^7}) | S_{sd2}, S_{si3^7} | C_{max}, E_{op}$$

La notación describe un taller de flujo híbrido de 6 etapas, con máquinas paralelas no relacionadas en las etapas 2 y 5, y máquinas paralelas idénticas en la etapas 3, 4, 6 y 7 (Figura 4). Se estiman tiempos de configuración dependientes de la secuencia en la etapa 2, e independientes en las etapas 3 a la 7. Se consideran los objetivos de minimizar el tiempo de terminación (C_{max}) y la energía consumida (E_{op}).

3.2.2. Representación del cromosoma

La representación de una solución (cromosoma) se da mediante una cadena de enteros, la cual es una permutación de números $1, 2, \dots, n$, y cada entero representa un lote (10 Kg) de un determinado trabajo. La enumeración de cada lote es dado entre un rango especificado por el orden de los trabajos y la cantidad de lotes.

En la Tabla 1 se muestra un ejemplo. Los 9 lotes del trabajo 1, son enumerados del 1 al 9, la enumeración de los 12 lotes del trabajo 2 continúa a partir de donde terminó el conteo anterior, hasta completar 12 enumeraciones, es decir, del 10 al 21. El proceso se repite hasta terminar la enumeración de los lotes de último trabajo.

Tabla 1. Enumeración de lotes por trabajo.

Trabajo	Lotes	Rango
1	9	1 - 9
2	12	10 - 21
3	16	22 - 37
4	15	38 - 52
5	10	53 - 62

La Figura 7, muestra un ejemplo de representación del cromosoma, su estructura se compone de columnas de acuerdo al número de máquinas por etapa (m_i), los lotes enumerados de cada trabajo se distribuyen aleatoriamente entre los espacios de las columnas (genes) calculados, dividiendo la cantidad de lotes entre la cantidad de máquinas por etapa.

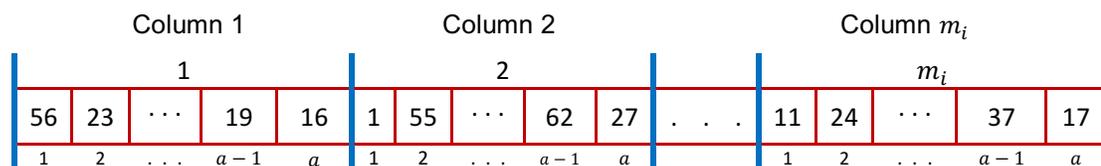


Figura 7. Ejemplo de representación del cromosoma.

En este ejemplo, se consideran 4 (máquinas por etapa) / 62 (lotes) = 15.5 (espacios por columna), al tener decimales el resultado se redondea al entero superior más cercano, terminando con 16 (k) espacios entre columnas.

En cada columna del cromosoma, las tareas de cada trabajo se componen de

agrupaciones de lotes formadas de acuerdo a las capacidades de procesamiento en Kg de las máquinas de la primera etapa y la compatibilidad de procesamiento de los trabajos. En la Figura 8 se muestra un ejemplo, tomando como referencia la primera columna del cromosoma de la Figura 7.

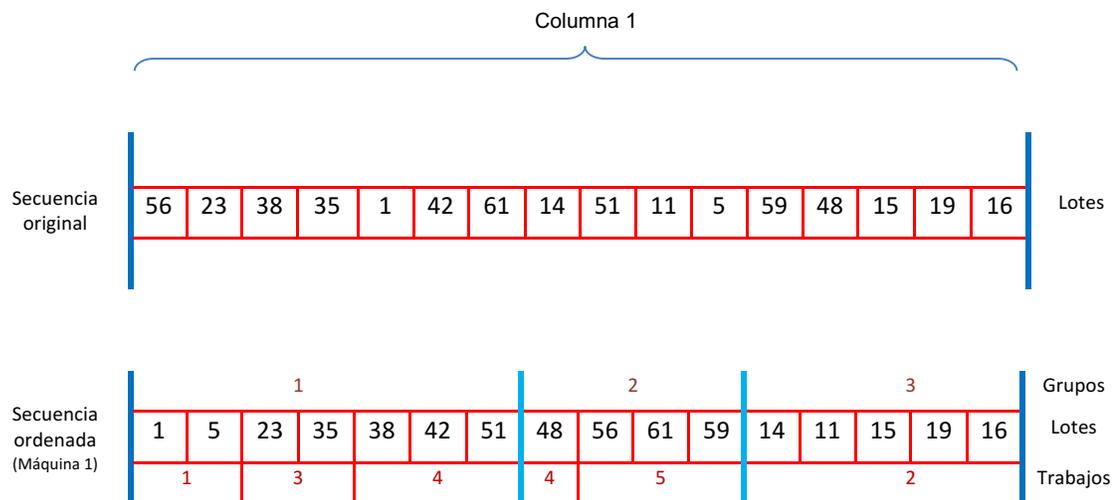


Figura 8. Ejemplo de obtención de tareas.

La secuencia dispone de las siguientes características:

- 1) Grupos: Indica el conjunto de lotes que la máquina procesará al mismo tiempo. Son indicados dentro de las separaciones azul claro.
- 2) Trabajos: Indica el número de trabajo al cual pertenece cada subgrupo de lotes. Son indicados dentro de las separaciones rojas.
- 3) Lotes: Secuencia de lotes de la columna que pertenecen al cromosoma. La secuencia acomodada son los lotes agrupados de acuerdo al punto 2.

Se consideran los siguientes supuestos:

- 4) La máquina dispone de 70 Kg (7 lotes) de capacidad de procesamiento.
- 5) El primer grupo lo componen 7 lotes del trabajo 1, 3 y 4. Al ser llenada la capacidad de la máquina, se forma el segundo.

6) El tercer grupo lo componen 5 lotes del trabajo 2, como este tipo de trabajo no es compatible con ninguno de los 4 trabajos restantes, los lotes se procesan por separado del segundo grupo.

La cantidad de tareas representan los subgrupos obtenidos en la fila “Trabajos” de la Figura 8. Las tareas serán asignadas a las demás etapas de acuerdo a su orden de terminación dentro de la primera etapa. En las etapas posteriores, las tareas permanecerán de acuerdo a como fueron ordenadas y serán procesadas de forma individual, asignadas a las máquinas de acuerdo a su disponibilidad; de otra forma las tareas deberán de esperar dentro del buffer.

La Figura 9 indica un ejemplo de lo que sería la asignación de las tareas obtenidas de acuerdo a la “Secuencia ordenada” de la columna del cromosoma de la Figura 8. En este ejemplo se consideran las primeras 2 etapas, con 2 máquinas en cada una. En la etapa 1, se encuentran dos máquinas con capacidad para 7 lotes, o sea, 70 Kg para procesar.

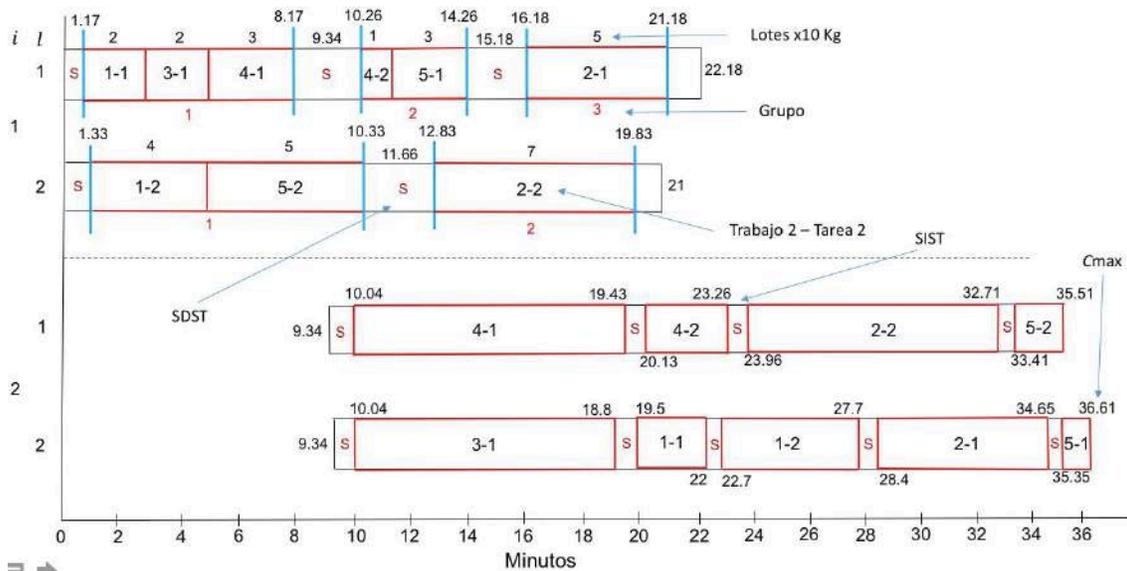


Figura 9. Ejemplo de calendario.

De acuerdo al cromosoma de la Figura 8, se da la asignación de tareas a las máquinas de cada etapa. En la etapa 1, cada máquina procesa una determinada cantidad de grupos de tareas. En la máquina 1 se observa el orden de asignación de las tareas de la “Secuencia ordenada” del cromosoma. Antes de asignar el primer grupo de tareas, se estima un tiempo de configuración para preparar la máquina. Primeramente se asignan las 3 tareas del grupo 1, lo cual todas las tareas pertenecientes al grupo se procesan al mismo tiempo. Al terminar se indica un tiempo de configuración dependiente de la secuencia (SDST) (Tabla 7) necesario para dejar la máquina lista para procesar el grupo 2 compuesto de 2 tareas (4-2 y 5-1).

En la etapa 2, las tareas son asignadas de acuerdo al orden de terminación que tuvieron en la primera etapa. Las tareas que estén en espera de ser procesadas, se asignarán a la máquina más próxima en terminar, dando prioridad a las tareas con mayor tiempo de procesamiento. Las tareas son asignadas a la etapa 2 a partir del tiempo de configuración necesario pasarlas al buffer. Cada tarea se asigna de forma individual en la etapa 2 con un tiempo de configuración independiente de la secuencia (SIST). En esta etapa y posteriores el tiempo de procesamiento es diferente para cada tipo de tarea de acuerdo al tipo de trabajo que pertenecen y cantidad de lotes (Tabla 5).

