# Un algoritmo genético para la solución al problema de programación de lotes económicos (ELSP), con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en un mismo ciclo de producción

**Mayra Alejandra Rojas Rueda**

**Trabajo de grado para Optar por el Título de Ingeniero Industrial**

**Director:**

**Carlos Eduardo Díaz Bohórquez**

**Magister en Ingeniería Industrial**

**Codirector:**

**Rafael Hernán Rojas Gualdrón**

**Ingeniero Mecatrónico**

**Universidad Industrial de Santander**

**Facultad de Ingenierías Fisicomecánicas**

**Escuela de Estudios Industriales y Empresariales**

**Bucaramanga**

**2019**

## Agradecimientos

Agradezco en primera instancia a mi Director de proyecto, el docente Carlos Eduardo Díaz Bohórquez por su guía, conocimientos, aportes y solución de dudas para que el trabajo diera continuidad y culminara satisfactoriamente.

Agradezco a mi Codirector, el ingeniero Rafael Hernán Rojas Gualdrón por su tiempo, asesoría, acompañamiento y conocimientos aportados para la realización del trabajo de grado, así mismo por su paciencia y motivación para finalizar con éxito este trabajo.

De igual manera agradezco al grupo de investigación OPALO, por acoger mi proyecto y encaminarlo correctamente a resultados satisfactorios.

Agradezco a la Universidad Industrial de Santander, por abrirme sus puertas durante estos años de aprendizaje, por forjar valiosas herramientas para enfrentar los retos futuros y por enriquecerme de valores y de orgullo de pertenencia.

Finalmente y no menos importantes gracias a la Escuela de Estudios Industriales y Empresariales, a sus administrativos y profesores que dieron los aportes y conocimientos suficientes para un futuro encaminado de grandes vivencias y éxito laboral.

## Dedicatoria

*La vida consta de varias etapas que se van superando al pasar los años con perseverancia, dedicación, entrega, humildad y el deseo de aprender cada día de aquello que nos ayuda a ser mejores, este proyecto y mi paso por la universidad es parte de esas etapas que me han enseñado a crecer como ser humano y a tener una razón social para servir y para retribuir los conocimientos adquiridos a los demás. Dedico esta etapa vivida a Dios que con su fuerza y voluntad logré cumplir con los objetivos trazados para este camino, a Rosalina Rueda y Juan Pablo Zárate, personas que por designios del destino y el ciclo natural de la vida no se encuentran presentes conmigo, pero siempre estarán en mi mente y corazón y que agradezco por cada palabra y enseñanza que aportaron para mí antes de su partida, su recuerdo vive en mi memoria y los amaré por siempre mi nonita y mi Juanpis.*

*Quiero dedicar este logro al ser más importante de mi vida, mi mamá Ana Smith Rojas, por tu cariño incondicional, por enseñarme desde pequeña a trabajar y perseguir mis metas. Gracias por tanto te amo mamita.*

*A Carlos Rojas, mi hermanito lindo, mi ser de luz, el amor de mi vida, gracias por cuidar y proteger de mi cuando yo no podía hacerlo, por darme la felicidad de tener dos hermosos angelitos como sobrinas, gracias por llenarme de fuerzas cuando todo se torna gris, te amo mi manito.*

*A mi tía Rosita, por creer en mí y apoyarme siempre y ser mi inspiración y ejemplo de amor, resistencia y disciplina. Te amo mi tía linda.*

*A Laura Zárate, mi primita hermosa mi hermanita de risas y aventuras, gracias por tu cariño constante, por tus palabras, por tus consejos, te amo de aquí al infinito.*

*Quiero dedicar también unas palabras a todas aquellas personas que me acompañaron en cada etapa, mis amigos y compañeros de universidad, mis compañeros de trabajo, gracias por ser una gran familia, por enseñarme y llenarme de luz cada día.*

*A Nelson Chang, gracias chinito, por el apoyo y amistad sincera en estos años, te admiro y te quiero mucho.*

*A Claudia López, eres el ser más maravilloso, lleno de luz y amor que he conocido, gracias por regalarme un poquito de ello, por ser mi compañera, mi amiga y mi mamá del destino, te quiero muchísimo.*

*A Steffanny Muñoz, mi tefy gracias por ser mi amiga fiel, por escucharme, por quererme y por apoyarme en mis proyectos, por ser una compañera de estudio excepcional y por ser tan hermoso ser humano, te quiero.*

*A Yesika León, mi yes gracias por estos años de amistad y de compañía, por tu paciencia tu cariño, tu calma y por ser mi apoyo incondicional, te quiero.*

*A Diego Amado, gracias por aparecer en mi vida y apoyarme en esta etapa final, por brindarme tu compañía y cariño, te quiero mucho.*

*Finalmente dedico unas palabras a mis hermanos de vida, los mejores amigos que Dios puso en mi camino para apoyarme, enseñarme tantas cosas valiosas, compartir y aprender juntos:*

*Jose Luis Grass, gracias por tantos años de inigualable amistad, por estar conmigo en mis peores y mejores momentos, por apoyarme a pesar de mis equivocaciones, por hacerme sonreír cuando no tenía el valor para hacerlo, eres la persona más sincera, fuerte y leal que conozco, te quiero y eres el mejor regalo que el destino pudo darme.*

*María Angélica Yunez, mi chiquitita gracias por querer lo mejor para mí siempre, por ser mi consejera, por enseñarme tantas cosas valiosas para mi camino, eres una persona disciplinada,* *incansable y capaz, te quiero por todo lo que aportas en mi vida y porque sin tu presencia no sería la misma.*

**Tabla de Contenido**

## Lista de Figuras

## Lista de Tablas

## Lista de Apéndices

## RESUMEN

**TITULO:** UN ALGORITMO GENÉTICO PARA LA SOLUCIÓN AL PROBLEMA DE PROGRAMACIÓN DE LOTES ECONÓMICOS (ELSP), CON TIEMPOS DE ALISTAMIENTO DEPENDIENTES DE LA SECUENCIA Y LA GENERACIÓN DE SUBLOTES EN UN MISMO CICLO DE PRODUCCIÓN.[*]

**AUTOR:** MAYRA ALEJANDRA ROJAS RUEDA[**]

**PALABRAS CLAVES:** ELSP, problema del lote económico y programación, métodos metaheurísticos, lotes, programación, lotes económicos, algoritmo genético, minimización, tiempos de alistamiento, producir diferentes artículos en una sola máquina.

**DESCRIPCIÓN**

Para efectos de esta investigación, se toma en consideración el Problema de Programación de Lotes Económicos, que ha sido caso de estudio por más de 50 años, el cual consiste en producir diferentes artículos en una sola máquina, teniendo en cuenta tiempos de alistamiento que dependen de la secuencia de producción con la particularidad de generar sublotes para el mismo producto dentro del mismo ciclo de producción. Al contemplar dichos aspectos surge un problema de tipo NP-HARD, lo que lleva a la necesidad de implementar métodos metaheurísticos, que permiten dar la mejor solución al problema, permitiendo así mismo la minimización de los costos y contribuyendo al mejoramiento continuo y la competitividad de las organizaciones. Se presenta el problema de ELSP con partición en el lote, para el cual se desarrolla un algoritmo genético que permite obtener una solución satisfactoria comparada con instancias encontradas en la literatura. Los resultados obtenidos muestran que el algoritmo genético propuesto es eficiente dado a que cumple con las restricciones y permite encontrar un valor óptimo del problema con las características planteadas.


[*] Proyecto de grado

[**] Facultad de Ingenierías Físicomecánicas. Escuela de Estudios Industriales y Empresariales. Programa de Ingeniería Industrial. Director. M.Sc. Carlos Eduardo Díaz Bohórquez. Codirector. Ing. Rafael Hernán Rojas Gualdrón.

## ABSTRACT

**TITLE:** A GENETIC ALGORITHM FOR THE SOLUTION TO THE ECONOMIC LOT SCHEDULING PROBLEM (ELSP), WITH TIMES OF DEPENDENT LIFTING OF THE SEQUENCE AND THE GENERATION OF SUBLOTS IN A SAME CYCLE OF PRODUCTION.*



**AUTHOR:** MAYRA ALEJANDRA ROJAS RUEDA**



**KEYWORDS:** ELSP, metaheuristic methods, lots, scheduling, economic lots, Genetic algorithm, minimization, times of enlistment, producing different articles in a single machine.



**DESCRIPTION**

For the purposes of this research, the Economic lots scheduling Problem (ELSP) is taken into account, which has been a case of study for more than 50 years, which consists of producing different articles in a single machine, taking into consideration the times of enlistment that depend of the production sequence with the particularity of generating sub lots for the same product within the same production cycle. When contemplating these aspects arise a problem of type NP-HARD, which leads to the need to implement metaheuristic methods, which allow to give the approximate solution of the problem, also allowing the minimization of costs and contributing to the continuous improvement and the competitiveness of organizations. The problem of ELSP with partition in the lot is presented, for which a genetic algorithm was developed that allows a satisfactory solution compared with instances found in the literature. The results obtained show that the proposed genetic algorithm is efficient given that it complies with the restrictions and allows finding an optimal value of the problem with the proposed characteristics.


---

* Bachelor Thesis


** Faculty of Physicomechanical Engineering. School of Industrial and Business Studies. Industrial Engineering Program. Director. M.Sc. Carlos Eduardo Díaz Bohórquez. Codirector. Ing. Rafael Hernán Rojas Gualdrón

## Introducción

En la actualidad, las organizaciones están en un proceso continuo de mejoramiento en los sistemas de producción y operaciones, por lo cual es importante el conocimiento y adaptación de nuevas metodologías que garanticen un alto nivel de competitividad en el mercado. Las áreas de producción y operación están en la búsqueda de modernización de la planeación, gestión y tecnología para lograr eficiencia y eficacia en los sistemas productivos, de no hacerlo las empresas pierden participación en el mercado. Por ello es necesario desarrollar métodos que generen valor y que sean capaces de estar en el entorno productivo de mejora continua.

Dicho lo anterior el presente documento hace referencia al Problema de Programación de Lotes Económicos (ELSP) por sus siglas en inglés, el cual consiste en la gestión de operaciones y la teoría de inventario que ha sido caso de estudio por un gran número de investigadores durante más de 50 años. El término de Problema de Programación de Lotes Económicos (ELSP), fue utilizado por primera vez por el profesor Rogers (1958), quién extendió el modelo de Cantidad de Orden Económica (EOQ) a la situación en la que se encuentran varios productos que deben ser producidos en la misma máquina, y para lo que se decide el tamaño de lote de cada producto y la secuencia de los mismos. Años más tarde aparecen diferentes consideraciones para enfocar el Problema de Programación de Lotes Económicos (ELSP) clásico, se establece el método de ciclo común por Hanssmann (1962), el cual asume que los ciclos productivos para todos los productos son los mismos. Más adelante está el enfoque de Bomberger (1966), el cual establece el método de periodo básico, que permite tener diferentes ciclos para los diferentes productos suponiendo que cada ciclo es múltiplo entero de un ciclo básico. En el mismo año surge el método del periodo básico extendido, siendo una mejora del mencionado anteriormente, el cual consiste en tener dos periodos básicos consecutivos, cargando cada articulo en esos dos periodos básicos. Por último está el método de modificación de los tamaños de lote atribuido a Dobson (1987), que hace referencia a tener distintos tamaños de lote para cualquier producto durante un programa cíclico. La solución completa al problema de programación de lotes económicos (ELSP) incluye establecer la secuencia de fabricación de los diferentes articulos, y esto implica decir cuándo y cuánto deben producirse cada uno de ellos, es decir el cálculo del tamaño de lote, de tal forma que los costos totales sean mínimos. Respecto al establecimiento de la secuencia se le atribuye a Delporte y Thomas (1977), quienes establecen una serie de heurísticas para decidir el orden de fabricación de los diferentes artículos. Dado a las aproximaciones para comprobar la factibilidad del problema, el (ELSP) es conocido como un problema de tipo NP- hard, (Hsu 1983).

Así mismo después de dar una revisión cronologíca en la literatura, se pretende abordar el problema de programación de lotes económicos (ELSP) para el presente estudio con las siguientes caracteristicas; tomar en cuenta los tiempos de alistamiento (Setup) y la dependencia de la secuencia de fabricación , generando sublotes del mismo producto resolviendolo a través de la metaheurística de algoritmo génetico.

Tabla 1.

*Cumplimiento de Objetivos*

| **Objetivos Específicos** | **Cumplimiento** |
|---|---|
| • Realizar la revisión de literatura sobre Programación de Lotes Económicos (ELSP) y sus variantes con el propósito de identificar la evolución y la aplicabilidad en un entorno real de producción. | Numeral 3 |
| • Formular un modelo matemático para la solución del ELSP con tiempos de alistamiento dependientes de la secuencia y generación de sublotes dentro del mismo ciclo de producción. | Numeral 5 |
| • Desarrollar un algoritmo genético para la solución del problema de ELSP con tiempos de alistamiento dependientes de la secuencia y generación de | Numeral 6<br>Apéndice A |

| | | |
|---|---|---|
| | sublotes dentro del mismo ciclo de producción y programarlo en el lenguaje Matlab. | |
| • | Evaluar el desempeño del algoritmo genético, analizando los datos resultantes y compararlo con instancias de la literatura. | Numeral 7<br>Apéndice B |
| • | Elaborar un artículo académico de carácter publicable que muestre los resultados del trabajo de investigación realizado. | Apéndice C |

## 1. Planteamiento del Problema

Las organizaciones actualmente se encuentran en la búsqueda de herramientas que ayuden a la planeación y gestión en las operaciones para obtener como resultado el mejoramiento en la productividad y estar al nivel de competitividad en el mercado. Tomando en consideración el problema de programación de lotes económicos (ELSP), el cual tiene relación con la gestión de operaciones y la teoría de inventario, programando la producción de lotes económicos con diferentes productos en una sola máquina que se repitan en un tiempo determinado, teniendo conocimiento que si se producen más artículos de diferente tipo en la misma instalación, la frecuencia de producción de cada tipo de artículo aumenta consecuente al balance de la tasa de producción y la tasa de demanda, lo que genera que el tiempo y los costos de configuración sean elevados. Para efectos de esta investigación se estudia el problema de programación de lotes económicos (ELSP) tomando en cuenta la fragmentación del lote para el mismo producto dentro del mismo ciclo de producción, los tiempos de alistamiento y la secuencia de fabricación de $n$ elementos. Conociendo que este problema es de tipo NP-hard, se da la necesidad de implementar métodos heurísticos, metaheurísticos, híbridos, entre otros, que permitan dar la mejor solución al problema. En la mayoría de casos, los esfuerzos de investigación han llegado a determinar la generación de un tiempo repetitivo óptimo, para que de esta manera se permita la minimización de los costos contribuyendo al mejoramiento continuo y la competitividad de las organizaciones.

Se propone para este estudio, un algoritmo genético que ayude a dar una solución aproximada al problema, dado a su aplicabilidad para la búsqueda eficiente, y su capacidad de trabajar con diferentes elementos de manera simultánea, permitiendo tener un conjunto de soluciones cada vez más satisfactorias.

# 2. Objetivos

## 2.1. Objetivo General

Plantear un algoritmo de solución para el problema de programación de lotes económicos (ESLP), con tiempos de alistamiento dependientes de la secuencia y generación de sublotes en un mismo ciclo de producción, satisfaciendo la demanda en el entorno productivo de una sola máquina basada en la metaheurística algoritmo genético.

## 2.2. Objetivos Específicos

✓ Realizar la revisión de literatura sobre Programación de Lotes Económicos (ELSP) y sus variantes con el propósito de identificar la evolución y la aplicabilidad en un entorno real de producción.

✓ Formular un modelo matemático para la solución del ELSP con tiempos de alistamiento dependientes de la secuencia y generación de sublotes dentro del mismo ciclo de producción.

✓ Desarrollar un algoritmo genético para la solución del problema de ELSP con tiempos de alistamiento dependientes de la secuencia y generación de sublotes dentro del mismo ciclo de producción y programarlo en el lenguaje Matlab.

✓ Evaluar el desempeño del algoritmo genético, analizando los datos resultantes y compararlo con instancias de la literatura.

✓ Elaborar un artículo académico de carácter publicable que muestre los resultados del trabajo de investigación realizado.

## 3. Revisión de la Literatura

El Problema de Programación de Lotes Económicos (ELSP) por sus siglas en inglés, está asociado con la configuración de diferentes artículos en una sola máquina, producidos a través de lotes, los cuales deben secuenciarse de tal forma que los tiempos y costos totales sean mínimos. Para efectos de este estudio se presenta el problema con la característica de fraccionar el lote tomando en cuenta el tiempo de alistamiento, y que éste dependa de la secuencia en la programación.

