

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



**IMPLEMENTACIÓN DE UN ALGORITMO MEMÉTICO PARA RESOLVER EL  
PROBLEMA DE CORTE DE MATERIALES APLICADO A LA PRODUCCIÓN  
DE BARRAS DE ACERO PARA HORMIGÓN**

**Tesis para obtener el título profesional de Ingeniero Informático**

**AUTOR**

**Andrés Enrique Gherzi Sayán**

**ASESOR**

**Mg. Rony Cueva Moscoso**

Lima, septiembre, 2021

A mi hermana Sandra  
y mis padres Enrique y Cecilia



## Resumen

El país ha vivido durante los últimos años un aumento en el PBI que lo ha colocado como uno de los principales países de la región. Este crecimiento se debe al aumento sostenido de la producción en diversos sectores. Por cifras del Ministerio de Economía y Finanzas (MEF), el sector de construcción, a cargo del Ministerio de Vivienda y Construcción representa uno de los principales contribuyentes al PBI interno (MEF), con un aporte del 5% al indicador. Además, se proyecta un crecimiento de 7% sostenido en los próximos años, por encima del promedio nacional. Esto ayudará a impulsar los sectores primarios y secundarios relacionados. Así que se vuelve prioritaria la elaboración de soluciones que aumenten la eficiencia en el consumo de recursos a todo nivel.

En particular, este proyecto plantea abordar la producción de las barras de acero para hormigón a nivel industrial. La actividad productiva por la cual el acero fundido es convertido en barras de acero grandes, que serán a su vez cortadas en longitud para conseguir barras más pequeñas, que son usadas en la manufactura y la construcción de edificios. Estas son las conocidas como barras de acero para hormigón.

El proceso consta de dos partes principalmente:

- La elaboración de barras largas estándar por el proceso de colada y
- Una segunda fase de corte para obtener los productos finales, barras de acero para hormigón, según el tamaño solicitado por los clientes.

Aunque la producción de la barra grande (primaria) se realiza en una línea de ensamblaje a partir de acero fundido, lo que la vuelve virtualmente infinita y modificable en tamaño, las barras pequeñas se elaboran a partir de un tamaño estándar que cada fábrica utiliza para la manufactura.

Para elaborar la producción de barras estándar, se toman en cuenta las negociaciones entre los ejecutivos de ventas y los clientes potenciales, obteniéndose requerimientos de producción. Dichos requerimientos son luego incluidos en la programación del periodo tomando en cuenta la disponibilidad de material (acero fundido). Debido a esto, aunque la primera etapa de corte es limpia por la naturaleza de la fabricación, en la segunda se pueden producir desperdicios y pérdidas, pues las barras grandes no siempre se utilizan al cien por ciento en la creación de barras de acero para hormigón.

El problema entonces es de corte de materiales, el cual consiste en conseguir un número de piezas de diferentes largos que deben ser cortadas de una fuente, de tal forma que se cumpla con la demanda de largos y se produzcan optimizando una función objetivo.

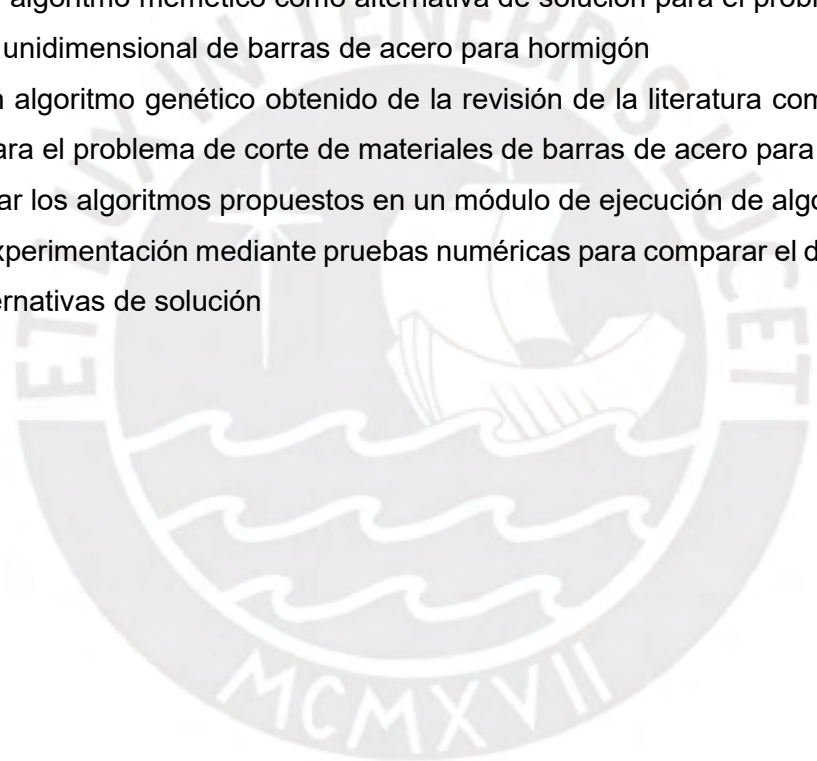
Por lo mencionado este trabajo de fin de carrera, plantea implementar un algoritmo memético para resolver el problema de corte de materiales aplicado a la producción de barras de acero para hormigón.

### **Objetivo General**

Implementar un algoritmo memético para resolver el problema de corte de materiales aplicado a la producción de barras de acero para hormigón.

### **Objetivos Específicos**

- O1. Definir la función objetivo a evaluar para los algoritmos genético y memético
- O2. Diseñar un algoritmo memético como alternativa de solución para el problema de corte de materiales unidimensional de barras de acero para hormigón
- O3. Adaptar un algoritmo genético obtenido de la revisión de la literatura como alternativa de solución para el problema de corte de materiales de barras de acero para hormigón
- O4. Implementar los algoritmos propuestos en un módulo de ejecución de algoritmos
- O5. Realizar experimentación mediante pruebas numéricas para comparar el desempeño entre ambas alternativas de solución



# Índice de Contenido

|   |     |
|---|-----|
| <b>Objetivo General</b>                               | iii |
| <b>Objetivos Específicos</b>                          | iii |
| Capítulo 1. Generalidades                             | 1   |
| 1.1 Problemática                                      | 1   |
| 1.2 Objetivos   | 4   |
| 1.2.1 Objetivo General                                | 4   |
| 1.2.2 Objetivos Específicos                           | 4   |
| 1.2.3 Resultados Esperados                            | 4   |
| 1.2.4 Mapeo de Objetivos, Resultados y Verificación   | 5   |
| 1.3 Herramientas y Métodos                            | 7   |
| 1.3.1 Herramientas                                    | 8   |
| 1.3.1.1 Apache Netbeans 11.3 (Netbeans IDE, 2020)     | 8   |
| 1.3.1.2 Lenguaje de Programación Java (Oracle, 2020)  | 8   |
| 1.3.2 Métodos y Metodologías                          | 9   |
| 1.3.2.1 Extreme Programming (Bleck, 2004)             | 9   |
| 1.3.2.2 Prueba Numéricas (Montgomery, 2013)           | 9   |
| Capítulo 2. Marco Teórico Y Conceptual                | 10  |
| 2.1 Corte de Materiales ( <i>Cutting Stock</i> )      | 10  |
| 2.1.1 Problemas NP y P                                | 12  |
| 2.1.1.1 Tiempo de Ejecución o Complejidad Algorítmica | 12  |
| 2.1.1.2 Problemas P                                   | 14  |
| 2.1.1.3 Problemas NP                                  | 15  |
| 2.1.1.4 Problemas P y NP                              | 15  |
| 2.1.1.5 Problemas NP-Hard                             | 15  |
| 2.1.1.6 Problemas NP-Complete                         | 16  |
| 2.2 Algoritmos Genéticos                              | 16  |
| 2.2.1 Fases de los Algoritmos Genéticos               | 17  |
| 2.2.1.1 Generación de la población inicial            | 17  |
| 2.2.1.2 Evaluación                                    | 17  |
| 2.2.1.3 Selección                                     | 18  |
|   | iv  |

|  |    |
|--|----|
| 2.2.1.4 Cruce o crossover                              | 18 |
| 2.2.1.5 Mutación                                       | 18 |
| 2.2.1.6 Otros operadores existentes                    | 18 |
| 2.2.2 Convergencia                                     | 19 |
| 2.2.2.1 Operadores de Reseteo                          | 19 |
| 2.2.2.2 Diversidad de Selección                        | 19 |
| 2.2.2.3 Operador de Crossover para mantener diversidad | 20 |
| 2.3 Algoritmos Meméticos                               | 20 |
| 2.3.1 Estructura de un algoritmo Memético              | 20 |
| 2.3.2 Operadores Globales                              | 21 |
| 2.3.3 Operador Local                                   | 21 |
| 2.3.4 Otros métodos meméticos                          | 22 |
| 2.4 Pruebas Numéricas                                  | 22 |
| Capítulo 3. Estado del Arte                            | 24 |
| 3.1 Método usado en la Revisión y Discusión            | 24 |
| 3.1.1 Cadenas de Búsqueda                              | 25 |
| 3.1.2 Fuentes de Información                           | 26 |
| 3.1.3 Criterios de Inclusion Y Exclusion               | 26 |
| 3.2 Revisión y Discusión                               | 26 |
| 3.2.1 Publicaciones Encontradas                        | 28 |
| 3.2.2 Preguntas de Investigación y Discusión           | 38 |
| 3.3 Conclusiones                                       | 39 |
| Capítulo 4. Definición del Problema                    | 41 |
| 4.1 Introducción                                       | 41 |
| 4.2 Definición de las variables                        | 41 |
| 4.3 Restricciones del Problema                         | 42 |
| 4.4 Función Objetivo                                   | 43 |
| 4.5 Conclusiones                                       | 44 |
| Capítulo 5. Estructuras de Datos                       | 45 |
| 5.1 Introducción                                       | 45 |
| 5.2 Cromosoma de la Solución                           | 45 |
| 5.3 Estructuras de datos                               | 46 |
| 5.4 Conclusiones                                       | 48 |

|  |    |
|--|----|
| Capítulo 6. Pseudocódigos de los Algoritmos                        | 49 |
| 6.1 Introducción   | 49 |
| 6.2 Diseño del Algoritmo Genético                                  | 49 |
| 6.2.1 Operador de Selección  | 50 |
| 6.2.2 Operador de Crossover  | 51 |
| 6.2.3 Operador de Mutación   | 52 |
| 6.2.4 Operador de Elitismo   | 53 |
| 6.3 Diseño del Algoritmo Memético                                  | 53 |
| 6.3.1 Operador de Búsqueda Local                                   | 54 |
| 6.4 Conclusiones   | 55 |
| Capítulo 7. Prueba de Verificación                                 | 56 |
| 7.1 Flujo de Validación funcional                                  | 57 |
| 7.1.1 Datos de entrada   | 57 |
| 7.1.2 Representación de la Solución                                | 58 |
| 7.1.3 Función Fitness  | 59 |
| 7.1.4 Operador de selección y Operador de crossover                | 60 |
| 7.1.5 Operador de Mutación   | 61 |
| 7.1.6 Operador de Búsqueda Local                                   | 62 |
| 7.2 Conclusiones   | 63 |
| Capítulo 8. Módulo de los Algoritmos                               | 64 |
| 8.1 Introducción   | 64 |
| 8.2 Pruebas Unitarias  | 64 |
| 8.2.1 Entrada de Datos   | 65 |
| 8.2.2 Algoritmo Genético   | 66 |
| 8.2.3 Algoritmo Memético   | 68 |
| 8.2.4 Salida de Datos  | 69 |
| Capítulo 9. Módulo de Entrada de Datos y Módulo de Salida de Datos | 70 |
| 9.1 Introducción   | 70 |
| 9.2 Módulo de Entrada de Datos                                     | 71 |
| 9.3 Módulo de Salida de Datos                                      | 74 |
| Capítulo 10. Experimentación Numérica                              | 78 |
| 10.1 Introducción  | 78 |
| 10.2 Recolección de datos  | 78 |

|  |     |
|--|-----|
| 10.3 Calibración de los Algoritmos           | 80  |
| 10.3.1 Población Inicial                     | 80  |
| 10.4 Algoritmo Genético                      | 81  |
| 10.4.1 Tasa de Casamiento                    | 81  |
| 10.4.2 Tasa de Mutación                      | 82  |
| 10.4.3 Cantidad de generaciones sin mejora   | 83  |
| 10.4.4 Cantidad máxima de generaciones       | 85  |
| 10.5 Algoritmo Memético                      | 86  |
| 10.5.1 Tasa de Voracidad (Tasa Alpha)        | 86  |
| 10.5.2 Cantidad de generaciones sin mejora   | 87  |
| 10.5.3 Cantidad máxima de generaciones       | 88  |
| 10.3 Experimentación Numérica                | 89  |
| 10.3.1 Pruebas Estadísticas                  | 89  |
| 10.3.1.2 Prueba de Kolmogorov-Smirnov        | 89  |
| 10.3.1.3 Prueba F de Fisher                  | 90  |
| 10.3.1.4 Prueba Z                            | 92  |
| 10.4 Conclusiones                            | 92  |
| Capítulo 11. Conclusiones y Trabajos Futuros | 93  |
| 11.1 Conclusiones                            | 93  |
| 11.2 Conclusión del Objetivo General         | 94  |
| 11.3 Trabajos Futuros                        | 95  |
| Referencias                                  | 96  |
| Anexo A: Calibración de la Población Inicial | 99  |
| Anexo B: Calibración del algoritmo genético  | 101 |
| Anexo C: Calibración del algoritmo memético  | 118 |
| Anexo D: Experimentación Numérica            | 130 |
| Anexo E: Plan de Proyecto                    | 132 |



## Índice de Imágenes

|   |    |
|---|----|
| Imagen 1: PBI por Sector (MEF, 2019)  | 1  |
| Imagen 2: Estructura básica de un problema de corte y uno de packing (Dyckhoff, 1990)                                       | 11 |
| Imagen 3: Clasificación de problemas P, NP, NPC y NP Hard (Autoría propia)  | 12 |
| Imagen 4: Algoritmo de Suma (Autoría propia)  | 13 |
| Imagen 5: Complejidad exponencial y polinómica (Autoría propia)   | 14 |
| Imagen 6: Clasificación de problemas P, NP y NPC. si $P = NP$ (Autoría propia)  | 15 |
| Imagen 7: Pseudocódigo del algoritmo Genético (Autoría propia)  | 17 |
| Imagen 8: Pseudocódigo del algoritmo Memético (Autoría propia)  | 21 |
| Imagen 9: Representación esquemática del proceso de corte (Mellouli, A, et al, 2019)  | 29 |
| Imagen 10: Flujo del algoritmo genético propuesto (Mellouli, A, et al, 2019)  | 29 |
| Imagen 11: Enfoque general del algoritmo de búsqueda Cuckoo (Adaptado de Monzón, D, et al, 2019)                            | 30 |
| Imagen 12: Descripción gráfica del corte binario (Adaptado de López, I.A., et al, 2017)                                     | 31 |
| Imagen 13: Tiempo de solución según diferentes complejidades (López, I.A., et al, 2017)                                     | 31 |
| Imagen 14: Comparación de resultados del algoritmo de Colonia de Hormigas (Evtimov, G, Fidanova, S., 2018)                  | 32 |
| Imagen 15: Flujo de trabajo del manejo de piezas irregulares o defectuosas (Fathy, M, et al, 2015)                          | 32 |
| Imagen 16: Comparación entre algoritmo Greedy y Algoritmo Genético sin mutación, con 10 poblaciones (Fathy, M, et al, 2015) | 33 |
| Imagen 17: Representación de la secuencia de corte (Peláez, J, et al, 2016)   | 34 |
| Imagen 18: Proceso de evolución (Peláez, J, et al, 2016)  | 35 |
| Imagen 19: Algoritmo genético con etapa de algoritmo greedy (Bonnevay, S, et al, 2015)                                      | 36 |
| Imagen 20: Costo de manufactura para un setup (Bonnevay, S, et al, 2015)  | 36 |
| Imagen 21: Ejemplo del planteamiento (Goncalvez, J., 2015)  | 37 |
| Imagen 22: Estrategia evolucionaria usada por BRKGA (Goncalvez, J., 2015)   | 37 |
| Imagen 23: Datos de Entrada (Autoría Propia)  | 57 |

|  |    |
|--|----|
| Imagen 24: Representación interna de datos (Autoría Propia)                              | 58 |
| Imagen 25: Población Inicial (Autoría Propia)  | 59 |
| Imagen 26: Fitness Inicial (Autoría Propia)  | 59 |
| Imagen 27: Población inicial en orden de función fitness (Autoría Propia)                | 60 |
| Imagen 28: Selección por ruleta (Autoría Propia)   | 60 |
| Imagen 29: Crossover (Autoría Propia)  | 61 |
| Imagen 30: Corrección aleatoria (Autoría Propia)   | 61 |
| Imagen 31: Mutación (Autoría Propia)   | 62 |
| Imagen 32: Búsqueda local 1 (Autoría Propia)   | 62 |
| Imagen 33: Búsqueda local 2(Autoría Propia)  | 63 |
| Imagen 34: Menú principal de la aplicación (Autoría Propia)                              | 70 |
| Imagen 35: Menú de selección de archivos (Autoría Propia)                                | 71 |
| Imagen 36: Datos de entrada 1(Autoría Propia)  | 71 |
| Imagen 37: Datos de entrada 2(Autoría Propia)  | 72 |
| Imagen 38: Datos de entrada 3(Autoría Propia)  | 72 |
| Imagen 39: Menú principal luego de cargar los datos (Autoría Propia)                     | 73 |
| Imagen 40: Reporte de resultado (Autoría Propia)   | 74 |
| Imagen 41: Reporte de solución por generación (Autoría Propia)                           | 75 |
| Imagen 42: Reporte de promedios por generación (Autoría Propia)                          | 76 |
| Imagen 43::Reporte de la última generación (Autoría Propia)                              | 76 |
| Imagen 44: Menú principal (Autoría Propia)   | 77 |
| Imagen 45:Resultados de experimentación (Autoría Propia)                                 | 79 |
| Imagen 46: Resultados de experimentación numérica (Autoría Propia)                       | 79 |
| Imagen 47: Calibración de la Población Inicial (Autoría Propia)                          | 81 |
| Imagen 48: Calibración de la Tasa de Casamiento (Autoría Propia)                         | 82 |
| Imagen 49: Calibración de la Tasa de Mutación (Autoría Propia)                           | 83 |
| Imagen 50: Calibración de la Cantidad de Generación sin mejora(Autoría Propia)           | 84 |
| Imagen 51: Calibración de la Cantidad Máxima de Generaciones(Autoría Propia)             | 85 |
| Imagen 52: Calibración de la Tasa de Voracidad (Autoría Propia)                          | 86 |
| Imagen 53: Calibración de la Cantidad Máxima de Generaciones sin mejora (Autoría Propia) | 87 |

## **Indice de Tablas**

|  |           |
|--|-----------|
| Tabla 1: Mapeo de Objetivos, Resultados e Indicadores (Autoría Propia)           | <b>5</b>  |
| Tabla 2: Herramientas, Métodos y Procedimientos (Autoría Propia)                 | <b>7</b>  |
| Tabla 3: Pregunta de Investigación PICOC (Autoría Propia)                        | <b>25</b> |
| Tabla 4: Criterios de Inclusión y Exclusión (Autoría Propia)                     | <b>26</b> |
| Tabla 5: Clasificación de Publicaciones encontradas (Autoría Propia)             | <b>26</b> |
| Tabla 6: Publicaciones Encontradas (Autoría Propia)                              | <b>28</b> |
| Tabla 7: Verificación de Indicadores (Autoría Propia)                            | <b>56</b> |
| Tabla 8: Calibración de la Población Inicial (Autoría Propia)                    | <b>80</b> |
| Tabla 9: Calibración de la Tasa de Casamiento (Autoría Propia)                   | <b>81</b> |
| Tabla 10: Calibración de la Tasa de Mutación (Autoría Propia)                    | <b>82</b> |
| Tabla 11: Calibración de la Cantidad de generaciones sin mejora (Autoría Propia) | <b>83</b> |
| Tabla 12: Calibración de la Cantidad Máxima de Generaciones(Autoría Propia)      | <b>85</b> |
| Tabla 13: Calibración de la Tasa de Voracidad (Autoría Propia)                   | <b>86</b> |
| Tabla 14: Calibración de la Cantidad de Generaciones sin mejora (Autoría Propia) | <b>87</b> |
| Tabla 15: Calibración de la Cantidad Total de Generaciones (Autoría Propia)      | <b>88</b> |
| Tabla 16: Resultados Finales (Autoría Propia)                                    | <b>93</b> |

# Capítulo 1. Generalidades

## 1.1 Problemática

El país ha vivido durante los últimos años un aumento en el PBI que lo ha colocado como uno de los principales países de la región. Este crecimiento se debe al aumento sostenido en la producción de diversos sectores. Uno de estos es el de construcción, a cargo del Ministerio de Vivienda y Construcción, que incluye las actividades económicas relacionadas al desarrollo de infraestructura pública y privada. Por cifras del Ministerio de Economía y Finanzas (MEF), este rubro representa uno de los principales contribuyentes al PBI interno (MEF, 2019), con un aporte del 5% al indicador.

**Cuadro 2**  
**PRODUCTO BRUTO INTERNO POR SECTORES**

|                                    | Peso Año Base<br>2007 | (Var. % real anual) |      |      |      |      | Prom<br>2018-2021 |
|------------------------------------|-----------------------|---------------------|------|------|------|------|-------------------|
|                                    |                       | 2017                | 2018 | 2019 | 2020 | 2021 |                   |
| <b>Agropecuario</b>                | 6.0                   | 2.6                 | 4.0  | 4.2  | 4.0  | 4.0  | 4.0               |
| Agrícola                           | 3.8                   | 2.6                 | 4.4  | 4.3  | 3.9  | 3.9  | 4.1               |
| Pecuario                           | 2.2                   | 2.7                 | 3.3  | 4.2  | 4.2  | 4.2  | 4.0               |
| <b>Pesca</b>                       | 0.7                   | 4.7                 | 24.8 | 4.0  | 4.2  | 4.4  | 9.3               |
| <b>Minería e hidrocarburos</b>     | 14.4                  | 3.2                 | 2.2  | 1.9  | 1.8  | 2.6  | 2.1               |
| Minería metálica                   | 12.1                  | 4.2                 | 2.2  | 2.4  | 1.6  | 3.5  | 2.4               |
| Hidrocarburos                      | 2.2                   | -2.4                | 0.9  | 0.8  | 4.2  | 0.0  | 1.5               |
| <b>Manufactura</b>                 | 16.5                  | -0.3                | 3.7  | 3.6  | 4.0  | 4.3  | 3.9               |
| Primaria                           | 4.1                   | 1.9                 | 7.6  | 3.3  | 3.4  | 3.5  | 4.4               |
| No primaria                        | 12.4                  | -0.9                | 2.8  | 3.8  | 4.2  | 4.5  | 3.8               |
| <b>Electricidad y agua</b>         | 1.7                   | 1.1                 | 3.0  | 3.5  | 4.5  | 4.5  | 3.9               |
| <b>Construcción</b>                | 5.1                   | 2.2                 | 9.0  | 7.8  | 7.8  | 7.8  | 8.1               |
| <b>Comercio</b>                    | 10.2                  | 1.0                 | 2.6  | 4.0  | 4.2  | 4.5  | 3.8               |
| <b>Servicios</b>                   | 37.1                  | 3.4                 | 3.8  | 4.8  | 5.2  | 5.5  | 4.9               |
| <b>PBI</b>                         | 100.0                 | 2.5                 | 3.6  | 4.3  | 4.5  | 5.0  | 4.3               |
| <b>PBI primario</b>                | 25.2                  | 2.9                 | 3.7  | 2.7  | 2.6  | 3.1  | 3.0               |
| <b>PBI no primario<sup>1</sup></b> | 66.5                  | 2.2                 | 3.9  | 4.8  | 5.1  | 5.4  | 4.8               |

1/ No considera derechos de importación y otros impuestos.

Fuente: BCRP, Proyecciones MEF.

Imagen 1: PBI por Sector (MEF, 2019)

Según estas mismas cifras (MEF, 2019), se proyecta un crecimiento de 7% sostenido en los próximos años, por encima del promedio nacional. Esto ayudará a impulsar los sectores primarios productivos relacionados, que a su vez aumentarán las actividades

secundarias, debido a que para su funcionamiento se hace uso de diferentes productos como el concreto, el plástico, madera y el acero, entre otros.

Esta variedad de productos se encuentra clasificada en el rubro de actividades de manufactura no primaria. Las actividades de metalmecánica y de manufactura de materiales de construcción, entonces, también se dinamizarán.

Se calcula que el segmento de manufactura no primaria crecerá un 2.8% (MEF, 2019). Así que se vuelve prioritaria la elaboración de soluciones que aumenten la eficiencia en el consumo de recursos sea mediante el reciclado de productos secundarios o la disminución de las pérdidas creadas en las diferentes etapas de las diferentes actividades.

En particular, este proyecto plantea abordar la producción de las barras de acero para hormigón a nivel industrial. La actividad productiva por la cual el acero fundido es convertido en barra de acero grandes, que serán a su vez cortadas longitudinalmente para conseguir las barras de acero que son usadas en la manufactura y la construcción de edificios. Estas son las conocidas como barras de acero para hormigón.

Formalmente, son llamadas “barras laminadas en caliente”, obtenidas desde un proceso de fundido continuo llamado de colada continua (CAPACERO, 2019). El proceso consta de dos partes principalmente:

- La elaboración de barras largas estándar por el proceso de colada y
- Una segunda fase de corte para obtener los productos finales, barras de acero para hormigón, según el tamaño solicitado por los clientes.

Aunque la producción de la barra grande (primaria) se realiza en una línea de ensamblaje a partir de acero fundido, lo que la vuelve virtualmente infinita y modificable en tamaño, las barras pequeñas se elaboran a partir de un tamaño estándar que cada fábrica utiliza para la manufactura. La fábrica CAPAcero, una de las principales siderúrgicas de Chile, produce barras largas en el tamaño estándar de 6000 y 12000 mm, pero también ofrece largos intermedios para sus clientes (CAPACERO, 2019)

Entonces, para elaborar la producción de barras estándar de un periodo, se toman en cuenta las negociaciones entre los ejecutivos de ventas y los clientes potenciales, obteniéndose requerimientos de producción. Dichos requerimientos son luego incluidos en la programación del periodo tomando en cuenta la disponibilidad de material (acero fundido). Debido a esto, aunque la primera etapa de corte es limpia por la naturaleza de la fabricación, en la segunda etapa de corte se pueden producir desperdicios y pérdidas,

pues las barras grandes que se generan en lotes no siempre se utilizan al cien por ciento en la creación de barras de acero para hormigón.

