

UNIVERSIDAD DE CONCEPCIÓN – CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

**Programación de una máquina considerando variados lotes
para múltiples productos y tiempos de setup dependientes de
la secuencia**

Por

Mónica Cecilia Riquelme Garrido

Profesor guía

Eduardo Salazar Hornig

Concepción, Diciembre de 2020

Tesis presentada a la

DIRECCIÓN DE POSTGRADO
UNIVERSIDAD DE CONCEPCIÓN

Para optar al grado de

MAGÍSTER EN INGENIERÍA INDUSTRIAL



*A mis padres y hermanas, por darme la
oportunidad de llegar a este momento, apoyarme
siempre y ser mi pilar en momentos difíciles.*

RESUMEN

PROGRAMACIÓN DE UNA MÁQUINA CONSIDERANDO VARIADOS LOTES PARA MÚLTIPLES PRODUCTOS Y TIEMPOS DE SETUP DEPENDIENTES DE LA SECUENCIA

Mónica Cecilia Riquelme Garrido

Diciembre 2020

PROFESOR GUÍA: Eduardo Salazar Hornig

PROGRAMA : Magíster en Ingeniería Industrial

El presente trabajo se enfoca en el problema de dimensionamiento de lotes y programación de producción para un sistema de una máquina, múltiples productos y setup dependientes de secuencia. La investigación surge motivada por el trabajo realizado en la literatura sobre un problema real de una empresa manufacturera, el cual es utilizado como punto de comparación. Debido a que el problema presentado posee una dificultad NP-Hard, se desarrolla una metodología de solución metaheurística que combina muestreo aleatorio y algoritmos genéticos. La metodología desarrollada es testada mediante problemas de prueba y comparada con los resultados obtenidos en el problema original, obteniéndose resultados de mejor calidad, llegando a mejorar la solución anterior hasta en un 88,2% en uno de los problemas de prueba resueltos. Se prueba que la metodología propuesta logra mejores soluciones de forma más eficiente y en tiempos computacionales reducidos.

Palabras clave: Dimensionamiento de lotes, Algoritmos genéticos, Metaheurística, Programación de producción de una máquina, Muestreo aleatorio

ABSTRACT

LOT SIZING AND SCHEDULING FOR A SINGLE MACHINE, MULTI-PRODUCTS AND SEQUENCE-DEPENDENT SETUP TIMES

Mónica Cecilia Riquelme Garrido

December 2020

THESIS SUPERVISOR: **Eduardo Salazar Hornig**

PROGRAM : **Master in Industrial Engineering**

The following work focus in the lot sizing and scheduling problem on a single machine, multi-product and sequence-dependent setup times. This investigation is motivated by the work find in the literature about the real problem of a manufacturing facility, which is later use as benchmark. Since the problem is NP-Hard, a metaheuristic solution method is developed which combines random sampling with genetic algorithms. The proposed method is tested using test problems and is compared with the results find in the benchmark investigation, where better solutions are obtained, improving the solution of the original problem up to 88,2% in one of the test instances. It is proven that the solution method proposed in this work provides better solutions in a more efficient manner and low computational times.

Key words: Lot sizing, Genetic Algorithms, Metaheuristic, Single machine scheduling, Random sampling

Índice

Capítulo 1: Introducción.....	1
Capítulo 2: Dimensionamiento de lotes y programación de producción	3
2.1. Historia y desarrollo	3
2.2. Revisión bibliográfica	5
Capítulo 3: Planteamiento del problema	9
3.1 Descripción del problema.....	9
3.2 Medida de desempeño.....	10
3.3 Ejemplo ilustrativo	11
Capítulo 4: Metaheurística MCGA	15
4.1 Conformación de tamaños de lotes mediante Montecarlo	16
4.2 Secuenciación de órdenes de producción mediante Algoritmos Genéticos	17
Capítulo 5: Experimentos.....	19
5.1 Problemas de prueba.....	19
5.2 Instancias aleatorias.....	23
Capítulo 6: Resultados	29
Capítulo 7: Conclusiones	32
Referencias:	34
Anexos:	36
A. Parámetros utilizados en instancias aleatorias de prueba	36
B. Resultados de instancias aleatorias de prueba.....	38

Lista de Tablas

Tabla 5.1: Parámetros generales Problemas 1, 2 y 3.....	20
Tabla 5.2: Demanda Problemas 1, 2 y 3.	21
Tabla 5.3: Parámetros de instancias correspondientes a Problemas 1, 2 y 3.	22
Tabla 5.4: Tamaño configuraciones por instancia de Problemas M1, M2 y M3.	24
Tabla 5.5: Ordenes de producción generadas para problema ilustrativo M3.....	26
Tabla 5.6: Valores de Déficit obtenidos para problema ilustrativo M3.....	27
Tabla 6.5: Tiempos de ejecución de problemas de prueba utilizando metodología MCGA.....	29
Tabla 6.1: Resultados obtenidos para Problema 1.....	30
Tabla 6.2: Resultados obtenidos para Problema 2.....	30
Tabla 6.3: Resultados obtenidos para Problema 3.....	30
Tabla 6.4: Resultados obtenidos para generados aleatoriamente M1, M2 y M3.	31

Lista de Figuras

Figura 3.1: Secuenciamiento ordenes de problema ilustrativo.....	12
Figura 3.2: Evolución del inventario para los productos 1, 2 y 3.....	13
Figura 4.1: Metaheurística MCGA (pseudocódigo).....	16
Figura 4.2: Operador de cruzamiento OX.....	17
Figura 4.3: Operador de mutación Swap.....	18
Figura 5.1: Evolución de valor Déficit para el problema ilustrativo M3.....	28

Capítulo 1: Introducción

La integración y globalización de los mercados, sumados con el desarrollo de nuevas tecnologías han asegurado el crecimiento del comercio internacional, lo cual añade mayor presión sobre las empresas para desarrollar estrategias que les permitan mantener un buen nivel de competitividad. Debido a lo anterior, las compañías se ven forzadas a diferenciarse de sus competidores y mantenerse en una búsqueda constante de nuevas formas de producir y realizar sus operaciones, logrando proporcionar productos y servicios con mayor eficacia y eficiencia.

Uno de los aspectos fundamentales en la gestión de una empresa enfocada en la optimización de sus procesos y mejora continua, es la gestión de los procesos logísticos, en especial el cumplimiento en las entregas a clientes y la gestión de inventarios, los cuales han recibido gran atención de investigadores y ha evolucionado constantemente hasta lograr convertirse en una de las herramientas esenciales en las organizaciones para alcanzar buenos niveles de competitividad.

Una gestión eficiente de inventarios requiere de la toma de decisiones en uno de los temas más desafiantes para la administración, como lo es la planificación y programación de la producción, la cual se encarga de la asignación eficiente de recursos buscando cumplir con los objetivos de producción en un determinado horizonte de tiempo. Parte importante de la planificación y programación de producción se centra en el problema de definir los tamaños de lotes a procesar, identificando cuánto y cuándo producir cierto producto.

La gestión de inventarios tiene una directa relación con la productividad y rendimiento de un sistema, lo cual influye en la capacidad de una organización para competir en el mercado. Es por lo anterior que el desarrollo y mejora de metodologías de solución para el problema de dimensionamiento de lotes de producción representa un aporte importante al desarrollo y crecimiento de las empresas, y debe continuar siendo estudiado por investigadores, para lograr desarrollar metodologías que puedan adaptarse a los distintos tipos de compañías y sus procesos.

Esta tesis se enfoca en la resolución del problema de dimensionamiento de lotes y programación de producción en una máquina, múltiples productos y setup dependientes de secuencia. Esta investigación surge como una generalización al trabajo realizado por Iribarren (2011) en una empresa manufacturera dedicada a la fabricación de bolas de molienda para la minería.

Se propone una metaheurística híbrida para lograr la conformación de lotes de producción y posterior secuenciación de las órdenes de producción, la cual combina muestreo aleatorio con un algoritmo genético, con el objetivo de disminuir la cantidad de producto entregada en forma atrasada a los clientes. Se busca así encontrar una mejora en la solución del problema, desarrollando un método que logre ser más eficiente en la resolución de este tipo de problemas.



Capítulo 2: Dimensionamiento de lotes y programación de producción

El dimensionamiento de lotes en la programación de la producción ha sido estudiado ampliamente en la literatura y existen una gran variedad de extensiones las cuales han surgido a través del tiempo con la intención de perfeccionar los modelos existentes adoptando diversas características de situaciones reales. Si bien es reconocido y estudiado no existe un consenso o clasificación general para este tipo de problemas y sus extensiones, por lo que se tienen diversas nomenclaturas y clasificaciones. Debido a lo anterior en este capítulo nos centraremos en la exposición de algunos de los principales tipos de problemas mencionados en la literatura, junto a una revisión de metodologías de solución para estos, los cuales tienen relación con el tema a tratar en esta investigación, con la finalidad de contextualizar los diversos tipos de problemas que existen y sus objetivos.

2.1. Historia y desarrollo

El problema de dimensionamiento de lotes y programación se centra en encontrar un programa de producción que logre satisfacer la demanda de los clientes, cuyo objetivo generalmente es minimizar los costos asociados al programa de producción. Las principales decisiones de las que se preocupa es determinar cuándo se producirá cierto producto y los tamaños de lotes a producir, donde un lote corresponde la producción de un mismo producto en uno o varios periodos de tiempo consecutivos.

Existe una gran variedad de problemas dependiendo de las características que se tomen en consideración en cuando a aspectos como el tipo de horizonte de planificación, la naturaleza de los parámetros, la estructura de suministro del problema y la cantidad de productos que se consideran, entre otros. Estos aspectos tienen una gran influencia en la dificultad de resolución del problema y en los métodos de solución a aplicar. Los problemas de dimensionamiento de lotes pueden considerar un horizonte de planificación finito o infinito, parámetros determinísticos o estocásticos, múltiples productos o un único tipo de producto en proceso, puede considerarse un problema enfocado en una cadena de suministro con múltiples niveles o una sola etapa del proceso.

Debido a esta diversidad de características es que el problema de dimensionamiento de lotes ha sido ampliamente estudiado y múltiples modelos y extensiones han sido investigados desde sus inicios.

La investigación del problema de dimensionamiento de lotes nace con el modelo EOQ por su nombre en inglés *Economic Order Quantity* (Harris, 1913), el cual se centra en encontrar el tamaño de una

orden de reposición de inventario que minimice los costos de almacenamiento, costos de preparación y producción. El EOQ asume un problema determinístico y estacionario, considerando un solo producto y sin restricciones de capacidad. El modelo EOQ ha tenido variadas extensiones y variaciones a través del tiempo con el objetivo de lograr mayor complejidad y mejorar las soluciones encontradas.

Una de las primeras variaciones publicadas corresponde al ELSP o *Economic lot scheduling problem* el cual mantiene las características del EOQ de ser un problema que considera múltiples productos, determinístico y constante en el tiempo, sin embargo, integra restricciones de capacidad, siendo un problema clasificado como NP-Hard (Drexl, 1997).

Otro modelo que ha cobrado importancia dentro de los investigadores es el CLSP o *Capacitated lot sizing problem*. El CLSP asume un horizonte de planificación finito el cual se encuentra dividido en períodos de tiempo discretos, posee una demanda asociada a cada período y que puede variar en el tiempo, además de incluir restricciones de capacidad. La principal característica de este modelo es que permite que múltiples productos sean procesados en un mismo período de tiempo, por ende, se debe planificar la producción con tal de que al finalizar el período se cumpla la demanda de todos los productos procesados en el período (Drexl, 1997).