A continuación, se presenta el análisis de la literatura encontrada a partir de la ecuación de búsqueda con palabras clave del problema en la base de datos de la Universidad Industrial de Santander y de esta manera justificar el interés en la investigación del problema. Tomando los diferentes enfoques de solución en los últimos diez años, así mismo de los artículos encontrados se mencionan los que se asocian con las características propuestas en el presente estudio de investigación.

En el artículo de Hernández, Hernández y Flores (2011), los autores trabajan el Problema de Programación de Lotes Económicos (ELSP) por sus siglas en inglés, con el enfoque del ciclo básico planteado por Bomberger (1966), para el cual existen diferentes propuestas de solución dadas por algoritmos genéticos y diferentes metaheurísticas , para la investigación que aborda estos autores utilizan el método recocido simulado, el cual consiste en la forma de obtener un espacio de búsqueda más restringido de las variables y una estrategia para dar control a la exploración del espacio de soluciones que realiza el algoritmo, de tal forma que se lleve a cabo una búsqueda eficiente a la solución óptima para el problema.

La investigación realizada por Zanoni, Sogerstedt, Tang y Mazzoldi (2011), se estudia el ELSP con productos múltiples de fabricación y la oportunidad de reutilizar las piezas gastadas dándoles una segunda vida útil en la misma línea de producción, cumpliendo con los estándares calidad. En el año 2006 Tang y Teunter, estudiaron por primera vez el lote económico con las características mencionadas anteriormente y presentaron un algoritmo complejo para su solución óptima. Tres años más adelante los autores Tang y Teunter con el refuerzo de un nuevo autor propusieron en el 2009 otras heurísticas para tratar el problema, utilizando un enfoque computacional más eficiente que el anterior. Sin embargo, ambos estudios limitaron la atención al método del ciclo común, suponiendo que se utilizaba un solo lote de fabricación para cada artículo en cada ciclo. Basados en esto Zanoni, Sogerstedt, Tang y Mazzoldi (2011) proponen un algoritmo simple para resolver el problema, utilizando el método del periodo básico, mostrando así la funcionalidad del método y el ahorro de los costos totales de producción.

En el artículo de Hochung y Chan (2012), los autores abordan el ELSP basados en los estudios realizados por los diferentes investigadores por más de 50 años, en donde se indaga que al ser conocido este problema como tipo NP-hard, se desarrollan diferentes heurísticas para su solución. En dichas heurísticas los investigadores adaptaban generalmente dos métodos de redondeo para definir la frecuencia de producción, dichos métodos son el enfoque de número entero y la potencia de dos más cercanos. La frecuencia en la que se fabrican los productos define la cantidad de veces que se produce dicho producto en un ciclo determinado, por ello, si se obtiene una frecuencia de producción diferente, conduce a resultados distintos de optimización incluyendo el conjunto de productos. Por esta razón los autores proponen un algoritmo genético de dos niveles que ayude a tratar el problema del ELSP, llevando a cabo la utilización de ejemplos numéricos encontrados en la literatura y conjuntos de problemas generados aleatoriamente que prueben el rendimiento del nuevo enfoque. En el estudio se muestra una comparación del enfoque propuesto por los autores con los enfoques ya existentes y se muestra la efectividad del método empleado y la importancia que tiene la frecuencia de la producción para obtener óptimos resultados.

El artículo de Mohammadi, Nurmaya y Bahreininejad (2014), está enfocado en la optimización del ELSP, tomando en consideración la secuencia de fabricación de los productos y su vida útil, empleando métodos metaheurísticos para su solución, abordando el problema desde la perspectiva de producir múltiples artículos en una sola máquina en un patrón de ciclo, asumiendo que cada elemento puede producirse más de una vez en cada ciclo productivo. Su función objetivo es determinar su tasa de producción y tiempo de ciclo, tomando en cuenta un programa viable para la familia de artículos, permitiendo minimizar los costos promedio a largo plazo. Para efectividad en los algoritmos se emplea el método Taguchi ajustando los parámetros para obtener éxito en sus posibilidades de solución.

El artículo de Yao y Huang (2014), presentan el ELSP con la particularidad de que utilizan elementos en deterioro y se basan en el enfoque del periodo básico, en el cual el problema se preocupa por el tamaño del lote y la decisión de programación de $n$ artículos, por lo cual los autores proponen un Algoritmo Genético Hibrido (HGA), ya que este se equipa con procedimientos de prueba de viabilidad y a su vez se obtiene una heurística de búsqueda binaria para dar solución eficiente al problema. Los autores utilizan Proc FT para probar la viabilidad del óptimo local en un conjunto de frecuencias obtenidas en el proceso evolutivo del algoritmo genético, cuando el óptimo local no se muestra factible emplean la heurística de búsqueda binaria propuesta para resolver el problema de manera satisfactoria con el mínimo valor objetivo, sus resultados numéricos demuestran que el algoritmo genético hibrido propuesto es un enfoque eficiente para la solución del ELSP con las características abordadas.

El artículo de investigación de la Universidad de los Andes realizado por Torres y Rojas (2015), abordan el problema con el enfoque del periodo básico propuesto por Bomberger (1966), mediante un nuevo algoritmo genético con la estrategia de selección ruleta, es decir el cromosoma binario no requiere la inclusión de una variable asociada con el periodo básico. Este enfoque de periodo básico se calcula para optimizar cada selección de las variables de decisión de enteros, para lo cual se crea un diseño experimental, en el que se modifican tres niveles de los parámetros de probabilidad de cruce y mutación, por lo que se concluye que el uso de una combinación de pequeñas probabilidades de mutación con probabilidades de cruce relativamente altas mejora el rendimiento del algoritmo significativamente. El algoritmo propuesto por los autores se prueba en 10 problemas clásicos de Bomberger y da como resultado mejores soluciones en términos de la función de evaluación de la condición física y el número de generaciones, al mismo tiempo se obtienen períodos básicos más confiables y precisos en todos los casos.

El artículo de Chatfield (2015), aborda un enfoque puro de búsqueda genética al problema, desarrollando un procedimiento de programación de lotes genéticos GLS (por sus siglas en inglés), en el cual el método combina la estructura de solución extendida con una perspectiva nueva en la programación de los artículos, por lo que se considera un número mayor de programas, en los que se indica la asignación de los artículos a periodos como parte de una estructura de solución y donde se mantiene la eficiencia y viabilidad de poder resolver el problema. Para ello el autor emplea la regla de secuencia simple y efectiva que genera programaciones “anidadas”, creando así una representación cromosómica binaria de formulación de un nuevo problema, utilizando el algoritmo genético para buscar soluciones efectivas del ELSP a un bajo costo.

Continuando con la línea de múltiples criterios y diferentes enfoques para solución del problema, se encuentran los autores Zohali, Naderi, Mohammadi y Roshanaei (2018), donde se hace referencia al problema de ELSP en el entorno de fabricación de taller de flujo híbrido en el cual hay un horizonte de planificación finito. Los autores abordan el problema con dos decisiones simultáneas en relación a la secuencia de fabricación de productos y la cantidad de producción, en lo cual el objetivo es minimizar el costo total. Para resolver el problema se desarrolla un modelo nuevo de programación no lineal de enteros mixtos (MINLP), el cual mejora el modelo existente en la literatura, posterior a esto presentan una técnica nueva de alineación que transforma los dos modelos de MINLP en modelos efectivos de programación lineal de enteros mixtos (MILP), a su vez desarrollan un algoritmo eficaz que híbrida el algoritmo de búsqueda local iterado con una función óptima aproximada, para lo cual realizan experimentos exhaustivos que comparan el rendimiento del modelo, comparando también el algoritmo propuesto con 4 algoritmos metaheurísticos ya existentes en la literatura. Finalmente, los estudios realizados logran demostrar que el algoritmo y la técnica propuesta mejora sustancialmente el rendimiento computacional de los modelos matemáticos y también de los algoritmos en la literatura en caso de gran tamaño del problema.

Teniendo en cuenta los artículos mencionados anteriormente se tiene que el ELSP ha estado bajo diferentes propuestas y métodos que ayudan a la gestión en las operaciones y se ha mantenido como referencia para un entorno real de producción en los últimos 10 años, por lo que influye para el presente estudio con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en un mismo ciclo de producción continuar con la línea de conseguir la mejor solución para determinadas instancias.

# 4. Marco Teórico

## 4.1. Problema de Programación del Lote Económico (ELSP)

Para la definición y enfoques del problema de programación de lotes económicos ELSP se toma en cuenta una revisión extensa de la literatura del ELSP realizada por Vidal Carreras (2006), donde se parte de la descripción hecha por Bomberger (1966), y se relaciona con la programación de la producción de lotes económicos de varios productos diferentes en una única máquina. Las características del sistema según Bomberger (1966) son las siguientes:

- Solo un producto puede producirse al mismo tiempo.
- Existen costos y tiempos de alistamiento constantes asociados a la producción de cada producto. Los costos y tiempos de alistamientos son dependientes solo del producto que se va a fabricar, es decir que son independientes de la secuencia de fabricación.
- El ratio de la demanda de cada producto es conocido y constante durante el horizonte de producción.
- Para cada producto el costo variable total es la suma del costo de alistamiento y del costo de almacenamiento de inventario.

En el Problema de Programación de Lotes Económicos (ELSP) clásico se lleva a cabo una notación que se muestra en la Tabla 2, la cual se emplea en el desarrollo de los diferentes enfoques de solución.

Tabla 2.

*Parámetro y Variables de Decisión ELSP. Adaptado por Vidal Carreras (2006)*

| Símbolo | Definición |
|---|---|

| | |
|---|---|
| $\boldsymbol{i}$ | Índice de ítems $i = 1 \ldots\ldots n$ |
| $\boldsymbol{d_i}$ | Demanda del producto $i$ (unidades por unidades de tiempo) |
| $\boldsymbol{P_i}$ | Ratio de producción del producto $i$ (unidades por unidades de tiempo) |
| $\boldsymbol{A_i}$ | Costo de preparación de la máquina para la producción del producto $i$ (unidades de tiempo) |
| $\boldsymbol{h_i}$ | Costo de almacenamiento del producto $i$ (por unidad por unidades de tiempo) |
| $\boldsymbol{c_i}$ | Tiempo de cambio de partida del producto $i$ (unidades de tiempo) |
| $\boldsymbol{RO_i}$ | Ratio de cobertura del producto $i$ (unidades de tiempo) |
| $\boldsymbol{T_i}$ | Tiempo de ciclo para el producto $i$ (unidades de tiempo) |
| $\boldsymbol{T_i^e}$ | Tiempo de ciclo del modelo EOQ exponencial para el producto $i$ (unidades de tiempo) |
| $\boldsymbol{T^{cc}}$ | Tiempo de ciclo modelo ciclo común (unidades de tiempo) |
| $\boldsymbol{CT}$ | Costo total (unidades monetarias) |

## 4.2. Solución Independiente

Este procedimiento consiste en el cálculo del tamaño de lote de cada producto de manera aislada a partir de la fórmula del tamaño de lote económico (EOQ). Se denomina de esta manera ya que se ignora la casuística de varios productos que puedan ser producidos en una misma máquina, lo cual implica compartir la capacidad de la misma. De tal modo, el valor obtenido con este método es siempre una cota inferior en el cálculo del coste mínimo del problema. Esta solución puede ser raramente factible por el hecho de considerar cada producto de manera aislada. El planteamiento matemático para un producto $i$ es el siguiente:

La función de los costos totales viene dada por: $CT_i = \frac{1}{\mathrm{T}_i}\mathrm{A}_i + \mathrm{h}_i \frac{\mathrm{d}_i \mathrm{T}_i}{2}(1 - \frac{\mathrm{d}_i}{\mathrm{P}_i})$ donde los costos totales son función del costo de alistamiento y del costo de almacenamiento de los productos dentro del ciclo $T_i$. Para obtener el ciclo que minimiza los costos totales se deriva con respecto a $T_i$ la ecuación anterior y se obtiene lo siguiente:

$$T_i = \sqrt{2A_i/h_i r_i(1-b_i)}$$

(Vidal Carreras, 2006)

### 4.3. Método del Ciclo Común

Este método fue propuesto por Hanssmann (1962), y está basado en asumir una longitud suficiente para el ciclo en la cual se pueda acomodar la producción de cada producto exactamente una vez en cada ciclo, siendo el tamaño del ciclo de todos los productos iguales. También es conocido como el ciclo de rotación o método del tornillo. (Vidal Carreras, 2006)

### 4.4. Método del Periodo Básico

Este método fue propuesto por Bomberger (1966) y consiste en admitir diferentes ciclos para cada uno de los productos, pero insistiendo en que cada $T_i$, debe ser un múltiplo entero del periodo básico que se denomina “W” y es la longitud suficiente para acomodar la producción de todos los productos. Este método fue retomado por Dobson (1987), y Gallego y Mom (1962). En el método del Período Básico (BP), se permite variación de ciclos $T_I = n_i W$, donde $n_i$ es un multiplicador entero. (Vidal Carreras, 2006)

### 4.5. Método del Periodo Básico Extendido

Este método permite diferentes tamaños de lote para cualquier producto durante un programa cíclico, considerando explícitamente los tiempos de alistamiento, obteniendo siempre una solución factible, tal y como se demuestra en Dobson (1987), donde comúnmente da mejores soluciones que las anteriores expuestas. (Vidal Carreras, 2006)

### 4.6. Problema de Programación de Lotes Económicos, Solución Matemática

Analizando el problema de programación de lotes económicos y llegando a un caso concreto, puede estar representado mediante el siguiente grafico (Figura 1), para el cual se presenta la producción de 3 artículos, el primer artículo es producido durante un periodo por ejemplo $t_1$, construyendo de esta manera un inventario el cual es utilizado para satisfacer la demanda el resto del periodo. Posteriormente se produce el segundo artículo que está en amarillo y se consume mientras se está produciendo y acumulando el inventario, el cual es utilizado para suplir la demanda el resto del ciclo. Seguidamente se produce el tercer elemento señalado en azul hasta el punto de alcance, no es necesario que los picos sean iguales, solo son indicativos. Así que, de esta manera en la producción del tercer artículo, se consume el inventario acumulado, y se regresa al primer artículo. A lo anterior expuesto se le denomina ciclo, este ciclo es T mayúscula y en el problema de la programación de lotes económicos se asume que el ciclo es el mismo para los 3 elementos, tal como lo muestra el gráfico.

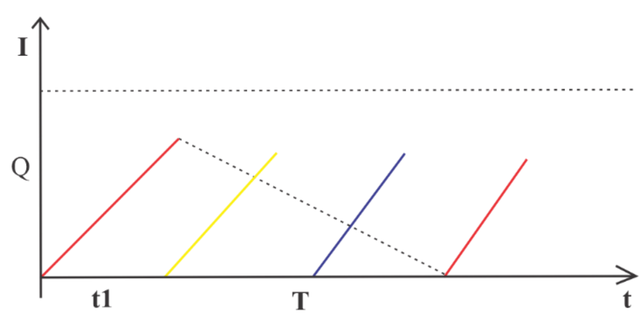



*Figura 1*. ELSP en la producción de 3 artículos durante el mismo ciclo T.