El problema entonces es de corte de materiales, que en la literatura se conoce como el problema del corte de materiales (cutting stock problem). Tal cual Dyckhoff define en la modernidad (1981), el problema del corte de materiales consiste en conseguir un número de piezas de diferentes largos que debe ser cortadas de una fuente, de tal forma que se cumpla con la demanda de largos y se produzcan optimizando una función objetivo.

A lo largo de los años se han desarrollado diferentes soluciones al problema. Inicialmente, se abordó como un problema de programación lineal (Kantorovich) para resolver el problema del uso de materiales. Se utilizó un método de generación de columnas. Más tarde, fue resuelto por Gilmore y Gomory en 1961 con una solución basada en los procedimientos de generación de columnas.

Sin embargo, estos métodos de programación lineal no siempre son ideales. Existen ocasiones en las que restricciones especiales causan una necesidad de utilizar métodos más elaborados.

Este caso particular es un problema alcanzable solo por soluciones complejas, debido a la cantidad de soluciones imperfectas posibles y por el gran esfuerzo en recursos requerido para alcanzar las mejores soluciones. por herramientas complejas. Y conforme a la teoría expuesta en el marco teórico, se reconoce que el problema es de tipo NP.

Esto nos llama a definir el corte de las barras, además, tanto en dimensionalidad como en geometría (Martí, 2018). El problema actual considera corte unidimensional de barras y por su naturaleza unidimensional, el orden en que son realizados los cortes es trivial. Los cortes son longitudinales, unidimensionales, uno a la vez, no existen mayores restricciones como las expuestas por Wascher (2007) acerca del tipo de corte que la maquinaria realiza.

Concluimos entonces, debido a la complejidad del problema y a las ventajas y desventajas de los métodos ya existentes en la industria, y conforme a la teoría expuesta en el Capítulo Marco y a la investigación del Estado del Arte, que una solución no exacta es requerida debido a las restricciones de tiempo y además es necesario especializar la solución utilizando el conocimiento acerca del caso particular: Es necesaria una solución memética.

Por lo que este proyecto espera brindar a la industria con una herramienta que le ayude en su objetivo de minimizar los costos asociados al uso de materiales y que lo haga considerando el caso real de la elaboración de barras de acero para hormigón.

## 1.2 Objetivos

### 1.2.1 Objetivo General

Implementar un algoritmo memético para resolver el problema de corte de materiales aplicado a la producción de barras de acero para hormigón

### 1.2.2 Objetivos Específicos

- O1. Definir la función objetivo a evaluar para los algoritmos genético y memético
- O2. Diseñar un algoritmo memético como alternativa de solución para el problema de corte de materiales unidimensional de barras de acero para hormigón
- O3. Adaptar un algoritmo genético obtenido de la revisión de la literatura como alternativa de solución para el problema de corte de materiales de barras de acero para hormigón
- O4. Implementar los algoritmos propuestos en un módulo de ejecución de algoritmos
- O5. Realizar experimentación mediante pruebas numéricas para comparar el desempeño entre ambas alternativas de solución

### 1.2.3 Resultados Esperados

- R1. Definición de la función objetivo, las variables y las restricciones del caso particular de corte de materiales (O1)
- R2. Estructura de datos que serán utilizadas por el algoritmo memético: Incluye la representación de la solución y las estructuras auxiliares (O2)
- R3. Diseño en pseudocódigo del Algoritmo memético (O2)
- R4. Estructura de datos que serán utilizadas por el algoritmo genético: Incluye la representación de la solución y las estructuras auxiliares (O3)
- R5. Diseño en pseudocódigo del algoritmo genético obtenido de la revisión de la literatura (O3)
- R6. Módulo de Entrada de Datos: Incluye la interfaz de entrada y los algoritmos para estructurar los datos (O4)
- R7. Implementación del algoritmo memético propuesto (O4)
- R8. Implementación del algoritmo genético adaptado (O4)
- R9. Módulo de Salida de Datos: Incluye la interfaz de muestra de resultados y los algoritmos para interpretar los resultados (O4)

R10. Informe de la experimentación: Incluye la calibración de las variables, la experimentación numérica y la comparación y conclusiones (O5)

#### 1.2.4 Mapeo de Objetivos, Resultados y Verificación

Tabla 1: Mapeo de Objetivos, Resultados e Indicadores (Autoría Propia)

| O1. Definir la función objetivo a optimizar para los algoritmos genético y memético   |  |  |
|---|--|--|
| Resultado   | Fuente de Verificación   | Indicador Verificable Objetivamente  |
| R1. Definición de la función objetivo, las variables y las restricciones del caso particular de corte de materiales                           | Documento <ul style="list-style-type: none"> <li>- Función objetivo a ser optimizada</li> <li>- Variables a considerar para la variante del problema</li> <li>- Restricciones y parámetros del caso</li> </ul> | -Validar: <ul style="list-style-type: none"> <li>a) que la función objetivo exprese las características que representan la mejor solución,</li> <li>b) que las variables consideren los valores a evaluar en la experimentación, y</li> <li>c) que las restricciones muestran límites del caso de estudio</li> </ul> |
| O2. Diseñar un algoritmo memético como alternativa de solución para el problema de corte de materiales unidimensional                         |  |  |
| Resultado   | Fuente de Verificación   | Indicador Verificable Objetivamente  |
| R2. Estructura de datos que serán utilizadas por el algoritmo memético: Incluye la representación de la solución y las estructuras auxiliares | Documento: <ul style="list-style-type: none"> <li>- Estructura y representación de la solución</li> </ul>  | Validar que la estructura soporte la representación de la solución propuesta   |
| R3. Diseño en pseudocódigo del Algoritmo memético   | Documento: <ul style="list-style-type: none"> <li>- Pseudocódigo</li> </ul>  | Validar que el diseño del pseudocódigo resuelva el caso de estudio, Prueba de Flujo de Datos   |



| O3. Adaptar un algoritmo genético como alternativa de solución para el problema de corte de materiales de barras de acero para hormigón       |  |  |
|---|--|--|
| Resultado   | Fuente de Verificación                                     | Indicador Verificable Objetivamente  |
| R4. Estructura de datos que serán utilizadas por el algoritmo genético: Incluye la representación de la solución y las estructuras auxiliares | Documento:<br>- Estructura y representación de la solución | Validar que la estructura soporte la representación de la solución adaptada                    |
| R5. Diseño en pseudocódigo del algoritmo  | Documento:<br>- Pseudocódigo                               | Validar que el diseño del pseudocódigo aplique el algoritmo adaptado, Prueba de Flujo de Datos |
| O4. Implementar los algoritmos propuestos en un módulo de algoritmos  |  |  |
| Resultado   | Fuente de Verificación                                     | Indicador Verificable Objetivamente  |
| R6. Módulo de Entrada de Datos: Incluye la interfaz de entrada y los algoritmos para estructurar los datos                                    | Software   | Prueba Unitaria  |
| R7. Implementación del algoritmo memético propuesto   | Software   | Prueba Unitaria  |
| R8. Implementación del algoritmo genético adaptado  | Software   | Prueba Unitaria  |
| R9. Módulo de Salida de Datos: Incluye la interfaz de muestra de resultados y los algoritmos para interpretar los resultados                  | Software   | Prueba Unitaria  |

| O5. Realizar experimentación mediante pruebas numéricas para comparar el desempeño de ambas alternativas de solución de datos            |  |   |
|--|--|---|
| Resultado  | Fuente de Verificación   | Indicador Verificable Objetivamente   |
| R10. Informe de la experimentación: Incluye la calibración de las variables, la experimentación numérica y la comparación y conclusiones | Documento: <ul style="list-style-type: none"> <li>- Calibración de los algoritmos</li> <li>- Experimentación numérica</li> <li>- Reporte de resultados y conclusiones</li> </ul> | Resultado de la Evaluación de la comparación de las soluciones propuesta y adaptada |

### 1.3 Herramientas y Métodos

Este proyecto pertenece al área de Ciencias de la Computación, en su rama de Algoritmia. Se hace uso de teoría de algoritmos metaheurísticos aplicados al problema de corte de materiales (y otros conceptos explicados a detalle en el capítulo “Marco”). Estos son algoritmos no exactos, algunos inspirados en comportamientos biológicos y que son adaptados para solucionar problemas informáticos, en conjunción con herramientas, métodos y metodologías, como se muestra en el siguiente cuadro

Tabla 2: Herramientas, Métodos y Procedimientos  
(Autoría Propia)

| N° Resultado | Resultado Esperado   | Herramienta, Método o Procedimiento a usarse |
|--------------|--|--|
| 1            | Definición de la función objetivo, las variables y las restricciones | Extreme Programming<br>Netbeans              |
| 2            | Estructura de datos que serán utilizadas por el algoritmo memético   | Extreme Programming<br>Netbeans              |
| 3            | Diseño en pseudocódigo del Algoritmo memético                        | Extreme Programming<br>Netbeans              |
| 4            | Estructura de datos que serán utilizadas por el algoritmo genético   | Extreme Programming<br>Netbeans              |
| 5            | Diseño en pseudocódigo del algoritmo                                 | Extreme Programming<br>Netbeans              |
| 6            | Módulo de Entrada de Datos   | Extreme Programming<br>Netbeans              |

|    |                                       |  |
|----|---------------------------------------|--|
|    |                                       | Java   |
| 7  | Implementación del algoritmo memético | Extreme Programming<br>Netbeans<br>Java  |
| 8  | Implementación del algoritmo genético | Extreme Programming<br>Netbeans<br>Java  |
| 9  | Módulo de Salida de Datos             | Extreme Programming<br>Netbeans<br>Java  |
| 10 | Informe de la experimentación         | Análisis de Varianza<br>Prueba Kolmogorov-Smirnov<br>Prueba F de Fisher<br>Prueba Z<br>Microsoft Excel |

### 1.3.1 Herramientas

#### 1.3.1.1 Apache Netbeans 11.3 (Netbeans IDE, 2020)

Apache Netbeans es un entorno de desarrollo de Apache para la programación en diversos lenguajes. Incluye herramientas especializadas para depuración en diferentes sistemas y además es soportado por terceros que desarrollan complementos para sus interfaces, tiene licenciamiento para estudiantes, funciona en el sistema operativo Windows y tiene soporte para Java.

El entorno Apache Netbeans es usado como plataforma de programación y desarrollo de los algoritmos y los módulos de la solución.

La versión utilizada es Apache Netbeans 11.3 Java.

#### 1.3.1.2 Lenguaje de Programación Java (Oracle, 2020)

El lenguaje Java es el lenguaje orientado a objetos desarrollado por Oracle en 1996 y actualmente en la versión 14. Compatible con la mayoría de sistemas informáticos por su facilidad de implementación, alto nivel de uso y gran documentación existente. No requiere licencia comercial y las demás herramientas usadas tienen implementación en este lenguaje.

El lenguaje Java es usado para programar tanto el algoritmo propuesto como para adaptar el algoritmo con el cual se realiza la comparación.

### 1.3.2 Métodos y Metodologías

#### 1.3.2.1 Extreme Programing (Bleck, 2004)

Como marco de desarrollo del proyecto, se utiliza una variante de Extreme Programming (Bleck, 2004) que consta de las siguientes herramientas:

- Historias de usuario: Documentación de los requisitos del sistema, para declarar las funcionalidades soportadas por la librería.
- Cronograma de entregables: Planificación de las fechas límite de los avances a entregar.
- Bucle de desarrollo: Lista de entregables considerados para cada iteración, tomando en cuenta que cada iteración aumenta las funcionalidades del sistema y es además un resultado completo del proyecto.

#### 1.3.2.2 Prueba Numéricas (Montgomery, 2013)

Las pruebas estadísticas son usadas para verificar diversas hipótesis matemáticas. Se mide la efectividad del algoritmo memético propuesto, comparado con un algoritmo genético que es usado para la comparación.

Se sigue el modelo de diseño de experimentos propuesto por Montgomery (Montgomery, 2013).

## Capítulo 2. Marco Teórico Y Conceptual

El presente proyecto de fin de carrera presenta el diseño e implementación de un algoritmo de cortes en una dimensión en el contexto del problema del corte de materiales. Para su elaboración, se hace utilizan diversas herramientas informáticas y conceptos pertenecientes a la teoría de algoritmos acerca de los algoritmos de cortes y métodos de optimalidad. Con el objetivo de facilitar la comprensión de este documento, los conceptos mencionados y la base teórica sobre la que se soportan serán presentados en las siguientes páginas, de tal manera que no sea necesaria la consulta de recursos externos para la lectura. Este capítulo es desarrollado con base en la teoría expuesta en las obras literarias de los autores Dyckhoff, Cook, Talbi, Brucker y otros.

### 2.1 Corte de Materiales (*Cutting Stock*)

Basados en la teoría expuesta por Dyckhoff (Dyckhoff, 1990), se presenta el corte de materiales en la literatura bajo diversos nombres y tipologías: Cutting stock, bin packing, strip packing, vector packing, vehicle loading, line balancing, knapsack, entre otros. En este proyecto se expone la teoría relacionada al cutting stock.

La forma de general del corte de materiales consiste de dos partes:

- **Dos grupos de data básica:** Objetos grandes que sirven de fuente de materiales y objetos pequeños (ítems) que deben ser obtenidos.
- **El proceso de corte o generación de patrones geométricos** de los ítems pequeños en los ítems grandes

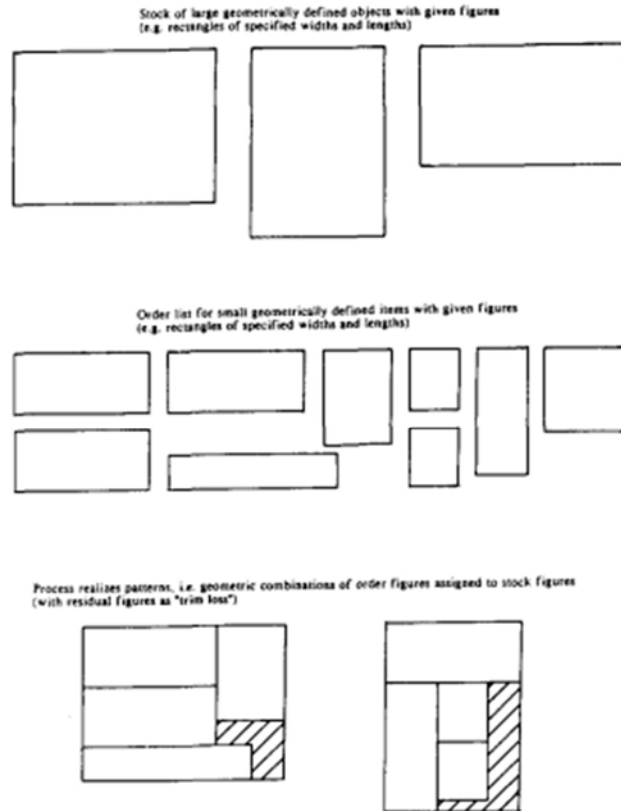


Figure 3. Basic structure of cutting and packing

Imagen 2: Estructura básica de un problema de corte y uno de packing (Dyckhoff, 1990)

El algoritmo de optimización entonces comprende la tarea interesada en el cálculo de la distribución de las piezas pequeñas en las piezas grandes de material. Es decir, el cálculo de los cortes de material para cubrir la demanda de piezas usando computadoras.

Esto se logra tomando como base algoritmos de optimización y aplicándolos al caso de estudio. Las variables a considerar en un algoritmo deberán considerar las siguientes características:

1. **Dimensionalidad:** Cantidad de dimensiones en las que existen las piezas grandes y pequeñas. Usualmente se clasifican en unidimensional, bidimensional, tridimensional y multidimensional.
2. **Medición de las cantidades:** Discreto o continuo. Cuando es discreto, se refiere a la cantidad de objetos en los cortes. Cuando es continuo, se refiere a propiedades de los objetos. Por ejemplo, el largo combinado de la selección o la fracción de los materiales.
3. **Contorno de las figuras:** La geometría de los objetos, su forma, tamaño u orientación. Además, la regularidad o irregularidad de la forma.

4. **Mezcla:** Cuáles ítems pueden ser seleccionados a la vez.
5. **Disponibilidad:** En cantidades y otros recursos. Además de las cantidades, la disponibilidad de tiempo o la precedencia.
6. **Restricciones de patrones:** Resaltan cuatro restricciones de patrones. Mínimos y máximos de cantidades, orientación de los objetos, frecuencia de objetos y el tipo de corte: Anidadas o de guillotina.
7. **Restricción de asignación:** Lo mismo que patrones, cuando se asignan a los materiales
8. **Criterio de optimalidad:** Optimización de la cantidad de objetos en un patrón, optimización de la geometría

### 2.1.1 Problemas NP y P

Para comprender el problema NP y P, se presentan los conceptos de Complejidad Algorítmica y las Clasificación

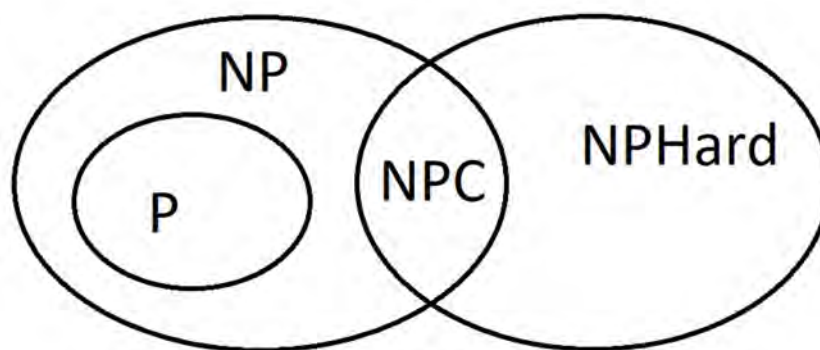


Imagen 3: Clasificación de problemas P, NP, NPC y NP Hard (Autoría propia)

#### 2.1.1.1 Tiempo de Ejecución o Complejidad Algorítmica

Para calificar la eficiencia de un algoritmo, se desarrolló la notación O (Bachman, 1894), que luego inspirará la creación de las notaciones O-grande y O-pequeña, entre otras (Landau, 1909). Estas evalúan la complejidad algorítmica de una solución mediante el cálculo de la cantidad de recursos que potencialmente pueden ser usados para ejecutar el programa. Para los cálculos de este proyecto, se utiliza la notación O-grande que se explica a continuación.

Para ilustrar, usaremos el siguiente programa que suma dos números.

```
Sumar a + b  
1. total = a  
2. Where (b > 0)  
   2.1. total = total + 1  
   2.2. b = b - 1  
3. End  
4. Return total  
End Sumar a + b
```

Imagen 4: Algoritmo de Suma (Autoría propia)

La notación O-grande define el escenario en el que el peor caso ocurre y se consume el máximo de recursos. Para el ejemplo, un algoritmo que suma  $a + b$ , sumando 1 al primer número una cantidad de veces  $b$ , se tendría en el peor caso una cantidad  $b$  de pasos (en el caso que  $b$  no sea trivial).

Mientras que, por ejemplo, la notación o-pequeña define el escenario de mejor caso, donde se consume el mínimo de recursos. Para este algoritmo, se tiene en el mejor caso una cantidad 1 de paso (en el caso que  $b = 0$ ).

Se define que el algoritmo tendrá una notación  $O(n)$  y una notación  $o(1)$ .

Con esta definición de complejidad algorítmica, podemos clasificar los problemas en categorías.

- **Problemas que se pueden resolver en tiempo Polinómico:** Aquellos cuya complejidad es una función polinómica ( $n^k$ ) de la magnitud de los datos.
- **Problemas que se pueden resolver en tiempo exponencial:** Aquellos cuya complejidad es una función exponencial ( $2^n$ ) de la magnitud de los datos.

La importancia de esta clasificación se basa en que para considerar el esfuerzo en recursos necesario para resolver un problema, podemos observar que un problema de complejidad exponencial es magnitudes más difícil de resolver que uno con magnitud polinómica.



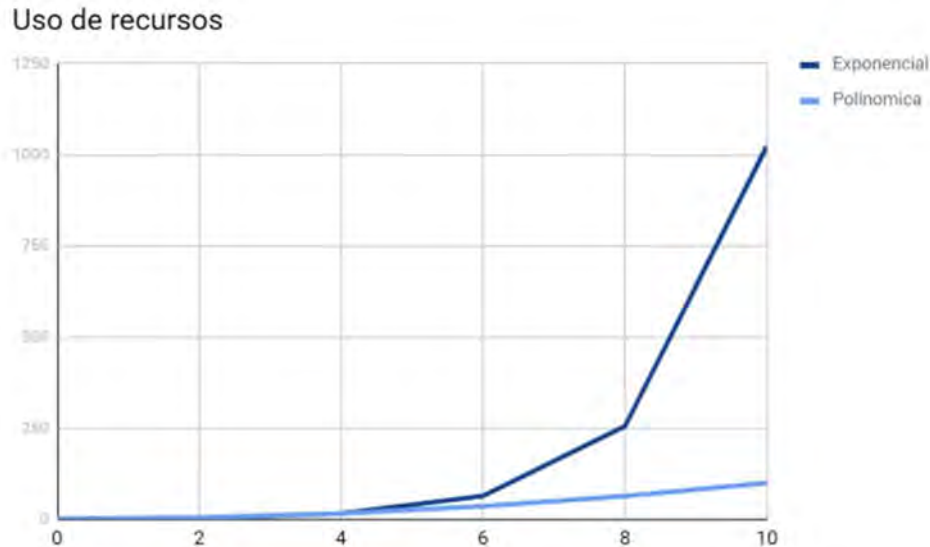


Imagen 5: Complejidad exponencial y polinómica (Autoría propia)

Esto nos servirá para definir si un problema tiene una solución difícil (exponencial) o fácil (polinómica). Que es parte de la pregunta de los problemas P y NP.

Como nota, la definición de una solución fácil o difícil es solo relativa. Pueden existir factores que ocasionen que el cálculo requiera una cantidad grande de recursos.

#### 2.1.1.2 Problemas P

Son todos los problemas de computación que actualmente tienen una solución exacta alcanzable en tiempo polinómico. (Cook, 1971) Ejemplos son el cálculo de una multiplicación simple, la verificación de un número como primo o una búsqueda lineal.

Para multiplicar  $m \times n$  sólo es necesario sumar  $n$  veces  $m$ . Con complejidad polinomial  $O(n)$ . Lo mismo para la búsqueda lineal. Solo habría que comparar  $m$  una cantidad  $n$  veces.

Incluso una búsqueda con el algoritmo MergeSort, inventado por Von Neumann, es un problema P pues puede ser resuelta en complejidad  $O(n \log n)$ . Y aunque originalmente no se conocía un método para verificar que un número sea primo (lo que lo clasificaría como no P), en 2002 el Test AKS se desarrolló para comprobar que un número es primo en tiempo (Agrawal et al, 2002)

### 2.1.1.3 Problemas NP

En esta categoría están los problemas que no son computables en tiempo polinómico y si en tiempo exponencial. Además, existe una segunda restricción: son problemas que dada una solución, esta puede ser verificable en tiempo polinómico (Cook, 1971)

Por ejemplo, el problema de Sudoku. Resolver un Sudoku requiere métodos con complejidad exponencial y la verificación de una solución requiere complejidad algorítmica.

### 2.1.1.4 Problemas P y NP

Entonces, El problema P y NP plantea la cuestión de si ambas categorías P y NP son equivalentes. Lo que implicaría que todos los problemas NP (que por ahora solo se resuelven en tiempo exponencial) podrían ser resueltos en tiempo polinómico. (Fortnow, 2009)

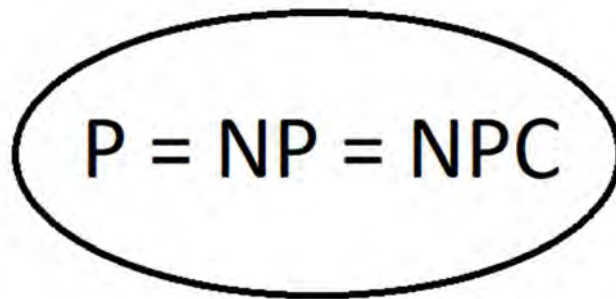


Imagen 6: Clasificación de problemas P, NP y NPC. si  $P = NP$  (Autoría propia)

Actualmente, no existe prueba formal de si son equivalentes ni de si no los son. Y una demostración permitiría encontrar solución a problemas complejos de uso de recursos, orden de pasos e incluso soluciones a enfermedades.

### 2.1.1.5 Problemas NP-Hard

Existe un tercer grupo de Problemas. Los NP-Hard, que se encuentran fuera de la categoría de NP (por lo que el nombre es solo referencial) e interseca con los NP. Estos problemas son incluso más complejos que los NP y no se tiene una solución ni una comprobación a las soluciones. (Leeuwen, 1998).

Por ejemplo, el decidir la mejor jugada en una partida de ajedrez. No existe un algoritmo para decidir la mejor jugada e incluso si se propone una jugada, no existe un método para verificar que sea la mejor.

Sin embargo, hay un grupo especial de problemas. Aquellos que clasifican como los NP-Hard, que pertenecen a los NP y que son la base de otros problemas NP.

#### 2.1.1.6 Problemas NP-Complete

Finalmente, la clasificación NP-Completa. De los problemas NP que son verificables, existe un grupo de problemas núcleo hacia los cuales otros problemas NP pueden ser reducidos (transformados). A este grupo se le llama problemas NP-Completos. Son los problemas base sobre los que se extienden otros. (Fortnow, 2009) Por ejemplo, hallar la cantidad mínima de colores necesarios para pintar un mapa puede reducirse al problema NP Completo de Vertex Labelling.

## 2.2 Algoritmos Genéticos

Los algoritmos genéticos son un tipo de algoritmos que se inspira en el comportamiento natural de los seres vivos de evolucionar para alcanzar mejores soluciones.

La idea consiste en imitar los fundamentos evolutivos de las especies para converger en soluciones óptimas. Los fundamentos son:

- Las especies evolucionan en poblaciones de individuos
- Nuevos individuos aparecen por la recombinación de individuos previos
- Los individuos mejor preparados tienen más probabilidad de sobrevivir y tener descendencia
- La mayor preparación depende del fenotipo, el genotipo y el entorno
- El genotipo se afecta por mutaciones

A partir de esto, se declara que los algoritmos reproducen estos fundamentos de la siguiente manera:

- Los individuos se conocen como cromosomas y representan soluciones al problema.
- Se codifican en genotipos que son interpretados para obtener la solución.
- La función objetivo indica cuán preparado está un individuo con el entorno, representado por el problema.
- Nuevos Individuos aparecen en una población evolutiva mediante la recombinación y mutación.