Se observa que las tareas 1-1 y 1-4 fueron asignadas primero a las máquinas en la etapa 2, ya que la 3-1, tiene menor tiempo de procesamiento que la otras dos. Posteriormente es asignada la tarea 4-2 de acuerdo a su tiempo de terminación, a la máquina mas próxima en terminar. En el tiempo que las tareas 3-1 y 4-2 finalizan en la etapa 2, las tareas 1-2, 5-2, 5-1, 2-2 y 2-1, ya finalizaron en la etapa 1, por lo que se encuentran en el buffer de espera. En este instante se hace una evaluación de cual tarea de las 5 tendrá mayor tiempo de finalización para ser asignada a la máquina mas próxima en terminar. Cuando una máquina va a estar disponible, la evaluación se repite con las tareas en el buffer. Esta estrategia se repite en las etapas posteriores.

En este ejemplo, se muestran las primeras dos etapas del modelo, de igual forma la asignación de las tareas continua individual en las etapas 3 a la 7, y en grupo solo en la etapa 2 (la primera del modelo).

3.2.3. Inicialización, evaluación y criterio de paro

La población se forma de 20, 30 y 50 individuos con sus genes obtenidos de manera aleatoria y uniforme. Las diferentes medidas de la población fueron consideradas por su frecuencia en la literatura para este tipo de problemas con cargas de trabajos pequeñas (Costa, et. al., 2013; Fadae & Zandieh, 2013). La función objetivo calcula la aptitud (Cmax y Energía) de cada individuo. Como criterio de paro se considera 50 generaciones de acuerdo a varias referencias en que el algoritmo genérico converge al óptimo con el número de poblaciones estimadas, también considerando que el tiempo de experimentación no sea tan exhaustivo (Barbosa, 2014; Dhingra, et al., 2014). El algoritmo propuesto asume que la asignación del trabajo a la primera máquina disponible podría resultar en el mejor tiempo de completar el trabajo (Ruiz y Maroto, 2006).

En un TFH con máquinas paralelas no relacionadas o idénticas, si la primera máquina disponible es muy lenta para un trabajo dado, asignarlo a esta máquina resulta en un tiempo retrasado, comparado con la asignación a otras máquinas. Ahora, con la consideración de tiempos de ajuste se empeoraría el resultado. Para resolver esta dificultad, el trabajo se asigna a la máquina que lo finaliza en el menor tiempo posible en cada etapa, tomando en consideración las diferentes velocidades, tiempos de ajuste y disponibilidad de máquinas.

El cálculo del tiempo de completar los trabajos es el siguiente (Baker, 1974; Pinedo, 2002; Yaurima et al., 2008): Sea $S_{i,l,q}$ el tiempo de ajuste de la máquina l en la etapa i para procesar la tarea $p \in T_j$ después de haber procesado la tarea $q \in T_j$. Sea $p_{i,q}$ el tiempo de procesamiento de la tarea $q \in T_j$. El tiempo $C_{i,p}$ de completar la tarea p en la etapa i se calcula de acuerdo a:

$$C_{i,p} = \min_{1 \leq l \leq m_i} \{ \max\{C_{i,p} + S_{i,l,q} ; C_{i-1,p}\} + p_{i,l,q} \}$$

Una vez que las tareas son asignadas a las máquinas en las etapas, el tiempo máximo de completar todas las tareas de los trabajos se calcula como:

$$C_{max} = \max_{p=1}^Q \{C_{m,p}\}$$

Q indica la cantidad total de tareas por todos los trabajos. $C_{m,p}$ indica el tiempo de completado de la tarea $p \in T_j$, en la última etapa (m). El consumo total de energía es calculado como:

$$E_{op} = \sum_{t=1}^m p_{i,l,q} \cdot E_{i_l}$$

E_{op} indica el total de la energía consumida por todas las tareas durante todo el calendario, E_{i_l} indica el consumo de energía eléctrica de la máquina l en la etapa i , $p_{i,l,q}$ se refiere al tiempo de procesamiento de la tarea $q \in T_j$.

3.2.4. Selección, cruzamiento y mutación

Para la selección de padres, se considera el torneo binario (*binary tournament*): De la población se toman aleatoriamente dos individuos y se selecciona el primero con mayor aptitud. Para evitar la convergencia a un óptimo local, se incorpora un operador de mutación, el cual permite introducir material genético perdido y

variabilidad en la población. Se consideran tres operadores de mutación (*insert*, *desplacement* y *switch*), y sus probabilidades de 0.05, 0.1 y 0.2; los cuales se usan más frecuentemente en la literatura (Michalewicz, 1996; Gen, 1997; Ruiz & Maroto, 2006; Costa et. al., 2013; Fadae & Zandieh, 2013):

1. INSERT: Se selecciona un valor de forma aleatoria y se inserta en una posición arbitraria (Figura 10).

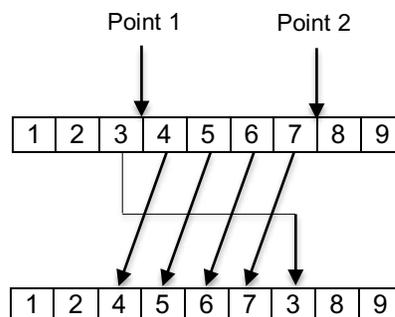


Figura 10. Mutación Insert.

2. DESPLACEMENT: Es una generalización de la mutación de inserción en la que, en vez de mover un solo valor, se cambian de lugar varios a la vez (Figura 11).

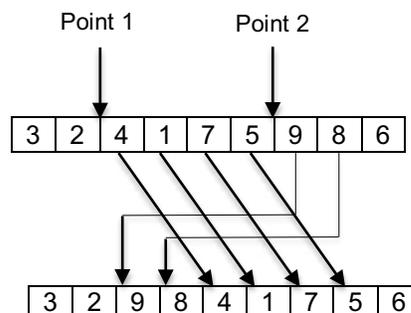


Figura 11. Mutación Displacement.

3. SWITCH: Se realiza un intercambio simple entre dos genes adyacentes. Se selecciona aleatoriamente un punto. El gen correspondiente se intercambia con su sucesor inmediato a la derecha. Si se selecciona el último gen, entonces éste se intercambia con el primero (Figura 12).

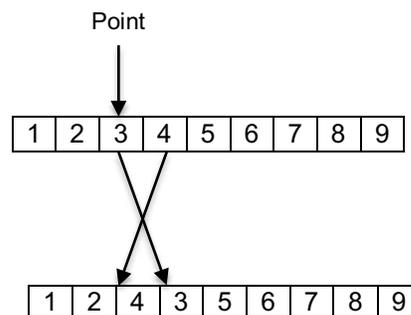


Figura 12. Mutación Switch.

Un operador de cruzamiento genera nuevas soluciones a partir de dos secuencias. El objetivo es generar soluciones con mejores valores para C_{max} .

Se consideran 2 operadores de cruzamiento conocidos y sus probabilidades de 0.5, 0.7 y 0.9 mas utilizadas en la literatura (Seidgar et. al., 2009; Costa et. al., 2013; Dhingra, et al., 2014):

1. OX - Order Crossover (Gog et al. 2011; Hwang, 2002; Larrañaga et al. 1999; Prins, 2004; Starkweather et al. 1991): Son seleccionados dos puntos de corte, la parte del primer padre localizado entre esos dos puntos de corte es copiado al hijo. Las posiciones siguientes en el hijo son llenadas comenzando después del segundo punto de corte, considerando los vértices en el orden encontrado en el segundo padre (Figura 13):

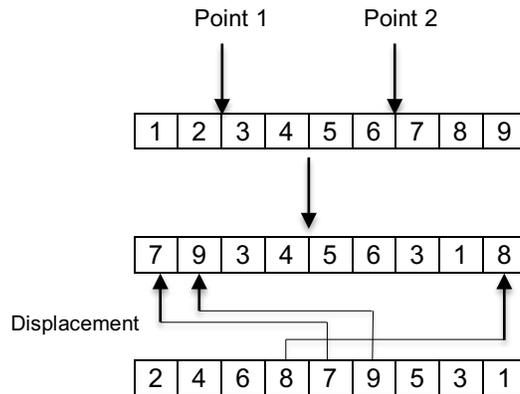


Figura 13. Operador de cruzamiento OX.

2. PMX - Partially Mapped Crossover (Larrañaga et al., 1999; Tan et al., 2001): El operador PMX es similar al OX. Se tienen dos puntos de corte. La parte del primer padre entre esos dos puntos de corte es copiado al hijo. Después las partes siguientes del hijo son llenadas con los vértices siguientes de tal forma que sus posiciones absolutas son heredadas como sea posible del segundo padre (Figura 14).

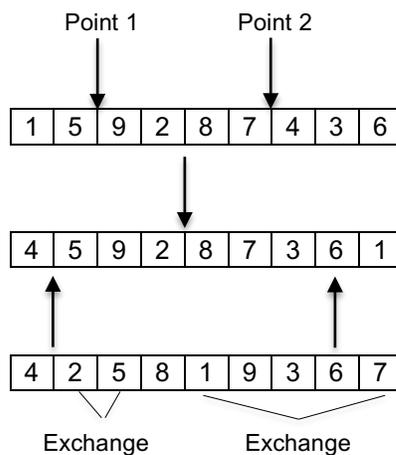


Figura 14. Operador de cruzamiento PMX.

3.3. Calibración del algoritmo

La calibración es un procedimiento para elegir los parámetros del algoritmo que proporcionan el mejor resultado en la variable respuesta. Se realiza a través de un experimento computacional. Incluye los siguientes pasos:

- a) Ejecutar cada instancia o carga de trabajo con todas las posibles combinaciones de parámetros.
- b) Obtener la mejor solución.
- c) Calcular el incremento relativo sobre la mejor solución (IRMS).
- d) Aplicar análisis de varianza (ANOVA) con 95% de nivel de confianza para encontrar los factores con más influencia en la solución.
- e) Seleccionar el conjunto de los mejores parámetros de los factores considerando las soluciones en el Frente de Pareto.

3.3.1. Diseño del experimento

En el experimento se analiza el efecto a la función objetivo (variable respuesta) de los factores: tamaño de la población, tipos de cruzamiento y mutación con sus probabilidades. Para calibrar el algoritmo se analizan los siguientes niveles de los factores mostrados en la siguiente tabla:

Tabla 2. Parámetros usados para la calibración.

Factores	Niveles
Población	20, 30 y 50
Operadores de Cruzamiento	OX, PMX
Prob. de Cruzamiento	0.5, 0.7, 0.9
Operadores de Mutación	Displacement, Exchange, Insertion
Prob. de Mutación	0.05, 0.1, 0.2
Selección	Torneo Binario
Criterio de paro	50 iteraciones.