Tomando en cuenta los parámetros del problema se tiene que:

$n$: Cantidad de productos $j$

$D_j$: Demanda del producto $j$

$Q_j$: Cantidad a producir del producto $j$

$Co_j$: Costo de cambio para el producto$j$

$C_j$: Costo de mantener en inventario el producto $j$

Para el costo del artículo $j$ se tiene la siguiente expresión:

$$TC_j = \frac{D_j}{Q_j}Co_j + \frac{1}{2}Im_jCc_j \qquad (1)$$

De acuerdo con la ecuación (1) el costo total es la suma de los costos asociados con los 3 artículos. En general cuando se tienen $n$ artículos, para cada artículo hay un costo de pedido y un costo de configuración. En primera medida se toma en cuenta que el costo de configuración es el equivalente al costo de pedido, y el costo de pedido es cuando se compra el artículo desde el exterior, mientras que el costo de configuración se incurre cuando se produce. Seguidamente se puede observar que en la figura1 se da una pequeña brecha entre el final de la producción del siguiente artículo, siendo esta pequeña brecha el tiempo para el cambio o la configuración para hacer el siguiente artículo. Para lo mencionado anteriormente hay un costo asociado con esa configuración. Por lo tanto, si se toma un artículo en particular, la cantidad de veces que se haría ese artículo es $\frac{D_j}{Q_j}$, para lo cual la expresión usual es $\frac{D_j}{Q_j}Co_j$. Donde, $Co$ para el artículo $j$ es el costo de cambio, $D_j$ es la demanda anual, $Q_j$ es la cantidad de pedido o la cantidad de producción para ese artículo, se tiene también que $Im$ es igual a $Q_j(1 - \frac{D_j}{P_j})$ , de esta manera quedaría la siguiente expresión:

$$TC_j = \frac{D_j}{Q_j} Co_j + \frac{1}{2} Q_j (1 - \frac{D_j}{P_j})\, Cc_j \qquad (2)$$

Ahora, una suposición importante en el problema de lotes económicos es que los tiempos del ciclo $T$ coinciden. Por lo tanto, el tiempo de ciclo para el segundo elemento también sería el mismo $T$ de tiempo de ciclo para el tercer elemento. Por lo cual, como los tiempos de ciclo son los mismos, la cantidad de producción $Q$ para el artículo $j$, sería igual a $Q_j = D_j T$ , donde $D_j$ es la demanda anual y $T$ es el tiempo de ciclo. Entonces, teniendo la expresión anterior se sustituye en (2), para obtener $TC$ y $Cc_j$ se escribe como $C_j$, esto sería igual a:

$$TC_j = \frac{1}{T} Co_j + \frac{1}{2} C_j D_j T (1 - \frac{D_j}{P_j}) \qquad (3)$$

La expresión (3) sería el costo de un artículo en particular suponiendo que el tiempo del ciclo es $T$, que es la variable que se quiere hallar. Ahora el costo total será para todos los artículos, por lo cual, se resume y se tiene que:

$$TC = \sum_{j}^{n} \left( \frac{1}{T} Co_j + \frac{1}{2} C_j D_j T (1 - \frac{D_j}{P_j}) \right) \qquad (4)$$

Ahora, la ecuación (4) es la función que se quiere minimizar y descubrir la $T$ que es común para todos los elementos. Además, en este punto se tiene también una restricción. La restricción se da de tal modo que, si se toma el comienzo del primer ciclo del artículo 1 y el comienzo del segundo ciclo del artículo 1, la duración del tiempo es $T$, que es lo mismo que el comienzo del artículo 2 en el primer ciclo y el comienzo del segundo ciclo del artículo 2. Ahora, entre este tiempo $T$, lo que se debe hacer es producir este artículo más el siguiente, para ello se tiene en cuenta el cambio de rojo a amarillo, que es el artículo 1 al artículo 2 y del amarillo al azul que es el artículo 2 al artículo 3 y el azul al rojo, que es el artículo 3 al artículo 1. Por lo tanto, dentro de este período de tiempo $T$ se debe tener la producción de los 3 artículos creados, así como el tiempo de configuración para que los 3 artículos sean acomodados. Así que se llamará como el tiempo de configuración para hacer el artículo j. Se debe tener en cuenta que estos tres tiempos no tienen que ser iguales, en segundo lugar, se asume que el tiempo de configuración, es el tiempo necesario para hacer el respectivo artículo y no debe depender del otro. Por ejemplo, si pasamos de rojo a amarillo o si tiene otra situación que se pasa de azul a amarillo, el tiempo que se tarda en configurar para el elemento amarillo que es el número de elemento 2 será el mismo. De esta forma se llamará $K_j$ al tiempo de configuración para producir o configurar la producción del artículo $j$. Así que se produce hasta un cierto período, siendo $P_j$ la tasa de producción de este ciclo, y $t_j$ el tiempo de producción del artículo $j$ expresándose de esta manera $P_j t_j$. De esta forma se tiene que lo que se produce en un ciclo debe ser igual a lo que se consume en ese mismo ciclo, es decir que $P_j t_j = D_j T$. Dicho esto, se tiene que el tiempo de ciclo para el artículo $j$, es la suma de $K_j$ y $P_j t_j$. Despejando $t_j$ para dejar en términos del tiempo de ciclo obtenemos la siguiente restricción:

$$\sum K_j + \sum \frac{D_j}{P_j} T \leq T \qquad (5)$$

Ahora, esta restricción debe ser reescrita ya que hay un término $T$ que viene dado. Por lo tanto, despejando la ecuación se tiene que:

$$\frac{\sum K_j}{1 - \sum \frac{D_j}{P_j}} \leq T \qquad (6)$$

Entonces la expresión (6) es la forma en que el limitador está en la función objetivo. Consecuente a esto, se debe resolver el problema de optimización restringida para obtener el valor de $T$. El método más simple para hallar el valor de $T$ es en realidad resolver la versión sin restricciones del problema. Siempre se puede relajar un problema eliminando un cierto conjunto de restricciones, y si la solución satisface la restricción, entonces es óptimo para el problema de la restricción. Por tanto, resolviendo la versión no restringida de este problema, que consiste en diferenciarlo de inmediato con respecto a $T$ y establecerlo en 0 se obtiene la siguiente expresión:

$$\frac{-\sum Co_j}{T^2} + \sum \frac{1}{2}\left[C_j D_j \left(1 - \frac{D_j}{P_j}\right)\right] = 0 \qquad (7)$$

Resolviendo de (7) obtenemos:

$$\frac{\sum Co_j}{T^2} = \sum \frac{1}{2}\left[C_j D_j \left(1 - \frac{D_j}{P_j}\right)\right] \qquad (8)$$

Despejando $T$ de la ecuación (8) obtenemos que:

$$T = \sqrt{\frac{\sum Co_j}{\sum \frac{1}{2}\left[C_j D_j \left(1 - \frac{D_j}{P_j}\right)\right]}} \qquad (9)$$

Posteriormente hay que comprobar si el valor de (9) satisface la restricción definida en particular. Si satisface, quiere decir que este costo es óptimo. Pero si esto no satisface esta restricción, se concluye una vez más la naturaleza misma de la función, que se realizará como una ecuación en el óptimo y luego se pondrá en práctica a través de un ejemplo numérico. (Srinvasan, 2012)

## 4.7. Métodos de Resolución para el Problema ELSP

El problema de programación de lotes económicos (ELSP), puede resolverse mediante métodos analíticos que consiguen el óptimo de una versión limitada del problema original y por medio de heurísticas. Hsu (1983), muestra que el problema es de tipo NP-hard y en su estudio se da un procedimiento de enumeración implicita para testear la factibilidad. Dado que las soluciones exactas solo se pueden alcanzar para pequeñas instancias, y debido a la magnitud del problema del ELSP se clasifica como tipo NP- hard. En la literatura se ha encontrado estudios que se dedican a la complejidad del problema, por lo cual es importante establecer la importancia de la teoria de complejidad computacional en la solución del ELSP. (Vidal Carreras, 2006)

## 4.8. Teoría de la Complejidad Computacional

La teoría de la complejidad computacional se centra en la clasificación de los problemas computacionales de acuerdo con su dificultad inherente y la relación entre dichas clases de complejidad, para ello se toma en consideración dos tipos de recursos requeridos durante el cómputo para resolver un problema:

- **Tiempo:** Número de pasos base de ejecución de un algoritmo para resolver un problema.
- **Espacio:** Cantidad de memoria utilizada para resolver un problema.

La teoría estudia la eficiencia de los algoritmos estableciendo su efectividad de acuerdo con los dos recursos mencionados anteriormente, ayudando a evaluar la viabilidad de la implementación práctica en tiempo y costo. La mayor parte de los problemas de teoría complejidad son relacionados con problemas de decisión, que corresponde a dar una respuesta positiva o negativa de un problema dado. Para este tipo de problemas el lenguaje equivalente es el conjunto de entradas para el cual la respuesta es positiva. Los problemas de decisión se clasifican en conjuntos de complejidad comparables llamados clases de complejidad. (Sanchis, Ledezma Espino, Iglesias Martinez, García Jimenez, & Alonso Weber )

**4.8.1. Clases de Complejidad.** Considerando la teoría de la complejidad computacional un problema de decisión podrá ser; P, NP, NP – hard y NP – completo. Sin embargo, para esta distinción es necesario considerar el modelo teórico de las máquinas de Turing.

**Clase P:** Contiene todos los problemas de decisión para los que existe un algoritmo de la máquina de Turing, los cuales tienen un tiempo polinómico y conducen a una respuesta afirmativa o negativa en una serie de pasos. Este tipo de problemas son triviales.

**Clase NP:** Contiene todos los problemas de tipo no – polinomial. Los cuales se resuelven mediante tiempos exponenciales, por lo cual no es fácil hallar una respuesta inmediata, y se recomienda solucionar en la máquina de Turing.

**Clase NP- hard:** Son problemas que se resuelven en un tiempo polinomial no determinista. Estos problemas pueden estar dados por problemas de decisión, problemas de optimización o problemas de búsqueda.

**Clase NP – completo:** Son aquellos problemas que son NP y NP- hard al mismo tiempo.(Academic, 2010)

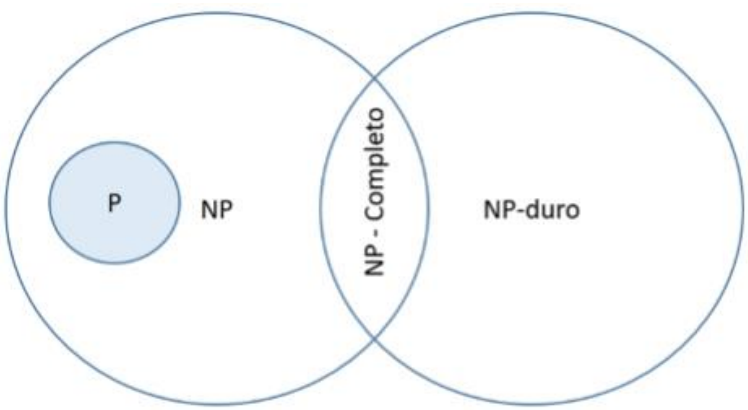



*Figura 2.* Identificación de los problemas P vs NP. Adaptado de Maldonado (2013)

El problema de programación de lotes económicos (ELSP), con las características ya mencionadas en el presente estudio, es clasificado desde la perspectiva computacional como tipo

NP – hard, debido a que debe darse aproximaciones del problema para hallar su factibilidad. (Hsu, 1983).

## 4.9. Algoritmos Genéticos

Los algoritmos genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización y se basan en el proceso genético de organismos vivos. A lo largo de todas las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural según la teoría evolutiva de Darwin. Basado en este proceso, los algoritmos genéticos son capaces de crear soluciones para problemas reales. Los principios básicos de los algoritmos genéticos fueron establecidos por Holland (1975), y se encuentran descritos en diferentes textos. Estos algoritmos utilizan una estructura simple de datos denominada cromosoma que representa las soluciones posibles en un problema específico y tiene aplicabilidad a esas estructuras en diferentes operadores y combinaciones de ellos de tal manera que la información importante se preserve. Los algoritmos genéticos se asocian generalmente a funciones de optimización, sus elementos básicos según la literatura se definen como:

**Población:** Se define como un conjunto de individuos que representan posibles soluciones a un problema, estos individuos se denominan como cadenas de bits que se evalúan después de codificarse en números enteros o reales que representan las variables en el problema. De forma general, la población inicial suele conformarse de manera aleatoria. Mediante un proceso de selección natural que se aplica sobre la población inicial y con el uso de operadores genéticos tales como el cruzamiento y la mutación, se van generando así los descendientes que constituirán una nueva generación.

**Gen o Cromosoma*:*** Se conoce también como genotipo, y es el individuo o elemento de la población que representa la posible solución al problema.

**Función Fitness*:*** También conocida como función aptitud, siendo una expresión matemática para evaluar la calidad del grupo de individuos en una generación. La clave a la hora de definir una función aptitud es que ésta debe regresar los valores más altos cuando se aplica a los individuos que más se aproximen a la solución óptima.

**Selección Natural de padres*:*** Es un mecanismo de selección aplicado sobre una población o generación en forma probabilística de acuerdo con el valor de la función de aptitud de cada individuo. De acuerdo con esta función los individuos que estén mejor calificados tienen mayor oportunidad de ser seleccionados como padres para producir la generación siguiente.

**Operadores Genéticos:** Son operadores que permiten obtener una generación nueva a partir de una población ya existente. Los más comunes son el "crossover" que son los de cruce o combinación genética y el operador mutación. **Crossover:** Mencionado anteriormente es un cruce mediante el cual dos individuos se aparean para la producción de descendencias individuales, esto se logra con el intercambio de los segmentos de los cromosomas de los padres. Para ellos se han propuesto diferentes modelos como el punto simple, el punto múltiple y el cruzamiento uniforme. **Mutación:** Es el mecanismo necesario para el aseguramiento de la diversidad de la población. Se selecciona de forma aleatoria un individuo para que sufra la mutación, el algoritmo de igual forma cambia el bit, teniendo como objetivo evitar un modelo fijo de soluciones que haya sido propagado mediante generaciones diferentes. (Moujahid, Inza , & Larrañaga, 2010).

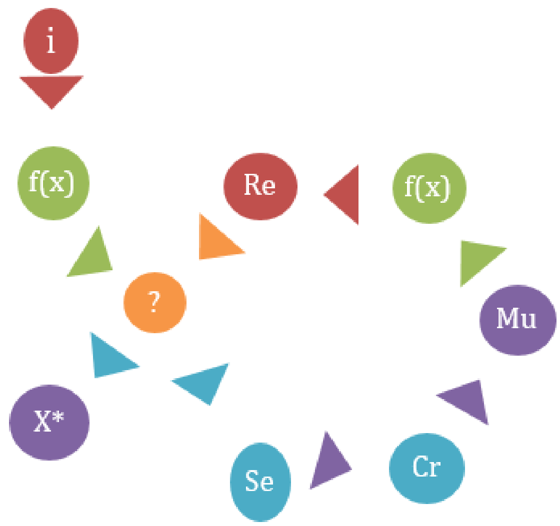



Figura 3. *Algoritmo Genético*

La figura anterior representa el proceso que realiza el algoritmo genético dado por **(i):** inicialización, **f(x):** evaluación **?:** condición de término, **Se:** selección, **Cr:** cruzamiento, **Mu:** mutación, **Re:** reemplazo, **X*:** mejor solución.

**4.9.1. Codificación de las variables.** Los algoritmos genéticos requieren que los individuos o en su defecto las posibles soluciones del problema se codifiquen en un cromosoma, donde cada cromosoma contiene varios genes, que son correspondientes a los parámetros que posea el problema. Para poder trabajar con estos genes en el ordenador es necesario que sean codificados en cadena, es decir, una ristra de símbolos que generalmente estará compuesta de 0s y 1s, no obstante, también se puede realizar la codificación mediante números enteros e incluso cadenas de palabras.

La elección de la codificación dependerá también del problema a resolver, ya que puede darse la situación en la que la resolución de un caso sea mejor el uso de una codificación basada en números reales mientras que esa codificación complique la solución en otro caso. Así pues, hay que estudiar la codificación que se mejor según el caso que se esté estudiando.

**Codificación Binaria:** Es la codificación más extendida debido a que los primeros algoritmos genéticos utilizaron este tipo de codificación. En este caso, cada cromosoma es una cadena de bits (0 o 1). A su favor tiene que puede abarcar muchos cromosomas incluso con un número reducido de genes. Sin embargo, esta opción no es la idónea para muchos problemas y en algunas ocasiones es necesario realizar correcciones después de la reproducción y/o mutación.

**Codificación Numérica:** En este tipo de codificación se utilizan cadenas de números que representan un número en una secuencia. Se utiliza en problemas en los que hay que ordenar algo, donde resulta muy útil. En algunos casos también es necesario como en el caso anterior realizar correcciones después de la reproducción y/o mutaciones.

**Codificación por Valor Directo:** Este tipo de codificación será el utilizado en caso de resolución de problemas en el que se requiera del uso de valores de cifrado complicado como podría ser en el uso de números reales, cuya codificación con números binarios sería muy complejo. En codificación por valor directo cada cromosoma es una cadena de valores relacionados con el problema a estudiar, pudiendo ser desde números decimales, cadenas de caracteres o incluso una combinación de varios de ellos. Su aplicación es muy buena en ciertos problemas concretos. Por el contrario, para la utilización de esta codificación, normalmente es necesario desarrollar nuevas técnicas de reproducción y mutación especificas hacia la resolución del problema.