Una variación del CLSP es el modelo DLSP o *Discrete lot sizing and scheduling problem*, el cual a diferencia del CLSP, permite que solo un producto sea procesado en cada período de tiempo y al hacerlo debe utilizar la capacidad completa de este, es decir, se procesa un producto a toda su capacidad o el período permanece vacío (Fleischmann, 1994).

Una extensión del DLSP corresponde al CSLP o *Continuous setup lot sizing problem*, el cual es similar al anterior, sin embargo, elimina la restricción de utilizar la capacidad completa de un período, permitiendo que las cantidades a producir sean de cualquier tamaño continuo, aún con la restricción de producir solo un ítem por período.

Una mejora para ocupar la capacidad restante de un período determinado del CLSP es el modelo PLSP o *Proportional lot sizing and scheduling problem* el cual permite ocupar la capacidad sin utilizar para procesar un segundo ítem en el período y el GLSP o *General lot sizing and scheduling problem* el cual permite que múltiples productos sean procesados en el mismo período (Glock et al., 2014).

2.2. Revisión bibliográfica

Se presenta una revisión de la literatura enfocada principalmente en el CLSP o *Capacitated lot sizing problem*, explicado en la sección anterior, con el objetivo de presentar las metodologías utilizadas en la resolución de este tipo de problemas. Se selecciona el CLSP en esta ocasión, debido a que es aquel que más se asemeja al problema a tratar en este informe.

Es conocido que el CLSP posee una dificultad NP-Hard, además, al incorporar tiempos de preparación o setup tiene una dificultad NP-Completo (Drexler, 1997). Debido a que resolver el problema de forma óptima tiene un alto nivel de dificultad es que el campo ha sido dominado principalmente por heurísticas o metaheurísticas.

Se observa que los métodos de solución óptimos son utilizados principalmente para resolver problemas de menor escala, como en el trabajo de Kwak & Jeong (2011) que considera el problema CLSP, de una máquina y múltiples productos con tiempos y costos de setup dependientes de secuencia, con el objetivo de minimizar costos. Propone una solución de dos niveles donde en el primer nivel se resuelve el problema de determinar los tamaños de lotes y en el segundo nivel se realiza el secuenciamiento de las ordenes de producción. El problema se resuelve de forma óptima para un número reducido de productos, lo cual es comparado con otras soluciones óptimas del CLSP y heurística existente, demostrando tener una mayor eficiencia computacional.

Por otro lado, Gómez-Herrera et al. (2013) se enfoca en el problema de dimensionamiento de lotes encontrado en una compañía productora de materias primas para la industria panadera, considerando un sistema productivo de dos etapas, determinístico, permitiendo órdenes atrasadas, con múltiples productos, cuyo setup es independiente de la secuencia, donde se tiene como objetivo minimizar los costos totales. Para resolver el problema se presenta un modelo de programación lineal entera, el cual es implementado en tres instancias diferentes. Se observa que, en comparación al método actual utilizado por la compañía, el método propuesto logra encontrar tamaños de lotes más equitativos que flexibiliza las operaciones y el manejo de inventario, teniendo lotes que permiten evitar déficit de stock de mejor manera, encontrándose una solución con una brecha de 10% con la solución óptima.

Una línea de trabajo que tiene una alta presencia en la resolución de problemas de dimensionamiento de lotes son las heurísticas, principalmente utilizadas para resolver problemas de una mayor cantidad de productos en menor tiempo.

Ríos-Solís et al. (2020) considera un problema de dimensionamiento de lotes para múltiples productos, los cuales deben ser procesados en múltiples máquinas, donde cada producto posee un tiempo de proceso que varía para cada máquina. Se propone una heurística iterativa de dos etapas basada en programación matemática, donde en una primera etapa se determina los tamaños de lotes y en una segunda etapa se realiza la programación de dichos lotes en las diversas máquinas. El método propuesto se prueba en problemas generados aleatoriamente y en problemas de compañías reales, demostrándose que es posible llegar a soluciones factibles en un número reducido de iteraciones.

Schmidt et al (2019) considera el problema de dimensionamiento de lotes y programación de máquinas paralelas, buscando mantener una sincronía entre las etapas del proceso productivo, estableciendo restricciones de tiempos muertos. Se desarrollan dos modelos matemáticos y diversas estrategias de resolución heurísticas para resolver problemas de prueba reales, donde se comprueba que los métodos heurísticos logran soluciones competitivas en tiempos computacionales reducidos, logrando desempeñarse de mejor forma en comparación a métodos de optimización.

Shim et al. (2011) consideran el CLSP con setup dependientes de secuencia, con el objetivo de minimizar los costos de setup y de mantenimiento de inventarios. Para resolver este problema se propone una heurística de dos etapas, la cual consiste en una mejora a una heurística existente. En la primera etapa se obtiene una solución inicial y se determinan los tamaños de lotes, secuenciando en cada periodo aquellos productos que posean demanda en dicho período y son secuenciados de acuerdo a los costos de setup, luego se trabaja de atrás hacia adelante ajustando la secuencia de acuerdo con la capacidad de cada periodo. En la siguiente etapa se mejora la solución encontrada en la etapa anterior, enfocándose en minimizar los costos, intentando disminuir los setup. Esto se hace a través de mejoras hacia atrás y hacia adelante en la secuencia, junto con un set de reglas de priorización para seleccionar los trabajos que serán movidos en la secuencia. Luego, se prueba la heurística en instancias de pequeñas, instancias grandes e instancias de artículos de la literatura. Se probó que la heurística propuesta es significativamente mejor que la existente en la literatura, además de demostrar una mejora frente al método del artículo utilizado de benchmark.

Gupta & Magnusson (2005) consideran el problema de una máquina con múltiples productos y periodos, el cual se formula como un problema de programación entera mixta. Debido a la complejidad del problema se formula una heurística para resolver instancias mayores, llamada Heurística ISI. La heurística consiste en un primer paso de inicialización donde se obtienen los

tamaños de lote a producir para cada producto en cada periodo, esto se hace secuenciando los productos de forma decreciente de acuerdo con la demanda acumulada a través del tiempo. Mediante una revisión desde el último periodo hasta el primero se revisa la capacidad y cada vez que se viola una restricción de capacidad el exceso se cambia al periodo inmediatamente anterior, considerando el producto con menor costo de inventario, para luego realizar el mismo procedimiento desde el primer periodo hasta el último, intentado eliminar todo el sobretiempo. Posteriormente, se continua con la etapa de secuenciamiento, la cual se lleva a cabo mediante una heurística greedy. Finalmente, se tiene la etapa de mejoramiento la cual intenta mejorar la solución obtenida disminuyendo los costos.

Ocampo & Salazar (2014) consideran el ELSP con múltiples productos y setup dependientes de secuencia, para el cual se propone un enfoque de solución mixto, donde se utiliza una heurística para la generación de secuencias de lotes, trabajando de forma creciente, para luego ser evaluadas por un modelo de optimización continua no lineal, el cual determina el tamaño de los lotes, con el objetivo de minimizar el costo total de administración de inventario.

Almada-Lobo et al. (2007) consideran el CLSP para una máquina, con tiempos y costos de setup dependientes de secuencia, para el cual se presentan dos formulaciones de programación lineal entera mixta. Debido a que el problema es *NP-Hard* se presenta una heurística de cinco pasos para resolver instancias de mayor tamaño, la cual consiste principalmente de secuenciar los lotes de productos en cada periodo con tal de satisfacer la demanda de dicho periodo, sin considerar restricciones de capacidad, para luego tener un procedimiento de secuenciamiento y una rectificación de la capacidad en cada periodo, donde desde el último periodo hacia el primero se mueven aquellos lotes que exceden la capacidad a periodos previos, finalmente, los siguientes pasos intentan mejorar la solución encontrada.

Otra línea de trabajo en este tipo de problemas se enfoca en el uso de metaheurísticas como se puede ver en el trabajo de Wang et al. (2012), el cual se enfoca en el problema de dimensionamiento de lotes de una compañía con una cadena de suministro enfocada en la fabricación a pedido, considera setup y no permite ordenes atrasadas. Debido a que el problema es considerado NP-Hard se propone una metodología de solución utilizando *tabu search*, la cual es testeada utilizando instancias aleatorias pequeñas y grandes. Para instancias pequeñas se compara con el método *branch and bound*, encontrando que la metodología alcanza soluciones optimas o cercanas en poco tiempo de

funcionamiento. Para instancias grandes se compara el método con una cota inferior, donde se logra verificar la eficiencia del método *tabu search*.

Ceschia et al. (2017) considera el problema de dimensionamiento y programación de lotes, discreto, para una máquina y múltiples productos, donde se propone una metodología metaheurística en base a *simulated annealing*, la cual es comparada con metodologías presentes en la literatura basados en programación entera mixta, para esto se utilizan problemas de prueba existentes y problemas nuevos. Se logra comprobar que el método propuesto logra encontrar soluciones cercanas al óptimo en tiempos reducidos para todas las instancias de prueba.

Wang & Luo (2016) se centran en el problema de dimensionamiento de lotes y programación de un sistema de producción de múltiples productos, se busca minimizar los costos logísticos, asegurando un cierto nivel de servicio al cliente. Para esto se propone una metodología híbrida utilizando algoritmos genéticos y *simulated annealing*, la cual es comparada con cotas inferiores del problema, teniéndose resultados favorables en cuando a calidad de solución y eficiencia.

Un problema ELSP que permite órdenes atrasadas es tratado por Gonçalves & Sousa (2011). Para resolverlo se plantea un modelo no lineal entero mixto, pero debido a la complejidad de este se desarrolla una solución híbrida combinando algoritmos genéticos y una formulación de programación lineal. La solución probó que la metodología es efectiva y robusta comparado con las metodologías comparadas.

Finalmente, Mirabi (2011) considera el CLSP de una máquina y múltiples productos con setup dependientes de secuencia, con el objetivo de disminuir los costos de setup, costos de inventario y costos de ordenes atrasadas. Debido a que se trata de un problema *NP-Completo* se utiliza una metaheurística para su resolución, la cual consta de una primera fase de inicialización para lograr obtener una solución inicial y una segunda fase de mejora de la solución utilizando *simulated annealing*. Para lograr evaluar el método utilizado se compara con heurísticas encontradas en la literatura y con una cota inferior del problema la cual se obtiene resolviendo una relajación lineal de este, donde se observa que el método tiene un mejor desempeño que otras heurísticas y logra resultados en un menor tiempo.

Capítulo 3: Planteamiento del problema

El presente trabajo trata del problema de dimensionamiento de lotes y programación de producción para una máquina, múltiples productos y tiempos de setup dependientes de la secuencia. En la Sección 3.1 se presenta una descripción detallada del problema tratado y su objetivo.

3.1 Descripción del problema

El presente problema se centra en un sistema constituido por una única máquina, donde se realiza el procesamiento de diversos tipos de productos, durante un horizonte de planificación finito y períodos de tiempo determinados. Se puede procesar un producto a la vez en la máquina y múltiples productos pueden ser procesados en cada periodo. Al comenzar el trabajo en la máquina y al realizar un cambio en el tipo de producto procesado se debe llevar a cabo un tiempo de preparación o setup, el cual sirve para preparar la máquina para el procesamiento de otro tipo de producto, la cantidad de tiempo empleado en esta actividad dependerá del tipo de producto que sale y el tipo de producto que entra a la máquina. Además, cada tipo de producto posee una tasa conocida de proceso en la máquina. La demanda de cada tipo de producto es conocida y constante en el tiempo. Se asume que la máquina tiene capacidad para procesar toda la demanda.