Como se aprecia, uno de los principales aportes de los algoritmos genéticos es la introducción de la idea de combinar información de diferentes soluciones (Martí, 2018).

## 2.2.1 Fases de los Algoritmos Genéticos

Programáticamente, Un algoritmo genético modela las soluciones con representaciones binarias que paso a paso, o generación a generación, se ven afectadas por operadores que simulan tanto el cruce como mutaciones y selección natural. A continuación, se presentan las fases de la metaheurística

### **Genetic Algorithm**

1. Objective function  $f(x)$
2. Generating initial population  $P_t$  ( $i=0$ )
3. Evaluate population  $P_t$
4. Where (Stop condition is not met)
  - 4.1. Select best population (candidates for reproduction)
  - 4.2. Breed new individuals  $P'_t$  (from candidates for reproduction)
  - 4.3. Mutation of new individuals  $P'_t$
  - 4.4. Evaluate population  $P'_t$
  - 4.5. Replace population  $P_t$  with  $P'_t$  when needed
5. End
6. Return best individual (solution) found

### **End Genetic**

Imagen 7: Pseudocódigo del algoritmo Genético (Autoría propia)

### 2.2.1.1 Generación de la población inicial

Los algoritmos genéticos buscan crear una solución a partir de la recombinación y variación de individuos que son parte de la población en un tiempo  $t$  anterior. Por lo tanto, se necesita una población inicial de tiempo  $t_0$  a partir de la cual se empieza la búsqueda de nuevas soluciones. Esta fase genera una población en tiempo  $t_0$  mediante diferentes métodos.

Algunos métodos son completamente aleatorios, generando individuos de cualquier tipo. Otros buscan mantener una distribución de los individuos en el espacio, como el Hipercubo Latino (Hypercube Latin) que genera concatenaciones de permutaciones de elementos para obtener diversidad en la población inicial. Un último enfoque es el heurístico donde se conoce el problema y se explotan sus características para mejorar la población inicial. (Yang, 1993)

### 2.2.1.2 Evaluación

De las soluciones existentes. En esta fase se asigna un valor de fitness, de idoneidad, a cada solución para ser seleccionada en siguientes fases. Este valor representa

usualmente un valor que referencia cuan óptima es la solución al problema. Algunos enfoques incluyen otros factores como la diversidad (Smith, et al, 1997) (Yeniay, 2005).

#### 2.2.1.3 Selección

Una vez todos los individuos son evaluados y tienen un valor de idoneidad, se seleccionan a los que podrán multiplicarse. Diversos métodos existen como selección aleatoria, ruleta, selección universal estocástica (Baker, 1987) y torneo (Goldberg, 1990).

#### 2.2.1.4 Cruce o crossover

Se escogen bits parciales de cada solución padre y se unen en una solución hijo. Existen métodos que intercambian partes de los padres, que intercambian la mitad de los bits diferentes (Half Uniform Crossover), lo que asegura que las diferencias provengan de cada padre. (Eshelman, 1991). Otros métodos son partially mapped (Goldberg, 1985) y order crossover (Davis, 1985).

#### 2.2.1.5 Mutación

El operador de mutación está más ligado a la representación en código de problema y las soluciones. Usualmente, sin embargo, se escogen aleatoriamente segmentos de la solución y se intercambian por su valor reflejo. Esto es llamado flipping y aplica principalmente para mutaciones a nivel de genes de cadenas binarias.

Otro método más complejo es el intercambio entre genes (Banzhaf, 1990). Para problemas donde se siguen secuencias de pasos, en vez de anular un paso aleatoriamente, se suelen intercambiar el orden de dos pasos. Por lo que la mutación en estos casos se realiza usando dos genes del cromosoma.

#### 2.2.1.6 Otros operadores existentes

- **Reemplazo:** Una vez generados los nuevos individuos, hay que agregarlos a la población. Diferentes métodos lo hacen de varias formas. Algunos eliminan completamente a la población original, otros seleccionan a los mejores individuos de la nueva generación y los agrega, y otros métodos insertan uno por uno a los nuevos individuos comparando. (Marti, 1998)
- **Condición:** El algoritmo se ejecuta hasta un límite. Usualmente, se escoge una cantidad máxima de generaciones, un tiempo límite de ejecución o un puntaje mínimo de fitness (Marti, 1998)

## 2.2.2 Convergencia

Debido a la forma en que funcionan los algoritmos genéticos, las nuevas soluciones suelen parecerse a sus predecesores y sus propias nuevas soluciones a las primeras, provocando que después de algunas iteraciones se termine con poblaciones invariables. Esto se conoce como convergencia prematura.

La convergencia prematura se debe principalmente a dos factores (Crepinsek, 2013):

- **La presión de selección**, que se encarga de borrar individuos con mala preparación
- **La consecuente falta de diversidad**, que ocasiona que la población consista de los mismos individuos preparados de una misma región de soluciones.

Para evitar la convergencia se han desarrollado diferentes enfoques

### 2.2.2.1 Operadores de Reseteo

El enfoque de incluir un operador que reinicie la población cuando se alcanza convergencia en un espacio de soluciones. Cuando las soluciones son muy parecidas, los subsecuentes individuos generados seguirán buscando en el mismo espacio, por lo que el operador de reset reinicia la búsqueda con otra población inicial.

Según la implementación, algunos operadores de reseteo toman las mejores soluciones y generan una población inicial basada en estos mejores individuos (Eshelman, 1991), mientras que otras implementaciones utilizan las mejores soluciones y las combinan con individuos aleatorios (Goldberg, 1989).

### 2.2.2.2 Diversidad de Selección

Donde se busca generar individuos de la mayor cantidad de subpoblaciones. Esto se logra por medio de técnicas de selección al momento de la generación de nuevos individuos.

Algunos autores, por ejemplo, le dan mayor cantidad de intentos de generación a los cromosomas con mayor diversidad (Ranking selection, Baker 1987), sin tomar en cuenta la preparación. Otros hacen lo mismo priorizando individuos con un puntaje de preparación más alejado del promedio (Disruptive selection, Kuo, 1996). FUSS (Hutter, 2006) en particular, genera individuos usando el método de selección, manteniendo una población uniformemente distribuida.

Una subcategoría de los métodos de diversidad de selección son los métodos de crowding. Estos métodos buscan que los nuevos individuos reemplacen solo a individuos parecidos genéticamente, de tal forma que se mantenga una diversidad (Crowding methods Sareni, 1998). Por ejemplo, RTS (Harik, 1995) genera un nuevo individuo y busca al individuo más parecido, comparándolos

#### 2.2.2.3 Operador de Crossover para mantener diversidad

Usualmente el crossover se realiza de manera aleatoria o seleccionando individuos con alto puntaje de preparación. Cuando se utiliza el crossover para diversidad, se eligen a individuos según su nivel de similitud. Si se eligen similares, se llama positive assortative mating. Si son diferentes, es negative assortative mating. (Marti, 1998) El método consiste en escoger aleatoriamente un padre y otros candidatos. Luego se evalúa la similitud de los candidatos y se escoge según el matting elegido. Otro método es CHC (Eshelman, 1991) que mezcla los alelos similares y cambia aleatoriamente los diferentes, consiguiendo

## 2.3 Algoritmos Meméticos

Los algoritmos meméticos son un tipo de algoritmos que incorporan conocimiento acerca del problema en las elecciones que se toman en el diseño del algoritmo solución.

La idea consiste en utilizar el conocimiento del problema para resolver las debilidades y explotar las ventajas. Este conocimiento es llamado por los autores como “meme” (Dawkin), debido a que se asemeja al término griego que significa “cosa imitada”, y da el nombre a los algoritmos Meméticos (Marti, 2018).

- Los individuos mejoran en base a algoritmos que explotan el problema
- Se mantiene un espacio de búsqueda variado en base al resto de operadores
- Se utilizan soluciones genéricas comprobadas adaptadas a problemas específicos.

### 2.3.1 Estructura de un algoritmo Memético

Programáticamente, los algoritmos meméticos funcionan junto con otros métodos, a los cuales toman como base y les agregan el conocimiento del problema. Por lo general, aplican fases de mejora poblacional (global) y luego fases de mejora individual (local).

A continuación, se presenta el pseudocódigo de un algoritmo memético que funciona en equipo con otro algoritmo para las operaciones globales.

### **Memetic Algorithm**

1. Objective Function  $f(x)$ .
2. Generating initial population  $P_t$  ( $i=0$ )
3. Evaluate population  $P_t$
4. Where (Stop condition is not met)
  - 4.1 Genetic/Other Algorithm's global operators
  - 4.2 Memetic's local search operator
5. End
6. Return best individual (solution) found

### **End Memetic**

Imagen 8: Pseudocódigo del algoritmo Memético (Autoría propia)

### 2.3.2 Operadores Globales

Estos operadores en los algoritmos meméticos son obtenidos principalmente a través de otros algoritmos. Lo que no excluye la modificación de los operadores basándonos en información del problema.

Sin embargo, la introducción de conocimiento específico a un operador global podría además dirigir la población hacia óptimos locales dentro del entorno global, por lo que lo más recomendable es mantener los métodos de generación de nuevos individuos en la configuración estable que nos permita tener variedad de soluciones y un espacio de búsqueda completo.

Por ejemplo, se podría utilizar algoritmos de colonias para encontrar espacios y búsquedas locales (en los operadores locales) para mejorar las soluciones. Otro método podría utilizar mutaciones de algoritmos genéticos para evitar la convergencia.

El óptimo se encuentra en equilibrar o configurar las fases en base al problema a resolver. Justamente, utilizar el conocimiento memético para la elección.

### 2.3.3 Operador Local

El operador local en los algoritmos meméticos es aquel que en base al conocimiento específico explota el problema para mejorar las soluciones de manera inteligente.



De las soluciones existentes, aplica operaciones de búsqueda local para aumentar el fitness de cada solución. Por lo general, de todas las soluciones modificadas.

Algunos beneficios de los operadores locales son presentados a continuación (Martí, 2018).

- Se puede escapar del óptimo local
- Pueden ser reemplazados por otros operadores locales a conveniencia
- Tienen un alto rango de posibles implementaciones. Desde métodos exactos para mejorar las soluciones hasta otros métodos heurísticos aplicados a la solución.

#### 2.3.4 Otros métodos meméticos

Finalmente, la estructura presentada no es necesariamente el modelo a seguir para los algoritmos meméticos. Otras soluciones algorítmicas pueden incluir métodos más complejos que utilizan varias estructuras a la vez funcionando para optimizar diferentes aspectos (Moscato, 1992), o hacer uso de algoritmos multi-meméticos (Krasnogor, 2002), que incluyen fases de optimización de los operadores de optimización.

### 2.4 Pruebas Numéricas

Para evaluar la calidad de la solución propuesta, se consideran tres características principalmente:

1. **Calidad de la Solución:** Tomando como referencia una solución óptima global. Ya sea porque se conoce por otros métodos o debido a cálculos parciales o triviales. Un problema de cortes, por ejemplo, considera la solución óptima como aquella en la que no se generan pérdidas.
2. **Esfuerzo Computacional:** Consumo de recursos del computador. Sea por cantidad de ciclos de hardware o por métodos teóricos de complejidad de algoritmos (notación  $O$ ).
3. **Robusticidad:** Definido en la literatura como la insensibilidad frente a desviaciones menores en los datos de entrada (o los parámetros de la metaheurística).

Estas características son comparadas entre diferentes soluciones mediante pruebas estadísticas. Algunas pruebas estadísticas comunes son:

- **T-test:** En condiciones de normalidad

- **Wilcoxon test:** Para análisis no paramétrico.
- **ANOVA:** Para comparar más de dos soluciones a la vez
- **ANOVA Multi Varianza:** Para comparar con varias características



# Capítulo 3. Estado del Arte

Con el objetivo de identificar los métodos algorítmicos que han sido usados para resolver el problema del cutting stock, se realiza una revisión del estado del arte.

Se ha encontrado que a través de los años se han presentado diversas soluciones para el problema de corte de materiales. Desde aplicaciones para cortes unidimensionales hasta métodos que incluyen parámetros temporales y de costo de recursos. En este capítulo se presenta una revisión de los planteamientos realizados para solucionar el corte en una dimensión, utilizando el método de revisión sistemática para la investigación.

## 3.1 Método usado en la Revisión y Discusión

Para la revisión del estado del arte se ha utilizado una pregunta de investigación en conjunto con cadenas de búsqueda y criterios de inclusión y exclusión.

Como método de formulación de la pregunta de investigación se utiliza el siguiente PICOC:

- **Population:** Aplicaciones para el corte de materiales en una dimensión.
- **Intervention:** Soluciones que utilizan algoritmos metaheurísticos para resolver el problema del corte de materiales.
- **Comparison:** Soluciones que utilizan algoritmos genéticos para resolver el problema
- **Outcome:** Soluciones para el corte de materiales en una dimensión que utilizan algoritmos metaheurísticos y genéticos
- **Context:** Todas las áreas donde se puedan aplicar algoritmos para el corte de materiales en una dimensión.

A continuación, se muestra la pregunta de investigación formulada:

| Pregunta de Investigación PICOC |   |
|---------------------------------|---|
| Población                       | Aplicaciones para el corte en una dimensión |
| Intervención                    | Algoritmos metaheurísticos                  |

|             |  |
|-------------|--|
| Comparación | Comparado con algoritmos genéticos   |
| Salida      | Soluciones al problema del corte de materiales en una dimensión que usan algoritmos metaheurísticos  |
| Contexto    | Todas las áreas de aplicación  |
| Pregunta    | P1 ¿Qué soluciones al problema del corte de materiales en una dimensión se han desarrollado usando algoritmos metaheurísticos y genéticos? |

Tabla 3: Pregunta de Investigación PICOC (Autoría Propia)

El siguiente paso es derivar preguntas secundarias a partir de la pregunta de investigación. Estas preguntas de subinvestigación son:

- **P1.1** ¿Qué algoritmos bio-inspirados han sido usados para resolver el problema del corte de materiales?
- **P1.2** ¿Cómo se desempeñan en comparación a los algoritmos genéticos?
- **P1.3** ¿Cómo se desempeñan los algoritmos meméticos en comparación a los algoritmos genéticos en corte de materiales unidimensional?

### 3.1.1 Cadenas de Búsqueda

A cada elemento del método PICOC se le asigna una cadena de búsqueda. Esta asignación se realiza mediante una búsqueda previa en las bases de datos e identificando palabras claves que ayuden a identificar resultados relevantes.

Estas son las cadenas:

- **Cadena de Population (C1):** cutting stock
- **Cadena de Intervention (C2):** metaheuristics
- **Cadena de Comparison (C3):** genetic
- **Cadena de Output (C4):** (“cutting stock”) AND (metaheuristics AND genetic)
- **Cadena de Contexto (C5):** Todas las áreas.

En resumen, la cadena C4 está incluida en C1, C2 y C3. La cadena C5 no existe. La cadena de búsqueda resultante para realizar la pregunta de investigación es: C4

### 3.1.2 Fuentes de Información

Debido a la cantidad de artículos, revistas indexadas y prestigio general, se ha seleccionado la base de datos SCOPUS como fuente de datos para la revisión sistemática.

### 3.1.3 Criterios de Inclusión Y Exclusión

| Criterios de Inclusión   | Criterios de Exclusión   |
|--|--|
| <b>CR1:</b> Artículos que traten acerca de soluciones al problema usando algoritmos metaheurísticos<br><b>CR2:</b> Artículos que traten acerca de soluciones al problema para cortes regulares | <b>CE1:</b> Artículos publicados en un idioma diferente al español o inglés<br><b>CE2:</b> Artículos publicados con una antigüedad mayor a 5 años. |

Tabla 4: Criterios de Inclusión y Exclusión (Autoría Propia)

El criterio de inclusión CR1 y CR2 se utilizan para escoger los artículos que analicen el uso de los algoritmos estudiados y los criterios de exclusión CR2 y CR1 sirven para obtener soluciones actualizadas y difundidas a nivel global.

## 3.2 Revisión y Discusión

Está este acápite se analizan los resultados encontrados tras aplicar el método de búsqueda escogido.

| Clasificación   | Cantidad |
|---|----------|
| Documentados encontrados por la cadena de búsqueda    | 264      |
| Documentos filtrados según criterios de exclusión     | 111      |
| Documentos seleccionados según criterios de inclusión | 8        |
| Documentos con duplicados                             | 8        |
| Documentos finales                                    | 8        |

Tabla 5: Clasificación de Publicaciones encontradas (Autoría Propia)

Los resultados encontrados y un extracto de los principales documentos se muestran en la tabla 6:

| Publicación  | Solución                                | Comparación         | Resultados  |
|--|---|---------------------|---|
| <i>An Innovative Genetic Algorithm for Multi-Objective Optimization of Two-Dimensional Cutting-Stock Problem</i> | Genético con función objetivo de Pareto | CPLEX               | Soluciones cercanas en menor tiempo (Mellouli, A, Mellouli, A, Masmoudi, F., 2019)                          |
| <i>A cuckoo search algorithm for 2d-cutting problem in decorative ceramic production lines with defects</i>      | Cuckoo                                  | Genético            | Se demuestra peor resultado que genético (Monzon, D, Cueva, R, Tupia, M, Bruzza, M., 2019)                  |
| <i>Solving binary cutting stock with matheuristics using particle swarm optimization and simulated annealing</i> | Recocido + Colonia de partículas        | Métodos exactos     | Mejor resultado que método exacto (López, I.A., Mora, J, Santos, C, Gonzalez-Mendoza, M, Monziel, C., 2017) |
| <i>Ant Colony Optimization Algorithm for 1D Cutting Stock Problem</i>  | Colonia de hormiga                      | Greedy              | Disminución de consumo de materiales, aumento de tiempo (Evtimov, G, Fidanova, S., 2018)                    |
| <i>Evolutionary-based hybrid algorithm for 2D cutting stock problem</i>  | Genético + Greedy                       | Genético + Mutación | Disminución de pérdida (Fathy, M, Osama, M, El-Mahallawy., 2015)  |
| <i>An evolutionary metaheuristic for the 2d guillotine cut problem</i>   | Genético                                | -                   | Eficiente en poblaciones pequeñas (Pelaez,  |

|  |                   |               |  |
|--|-------------------|---------------|--|
|  |                   |               | <i>J, Yanez, A, Santos, E, Moncayo, L., 2016)</i>                                |
| <i>A genetic algorithm to solve a real 2-D cutting stock problem with setup cost in the paper industry</i> | Genético + Greedy | Método humano | Reducción de costos en 30% ( <i>Bonnevay, S, Aubertin, P, Gavin, G., 2015)</i> ) |
| <i>A hybrid biased random key genetic algorithm for a production and cutting problem</i>                   | Genético + BRKGA  | -             | <1.5% de residuos ( <i>Goncalvez, J., 2015)</i> )                                |

Tabla 6: Publicaciones Encontradas (Autoría Propia)

### 3.2.1 Publicaciones Encontradas

#### **An Innovative Genetic Algorithm for Multi-Objective Optimization of Two-Dimensional Cutting-Stock Problem** (*Mellouli, A, et al, 2019*)

El paper presenta una solución para el problema de corte en dos dimensiones, considerando como parámetros la cantidad de residuos generados y la cantidad de setups (configuraciones iniciales) que deben realizarse para cubrir las necesidades del cliente.

Los residuos son contados en metros cuadrados del material que es generado y no se pertenece a ningún patrón de corte y las configuraciones iniciales se basan en la colocación de cuchillas de corte que generan bandas de diferente ancho que son procesadas por guillotinas, para conseguir los cortes bidimensionales, como se muestra en la imagen:

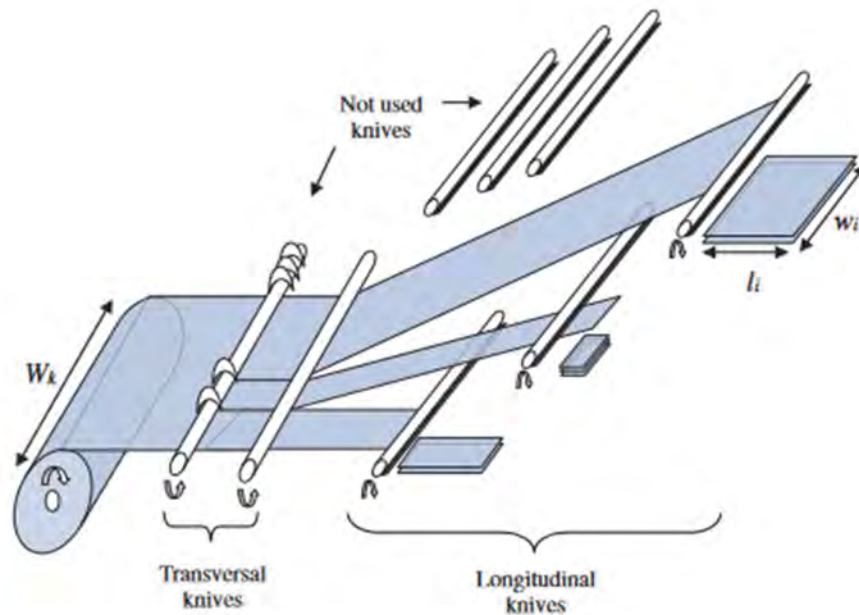


Imagen 9: Representación esquemática del proceso de corte (Mellouli, A, et al, 2019)

Para definir la función objetivo con dos parámetros, se utiliza el método PFGA de Eficiencia de Pareto, que busca hallar el punto en el que para ambos parámetros alcanzan su mayor punto sin afectar al otro.

La solución utiliza el siguiente flujo para el algoritmo genético

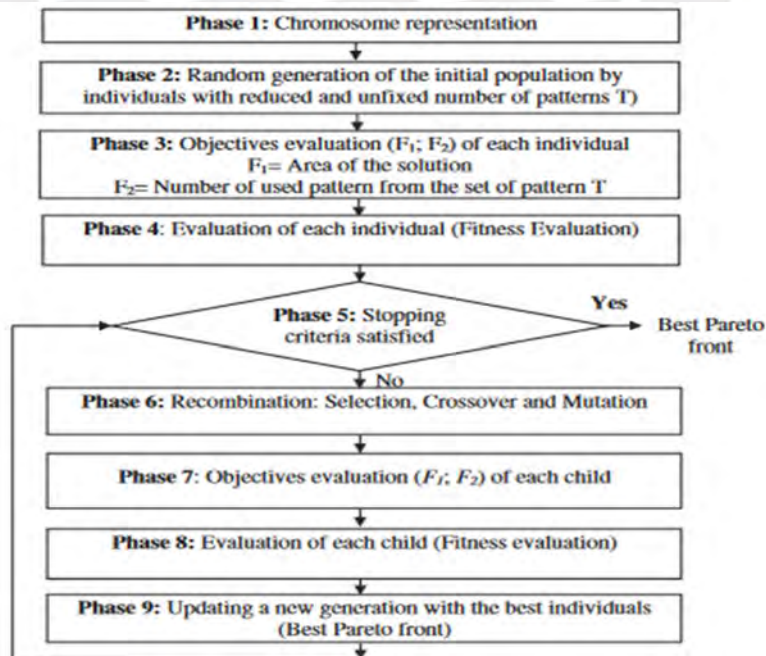


Imagen 10: Flujo del algoritmo genético propuesto (Mellouli, A, et al, 2019)



La evaluación del algoritmo se realiza comparado con un Solver CPLEX. El algoritmo genético es más rápido y alcanza soluciones cercanas a las alcanzadas por CPLEX

### **A cuckoo search algorithm for 2d-cutting problem in decorative ceramic production lines with defects (Monzon, D, et al, 2019)**

Este paper utiliza un algoritmo bioinspirado en el comportamiento de los cuckoo, aves que ponen sus huevos en nidos de otras aves para que nazcan sin costo para el cuckoo, para resolver el corte en dos dimensiones de cerámica. Se busca reducir el residuo por piezas en mal estado y por cerámicas no usadas.

#### **Cuckoo Search Algorithm (Instance of problem)**

1. Objective function  $f(x), x = (x_1, \dots, x_n)^t$
2. Generating initial population of  $N_f$  nests  $x_i (i=1, 2, \dots, n)$
3. Where ( $t < \text{Maximum of generations}$ ) or ( $\text{Stop condition is not met}$ )
  - 3.1. Obtain a cuckoo egg of "i" randomly with Levy flight
  - 3.2. Calculate its  $F_i$  fitness
  - 3.3. Select a "j" nest of the NP nests randomly
  - 3.4. Calculate its fitness  $F_j$
  - 3.5. If ( $F_i > F_j$ ) then
    - 3.5.1. Replace "j" with the new solution "i"
  - 3.6. End Si
  - 3.7. Abandon a fraction ( $P$ ) of the worst nests
  - 3.8. Build a new solution with Levy's flight
  - 3.9. Maintain best solutions (nests) for next generation
  - 3.10. Arrange from best to worst solution (nest) and save the best current one
4. End
5. Return best nest (solution) found

#### **End Cuckoo Search**

Imagen 11: Enfoque general del algoritmo de búsqueda Cuckoo (Adaptado de Monzón, D, et al, 2019)

Como especificidades de la solución se utilizan los espacios defectuosos como piezas y el tamaño inicial de los materiales a usar es suficiente para cubrir todas las órdenes sin overlap.

Los resultados hallados comparados contra un algoritmo genético muestran que el genético obtiene mejores resultados de la función objetivo. Se concluye que la diferencia puede ocurrir debido a que el algoritmo genético utiliza crossover y mutaciones para mejorar la solución, mientras que el cuckoo solo usa aleatoriedad.

**Solving binary cutting stock with matheuristics using particle swarm optimization and simulated annealing (Lopez, I.A., et al, 2017)**

Se presenta el problema de corte de materiales en la variante de corte binario. Se presenta una solución de algoritmo híbrido metaheurístico. Los algoritmos usados en el híbrido son recocido simulado y optimización de colonia de partículas.

El problema de corte binario es el corte unidimensional de rollos de material que son divididos en rollos de menor anchura.

| Material Roll |    |       |    | L=150 |       |
|---------------|----|-------|----|-------|-------|
| 10            | 50 |       |    | 80    | WASTE |
| 15            | 20 | 5     | 45 | 65    |       |
| 24            |    | WASTE |    |       |       |

Imagen 12: Descripción gráfica del corte binario (Adaptado de López, I.A., et al, 2017)

El algoritmo recocido se basa en dos fases de cocinado de las soluciones. Una primera etapa en la que las soluciones posibles son elegidas si supera un nivel mínimo de optimalidad y otra etapa en la que se agregan soluciones para permitir que se creen soluciones divergentes

El algoritmo de colonia se utiliza para simular el movimiento de población de las colonias de animales, en las que aunque existen diferentes movimientos internos, el movimiento completo de la masa de población guía al grupo.