**Codificación por Árbol:** Este tipo de codificación se utiliza principalmente en el desarrollo de programas o expresiones para programación genética. Cada cromosoma será en este caso un árbol con ciertos objetos. En este método, los cambios aleatorios pueden generarse cambiando el operador, alterando el valor de un cierto nodo del árbol o simplemente sustituyendo un subárbol por otro. (Arranz de la Peña & Parra Truyol, 2007)

**4.9.2. Algoritmo Genético Simple.** El algoritmo genético simple (SGA), es a lo que normalmente se le conoce como algoritmo genético, ya que es su primera versión. Las implementaciones que se le han hecho recientemente con mayor complejidad son conocidas como algoritmos evolutivos, de igual manera siguen manteniendo en la mayoría de los casos la estructura básica del algoritmo genético simple.

El objetivo del algoritmo genético es la búsqueda de parámetros que minimicen una función, y el primer paso para dar solución a un problema con SGA es identificar la función y sus variables objetivo de las cuales se desea encontrar los valores óptimos. Los algoritmos genéticos simples trabajan sobre cadenas codificadas de los valores objetivos, y seguidamente se define su codificación para una posible representación. A esta codificación se le conoce como genotipo ya definido anteriormente y a una instancia de tal codificación se denomina como cromosoma, este último se codifica en un conjunto de valores reales a los cuales se les llama fenotipo.

El genotipo debe representarse para el problema por una cadena de tamaño y en un alfabeto finito, las formas más comunes para esta representación son las cadenas binarias, de valores reales o codificación gris, siendo la última mencionada una codificación que también utiliza valores binarios, a su vez el genotipo puede segmentarse lógicamente para contener la codificación para más de una variable objetivo, y estas variables pueden codificarse de manera distinta, es decir que se pueden definir genotipos para algunas variables que utilicen cierta representación y para otras una representación totalmente diferente.

Una vez se define la función objetivo y la codificación para las posibles soluciones del genotipo se aplica el algoritmo cuyos pasos son los siguientes:

1. Iniciar la población P con N individuos al azar.
2. While, define el máximo de generaciones.
3. Evaluar la aptitud de los individuos de P.
4. Aplicar selección sobre P.
5. Aplicar crossover.
6. Aplicar mutación.
7. Generación + = 1.
8. Fin.

Iniciar la población quiere decir que se crean un conjunto de soluciones iniciales, estas soluciones están definidas por el programador. Un punto a decidir es el número N de individuos en la población y este valor se mantiene constante durante toda la ejecución y es importante que sea asignado a través de la experimentación preliminar para poder observar el comportamiento del algoritmo, y ajustar si el tamaño de la población afecta negativamente el tiempo de ejecución o la amplitud de búsqueda.

Una vez se da entrada a la población inicial, comienza el ciclo generacional que producirá mejores soluciones en cada iteración realizada. Este ciclo generacional consiste en aplicar diferentes procedimientos sobre la población actual, y estos operadores son llamados operadores genéticos, que son el equivalente a los procesos naturales de selección, reproducción y mutación. El ciclo generacional se repite mientras la condición de terminación no se haya cumplido, una condición común es que el número de iteraciones o generaciones lleguen a alcanzar un valor determinado.

El primer paso del ciclo generacional es evaluar la aptitud de las soluciones presentes en la población actual, lo cual consiste en aplicar la función objetivo sobre cada uno de los valores que se decodifican de la población. El siguiente paso es seleccionar los individuos para el apareamiento, este paso está relacionado con los principios de selección natural, dónde la idea es escoger aquellos individuos que permitan el acercamiento a la solución óptima. Para ello hay diferentes formas de hacer la selección y de la manera que se haga en este paso afectará de manera simbólica el comportamiento del algoritmo genético.

Los individuos que son seleccionados forman el conjunto de individuos que serán reproducidos, el operador de reproducción es llamado crossover, la idea de este operador es utilizar información de los padres para reproducir hijos, el operador es ciego a las propiedades de la cadena decodificada, dado a que trabaja solamente con la información codificada, y agregado a ello los procedimientos que generan los nuevos hijos también son estocásticos. Una vez que se generan los hijos se procede a aplicar el operador de mutación, este operador trabaja también sobre las cadenas codificadas. Su funcionamiento es simple, ya que introduce cambios aleatorios a los hijos generados en el paso anterior, el propósito de este operador es permitir que puedan explorarse regiones del espacio solución, diferentes a las que no sean posibles de llegar a través de la selección y reproducción, la tasa de mutación es un factor que define el usuario.

Por último, es necesario insertar la nueva descendencia a la población ya definida, hay que recordar que el número de individuos de la población establecida permanece estable, así que es necesario seleccionar los individuos que serán insertados y los que serán desechados, de igual manera que en las otras etapas del ciclo evolutivo, en esta etapa existen diferentes maneras de implementación.

En resumen de lo dicho anteriormente, se comienza con una población aleatoria, cuyos individuos contienen características que combinados pueden contribuir a encontrar mejores individuos, a través del tiempo el ciclo generacional explora el espacio de solución buscando individuos cada vez más cercanos a la mejor solución, siendo guiado por los mecanismos de selección, dando la combinación de características de los individuos existentes a través de la reproducción y dando la generación de individuos con características no existentes en la población actual a través de la mutación.

**4.9.2.1. *Diseño del Experimento.*** Como se mencionó anteriormente el algoritmo genético simple es la base los demás algoritmos genéticos avanzados, y para cada uno de los pasos del SGA existen diferentes métodos de los cuales puede escogerse el mejor para adaptarlo al problema que se esté trabajando en cuestión y de ser necesario también puede ser posible modificar en detalle cualquiera de los procedimientos para que se puedan explorar diferentes caminos de búsqueda.

**Generación de la población inicial**

La aproximación clásica para la generación de la población inicial es la generación al azar, en este tipo de inicialización son generados N cromosomas, en el caso de las representaciones con cadenas binarias, cada posición tiene la misma posibilidad de ser 0 o 1. Otra técnica que suele aplicarse es sembrar la población inicial con puntos que se creen cercanos a la solución óptima con el propósito de que se acelere la convergencia del algoritmo, esta técnica tiene las desventaja de que se aumente la posibilidad de que el algoritmo converja a un óptimo local en caso de que el óptimo global se encuentre lejano del punto sembrado.

**Selección de la Población para Reproducción**

Existen diferentes técnicas para la selección de individuos para la reproducción y la reinserción, a continuación, se menciona las que se usan con mayor frecuencia.

- **Selección por Ruleta:** en la literatura se describe como la versión estocástica de la supervivencia del más apto, el muestro estocástico con reemplazo o selección por ruleta es la técnica de selección que se usa con mayor frecuencia, ya que consiste en asignar un segmento de la ruleta a los individuos en base a la aptitud que tienen estos y a la aptitud total de la población actual, y se gira la ruleta tantas veces como se requiera. Para esto se tiene el siguiente procedimiento:

1. Se calcula el valor objetivo $f(x_i)$ para cada cromosoma $x_i$.
2. Se calcula el valor objetivo total para la población:

$$F = \sum_{n}^{i=1} f(x_i) \qquad i = 1, 2 \ldots\ldots n \; donde \; n \; es \; el \; tamaño \; de \; la \; población \qquad (10)$$

3. Se calcula la probabilidad de selección $P_i$ para cada cromosoma $x_i$:

$$P_i = \frac{f(x_i)}{F} \qquad i = 1{,}2 \ldots\ldots.. n \qquad (11)$$

4. Se calcula la probabilidad acumulativa $P_i$ para cada cromosoma $x_i$:

$$q_i = \sum_{i}^{l=1} p_l \qquad i = 1{,}2 \ldots\ldots.. n \qquad (12)$$

Después de la selección se hace lo siguiente $n$ repeticiones:

1. Se genera un número al azar $\rho$ en un rango [0,1].
2. Se escoge el i-ésimo cromosoma $x_i$ tal que $q_{i-1} < \rho \leq q_i$.

Si bien esta técnica es una de las que se usa comúnmente, no quiere decir que genere los mejores resultados, debido a que esta técnica tiene diversas dificultades, entre ellas una de las de mayor relevancia es el hecho de un cromosoma con segmento de tamaño > 0 pueda dominar las selecciones ya realizadas.

**Muestreo Estocástico Universal:** Es un algoritmo de muestreo que se implementa en una sola fase, y surge con el fin de corregir algunos problemas del algoritmo de muestro de ruleta que se explicó anteriormente, fue desarrollado por Baker en el año de 1987. Este algoritmo es simple y eficiente, dado un conjunto de $n$ individuos y sus valores objetivos asociados, el algoritmo los acomoda en ruleta donde el tamaño de los cortes asignados a cada individuo es proporcional al valor objetivo, tal y como se hace en el algoritmo de ruleta. Seguidamente una segunda ruleta, es marcada con $y$ marcadores espaciados entre sí igualmente, donde $y$ es el número de selecciones que se desean efectuar, y por último se gira la ruleta y se selecciona el individuo por cada marcador. Las posiciones de los marcadores indican los individuos seleccionados, si $l$ marcadores caen sobre el mismo individuo, este se selecciona $l$ veces, y esto garantiza que ningún individuo sea seleccionado ni más ni menos veces que las esperadas debido al corte del tamaño $v$. (Mariano, 2005).

**Selección por torneo:** Este método fue el utilizado para desarrollar el algoritmo del presente estudio, consiste en realizar la selección en base a comparaciones directas entre individuos. Hay dos maneras de selección mediante torneo, las cuales son de forma determinística y la probabilística. En la versión determinística se selecciona al azar un número $p$ de individuos, dónde generalmente se escoge $p = 2$, y dentro de los individuos seleccionados se elige el más apto para que pase a la siguiente generación. La versión probabilística la cual se utiliza en esta investigación, únicamente se diferencia en el paso de selección del ganador del torneo, y en lugar de escoger siempre el mejor, se genera un número aleatorio del intervalo [0,1], si es mayor que un parámetro $p$ que se fija en todo el proceso evolutivo, se escoge el individuo más apto y en caso de que sea contrario el menos apto, normalmente $p$ toma valores en el rango de $0{,}5 < p \leq 1$.

Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan varios individuos en cada torneo, la presión de selección es elevada y los peores individuos tienen pocas oportunidades de reproducirse. Mencionando un caso en particular es el elitismo global, el cual consiste en un torneo donde participan todos los individuos de la población por lo que la selección se vuelve determinística. Cuando el tamaño del torneo se reduce, la presión de selección también se disminuye y los peores individuos tienen más oportunidades de que sean seleccionados.

Elegir cualquiera de los métodos de selección determinará la estrategia de búsqueda del algoritmo genético. (Brad & Golberg, 1995)

**Reproducción**

La reproducción o crossover, es el operador genético principal, por el cual se generan nuevas soluciones a partir de las soluciones actuales, desplazándose en el espacio solución hacia el mejor que considere. Este opera sobre dos cromosomas cada vez y genera descendencia combinando algunas de las características de ambos cromosomas, el caso más simple crossover de un solo punto, escoge un punto de corte al azar en las dos cadenas de los cromosomas padres para formar dos subcadenas en cada una, una a la izquierda de un padre con la subcadena derecha del otro para formar una cadena hija. De igual manera se genera el hijo pero con las subcadenas restantes, dando un ejemplo si se tiene los cromosomas $c_1$ y $c_2$:

$$c_1 = [000|00000]$$

$$c_2 = [111|11111]$$

Y luego se genera un punto de corte como el marcado por el símbolo |, así que se generan dos hijos de la siguiente manera:

$$c_3 = [000|11111]$$

$$c_4 = [111|00000]$$

Ahora, se hace que $p_c$ sea llamado el factor de reproducción o crossover rate. Este se define en proporción, en base al $J$ tamaño de la población, del número de descendencia producida en cada generación. Este factor es el que controla el número esperado $p_c * J$ de cromosomas que participarán en un crossover. Dado un ejemplo, si se tuviera un $p_c = 0{,}25$ se espera que en promedio un 25% de los cromosomas participen en la reproducción. Si se tiene un conjunto de cromosomas seleccionados para reproducción de tamaño 10, se espera 2 o 3 cromosomas que se reproduzcan. El procedimiento es para cada cromosoma en la población seleccionada para la reproducción se genera un número $\rho$ en el rango [0,1], si $\rho < \rho_c$ ese cromosoma es seleccionado para participar en un cruzamiento.

**Multipoint Crossover**

Los operadores crossover cortan las cadenas padres en más de un punto, los puntos de corte se generan al azar y se mezcla la información de los padres intercalando la información de sus respectivas cadenas, y a este tipo de operador se le conoce como crossover multipunto. El procedimiento de un crossover multipunto de 2 puntos, es el siguiente, el símbolo | indica los puntos de corte.

$$Padre_1 = [000|00000|0000]$$

$$Padre_2 = [111|11111|1111]$$

Genera

$$Hijo_1 = [000|11111|0000]$$

$$Hijo_2 = [111|00000|1111]$$

Finalmente se tiene que el cruzamiento multipunto decrementa la efectividad de los algoritmos genéticos y este decremento aumenta mientras más puntos de corte sean utilizados. El crossover multipunto acerca al algoritmo genético a una búsqueda simple al azar, pero en algunos casos se pueden generar mejores resultados. (Mariano, 2005)

## 5. Formulación Matemática

El problema de programación de lotes económicos (ELSP) con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en un mismo ciclo de producción se define de la siguiente manera:

### 5.1. Parámetros

$i$: Índice del producto $i: 1, \ldots \ldots \ldots . n\ productos$

$j$: Índice del producto anterior $j: 1, \ldots \ldots \ldots n\ productos$

$P_i$: Tasa de producción para el producto $i$

$D_i$: Demanda para el producto $i$

$h_i$: Costo de mantenimiento de inventario para el producto $i$

C: Matriz de costos de alistamiento

$C_{ij}$: Costo de alistar el producto $i$ dependiente del producto anterior $j$

S: Matriz de tiempos de alistamiento

$S_{ij}$: Tiempo de alistar el producto $i$ dependiente del producto anterior $j$

## 5.2. Variables de Decisión

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada lote $i$

$a_i$: Partición del lote $i$ en $t$

En el problema se halla la duración del ciclo $T$, existiendo una secuencia de producción, donde el tamaño del lote puede fraccionarse en partes durante el mismo ciclo productivo quedando en notación como $a_i$, para que la secuencia de producción pueda llevarse a cabo en la duración del ciclo elegido, el ciclo puede repetirse indefinidamente satisfaciendo la demanda y minimizando los costos de inventario y alistamiento por unidad de tiempo. Ver Figura 4.

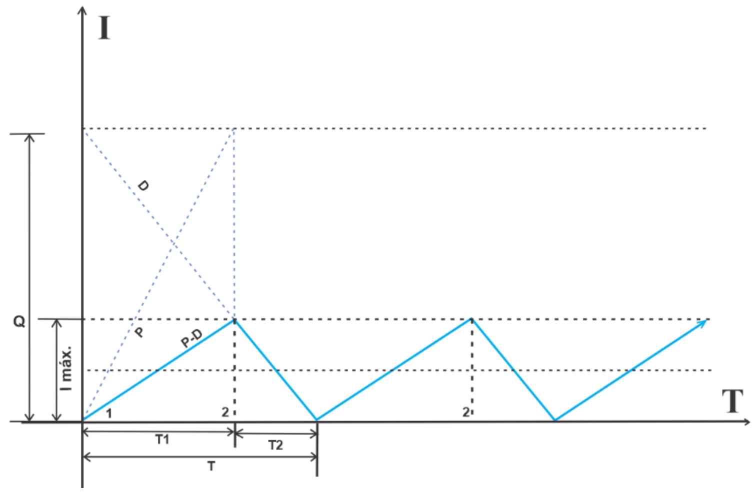



*Figura 4.* Inventario por producto sin faltantes.