El problema está enfocado en aumentar el nivel de servicio al cliente, el cual representa el grado de satisfacción de los clientes y está ligado a la capacidad de la empresa de satisfacer la demanda de sus clientes en los tiempos deseados. Esto se puede ver reflejado en la cantidad de entregas atrasadas que se tienen en cierto periodo de tiempo

De esta forma, el objetivo del problema es establecer los tamaños de lotes de cada tipo de producto y secuenciar dichas ordenes de producción de forma que se logre el cumplimiento de la demanda disminuyendo la cantidad de entregas atrasadas.

Parámetros

A continuación, se describen los parámetros del problema:

$T = n^{\circ}$ de periodos del horizonte de programación

$N = n^{\circ}$ de tipos de productos

$loteMin_i =$ lote mínimo del producto tipo i ($i = 1, \dots, N$)

$tasa_i =$ tasa de producción del producto tipo i ($i = 1, \dots, N$)

$s_{ij} =$ tiempo de setup dependiente de la secuencia (paso de producto i al j)

$d_{it} =$ demanda producto i en periodo t ($i = 1, \dots, N; t = 1, \dots, T$)

$nMax_i =$ máximo de lotes del producto tipo i ($i = 1, \dots, N$)

$$nMax_i = \left\lceil \frac{\sum_{t=1}^T d_{it}}{loteMin_i} \right\rceil$$

$M_i =$ número de lotes del producto tipo i ($i = 1, \dots, N; 1 \leq M_i \leq nMax_i$)

$M =$ número total de órdenes de producción (lotes) a secuenciar; $M = \sum_{i=1}^N M_i$

$tipo(k) =$ tipo de producto de orden k ($k = 1, \dots, M$)

$O_k =$ tamaño de orden k ($k = 1, \dots, M$)

$p_k =$ tiempo de producción de orden k ($k = 1, \dots, M$); $p_k = O_k / tasa_{tipo(k)}$

$x_{it} =$ cantidad producida del producto tipo i en el periodo t ($i = 1, \dots, N; t = 1, \dots, T$)

$I_{it} =$ inventario del producto tipo i al final del periodo t ($i = 1, \dots, N; t = 1, \dots, T$)

Supuestos del problema

Para dar contexto al problema se supone, sin pérdida de generalidad, la producción de un horizonte de un mes subdividido en 4 periodos de una semana ($T = 4$), sin inventario al inicio del horizonte de programación (en caso de que éste exista se corrige la demanda restándole este inventario).

3.2 Medida de desempeño

Dada la importancia de la entrega a tiempo de la mayor cantidad de productos a los clientes, se define la medida de desempeño *Déficit*, la que permite medir la calidad de las soluciones y de esta forma tener un criterio de selección de las mejores soluciones.

El *Déficit* consiste en la cantidad de producción entregada en forma retrasada en el horizonte de programación, compuesta por la suma del déficit observado periodo a periodo (semana a semana) para cada producto: Def_{it} . Se realiza el balance de inventario para cada tipo de producto al final de cada periodo, esto es, al inventario inicial del periodo se le suma la producción del periodo y se le resta la demanda del periodo:

$$I_{it} = I_{i,t-1} + x_{it} - d_{it}$$

Si $I_{it} < 0$ existe un déficit igual a $-I_{it}$ para el producto i en el periodo t , esto es:

$$Def_{it} = \max(0; -I_{it})$$

De esta forma, la mejor solución será aquella que posea el menor déficit total dentro de todas las soluciones encontradas. El déficit total de una solución será la suma del déficit resultante para cada tipo de producto en todos los periodos del horizonte de programación:

$$Déficit = \sum_{t=1}^T \sum_{i=1}^N Def_{it}$$

3.3 Ejemplo ilustrativo

Para ejemplificar la medida de desempeño descrita anteriormente, consideramos un problema sencillo con 3 tipos de productos, donde el número de órdenes de producción y la cantidad de producción en cada lote se encuentra definida arbitrariamente. Se consideran los siguientes parámetros:

- Horizonte de planificación: $T = 1$ (equivalente a 100 hrs.)

- Demanda (ton): $d_{it} = \begin{pmatrix} 45 \\ 30 \\ 50 \end{pmatrix}$

- Tasa de producción (ton/hr): $tasa_i = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

- Tiempos de setup (hr): $s_{ij} = \begin{pmatrix} 0 & 6 & 7 \\ 6 & 0 & 8 \\ 7 & 8 & 0 \end{pmatrix}$

En cuanto a las órdenes de producción, se consideran órdenes de igual tamaño para cada tipo de producto:

- Tamaño orden (ton): $O_k = (15 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15 \ 15)$

- Tipo de producto de cada orden: $tipo(k) = (1 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 3 \ 3)$

Se tienen la siguiente cantidad de lotes del mismo tamaño por tipo de producto, la suma de los cuales componen la demanda total del periodo:

- Número de lotes por tipo de producto: $M_i = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix}$

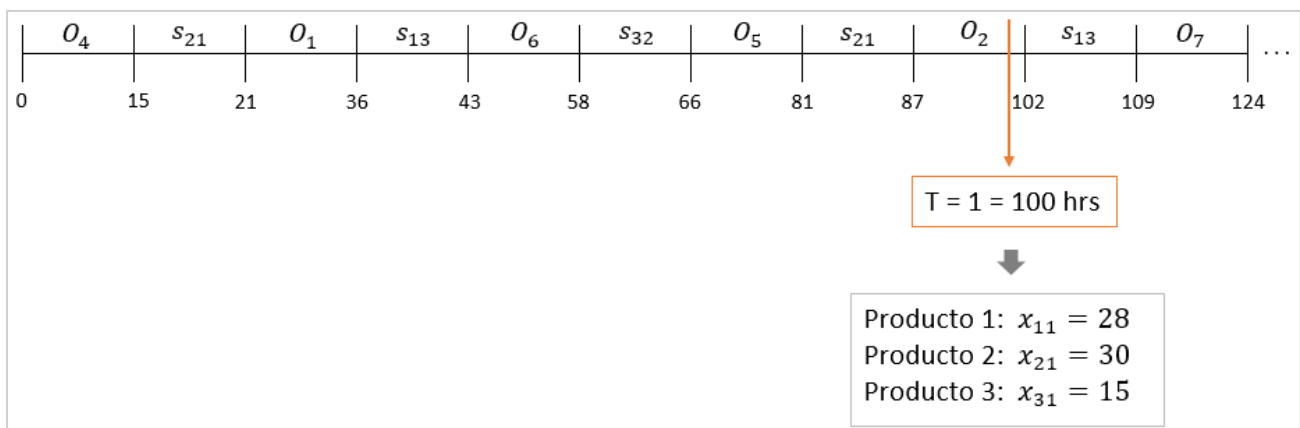
Para el secuenciamiento de las órdenes de producción (lotes) se considera la producción alternada de lotes de forma creciente según la demanda total de cada producto. Por ende, la secuencia de producción es la siguiente:

- Secuencia: $O_4 - O_1 - O_6 - O_5 - O_2 - O_7 - O_3 - O_8 - O_9$

Al realizar la programación de esta secuencia de producción se obtienen las siguientes producciones por tipo de producto para el horizonte de planificación establecido:

$$x_{it} = \begin{pmatrix} 15 \\ 30 \\ 15 \end{pmatrix}$$

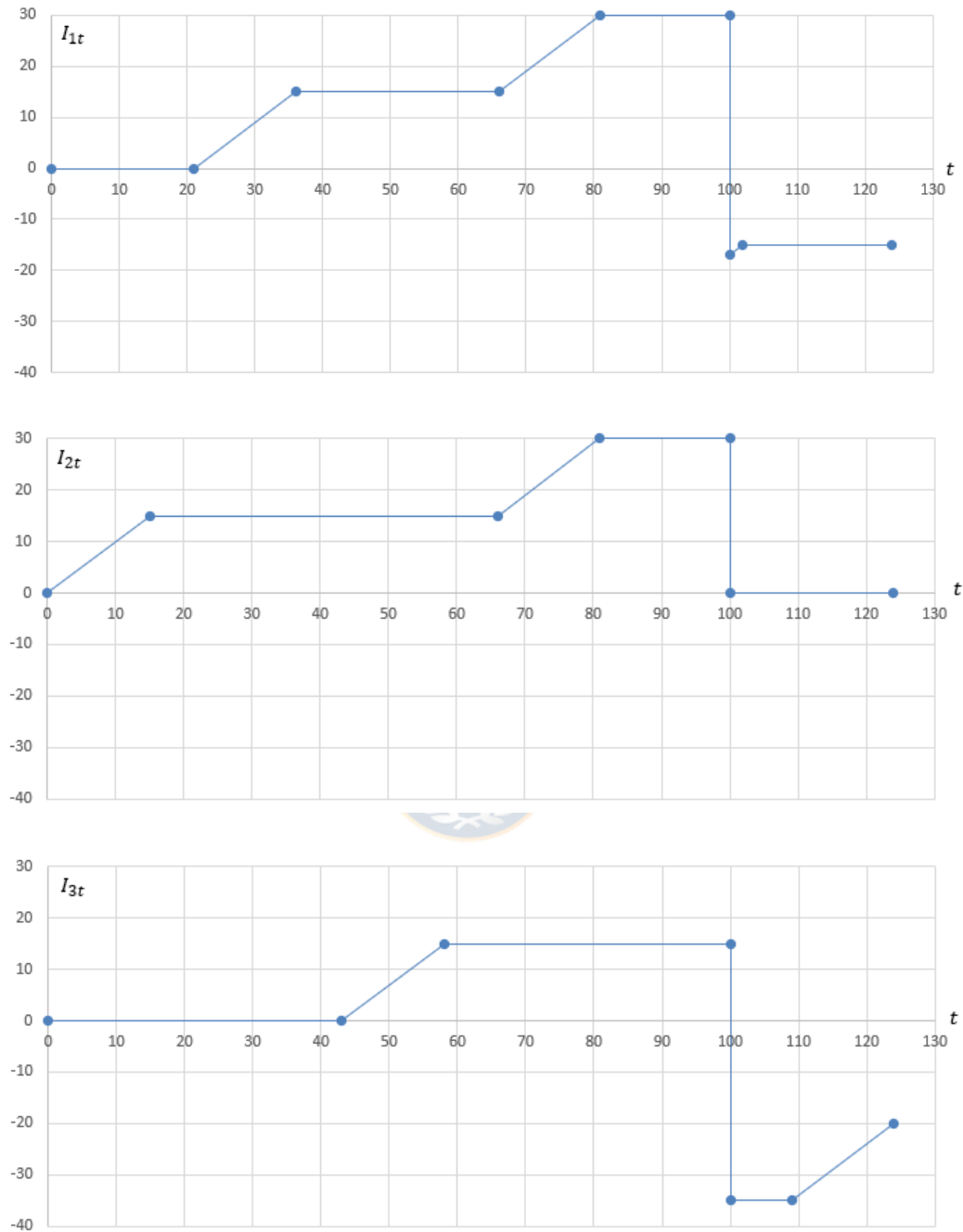
En la Figura 3.1 se ilustra la secuencia de proceso de las órdenes de producción (O_k) y los respectivos tiempos de setup (s_{ij}) que se llevan a cabo, según lo descrito anteriormente. También, se observa la línea temporal en la que se desarrollan las distintas operaciones.