Se tienen $3 \times 2 \times 3 \times 3 \times 3 = 162$ diferentes algoritmos o combinaciones. Cada combinación es ejecutada 30 veces, siendo un total de $162 \times 30 = 4,860$ experimentos. Para cada una de las 30 ejecuciones por combinación, hay una carga de trabajo diferente, teniendo un total de 30 cargas de trabajo que son repetidas en cada combinación.

Al considerar cada caso de máquinas por etapa (2, 4 y 6) se tiene un total de $4,860 \times 3 = 14,580$ experimentos.

En cada una de las 30 ejecuciones por combinación, se obtiene el mejor individuo a través de funciones de deseabilidad, obteniendo los mejores 30, uno por cada ejecución. Posteriormente valores de cada objetivo (Cmax y Energía) son usados para obtener un promedio de los 30, obteniendo un solo valor para cada objetivo. Este valor promedio representa una combinación de las 162.

La eficiencia de cada algoritmo o combinación es calculada de acuerdo al incremento relativo sobre la mejor solución (IRMS), mediante la siguiente fórmula:

$$IRMS = \frac{Heu_{sol} - Best_{sol}}{Best_{sol}} \times 100$$

dónde Heu_{sol} es el valor obtenido de la función objetivo por el algoritmo en cuestión, $Best_{sol}$ es el mejor valor obtenido durante la ejecución de todas las posibles combinaciones de parámetros.

3.3.2. Especificaciones computacionales

El equipo de cómputo utilizado para el desarrollo del AG bi-objetivo fue el siguiente: PCs con procesador Intel Core i3, 3.30GHz, 4 GB de memoria principal y sistema operativo Windows 7 Profesional SP1 de 64 bits. El lenguaje de programación utilizado para codificar el algoritmo fue R, utilizando como entorno de desarrollo

RStudio. Se eligió este lenguaje de programación por sus múltiples ventajas, entre ellas la escalabilidad que proporciona (R-project.org, 2016).

3.3.3. Configuración de parámetros de entrada

Los parámetros de entrada corresponden a un entorno de producción real. Se considera el modelo de recursos de 6 etapas, con 3 casos de 2, 4 y 6 máquinas por etapa. Las Tablas 3 y 4 presentan la capacidad de producción de las máquinas en cada etapa. La capacidad de producción indica la cantidad de trabajo (Kg de tortilla) que puede ser procesado en la máquina. Las etapas 3, 4, 6 y 7 corresponden a máquinas idénticas, las etapas 2 y 5 contienen máquinas no relacionadas. En cada etapa se consideran 6 máquinas enumeradas del 1 al 6, para procesar cada tipo de trabajo.

Tabla 3. Capacidad de procesamiento de máquinas (Kilogramos x minuto) en la etapa 2.

Etapa	2					
Máquinas	1	2	3	4	5	6
Trabajos	80	80	100	130	160	200

Tabla 4. Capacidad de procesamiento de máquinas (Kilogramos x minuto) en las etapas 3 a la 7.

Etapa <i>i</i>	3	4	5		6	7
Máquinas	1-6	1-6	1-3	4-6	1-6	1-6
1	1.3	200	7.2	9.6	9.6	9.6
2	1.3	200	6.4	8.8	8.8	8.8
Trabajos	3	0.44	200	2.4	4	4
	4	0.31	200	3.2	4.8	4.8
	5	1.3	200	4	8.8	8.8

En la etapa 2 se indica que todos los trabajos tendrán un tiempo de procesamiento igual de acuerdo a la máquina asignada. Para cada una de las 30 ejecuciones por combinación de parámetros se considera una carga compuesta de 5 trabajos para producir 5 tipos de tortillas de las más demandadas en la industria de estudio. La siguiente tabla muestra las características.

Tabla 5. Características de la carga de trabajo.

Trabajo <i>j</i>	Nombre Clave	Tipo	Grupo	Limites de lotes
1	H_Regular	1	1	356 - 446
2	H_Integral	2	2	34 - 61
3	Class_18pcs.	3	1	7 - 10
4	H_T50	4	1	13 - 22
5	H_MB	5	1	12 - 20

A cada trabajo le corresponde un tipo, grupo y la cantidad de lotes a producir entre un determinado rango, el cual es obtenido de las desviaciones estándar del promedio mensual de producción en Kg de tortilla durante un año.

Los lotes indican la cantidad de 10 Kg de tarea que deben de ser procesadas. Por ejemplo, las tareas con 1 lote indican 10 Kg a producir, con 5 lotes indican 1 tarea de 50 Kg. Los grupos indican las tareas de los trabajos que se pueden procesar al mismo tiempo al pertenecer al mismo grupo. La formación de grupos se debe a que los trabajos con el mismo tipo de masa se pueden combinar, para este caso los trabajos de tipo 1, 3, 4 y 5 pertenecen al grupo 1, y el tipo 2 forma un segundo grupo.

Los grupos formados podrán de ser procesados simultáneamente sólo en las máquinas no relacionadas de la etapa 2, considerando que el peso total de las tareas sea menor o igual a la capacidad en Kg de las máquinas.

Para cada una de las 30 ejecuciones por combinación, considerando cada carga de trabajo, se aplica una variación en las capacidades de procesamiento por minuto de las máquinas de las etapas 2 y 5 (Tablas 3 y 4). Se estima que en la etapa 2 hay 6

máquinas con diferentes capacidades, en la etapa 5 se tienen 6 máquinas, 3 de diferentes capacidades. Considerando cada caso de máquinas por etapa se selecciona la cantidad y capacidad de las máquinas de la siguiente forma: Las máquinas son enumeradas del 1 al 6 y son seleccionadas de forma aleatoria uniforme. Por ejemplo, en el caso de 2 máquinas por etapa, se realiza una selección aleatoria de 2 máquinas de las 6 de diferente capacidad, igual en los casos de 4 y 6 máquinas por etapa. Por lo tanto, considerando lo anterior, se estima una carga de trabajo diferente para cada una de las 30 ejecuciones. De igual manera para cada ejecución le corresponde una variación en las capacidades de las máquinas de las etapas 2 y 5, considerando cada caso de máquinas por etapa.

Estas variaciones se mantendrán en cada una de las 30 ejecuciones de cada combinación, Por ejemplo la ejecución 2 de la combinación 1, tendrá la misma carga de trabajo y variación de capacidades de máquinas, que la ejecución 2 de la combinación 2, así sucesivamente hasta la combinación 162.

La Tabla 6 muestra los tiempos de procesamiento en minutos por cada 10 Kg de tortilla de cada tipo de trabajo. Se observa que en las etapas 2 y 4 todas las máquinas tienen el mismo tiempo de procesamiento, debido a que no se depende del tipo de trabajo.

Tabla 6. Tiempo de procesamiento en minutos.

Etapa <i>i</i>	2	3	4	5	6	7		
Máquinas	1-6	1-6	1-6	1-3	4-6	1-6	1-6	
Trabajos	1	1	1.25	45	1.43	1.07	12.15	17.25
	2	1	1.25	45	1.48	1.11	12.15	17.25
	3	1	4.38	45	3.69	2.77	21.38	26.48
	4	1	3.13	45	3.00	2.25	24.50	29.60
	5	1	0.42	45	2.28	1.14	18.24	23.34

La Tabla 7 muestra los tiempos de configuración (ajuste) de las máquinas en cada etapa.

Tabla 7. Tiempos de configuración de las máquinas por etapa.

Etapa i	2	3	4	5	6	7
Tiempo de configuración (minutos)	st_2	0.7	0.7	0.5	0	0

En las etapas 3 a las 7, el tiempo de configuración es independiente de la secuencia de tareas, tipo de trabajo y peso. En la etapa 2, el tiempo de configuración es calculado de acuerdo a:

$$st_2 = \frac{35 + \left(\frac{t_w}{10} \times 5\right)}{60}$$

t_w es la suma total en Kg de las tareas a ser procesadas simultáneamente en las máquinas. Estas tareas deberán de pertenecer a los trabajos que pertenecen a una misma agrupación. Por ejemplo, si el total en Kg de las tareas es de 80 Kg, el tiempo de configuración será de 1.25 minutos, que es igual a 75 segundos.

La Tabla 8 muestra el consumo de energía de las máquinas en cada etapa. El consumo es independiente del tipo de trabajo a ser procesado. En la etapa 4 no se consume energía debido a que las bolas de masa reposan por un periodo de tiempo.

Tabla 8. Consumo de energía de las máquinas por etapa (kW x minuto).

Etapa i	2	3	4	5	6	7	
Máquina l	1	0.09	0.03	0	0.62	0.04	0.12
	2	0.09	0.03	0	0.62	0.04	0.12
	3	0.11	0.03	0	0.62	0.04	0.12
	4	0.14	0.03	0	0.9	0.04	0.12
	5	0.23	0.03	0	0.9	0.04	0.12
	6	0.23	0.03	0	0.9	0.04	0.12

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. Análisis de varianza

El análisis de varianza es aplicado para evaluar la diferencia estadística entre los resultados experimentales y observar el efecto de los parámetros en la calidad de los resultados. Es utilizado para determinar los factores que tienen un efecto significativo y conocer cuáles son los que influyen en la solución. Los parámetros del problema son considerados como factores y sus valores como niveles. Se asume que no hay interacción entre los factores.

La Tabla 9 muestra los resultados del análisis de varianza para el caso de 2 máquinas por etapa a un nivel de confianza del 95%.

Tabla 9. Análisis de varianza del IRSM para el caso de 2 máquinas por etapa.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A: Crossover	1747	1	1747	9.755	0.00213
B: Mutation	11967	1	11967	66.833	9.58e-14
C: Cr. Prob.	1519	1	1519	8.481	0.00411
D: Mu. Prob.	654	1	654	3.652	0.05782
E: Population	1732	1	1732	9.673	0.00222
Residuals	27934	156	179		

El valor del F-Ratio es la relación entre la media de cuadrados (*Mean Square*) del factor y la media de cuadrados de residuos. Un alto F-Ratio significa que este valor afecta significativamente la variable respuesta. El valor del P-Value muestra la significancia estadística de los factores. Los valores cuyos P-Value es menor a 0.05, tienen estadísticamente un efecto significativo en la variable respuesta (IRMS). De acuerdo al F-Ratio y P-Value, el factor más influyente en el caso de 2 máquinas por etapa es el operador de mutación y en segundo lugar es el operador de cruzamiento.