El nivel de inventario más alto es $(P_i - D_i)$. El costo total del inventario para la producción tomando en cuenta la partición del lote es igual a $\frac{1}{2}\ h_i(P_i - D_i)({}^{D_i}/_{P_i a_i})(T)$

## 5.3. Ecuación del Costo Total de Producir el Lote $i$ con $n$ Productos y la Partición en $a_i$

Teniendo en cuenta la notación anterior el costo total para ELSP con tiempos de alistamiento dependientes de la secuencia y generación de sublotes en un mismo ciclo de producción se escribe como la sumatoria del producto $i$ hasta $n$, donde $n$ viene siendo la cantidad de productos considerados, el costo de alistamiento dado como $C_i$ que viene dado por una matriz $C_{ij}$ que representa el costo de alistar el producto $i$ dependiendo del producto anterior $j$ sobre el tiempo de ciclo más la suma de los costos de mantener inventario, teniendo en cuenta que $CT$ viene dado en función del tiempo es decir en qué periodo se generan los costos ($/t), por lo que para este caso se toman los costos totales anuales, obteniendo de esta forma la siguiente expresión:

$$CT = \sum_{i}^{n} \left( \frac{C_i}{T} + \frac{1}{2} h_i \frac{D_i T}{P_i a_i} (P_i - D_i) \right) \qquad (13)$$

### 5.4. Restricciones del Problema

Para el ELSP con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en el mismo ciclo de producción se tienen en cuenta las siguientes restricciones que deben cumplirse para que el problema tenga una solución factible.

$$\sum_{i}^{n} S_i + t_i^p \leq T \qquad (14)$$

$$P_i t_i^p \geq D_i T \qquad (15)$$

La restricción (14) hace relación al tiempo de ciclo el cual debe ser mayor a la sumatoria del tiempo de alistamiento, que viene dado por la matriz $S_{ij}$ que es el tiempo de alistar el producto $i$ dependiente del producto anterior $j$ y la sumatoria del tiempo de producción para cada producto $i$, la restricción (15) hace cumplir que la producción debe ser mayor a la demanda dependiente del tiempo de producción y de ciclo respectivamente.

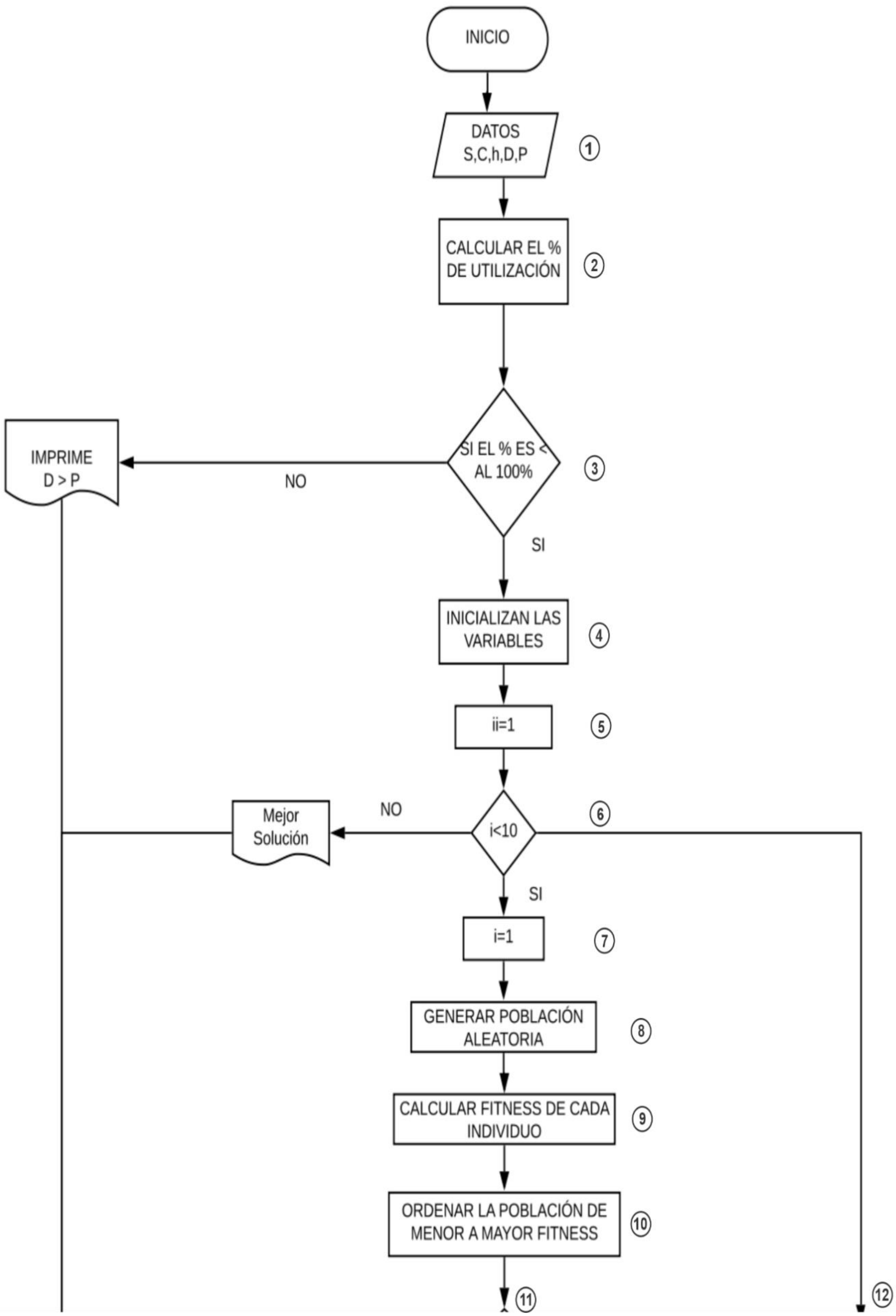
## 6. Diseño del Algoritmo

Para el problema de programación de lotes económicos (ELSP) con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en un mismo ciclo de producción se diseñó un algoritmo genético mediante el método de selección por torneo, que genera la mejor solución para el problema, su codificación se presenta en el apéndice A, al igual que los datos de prueba que están registrados en una hoja de cálculo de Excel llamada datos, anexa en el mismo apéndice.

**Procedimiento:** Algoritmo genético.

**Entrada:** Datos de entrada del problema, parámetros de los componentes genéticos.

**Salida:** Solución cercana al óptimo.


INICIO
DATOS
S,C,h,D,P
1
CALCULAR EL %
DE UTILIZACIÓN
2
IMPRIME
D > P
NO
SI EL % ES <
AL 100%
3
SI
INICIALIZAN LAS
VARIABLES
4
ii=1
5
Mejor
Solución
NO
i<10
6
SI
i=1
7
GENERAR POBLACIÓN
ALEATORIA
8
CALCULAR FITNESS DE CADA
INDIVIDUO
9
ORDENAR LA POBLACIÓN DE
MENOR A MAYOR FITNESS
10
11
12

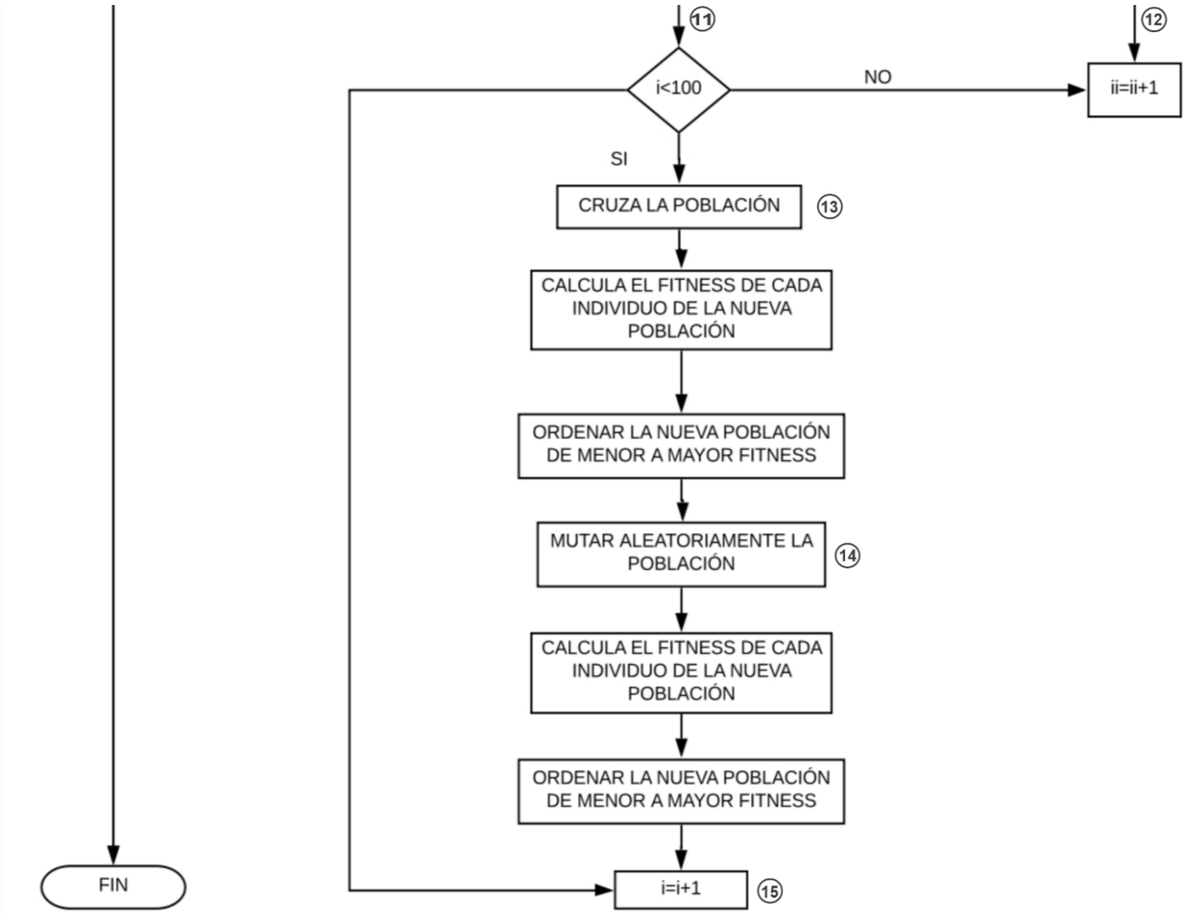



*Figura 5.* Diagrama de flujo del Algoritmo

Los pasos del diseño del algoritmo contenidos en la Figura 5 se describen a continuación:

Paso 1. Leer los datos de la hoja de cálculo Excel, en la cual se tiene Global (S C h D P): Son vectores con tamaño variable (h D P) dependiendo de la cantidad de productos y son valores constantes, h= costos de mantener el inventario ($ -unid/día), D= demanda diaria del producto (unid/día), P= capacidad de producción (unid/día). Son matrices (S C). Siendo S= tiempos de alistamiento (días), C= costos de alistamiento ($).

Paso 2. Se calcula el porcentaje de utilización de la máquina para cumplir la demanda. Se define una variable de control, a=0, seguidamente de if el cual hace referencia a una restricción de capacidad, la cual determina que, si la capacidad de producción es menor que la demanda, el programa termina de preguntar y se devuelve para cambiar los valores y que la restricción sea cumplida.

Paso 3. Por tanto, si el porcentaje de utilización es mayor al 100% no se puede calcular, imprime que la demanda es mayor que la producción se finaliza.

Paso 4. Se leen las variables globales que se van a utilizar en la función.

Paso 5. Se tiene las veces que se correrá el algoritmo.

Paso 6. Se tiene el contador de los números de torneos a realizar contenido por un for.

Paso 7. Entra e inicializa el contador de generaciones.

Paso 8. Genera la población aleatoria para cada uno de los 60 genes, tal y como se muestra en la Figura 6.

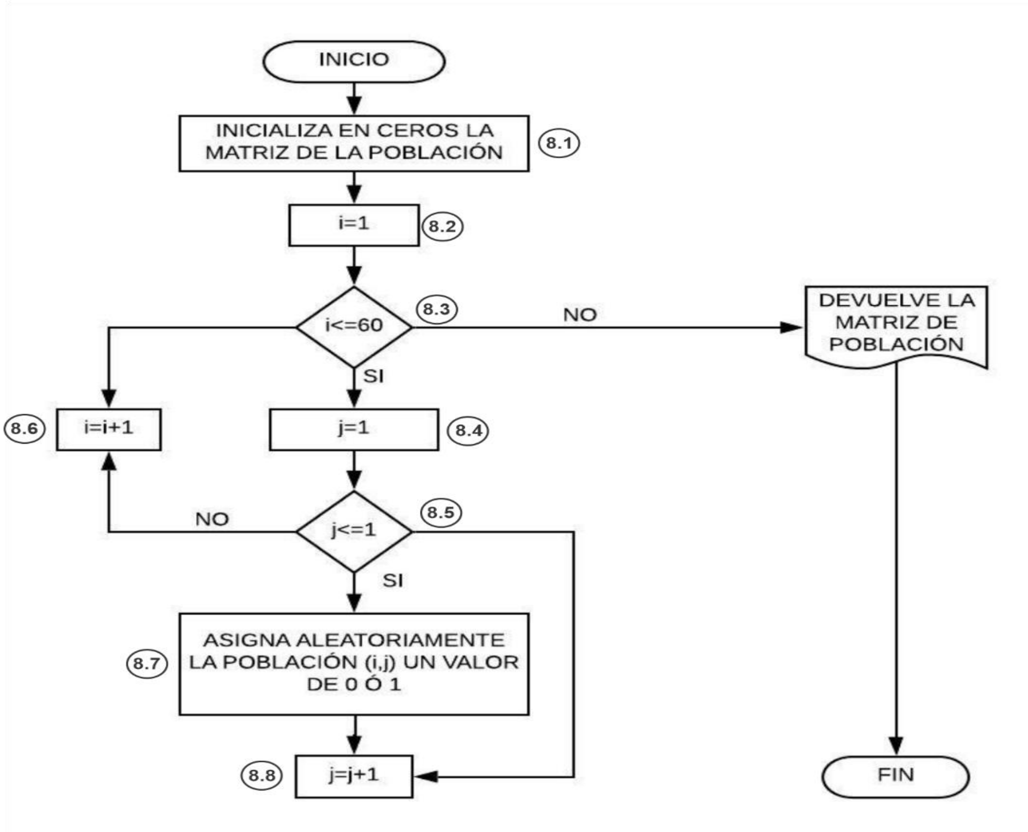



*Figura 6.* Diagrama de flujo de la función generar.

Paso 8.1. Se inicializa en ceros las matrices de la población de tamaño 60x11, donde 60 son los individuos de la población y 11 los individuos del cromosoma de solución.

Paso 8.2. Se tiene un contador para las filas es decir para los 60 individuos.

Paso 8.3. Entra e inicializa el contador de los 60 individuos, si no cumple la condición se devuelve a la matriz de la población y finaliza.

Paso 8.4. Se tiene el contador para las columnas es decir para los 11 genes.

Paso 8.5. Entra e inicializa el contador de los 11 genes.

Paso 8.6. Se le suma 1 al número de individuos de no cumplir con el for anterior.

Paso 8.7. Se le asigna aleatoriamente a la población de tamaño (i,j) un valor que va en un rango de (0 a 1).

Paso 8.8. Se le suma 1 al número de genes y finaliza la condición.

Paso 9. Se calcula la función del costo total. Su procedimiento se muestra en la Figura 7.

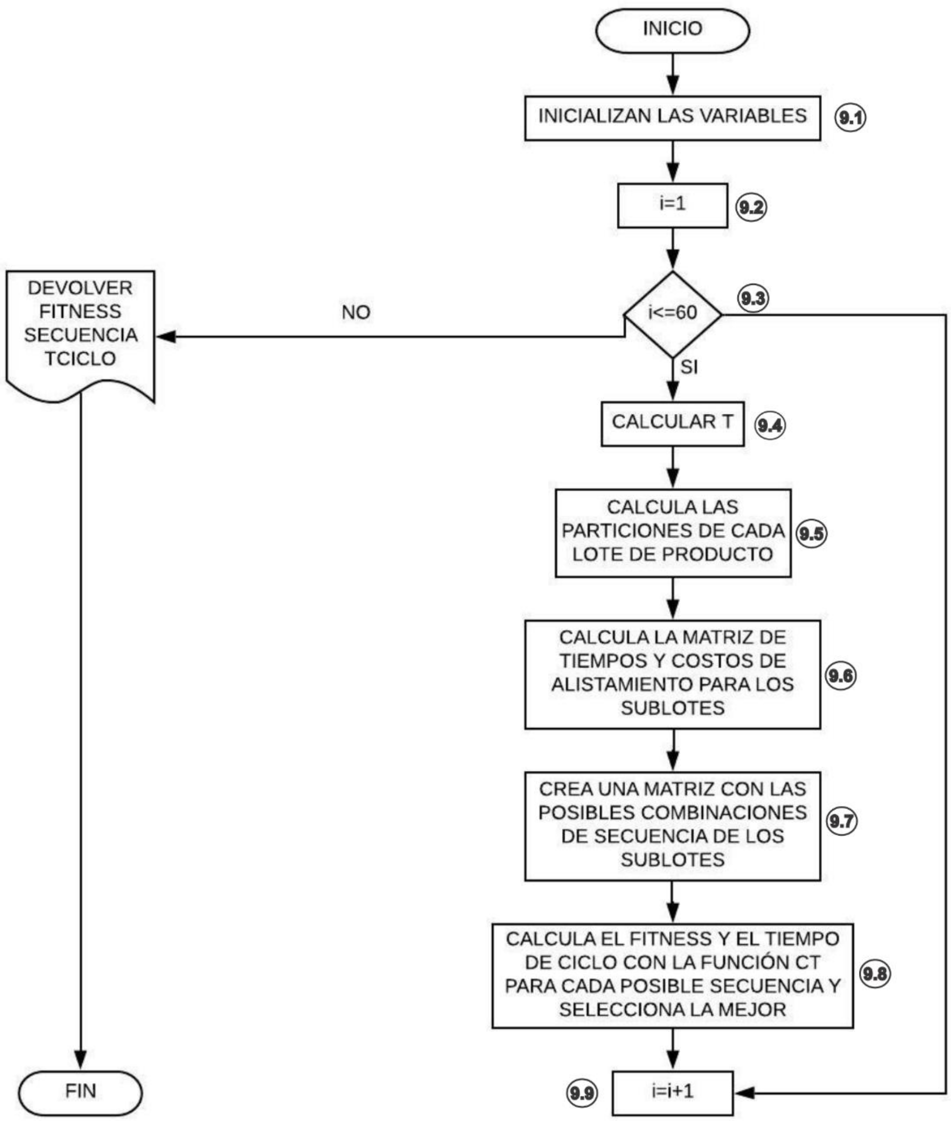



*Figura 7.* Diagrama de flujo de la función fit.