Fuente: Elaboración propia

Figura 3.1: Secuenciamiento ordenes de problema ilustrativo

La Figura 3.2 muestra la evolución del inventario para los productos 1, 2 y 3.



Fuente: Elaboración propia

Figura 3.2: Evolución del inventario para los productos 1, 2 y 3

Con respecto a la secuencia de órdenes de producción programada, la medida de desempeño *Déficit* se calcula de la siguiente forma para cada tipo de producto (ver evolución del inventario para cada tipo de producto en la Figura 3.2):

▪ **Producto tipo 1:**

- Balance de inventario ($i = 1 ; t = 1$):

$$I_{11} = I_{1,0} + x_{11} - d_{11} = 0 + 28 - 45 \quad \rightarrow \quad I_{11} = -17$$

- Medida de desempeño ($i = 1 ; t = 1$):

$$Def_{11} = \max(0 ; -I_{11}) = \max(0 ; 17) \quad \rightarrow \quad Def_{11} = 17$$

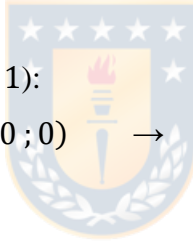
▪ **Producto tipo 2:**

- Balance de inventario ($i = 2 ; t = 1$):

$$I_{21} = I_{2,0} + x_{21} - d_{21} = 0 + 30 - 30 \quad \rightarrow \quad I_{21} = 0$$

- Medida de desempeño ($i = 2 ; t = 1$):

$$Def_{21} = \max(0 ; -I_{21}) = \max(0 ; 0) \quad \rightarrow \quad Def_{21} = 0$$



▪ **Producto tipo 3:**

- Balance de inventario ($i = 3 ; t = 1$):

$$I_{31} = I_{3,0} + x_{31} - d_{31} = 0 + 15 - 50 \quad \rightarrow \quad I_{31} = -35$$

- Medida de desempeño ($i = 3 ; t = 1$):

$$Def_{31} = \max(0 ; -I_{31}) = \max(0 ; 35) \quad \rightarrow \quad Def_{31} = 35$$

Por lo tanto, el resultado final obtenido para la secuencia programada es:

$$Déficit = \sum_{t=1}^1 \sum_{i=1}^3 Def_{it} = Def_{11} + Def_{21} + Def_{31} = 17 + 0 + 35 \quad \rightarrow \quad Déficit = 52$$

Este proceso se debe realizar para todas las secuencias de producción probadas, donde la que posea el menor valor de Déficit será aquella que entrega los mejores resultados y, por ende, la que será elegida finalmente para la resolución del problema.

Capítulo 4: Metaheurística MCGA

Considerando la dificultad del problema presentado es que se opta por el desarrollo de una metaheurística para su resolución. Para ello se decide utilizar una integración de algoritmos genéticos con muestreo Montecarlo. Los algoritmos genéticos es uno de los métodos más utilizados en la resolución de problemas clásicos de programación de producción, debido a su efectividad en la resolución de problemas de optimización combinatoria y la calidad de solución que entregan (Santander-Mercado et al., 2016).

La determinación del número de lotes para satisfacer la demanda total de un tipo de producto i (M_i) se considera entre 1 y $nMax_i$ (considerar un mayor número de lotes llevaría a tener lotes por bajo el tamaño mínimo de lote para el respectivo tipo de producto). Por otro lado, el problema de determinar el tamaño del lote dentro de la limitación de lote mínimo no es un problema trivial, como fue probado por Goerler, et al (2016), el tamaño de lote mínimo puede influenciar la solución del problema de forma positiva acelerando el tiempo de resolución de este, o de forma negativa llevando a un aumento en la cantidad de producción almacenada en inventario o incluso a un problema con una solución no factible.

En este trabajo se plantea que los M_i lotes del producto i sean de igual tamaño. Así, M_i , el número de lotes del producto i , se genera en forma aleatoria con distribución uniforme discreta entre 1 y $nMax_i$.

La Figura 4.1 muestra la estructura (pseudocódigo) de la metaheurística MCGA diseñada para resolver este problema, la que refleja la integración entre el muestreo aleatorio mediante del número de lotes M_i para cada tipo de producto i , definiendo las $M = \sum_{i=1}^N M_i$ órdenes de producción a secuenciar en la máquina aplicando algoritmos genéticos.

La metaheurística genera un número de problemas de secuenciación de una máquina con tiempos de preparación dependiente de la secuencia igual a *muestra*. Para cada uno de estos problemas se generan en forma aleatoria el número de lotes para cada tipo de producto, lo que permite definir el conjunto de M órdenes de producción a secuenciar y sus características (tipo de producto, cantidad de producto y tiempo de proceso), para luego resolver el problema de secuenciación de una máquina con tiempos de preparación dependientes de la secuencia mediante un algoritmo genético. El proceso

completo se repetirá un número definido de veces (tamaño de muestra), guardando cada vez la mejor solución encontrada.

```

metaheurística MCGA()
{
  sb ← solución_inicial()
  m ← 1
  while (m ≤ muestra)
  {
    for (i=1, ..., N) {generar Mi}
    DefinirOP()
    s ← SecuenciarOP()
    if ( f(s) < f(sb) ) {sb ← s }
    m ← m+1
  }
  Salida: sb
}

```

Fuente: Elaboración propia

Figura 4.1: Metaheurística MCGA (pseudocódigo)

4.1 Conformación de tamaños de lotes mediante Montecarlo

Mediante la información de la demanda total de cada producto y el tamaño de lote mínimo a producir se obtiene el número máximo de lotes posibles de conformar para cada tipo de producto i :

$$nMax_i = \left\lfloor \frac{\sum_{t=1}^T d_{it}}{loteMin_i} \right\rfloor$$

Luego, mediante la utilización del método de Montecarlo, se genera el número M_i de lotes a conformar para cada tipo de producto i . Este se genera en el intervalo discreto $[1, nMax_i]$ asumiendo una distribución uniforme discreta (todos los valores tienen igual probabilidad de ser generados). Si $M_i = 1$, el único lote conformado es por el total de la demanda del producto i ; si $M_i > 1$ los M_i lotes serán del mismo tamaño.

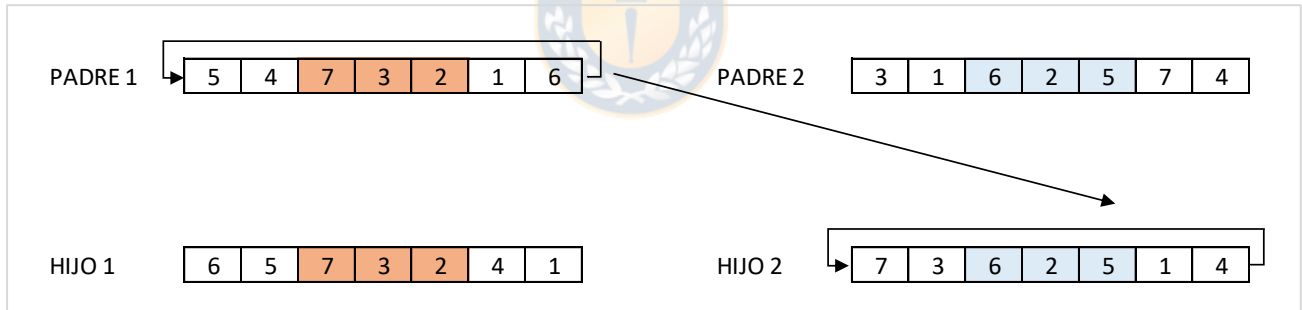
El tiempo de proceso para la orden k ($k = 1, \dots, M$) se obtiene multiplicando la tasa de producción del tipo de producto asociado a la orden k por la cantidad asociada a la orden k , esto es: $p_k = tasa_{tipo(k)} \cdot o_k$.

4.2 Secuenciación de órdenes de producción mediante Algoritmos Genéticos

Los lotes conformados por cada tipo de producto en el paso anterior corresponden a las órdenes de producción a secuenciar a continuación, las cuales serán programadas mediante un algoritmo genético. Se define un individuo (cromosoma) como la secuencia de todas las órdenes de producción.

La probabilidad de selección de un individuo se basa en un criterio de aptitud, este criterio corresponde al *Déficit*, el cual fue definido en el Capítulo 3. A menor valor de *Déficit* los individuos tendrán mayor probabilidad de ser seleccionados.

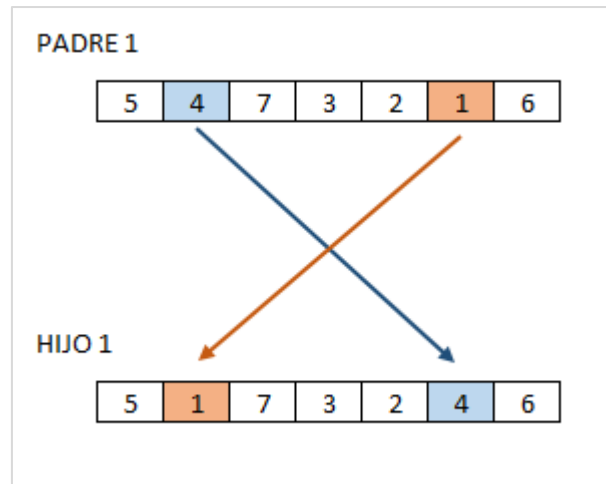
El algoritmo utiliza el operador genético de cruzamiento *OX* (*order crossover*), el cual selecciona aleatoriamente dos puntos de corte en la cadena del cromosoma (secuencia de trabajos) en ambos padres. Los trabajos comprendidos dentro de los dos puntos de corte se transfieren manteniendo las posiciones a la descendencia. Luego, los trabajos restantes de un padre se transfieren a partir de la parte final del cromosoma del respectivo padre manteniendo el orden de los trabajos excluyendo aquellos trabajos que ya fueron transferidos desde el otro padre a la parte central del hijo. En la Figura 4.2 se ilustra el operador *OX*.



Fuente: Elaboración propia

Figura 4.2: Operador de cruzamiento OX

Además, se utiliza el operador genético de mutación de intercambio (*swap*), el cual selecciona aleatoriamente dos trabajos en el cromosoma elegido intercambiando dichas posiciones. En la Figura 4.3 se muestra una representación del operador de mutación *swap*.



Fuente: Elaboración propia
Figura 4.3: Operador de mutación Swap



Capítulo 5: Experimentos

Para probar la efectividad del método desarrollado, se procede a realizar una serie de experimentos numéricos. Para ello se utiliza como punto de comparación el trabajo realizado en una empresa manufacturera, el cual sirvió de motivación para la realización y desarrollo de esta investigación. Para realizar lo anterior, se utilizan los problemas evaluados en dicho trabajo y se procede a resolverlos mediante la metaheurística desarrollada en esta tesis. De esta forma se comparan los resultados obtenidos mediante el método desarrollado en esta tesis con los obtenidos en el problema original.

Adicionalmente, se realiza la misma comparación de resultados con ambos métodos utilizando instancias de prueba generadas aleatoriamente.

En el presente capítulo se exponen los problemas de prueba que se utilizaron para la realización de los experimentos y la metodología que servirá de punto de comparación para establecer el desempeño del método desarrollado en esta tesis.