En el caso de 4 máquinas por etapa, el factor más influyente es el operador de mutación y en segundo lugar la probabilidad de cruzamiento.

Tabla 10. Análisis de varianza del IRSM para el caso de 4 máquinas por etapa.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A: Crossover	760	1	760	4.602	0.03349
B: Mutation	13471	1	13471	81.565	6.09e-16
C: Cr. Prob.	5157	1	5157	31.223	1.00e-07
D: Mu. Prob.	1540	1	1540	9.326	0.00266
E: Population	3395	1	3395	20.557	1.15e-05
Residuals	25765	156	165		

En el caso de 6 máquinas por etapa el factor más influyente es el operador de mutación y en segundo la probabilidad de mutación.

Tabla 11. Análisis de varianza del IRSM para el caso de 6 máquinas por etapa.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A: Crossover	2587	1	2587	18.39	3.14e-05
B: Mutation	17785	1	17785	126.47	2e-16
C: Cr. Prob.	2886	1	2886	20.52	1.16e-05
D: Mu. Prob.	4840	1	4840	34.42	2.58e-08
E: Population	2881	1	2881	20.49	1.18e-05
Residuals	21937	156	165		

En resumen, en cada análisis de varianza, el *P-Value* muestra que los cinco factores contemplados tienen efectos significativos a la variable respuesta, sin embargo, para cada caso de máquinas por etapa, el factor que tiene mayor significancia es el

operador de mutación que es el que tiene mayor influencia en la variable respuesta en todos los casos. En segundo lugar, se encuentra el operador de cruzamiento para el caso de 2 máquinas por etapa, la probabilidad de cruzamiento para el caso de 4 máquinas, y el factor probabilidad de mutación para el caso de 6 máquinas.

4.2. Análisis mediante el Frente de Pareto

Las soluciones del Frente de Pareto (FP) son soluciones de problemas con múltiples objetivos, en vez de considerar una sola solución. Estas soluciones proveen al tomador de decisiones (TD) mayor comprensión al momento de tomar una decisión final. Las soluciones óptimas de Pareto son aquellas que dominan a las demás con respecto a que no contienen los peores valores para cada objetivo, y el mejor para al menos un objetivo.

Hay varios factores que afectan el algoritmo en la búsqueda de una apropiada y decisiva solución, donde la búsqueda de valores de los parámetros tiende a ser una de ellas. Mientras que las diferentes combinaciones de parámetros dirigen a soluciones diferentes, se trata de encontrar una mejor combinación de parámetros para el algoritmo genético bi-objetivo.

En las figuras 15, 16 y 17 se muestran los individuos obtenidos de los experimentos para cada caso de máquinas por etapa. El eje “x” representa la energía usada por las máquinas, el eje “y” representa el tiempo de completado de los trabajos. Para tratar de determinar a los mejores individuos fue calculado un FP para cada caso. Cada punto representa una combinación de parámetros, de las 162 combinaciones.

Cada FP consiste de puntos enumerados de menor a mayor de acuerdo a su índice de deseabilidad (ID) obtenido mediante la aplicación de funciones de deseabilidad a cada solución en el frente. La enumeración es obtenida de acuerdo a la aproximación del ID a 1. Los más cercanos tendrán un mejor lugar en la

enumeración, concluyendo como los mejores resultados considerando los objetivos perseguidos.

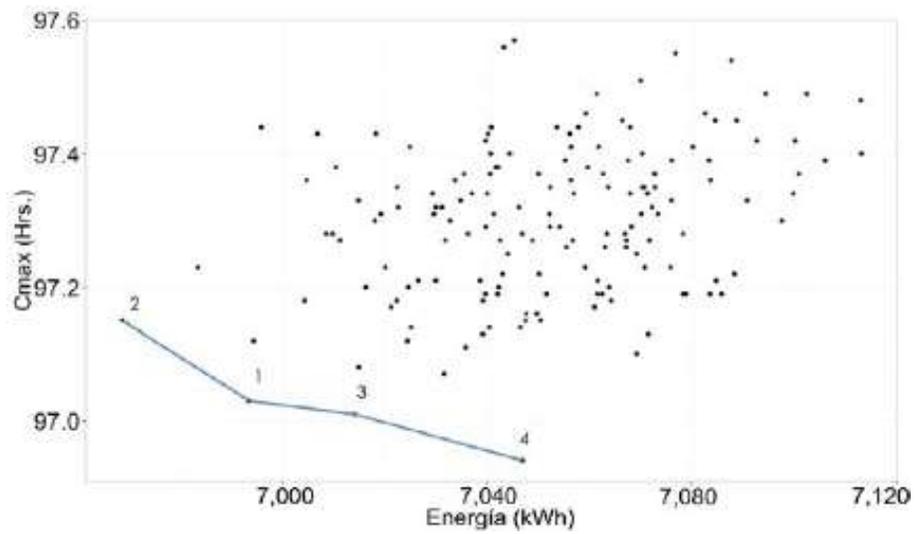


Figura 15. Frente de Pareto para el caso de 2 máquinas por etapa.

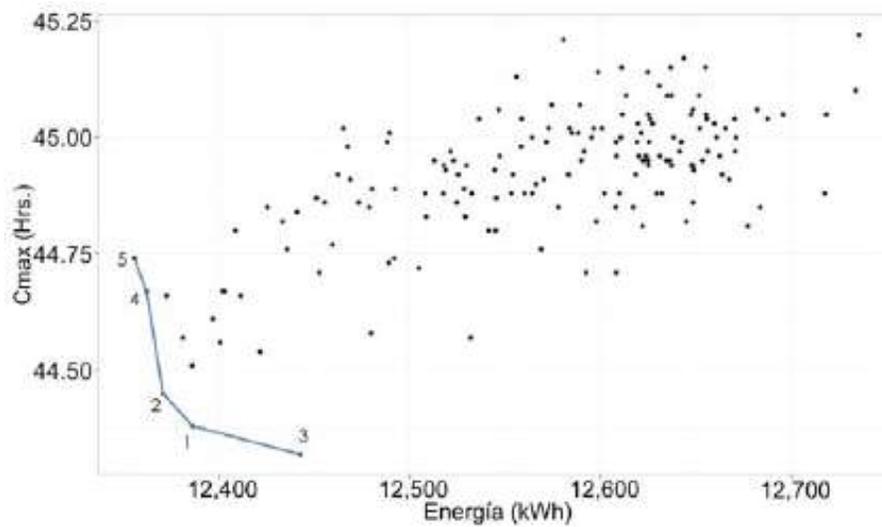


Figura 16. Frente de Pareto para el caso de 4 máquinas por etapa.

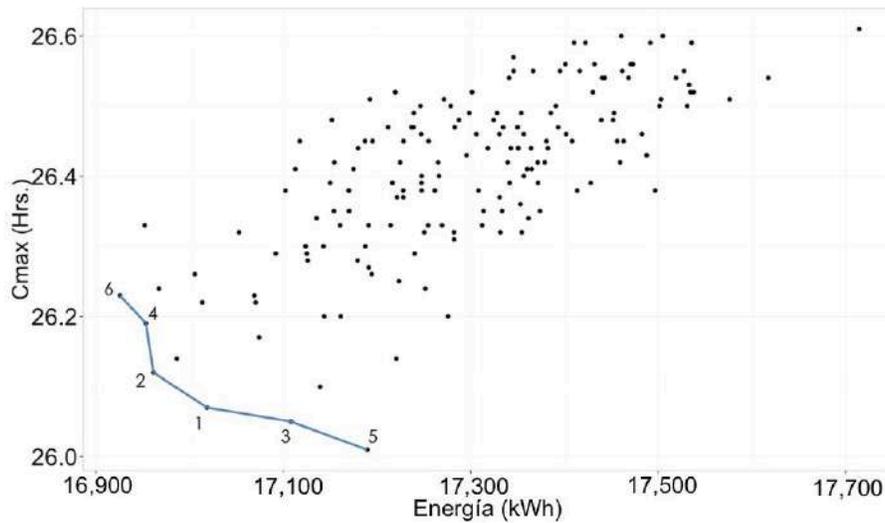


Figura 17. Frente de Pareto para el caso de 6 máquinas por etapa.

Las tablas 12, 13 y 14 muestran el orden de los puntos de los FP respecto a una combinación, también se muestra la energía consumida, el Cmax y el índice de deseabilidad. Para cada caso de máquinas por etapa el conjunto de soluciones (combinaciones) es casi similar, variando solo en una o dos combinaciones. Aquellos que aparecen en cada caso, son la 120, 129 y la 156, indicando que son las combinaciones más significativas encontradas en el FP en los 3 casos de máquinas por etapa. A la combinación 156 es la que le corresponde el mayor índice de deseabilidad, seguida de la 129 en los 3 casos.

Tabla 12. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 2 máquinas por etapa.

No.	Combinación	Energía (kWh)	Cmax (Hrs)	Deseabilidad (ID)	
1	156	6,993.41	97.03	0.843	Select
2	129	6,968.70	97.15	0.816	
3	128	7,014.08	97.01	0.782	
4	120	7,046.87	96.94	0.677	

Tabla 13. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 4 máquinas por etapa.

No.	Combinación	Energía (kWh)	Cmax (Hrs)	Deseabilidad (ID)	
1	156	12,381.55	44.38	0.926	Select
2	129	12,370.82	44.45	0.906	
3	147	12,442.75	44.32	0.878	
4	128	12,362.17	44.67	0.775	
5	120	12,355.79	44.74	0.730	

Tabla 14. Selección de la mejor combinación de parámetros para el AG bi-objetivo en el caso de 6 máquinas por etapa.

No.	Combinación	Energía (kWh)	Cmax (Hrs)	Deseabilidad (ID)	
1	156	17,018.73	26.07	0.891	Select
2	129	16,961.46	26.12	0.883	
3	120	17,108.33	26.05	0.847	
4	147	16,953.30	26.19	0.823	
5	138	17,189.52	26.01	0.816	
6	75	16,925.85	26.23	0.796	

La Tabla 15 muestra los parámetros de las combinaciones más sobresalientes o significativas. Se observa que el cruzamiento PMX se mantiene sólo en la combinación 156, que es la que tiene mayor índice de deseabilidad. El operador de mutación *Desplazamiento* se mantiene en las tres. La probabilidad de cruzamiento se mantiene en 0.9 en las dos combinaciones más significativas. La probabilidad de mutación en todas las combinaciones es 0.2. La población se mantiene con 50 individuos en los 3 casos.