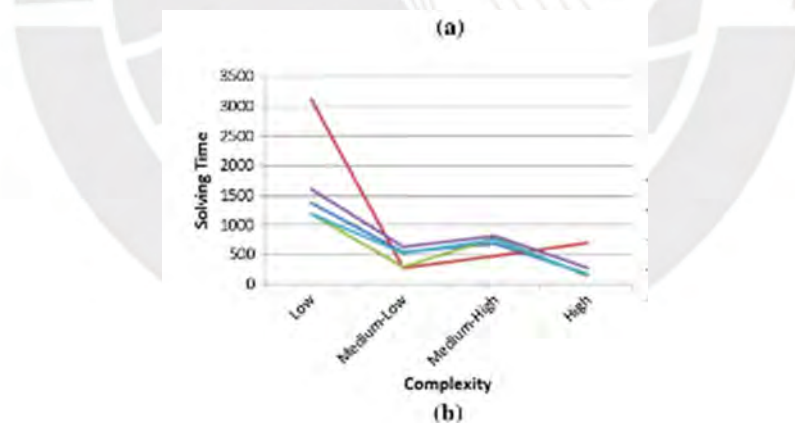


Imagen 13: Tiempo de solución según diferentes complejidades (López, I.A., et al, 2017)

Los resultados muestran menor tiempo de resolución para el algoritmo híbrido, comparado con métodos exa

**Ant Colony Optimization Algorithm for 1D Cutting Stock Problem** (Evtimov, G, Fidanova, S., 2018)

Como el nombre lo sugiere, se presenta un algoritmo inspirado en colonias de hormigas para el corte de materiales en una dimensión. El método utiliza un paralelo de la deposición de feromonas de las hormigas en un grafo que representa las soluciones. A mayor cantidad de feromonas, mayor la probabilidad de investigar en el dominio. Se actualizan las feromonas inversamente al desperdicio generado y se itera

| Properties      | Greedy | Commercial | ACO |
|-----------------|--------|------------|-----|
| Time in minutes | 2      | 20         | 10  |
| Used bars       | 234    | 234        | 232 |

Imagen 14: Comparación de resultados del algoritmo de Colonia de Hormigas (Evtimov, G, Fidanova, S., 2018)

Se compara la solución con un algoritmo greedy y se encuentra que consigue resultados mejores.

**Evolutionary-based hybrid algorithm for 2D cutting stock problem (Fathy, M, et al, 2015)**

Se presenta una solución al problema de corte de material en dos dimensiones, con manejo de materiales defectuosos. Usualmente se utilizan planchas en perfecto estado y las defectuosas se desechan. En esta solución, las planchas defectuosas son procesadas para obtener el mayor rectángulo de material disponible.

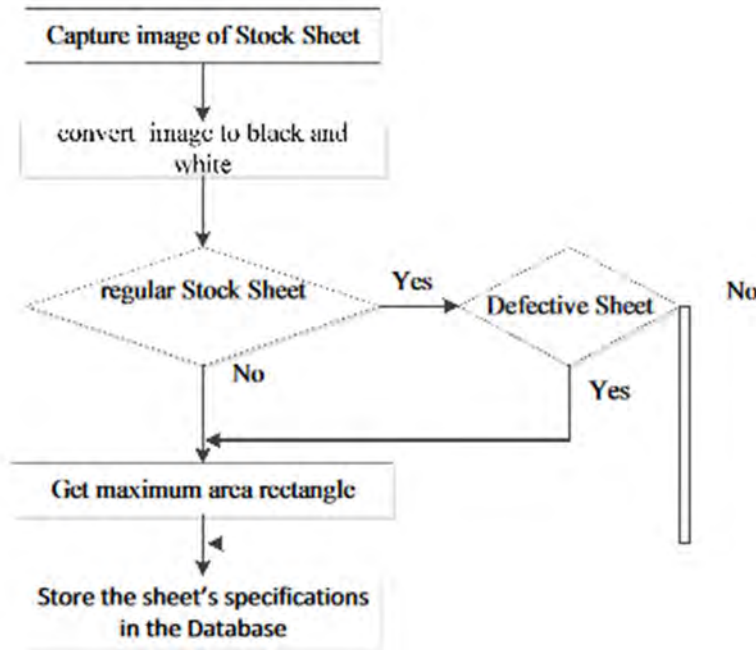


Imagen 15: Flujo de trabajo del manejo de piezas irregulares o defectuosas (Fathy, M, et al, 2015)

La solución propuesta utiliza las fases de un algoritmo genético y agrega un algoritmo greedy para la selección de las poblaciones.

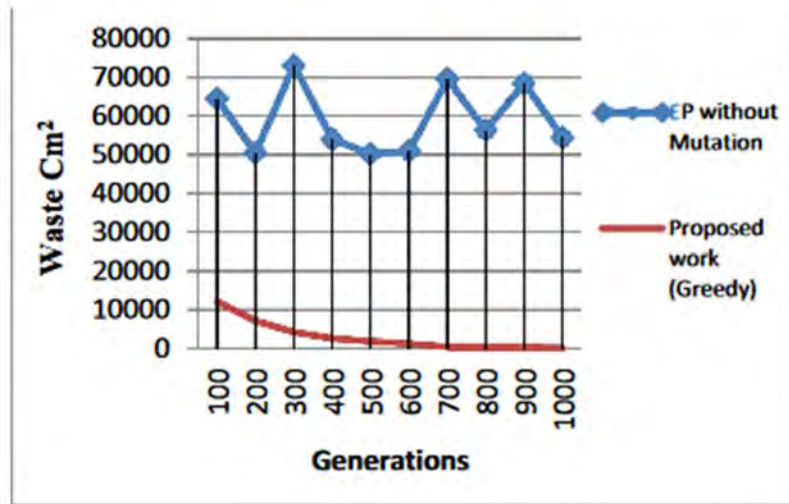


Imagen 16: Comparación entre algoritmo Greedy y Algoritmo Genético sin mutación, con 10 poblaciones (Fathy, M, et al, 2015)

Los resultados muestran una disminución en la pérdida del algoritmo híbrido comparado con un algoritmo genético con mutación.

**An evolutionary metaheuristic for the 2d guillotine cut problem** (Pelaez, J, et al, 2016)

Este paper resuelve el problema de corte, modelando con cromosoma que representan etapas de corte de un material. Los cortes se realizan en etapas y siguiendo el corte de guillotina.

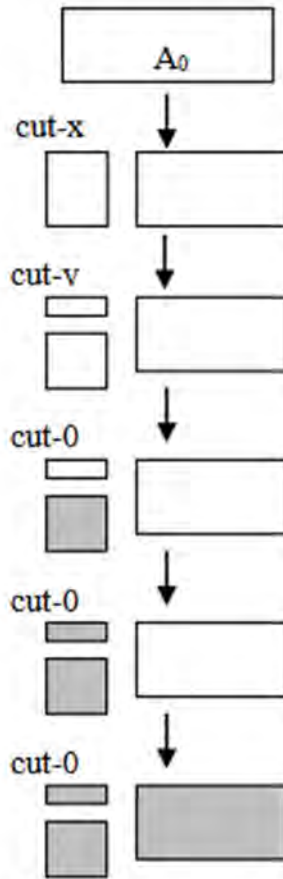


Imagen 17: Representación de la secuencia de corte (Peláez, J, et al, 2016)

El algoritmo utiliza cromosomas en tuplas de tipo de corte y coordenadas del corte. Debido a que la fase de crossover se realiza entre cromosomas del mismo nivel de corte, los elementos son agrupados en poblaciones y se eliminan elementos cuyas poblaciones no superen el mínimo de nivel de corte.

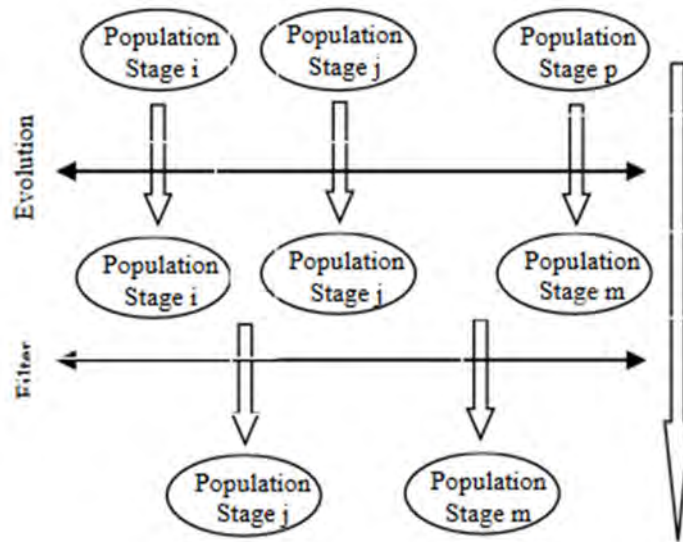


Imagen 18: Proceso de evolución (Peláez, J, et al, 2016)

Los resultados muestran soluciones exactas en el 80% de los problemas, cuando las poblaciones son pequeñas, aunque una cantidad menor a 150 poblaciones ocasiona que la eficiencia computacional se reduzca

**A genetic algorithm to solve a real 2-D cutting stock problem with setup cost in the paper industry (Bonnevay, S, et al, 2015)**

Se presenta una solución al corte de material en dos dimensiones, aplicado a una imprenta. El método usado combina un algoritmo de Bin Packing para generar cromosomas iniciales y un algoritmo genético para iterar las generaciones. Finalmente, un algoritmo greedy mejora las soluciones finales.

---

**Algorithm 1** GA-2CSP-S

---

```

1: function GA-2CSP-S
2:   BestKnown  $\leftarrow$  new Solution()
3:    $P_0 \leftarrow$  CreateInitialPopulation()
4:   BestKnown  $\leftarrow$  UpdateBestSolution( $P_0$ )
5:   for  $k \leftarrow 1$  to NbGen do
6:      $P_{k-1}^* \leftarrow$  RouletteWheelReproduction( $P_{k-1}$ )
7:      $P_k \leftarrow$  BestSolutionsReproduction(NbBest,  $P_{k-1}$ )
8:     for  $i \leftarrow$  NbBest+1 to NbPop do
9:       if Random(0,1) < ProbaCross then
10:        add Crossover( $P_{k-1}^*$ ) to  $P_k$ 
11:       else
12:        add Mutation( $P_{k-1}^*$ ) to  $P_k$ 
13:       end if
14:     end for
15:     BestKnown  $\leftarrow$  UpdateBestSolution( $P_k$ )
16:   end for
17:   HillClimbing(BestKnown)
18:   DeleteOverproduction(BestKnown)
19:   return BestKnown
20: end function

```

---

Imagen 19: Algoritmo genético con etapa de algoritmo greedy (Bonnevay, S, et al, 2015)

Se comprueba la eficiencia con datos reales y artificiales. En el caso real, se mejor el costo de la imprenta en un 30% y con datos artificiales se consigue información acerca de la proporción de patrones según el costo de setup

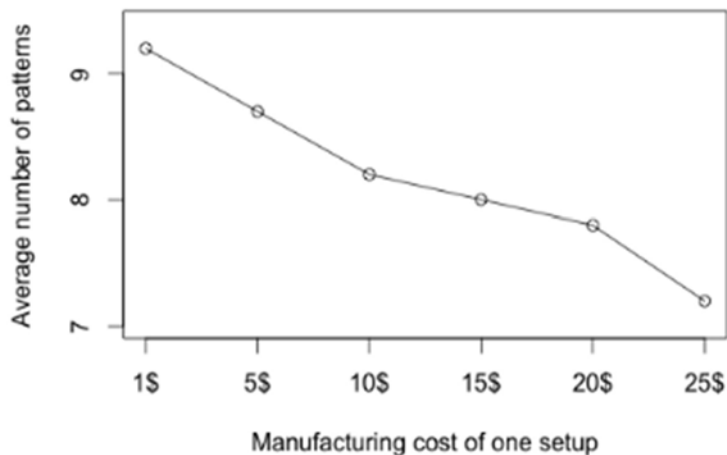


Imagen 20: Costo de manufactura para un setup (Bonnevay, S, et al, 2015)

A mayor costo de setup, la cantidad de patrones por diseño es menor.

**A hybrid biased random key genetic algorithm for a production and cutting problem (Goncalvez, J., 2015)**

El artículo presenta una solución al problema del corte de material en dos dimensiones. Se busca reducir el consumo de materiales. El problema se plantea utilizando una plancha de material en la que hay que cortar patrones rectangulares y se plantea como un problema de enteros.

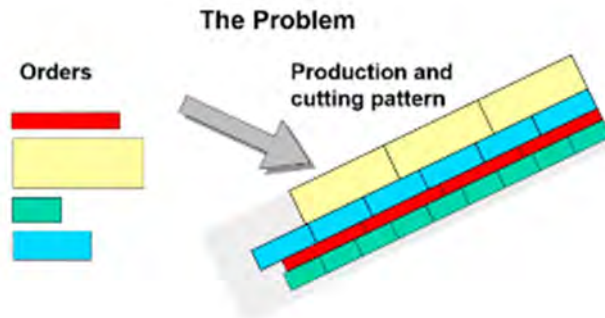


Imagen 21: Ejemplo del planteamiento (Goncalvez, J., 2015)

La solución expuesta utiliza la formulación de enteros y la resuelve con un algoritmo genético con generación parcializada de claves. La generación parcializada (BRKGA) ocasiona que los nuevos cromosomas generados en cada fase de nacimiento tengan al menos un padre elegido debido a su mayor capacidad de fitness. Esta selección ocasiona una convergencia de las soluciones

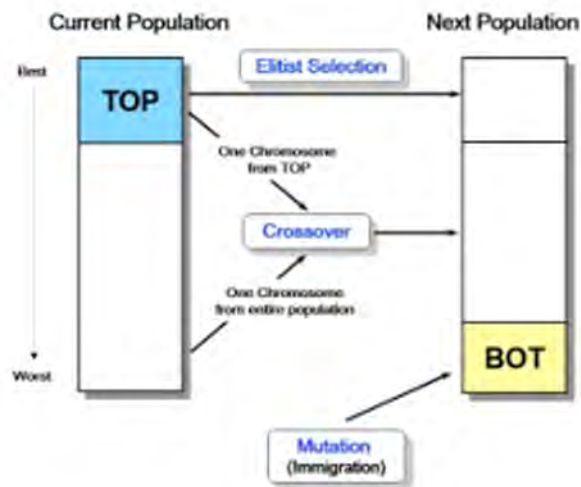


Imagen 22: Estrategia evolucionaria usada por BRKGA (Goncalvez, J., 2015)

Los resultados encontrados con el método presentan todos menos de 1.5% de material perdido.



### 3.2.2 Preguntas de Investigación y Discusión

Ahora se responden las preguntas de investigación y sub investigación.

P1. ¿Qué soluciones al problema del corte de materiales en una dimensión se han desarrollado usando algoritmos meméticos o híbridos?

Se encontraron soluciones híbridas que mayormente utilizan una fase de algoritmo greedy para seleccionar las mejores soluciones entre generaciones. Este enfoque parece ser el mejor para encontrar soluciones con bajo grado de residuos (comparado a otros algoritmos) e incluso en menor tiempo. (Fathy, M, et al, 2015)

Otra combinación es la de algoritmo genético con generación aleatoria de soluciones. Los resultados muestran soluciones con menos de 1.5% de pérdida por residuos. Esta generación aleatoria (BRKGA) asegura que al menos uno de los progenitores tenga alta capacidad en la función objetivo (Goncalvez, J., 2015)

P1.1 ¿Qué algoritmos genéticos han sido usados para resolver el problema del corte de materiales?

Se encontraron tanto soluciones puras como algoritmos con funciones objetivo multi-objetivo. En particular, se aprovecha el frente de Pareto (optimización de Pareto) para hallar una función de fitness que combine tanto costo de setup como cantidad de residuos generados. Se aplica a corte de guillotina por fases y al corte de papel en una imprenta (Mellouli, A, et al, 2019)

Se han encontrado soluciones que utilizan algoritmos genéticos puros con funciones objetivo inspirada por el teorema de Pareto y algoritmos genéticos puros. En ambos casos los algoritmos no logran mejorar los resultados.

P1.2 ¿Cómo se desempeñan en comparación a los algoritmos genéticos?

Los algoritmos híbridos consiguen mejores resultados frente a sus propios benchmark. (Fathy, M, et al, 2015). Los algoritmos genéticos puros no logran superar a sus comparaciones. No hay presencia de algoritmos meméticos comparados a algoritmos meméticos, pero se infiere que la mejor solución se obtiene con los híbridos.

P1.3 ¿Cómo se desempeñan en comparación a los algoritmos genéticos en corte de materiales unidimensional?

En corte unidimensional no existe gran cantidad de literatura. Lo más cercano es el corte de rollos de material, pero considerando rollos de cintas bidimensionales que son cortados a lo ancho para cumplir con diferentes especificaciones. (*Mellouli, A, et al, 2019*)

### 3.3 Conclusiones

Como se ha podido ver, el problema del corte de material ha sido abordado desde enfoques puros de programación evolutiva, hasta métodos híbridos y programación lineal. A continuación, se exponen las ventajas y desventajas en cada tipo de solución.

Empezando por los algoritmos genéticos puros, se aprecia un desempeño constante entre implementaciones, que por lo general consiguen soluciones cercanas incluso a las conseguidas por los métodos exactos con los que se les compara. Sin embargo, son superados por otras soluciones planteadas más complejas. (*Peláez, J, et al, 2016*) Esto podría deberse a que un método general carece de los ajustes necesarios para el problema en estudio. Fases de crossover y mutación, y fases de filtrado aseguran una mejora en las soluciones, pero no aprovechan especificidades del problema que pueden ser explotadas.

Luego están las soluciones basadas en algoritmos inspirados. Estas soluciones bioinspiradas consiguen mejores resultados en tiempo, mas no en calidad de las soluciones que los métodos genéticos. (*Monzón, D, et al, 2019*) A cambio se exhaustividad, se consigue simplificar el método algorítmica. No superan a los métodos híbridos y las versiones encontradas suelen basarse en enfoques estadísticos.

Finalmente, los métodos híbridos han conseguido buenas soluciones, tanto en calidad como en tiempo de ejecución. Incluso versiones simples de algoritmos genéticos con fases voraces consiguen mejores soluciones. (*Fathy, M, et al, 2015*). En este grupo se encuentran mezclas de algoritmos bioinspirados, genéticos e incluso recodidos simulados. (*López, I.A., et al, 2017*)

La conclusión de esta discusión del estado del arte es la recomendación de utilizar algoritmos que aprovechen las especificidades del problema a resolver. Muchos de los algoritmos encontrados en la literatura plantean soluciones en base arquetipos genéricos adaptados al problema, mas no aprovechan conocimiento del mismo para aumentar la eficacia al encontrar soluciones y terminan fallando justamente en la ejecución, en los casos reales

Los algoritmos genéticos consiguen las mejores soluciones, pero se basan en operadores de recombinación y mutación, que son operaciones genéricas, cuando una fase de búsqueda local podría aumentar la calidad. Como se demuestra en Fathy, 2015, donde se introduce un algoritmo híbrido (de programación evolutiva y lineal) capaz de procesar piezas irregulares de material en dos dimensiones, algo que sus pares no hacen, y logra superar a los mismos en el consumo de materiales.

Esto nos guía hacía el uso de algoritmos preparados para las características del problema de corte de material. Algoritmos que utilicen operaciones comprobadas en la generación de soluciones y que además aprovechen el conocimiento específico para explotar las soluciones.



# Capítulo 4. Definición del Problema

## 4.1 Introducción

En este capítulo se presenta el **Resultado Esperado 1 (RE1)**, que consiste en la definición de la función objetivo a ser optimizada, las variables del problema que intervienen en los cálculos y las restricciones del caso particular de corte de materiales unidimensional.

La forma en que está organizada cada parte es la siguiente: primero se define verbalmente la solución o las variables usadas, luego se explica la lógica seguida para formular la ecuación utilizada o las variables planteadas y finalmente se presenta la representación matemática, sea de las fórmulas o de las variables.

## 4.2 Definición de las variables

El problema a abordar consiste en minimizar la pérdida de materiales ocasionada al cortar barras de acero grandes en tamaños pequeños. Estas barras, tanto las grandes como las pequeñas, tienen forma cilíndrica con un diámetro estándar (que no interviene en los cálculos a realizar) y provienen de un origen finito de materiales. Es decir, el problema consiste en una variable unidimensional del corte de materiales, en el que se busca calcular la asignación de piezas a producir sobre piezas iniciales de materiales.

Por lo tanto, se tiene que las variables que intervienen en los cálculos son:

- Variables de la Orden de Producción

| Nombre                  | Descripción                                     |
|-------------------------|---|
| <i>CantBarrasPedido</i> | Cantidad de barras del pedido                   |
| <i>LongBarrasPedido</i> | Suma de las longitudes de las barras del pedido |
| <i>CantBarrasProd</i>   | Cantidad de barras del pedido producidas        |

- Variables de las piezas de material disponibles para la Orden de Producción

| Nombre                | Descripción   |
|-----------------------|---|
| <i>CantMateriales</i> | Cantidad de barras de material que están disponibles para cubrir la demanda de producción |
| <i>TamMaterial</i>    | Tamaño de cada barra de material  |

- Variables de las barras de producción

| Nombre                | Descripción  |
|-----------------------|--|
| <i>CantBarrasTipo</i> | Cantidad de barras del tipo, en la orden de producción     |
| <i>LongBarraTipo</i>  | Longitud de cada barra del tipo, en la orden de producción |

- Variables de las barras producidas

| Nombre                  | Descripción  |
|-------------------------|--|
| <i>PosiciónMaterial</i> | Posición de la barra producida en la lista de materiales |
| <i>PosicionBarra</i>    | Posición de la barra en un material                      |
| <i>LongBarra</i>        | Longitud de la barra cortada                             |

### 4.3 Restricciones del Problema

- **RES1: La cantidad de barras a producir debe ser menor o igual a la cantidad de barras de material:** Porque, aunque las soluciones serán por lo general de varias barras a producir en una barra de material, es necesario cubrir todas las posibilidades. Además, la Orden de Producción debe contener por lo menos una barra a producir.

$$CantBarrasPedido \leq CantMateriales$$

- **RES2: El tamaño de una barra a producir debe ser menor o igual al tamaño de la barra de material y el tamaño total de las barras a producir debe ser menor o igual al tamaño total de las barras de material.** Para que una Orden de Producción sea calculada, los tamaños deben ser consistentes.

$$\forall CantBarrasPedido / LongBarraTipo \leq TamMaterial$$

$$LongBarrasPedido \leq \sum_0^n LongBarraTipo * CantBarrasTipo$$

- **RES3: Las barras a producir no deben traslaparse entre sí.** Porque representan barras de material real que ocupa todo el espacio asignado.

$$\exists i, j / PosicionBarra(j) \leq PosicionBarra(i) + LongBarra(i)$$

- **RES4: Todas las barras a producir deben ser producidas.** Es decir, todas las barras a producir deben ser asignadas.

$$CantBarrasPedido = CantBarrasProd$$

- **RES5: El total del tamaño de la barra producida debe caber en una y solo una barra de material:** Las barras producidas no pueden pertenecer a más de una barra de material, porque son cortadas en una pieza.

$$\exists i, j / PosicionMaterial = j \wedge PosicionMaterial = i$$

## 4.4 Función Objetivo

Con las variables presentadas y las restricciones del problema, realizamos algunas consideraciones.

Analizamos que:

- Si la solución fuese minimizar el consumo, bastaría con no producir.
- Si la solución fuese aumentar la ganancia, bastaría con aumentar la ganancia incluso si se consume más material o si se produce con menor eficiencia.

Por lo que:

- Concluimos que el problema no solo busca minimizar el espacio utilizado al cortar las piezas grandes en barras pequeñas, pues bastaría cualquier configuración que maximice la cantidad de piezas producidas; sino que el problema debe considerar el consumir la menor cantidad de barras de material posible, cumpliendo con la demanda de la orden de producción.

Es así que la función objetivo se puede convertir a la forma de minimizar el desperdicio en el consumo de materiales necesario para cumplir con la demanda de producción de una orden de fabricación:

$$\text{Función Objetivo : } \min \sum_0^n \text{ Sobras Material}$$

Donde:

- N: Es el número de piezas de material disponibles. En este caso, barras de acero.
- Y considerando las restricciones del problema

## 4.5 Conclusiones

**Conclusión:** Tras analizar la naturaleza del problema de Cutting stock, concluimos que la función objetivo, las variables y las restricciones aquí definidas representan fielmente el problema abordado.

# Capítulo 5. Estructuras de Datos

## 5.1 Introducción

En este capítulo se presentan el **Resultado Esperado 2 (RE2)** y el **Resultado Esperado 4 (RE4)**, que consiste en las estructuras de datos que son utilizadas por el algoritmo memético (RE2) y el algoritmo genético (RE4).

Como existen estructuras que son usadas por ambos enfoques en los cálculos que realizan para encontrar la solución al problema, este capítulo se separa en dos partes: La estructura de la solución (cromosoma y datos modelados) y Las estructuras de datos que utilizan los algoritmos para sus cálculos.

## 5.2 Cromosoma de la Solución

Para realizar una comparación de ambos métodos, se plantea evaluar su eficacia y eficiencia sobre la misma base de información y misma representación: Mismos datos de entrada y misma representación de la solución (cromosoma).

**Esta es la forma del cromosoma:** Un arreglo unidimensional de enteros, que referencian el identificador del producto que se va a producir. De esa forma, el entero uno (1) referencia al producto P001, el entero dos (2) referencia al producto P002 y así sucesivamente.

| Pos1 | Pos2 | Pos3 | .. | .. | .. | PosN |
|------|------|------|----|----|----|------|
| 1    | 2    | 4    | .. | .. | .. | 3    |

Por lo tanto, se representa a una solución válida con la forma de un arreglo unidimensional de tamaño N, cuyas columnas contienen todas las barras que serán producidas según la solución, para cubrir el pedido de producción.

La relación consiste en que la posición del índice de una barra de producto final declara que dicho producto se está elaborando usando la primera barra de material disponible a la que le quede material suficiente.