Los valores de los parámetros utilizados en la experimentación se decidieron en base a valores frecuentemente utilizados en la literatura:

- Tamaño de muestra Montecarlo = 100
- Tamaño población algoritmo genético = 50
- Generación aleatoria de la población inicial del algoritmo genético.
- Número de generaciones del algoritmo genético = 100
- Probabilidad de cruzamiento = 0.7, 0.8 y 0.9
- Probabilidad de mutación = 0.01, 0.05 y 0.1
- Número de réplicas del algoritmo genético en cada muestra = 10

5.1 Problemas de prueba

Para la realización de experimentos numéricos se utiliza como base el trabajo realizado en una empresa productora de bolas de molienda para la minería, enfocándose en el proceso de una de las máquinas de producción donde se procesan diversos tipos de productos, manteniendo los tiempos de producción diferentes para cada producto y tiempos de setup dependientes de la secuencia (Iribarren, 2011). Además, se tiene que la compañía posee una política de tamaño de orden mínimo para poder

comenzar a producir. En dicho trabajo se propone un método de solución de búsqueda en vecindad IP. La determinación de tamaños de lotes se realiza de forma manual (externa al método implementado). Siguiendo criterios técnicos, la demanda total de cierto producto es dividida en partes iguales para cada lote conformado, teniendo en consideración la restricción de tamaño de lote mínimo. Una vez determinados los lotes se procede a secuenciar dichas ordenes de producción mediante una búsqueda de vecindad.

Se seleccionan 3 problemas evaluados en Iribarren (2011), los que se diferencian en el número de productos considerados y sus demandas. Cada uno de estos problemas fue resuelto repetidas veces, variando el número y tamaños de lotes, donde cada combinación corresponde a un subproblema. Como se mencionó anteriormente, en el trabajo de Iribarren (2011) la determinación de lotes se realiza de forma manual externa al método utilizado, por lo que también se detallan aquí las combinaciones de números y tamaños de lotes que se utilizaron para resolver el problema.

Tabla 5.1: Parámetros generales Problemas 1, 2 y 3.

Parámetro	Valor parámetro
N° de productos	$N = 6$
Tasas de producción	$tasa_i = \begin{pmatrix} 4.1 \\ 5.8 \\ 6.6 \\ 9.2 \\ 9.2 \\ 9.2 \end{pmatrix}$
Tamaño de lote mínimo	$M_i = \begin{pmatrix} 500 \\ 500 \\ 500 \\ 500 \\ 500 \\ 500 \end{pmatrix}$
Tiempos de setup	$s_{ij} = \begin{pmatrix} 0 & 4 & 5 & 6 & 7 & 8 \\ 4 & 0 & 4 & 5 & 6 & 7 \\ 5 & 4 & 0 & 4 & 5 & 6 \\ 6 & 5 & 4 & 0 & 4 & 5 \\ 7 & 6 & 5 & 4 & 0 & 4 \\ 8 & 7 & 6 & 5 & 4 & 0 \end{pmatrix}$

Fuente: Iribarren (2011)

En la Tabla 5.1 se muestran los parámetros generales de los 3 problemas considerados (Problema 1: serie 950, Problema 2: serie 1040; Problema 3: serie 1160). Se considera la producción mensual (periodificada en 4 semanas) de 6 productos de diferente tipo con tasas de producción expresadas en toneladas/hora, tamaño mínimo del lote de producción de 500 toneladas para todos los productos y tiempos de preparación (setup) dependientes de secuencia expresados en horas.

La demanda mensual por tipo de producto expresada en toneladas/semana se muestra en la Tabla 5.2. Notar que no en todos los problemas se considera la producción de todos los productos (el Problema 1 no considera la producción del producto tipo 2, el Problema 2 no considera la producción de los productos tipo 2 y 4, mientras que el Problema 3 considera la producción de todos los productos).

Tabla 5.2: Demanda Problemas 1, 2 y 3.

Problema	Demanda (d_{it})
Problema 1: serie 950	$d_{it} = \begin{pmatrix} 0 & 0 & 140 & 140 \\ 0 & 0 & 0 & 0 \\ 84 & 84 & 196 & 168 \\ 224 & 28 & 28 & 0 \\ 128 & 0 & 156 & 28 \\ 812 & 644 & 952 & 1176 \end{pmatrix}$
Problema 1: serie 1040	$d_{it} = \begin{pmatrix} 0 & 0 & 168 & 84 \\ 0 & 0 & 0 & 0 \\ 560 & 140 & 476 & 224 \\ 0 & 0 & 0 & 0 \\ 84 & 56 & 84 & 56 \\ 840 & 476 & 350 & 420 \end{pmatrix}$
Problema 1: serie 1160	$d_{it} = \begin{pmatrix} 112 & 112 & 224 & 28 \\ 0 & 28 & 0 & 28 \\ 0 & 56 & 168 & 84 \\ 0 & 0 & 56 & 0 \\ 56 & 56 & 56 & 56 \\ 784 & 756 & 84 & 84 \end{pmatrix}$

Fuente: Iribarren (2011)

Tabla 5.3: Parámetros de instancias correspondientes a Problemas 1, 2 y 3.

Problema	Instancia	Parámetros
Problema 1: serie 950	1.1	$O_k = (500, 532, 500, 500, 3584)$ $tipo(k) = (1, 3, 4, 5, 6)$ SI: $O_5 - O_3 - O_4 - O_2 - O_1$
	1.2	$O_k = (500, 532, 500, 500, 1792, 1792)$ $tipo(k) = (1, 3, 4, 5, 6, 6)$ SI: $O_5 - O_2 - O_4 - O_1 - O_3 - O_6$
	1.3	$O_k = (500, 532, 500, 500, 1195, 1195, 1194)$ $tipo(k) = (1, 3, 4, 5, 6, 6, 6)$ SI: $O_5 - O_6 - O_2 - O_4 - O_3 - O_1 - O_7$
Problema 2: Serie 1040	2.1	$O_k = (500, 1400, 500, 2086)$ $tipo(k) = (1, 3, 5, 6)$ SI: $O_4 - O_2 - O_3 - O_1$
	2.2	$O_k = (500, 1400, 500, 1043, 1043)$ $tipo(k) = (1, 3, 5, 6, 6)$ SI: $O_1 - O_3 - O_2 - O_5 - O_4$
	2.3	$O_k = (500, 700, 700, 500, 2086)$ $tipo(k) = (1, 3, 3, 5, 6)$ SI: $O_1 - O_4 - O_3 - O_2 - O_5$
Problema 3: serie 1160	3.1	$O_k = (500, 500, 500, 500, 500, 1708)$ $tipo(k) = (1, 2, 3, 4, 5, 6)$ SI: $O_6 - O_1 - O_3 - O_4 - O_5 - O_2$
	3.2	$O_k = (500, 500, 500, 500, 500, 854, 854)$ $tipo(k) = (1, 2, 3, 4, 5, 6, 6)$ SI: $O_7 - O_1 - O_6 - O_3 - O_4 - O_2$

Fuente: Elaboración propia

La Tabla 5.3 muestra el detalle de las instancias evaluadas para los Problemas 1, 2 y 3. Por ejemplo para el Problema 1 (serie 950) se evaluaron 3 instancias (1.1, 1.2 y 1.3). La instancia 1.1 contempla la producción de un lote por producto con la demanda total, la instancia 1.2 contempla la subdivisión de la demanda total en dos lotes siempre y cuando la subdivisión mantenga la restricción de tamaño mínimo de lote (esto lo cumple sólo el producto 6), y la instancia 1.3 contempla la subdivisión de la demanda total en 3 lotes de igual tamaño siempre y cuando la subdivisión mantenga la restricción de tamaño mínimo de lote (esto lo cumple sólo el producto 6).

A modo de ilustración, para la instancia 1.1 se tienen por lo tanto 5 órdenes de producción: O_1 , O_2 , O_3 , O_4 y O_5 de tipo de producto 1, 3, 4, 5 y 6, con tamaño de lote de 500, 532, 500, 500 y 3584 toneladas respectivamente. La solución inicial (SI) considerada para resolver la instancia 1.1 con el método de búsqueda en vecindad IP fue la secuencia $O_5 - O_3 - O_4 - O_2 - O_1$ (de acuerdo con el criterio de priorizar aquellas órdenes de productos con mayores demandas en las primeras semanas del mes).

5.2 Instancias aleatorias

Para la generación de instancias aleatorias se toma como base el método de generación de la literatura que trata el problema CLSP con tiempos de setup dependientes de secuencia (Almada-Lobo, et al., 2007), cuyas características se asemejan al problema tratado en esta investigación.

Los parámetros se generan de la siguiente manera:

- *Demanda*: $d_{it} = \text{Uniforme}(40, 60)$
- *Tiempos de setup*: $s_{ij} = \text{Uniforme}(5, 10)$
- *Tasa de producción*: $tasa_i = 1$
- *Tamaño de lote mínimo*: $M_i = 30$

Los problemas generados son resueltos utilizando la metaheurística MCGA propuesta en esta investigación, comparando los resultados mediante el método de búsqueda en vecindad IP utilizado por Iribarren (2011). Para este método, por cada problema generado se resuelven 3 instancias distintas donde se prueba con diferentes configuraciones de tamaños de lotes elaborados de forma manual siguiendo la siguiente lógica y verificando el cumplimiento de lote mínimo:

- Configuración 1: Órdenes completas, sin subdivisión
- Configuración 2: La demanda de cada tipo de producto es subdividida en 2 lotes
- Configuración 3: La demanda de cada tipo de producto es subdividida en 3 lotes

Para el establecimiento de la solución inicial, se secuencian las órdenes siguiendo la regla SPT (*Shortest Processing Time*), donde los trabajos se secuencian en orden creciente en cuanto a su tiempo de proceso. Debido a que las tasas de proceso son unitarias para todos los productos, la secuencia corresponde al orden creciente en la cantidad de demanda de cada orden de producción.

Se generan 3 problemas, denominados M1, M2 y M3, donde se varía el número de productos (N), tiempos de setup y demanda (ver Tabla A.1 del Anexo) y el detalle de las configuraciones de N , $2N$

y $3N$ para cada problema (o_k , $tipo(k)$ y $solución\ inicial$) que se muestran en la Tabla A.2 del Anexo.

La Tabla 5.4 resume el número de productos y el número de órdenes de producción de cada configuración para los Problemas M1, M2 y M3.

Tabla 5.4: Tamaño configuraciones por instancia de Problemas M1, M2 y M3.

Problema	N	Configuración	Órdenes de producción
M1	10	1.1	10
		1.2	20
		1.3	30
M2	9	2.1	9
		2.2	18
		2.3	27
M3	8	3.1	8
		3.2	16
		3.3	24

Fuente: Elaboración propia

Con el fin de ilustrar el proceso de optimización que lleva a cabo la metaheurística MCGA aplicada, se considera la resolución del problema M3 expuesto anteriormente. A continuación, se procede a exponer el desarrollo de la metaheurística propuesta, modificando el tamaño de muestra Montecarlo a 3 y el número de réplicas del algoritmo genético en cada muestra a 2 para simplificar el proceso y lograr ejemplificarlo de mejor forma.