Tabla 15. Óptima combinación de parámetros para el AG bi-objetivo en cada caso de máquinas por etapa.

No.	Comb.	Cr	Mu	Pcr	Pmu	Pob	
1	156	PMX	Displacement	0.9	0.2	50	Select
2	129	OX	Displacement	0.9	0.2	50	
3	120	OX	Displacement	0.7	0.2	50	

Se concluye que al seleccionar los parámetros de esas 3 combinaciones se pueden conseguir los mejores resultados en el AG bi-objetivo para cada caso de máquinas por etapa, ya que tienen similar combinación de parámetros, en especial la 156 por mantenerse como la mejor en los 3 casos de máquinas por etapa.

Las figuras 18 a la 22 muestran la media y el 95% de nivel de confianza de los parámetros más influyentes para el caso de máquinas por etapa donde se dio el más alto índice de deseabilidad, siendo el caso de 4 máquinas por etapa con 0.926 de ID (Tabla 12). La Figura 18 muestra los resultados obtenidos de los operadores de cruzamiento. Se observa que el operador PMX es el mejor de los dos probados.

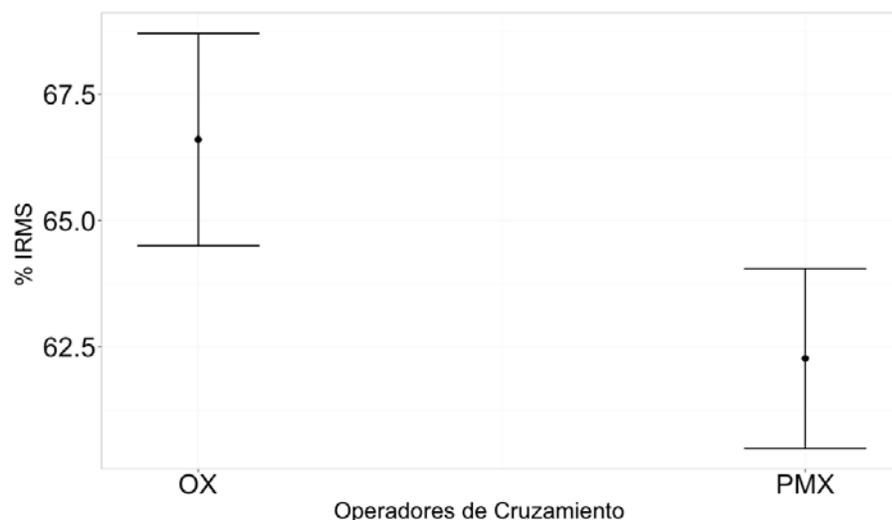


Figura 18. Operador de cruzamiento del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.

La Figura 19 presenta el gráfico de los operadores de mutación donde el operador “Desplazamiento” es el que muestra el mejor resultado con una gran diferencia.

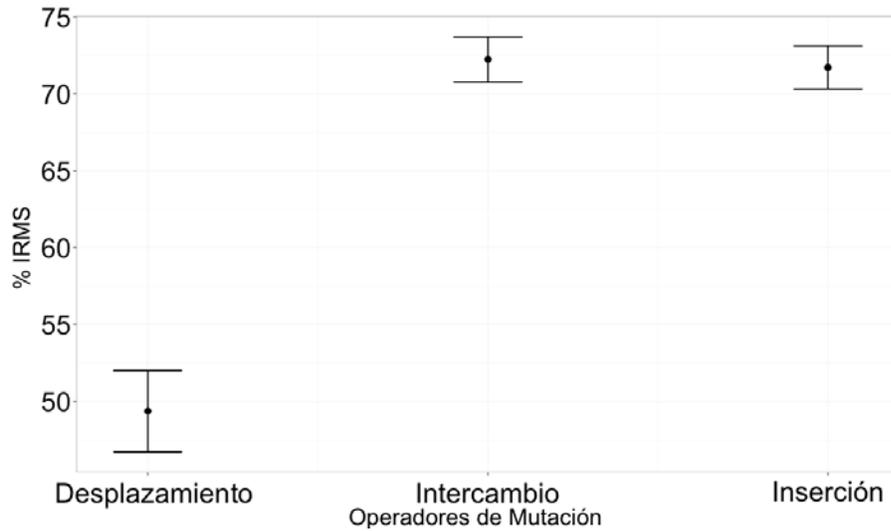


Figura 19. Operador de mutación del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.

En la Figura 20 se muestran los resultados de las probabilidades de cruzamiento. Se puede observar que la mejor probabilidad de cruzamiento ocurre en 0.9.

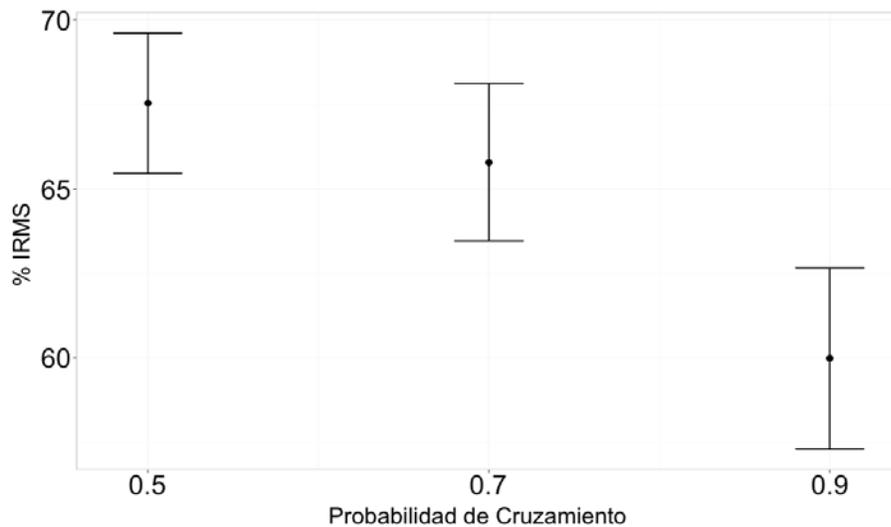


Figura 20. Probabilidad de cruzamiento del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.

La Figura 21 muestra el gráfico para la probabilidad de mutación. La mejor probabilidad ocurre en 0.2.

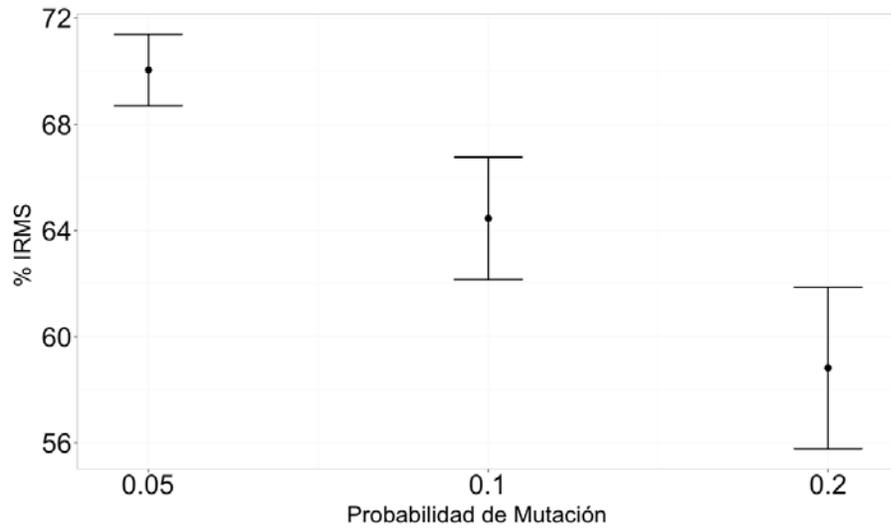


Figura 21. Probabilidad de mutación del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.

La Figura 22 presenta el gráfico del tamaño de la población donde se indica que una población de 50 individuos es estadísticamente más significativa.

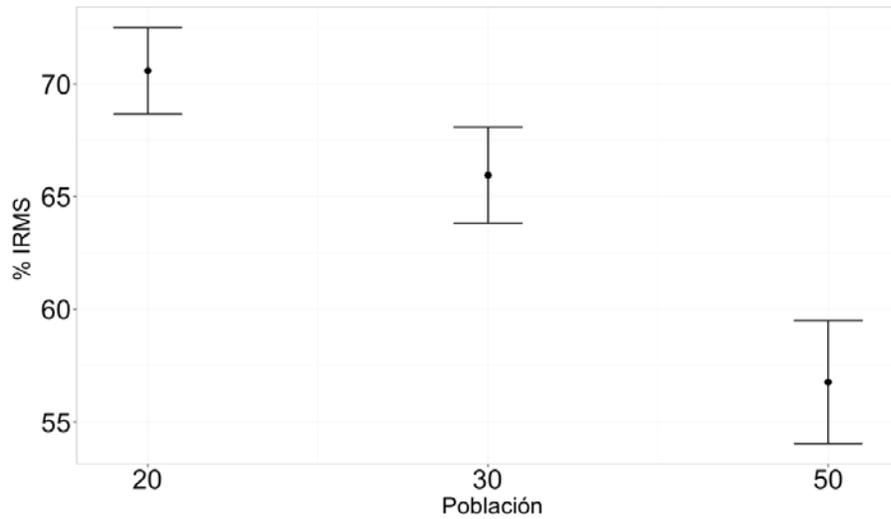


Figura 22. Población del caso de 4 máquinas por etapa. Media y 95% de nivel de confianza.

En las figuras se observa que los parámetros más influyentes corresponden a la combinación 156, el cual tiene mayor índice de deseabilidad en los 3 casos de máquinas por etapa (ver tablas 11, 12 y 13).

Los resultados obtenidos de los parámetros con más influencia en cada caso, son similares, después de aplicar funciones de deseabilidad a las combinaciones que se encuentran en el frente de Pareto, comparados con los resultados obtenidos aplicando el IRMS.

4.3. Comparación de resultados con la industria de estudio

La Tabla 16 muestra los resultados de energía y Cmax de la industria y el AG bi-objetivo. Los datos de la industria fueron obtenidos considerando el promedio de producción de la carga de kilogramos de tortilla durante un año.

Tabla 16. Resultados de los datos de producción de la industria y el AG bi-objetivo.