Paso 9.1. Se inicializan las variables globales mencionadas en el paso 1.

Paso 9.2 Se tiene el contador de genes.

Paso 9.3. Entra e inicializa el contador de genes, tomando en cuenta que si no cumple la condición imprime devolver, fitness, la secuencia y el tiempo de ciclo, de esta forma se finaliza.

Paso 9.4. Se calcula el tiempo de ciclo contenido en 8 genes del cromosoma de binario a decimal.

Paso 9.5. Se calcula las particiones de cada lote del producto y genera el siguiente cromosoma que se muestra en la Figura 8.

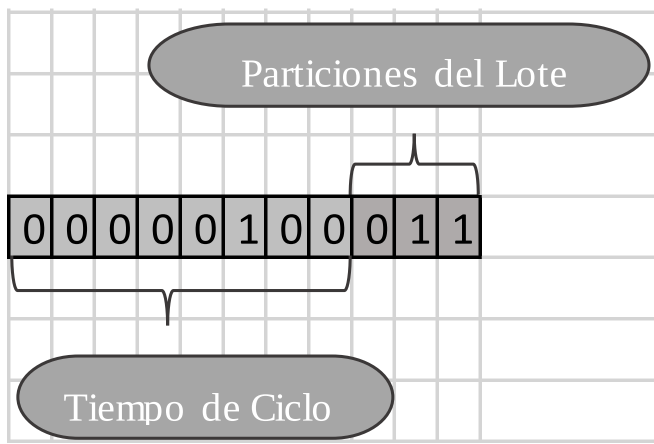



*Figura 8.* Cromosoma de mejor solución.

En la figura 8 se tiene el cromosoma que genera el tiempo de ciclo y las particiones del lote, teniendo que el tiempo de ciclo está dado en números binarios tal y como se explicó en el paso 9.4 y las particiones del lote están dadas por números binarios, teniendo que 1 hace relación a que el lote se parte y 0 a que no.

Paso 9.6. Se calcula la matriz de tiempos y costos de alistamiento para los sublotes.

Paso 9.7. Se crea una matriz que contiene las posibles combinaciones de secuencias de los sublotes es decir en qué orden con respecto a los lotes se van a producir.

Paso 9.8. Se calcula el fitness y el tiempo de ciclo con la función del costo total para cada posible secuencia y se selecciona la mejor.

Paso 9.9. Se le suma 1 a los genes y finaliza.

Paso 10. Se ordena la población de menor a mayor fitness y se guarda el mejor individuo, comparándolos cada vez que ordene una población. Su procedimiento se muestra en la Figura 9.

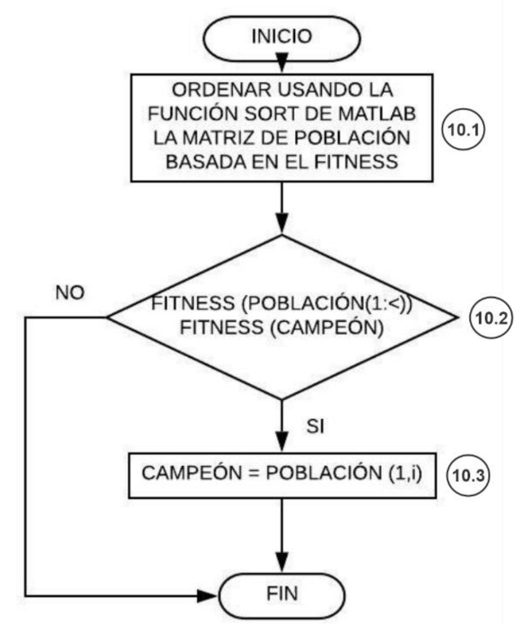



*Figura 9.* Diagrama de flujo de la función ordenar.

Paso 10.1. Se ordena la población usando la función SORT de Matlab de la matriz de población basada en el fitness ya calculado anteriormente.

Paso 10.2. Entra e inicializa el fitness teniendo en cuenta la población y los campeones del torneo, para lo que se asigna un número muy grande al mejor fit para que este cromosoma no sea tomado como ganador.

Paso 10.3. Se obtiene los campeones de la población de tamaño (1, i).

Paso 11. Se le suma 1 al número de torneos se finaliza el contador de torneos.

Paso 12. Se tiene el número de generaciones para cada torneo.

Paso 13. Se cruzan los individuos de forma aleatoria dando mayor probabilidad a los de mejor fitness es decir los de menor valor y dando menor probabilidad a los de menor fitness es decir a los de mayor valor. Su procedimiento se muestra en la Figura 10.

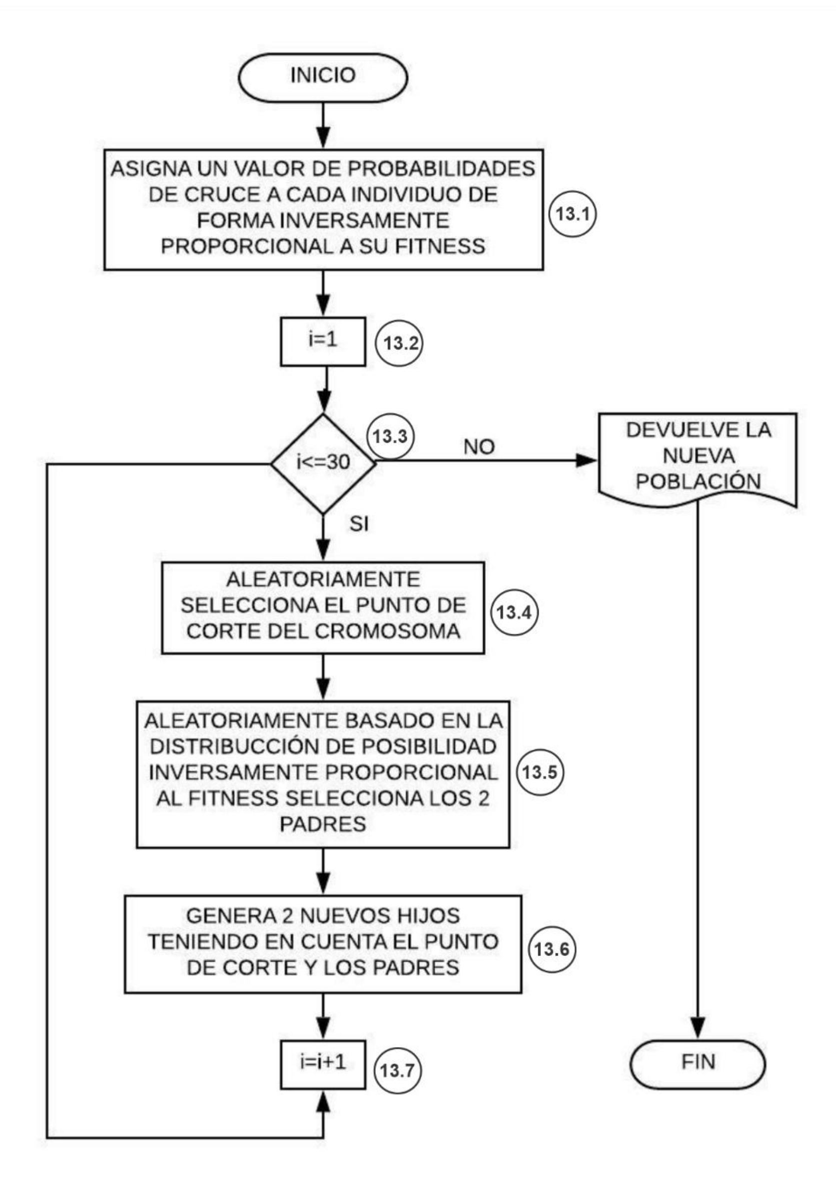



*Figura 10.* Diagrama de flujo de la función cruzar

Paso 13.1. Se le asigna un número de probabilidades de cruce para cada individuo de forma inversamente proporcional a su fitness.

Paso 13.2. Se tiene el contador para la población.

Paso 13.3 Entra e inicializa el contador de la población, donde se tienen 30 individuos ya que para cada uno creará dos nuevos dando así un total de 60 individuos, si no cumple la condición devuelve la nueva población y finaliza.

Paso 13.4. Aleatoriamente selecciona el punto de corte del cromosoma.

Paso 13.5. Aleatoriamente basado en la distribución de probabilidades inversamente proporcionales al fitness selecciona 2 padres para aparearse y crear los descendientes.

Paso 13.6. Genera 2 nuevos hijos teniendo en cuenta el corte anterior de los padres, tal y como se muestra en el ejemplo de la Figura 11.

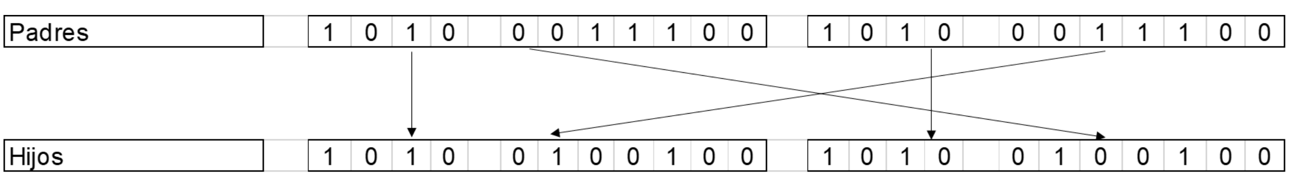



*Figura 11.* Operador de cruce basado en el orden.

Paso 13.7. Se le suma 1 al contador de población y finaliza.

Seguidamente calcula el nuevo fitness de cada individuo, y lo ordena tal y como se describió en los pasos 9 y 10.

Paso 14.  Se realiza la mutación de forma aleatoria con 5% de probabilidad de mutar uno de los genes de cada individuo. El proceso se describe en la Figura 12.

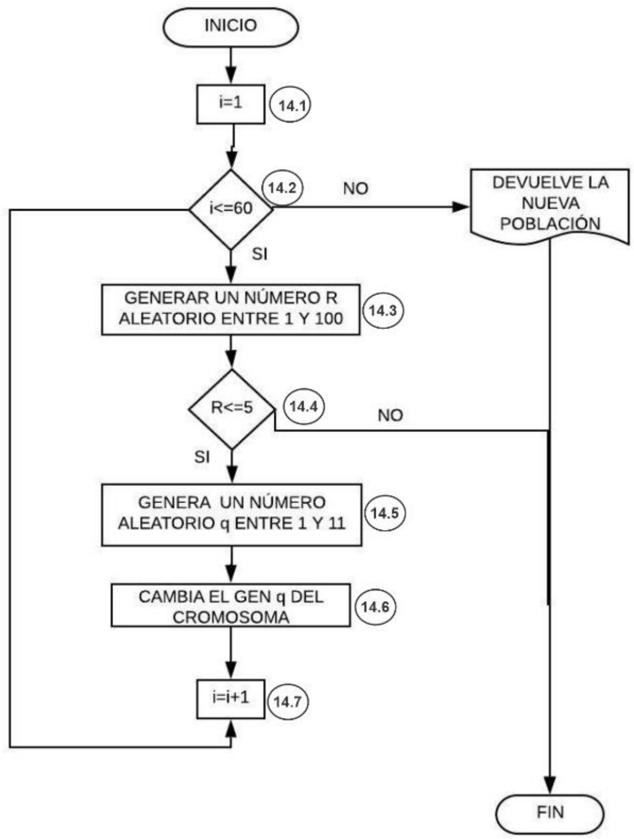



*Figura 12.* Diagrama de flujo de la función mutar.

Paso 14.1. Se tiene el contador de la población.

Paso 14.2. Entra e inicializa el contador de la población, teniendo en cuenta que si no cumple la condición devuelve la población y finaliza.

Paso 14.3. Genera un número R aleatorio que es la probabilidad de mutación dada en un rango de 1 a 100.

Paso 14.4. Entra e inicializa el contador de probabilidad de ser mutado, que no debe ser mayor al 5%, si no se cumple finaliza y no muta.

Paso 14.5. Genera un número aleatorio q entre 1 y 11, que hace relación al cromosoma con partición en el lote.

Paso 14.6. Cambia el gen q del cromosoma aleatoriamente.

Paso 14.7. Se le suma 1 a la población y finaliza.

Paso 15. Se le suma 1 al número de generaciones y se finaliza.

## 7. Validación del Algoritmo

Para la validación del algoritmo genético propuesto, se toman datos experimentales, planteando 5 escenarios difrentes en los que se generan 3 productos, dado a la complejidad del problema el algoritmo diseñado para matlab solo permite incluir maximo 3 productos por la capacidad del Hardware, los datos se encuentran en el apéndice B, y su descripción se plantea a continuación con su respectivo análisis.

**Parámetros:**

$i$: Índice del producto $i: 1, \ldots \ldots \ldots .3\ productos$

$j$: Índice del productor anterior $j: 1, \ldots \ldots \ldots .3\ productos$

$P_i$: Tasa de producción para el producto $i$

$D_i$: Demanda para el producto $i$

$h_i$: Costo de mantenimiento de inventario para el producto $i$

C: Matriz de costos de alistamiento

$C_{ij}$: Costo de alistar el producto $i$ dependiente del producto anterior $j$

S: Matriz de tiempos de alistamiento

$S_{ij}$: Tiempo de alistar el producto $i$ dependiente del producto anterior $j$

**Escenarios**

**Escenario 1:** Se presentan los datos de prueba para el problema

**Escenario 2:** En base a los datos presentados en el escenario número 1 se elevan los costos de la matriz de los costos de alistamiento.

**Escenario 3:** En base a los datos de prueba se incrementan los costos de mantener inventario.

**Escenario 4:** En base a los datos originales se reducen los costos de la matriz de costos de alistamiento para algunos productos y de igual manera se reducen los tiempos en la matriz de tiempos de alistamiento.

**Escenario 5:** En base a los datos originales se aumentan los costos en la matriz de costos de alistamiento y se reducen los costos de mantener inventario.

**Escenario 1**

Para el primer escenario se tienen en cuenta los 3 productos con un dia de trabajo de 24 horas. Se tienen los siguientes resultados mostrados en la Tabla 3, adicional a esto se muestra el orden de la secuencia de producción y la partición del lote. Por último se muestra la mejor solución en la Figura 13.

$TC$: Costo total de producir en función al tiempo de ciclo

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada producto $i$

Tabla 3.

*Resultados para el Escenario 1 para 3 productos*

| TABLA DE RESULTADOS | |
|---|---|
| TC | 15,212 ($) |
| T | 5 (días) |
| tp1 | 0,750 (días) |
| tp2 | 1 (días) |
| tp3 | 1 (días) |

Productos: 1 2 3

Partición de los lotes: 1 2 1

Orden de producción:1 2 3 2

| MEJOR SOLUCIÓN | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

*Figura 13.* Mejor solución para el Escenario 1 para 3 Productos

Los datos obtenidos en la Tabla 3 dan a entender que para los 5 días que es el tiempo de ciclo cuesta producir los 3 elementos $ 15.212 en ese periodo de tiempo, dado por el cromosoma de la mejor solución con los primeros 8 genes del cromosoma tal y como se explicó en el capítulo anterior, teniendo en cuenta que a esto se le suman los tiempos de alistamiento y de esta forma se obtiene un tiempo de ciclo de 5 días.