De esta forma, por ejemplo:



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 5 | 6 | 7 | 3 | 8 |
|---|---|---|---|---|---|---|---|

- Considerando los tamaños: 1m, 2m, 3m, 4m, 5m, 6m, 7m, 8m para los productos 1, 2, 3, 4, 5, 6, 7 y 8, en ese orden.
- Considerando los largos: 20m por barra de material

Define que las barras de producto 1, 2, 4, 5 y 6 se producen en la primera barra de material disponible (pues sus largos suman 18 y el producto 7 no podría producirse en la misma barra de material) y las barras de producto 7, 3 y 8 se producen en la barra de material siguiente (sus largos suman 18).

Esto se calcula debido a que los tamaños de las barras de material son estándar y es indiferente el orden en que se asignan a los productos a producir. Para saber cuándo cambiar a la siguiente barra de material, basta con sumar los largos de los productos y comparar la suma con el largo del material (como se ha hecho en el ejemplo).

### 5.3 Estructuras de datos

Estas estructuras se utilizan en ambos algoritmos para modelar los datos del problema y realizar cálculos sobre ellas.

- **Orden de Pedido:** Representa el requerimiento consolidado de los clientes que desean comprar barras de acero. Contiene un identificador, las cantidades a producir de cada producto y una referencia al material disponible para asignar para la producción.

| Identificador | Líneas de Pedido        | Material disponible |
|---------------|-------------------------|---------------------|
| O004          | {L001,L002,L003,...,Ln} | M001                |

- **Línea de Pedido:** Representa el detalle de la orden de pedido, en la forma de cada línea del pedido. Cada línea contiene un identificador, el producto que se requiere producir y la cantidad de unidades que se requieren producir.

| Identificador | Producto | Cantidad |
|---------------|----------|----------|
|---------------|----------|----------|

|      |      |    |
|------|------|----|
| L001 | P001 | 10 |
|------|------|----|

- **Producto:** Representa un tipo único de producto que se va a producir. Lo que se conoce como SKU. En el problema, un producto es un tipo de barra de acero final y se representa por un identificador y el largo de la barra en metros.

| Identificador | Largo (m2) |
|---------------|------------|
| P001          | 1.2        |

- **Material Disponible:** Representa el material disponible para la producción de las barras de acero finales. Contiene un identificador, el largo estándar del material disponible y la cantidad de unidades de material disponible. En el problema, el material son las barras estándar grandes desde las que se producen barras de acero final.

| Identificador | Largo Unitario (m2) | Cantidad Disponible |
|---------------|---------------------|---------------------|
| M001          | 100                 | 20                  |

- **Población:** Representa el universo de soluciones al problema que están siendo calculadas. Contiene referencias a las soluciones (los cromosomas) y meta-data como la mejor solución actual y las cantidades de generaciones.

| Mejor Solución | Soluciones         | Cantidad de Generaciones Actual | Cantidad de Generaciones Máxima |
|----------------|--------------------|---------------------------------|---------------------------------|
| C001           | {C001,C002,...,Cn} | 20                              | 40                              |

Además, para la presentación de los resultados por generación y del algoritmo, hace uso de una estructura auxiliar que guarda la información.

- **Resultados:** Representa un índice de las mejores soluciones alcanzadas en cada generación y en el algoritmo en total. La mejor solución inicial, el promedio de la

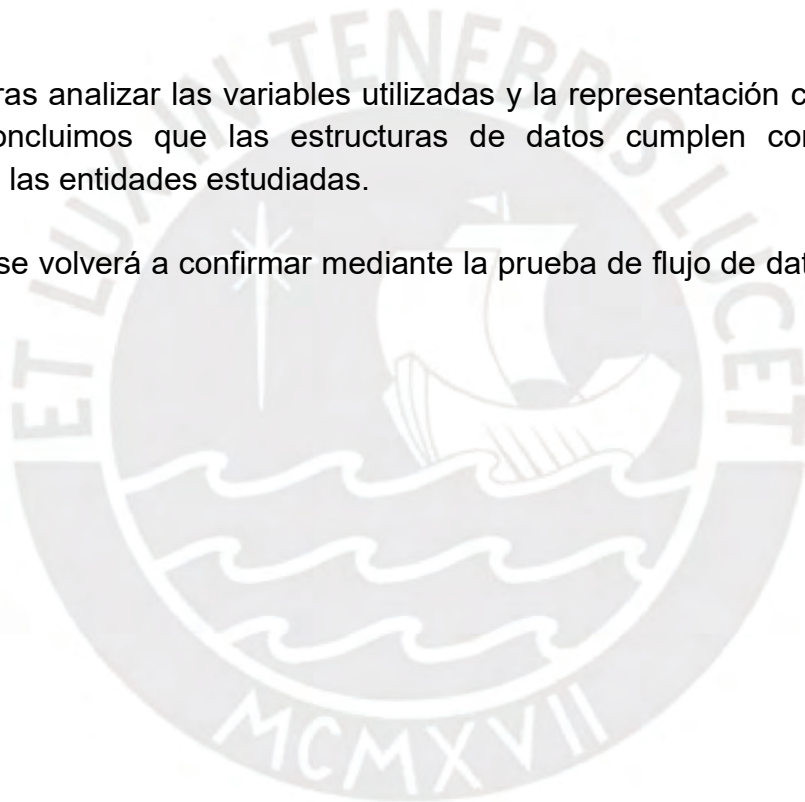
población inicial, la mejor solución de cada generación, el promedio de cada generación y las soluciones de la última generación.

| Mejor Inicial | Promedio Inicial | Puntajes              | Promedios             | Soluciones                     |
|---------------|------------------|-----------------------|-----------------------|--------------------------------|
| 0.80          | 0.75             | {0.80, 0.82, ..., Pn} | {0.75, 0.78, ..., Pn} | {Crom001, Crom002, ..., Cromn} |

## 5.4 Conclusiones

**Conclusión:** Tras analizar las variables utilizadas y la representación cromosómica de la solución, concluimos que las estructuras de datos cumplen con expresar las propiedades de las entidades estudiadas.

Esto, además, se volverá a confirmar mediante la prueba de flujo de datos en capítulos posteriores.



# Capítulo 6. Pseudocódigos de los Algoritmos

## 6.1 Introducción

En este capítulo se presentan el **Resultado Esperado 3 (RE3)** y el **Resultado Esperado 5 (RE5)**, que consisten en el pseudocódigo correspondiente a la solución propuesta, el algoritmo memético, (RE3) y el pseudocódigo correspondiente a la adaptación de una solución de la literatura al caso particular de estudio, el algoritmo genético, (RE5)

Debido a que el algoritmo memético propuesto se basa en el algoritmo genético, este capítulo se separa en dos partes: El pseudocódigo del algoritmo genético y una explicación de sus funcionalidades y de cómo se adapta al problema, y El pseudocódigo del algoritmo memético explicando las mejoras que se han realizado a partir de la solución adaptada y el diseño seguido.

## 6.2 Diseño del Algoritmo Genético

En esta sección se presenta el pseudocódigo del algoritmo, incluyendo la función principal y las funciones de los operadores genéticos.

Código: Algoritmo Genético

|    |  |
|----|--|
|    | <b>Inicio Algoritmo Genético {</b>                                       |
| 1. | srand(time(NULL));   |
| 2. | Poblacion poblacion = Poblacion();                                       |
| 3. | poblacion.generalInicial();  |
| 4. | Evolucion evolucion = Evolucion(poblacion);                              |
| 5. | cantidadGeneraciones = 0;  |
| 6. | <b>Mientras</b> (cantidadGeneraciones <= poblacion.maximoGeneraciones) { |
| a. | vector<Solucion> seleccionados = evolucion.seleccion();                  |
| b. | vector<Solucion> nuevosHijos = evolucion.crossover(seleccionados);       |
| c. | evolucion.mutacion(nuevosHijos);   |
| d. | evolucion.agregarHijos(poblacion, nuevosHijos);                          |
| e. | evolucion.elitismo();  |
| f. | cantidadGeneraciones = cantidadGeneraciones + 1;                         |
|    | }  |
| 7. | poblacion.imprimeMejorSolucion();  |
|    | <b>Fin</b>   |

1. Inicializa semilla de los números random
2. Inicializa objeto población
3. Llena población con datos de entrada
4. Inicializa objeto Evolución, que realiza operaciones de selección, crossover y elitismo.
5. Inicializa cantidad de generaciones en 0.
6. Mientras no se supere el límite de generaciones
  - a. Se seleccionan los padres de la nueva generación.
  - b. Se generan nuevos hijos con crossover.
  - c. Se muta a un porcentaje de los nuevos hijos.
  - d. Se agregan los nuevos hijos a la población.
  - e. Se borran a los menos preparados.
  - f. Se aumenta el número de la generación.
7. Se muestra la mejor solución.

### 6.2.1 Operador de Selección

Se utiliza el método de ruleta con probabilidades proporcionales a la función fitness y con tiradas dispersas. A mayor función fitness, más probabilidad de ser seleccionada para luego hacer el crossover. Además, se escogen números separados en la ruleta.

Código: Operador de Selección

|  |   |
|--|---|
| <ol style="list-style-type: none"> <li>1.</li> <li>2.</li> <li>3.</li> <li>4.</li> <li>5.</li> <li>6.</li> <li>a.</li> <li>b.</li> <li>7.</li> <li>8.</li> <li>a.</li> <li>b.</li> <li>c.</li> <li>i.</li> <li>1.</li> <li>2.</li> </ol> | <pre> <b>Inicio Selección {</b> vector&lt;Solucion&gt; seleccionados; tamSoluciones = poblacion.soluciones.size(); cantSeleccionados = 20; probabilidades = new float[tamSoluciones]; fAcumulada = 0; <b>Repetir</b> (i = 0; i &lt; tamSoluciones; i++) {     fAcumulada = fAcumulada + poblacion.soluciones.at(i).puntaje;     probabilidades[i] = fAcumulada; } rx = fAcumulada / cantSeleccionados; <b>Repetir</b> (i = 0; i &lt; cantSeleccionados; i++) {     a. randomR =         rx * i + static_cast&lt;float&gt; (rand()) / (static_cast&lt;float&gt; (RAND_MAX         / rx));     b. found = false;     c. <b>Repetir</b> (j = 0; j &lt; sizeSoluciones &amp;&amp; found == false; j++) {         i. if (randomR &lt; probabilidades[j]) {             1. seleccionados.push_back(poblacion.soluciones.at(j));             2. found = true; </pre> |
|--|---|

|   |   |
|---|---|
| 9 | <pre>         }     } } return seleccionados; } <b>Fin</b> </pre> |
|---|---|

1. Crea lista vacía de seleccionados.
2. Copia el tamaño del vector de soluciones.
3. La cantidad de seleccionados a seleccionar es 20.
4. Crea un arreglo donde están las probabilidades acumuladas de ser seleccionados.
5. Inicializa el máximo de la probabilidad acumulada.
6. Para cada solución
  - a. Setea su probabilidad de ser seleccionada
  - b. Aumenta el máximo de la probabilidad de acumulada.
7. Calcula el diferencial de la probabilidad.
8. Para cada tirada de la ruleta
  - a. Se escoge un random en el rango
  - b. Inicializa un bool que indica si ya se encontró la solución
  - c. Para cada solución
    - i. Se compara si está en el rango
      1. Se selecciona
      2. Se cambia el bool a verdadero.
9. Se retornan los seleccionados.

### 6.2.2 Operador de Crossover

El operador de crossover o cruce que se usa en el algoritmo escoge aleatoriamente un punto en el cual intercambiar los genes de los dos cromosomas. El primer hijo tendrá los genes de la primera solución hasta el punto aleatorio y de ahí hasta el final de la segunda solución y viceversa para el segundo hijo.

Código: Operador de Crossover

|                  |   |
|------------------|---|
| 1<br>2<br>3<br>4 | <pre> <b>Inicio Crossover (vector &lt;Solucion&gt; seleccionados) {</b>     vector&lt;Solucion&gt; nuevosHijos;     cantidadCrossover = seleccionados.size();     randIndex = rand() % cantidadBarrasPedido;     <b>Repetir</b> (int i = 0; i &lt; cantidadCrossover; i++) { </pre> |
|------------------|---|

|   |  |
|---|--|
| a | Solucion solucion1(seleccionados.at(i), seleccionados.at(i + 1), randIndex); |
| b | Solucion solucion2(seleccionados.at(i + 1), seleccionados.at(i), randIndex); |
| c | nuevosHijos.push_back(solucion1);  |
| d | nuevosHijos.push_back(solucion2);  |
| e | i = i + 1;   |
| 5 | }  |
|   | <b>return</b> nuevosHijos;   |
|   | }  |
|   | <b>Fin</b>   |

1. Inicializa vector vacío de nuevos hijos
2. Se copia el tamaño de los seleccionados.
3. Se escoge un número aleatorio para el crossover
4. Para cada seleccionado para el crossover
  - a. Se crea un hijo con la primera mitad de la primera solución y la segunda mitad de la segunda.
  - b. Se crea un hijo con la primera mitad de la segunda solución y la segunda mitad de la primera.
  - c. Agrega el hijo a la lista de hijos nuevos.
  - d. Agrega el hijo a la lista de hijos nuevos.
  - e. Aumenta el número de crossover
5. Retorna la lista de hijos nuevos.

### 6.2.3 Operador de Mutación

El operador de mutación calcula probabilidad de mutación de cada gen en la solución y si muta, se intercambia de lugar con cualquier otra gene.

Código: Operador de Mutación

|    |   |
|----|---|
|    | <b>Inicio Mutacion (vector&lt;Solucion&gt; nuevosHijos) {</b> |
| 1  | tamNuevosHijos = nuevosHijos.size();                          |
| 2  | cantMutaciones = 0;   |
| 3  | <b>Repetir</b> (int k = 0; k < tamNuevosHijos; k++) {         |
| a  | <b>Repetir</b> (int i = 0; i < cantidadBarrasPedido; i++) {   |
| i  | int r = rand() % 20;  |
| li | if (r == 1) {   |
| 1  | cantMutaciones = cantMutaciones + 1;                          |
| 2  | r2 = rand() % cantidadBarrasPedido;                           |
| 3  | cromosomaR2 = nuevosHijos.at(k).cromosoma[r2];                |
| 4  | nuevosHijos.at(k).cromosoma[r2] =                             |
|    | nuevosHijos.at(k).cromosoma[i];                               |

|        |   |
|--------|---|
| 5<br>6 | <pre> nuevosHijos.at(k).cromosoma[i] = cromosomaR2; } } } } } } } <b>Fin</b> </pre> |
|--------|---|

1. Copia el tamaño de los nuevos hijos.
2. Inicializa la cantidad de mutaciones en 0.
3. Para cada nuevo hijo
  - a. Para cada gen
    - i. Calcula la probabilidad
    - ii. Si acierta
      1. Aumenta la cantidad de mutaciones
      2. Calcula cual gen intercambiar
      3. Cambia el gen
      4. Cambia el gen
      5. Cambia el gen

#### 6.2.4 Operador de Elitismo

Es el operador usado para aumentar la calidad de la población en general. Selecciona a los peores 10 soluciones y la borra.

Código: Operador de Elitismo

|          |  |
|----------|--|
| 1.<br>2. | <pre> <b>Inicio Elitismo(Poblacion p) {</b>     sort(p.soluciones.begin(), p.soluciones.end(), funcCompara);     p.soluciones.erase(p.soluciones.end() - 9, p.soluciones.end()); } <b>Fin</b> </pre> |
|----------|--|

1. Ordena las soluciones por puntaje de la función fitness
2. Borra las soluciones con menos fitness.

### 6.3 Diseño del Algoritmo Memético

En esta sección se presenta el pseudocódigo del algoritmo memético, incluyendo la función principal y la función del operador de búsqueda local.



Para la realización del algoritmo memético, se usa como base el algoritmo genético y se agrega una función de búsqueda local y la calibración propia del algoritmo. Por lo tanto, aquí se presenta el pseudocódigo de la función de búsqueda local

### 6.3.1 Operador de Búsqueda Local

El operador de búsqueda local mejora la solución completando los espacios vacíos de las soluciones parciales con secciones pequeñas de otras soluciones parciales. Se utiliza una tasa de voracidad ( $\alpha$ ) para escoger si una sección es elegida para completar alguna solución parcial o se ignora y se continúa la búsqueda de otra posible sección.

Código: Operador de Búsqueda Local

|     |  |
|-----|--|
|     | <b>Inicio BusquedaLocal(Solucion s) {</b>                  |
| 1.  | vector<tuple<int*, int, float, float>> bloques;            |
| 2.  | bloques.clear();   |
| 3.  | Repetir (int i = 0; i < this->cantidadCromosomas; i++) {   |
| a   | Si (actualLargoMaterial - sumaLargos >= largoBarra) {      |
| i   | sumaLargos = sumaLargos + largoBarra;                      |
| ii  | cantidadGenes++;   |
|     | }  |
| b   | Sino {   |
| i   | bloques.agregaBloque();                                    |
| ii  | cantidadGenes = 1;   |
| iii | actLargoMaterial = largoMaterial[i+1];                     |
|     | }  |
| c   | sumaLargos = sumaLargos + largoBarra;                      |
|     | }  |
|     | }  |
| 4.  | if (cantidadCromosomas > totalGenes) {                     |
| d   | bloques.agregaBloques();                                   |
|     | }  |
| 5.  | nuevoCromosoma = new int[cantidadCromosomas];              |
| 6.  | Para (int i = 0; i < cantidadBloques; i++) {               |
| e   | Si (tamañoBloque >= 0) {                                   |
| i   | llena(nuevoCromosoma, bloque);                             |
| ii  | Para (int j = i+1; j < cantidadCromosomasDelBloque; j++) { |
| iii | Si (residuoBloque > tamañoCromosoma)                       |
| 1   | int r = rand() % 100;                                      |
| 2   | if (r <= alpha) {  |

|    |                                      |
|----|--------------------------------------|
| a  | llena(nuevoCromosoma, bloqueActual); |
| 7. | }                                    |
| 8. | } cromosoma = nuevoCromosoma;        |
| 8. | } <b>Fin</b>                         |

3. Crea un arreglo de bloques
4. Vacío.
5. Para cada cromosoma
  - a. Si entra en el material actual, sigue
  - b. Sino, se agregan los cromosomas desde el último hasta el actual al arreglo de bloques
6. Se agregan los cromosomas
7. Se crea un buffer de cromosoma
8. Para cada bloque del arreglo
  - a. Se llena el cromosoma buffer
  - b. Si hay espacio suficiente para un cromosoma de otra solución parcial
    - i. Se calcula con la tasa de voracidad (alpha) si se llena o se ignora.

## 6.4 Conclusiones

Tras analizar el pseudocódigo y cómo se integran las partes, cómo modifican la data y cómo producen resultados, concluimos que el pseudocódigo de los algoritmos resuelve el caso de estudio.

Esto, además, se volverá a confirmar mediante la prueba de flujo de datos en capítulos posteriores.

## Capítulo 7. Prueba de Verificación

En este capítulo se presenta la verificación funcional de los algoritmos de tal forma que se cumple con los Indicadores Objetivamente Verificables para los resultados **R1** a **R5**.

La forma en que se presentan las verificaciones es mediante la simulación de un flujo de datos regular, que pasará a través de las etapas de las soluciones, y donde se observará objetivamente que se cumple con los indicadores propuestos.

Aquí una explicación de las pruebas verificadas y el resultado que cubren.

| Resultado Obtenido  | Indicador Objetivamente Verificable   | Modo de verificación  |
|---|---|---|
| R1. Definición de la función objetivo, las variables y las restricciones. | Prueba funcionales con muestras pequeñas  | El flujo presentado aplica la función objetivo, realiza cálculos en las variables y funciona dentro de las restricciones del caso   |
| R2 y R4. Estructuras de datos   | Validar que las estructuras soporten la representación de los datos y la solución | El flujo presentado soporta representar los datos de entrada en las estructuras de datos, mantiene la coherencia en la solución (el cromosoma) y convierte las soluciones en los datos de salida. |
| R3 y R5. Pseudocódigo   | Validar que el pseudocódigo aplique los algoritmos                                | El flujo presentado aplica los operadores evolutivos de los algoritmos y soporta las estructuras y las definiciones anteriores.   |

Tabla 7: Verificación de Indicadores (Autoría Propia)

## 7.1 Flujo de Validación funcional

### 7.1.1 Datos de entrada

El primer paso en el flujo es representar los datos de entrada en forma de archivos en una estructura inicial interna de datos que luego será convertida en la población inicial de soluciones.

Aquí presentamos una muestra de datos de entrada.

|   |  |
|---|--|
| <p>Pedido:<br/>2 x Barra Tipo 1<br/>1 x Barra Tipo 2<br/>1 x Barra Tipo 4<br/>Materiales:<br/>1 x Material Tipo 1<br/>2 x Material Tipo 2</p> | <p>Catalogo:<br/>Barra Tipo 1 5m<br/>Barra Tipo 2 6m<br/>Barra Tipo 3 3m<br/>Barra Tipo 4 2m</p> |
|   | <p>Catalogo Materiales:<br/>Material Tipo 1 10m<br/>Material Tipo 2 9m</p>                       |

Imagen 23: Datos de Entrada (Autoría Propia)

Esta muestra representa los datos de entrada en forma de archivos de texto, que serán convertidos a datos internos:

- Cada línea del archivo de productos del pedido: Se convierte un índice en un arreglo múltiple que lista los productos a ser producidos. En este caso, se obtiene un arreglo con el índice 1 repetido 2 veces, el índice 2 repetido 1 vez y el índice 4 repetido 1 vez.
- Cada línea del archivo del catálogo (de productos): Se convierte en un valor en un arreglo múltiple que lista los tamaños de los productos disponibles para la producción. En este caso, se obtiene un arreglo con los valores 5, 6, 3 y 2, representando los largos de las barras de tipo 1, 2, 3 y 4.
- Cada línea del archivo de materiales del pedido: Se convierte en un índice de un arreglo múltiple que lista los materiales disponibles para la producción del pedido. En este caso, se obtiene un arreglo con el índice 1 repetido 1 vez y el índice 2 repetido 2 veces.

- Cada línea del archivo de catálogo de materiales: Se convierte en un valor en un arreglo múltiple que lista los tamaños de los materiales disponibles para la producción. En este caso, se obtiene un arreglo con los valores 10 y 9, representando los largos 10 y 9.

Aquí la representación interna de los datos:

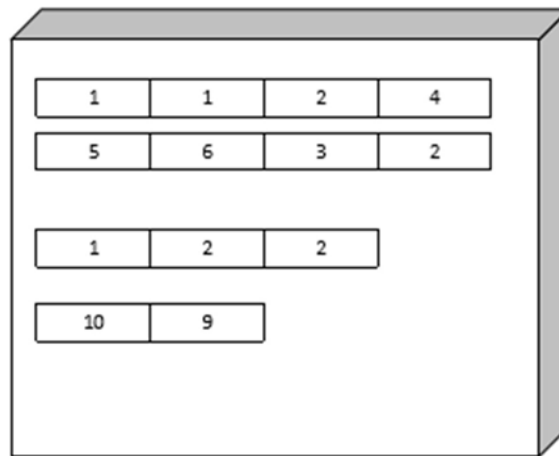


Imagen 24: Representación interna de datos (Autoría Propia)

Estos datos son utilizados en los cálculos de la función fitness de los algoritmos, con la cual se evalúa si una solución es viable y cuanto lo es.

Además, con la cantidad de productos pedidos se genera el cromosoma que representa la solución. En este caso, la cantidad de cantidad de productos pedidos es 4.

### 7.1.2 Representación de la Solución

A partir de la representación interna de los datos, se generan soluciones iniciales que son parte de la población inicial de los algoritmos.

El método que se utiliza es de generación aleatoria de permutaciones de los índices de productos a producir. De esta forma, los 4 productos pedidos ocasionan que se generen cromosomas (representaciones de la solución para los algoritmos) de largo 4 y cuyo contenido son las permutaciones de los índices.

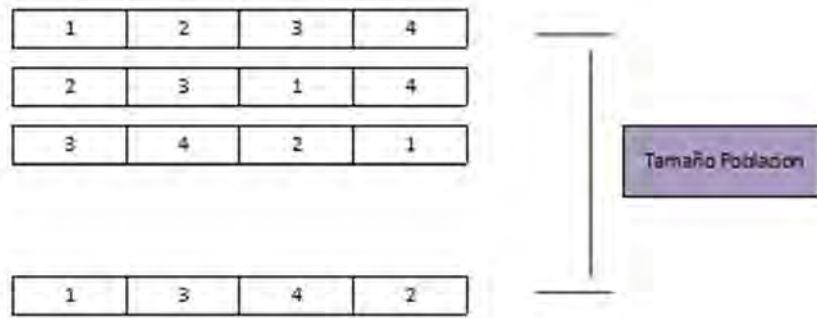


Imagen 25: Población Inicial (Autoría Propia)

Esta población inicial de soluciones será evolucionada por los operadores evolutivos tanto en el algoritmo adaptado (genético) como en el propuesto (memético). Los operadores modificarán los cromosomas de diferentes formas y los resultados serán evaluados por la función fitness para saber cuáles soluciones son mejores.

### 7.1.3 Función Fitness

Para conocer la optimalidad de las soluciones generadas, se calcula la función fitness a partir de la representación cromosómica.

El cálculo se realiza restando el largo de las barras a producir del largo de las barras disponibles de material. Aquí la muestra con 4 soluciones en forma de cromosoma.

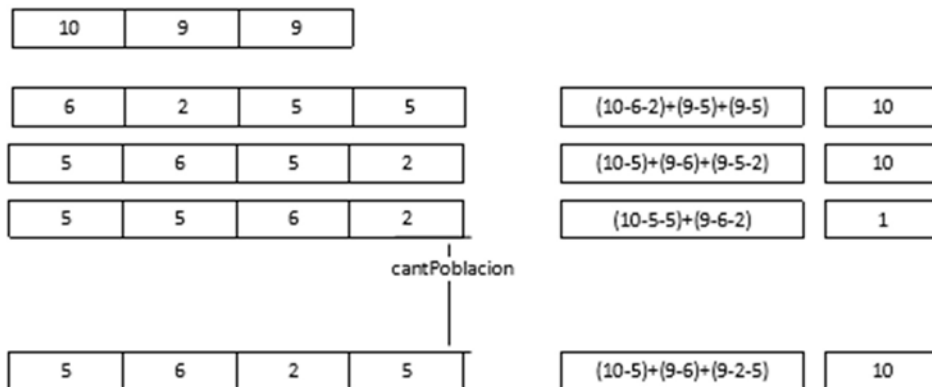


Imagen 26: Fitness Inicial (Autoría Propia)

Como se observa, para calcular las funciones fitness de las 4 soluciones (fitness 10, 10, 1 y 10), se han restado los largos de las barras a producir, obtenidos de información

sacada de los arreglos de largos de productos (6, 2, 5, 5; 5, 6, 5, 2; 5, 5, 6, 2; 5, 6, 2, 5), de los largos de las barras de material disponible para la producción (10, 9, 9).