Máquinas por etapa	Industria		Algoritmo	
	Energía (kWh)	Cmax (Hrs.)	Energía (kWh)	Cmax (Hrs.)
2	13,050.9	186	6,993.41	97.03
4	26,101.8	93	12,381.55	44.38
6	39,152.7	62	17,018.73	26.07

Las figuras 23 y 24 muestran una comparación de los resultados; en la Figura 23 se muestra una comparación del Cmax, el tiempo es considerado en horas de trabajo continuas.

En la Figura 23 los resultados de la industria para el caso de 2 máquinas por etapa, el tiempo de completar una carga mensual promedio de tortillas es de 186 horas. En el caso de 4 máquinas es de 93 horas y en el caso de 6 máquinas es de 62 horas. En los resultados del algoritmo, para el caso de 2 máquinas por etapa el

tiempo de completado es de 97.03 horas. En el caso de 4 máquinas por etapa el tiempo es de 44.38 horas, y en el caso de 6 máquinas es de 26.07 horas.

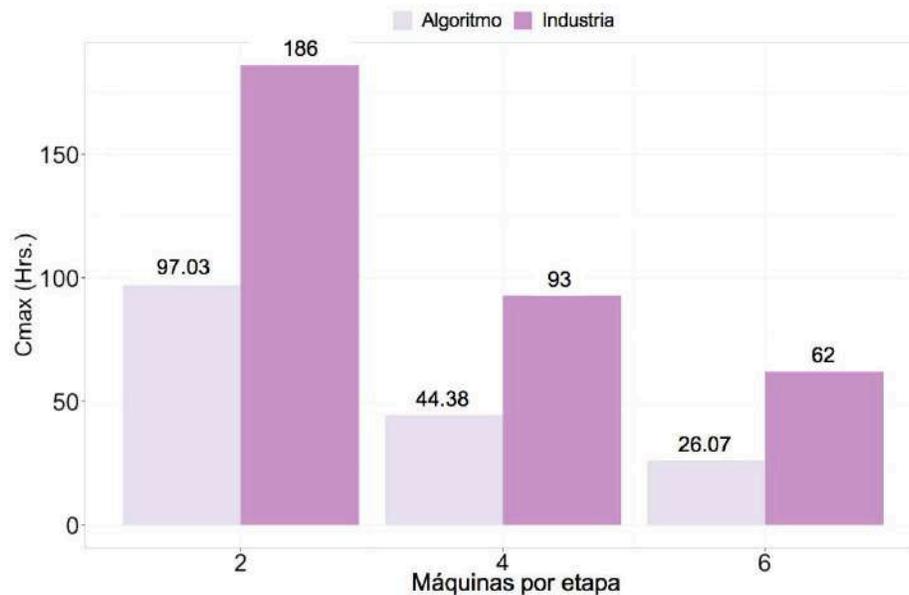


Figura 23. Comparación del Cmax de los datos de producción de la industria y el AG bi-objetivo.

En la Figura 24, se muestra la comparación de la energía, la cual es considerada de acuerdo al tiempo de procesamiento de las máquinas (Tabla 6). En los resultados de la industria para el caso de 2 máquinas por etapa, la energía consumida para la producción de una carga mensual promedio de tortillas es de 13,050.9 kWh. En el caso de 4 máquinas es de 26,101.8 kWh, y en el caso de 6 máquinas es de 39,152.7 kWh. En los resultados del algoritmo, para el caso de 2 máquinas por etapa la energía obtenida es de 6,993.41 kWh. En el caso de 4 máquinas es de 12,381.55 kWh, y en el caso de 6 máquinas es de 17,018.73 kWh.

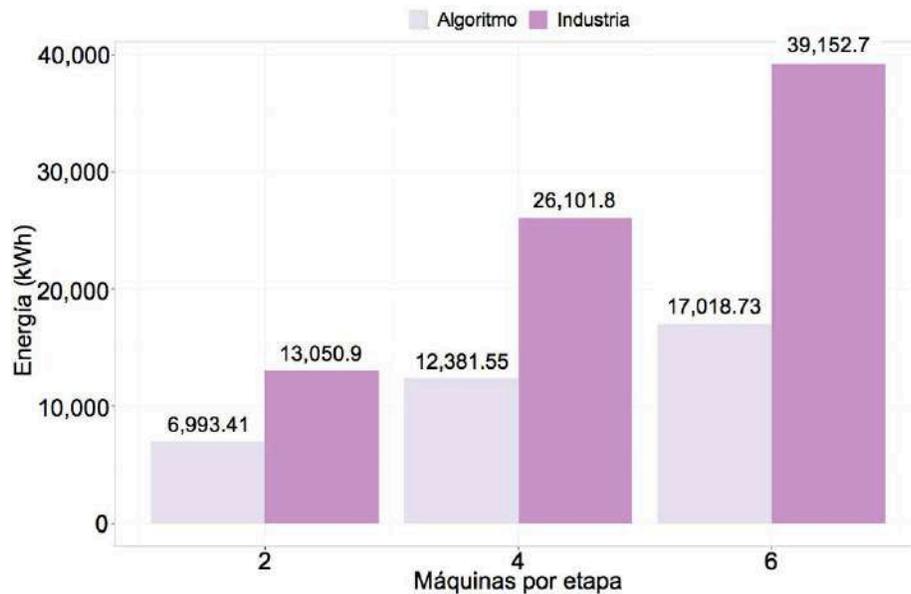


Figura 24. Comparación de la energía de los datos de producción de la industria y el AG bi-objetivo.

La Tabla 17 muestra el porcentaje de diferencia de los resultados del AG bi-objetivo comparados con los resultados de la industria. Los resultados de energía en el algoritmo, para el caso de 2 máquinas por etapa se diferencian un 46.4% comparados a los resultados de la industria. Para el caso de 4 máquinas es de 52.6% y para el caso de 6 máquinas es de 56.5%. En los resultados de Cmax, para el caso de 2 máquinas por etapa se alcanza una diferencia del 47.8% con respecto al tiempo de completado de la industria. Para el caso de 4 máquinas es de 52.3%, y para el caso de 6 máquinas es de 58%.

Tabla 17. Porcentaje de rendimiento en la comparación de los resultados del AG bi-objetivo con los datos de producción de la industria.

Máquinas por etapa	Rendimiento en %	
	Energía	Cmax
2	46.4	47.8
4	52.6	52.3
6	56.5	58

En todos los casos de máquinas por etapa, los porcentajes de energía mejoran en promedio un 51.8%, y en C_{max} mejoran en un 52.7%. Con esto se concluye que los porcentajes de energía y C_{max} en cada caso de máquinas favorecen en gran medida a los resultados del AG bi-objetivo.

Por último, en la Figura 25 se muestra el porcentaje de rendimiento del CO_2 producido debido al consumo de energía eléctrica, de los resultados de energía del AG bi-objetivo comparados con los resultados de energía de la industria. En los resultados de la industria, para el caso de 2 máquinas por etapa, el CO_2 producido durante la producción de una carga mensual promedio de tortillas es de 8.53 toneladas. En el caso de 4 máquinas es de 17.07 toneladas, y en el caso de 6 máquinas es de 25.6 toneladas. En los resultados del algoritmo, para el caso de 2 máquinas por etapa el CO_2 producido es de 4.57 toneladas. En el caso de 4 máquinas es de 8.1 toneladas, y en el caso de 6 máquinas es de 11.3 toneladas.

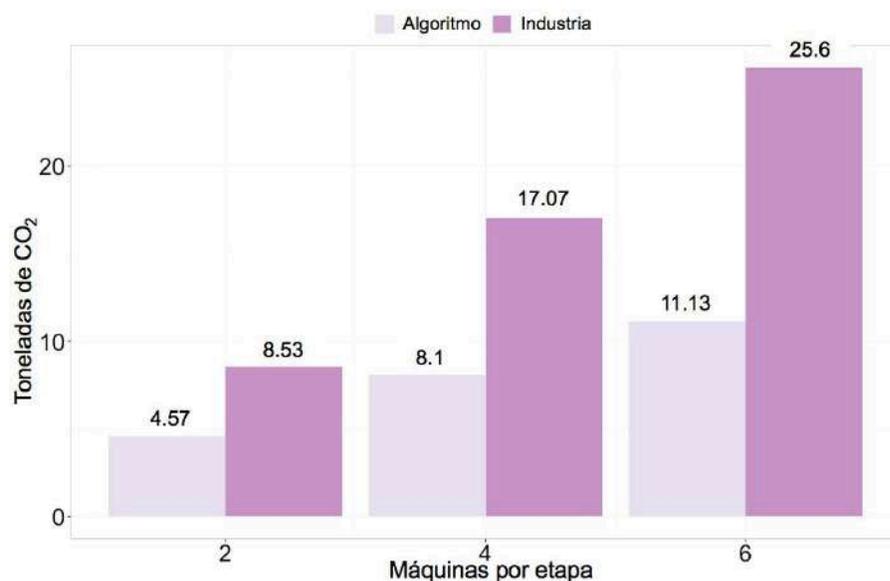


Figura 25. Comparación del CO_2 producido por la energía utilizada en la industria y el AG bi-objetivo.

La Tabla 18 muestra el porcentaje de rendimiento de los resultados del AG bi-objetivo comparados con los resultados de la industria. En el caso de 2 máquinas por etapa los porcentajes llegan al alcanzar el 46.4% de rendimiento, en el caso de 4 máquinas llegan a 52.6% y en el caso de 6 llegan a 56.5%, mejorando en más del 50%.

Tabla 18. Porcentaje de rendimiento respecto al CO₂ del AG bi-objetivo con los obtenidos por la industria.

Máquinas por etapa	Rendimiento en %
	CO₂
2	46.4
4	52.6
6	56.5

Se concluye que los porcentajes de CO₂ en cada caso de máquinas por etapa el rendimiento promedio es de 51.8%, favoreciendo de nuevo a los resultados del AG bi-objetivo.

CONCLUSIONES Y APORTACIONES

En esta tesis se aborda un complejo problema de programación de la producción, común en la industria de producción de tortillas, dónde las máquinas deben de procesar la materia prima a través de procesos que implican 6 etapas que cuentan con 2, 4 y 6 máquinas por etapa.

Se ha formalizado el modelo de la planta productiva que se describe, a través de un taller de flujo híbrido con máquinas paralelas no relacionadas e idénticas, tiempos de ajuste dependientes e independientes de la secuencia de los trabajos y espacio ilimitado para el almacenamiento temporal de los trabajos.