En primer lugar, se tiene que el tamaño de orden mínimo corresponde a: $loteMin_i = 3$. Por otro lado, la demanda total de cada producto es

$$\sum_{t=1}^T d_{it} = \begin{pmatrix} 199 \\ 209 \\ 203 \\ 200 \\ 202 \\ 192 \\ 176 \\ 201 \end{pmatrix}$$

El número máximo de lotes por producto es calculado

$$nMax_i = \left\lfloor \frac{\sum_{t=1}^T d_{it}}{loteMin_i} \right\rfloor$$

En caso de tener un resultado de $nMax_i$ decimal, este valor será aproximado al entero inferior. De esta forma, para el problema se tienen los siguientes valores de número máximo de lotes para cada producto:

$$\text{Producto 1: } nMax_1 = \left\lfloor \frac{199}{30} \right\rfloor = \lfloor 6.6 \rfloor = 6 \quad \text{Producto 5: } nMax_5 = \left\lfloor \frac{202}{30} \right\rfloor = \lfloor 6.7 \rfloor = 6$$

$$\text{Producto 2: } nMax_2 = \left\lfloor \frac{209}{30} \right\rfloor = \lfloor 6.9 \rfloor = 6 \quad \text{Producto 6: } nMax_6 = \left\lfloor \frac{192}{30} \right\rfloor = \lfloor 6.4 \rfloor = 6$$

$$\text{Producto 3: } nMax_3 = \left\lfloor \frac{203}{30} \right\rfloor = \lfloor 6.7 \rfloor = 6 \quad \text{Producto 7: } nMax_7 = \left\lfloor \frac{176}{30} \right\rfloor = \lfloor 5.8 \rfloor = 5$$

$$\text{Producto 4: } nMax_4 = \left\lfloor \frac{200}{30} \right\rfloor = \lfloor 6.6 \rfloor = 6 \quad \text{Producto 8: } nMax_8 = \left\lfloor \frac{201}{30} \right\rfloor = \lfloor 6.7 \rfloor = 6$$

Luego de determinado el número máximo de lotes posibles por tipo de producto, se procede a determinar el número de lotes a procesar por cada tipo de producto (M_i), lo cual se realiza aplicando el método Montecarlo a una distribución uniforme discreta en el intervalo $[1, nMax_i]$. Con el valor de número de lotes por producto, se determina el tamaño de cada lote (O_k), lo cual se realiza de la siguiente forma dividiendo el total de la demanda por el número de lotes, teniendo así tamaños de lotes iguales. Este proceso, que llamaremos Muestreo, se realiza en total 3 veces. Los resultados de número de lotes y tamaño de lotes para cada muestreo se observan en la Tabla 5.5.

Tabla 5.5: Ordenes de producción generadas para problema ilustrativo M3

Muestreo 1				Muestreo 2				Muestreo 3			
Lotes por producto	Orden	Tipo	Lote	Lotes por producto	Orden	Tipo	Lote	Lotes por producto	Orden	Tipo	Lote
$M_i = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 3 \\ 6 \\ 2 \\ 4 \\ 3 \end{pmatrix}$	O_1	1	199	$M_i = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 3 \\ 2 \\ 4 \\ 2 \\ 4 \end{pmatrix}$	O_1	1	66	$M_i = \begin{pmatrix} 6 \\ 1 \\ 3 \\ 6 \\ 6 \\ 4 \\ 5 \\ 2 \end{pmatrix}$	O_1	1	33
	O_2	2	104		O_2	1	66		O_2	1	33
	O_3	2	105		O_3	1	67		O_3	1	33
	O_4	3	67		O_4	2	209		O_4	1	33
	O_5	3	67		O_5	3	203		O_5	1	33
	O_6	3	69		O_6	4	66		O_6	1	34
	O_7	4	66		O_7	4	66		O_7	2	209
	O_8	4	66		O_8	4	68		O_8	3	67
	O_9	4	68		O_9	5	101		O_9	3	67
	O_{10}	5	33		O_{10}	5	101		O_{10}	3	69
	O_{11}	5	33		O_{11}	6	48		O_{11}	4	33
	O_{12}	5	33		O_{12}	6	48		O_{12}	4	33
	O_{13}	5	33		O_{13}	6	48		O_{13}	4	33
	O_{14}	5	33		O_{14}	6	48		O_{14}	4	33
	O_{15}	5	37		O_{15}	7	88		O_{15}	4	33
	O_{16}	6	96		O_{16}	7	88		O_{16}	4	35
	O_{17}	6	96		O_{17}	8	50		O_{17}	5	33
	O_{18}	7	44		O_{18}	8	50		O_{18}	5	33
	O_{19}	7	44		O_{19}	8	50		O_{19}	5	33
	O_{20}	7	44		O_{20}	8	51		O_{20}	5	33
	O_{21}	7	44						O_{21}	5	33
	O_{22}	8	67						O_{22}	5	37
	O_{23}	8	67						O_{23}	6	48
	O_{24}	8	67						O_{24}	6	48
						O_{25}	6		48		
						O_{26}	6		48		
						O_{27}	7		35		
						O_{28}	7		35		
						O_{29}	7		35		
						O_{30}	7		35		
						O_{31}	7		36		
						O_{32}	8		100		
						O_{33}	8		101		

Fuente: Elaboración propia

Por cada muestreo realizado se ejecutan 2 réplicas del algoritmo genético, el cual se encarga de secuenciar las ordenes de producción establecidas en el proceso de anterior. En cada réplica se ejecutan hasta un máximo de 100 generaciones hasta lograr llegar a la mejor solución considerando la medida de desempeño *Déficit*. De esta forma, se considerará como la mejor solución para el muestreo realizado, aquella que haya generado el menor déficit entre ambas réplicas del algoritmo genético ejecutado.

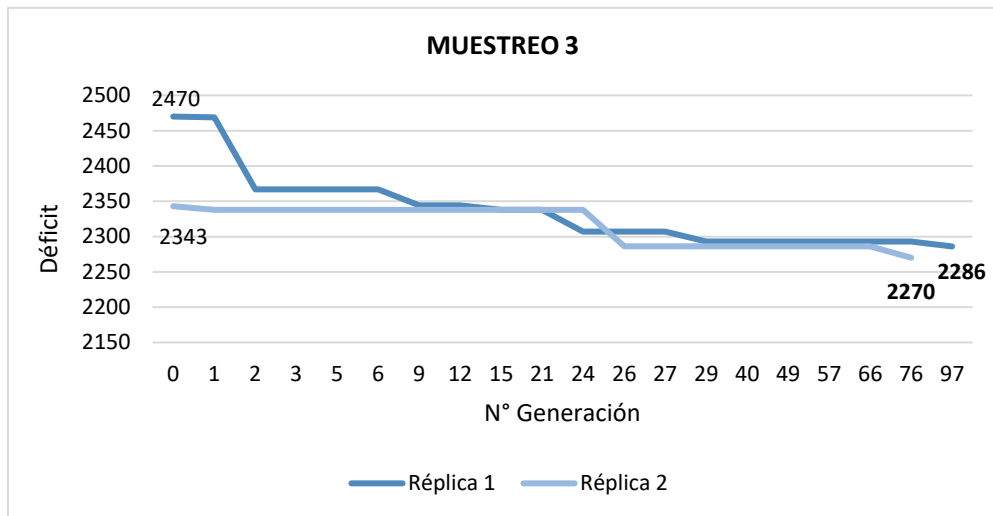
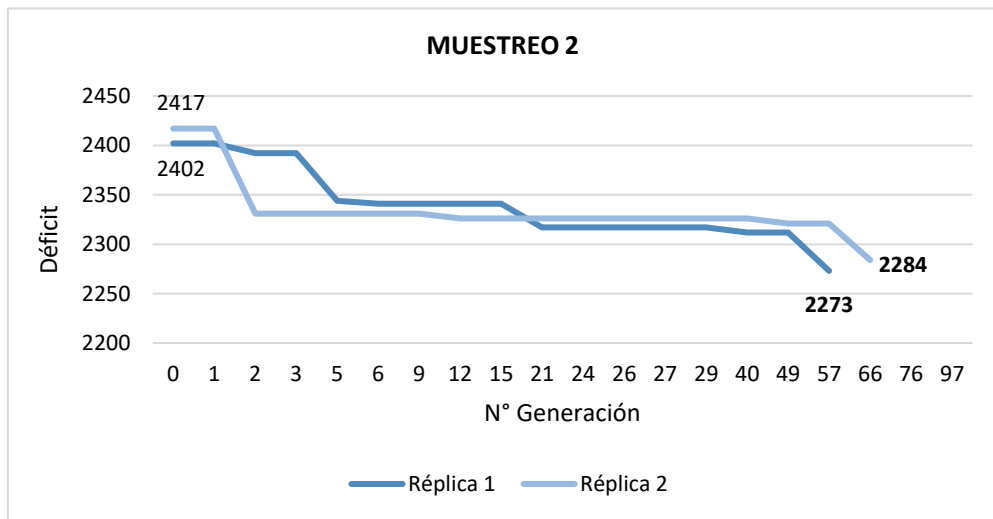
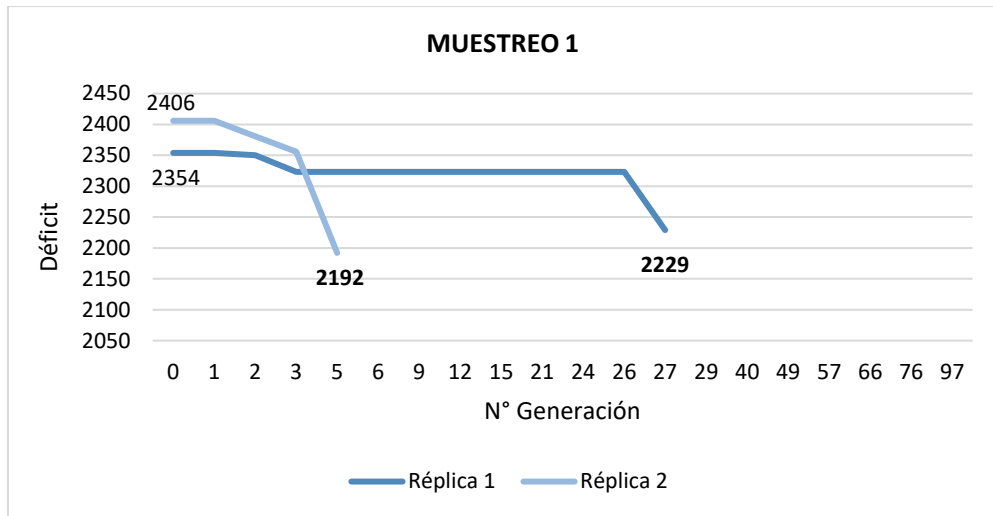
En la Figura 5.1 se puede apreciar la evolución que tiene el valor del *Déficit* a lo largo del proceso iterativo utilizado por la metaheurística MCGA en cada uno de los muestreos generados y réplicas correspondientes.

Tabla 5.6: Valores de Déficit obtenidos para problema ilustrativo M3

Muestreo 1				Muestreo 2				Muestreo 2			
Réplica 1		Réplica 2		Réplica 1		Réplica 2		Réplica 1		Réplica 2	
Déficit	Gen.	Déficit	Gen.	Déficit	Gen.	Déficit	Gen.	Déficit	Gen.	Déficit	Gen.
2354	0	2406	0	2402	0	2417	0	2470	0	2343	0
2350	2	2381	2	2392	2	2331	2	2469	1	2338	1
2323	3	2356	3	2344	5	2326	12	2367	2	2286	26
2229	27	2192	5	2341	6	2321	49	2344	9	2270	76
				2317	21	2284	66	2338	15		
				2312	40			2307	24		
				2273	57			2293	29		
								2286	97		

Fuente: Elaboración propia

Luego de completadas todas las iteraciones de la metaheurística se selecciona como resultado final del problema aquella configuración de lotes y secuencia de producción que haya generado el menor valor de déficit dentro de todos los muestreos generados. Los resultados obtenidos para cada muestreo y sus correspondientes réplicas se encuentran en la Tabla 5.6.