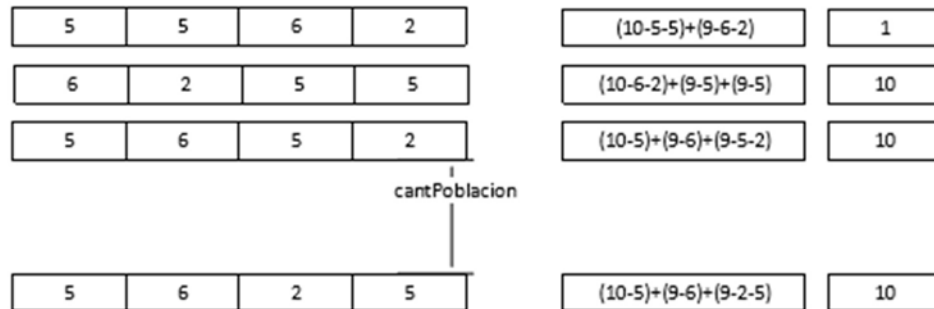


Imagen 27: Población inicial en orden de función fitness (Autoría Propia)

Finalmente, se ordenan las soluciones en orden de función fitness.

Con esta población inicial evaluada, se opera con los algoritmos. En caso del algoritmo genético se usan los operadores de selección y crossover y de mutación, y en el algoritmo memético se utiliza además el operador de búsqueda local

#### 7.1.4 Operador de selección y Operador de crossover

Para realizar la evolución de la población, se utiliza el cruce de soluciones mediante el operador de crossover, que mezcla dos soluciones y genera hijos similares. Para saber qué soluciones son escogidas para el crossover, se utiliza un método de ruleta estocástica que da mayor probabilidad de selección a las soluciones con mejor función fitness.

Aquí la selección aplicada a la muestra.

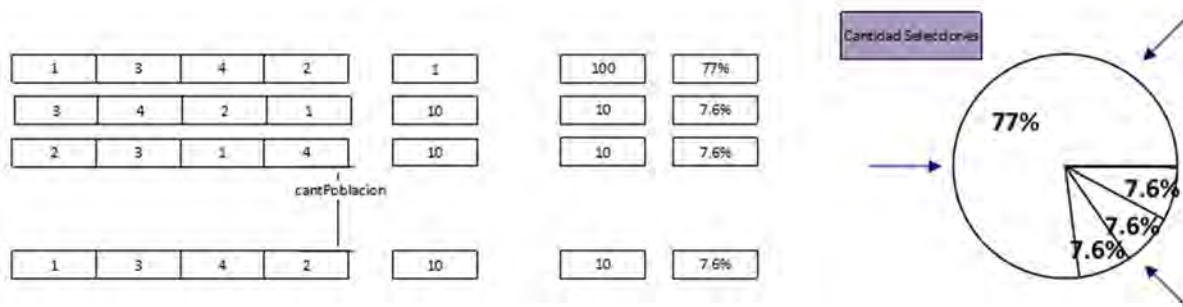


Imagen 28: Selección por ruleta (Autoría Propia)

Como se observa, se asigna una probabilidad (77%, 7,6%) a cada solución en base a su función fitness. Luego, se seleccionan las soluciones en base a la ruleta y la cantidad de selecciones a realizar.

En este caso, se realizan 3 selecciones. Dos de ellas se realizan a la solución con fitness 1 y 1 vez se realiza a una solución con fitness 10.

Estas 3 soluciones elegidas son mejoradas por el operador de crossover para generar 6 nuevas soluciones. El método usado es de punto único de crossover (Single Point Crossover), que escoge al azar un punto de los cromosomas y genera dos hijos: Uno con la cabeza de la primera solución y la cola de la segunda, y viceversa.

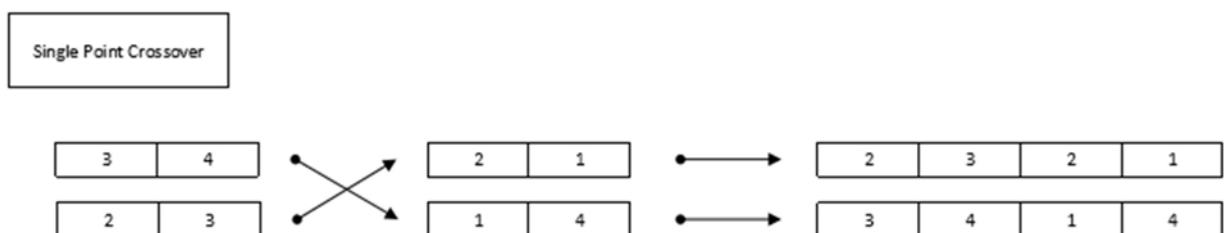


Imagen 29: Crossover (Autoría Propia)

Como se observa, cada solución nueva tiene la mitad de cada solución original.

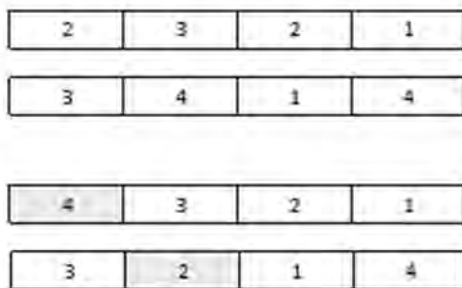


Imagen 30: Corrección aleatoria (Autoría Propia)

En caso de repeticiones en los índices, se corrigen aleatoriamente. Finalmente, estas nuevas soluciones son mutadas por el operador de mutación para evitar convergencia.

### 7.1.5 Operador de Mutación

Lo que hace este operador es cambiar los genes (valores) del cromosoma de manera aleatoria y siguiendo la probabilidad de mutación (tasa de mutación) para evitar convergencia de soluciones.



Se escoge un gen aleatorio y se intercambia con otro. En este caso, se intercambi6 el primer

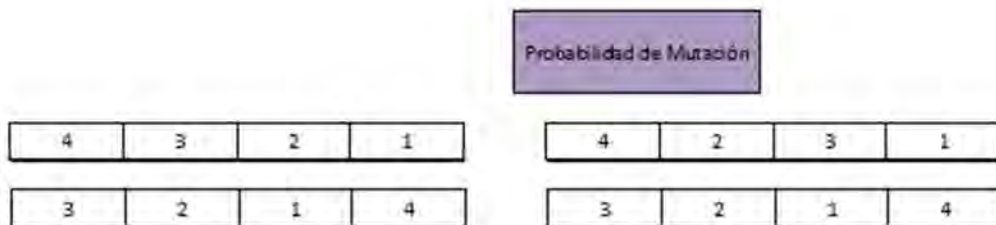


Imagen 31: Mutaci6n (Autoría Propia)

Tras la mutaci6n, el primer cromosoma (4, 3, 2, 1) se convirti6 en una variante (4, 2, 3, 1).

Finalmente, en el caso del algoritmo gen6tico, se evalúan las soluciones, se conservan las mejores y se reinicia el ciclo de operaciones. En el caso del algoritmo mem6tico, se utiliza un operador m6s: el Operador de Búsqueda Local.

### 7.1.6 Operador de Búsqueda Local

El operador de b6squeda local tiene como objetivo mejorar una soluci6n existente, basándose en su propia composici6n. Sin combinaci6n con otras soluciones. En este caso, el operador particiona la soluci6n en bloques de productos y espacios vacíos en un material.



Imagen 32: Búsqueda local 1 (Autoría Propia)

En la soluci6n presentada, por ejemplo, encuentra que para la primera barra de material (de largo 10), existen 2 productos asignados actualmente (de largo 2 y 5), por lo que se reconoce un bloque de residuo 3 (representado por un -3), debido a que  $10 - 2 - 5 = 3$ . Se realiza lo mismo con todos los cromosomas.

Luego, a cada bloque con valor negativo se le busca un producto de menor tamaño. En el caso de la segunda solución (6, -4, 5, -4, 5, 2, -2), escoge el valor -4 y se busca un valor positivo menor que 4. En este caso, 2.

|   |    |    |    |    |   |    |    |
|---|----|----|----|----|---|----|----|
| 5 | -4 | 5  | -4 | 5  | 2 | -2 | 10 |
| 6 | 2  | -2 | 5  | -4 | 5 | -4 | 10 |

Imagen 33: Búsqueda local 2(Autoría Propia)

Y se coloca en la nueva posición. Esto se repite considerando el valor alpha de voracidad, debido a que antes de realizar el intercambio, se calcula la probabilidad de realizarse según la tasa de voracidad.

Finalmente, al igual que con el algoritmo genético, se repite el ciclo de evolución.

## 7.2 Conclusiones

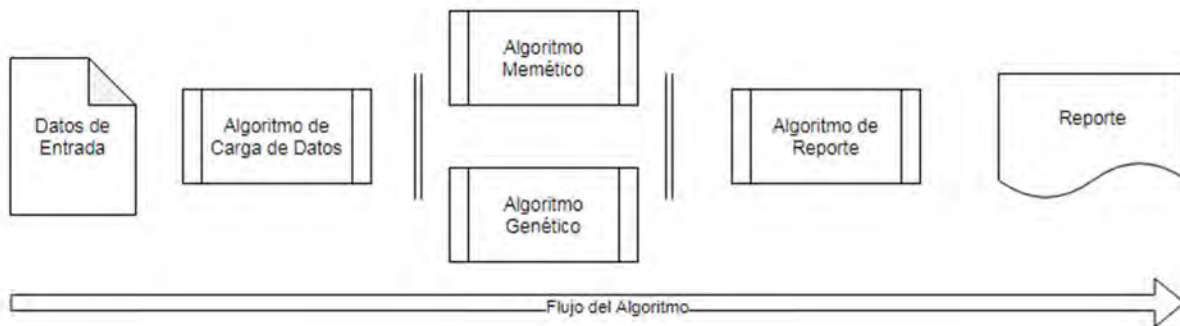
**Conclusión:** Tras realizar la prueba de flujo de datos, podemos confirmar que se cumple con la representación de los datos y el problema, con demostrar que las estructuras de datos soportan los operadores evolutivos, manteniendo la coherencia, y que el algoritmo (en forma de pseudocódigo y en forma ejecutada) genera resultados correctos.

# Capítulo 8. Módulo de los Algoritmos

## 8.1 Introducción

En este capítulo se presentan el **Resultado Esperado 7 (R7)** y el **Resultado Esperado 8 (R8)**: Implementación del algoritmo propuesto y la Implementación del algoritmo adaptado.

La presentación se realiza mediante un medio digital, sin embargo, aquí se presenta un diagrama con el flujo del algoritmo y los módulos desarrollados y, luego, las pruebas unitarias.



Flujo 1: Flujo del Programa (Autoría Propia)

El flujo regular del programa consiste en la entrada de datos por medio de archivos con la información de los pedidos, productos y materiales disponibles, que es formateado por el módulo de carga de datos. Luego, a elección, se calcula la solución al problema con el algoritmo memético o genético, y los resultados son formateados por el módulo de reporte. Finalmente, se muestra un reporte. Los módulos han sido desarrollados en Java, con la librería gráfica Swing y en el entorno de desarrollo Apache Netbeans IDE.

## 8.2 Pruebas Unitarias

En este capítulo además se presentan los casos de uso que sirven para verificar el funcionamiento de la implementación.

## 8.2.1 Entrada de Datos

| PU-001                 |  |
|------------------------|--|
| Nombre del caso de uso | Carga de datos   |
| Fase del algoritmo     | Entrada de Datos - Carga de archivos   |
| Entrada                | Ruta de los archivos a cargar  |
| Descripción            | La función comprueba que existen los archivos, los abre y crea las estructuras internas de materiales y pedidos.   |
| Salida Esperada        | Lista de materiales disponibles y Lista de pedidos   |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se presiona el botón elegir, de la pantalla de carga de archivos.</li> <li>2. Se selecciona el archivo a cargar.</li> <li>3. Se presiona el botón cargar datos.</li> </ol> |
| Resultado Obtenido     | Se cargaron los pedidos y se generaron los archivos de pedidos y de materiales.  |

|                        |  |
|------------------------|--|
| Nombre del caso de uso | Creación de la población inicial   |
| Fase del algoritmo     | Entrada de datos - Población inicial   |
| Entrada                | Lista de pedidos y de materiales   |
| Descripción            | La función genera una población inicial en base a los pedidos de producción.   |
| Salida Esperada        | Población inicial  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se llama a la función después de generar las listas de pedidos y de materiales.</li> <li>2. El algoritmo crea una población inicial aleatoria que corresponda a la cantidad de pedidos.</li> </ol> |
| Resultado Obtenido     | Se ejecutó la función y se generó la población inicial.  |

## 8.2.2 Algoritmo Genético

| PU-002                 |   |
|------------------------|---|
| Nombre del caso de uso | Ejecución del Algoritmo Genético  |
| Fase del algoritmo     | Algoritmo genético - Ejecución  |
| Entrada                | Lista de pedidos y de materiales, y la Población Inicial  |
| Descripción            | La función ejecuta el algoritmo genético, durante varias generaciones y devuelve una solución.  |
| Salida Esperada        | Estructura de resultados  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se presiona el botón Algoritmo Genético, de la pantalla de selección de algoritmo.</li> <li>2. Se muestra una pantalla de progreso del algoritmo.</li> <li>3. Se presiona el botón volver o el botón Reporte..</li> </ol> |
| Resultado Obtenido     | Se ejecutó el algoritmo y se generó la solución. Esta solución puede mostrarse con el botón Reporte.  |

| PU-003                 |   |
|------------------------|---|
| Nombre del caso de uso | Ejecución del Algoritmo Genético  |
| Fase del algoritmo     | Algoritmo genético - Selección de padres  |
| Entrada                | Población actual de soluciones  |
| Descripción            | Se escoge un subgrupo de la población que será usado en las fases evolutivas.   |
| Salida Esperada        | Subgrupo seleccionado   |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se llama a la función después de generar la población actual.</li> <li>2. El algoritmo elige mediante ruleta universal estocástica, un subgrupo de soluciones.</li> </ol> |
| Resultado Obtenido     | Se ejecutó la función y se seleccionó un subgrupo de soluciones.  |

| PU-004                 |   |
|------------------------|---|
| Nombre del caso de uso | Ejecución del Algoritmo Genético  |
| Fase del algoritmo     | Algoritmo genético - Cruce de padres  |
| Entrada                | Subgrupo de soluciones  |
| Descripción            | Se generan nuevo hijos en base al subgrupo de soluciones seleccionadas.   |
| Salida Esperada        | Conjunto de nuevos hijos  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se llama a la función después de seleccionar a los padres.</li> <li>2. El algoritmo genera hijos a partir del método de cruce de punto de cruce.</li> </ol> |
| Resultado Obtenido     | Se ejecutó la función y se generó un conjunto de nuevos hijos.  |

| PU-005                 |   |
|------------------------|---|
| Nombre del caso de uso | Ejecución del Algoritmo Genético  |
| Fase del algoritmo     | Algoritmo genético - Mutación   |
| Entrada                | Conjunto de nuevos hijos  |
| Descripción            | Se modifican algunos hijos en base al algoritmo de mutación para obtener soluciones nuevas a la población.  |
| Salida Esperada        | Conjunto de nuevos hijos  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se llama a la función después de generación de los nuevos hijos.</li> <li>2. El algoritmo modifica algunos hijos a partir del método de intercambio.</li> </ol> |
| Resultado Obtenido     | Se ejecutó la función y se mutó algunos hijos.  |

| PU-006                 |                                  |
|------------------------|----------------------------------|
| Nombre del caso de uso | Ejecución del Algoritmo Genético |
| Fase del algoritmo     | Algoritmo genético - Elitismo    |

|                    |  |
|--------------------|--|
| Entrada            | Conjunto de nuevos hijos y Población   |
| Descripción        | Se agregan los nuevos hijos a la población, se ordena la población y los peores individuos son eliminados.   |
| Salida Esperada    | Nueva Población  |
| Pasos realizados   | <ol style="list-style-type: none"> <li>1. Se llama a la función después de mutar a los nuevos hijos.</li> <li>2. El algoritmo agrega los hijos a la población</li> <li>3. El algoritmo ordena las soluciones y elimina a los peores individuos.</li> </ol> |
| Resultado Obtenido | Se ejecutó la función y se generó la nueva población.  |

### 8.2.3 Algoritmo Memético

| PU-007                 |   |
|------------------------|---|
| Nombre del caso de uso | Ejecución del Algoritmo Memético  |
| Fase del algoritmo     | Algoritmo memético - Ejecución  |
| Entrada                | Lista de pedidos y de materiales, y la Población Inicial  |
| Descripción            | La función ejecuta el algoritmo memético, durante varias generaciones y devuelve una solución.  |
| Salida Esperada        | Estructura de resultados  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se presiona el botón Algoritmo Memético, de la pantalla de selección de algoritmo.</li> <li>2. Se muestra una pantalla de progreso del algoritmo.</li> <li>3. Se presiona el botón volver o el botón Reporte..</li> </ol> |
| Resultado Obtenido     | Se ejecutó el algoritmo y se generó la solución. Esta solución puede mostrarse con el botón Reporte.  |

| PU-008                 |                                     |
|------------------------|-------------------------------------|
| Nombre del caso de uso | Ejecución del Algoritmo Genético    |
| Fase del algoritmo     | Algoritmo genético - Búsqueda local |

|                    |  |
|--------------------|--|
| Entrada            | Conjunto de nuevos hijos, algunos modificados  |
| Descripción        | Se mejoran algunos hijos en base a la probabilidad alpha y el espacio de búsqueda.   |
| Salida Esperada    | Conjunto de nuevos hijos   |
| Pasos realizados   | <ol style="list-style-type: none"> <li>1. Se llama a la función después de la mutación de los nuevos hijos.</li> <li>2. El algoritmo mejora algunos hijos llenando espacios vacíos con soluciones parciales..</li> </ol> |
| Resultado Obtenido | Se ejecutó la función y se mejoró algunos hijos.   |

#### 8.2.4 Salida de Datos

| PU-009                 |   |
|------------------------|---|
| Nombre del caso de uso | Salida de datos   |
| Fase del algoritmo     | Salida de Datos - Mostrar el Reporte  |
| Entrada                | Estructura de soluciones (Resultados) obtenidos por el algoritmo  |
| Descripción            | La función muestra un diálogo con las pestañas de reporte. Las pestañas contienen: La solución en una tabla, un gráfico de las mejores soluciones, un gráfico de los promedios de las soluciones, y un gráfico de todas las soluciones de la última generación. |
| Salida Esperada        | Un diálogo conteniendo el reporte.  |
| Pasos realizados       | <ol style="list-style-type: none"> <li>1. Se presiona el botón elegir, de la pantalla de carga de archivos.</li> <li>2. Se selecciona el archivo a cargar.</li> <li>3. Se presiona el botón cargar datos.</li> </ol>  |
| Resultado Obtenido     | Se cargaron los pedidos y se generaron los archivos de pedidos y de materiales.   |



# Capítulo 9. Módulo de Entrada de Datos y Módulo de Salida de Datos

## 9.1 Introducción

En este capítulo se presentan el **Resultado Esperado 6 (RE6)** y el **Resultado Esperado 9 (RE9)**, que corresponden a la interfaz gráfica de la aplicación: Los módulos de entrada de datos, (RE6) y de salida de datos (RE9).

En este capítulo se presentan las interfaces gráficas mediante las cuales el usuario podrá seleccionar los datos de entrada para los algoritmos y podrá visualizar los resultados obtenidos de manera gráfica.



Imagen 34: Menú principal de la aplicación (Autoría Propia)

## 9.2 Módulo de Entrada de Datos

La aplicación hace uso de las siguientes ventanas para seleccionar los archivos que contienen la información del pedido, de los materiales disponibles y del catálogo de productos.

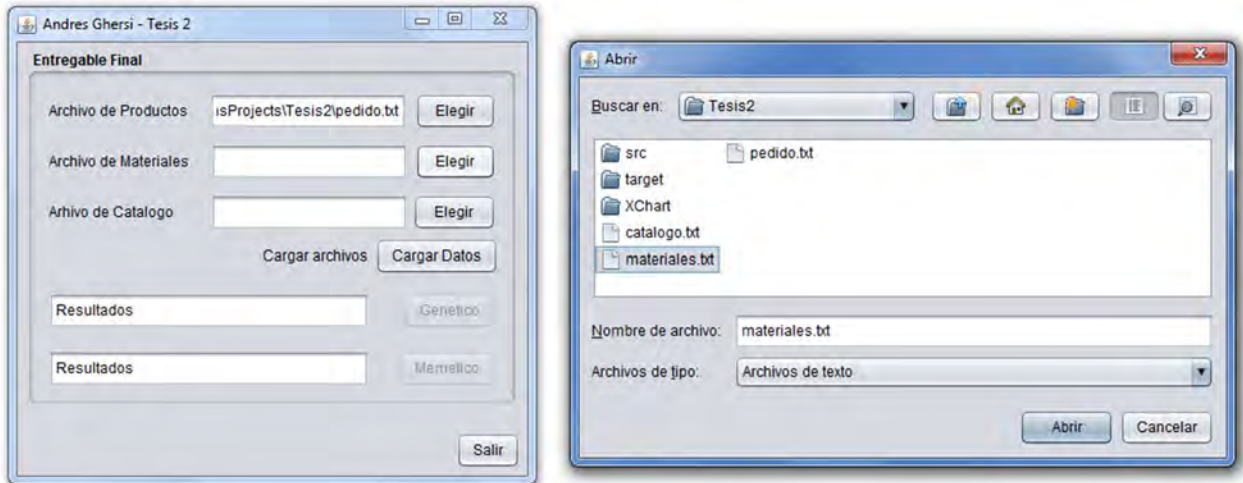


Imagen 35: Menú de selección de archivos (Autoría Propia)

El formato en el que se encuentran los datos es el de archivos de texto con formato. Y siguen la siguiente estructura:

- Archivo de Pedido: Cada línea del archivo representa el contenido de una línea del Pedido. Y en cada línea y separados por caracteres de tabulación, se registra el código del Producto requerido y la cantidad a producir.

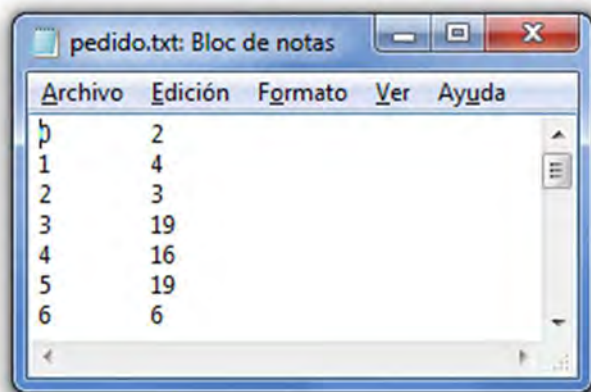


Imagen 36: Datos de entrada 1(Autoría Propia)

- Archivo de Productos: Cada línea del archivo representa un Producto, un SKU. Y en cada línea y separados por caracteres de tabulación, se registra el código del Producto y el largo del Producto.

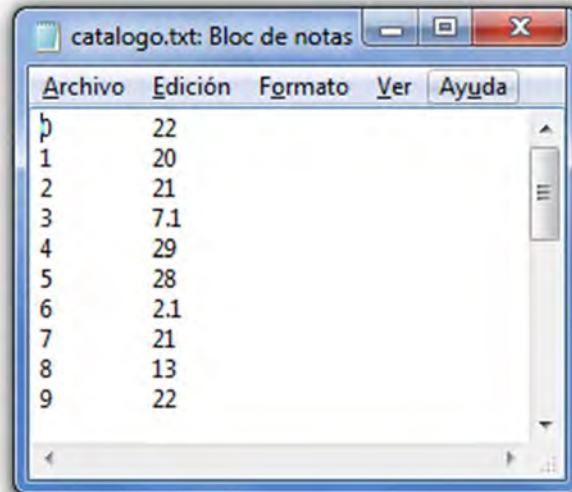


Imagen 37: Datos de entrada 2(Autoría Propia)

- Archivo de Materiales Disponibles: Cada línea del archivo representa un tipo de Barra de Material Disponible para cubrir la producción del Pedido. Y en cada línea y separados por caracteres de tabulación, se registra el código del Material Disponible y la cantidad.

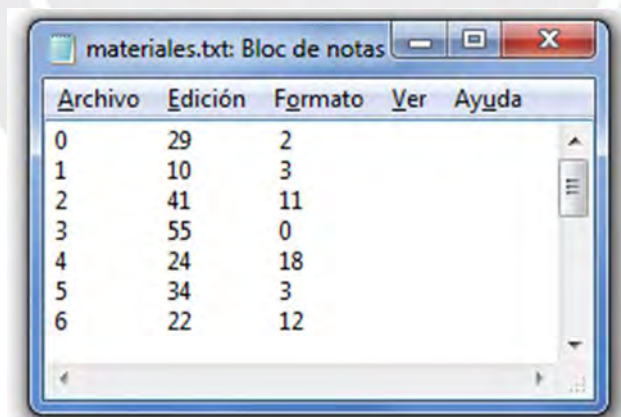


Imagen 38: Datos de entrada 3(Autoría Propia)

Estos archivos son cargados por los algoritmos de entrada de datos en las estructuras presentadas en los capítulos anteriores.

Luego de seleccionar los tres archivos, al presionar el botón “Cargar Datos”, se generan las estructuras internas de los pedidos y materiales.