En los últimos años ha aumentado el número de publicaciones hacia entornos de producción reales. Sin embargo, sigue siendo una realidad que aunque se ha investigado mucho en esta área, existe todavía una brecha importante entre la teoría y las aplicaciones prácticas.

Para el modelo de producción que se aborda en esta tesis se propone para su solución: un algoritmo genético bi-objetivo.

La asignación de los trabajos a las máquinas en cada etapa se realiza tomando en cuenta la máquina que permita terminar el trabajo en el menor tiempo posible, considerando los tiempos de ajuste de la secuencia de los trabajos. Este algoritmo se compara con los resultados de producción de la industria superándolo en más de un 50%.

Los datos de entrada son tomados considerando el ambiente de producción real en los que respecta a los tipos de tortilla más demandados en la industria.

El algoritmo propuesto ha sido calibrado a través de experimentos computacionales aplicando 162 algoritmos alternativos (combinaciones de factores) con 30 ejecuciones cada uno, en total 4,860 evaluaciones, considerando 3 casos de máquinas por etapa (2, 4 y 6), 14,580 evaluaciones fueron realizadas. Los factores

tomados en cuenta fueron: tipo de cruzamiento, tipo de mutación, probabilidad de cruzamiento, probabilidad de mutación y el tamaño de la población.

Se toma como variable respuesta el porcentaje de incremento relativo sobre la mejor solución conocida. Los resultados de los experimentos fueron sometidos a un análisis de varianza multifactorial donde los resultados muestran que los cinco factores contemplados tienen efectos significativos a la variable respuesta a un 95% de nivel de confianza.

En cada caso de máquinas por etapa, el operador de mutación fue el que tuvo el efecto más significativo, esto es, que este operador juega un rol importante en algoritmo multiobjetivo aplicado a este tipo de problemas.

Se utilizó el Frente de Pareto para modelar los dos objetivos perseguidos: Cmax y consumo de energía. De los 162 algoritmos evaluados en cada caso de máquinas por etapa, se aplicaron funciones de deseabilidad a las soluciones en el frente de Pareto, obteniendo de acuerdo a su índice de deseabilidad que la combinación 156 es la que se mantiene con el más alto valor en todos los casos. Los parámetros de esta combinación, son: 1) Cruzamiento: PMX; 2) Mutación: Desplazamiento; 3) Probabilidad de Cruzamiento: 0.9; 4) Probabilidad de mutación: 0.2; 5) Población: 50 individuos.

Resultados iguales fueron obtenidos de los parámetros más significativos obtenidos mediante el IRMS. Por lo que esta combinación de parámetros puede conseguir los mejores resultados en el algoritmo genético bi-objetivo en cada caso de máquinas por etapa.

Comparando los objetivos de energía y Cmax de los resultados del algoritmo genético bi-objetivo con los datos de producción de la industria, en todos los casos de máquinas por etapa, los porcentajes de energía mejoran en promedio un 51.8%, y en Cmax mejoran en un 52.7%.

En materia ambiental, al comparar los resultados del algoritmo con los datos de producción de la industria, respecto a la producción de CO₂ hecha por el consumo

de energía, para el caso de 2 máquinas por etapa se logra una disminución del 46.4%, en el caso de 4 máquinas llegan a 52.6% y en el caso de 6 disminuyen un 56.5%, mejorando en promedio más del 50%.

Se concluye que el algoritmo propuesto consigue un mayor rendimiento comparado con la estrategia actual de producción de la industria. La reducción es considerable en el tiempo de completar los pedidos y el consumo de energía, y por consiguiente se reduce también la producción de CO₂.

Como trabajo futuro, se espera comparar el algoritmo calibrado con otros algoritmos propuestos a problemas similares.

BIBLIOGRAFÍA

- Arthanary, L.S., & Ramaswamy, K.G. (1971). An Extension of Two Machine Sequencing Problem. *OPSEARCH, The Journal of the Operational Research Society of India*, 8(4), 10-22.
- Aguilar, P. V. (2014). *Resolución de problemas de optimización combinatoria utilizando técnicas de computación evolutiva*. Una aplicación a la biomedicina. Tesis Doctoral, Universidad de Coruña, La Coruña, España.
- Allaoui, H., & Artiba, A. (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research*, 33, 1399-1419.
- Arroyave, R. J. A. & Garcés, G. L. F. (2006). Tecnologías ambientalmente sostenibles. *Producción + Limpia*, 1(2), 79-86.
- Baños, R., Manzano-Agugliaro, F., Montoya, F. G., Gil, C., Alcayde, A., & Gómez, J. (2011). Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4), 1753-1766. doi:10.1016/j.rser.2010.12.008
- Barbosa, S. M. (2014). *Aplicaciones de algoritmos evolutivos a finanzas*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California, México.
- Baesler, F., Ceballos, L. & Ramírez, M. (2006). Programación multiobjetivo de máquinas moldureras a través de algoritmos genéticos. *Maderas. Ciencia y tecnología*, 8(3), 183-192.
- Baker, K. R. (1974). *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons.
- Bellman R. (1957). *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 366p.
- Beltran. L., Contreras, C., Valenzuela, J. M., Dávila, M., Arévalo, V., Jiménez, O.,

Ortega, H. (2012). Atlas de almacenamiento geológico de CO₂, México. Obtenido de <http://co2.energia.gob.mx/res/ATLAS%20FINAL.pdf>

Blum, C., Chiong, R., Clerc, M., De Jong, K., Michalewicz, Z., Neri, F., Weise, T., (2012). Evolutionary optimization. In: Chiong, R., Weise, T., Michalewicz, Z. (Eds.), *Variants of Evolutionary Algorithms for Real-world Applications*. Springer Berlin Heidelberg, pp. 1e29.

Brah, S.A., & Hunsucker, J.L. (1991). Branch and bound algorithm for the flow shop with multiple processors. *European Journal of Operational Research*, 51, 88-99. Chen, S.-H. y Yeh, C.-H. (1997). Toward a computable approach to the efficient market hypothesis: An application of genetic programming. *Journal of Economic Dynamics and Control*, 21(6): 1043–1063.

Campbell, H. G., Dudek, R.A., & Smith, M. L. (1970). An heuristic algorithm for the n jobs m machine sequencing problem. *Management Science*, 16(10), 630-637.

Cerny, V. (1985). Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, 45, 41-51.

Comisión Federal de Electricidad. (2014). Datos de facturación, 2013 a Octubre del 2014. San Luis Rio Colorado, Sonora, México.

Coello, C. C. A. & Landa, B. R. (2009). Evolutionary multi-objective optimization in materials science and engineering. *Materials and Manufacturing Processes*, 24(2), 119-129. doi: 10.1080/10426910802609110

Colomi, A., Dorigo, M. & Maniezzo, V. (1992). An Investigation of Some Properties of an Ant Algorithm, *Proceeding of the PPSN-II, Second International Conference on Parallel Problem Solving from Nature*, Manner, R. & Manderick (Eds.), Elsevier, Amsterdam, The Netherlands, 509-520.

Costa, A., Cappadonna, F. A., & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *The International Journal of Advanced*

- Manufacturing Technology*, 69(9), 2799-2817. doi:10.1007/s00170-013-5221-5
- Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 29(5), 418-429. doi:10.1016/j.rcim.2013.04.001
- Deb, K. (2011). Multi-objective optimization using evolutionary algorithms: An introduction. In Wang L., N-Amos, H. C. & Deb, K. (Eds.), *Multi-objective evolutionary optimization for product design and manufacturing* (pp. 3-34). London : Springer. doi: 10.1007/978-0-85729-652-8_1
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197. doi:10.1109/4235.996017
- Dhingra, S., Gupta, S. B., & Biswas, R. (2014). Hybrid GASA for bi-criteria multiprocessor task scheduling with precedence constraints. *Computer Applications: An International Journal*, 1(1), 11-21.
- Eiben, A. E. y Smith, J. E. (2003). *Introduction to Evolutionary Computing*. EUA : Springer.
- Elkington, J. (1998). *Cannibals with forks: The triple bottom line of 21st century business*. Gabriola Island, BC: New Society Publishers.
- El-Zahar, M. H., & Rival, I. (1985). Greedy linear extensions to minimize jumps. *Discrete Applied Mathematics*, 11, 143-156.
- Estévez, V. P. (1997). Optimización mediante algoritmos genéticos. *Anales del Instituto de Ingenieros de Chile*, 83-92.
- Fadaei, M., & Zandieh, M. (2013). Scheduling a bi-objective hybrid flow shop with sequence-dependent family setup times using metaheuristics. *Arabian Journal for Science and Engineering*, 38(8), 2233-2244. doi:10.1007/s13369-013-0611-4
- Feo, T. A., & Resende, M. G. C. (1989). A probabilistic heuristic for computationally

difficult set covering problem. *Operations Research Letters*, 8, 67-71.

Gahm, C., Denz, F., Dirr, M., & Tuma, A. (2016). Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research*, 248(3), 744-757. doi:10.1016/j.ejor.2015.07.017

Garey, M. R., & Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-completeness. New York: W. H. Freeman.

Ghosh, A. & Dehuri, S. (2005). Evolutionary algorithms for Multi-Criterion Optimization: A Survey. *International Journal of Computing & Information Sciences*. 1(2).

Giffer, B., & Thompson, G. L. (1960). Algorithms for Solving Production Scheduling Problems. *Operations Research*, 8, 487-503.

Giret, A., Trentesaux, D., & Prabhu, V. (2015). Sustainability in manufacturing operations scheduling: A state of the art review. *Journal of Manufacturing Systems*, 37, 126-140. doi:10.1016/j.jmsy.2015.08.002

Glover, F. (1990). Tabu Search: A Tutorial. *Interfaces*, 20, 74-94.

Gog, A., & Chira, C. (2011). Comparative analysis of recombination operators in genetic algorithms for the travelling salesman problem. (pp. 10-17). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-21222-2_2

Goldberg, D.E. (1989). *Genetic Algorithms in search, optimization and machine learning*. EUA : Addison-Wesley.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy-Kan, A. H. G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, 5, 287-326.

Gourgand, M., Grangeon, N., & Norre, S. (1999). Metaheuristics for the deterministic hybrid flow shop problem. *Proceedings of the International Conference on Industrial Engineering and Production Management, IEPM'99* (pp. 136–145). Glasgow: FUCAM - INRIA.