Fuente: Elaboración propia

Figura 5.1: Evolución de valor Déficit para el problema ilustrativo M3

Capítulo 6: Resultados

Se llevan a cabo los experimentos numéricos, utilizando los problemas de prueba expuestos en el capítulo anterior, de acuerdo con la metaheurística MCGA propuesta en este trabajo y explicada en los Capítulos 4 y 5, utilizando el programa SPS_MCGA_Scheduler adaptando rutinas en lenguaje C/C++ de SPS_Optimizer (Salazar, 2010). Los resultados producidos por la metaheurística MCGA se comparan con la búsqueda en vecindad IP de Iribarren (2011). Todos los experimentos fueron realizados en un computador equipado con un procesador Intel Core i5-8350U, con 1,70 GHz y 4 GB de RAM.

Los tiempos de ejecución, expresados en segundos, de todos los problemas resueltos utilizando la metodología MCGA se encuentran en la Tabla 6.1.

Tabla 6.1: Tiempos de ejecución de problemas de prueba utilizando metodología MCGA

Problema	Tiempo ejecución (seg)
Problema 1	34,29
Problema 2	31,62
Problema 3	34,23
M1	62,67
M2	55,4
M3	53,56

Fuente: Elaboración propia

En la realización de los experimentos numéricos se varían los parámetros utilizados en el algoritmo genético, específicamente la probabilidad de cruzamiento y mutación, según lo descrito en el Capítulo 5, en el caso de los Problemas 1, 2 y 3 no se observan diferencias en los resultados obtenidos al variar los parámetros indicados.

Para el Problema 1 se observa que la metodología MCGA resuelve el problema utilizando 10 órdenes de producción, mejorando la solución en un 37,3% en comparación con la mejor solución del problema original, la cual considera 7 órdenes de producción. Además, de mejorar la solución un 68,3% en comparación a la peor solución, la cual no considera ninguna subdivisión de lotes. Los resultados obtenidos para el Problema 1 se pueden observar en la Tabla 6.2.

Tabla 6.2: Resultados obtenidos para Problema 1

Método	Problema	Medida de desempeño (Déficit)
MCGA	Problema 1	305
Búsqueda de vecindad IP	Instancia 1.1	963
	Instancia 1.2	387
	Instancia 1.3	487

Fuente: Elaboración propia

Por otro lado, el Problema 2 es el que presenta la mayor reducción del déficit resolviéndolo con la metaheurística MCGA, teniéndose una mejora en la solución de un 88,2%, utilizando 8 órdenes de producción, en comparación con la mejor solución encontrada con IP, la cual utiliza 5 lotes para su resolución, y de un 91,2% en comparación con la peor solución del problema original. Los resultados obtenidos para este problema se pueden observar en la Tabla 6.3.

Tabla 6.3: Resultados obtenidos para Problema 2

Método	Problema	Medida de desempeño (Déficit)
MCGA	Problema 2	84
Búsqueda de vecindad IP	Instancia 2.1	965
	Instancia 2.2	715
	Instancia 2.3	901

Fuente: Elaboración propia

Observando los resultados obtenidos para el Problema 3, los cuales se encuentran en la Tabla 6.4, se puede observar que también la solución generada por el método MCGA posee un mejor desempeño que aquella generadas utilizando el método IP, observándose una mejora de hasta un 39,6%.

Tabla 6.4: Resultados obtenidos para Problema 3

Método	Problema	Medida de desempeño (Déficit)
MCGA	Problema 3	169
Búsqueda de vecindad IP	Instancia 3.1	280
	Instancia 3.2	178

Fuente: Elaboración propia

Por otro lado, se realizó la misma comparación para los problemas generados aleatoriamente, en este caso se puede observar la misma tendencia anterior, donde la metodología MCGA entrega soluciones de mejor calidad. Los resultados obtenidos se pueden observar en la Tabla 6.5.

En los Problemas M1, M2 y M3 se observaron diferencias en el Déficit según los valores de la probabilidad de cruzamiento y mutación, por lo que en este capítulo se expondrán los mejores valores en cada problema, para revisar la totalidad de los resultados obtenidos ver Anexo B.

Tabla 6.5: Resultados obtenidos para generados aleatoriamente M1, M2 y M3.

Método	Problema	Medida de desempeño (Déficit)
MCGA	Problema M1	2876
	Problema M2	2538
	Problema M3	2009
Búsqueda en vecindad IP	Configuración 1.1	3506
	Configuración 1.2	3189
	Configuración 1.3	3146
	Configuración 2.1	3132
	Configuración 2.2	2719
	Configuración 2.3	2735
	Configuración 3.1	2622
	Configuración 3.2	2175
	Configuración 3.3	2253

Fuente: Elaboración propia

Capítulo 7: Conclusiones

La presente investigación se centró en el problema de dimensionamiento de lotes y programación de producción en una máquina, múltiples productos y setup dependientes de la secuencia. Este trabajo propone una mejora al método de Iribarren (2011) que resuelve el problema de programación de producción y dimensión de lote en una empresa manufacturera de bolas de molienda para la minería, y al mismo tiempo propone una generalización de este problema que puede ser aplicado a otros entornos productivos. Entendiendo además la importancia de los inventarios en la economía y como las decisiones tomadas en este aspecto afecta directamente la capacidad de una empresa de competir en el mercado.

Para la resolución del problema se desarrolla una metaheurística que integra muestreo aleatorio utilizando el método Montecarlo que genera el número de lotes de igual tamaño a procesar por tipo de producto, los cuales pasan a ser las órdenes de producción del problema. Luego, las órdenes de producción se secuencian utilizando un algoritmo genético. Se tiene como objetivo principal la minimización del Déficit, es decir, la cantidad total de producto entregado atrasado como medida de desempeño.

La metodología propuesta, llamada MCGA, es testada, utilizando como punto de comparación el método y los problemas de prueba evaluados en el trabajo de Iribarren (2011), trabajo que motivó esta investigación. Además, se generaron instancias de prueba aleatorias, las cuales son probadas con ambos métodos, para así comprobar la efectividad de la metodología desarrollada en esta tesis.

En los resultados obtenidos se puede apreciar que la metodología MCGA logra obtener mejores resultados en todas las instancias probadas, en comparación con el método IP. Se tiene una mejora de hasta un 68,3% en la resolución del Problema 1 presentado, donde se llega a la solución utilizando una mayor cantidad de lotes de producción. Por otro lado, en el Problema 2 se observa una mejora de hasta un 88,2% en comparación con la mejor solución encontrada con el método IP y de un 39,6% en el Problema 3.

Se puede observar la misma tendencia de mejora en las instancias aleatorias generadas, las cuales presentaban una mayor flexibilidad en el número de lotes considerados, sin embargo, la metodología MCGA logra soluciones que mejoran los resultados en un 6,8% en el Problema M1, además, de lograr

resultados de un 4,9% mejores en el Problema M2 y de un 3,2% en el Problema M3, en comparación con la mejor solución obtenida con IP. Además, se ve una mejora en los resultados al variar el número de lotes a conformar en las distintas soluciones propuesta con IP, lográndose soluciones de mejor calidad al subdividir los productos en más de un lote.

Con lo anterior, podemos observar un desempeño del método propuesto que cumple con el objetivo de la investigación, logrando proporcionar soluciones de mejor calidad en cuando a la disminución de las órdenes atrasadas en un sistema, lo cual puede llegar a proporcionar un aumento en el nivel de servicio al cliente en una organización.

Se logra comprobar el buen desempeño de las soluciones metaheurísticas, especialmente integrando metaheurísticas como en este caso lo fue la integración de muestreo aleatorio Montecarlo con algoritmos genéticos, a la hora de resolver problemas de elevada dificultad, comportándose de forma deseada entregando soluciones de calidad en tiempos computacionales reducidos.

Con los resultados obtenidos, se aprecia la importancia de la conformación de los lotes a producir, donde encontrar el tamaño y número ideal proporciona mejores soluciones. Debido a que no existe una regla general, donde una mayor cantidad de lotes no significa necesariamente una mejor solución, es necesario tener una metodología que pueda proveer flexibilidad a la hora de realizar el dimensionamiento de lotes, pudiendo hacer de este un proceso eficaz, pero en tiempos que sean compatibles con los tiempos de las decisiones en sistemas reales. Es en este aspecto donde la solución propuesta posee también la flexibilidad para la conformación de lotes en tiempos adecuados.

Finalmente, debido a que esta investigación se basa en un problema aplicado en una organización real, se muestra la importancia de tener un sistema de programación de producción y dimensionamiento de lotes de producción desarrollado, utilizando las herramientas existentes en la literatura, ya que se puede mejorar el desempeño de un proceso de forma significativa. Este trabajo aporta un nuevo método para resolver el problema conjunto de la programación y determinación de lotes de producción.

Referencias

- Almada-Lobo, B., Klabjan, D., Carravilla, M. A., & Oliveira, J. F., 2007. Single machine multi-product capacitated lot sizing with sequence-dependent setup. *International Journal of Production Research*, pp. 4873-4894.
- Drexl, A., & Kimms, A., 1997. Lot sizing and scheduling - Survey and extensions. *European Journal of operational research*, pp. 221-235.
- D. Wang and H. Luo, "Simultaneous Lot-Sizing and Scheduling for Single-Stage Multi-product Production-Distribution System with Due Date Considerations to Minimize Total Logistics Cost," 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2016, pp. 200-203.
- Fleischmann, B., 1994. The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. *European Journal of Operational Research*, pp. 395-404.
- Glock, C. H., Grosse, E. H., & Ries, J. M., 2014. The lot sizing problem: A tertiary study. *Int. J. Production Economics*, pp. 39-51.
- Gómez-Herrera, J. A., Escobar, J. W., & Figueroa-Cabrera, Á., 2013. A multi-product lot-sizing model for a manufacturing company. *Ingeniería Investigación y Tecnología*, pp. 413-419.
- Gonçalves, J. F., & Sousa, P. S., 2011. A genetic algorithm for lot-sizing and scheduling under capacity constraints and allowing backorders. *International Journal of Production Research*, pp. 2683-2703.
- Gupta, D., & Magnusson, T., 2005. The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. *Computers and operations research*, pp 727-747.
- Harris, F., 1913. Hoy many parts to make at once. *The magazine of management*, pp. 135-136.
- Iribarren, R. A., 2011. Modelo para la programación de la producción con configuración de lotes - Caso de Moly-Cop Chile S.A. Concepción, Chile: Universidad de Concepción.
- Kwak, I.-S., & Jeong, I.-J., 2011. A hierarchical approach for the capacitated lot-sizing and scheduling problem with a special structure of sequence-dependent setups. *International Journal of Production Research*, pp. 7425-7439.