0 0 0 0 0 1 0 0

8 7 6 5 4 3 2 1

Dado que el cromosoma es un número binario se lee de la siguiente forma:

$$2^{n-1}$$

Dado que n es la posición de cada bit, entonces se obtiene

$$T = 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$T = 4$$

Y por útlimo con la suma de los tiempos de alistamiento se tiene que

$$T = 5$$

La suma de los tiempos de producción sobre el tiempo de ciclo genera el porcentaje de utilización de la máquina. Dada mejor solución según muestra la Figura 13 se tiene que el producto 2 genera sublotes por lo que en este primer escenario se tendrían 4 lotes generados tal y como muestra el orden de producción.

**Escenario 2**

Para el siguiente escenario, se hace un incremento en la matriz de costos de alistamiento en pasar del producto 1 al producto 3, del producto 2 al producto 1 y del producto 2 al producto 3, se espera que el costo total sufra un incremento, a su vez se espera que el tiempo de ciclo, los tiempos de producción y el número de particiones sea menor.

$TC$: Costo total de producir en función al tiempo de ciclo

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada producto $i$

Tabla 4

*Resultados del Escenario 2 para 3 Productos*

| TABLA DE RESULTADOS | |
|---|---|
| TC | 18,338 ($) |
| T | 4,8 (días) |
| tp1 | 0,720 (días) |
| tp2 | 0,960 (días) |
| tp3 | 0,960 (días) |

Productos: 1 2 3

Partición de los lotes: 1 1 1

Orden de producción: 3 1 2

| MEJOR SOLUCIÓN | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

*Figura 14.* Mejor Solución para el Escenario 2 para 3 Productos

Observando los resultados obtenidos anteriormente se cumplen las suposiciones planteadas inicialmente, y se tiene que para este escenario no se genera partición en el lote dado que considera menos costoso no partirlo, tal comportamiento se observa en el cromosoma de la mejor solución representado en la Figura 14.

**Escenario 3**

Para el escenario número 3 se incrementan los costos de mantener inventario para todos los productos y permanecen constantes los costos de alistamiento, por lo que se espera que haya incremento en el costo total, que el tiempo de ciclo y los tiempos de producción disminuyan, y finalmente que el número de particiones aumente en consideración al escenario base es decir el número 1.

$TC$: Costo total de producir en función al tiempo de ciclo

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada producto $i$

Tabla 5.

*Resultados Escenario 3 para 3 Productos*

| TABLA DE RESULTADOS | |
|---|---|
| TC | 50,223 ($) |
| T | 2,250 (días) |
| tp1 | 0,338 (días) |
| tp2 | 0, 450 (días) |
| tp3 | 0,450 (días) |

Productos: 1 2 3

Partición de los lotes: 1 2 2

Orden de producción: 2 3 3 2 1

| MEJOR SOLUCIÓN |
|---|
| 0 0 0 0 0 0 0 1 0 1 1 |

*Figura 15.* Mejor Solución para el Escenario 3 para 3 Productos

Tal y como lo muestran los resultados arrojados anteriormente se comprueban las suposiciones planteadas obteniendo que si se incrementan los costos de mantener inventario el costo total aumenta de una forma considerable comparado con la instancia inicial y los tiempos de producción tanto como el tiempo de ciclo disminuyen de forma favorable. De igual forma el cromosoma de la mejor solución representa que se parte el lote 2 y 3.

**Escenario 4**

Para este escenario se plantea disminuir la matriz costos de alistamiento para algunos productos sobre los datos originales y se reduce el tiempo de alistamiento para algunos productos igualmente, esperando que el costo total, los tiempos de producción y el tiempo de ciclo sean menores comparados con los resultados del escenario original es decir el número 1, también se espera que la partición de los lotes disminuya.

$TC$: Costo total de producir en función al tiempo de ciclo.

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada producto $i$

Tabla 6.

*Resultados Escenario 4 para 3 Productos*

| TABLA DE RESULTADOS | |
|---|---|
| TC | 12,662 ($) |
| T | 3,350 (días) |
| tp1 | 0,502 (días) |
| tp2 | 0,670 (días) |

| | |
|---|---|
| tp3 | 0,670 (días) |

Productos: 1 2 3

Partición de los lotes: 1 1 1

Orden de producción: 3 1 2

| MEJOR SOLUCIÓN |
|---|
| 0 0 0 0 0 0 1 1 0 0 0 |

*Figura 16.* Mejor Solución para el Escenario 4 para 3 Productos

Observando los resultados obtenidos se puede concluir que se cumple con las suposiciones realizadas, a su vez permite ver que el lote no se fracciona ya que según los datos aportados el programa toma en consideración que partirlo más de una vez no genera una solución factible por lo que escoge la mejor solución generada.

**Escenario 5**

Para el último escenario, se tiene que la matriz de costos de alistamiento incrementa y a su vez se disminuyen los costos de mantener inventario para los 3 productos tal y como se registra en el apéndice B. Para esto se espera como resultado que el costo total se eleve, los tiempos de producción, el tiempo de ciclo disminuyan y la partición del lote aumente.

$TC$: Costo total de producir en función al tiempo de ciclo.

$T$: Tiempo de ciclo

$t_i^p$: Tiempo de producción para cada producto $i$

Tabla 7.

*Resultados del Escenario 5 para 3 Productos*

| TABLA DE RESULTADOS | |
|---|---|
| TC | 5,801 ($) |
| T | 17 (días) |
| tp1 | 2,550 (días) |
| tp2 | 3,4(días) |
| tp3 | 3,4 (días) |

Productos: 1 2 3

Partición de los lotes: 1 2 1

Orden de producción: 1 2 3 2

| MEJOR SOLUCIÓN |
|---|
| 0 0 0 1 0 0 0 0 0 1 0 |

*Figura 17.* Mejor Solución del Escenario 5 para 3 Productos

Los resultados obtenidos para el último escenario se tienen como conclusión que las suposiciones inicialmente realizadas se cumplen y a su vez da entender que si es más económico mantener en inventario los productos sus costos son menores y sus tiempos de producción se reducen notablemente frente al escenario planteado. Y de igual forma el cromosoma de mejor solución muestra que se parte el lote del producto 2.

# 8. Resultados de Validación

En la siguiente sección se presenta una comparación entre los resultados del algoritmo diseñado para fraccionar el lote y los resultados del algoritmo con el modelo clásico del ELSP, de esta manera se espera evaluar la efectividad del algoritmo propuesto para el presente estudio. Para ello se registran las tablas de resultados por cada algoritmo en cada uno de los escenarios propuestos anteriormente para la situación en la que se tiene 3 productos, tal y como se registra en el apéndice B en la hoja de cálculo llamada comportamiento de los algoritmos.

Seguidamente del registro de datos se procede a realizar 10 gráficas que representan cada escenario para medir la efectividad de los algoritmos en relación con los costos y los tiempos de producción y del tiempo de ciclo, estas están representadas por medio del gráfico de líneas.

**Análisis del Escenario 1**

En la Figura 18 se observa el comportamiento del costo total para el problema de programación de lotes económicos con partición en el lote y sin partición en el mismo, dando como resultado factible para los datos estándar de prueba la opción de fraccionar el lote para que los costos totales sean mínimos.

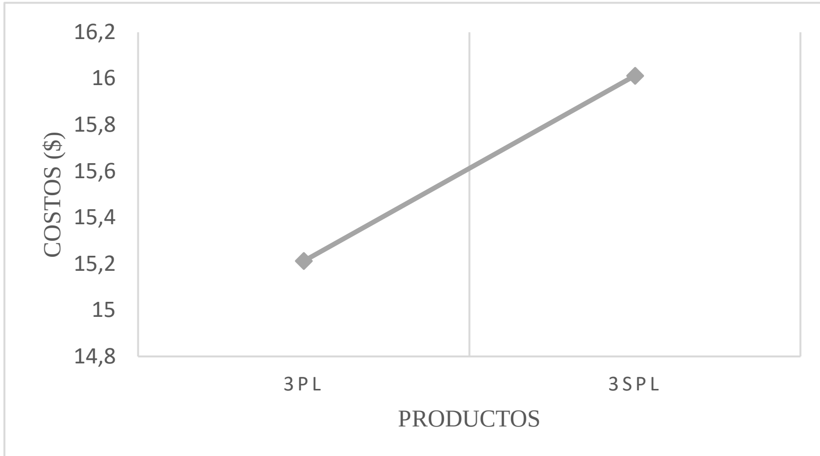



*Figura 18.* Análisis del Escenario 1 con Partición y sin Partición del Lote en Relación al Costo Total para los 3 Productos

En la Figura 19 se observa el comportamiento del modelo en cuanto al tiempo de ciclo, y los tiempos de producción, permitiendo concluir que es más demorada la fabricación de los productos si se fracciona el lote, dado a que para este caso el producto número 2 se fracciona en 2 partes lo que aumenta el tiempo de ciclo y los tiempos de producción en relación a tomar el problema sin la opción de partir los lotes, sin embrago dado los resultados anteriores se demuestra que el método presentado en el estudio es factible.

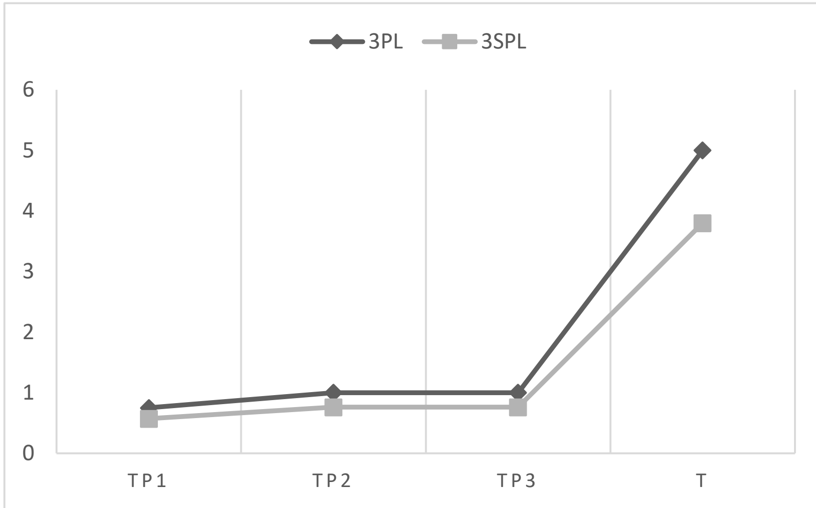



*Figura 19.* Análisis del Escenario 1 con Partición y sin Partición del Lote en relación con los Tiempos de Producción y el Tiempo de Ciclo

**Análisis del Escenario 2**

Para este escenario con los datos planteado se tiene que el programa que genera la partición del lote arroja iguales resultados en relación con el problema clásico de ELSP, debido a que como lo mostraban los resultados de validación parte el lote sólo 1 vez para cada producto, por ello se observa en la Figura 20 una línea que representa un valor constante para ambos casos.

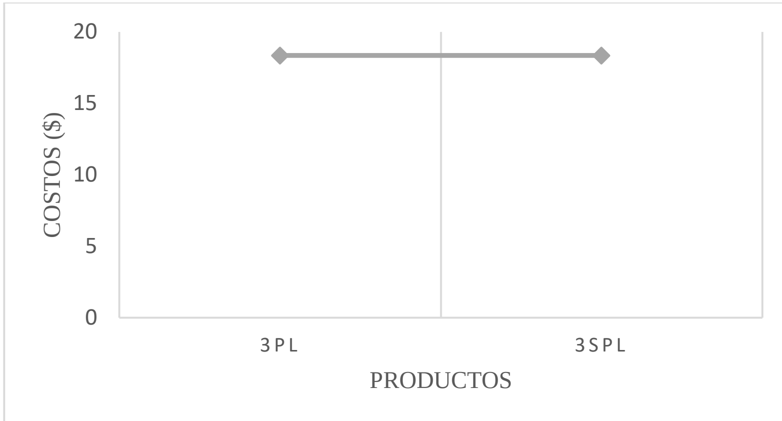



*Figura 20.* Análisis del Escenario 2 con Partición y sin Partición del Lote en relación con el Costo Total para los 3 Productos

Igualmente, en la Figura 21 se observa que los resultados para los tiempos de producción y el tiempo de ciclo son los mismos para ambas situaciones, es decir se parta o no el lote dado las condiciones del escenario y los datos de prueba utilizados, tal y como se mencionó anteriormente.

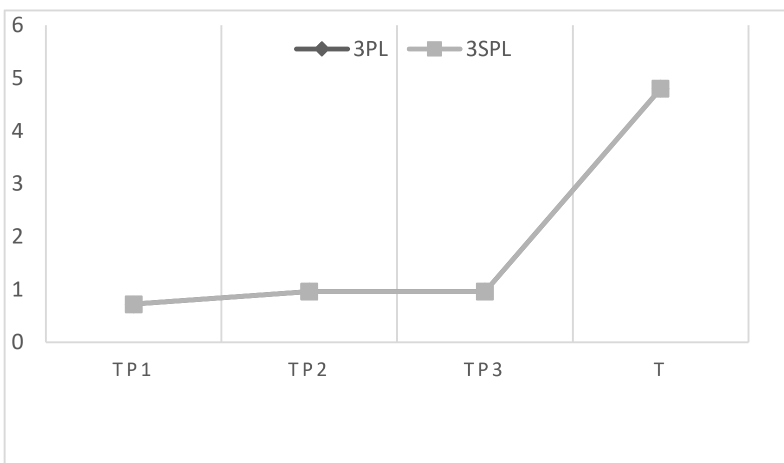



*Figura 21.* Análisis del Escenario 2 con Partición y sin Partición del Lote en relación con los Tiempos de Producción y el Tiempo de Ciclo

**Análisis del Escenario 3**

En la Figura 22 se muestra que para este escenario planteado se obtiene de igual forma que en los escenarios número 1, que los costos totales son menores partiendo el lote en comparación al problema clásico.

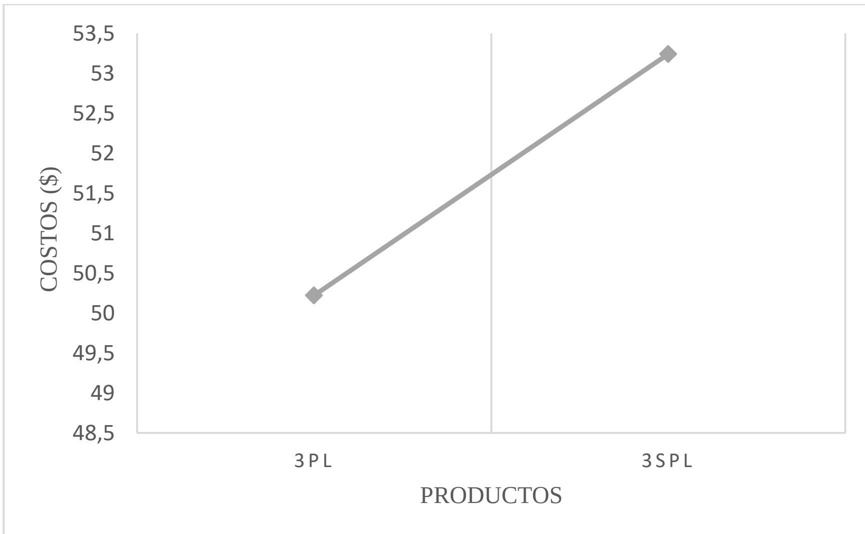



*Figura 22.* Análisis del Escenario 3 con Partición y sin Partición del Lote en relación con el Costo Total para los 3 Productos

De igual forma se muestra en la Figura 23 que los tiempos de producción y de ciclo son mayores si se hace partición en el lote de acuerdo a las características dadas del escenario, sin embargo, permite también tener una solución factible y se recomienda hacer la partición para el escenario en las condiciones propuestas.