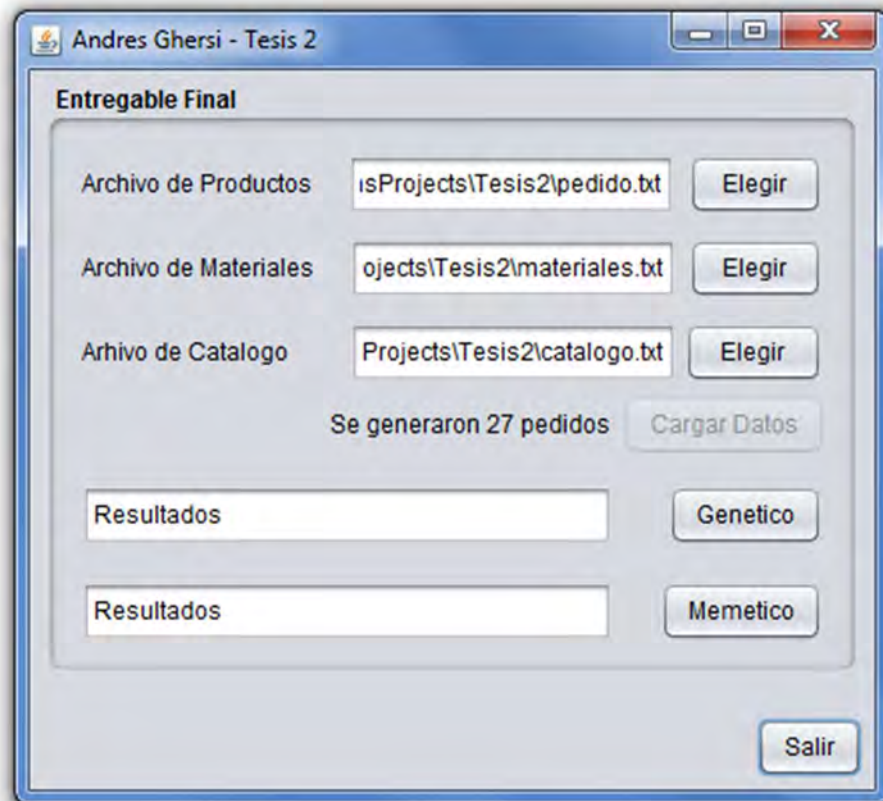
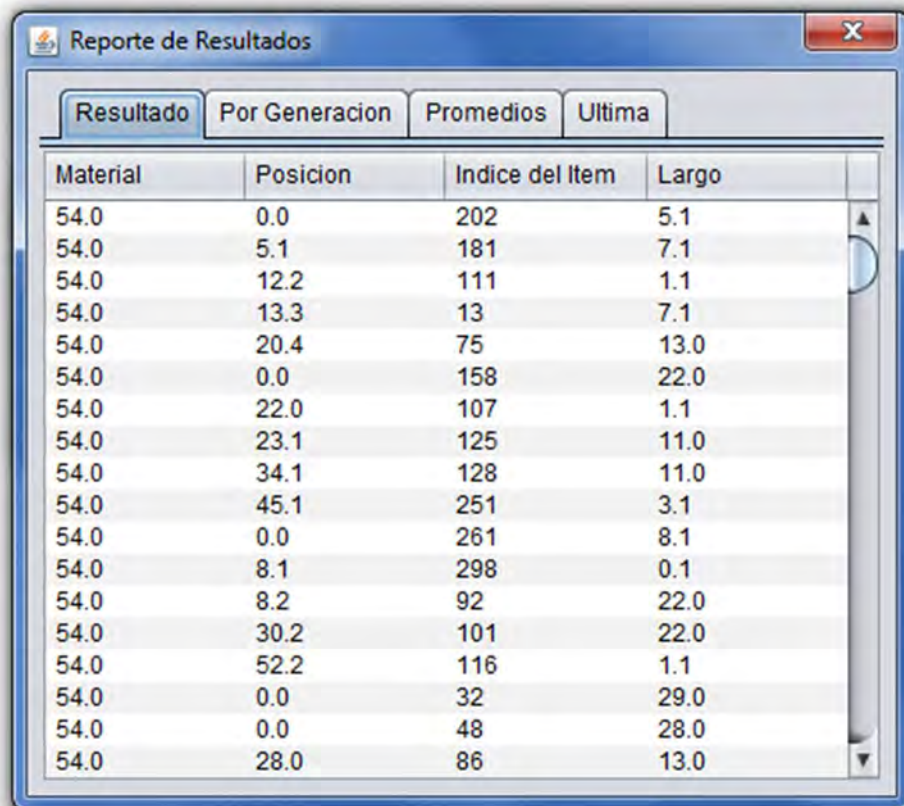


Imagen 39: Menú principal luego de cargar los datos (Autoría Propia)

### 9.3 Módulo de Salida de Datos

La aplicación hace uso de las siguientes ventanas para mostrar los resultados de los cálculos realizados.

Al terminar, el resultado de la experimentación es exportado a una tabla de texto con formato.



| Material | Posicion | Indice del Item | Largo |
|----------|----------|-----------------|-------|
| 54.0     | 0.0      | 202             | 5.1   |
| 54.0     | 5.1      | 181             | 7.1   |
| 54.0     | 12.2     | 111             | 1.1   |
| 54.0     | 13.3     | 13              | 7.1   |
| 54.0     | 20.4     | 75              | 13.0  |
| 54.0     | 0.0      | 158             | 22.0  |
| 54.0     | 22.0     | 107             | 1.1   |
| 54.0     | 23.1     | 125             | 11.0  |
| 54.0     | 34.1     | 128             | 11.0  |
| 54.0     | 45.1     | 251             | 3.1   |
| 54.0     | 0.0      | 261             | 8.1   |
| 54.0     | 8.1      | 298             | 0.1   |
| 54.0     | 8.2      | 92              | 22.0  |
| 54.0     | 30.2     | 101             | 22.0  |
| 54.0     | 52.2     | 116             | 1.1   |
| 54.0     | 0.0      | 32              | 29.0  |
| 54.0     | 0.0      | 48              | 28.0  |
| 54.0     | 28.0     | 86              | 13.0  |

Imagen 40: Reporte de resultado (Autoría Propia)

Y sigue la siguiente estructura.

- Columnas: Contiene cuatro columnas. Material (en punto decimal), posición (en punto decimal), índice (entero) y largo (en punto decimal). Y representa la información por cada producto a producirse.
- Filas: Cada línea representa un producto del archivo de pedido que va a ser elaborado en el material que precede en la primera columna. El material está designado por largo, debido a que son barras de material disponible. Además, se

indica la posición en la barra (0 para el inicio de la barra) desde la que el producto será producido. Finalmente, se indica el largo de la barra a producir.

Además, existen gráficas para observar información de las soluciones de manera visual.

- Gráfico por Generación: Gráfica de la función de puntaje de las mejores soluciones de cada generación.

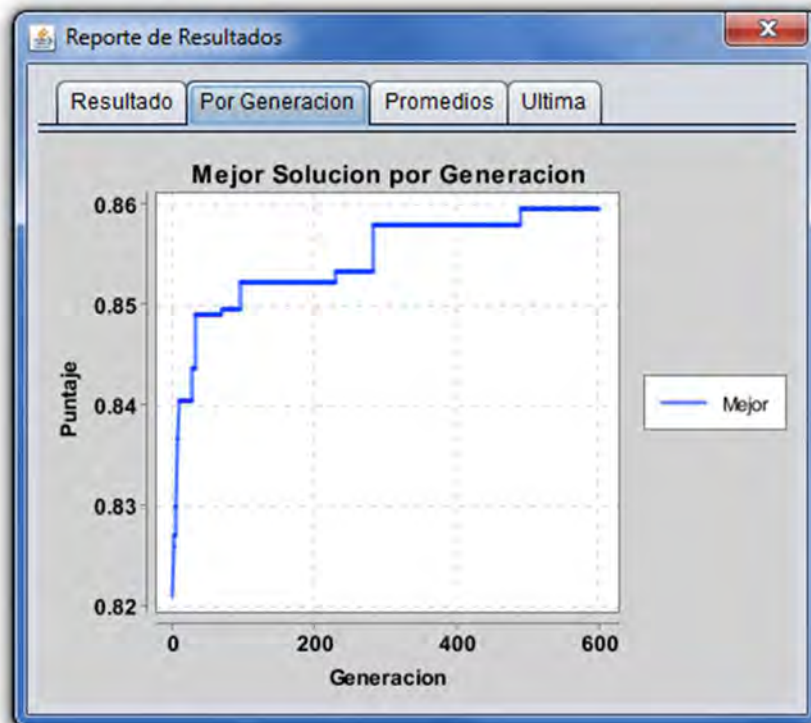


Imagen 41: Reporte de solución por generación (Autoría Propia)

- Gráfico de Promedios: Gráfica del promedio de la función de puntaje de las generaciones.

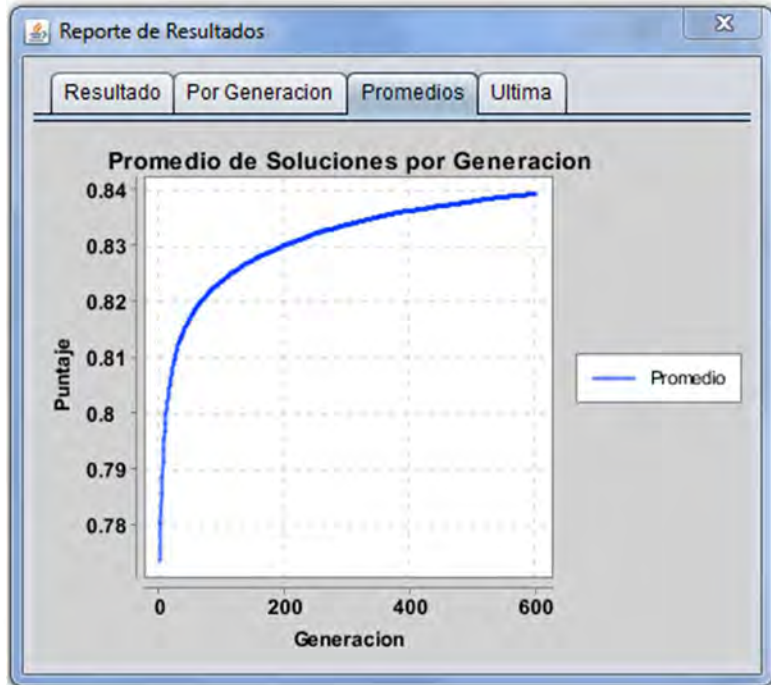


Imagen 42: Reporte de promedios por generación (Autoría Propia)

- Gráfico de la Última Generación: Gráfica del puntaje de cada solución de la última generación

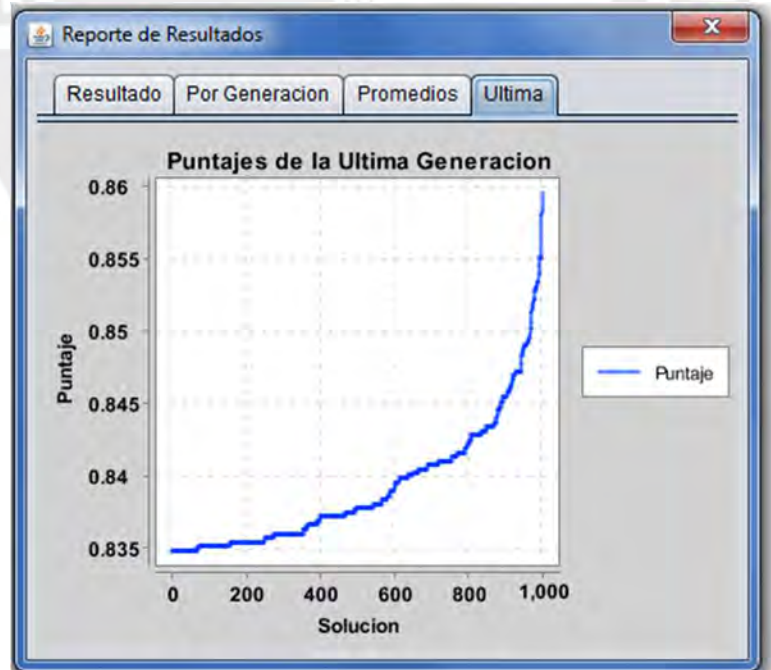


Imagen 43: Reporte de la última generación (Autoría Propia)

Finalmente, se observan los resultados de ambos algoritmos.

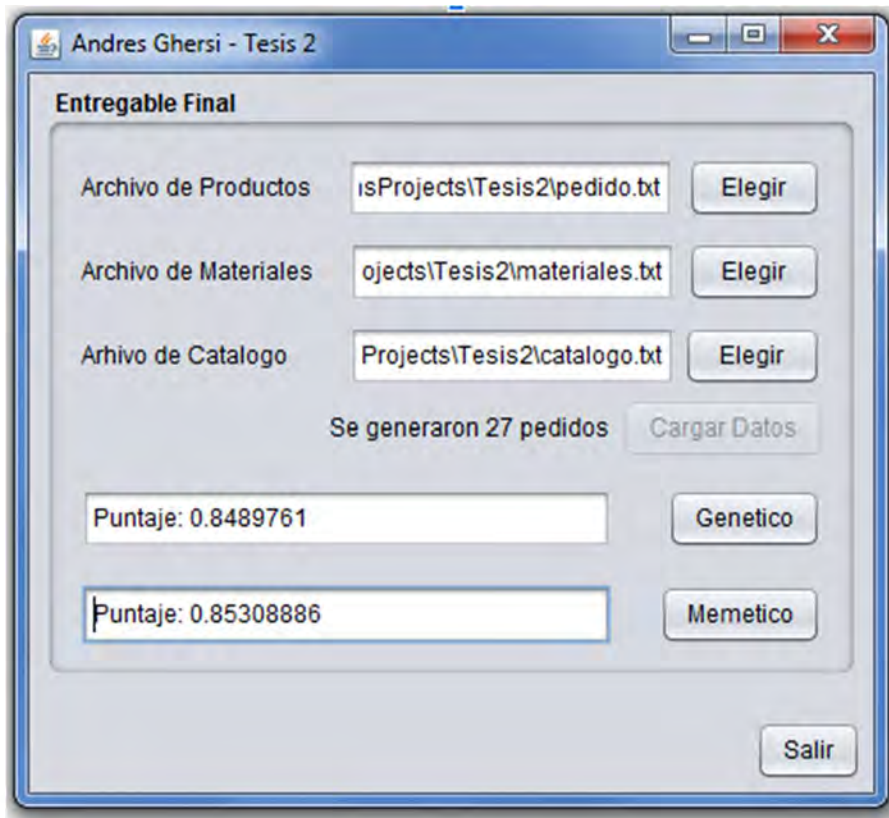


Imagen 44: Menú principal (Autoría Propia)

En este caso, por ejemplo, el algoritmo genético alcanzó un puntaje de 0.849 y el algoritmo memético alcanzó 0.853. Una mejora de 4.7% con los mismos datos de entrada.



# Capítulo 10. Experimentación Numérica

## 10.1 Introducción

En este capítulo se presenta el **Resultado Esperado 10 (RE10)**, que corresponde a la calibración de los algoritmos, la experimentación numérica y el reporte de resultados y conclusiones.

En este capítulo se presentan los resultados numéricos de la calibración de las variables de ambos algoritmos: tamaños, probabilidades y cantidades, además de la probabilidad de selección de una posible mejora (variable alpha) para el algoritmo memético. Finalmente, se presentan los resultados de la experimentación numérica y las conclusiones.

## 10.2 Recolección de datos

Para la obtención de los datos de prueba, se generaron 40 sets de datos de entrada, los cuales fueron procesados por los algoritmos 100 veces y se conservaron los puntajes de la función fitness con mejor valor. Estos valores son promediados y evaluados para decidir cuál es la mejor configuración de la variable estudiada en la calibración. Los valores de los resultados de la Calibración de los Algoritmos se encuentran en el anexo A (Calibración de la Población Inicial), anexo B (Calibración del Algoritmo Genético) y anexo C (Calibración del algoritmo Memético).

En el caso de la experimentación numérica, se escogió la mejor configuración de las variables en cada algoritmo para luego ejecutar cada solución con el set original de datos (los 40 sets). Los valores de los resultados de la Experimentación Numérica se encuentran en el anexo D.

Aquí se muestra una representación gráfica de los resultados encontrados.

### Poblacion Inicial, Genético y Memético

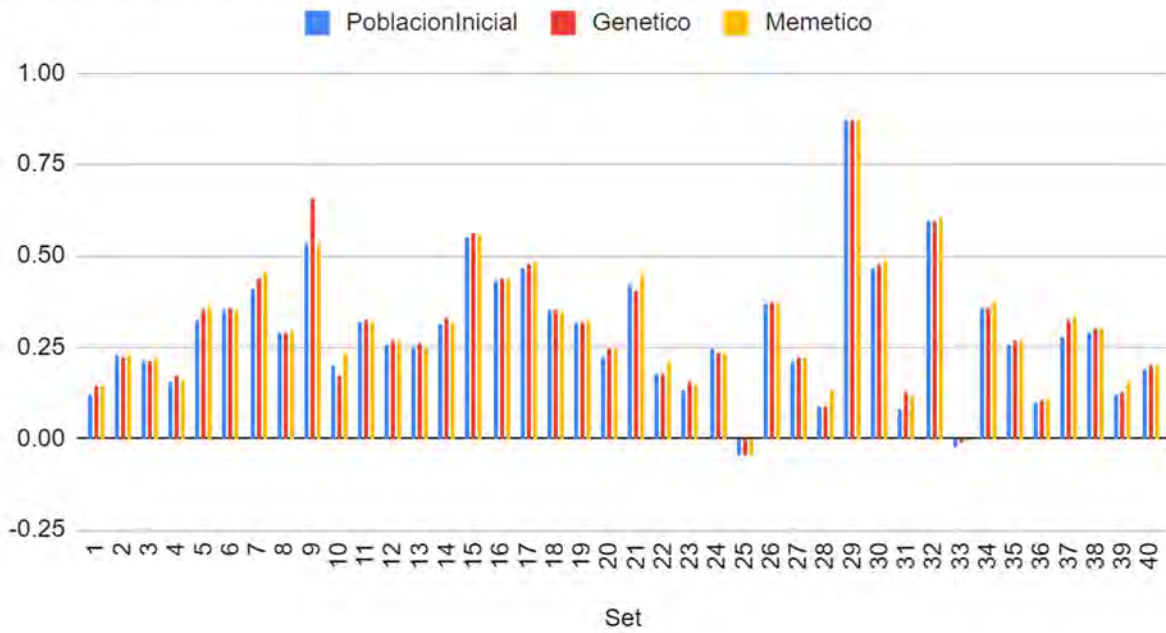


Imagen 45: Resultados de experimentación (Autoría Propia)

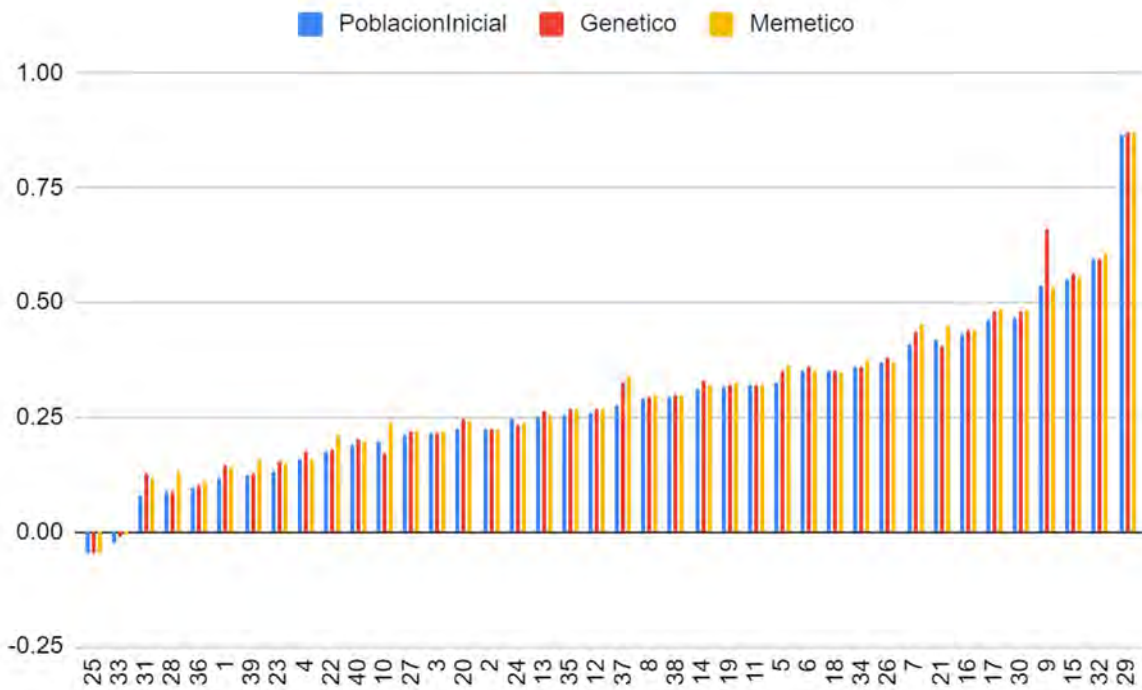


Imagen 46: Resultados de experimentación numérica (Autoría Propia)

## 10.3 Calibración de los Algoritmos

Para la calibración de los algoritmos, se realizó una evaluación de las soluciones iniciales generadas por el algoritmo de población inicial. Luego, se ejecutaron los algoritmos con diferentes valores de las variables y se conservó la mejor configuración, comparando la función fitness con la población inicial.

Cada nueva variable es probada utilizando los mejores valores de las fases anteriores.

### 10.3.1 Población Inicial

Para la calibración de la población inicial, se ejecutó el algoritmo de generación con diferentes tamaños de poblaciones. Los resultados de la función fitness de la población inicial son los siguientes:

|                      | Tamaño de la Población Inicial |                 |               |                 |               |              |               |                 |                |                 |
|----------------------|--------------------------------|-----------------|---------------|-----------------|---------------|--------------|---------------|-----------------|----------------|-----------------|
|                      | 50                             | 100             | 150           | 200             | 250           | 300          | 350           | 400             | 450            | 500             |
| Promedio             | 0.27492<br>14                  | 0.28274<br>3625 | 0.28819<br>63 | 0.28795<br>2075 | 0.29054<br>87 | 0.29190<br>8 | 0.28938<br>23 | 0.29359<br>1775 | 0.29566<br>865 | 0.29861<br>2925 |
| Porcentaje de mejora |                                | 2.8453<br>%     | 4.8286<br>%   | 4.7398<br>%     | 5.6843<br>%   | 6.1787<br>%  | 5.2600<br>%   | 6.7912<br>%     | 7.5466<br>%    | 8.6176<br>%     |
| Porcentaje / Tamaño  |                                | 0.0285<br>%     | 0.0322<br>%   | 0.0237<br>%     | 0.0227<br>%   | 0.0206<br>%  | 0.0150<br>%   | 0.0170<br>%     | 0.0168<br>%    | 0.0172<br>%     |

Tabla 8: Calibración de la Población Inicial (Autoría Propia)

Se eligió el valor de 150 debido a que presenta el mejor ratio Porcentaje de Mejor / Tamaño de la Población. Por la naturaleza de la generación (aleatoria), un aumento indefinido siempre producirá mejores soluciones máximas, por lo que el enfoque cambia se convierte en buscar el valor más adecuado para el experimento. En este caso, el valor que aumenta más el fitness, sin generar muchas soluciones. Es decir, el ratio porcentaje de mejora entre tamaño de la población.

## Mejora / Tamaño vs. Tamaño Poblacion

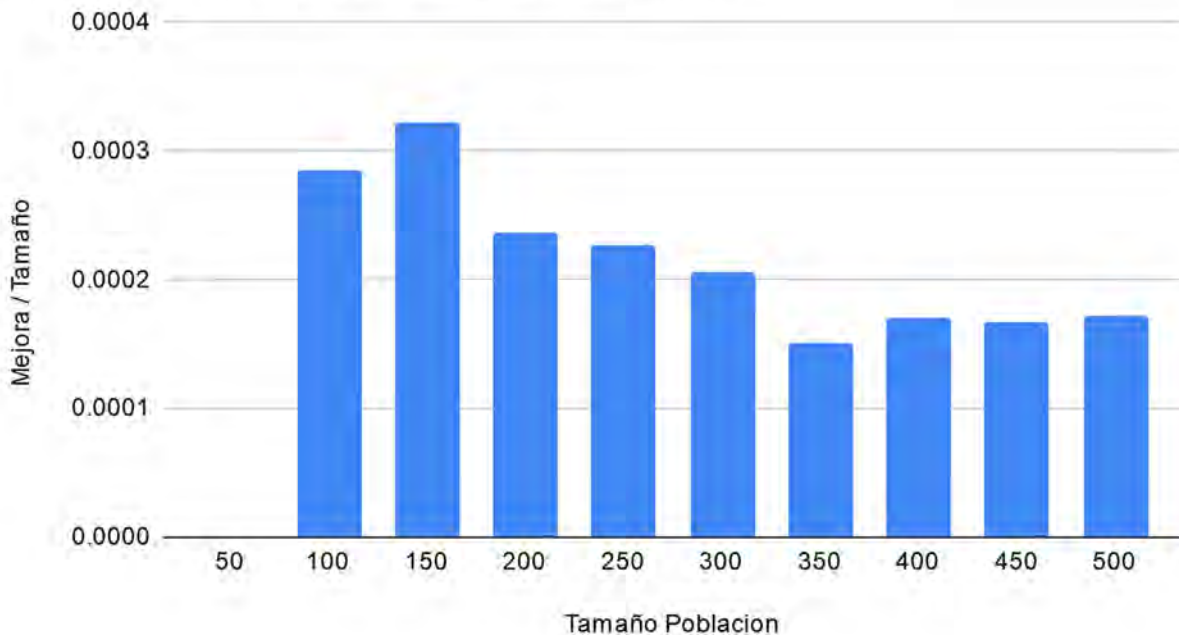


Imagen 47: Calibración de la Población Inicial (Autoría Propia)

El mayor porcentaje de mejora por tamaño de la población es de 0.0322%, alcanzado en la población de tamaño 150. Se configura la variable tamaño de la población inicial en el valor 150. Este valor, además, será usado en las siguientes fases.

## 10.4 Algoritmo Genético

Aquí se presenta la calibración del algoritmo genético y sus variables.

### 10.4.1 Tasa de Casamiento

Tomando como base la población inicial generada, se realizó la fase de crossover con diferentes valores de la tasa de casamiento. Los resultados de la función fitness de la población son los siguientes:

|          | Tasa de Casamiento |         |         |         |         |         |         |         |         |         |
|----------|--------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
|          | 5%                 | 15%     | 25%     | 35%     | 45%     | 55%     | 65%     | 75%     | 85%     | 95%     |
| Promedio | 0.28806            | 0.28829 | 0.29123 | 0.29025 | 0.29476 | 0.29292 | 0.29299 | 0.29110 | 0.29337 | 0.29349 |
|          | 9025               | 465     | 41      | 975     | 3825    | 8375    | 33      | 3875    | 9975    | 375     |
| Mejora   | -                  | 0.0341  | 1.0541  | 0.7160  | 2.2788  | 1.6420  | 1.6645  | 1.0089  | 1.7987  | 1.8381  |

|  |        |   |   |   |   |   |   |   |   |   |
|--|--------|---|---|---|---|---|---|---|---|---|
|  | 0.0442 | % | % | % | % | % | % | % | % | % |
|  | %      |   |   |   |   |   |   |   |   |   |

Tabla 9: Calibración de la Tasa de Casamiento (Autoría Propia)

Después de probar la variable en valores desde el 5% al 95%, se observa que la mayor mejora respecto a la población inicial se genera cuando la tasa de casamiento es de 45%.