- Mirabi, M., 2011. A hybrid simulated annealing for the single-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times and costs and dynamic release of jobs. *International Journal of Advanced Manufacturing Technology*, pp. 1109-1119.
- Ocampo Azocar, H., & Salazar Hornig, E., 2014. Dimensionamiento de lotes y programación de una máquina para múltiples productos con setup y escasez. *Ingeniería Industrial*, pp. 49-62.
- Salazar, E., 2010. Programación de Sistemas de Producción con SPS_Optimizer. *Revista ICHIO*, 1 (2), pp 33–46.
- Santander-Mercado, A., & Jubiz-Diaz, M., 2016. The economic lot scheduling problem: a survey. *International Journal of Production Research*, pp. 4973-4992.
- Sara Ceschia, Luca Di Gaspero, Andrea Schaerf. Solving discrete lot-sizing and scheduling by simulated annealing and mixed integer programming. *Computers & Industrial Engineering*, Volume 114, 2017, pp. 235-243
- Shim, I.-S., Kim, H.-C., Doh, H.-H., & Lee, D.-H., 2011. A two-stage heuristic for single machine capacitated lot-sizing and scheduling with sequence-dependent setup costs. *Computers & Industrial Engineering*, pp. 920-929.
- T. M. Pinho Schimidt, S. Cassius Tadeu, G. Valentim Loch and C. M. Schenekemberg, 2019. Heuristic approaches to solve a two-stage lot sizing and scheduling problem. *IEEE Latin America Transactions*, pp. 434-443.
- Goerler, Andreas & Voss, Stefan, 2015. Dynamic lot-sizing with rework of defective items and minimum lot-size constraints. *International Journal of Production Research*, pp. 2284-2297.
- Wang, D., Grunder, O., & EL Moudni, A., 2012. A tabu search approach for lot sizing and scheduling problem with arbitrary job sizes and due date considerations. *IFAC Proceedings Volumes*, pp. 170-175.
- Yasmín Á Ríos-Solís, Omar J. Ibarra-Rojas, Marta Cabo, Edgar Possani, 2020. A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production, *European Journal of Operational Research*, pp. 861-873.

Anexos

A. Parámetros utilizados en instancias aleatorias de prueba

Tabla A.1: Parámetros generales instancias aleatorias de problemas M1, M2 y M3

Parámetro	Problema M1	Problema M2	Problema M3
Número de productos (N)	$N = 10$	$N = 9$	$N = 8$
Tiempos de setup (s_{ij})	$\begin{pmatrix} 0 & 5 & 7 & 7 & 5 & 7 & 8 & 7 & 7 & 8 \\ 10 & 0 & 7 & 7 & 6 & 7 & 8 & 10 & 7 & 9 \\ 10 & 9 & 0 & 5 & 6 & 5 & 9 & 9 & 8 & 9 \\ 8 & 8 & 9 & 0 & 10 & 6 & 9 & 8 & 6 & 9 \\ 6 & 6 & 8 & 5 & 0 & 10 & 8 & 7 & 9 & 7 \\ 10 & 5 & 8 & 9 & 7 & 0 & 5 & 5 & 8 & 6 \\ 5 & 8 & 7 & 7 & 8 & 6 & 0 & 6 & 7 & 9 \\ 5 & 5 & 7 & 9 & 10 & 10 & 7 & 0 & 9 & 6 \\ 10 & 7 & 5 & 5 & 9 & 8 & 8 & 7 & 0 & 6 \\ 10 & 7 & 6 & 7 & 6 & 5 & 10 & 9 & 7 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 9 & 6 & 8 & 5 & 6 & 8 & 5 & 7 \\ 10 & 0 & 10 & 7 & 10 & 9 & 6 & 5 & 7 \\ 10 & 6 & 0 & 6 & 9 & 8 & 6 & 6 & 5 \\ 10 & 6 & 7 & 0 & 9 & 7 & 6 & 5 & 7 \\ 9 & 8 & 8 & 10 & 0 & 6 & 10 & 10 & 8 \\ 9 & 9 & 6 & 5 & 5 & 0 & 6 & 8 & 7 \\ 9 & 10 & 7 & 10 & 8 & 6 & 0 & 9 & 10 \\ 5 & 8 & 6 & 9 & 6 & 10 & 8 & 0 & 5 \\ 6 & 7 & 7 & 9 & 8 & 10 & 8 & 6 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 8 & 7 & 8 & 5 & 10 & 6 & 7 \\ 9 & 0 & 8 & 8 & 5 & 5 & 9 & 5 \\ 6 & 6 & 0 & 8 & 6 & 10 & 6 & 7 \\ 8 & 7 & 8 & 0 & 8 & 8 & 10 & 5 \\ 9 & 8 & 10 & 9 & 0 & 7 & 10 & 10 \\ 10 & 6 & 6 & 6 & 8 & 0 & 8 & 10 \\ 10 & 7 & 8 & 9 & 7 & 6 & 0 & 8 \\ 5 & 5 & 7 & 9 & 10 & 8 & 6 & 0 \end{pmatrix}$
Demanda (d_{it})	$d_{it} = \begin{pmatrix} 42 & 47 & 41 & 40 \\ 56 & 41 & 44 & 52 \\ 45 & 42 & 49 & 42 \\ 40 & 57 & 45 & 42 \\ 45 & 44 & 59 & 46 \\ 52 & 50 & 53 & 49 \\ 53 & 59 & 51 & 60 \\ 44 & 47 & 56 & 40 \\ 45 & 59 & 48 & 50 \\ 43 & 49 & 50 & 56 \end{pmatrix}$	$d_{it} = \begin{pmatrix} 55 & 53 & 53 & 41 \\ 51 & 41 & 53 & 54 \\ 43 & 44 & 41 & 48 \\ 59 & 41 & 46 & 46 \\ 52 & 56 & 54 & 49 \\ 50 & 55 & 53 & 47 \\ 47 & 60 & 52 & 50 \\ 56 & 42 & 40 & 43 \\ 55 & 40 & 44 & 41 \end{pmatrix}$	$d_{it} = \begin{pmatrix} 41 & 56 & 53 & 49 \\ 58 & 52 & 59 & 40 \\ 53 & 50 & 53 & 47 \\ 45 & 47 & 49 & 59 \\ 55 & 47 & 52 & 48 \\ 45 & 51 & 52 & 44 \\ 45 & 41 & 41 & 49 \\ 48 & 51 & 48 & 54 \end{pmatrix}$

Fuente: Elaboración propia

Tabla A.2: Parámetros de configuraciones correspondientes a problemas M1, M2 y M3.

Problema	Configuración	Tamaño orden (O_k)	Tipo producto orden ($tipo(k)$)	Solución inicial (SI)
M1	Configuración 1.1	(170; 193; 178; 184; 194; 204; 223; 187; 202; 198)	(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	$O_1 - O_3 - O_4 - O_8 - O_2 - O_5 - O_{10} - O_9 - O_6 - O_7$
	Configuración 1.2	(85; 85; 96; 97; 89; 89; 92; 92; 97; 97; 102; 101; 111; 112; 93; 94; 101; 101; 99; 99)	(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10)	$O_1 - O_2 - O_5 - O_6 - O_7 - O_8 - O_{15} - O_{16} - O_3 - O_4 - O_9 - O_{10} - O_{19} - O_{20} - O_{17} - O_{18} - O_{11} - O_{12} - O_{13} - O_{14}$
	Configuración 1.3	(56; 57; 57; 64; 64; 65; 59; 59; 60; 61; 61; 62; 64; 65; 65; 68; 68; 68; 74; 74; 75; 62; 62; 63; 67; 67; 68; 66; 66; 66)	(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 9, 10, 10, 10)	$O_1 - O_2 - O_3 - O_7 - O_8 - O_9 - O_{10} - O_{11} - O_{12} - O_{22} - O_{23} - O_{24} - O_4 - O_5 - O_{13} - O_6 - O_{14} - O_{15} - O_{28} - O_{29} - O_{30} - O_{25} - O_{26} - O_{16} - O_{17} - O_{18} - O_{27} - O_{19} - O_{20} - O_{21}$
M2	Configuración 2.1	(202; 199; 176; 192; 211; 205; 209; 181; 180)	(1, 2, 3, 4, 5, 6, 7, 8, 9)	$O_3 - O_9 - O_8 - O_4 - O_2 - O_1 - O_6 - O_7 - O_5$
	Configuración 2.2	(101; 101; 99; 100; 88; 88; 96; 96; 105; 106; 102; 103; 104; 105; 90; 91; 90; 90)	(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9)	$O_5 - O_6 - O_{15} - O_{17} - O_{18} - O_{16} - O_7 - O_8 - O_3 - O_4 - O_1 - O_2 - O_{11} - O_{12} - O_{13} - O_9 - O_{14} - O_{10}$
	Configuración 2.3	(67; 67; 68; 66; 66; 67; 58; 59; 59; 64; 64; 64; 70; 70; 71; 68; 68; 69; 69; 70; 70; 60; 60; 61; 60; 60; 60)	(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 9)	$O_7 - O_8 - O_9 - O_{22} - O_{23} - O_{25} - O_{26} - O_{27} - O_{24} - O_{10} - O_{11} - O_{12} - O_4 - O_5 - O_1 - O_2 - O_6 - O_3 - O_{16} - O_{17} - O_{18} - O_{19} - O_{13} - O_{14} - O_{20} - O_{21} - O_{15}$
M3	Configuración 3.1	(199; 209; 203; 200; 202; 192; 176; 201)	(1, 2, 3, 4, 5, 6, 7, 8)	$O_7 - O_6 - O_1 - O_4 - O_8 - O_5 - O_3 - O_2$
	Configuración 3.2	(99; 100; 104; 105; 101; 102; 100; 100; 101; 101; 96; 96; 88; 88; 100; 101)	(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8)	$O_{13} - O_{14} - O_{11} - O_{12} - O_1 - O_2 - O_7 - O_8 - O_{15} - O_5 - O_9 - O_{10} - O_{16} - O_6 - O_3 - O_4$
	Configuración 3.3	(66; 66; 67; 69; 70; 70; 67; 68; 68; 66; 67; 67; 67; 68; 64; 64; 64; 58; 59; 59; 67; 67; 67)	(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8)	$O_{19} - O_{20} - O_{21} - O_{16} - O_{17} - O_{18} - O_1 - O_2 - O_{10} - O_3 - O_7 - O_{11} - O_{12} - O_{13} - O_{14} - O_{22} - O_{23} - O_{24} - O_8 - O_9 - O_{15} - O_4 - O_5 - O_6$

Fuente: Elaboración propia

B. Resultados de instancias aleatorias de prueba

Problema	Probabilidad Mutación	Probabilidad Cruzamiento	Déficit	Tiempo	N° ordenes
M1	0,01	0,7	2980	54,86	32
	0,01	0,8	2961	53,62	33
	0,01	0,9	2966	56,45	30
	0,05	0,7	2894	54,05	23
	0,05	0,8	2876	55,33	27
	0,05	0,9	2881	59,75	30
	0,1	0,7	2937	52,57	34
	0,1	0,8	2932	53,86	30
	0,1	0,9	2981	62	33
M2	0,01	0,7	2538	49,08	31
	0,01	0,8	2600	50,94	20
	0,01	0,9	2582	51,23	23
	0,05	0,7	2581	47,97	27
	0,05	0,8	2604	57,27	30
	0,05	0,9	2605	55,95	29
	0,1	0,7	2596	53,41	33
	0,1	0,8	2584	50,1	20
	0,1	0,9	2515	52,18	28
M3	0,01	0,7	2009	47,66	28
	0,01	0,8	2016	47,17	25
	0,01	0,9	2077	47,14	32
	0,05	0,7	2083	46,58	24
	0,05	0,8	2086	48,1	28
	0,05	0,9	2107	48,57	25
	0,1	0,7	2078	45,58	24
	0,1	0,8	2105	44,95	23
	0,1	0,9	2060	49,41	28

Fuente: Elaboración propia