- Gupta, J. N. D. (1988). Two-Stage, Hybrid Flowshop Scheduling Problem. *Journal of the Operational Research Society*, 39(4), 359-364.
- Hecker, F. T., Hussein, W. B., Paquet-Durand, O., Hussein, M. A., & Becker, T. (2013). A case study on using evolutionary algorithms to optimize bakery production planning. *Expert Systems with Applications*, 40(17), 6837. doi:10.1016/j.eswa.2013.06.038
- Hecker, F. T., Stanke, M., Becker, T., & Hitzmann, B. (2014). Application of a modified GA, ACO and a random search procedure to solve the production scheduling of a case study bakery. *Expert Systems with Applications*, 41(13), 5882-5891. doi:10.1016/j.eswa.2014.03.047
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 228p.
- Hwang, H. (2002). An improved model for vehicle routing problem with time constraint based on genetic algorithm. *Computers & Industrial Engineering*, 42(2), 361-369. doi:10.1016/S0360-8352(02)00033-5
- Jaimes, A. L. and Coello, C. A. C. (2013) Interactive Approaches Applied to Multiobjective Evolutionary Algorithms, in *Multicriteria Decision Aid and Artificial Intelligence* (eds M. Doumpos and E. Grigoroudis), John Wiley & Sons, Ltd, Chichester, UK. doi: 10.1002/9781118522516.ch8
- Johnson, S. M. (1954). Optimal two and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61-68.
- Konak, A., Coit, D. W. & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. In Guedes, S. C., Saleh, J. H. & Xie, M. (Eds.), *Reliability Engineering & System Safety Editorial Board*, 992–1007. Reino Unido : Elsevier.
- Lagunas, J. R., Moo, Y. V. & Ortiz, M. B. (2013). Optimización multiobjetivo mediante algoritmos genéticos: Aplicación a controladores PID robustos. *Programación Matemática y Software*, 5(1).

- Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2), 129-170. doi:10.1023/A:1006529012972
- Liu, Y., Dong, H., Lohse, N., & Petrovic, S. (2015). Reducing environmental impact of production during a rolling blackout policy - A multi-objective schedule optimisation approach. *Journal of Cleaner Production*, 102, 418-427. doi:10.1016/j.jclepro.2015.04.038
- Liu, C., & Huang, D. (2014). Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *International Journal of Production Research*, 52(2), 337-352. doi:10.1080/00207543.2013.825740
- López, R. B. & Mezura, M. E. (2007). Comparación de algoritmos evolutivos y bio-inspirados en problemas de optimización con restricciones. *IV encuentro de la Mujer en la Ciencia*.
- Lu, L., Anderson-Cook, C., & Lin, D. (2014). Optimal designed experiments using a pareto front search for focused preference of multiple objectives. *Computational Statistics & Data Analysis*, 71, 1178-1192. doi:10.1016/j.csda.2013.04.008
- Martin, B., & Sanz, A. (1997). *Redes neuronales y Sistemas borrosos*. Madrid, España: Edit. RAMA, 442p.
- May, G., Stahl, B., Taisch, M., & Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, 53 (23), 7071-7089. doi:10.1080/00207543.2015.1005248
- Mejía, C. J. (2008). *Asignación de horarios de clases universitarias mediante algoritmos evolutivos*. Tesis de maestría, Universidad de Norte, Barranquilla, Atlantico, Colombia.
- Mori, M., Fujishima, M., Inamasu, Y., & Oda, Y. (2011). A study on energy efficiency improvement for machine tools. *CIRP Annals - Manufacturing Technology*, 60(1),

145-148. doi:10.1016/j.cirp.2011.03.099

Mouzon, G., Yildirim, M. B., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 45(18), 4247-4271. doi:10.1080/00207540701450013

Newman, S. T., Nassehi, A., Imani-Asrai, R., & Dhokia, V. (2012). Energy efficient process planning for CNC machining. *CIRP Journal of Manufacturing Science and Technology*, 5(2), 127-136. doi:10.1016/j.cirpj.2012.03.007

Nishant, R., Teo, T. S. H., & Goh, M. (2014). Energy efficiency benefits: Is technophilic optimism justified? *IEEE Transactions on Engineering Management*, 61(3), 476-487. doi:10.1109/TEM.2014.2314703

Osman, I. H., & Potts, C. N. (1989). Simulated Annealing for permutation flowshop sequencing. *OMEGA International Journal of Management Science*, 17, 551-557.

Osman, I., H. (1995). An introduction to Meta-Heuristics. Lawrence, M., & Wilson, C. Editores. *Operational Research, Tutorial Papers*, 92-122.

Palmer, D. S. (1965). Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time - A Quick Method of Obtaining a Near Optimum. *Operational Research Quarterly*, 16, 101-107.

Persson, A., Andersson, M., Grimm, H. & Ng, A. (2008). Metamodel-assisted simulation-based optimization of a real-world manufacturing problem.

Peñuela, M. C. A. & Granada, E. M. (2007). Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no-dominado (NSGA-II). *Scientia Et Technica*, 8(35), 175-180.

Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems*. New Jersey: Prentice Hall, 586p.

Pinedo, M. L., & SpringerLink (Online service). (2012). *Scheduling: Theory, algorithms, and systems* (4th 2012. ed.). Boston, MA: Springer US.

- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985-2002. doi:10.1016/S0308-0548(03)00158-8
- Portmann, M. C. (1997). Scheduling methodology: optimization and compusearch approaches I. En Artiba, A., & Elmaghraby, S. E., editores, *The Planning and Scheduling of Production Systems. Methodologies and Applications*, London: Chapman & Hall, pp. 271-300.
- Programa de Desarrollo Urbano del Centro de Población de la Ciudad de San Luis Rio Colorado, Sonora. (2013). Obtenido de http://ordenamientoterritorial.gob.mx/SEIOT/DPT/PDUCP/san%20luis%20rio%20colorado/DOCUMENTO2013/0_RESUMENEJECUTIVOPDUCP_SLRC_2013.pdf
- Pusavec, F., Krajnik, P., & Kopac, J. (2010). Transitioning to sustainable production – part I: Application on machining technologies. *Journal of Cleaner Production*, 18(2), 174-184. doi:10.1016/j.jclepro.2009.08.010
- Rodríguez, Z. M. (2005). *Un algoritmo genético para el ensamble de secuencias de ADN*. Tesis de maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California, México.
- Rosen, K. H. (1999). *Discrete Mathematics and Its Applications*. Boston: WCB/McGraw- Hill, 678p.
- R-project.org. (2016). *R: The R Project for Statistical Computing*. [online] Available at: <https://www.r-project.org/> [Accessed 4 Jul. 2016].
- Ruiz, R., & Maroto, C. (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169, 781-800.
- Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1-18. doi:10.1016/j.ejor.2009.09.024

- Salvador, M. S. (1973). A solution to a special case of flow shop scheduling problems. En Elmaghraby, S. E., editor, *Symposium of the Theory of Scheduling and Applications*, New York: Springer-Verlag, pp. 83-91.
- Seidgar, H., Ezzati, M., Kiani, M., & Tavakkoli-Moghaddam, R. An Efficient Genetic Algorithm for Two-stage Hybrid Flow Shop Scheduling with Preemption and Sequence Dependent Setup Time.
- Shapiro, J. F. (1979). A Survey of Lagrangian Techniques for Discrete Optimization. *Annals of Discrete Mathematics*, 5, 113-138.
- Silver, E. A., Vidal, R. V., & De-Werra, D. (1980). A Tutorial on Heuristics Methods. *European Journal of Operational Research*, 5(3), 153-162.
- Starkweather, T., McDaniel, S., Mathias, K. E., L. D. Whitley, C. (1991). A comparison of genetic sequencing operators, In R.K.Belew, L.B.Booker (Eds), *Morgan Kaufmann*, Proceedings of the 4th International Conference on Genetic Algorithms,. (pp. 69–76), San Francisco, CA.
- Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15(3), 281-295. doi:10.1016/S0954-1810(01)00005-X
- Tang, L., Luh, P.B., Liu J., & Fang, L. (2002). Steel-making process scheduling using Lagrangian relaxation. *International Journal of Production Research*, 40(1), 55-70.
- Tobón, O. D. & Agudelo, F. S. (2008). *Optimización de herramientas multiobjetivo para la toma de decisiones de inversión aislados sostenibles de energía*. Medellín, Colombia : ISA – COLCIENCIAS.
- Thomas, J. L., & McClain, J. O. (1993). An overview of production planning. En Graves, S. C., Rinnooy Kan, A. H. G., y Zipkin, P. H., editores, *Logistics of Production and Inventory*, volumen 4 de *Handbooks in Operations Research and Management Science*, Amsterdam: Elsevier Science Publishers, B. V., pp. 333–370.
- Toscano, P. G. (2001). *Optimización multiobjetivo usando un micro algoritmo*

genético. Tesis de maestría, Universidad Veracruzana - LANIA, Veracruz, México.

Trentesaux, D. & Prabhu, V. V. (2013). Introduction to Shop-Floor Control. Wiley Encyclopedia of Operations Research and Management Science. 1–9.

United Nations (1987, December). Report of the World Commission on Environment and Development. A/RES/42/187. 96th plenary meeting.

Wagner, T., & Trautmann, H. (2010). Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation*, 14(5), 688-701. doi:10.1109/TEVC.2010.2058119

Wang, S., Lu, X., Li, X., & Li, W. (2015). A systematic approach of process planning and scheduling optimization for sustainable machining. *Journal of Cleaner Production*, 87(C), 914-929. doi:10.1016/j.jclepro.2014.10.008.

Wirth, N. (1976). *Algorithms + data structures = Programs*. Englewood Cliffs, N. J.: Prentice-Hall, 366p.

Yaurima, V., Burtseva, L., & Tchernykh, A. (2009). Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Computers & Industrial Engineering*, 56(4), 1452-1463. doi:10.1016/j.cie.2008.09.004

Yu, X. & Gen, M. (2010). *Introduction to Evolutionary Algorithms*. EUA: Springer.

Zhang, R., & Chiong, R. (2016;2015;). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, 112, 3361-3375. doi:10.1016/j.jclepro.2015.09.097

Zheng, Y., Chen, S., Lin, Y., & Wang, W. (2013). Bio-inspired optimization of sustainable energy systems: A review. *Mathematical Problems in Engineering*,

2013, 1-12. doi:10.1155/2013/354523

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Tesis doctoral, Escuela Politécnica Federal de Zúrich, Suiza.

Zhou, A., Qu, B., Li, H., Zhao, S., Nagaratnam, S. P. & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32–49. doi:10.1016/j.swevo.2011.03.001