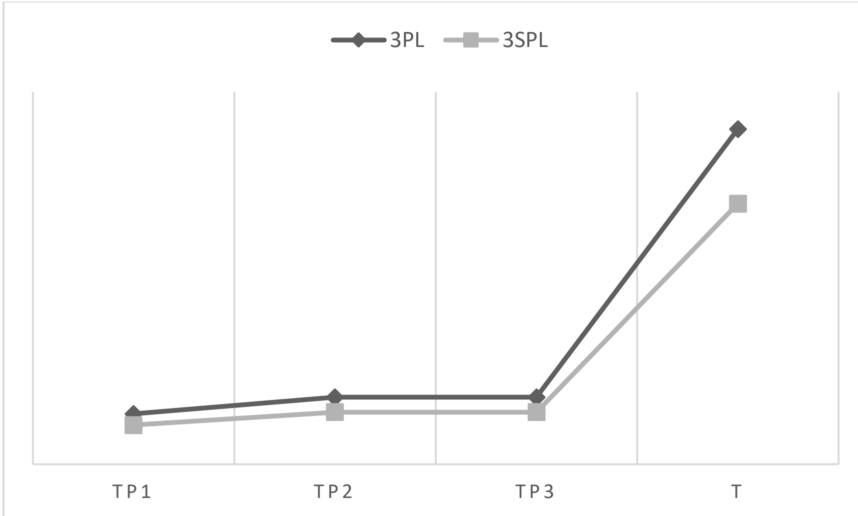



*Figura 23.* Análisis del Escenario 3 con Partición y sin Partición del Lote en relación con los Tiempos de Producción y el Tiempo de Ciclo

**Análisis del Escenario 4**

En la Figura 24 se tiene el mismo comportamiento presentado en el escenario número 2, dado que para las características presentadas arroja lo mismos resultados partiendo los productos 1 sola vez tal y como se muestran en los resultados de validación del correspondiente escenario, comprobando de igual manera una solución factible para el problema.

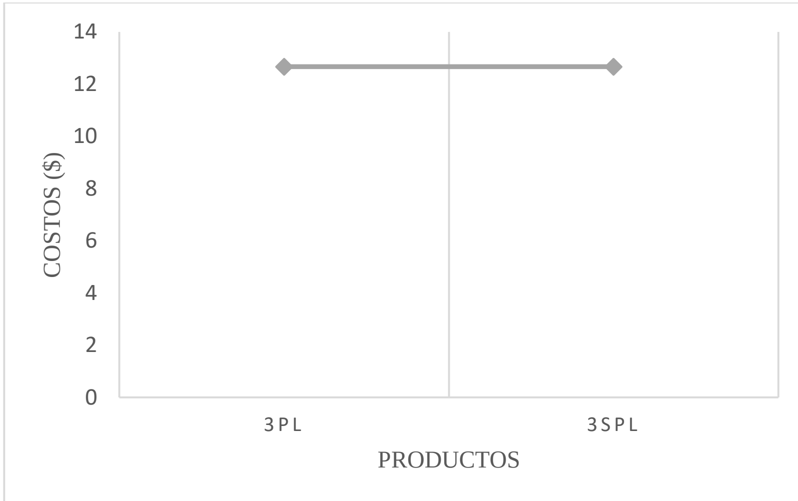



*Figura 24.* Análisis del Escenario 4 con Partición y sin Partición del Lote en Relación al Costo Total para los 3 Productos

De igual manera se repite el caso para los tiempos de producción y el tiempo de ciclo, tal y como se observa en la Figura 25.

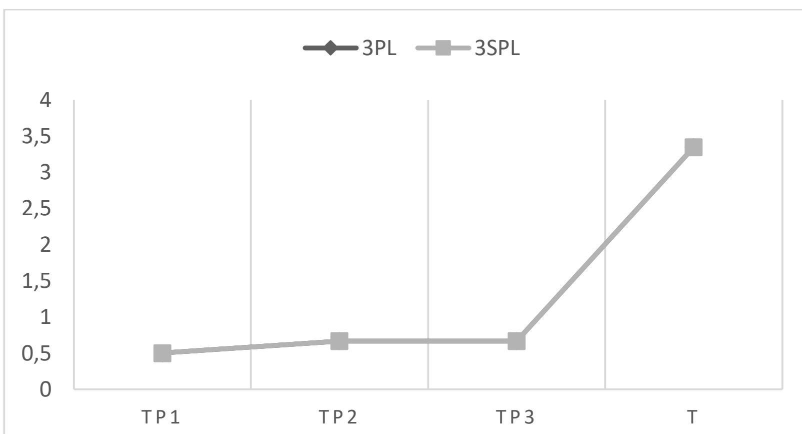



*Figura 25.* Análisis del Escenario 4 con Partición y sin Partición del Lote en relación con los Tiempos de Producción y el Tiempo de Ciclo

**Análisis del Escenario 5**

Para el último escenario presentado se tiene que el costo total es menor partiendo el lote en comparación al problema clásico, como se puede observar en la Figura 26, comprobando una vez más que el algoritmo propuesto funciona y da como resultado una solución factible para un entorno real de producción.

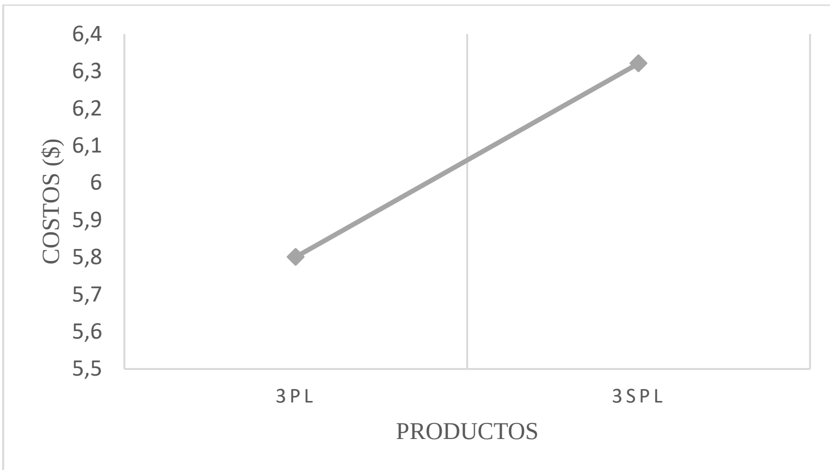



*Figura 26.* Análisis del Escenario 5 con Partición y sin Partición del Lote en Relación al Costo Total para los 3 Productos

Así mismo en la Figura 27 se observa que el problema del ELSP con las características propuestas en el estudio realizado obtiene tiempos de producción y tiempos de ciclo más largos en comparación al problema clásico.

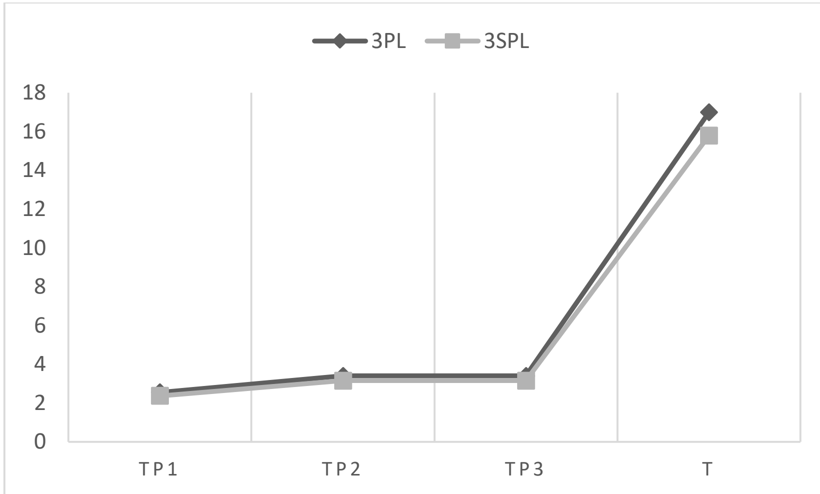



*Figura 27.* Análisis del Escenario 5 con Partición y sin Partición del Lote en Relación a los Tiempos de Producción y el Tiempo de Ciclo

Lo anterior concluye que para cada uno de los escenarios propuestos como prueba para el presente estudio con las características planteadas arrojan una solución factible y efectiva en comparación al problema clásico del ELSP.

## 9. Conclusiones

Según la revisión de literatura abordada sobre el Problema de Programación de Lotes económicos (ELSP) y sus variantes se logró identificar el interés de la comunidad científica en la solución por medio de metaheurísticas a un problema que ha sido caso de estudio por más de 50 años en un entorno real de producción, es decir que es aplicable aún en la actualidad para ayudar a la gestión de las operaciones y la mejora continua en las organizaciones. Sin embargo, en la revisión abordada no se encuentra autores que hayan desarrollado el problema del ELSP con las características propuestas en la presente investigación, lo cual permite que este proyecto sea un referente para futuras investigaciones del mismo tema.

La formulación del modelo matemático para la solución del Problema de Programación de Lotes Económicos (ELSP) con tiempos de alistamiento dependientes de la secuencia y generación de sublotes dentro del mismo ciclo de producción, permite de manera clara y precisa cumplir con las características planteadas para el problema, generando soluciones comparables entre los escenarios propuestos para las pruebas.

La aplicación de un algoritmo genético como método de solución para el ELSP con tiempos de alistamiento dependientes de la secuencia y generación de sublotes dentro del mismo ciclo de producción, logra un conjunto de mejores soluciones para las características propuestas, comprobando la efectividad de estos algoritmos para este tipo de problemas, en las instancias tomadas en cuenta el presente estudio.

Evaluando el desempeño del algoritmo genético, y analizando los datos resultantes se pudo comprobar que el algoritmo propuesto para el ELSP con partición en el lote era efectivo frente al modelo estándar para las instancias del problema encontradas en la literatura.

# 10. Recomendaciones

Dar solución al problema de programación de lotes económicos con tiempos de alistamiento dependientes de la secuencia y la generación de sublotes en un mismo ciclo de producción con demanda estocástica para futuros estudios sobre el tema.

Se recomienda para futuras investigaciones trabajar el ELSP con las características planteada en base a datos generados en un entorno real de producción.

Se recomienda para futuras investigaciones desarrollar un algoritmo dentro del algoritmo planteado para extender las posibilidades de solución y poder trabajar con más elementos, dado que para las características planteadas la memoria del Hardware no es suficiente para desarrollar el problema con datos de mayor magnitud.

## Referencias Bibliográficas

*Academic.* (2010). Obtenido de http://www.esacademic.com/dic.nsf/eswiki/1279757

Arranz de la Peña, J., & Parra Truyol, A. (2007). *Algoritmos Genéticos.* Madrid.

Berzal, F. (s.f.). *Algoritmos Geneticos.* Obtenido de Departamento de ciencias de la computacion: https://elvex.ugr.es/decsai/iaio/slides/G2%20Genetic%20Algorithms.pdf

Bomberger, E. (1966). A dynamic programming approach to a lot size scheduling probelm. *Management Science, 12*(11), 778.

Brad, M., & Golberg, D. (1995). Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*, 193-212.

Chatfield, D. (2015). The economic lot scheduling problem: A pure. *Computers & Operations Research*, 2865 - 2881. Obtenido de Available online at www.sciencedirect.com

Delporte, C., & Thomas, L. (1977). Lot sizing and sequencing for N - Products on oe facility. *Management Science, 23*(10), 1070 - 1079.

Dobson. (1987). The economic lot- scheduling roblem - achieving feasibility using time - varying lot sizes. *Operations Research, 20*(1), 50-55.

Fernandez-Baños Marín, I. (Diciembre de 2013). Programación de la secuencia de fabricación en una máquina, con tiempos de preparación variables, mediante la aplicación de algoritmos genéticos. 83. Barcelona, España. Obtenido de https://upcommons.upc.edu/bitstream/handle/2099.1/3638/31132-1.pdf

Gallego, G., & Mom, I. (1962). The effect of externalizing setups in the economic lot schedulinng problem. *Operation Research, 40*(3), 614-619.

Hanssmann, F. (1962). Operations - Research in production and inventory control. *New York*.

Hernández, J. O., Hernández , S., & Flores, I. (Septiembre de 2011). Algoritmo recocido simulado para el problema de la programación del tamaño del lote económico bajo el enfoque de ciclo básico. *19*(3).

Hochung, S., & Chan, H. (2012). A Two-Level Genetic Algorithm to Determine. *IEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 59*, 611-619.

Hsu, W. (1983). on the general feasibility test of scheduling lot sizes for several products on one machine. *Management Science, 29*(1), 93-105.

Khouja, M. (1997). The scheduling of economic lot sizes on volume flexible production systems. *International Journal of production Economics, 48*(1), 73-86.

Maldonado, C. (2013). Un problema Fundamental en la investigación: Los problemas P vs NP. *Logos, Ciencia & Tecnología, 4*(2). Obtenido de https://www.redalyc.org/html/5177/517751544002/

Mariano, R. (10 de mayo de 2005). Selección usando algortimos genéticos, Capitulo 3. *Trabajo de mastría, Optimización de la Genotipificación de ADN como un Problema de Selección de Características*. Cholula, Puebla, Mexico. Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/msp/rodriguez_m_m/capitulo3.pdf

Mohammadi, M., Nurmaya , M. S., & Bahreininejad, A. (2014). Optimization of economic lot scheduling problem with backordering and shelf-life considerations using calibrated metaheuristic algorithms. *Applied Mathematics and Computation*, 401- 422. Obtenido de journal homepage: www.elsevier.com/locate/amc

Moujahid, , A., Inza , I., & Larrañaga, P. (2010). Algoritmos genéticos. *Departamento de ciencias de la computación e inteligencia artificial*. País Vasco. Obtenido de http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t2geneticos.pdf

Ocampo Azocar, H., & Salazar Horning, E. (2014). Dimensionamiento de lotes y programación de una máquina para múltiples productos con setup y escasez. *Revista Ingeieria Industrial, 13*(2), 49-62. Obtenido de https://dialnet.unirioja.es/servlet/articulo?codigo=4999960

Peña Arenas, I. G., & Lopez Castro, L. F. (Junio de 2016). Algoritmo genetico para solucionar el problema de dimensionamiento y programación de lotes con costos de alistamiento dependientes de la secuencia. *Algoritmo genetico para solucionar el problema de dimensionamiento y programación de lotes con costos de alistamiento dependientes de la secuencia, 34(1)*. Bogotá, Cundinamarca, Colombia. Obtenido de http://rcientificas.uninorte.edu.co/index.php/ingenieria/article/view/6689/8965

Pinilla Manrique, J. (sábado 26 de febrero de 2011). *Investigación de Operaciones una Introducción.* Obtenido de Modelo LEP sin faltante: http://ingjox.blogspot.com/2011/02/modelo-lep-sin-faltante.html

Rogers, J. D. (Abril de 1958). A Computational Aproach to the Economic Lot Scheduling Problem. *Management Science, 4*, 264-291. Obtenido de https://pubsonline.informs.org/doi/abs/10.1287/mnsc.4.3.264

Sanchis, A., Ledezma Espino, A., Iglesias Martinez, J., García Jimenez, B., & Alonso Weber , J. M. (s.f.). Complejidad computacional. *Teoría de automatas y lenguajes formales*. Madrid: Universidad Carlos III de Madrid. Obtenido de http://ocw.uc3m.es/ingenieriainformatica/teoria-de-automatas-y-lenguajes-formales/material-de-clase-1/tema-8-complejidad-computacional

SImone Zanoni, A. S. (2011). Multi-product economic lot scheduling problem with manufacturing. *elsevier*, 1025-1033.

Srinvasan, G. (25 de Junio de 2012). Operations and Suply Chain Management. *Scheduling Problem, Supply Chain Inventory*. Obtenido de http://nptel.iitm.ac.in

Sun, H., Huang, H. -c., & Jaruphongsa, W. (2009). A genetic algorithm for the economic lot scheduling problem under the extended. *CIRP Journal of manufacturing science and technology*, 29-34. Obtenido de journal homepage: www.elsevier.com/locate/cirpj

Torres, J. F., & Rojas, G. S. (2015). A New Genetic Algortihm for the Economic Lot Scheduling Problem. *Industrial Engineering Department*.

Vidal Carreras, P. I. (2006). Enfoques para la resolución del problema ELSP. *Universidad Politecnica de valencia, 1(2)*, 31-43. Obtenido de file:///D:/Asus/Downloads/Dialnet-EnfoquesParaLaResolucionDelProblemaELSP-4787158%20(3).pdf

Yao, M. -J., & Huang, J. -X. (2014). Solving the economic lot scheduling problem with. *Journal of Food Engineering*, 309-322. Obtenido de www.sciencedirect.com

Zanoni, S., Sogerstedt, A., Tang, O., & Mazzoldi, L. (2011). Multi-product economic lot scheduling problem with manufacturing. *computers & industrial engineering*, 1025-1033. Obtenido de https://doi.org/10.1016/j.cie.2011.12.030

Zohali, H., Naderi, B., Mohammadi, M., & Roshanaei, V. (2018). Reformulation, linearization, and a hybrid iterated local search. *Computer and Operations Research*, 127-138. Obtenido de journal homepage: www.elsevier.com/locate/cor