### Mejora vs. Tasa de Casamiento

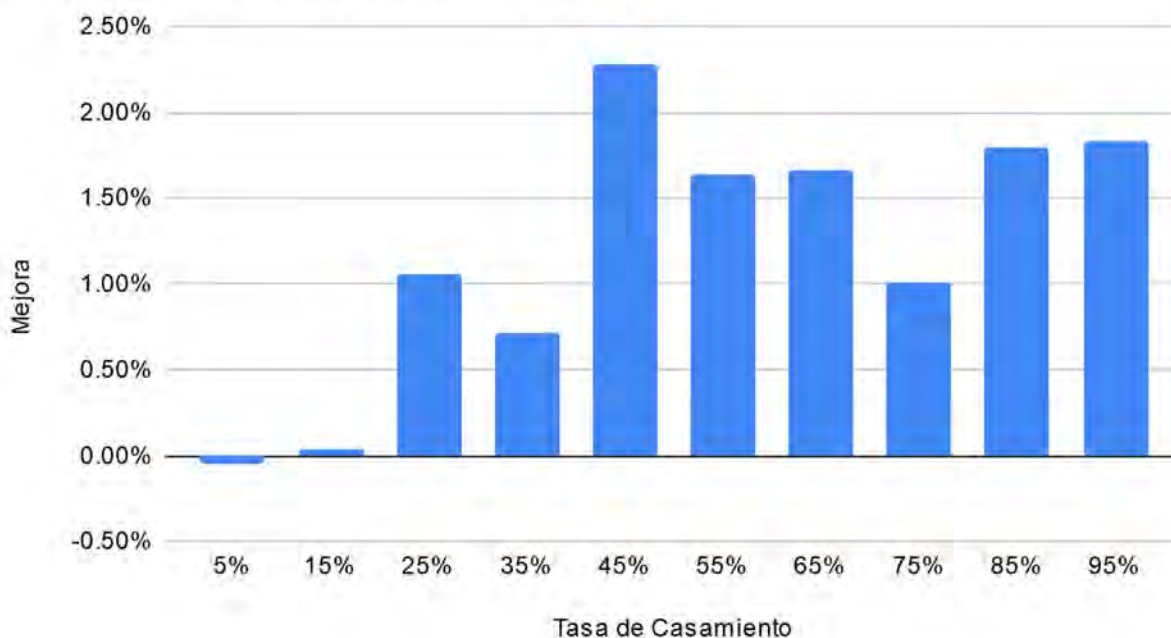


Imagen 48: Calibración de la Tasa de Casamiento (Autoría Propia)

El mayor porcentaje de mejora es de 2.2788%, alcanzado en la tasa de casamiento de 45%. Se configura la variable Tasa de Casamiento en el valor 45%. Este valor, además, será usado en las siguientes fases.

### 10.4.2 Tasa de Mutación

Tomando como base la población inicial generada y la fase de crossover con la variable Tasa de Casamiento al 45%, se realizó la fase de mutación con diferentes valores. Los resultados de la función fitness de la población son los siguientes:

|         | Tasa de Mutación |         |         |         |         |         |
|---------|------------------|---------|---------|---------|---------|---------|
|         | 5%               | 6%      | 7%      | 8%      | 9%      | 10%     |
| Promedi | 0.28966          | 0.29133 | 0.29247 | 0.29232 | 0.29101 | 0.29189 |

|        |             |             |             |             |             |             |
|--------|-------------|-------------|-------------|-------------|-------------|-------------|
| o      | 4425        | 7875        | 7175        | 0025        | 98          | 195         |
| Mejora | 0.5094<br>% | 1.0901<br>% | 1.4854<br>% | 1.4309<br>% | 0.9797<br>% | 1.2823<br>% |

Tabla 10: Calibración de la Tasa de Mutación (Autoría Propia)

Después de probar la variable en valores desde el 5% al 10%, se observa que la mayor mejora respecto a la población inicial se genera cuando la tasa de mutación es de 45%.

### Mejora vs. Tasa de Mutacion

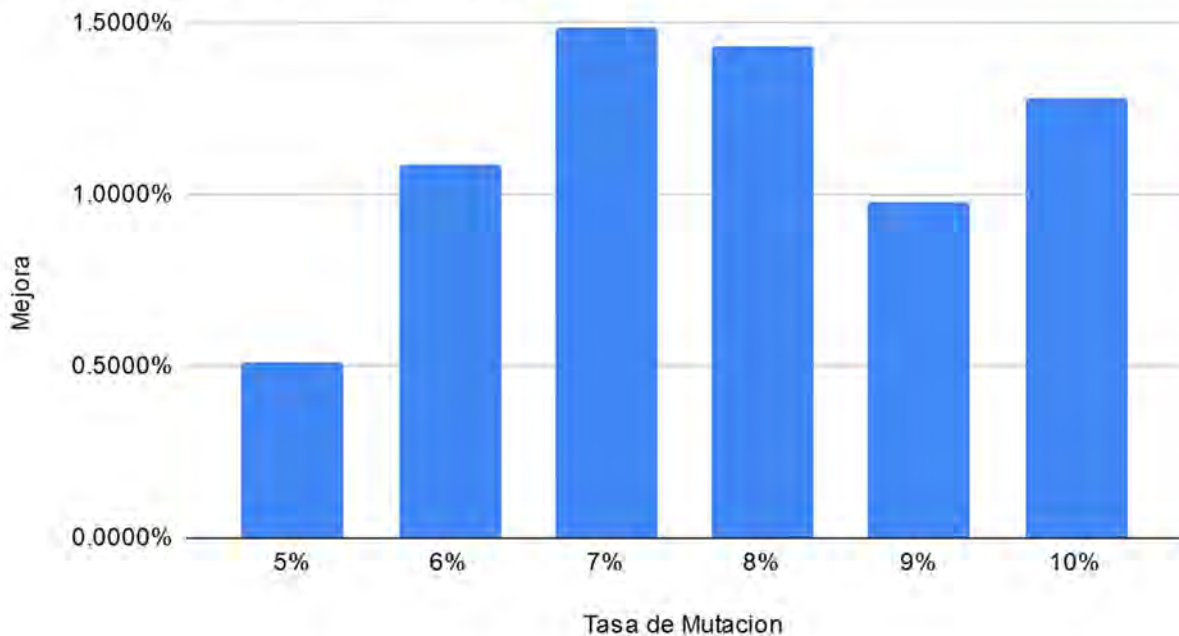


Imagen 49: Calibración de la Tasa de Mutación (Autoría Propia)

El mayor porcentaje de mejora es de 1.4854%, alcanzado en la tasa de mutación de 7%. Se configura la variable Tasa de Mutación en el valor 7%. Este valor, además, será usado en las siguientes fases.

#### 10.4.3 Cantidad de generaciones sin mejora

Tomando como base la población inicial generada, la fase de crossover con la variable Tasa de Casamiento al 45% y la fase de mutación con la variable Tasas de Mutación al 7% se realizó la fase de evolución de múltiples generaciones con diferentes valores en el máximo de generaciones sin mejora. Los resultados de la función fitness de la población son los siguientes:

|                                     |
|-------------------------------------|
| Cantidad de generaciones sin mejora |
|-------------------------------------|

|                     | 1               | 2              | 3               | 4             | 5            | 6             | 7               | 8              | 9             | 10            |
|---------------------|-----------------|----------------|-----------------|---------------|--------------|---------------|-----------------|----------------|---------------|---------------|
| Promedio            | 0.29087<br>6325 | 0.29573<br>485 | 0.30117<br>6925 | 0.29976<br>35 | 0.29978<br>4 | 0.29922<br>96 | 0.30262<br>5825 | 0.30191<br>515 | 0.30569<br>02 | 0.30121<br>06 |
| Mejora              | 0.9299<br>%     | 2.6158<br>%    | 4.5041<br>%     | 4.0137<br>%   | 4.0208<br>%  | 3.8284<br>%   | 5.0068<br>%     | 4.7602<br>%    | 6.0701<br>%   | 4.5158<br>%   |
| Porcentaje / Tamaño | 0.93%           | 1.31%          | 1.50%           | 1.00%         | 0.80%        | 0.64%         | 0.72%           | 0.60%          | 0.67%         | 0.45%         |

Tabla 11: Calibración de la Cantidad de generaciones sin mejora (Autoría Propia)

En este caso, siguiendo la lógica expuesta en puntos anteriores, debido al costo de aumentar las generaciones, se elige el valor de 3 generaciones debido a que presenta el mayor aumento en fitness por generación.

### Mejora / Tamaño vs. Cantidad de generaciones

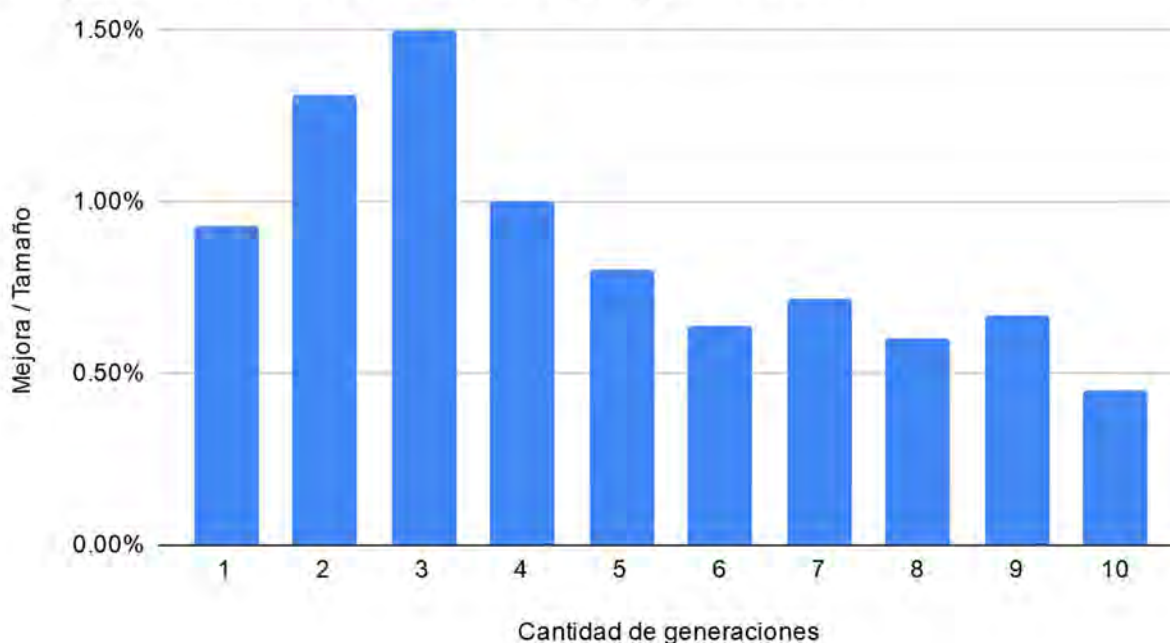


Imagen 50: Calibración de la Cantidad de Generación sin mejora(Autoría Propia)

El mayor ratio de mejora es de 4.5041%, alcanzado en la cantidad de generaciones sin mejora de 3. Se configura la variable Cantidad de generaciones sin mejora en el valor de 3. Este valor, además, será usado en las siguientes fases.

#### 10.4.4 Cantidad máxima de generaciones

Tomando como base la población inicial generada, la fase de crossover con la variable Tasa de Casamiento al 45% y la fase de mutación con la variable Tasas de Mutación al 7% se realiza la fase de evolución con el máximo de generaciones sin mejora en 3 y la cantidad total de generaciones en valores de 100 a 600. Los resultados de la función fitness de la población son los siguientes:

|          | Cantidad máxima de generaciones |         |         |         |         |         |
|----------|---------------------------------|---------|---------|---------|---------|---------|
|          | 100                             | 200     | 300     | 400     | 500     | 600     |
| Promedio | 0.29692                         | 0.29667 | 0.29696 | 0.29653 | 0.29765 | 0.29999 |
|          | 9275                            | 47      | 7575    | 0325    | 3075    | 09      |
| Mejora   | 3.0302                          | 2.9419  | 3.0435  | 2.8918  | 3.2814  | 4.0926  |
|          | %                               | %       | %       | %       | %       | %       |

Tabla 12: Calibración de la Cantidad Máxima de Generaciones(Autoría Propia)

Después de probar la variable en valores desde el 100 hasta el 600, se observa que la mayor mejora respecto a la población inicial se genera cuando la cantidad es de 600.

Mejora vs. Cantidad de generaciones

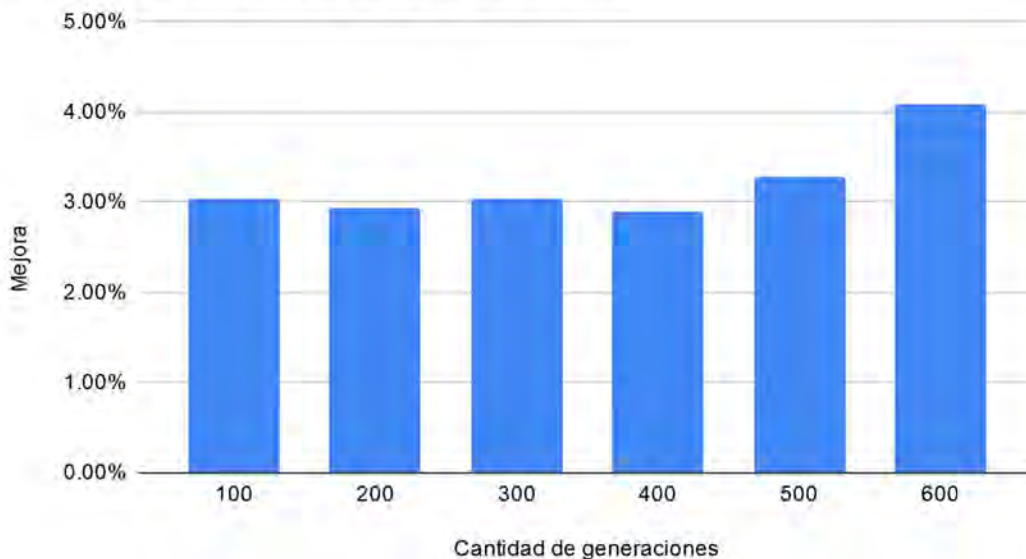


Imagen 51: Calibración de la Cantidad Máxima de Generaciones(Autoría Propia)

El mayor porcentaje de mejora es de 4.0926%, alcanzado en la cantidad de generaciones de 600. Se configura la variable Cantidad Máxima de Generaciones en el valor 600.



# 10.5 Algoritmo Memético

Aquí se presenta la calibración del algoritmo genético y sus variables.

## 10.5.1 Tasa de Voracidad (Tasa Alpha)

Tomando como base la población inicial generada, la fase de crossover con la variable Tasa de Casamiento al 45% y la fase de mutación con la variable Tasas de Mutación al 7% se realizó la fase de búsqueda local en la que se calibra la variable alpha o tasa de Voracidad con diferentes valores del 0% al 100%. Los resultados de la función fitness de la población son los siguientes:

|          | Tasa de Voracidad |                |               |                |                 |                 |
|----------|-------------------|----------------|---------------|----------------|-----------------|-----------------|
|          | 0%                | 20%            | 40%           | 60%            | 80%             | 100%            |
| Promedio | 0.29270<br>0075   | 0.28996<br>275 | 0.29633<br>49 | 0.29847<br>495 | 0.29718<br>1475 | 0.29712<br>4625 |
| Mejora   | 1.5627<br>%       | 0.6129<br>%    | 2.8240<br>%   | 3.5665<br>%    | 3.1177<br>%     | 3.0980<br>%     |

Tabla 13: Calibración de la Tasa de Voracidad (Autoría Propia)

Después de probar la variable en valores desde el 0% hasta el 100%, se observa que la mayor mejora respecto a la población inicial se genera cuando la tasa de voracidad es de 60%.

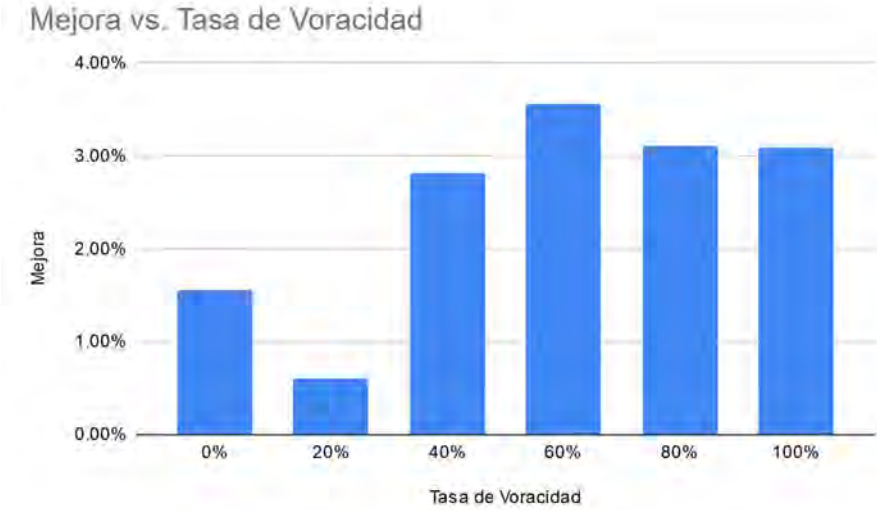


Imagen 52: Calibración de la Tasa de Voracidad (Autoría Propia)

## 10.5.2 Cantidad de generaciones sin mejora

Tomando como base la población inicial generada, la fase de crossover con la variable Tasa de Casamiento al 45%, la fase de mutación con la variable Tasas de Mutación al 7% y la fase de búsqueda local con la variable alpha o tasa de Voracidad al 60%, se realiza la fase de evolución. Los resultados de la función fitness de la población son los siguientes:

|                      | Cantidad máxima de generaciones |                |                |                |               |                 |
|----------------------|---------------------------------|----------------|----------------|----------------|---------------|-----------------|
|                      | 1                               | 2              | 3              | 4              | 5             | 6               |
| Promedio             | 0.29636<br>115                  | 0.30582<br>835 | 0.30220<br>365 | 0.30056<br>995 | 0.30286<br>95 | 0.30851<br>6925 |
| Mejora               | 2.8331<br>%                     | 6.1181<br>%    | 4.8604<br>%    | 4.2935<br>%    | 5.0914<br>%   | 7.0510<br>%     |
| Mejora /<br>Cantidad | 2.83%                           | 3.06%          | 1.62%          | 1.07%          | 1.02%         | 1.18%           |

Tabla 14: Calibración de la Cantidad de Generaciones sin mejora (Autoría Propia)

Después de probar la variable en valores desde el 1 hasta 6, se observa que la mayor mejora respecto a la población inicial se genera cuando la cantidad es de 2.

Mejora vs. Cantidad de Generaciones

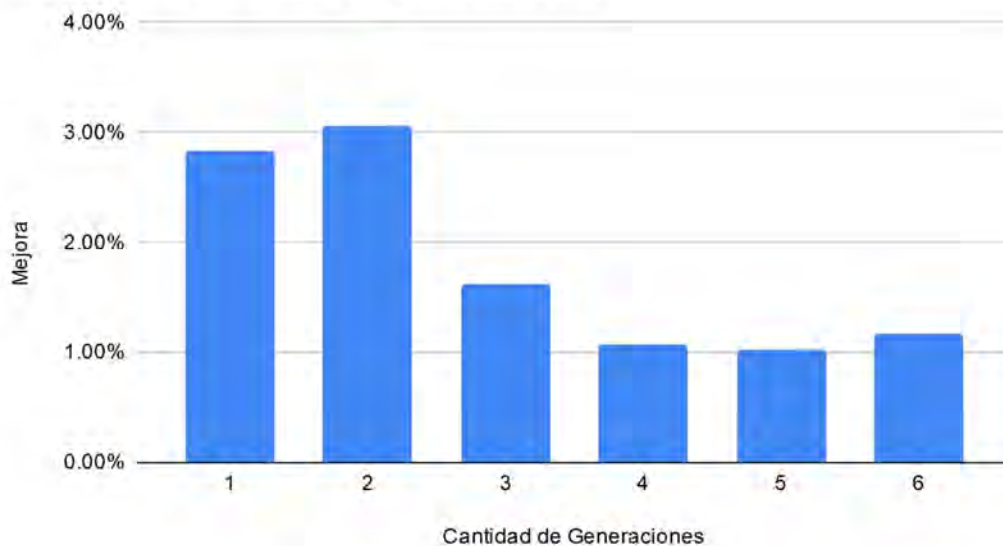


Imagen 53: Calibración de la Cantidad Máxima de Generaciones sin mejora (Autoría Propia)

El mayor porcentaje de mejora es de 3.06%, alcanzado en la cantidad de generaciones de 2. Se configura la variable Cantidad Máxima de Generaciones sin mejora en el valor 2.

### 10.5.3 Cantidad máxima de generaciones

Tomando como base la población inicial generada, la fase de crossover con la variable Tasa de Casamiento al 45%, la fase de mutación con la variable Tasas de Mutación al 7%, la fase de búsqueda local con la variable alpha o tasa de Voracidad al 60%, se realiza la fase de evolución con el máximo de generaciones sin mejora en 2 y la cantidad total de generaciones en valores de 100 a 600. Los resultados de la función fitness de la población son los siguientes:

|          | Cantidad máxima de generaciones |              |                 |               |               |               |
|----------|---------------------------------|--------------|-----------------|---------------|---------------|---------------|
|          | 100                             | 200          | 300             | 400           | 500           | 600           |
| Promedio | 0.29959<br>27                   | 0.30023<br>9 | 0.30275<br>4325 | 0.30151<br>57 | 0.30069<br>79 | 0.29961<br>01 |
| Mejora   | 3.9544<br>%                     | 4.1786<br>%  | 5.0514<br>%     | 4.6216<br>%   | 4.3379<br>%   | 3.9604<br>%   |

Tabla 15: Calibración de la Cantidad Total de Generaciones (Autoría Propia)

Después de probar la variable en valores desde el 100 hasta el 600, se observa que la mayor mejora respecto a la población inicial se genera cuando la cantidad es de 200.

Mejora vs. Cantidad de Generaciones

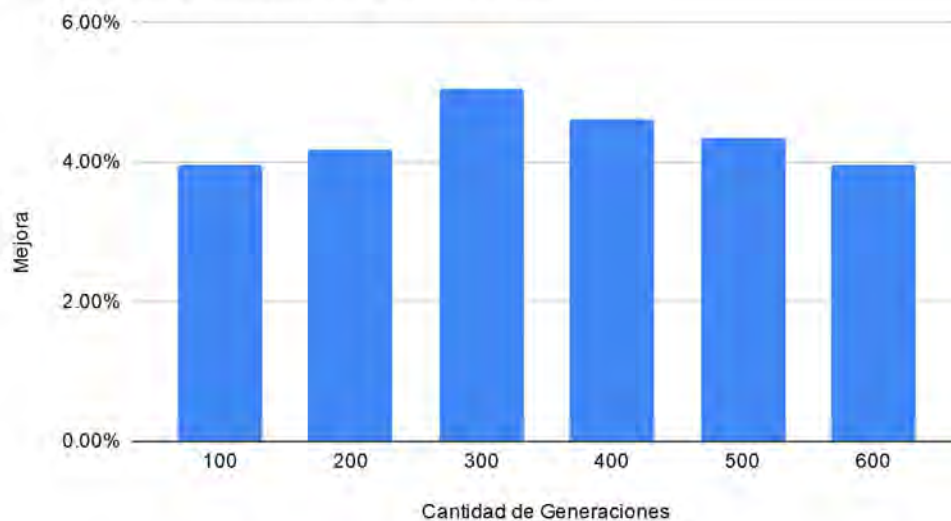


Imagen 54: Calibración de la Cantidad Total de Generaciones (Autoría Propia)

El mayor porcentaje de mejora es de 4.1786%, alcanzado en la cantidad de generaciones de 200. Se configura la variable Cantidad Máxima de Generaciones en el valor 200.

## 10.3 Experimentación Numérica

Una vez encontrados los valores óptimos de las variables, se realizó la Experimentación Numérica (Anexo D) y las Pruebas Estadísticas. Estos son los resultados.

Los datos de la experimentación numérica (algoritmo, set de datos y puntajes) se encuentran en el anexo D.

### 10.3.1 Pruebas Estadísticas

Una vez realizada la Experimentación Numérica (Anexos), se realiza una prueba de hipótesis para saber si el algoritmo memético genera mejores soluciones que el algoritmo genético (y la población inicial).

Se hace uso de la prueba Z para evaluar la hipótesis. Se evalúa si ambas muestras (de ambos algoritmos) son estadísticamente diferentes o iguales.

| Prueba Z para medias de dos muestras |  |
|--------------------------------------|--|
| Hipótesis Nula (H0)                  | H0: Las medias de ambas muestras son estadísticamente iguales    |
| Hipótesis Alternativa (H1)           | H1: Las medias de ambas muestras son NO estadísticamente iguales |

Para realizar la prueba Z, existen dos requisitos que serán verificados con el uso de las pruebas KS y F de Fisher.

| Requisito   | Prueba para comprobar        |
|---|------------------------------|
| Cada muestra por separado sigue una distribución normal | Prueba de Kolmogorov-Smirnov |
| Las varianzas de ambas muestras son homogéneas          | Prueba de F de Fisher        |

A continuación, se presentan los resultados.

#### 10.3.1.2 Prueba de Kolmogorov-Smirnov

Esta prueba busca verificar si una muestra de datos sigue la distribución normal.

|                       |  |
|-----------------------|--|
| Objetivo de la prueba | Verificar si las muestras siguen una distribución normal |
| Hipótesis Nula        | La muestra sigue una distribución normal                 |
| Hipótesis Alternativa | La muestra NO sigue una distribución normal              |

Resultado para la muestra del algoritmo genético:

|   |  |
|---|--|
| Estadístico K-S   | 0.11262  |
| p-value   | 0.64964  |
| P-value > valor de significancia (0.05)   | P-value < valor de significancia (0.05)  |
| Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que la muestra sigue una distribución normal. | Debido a que el valor p-value es menor que la significancia de la prueba, Se rechaza la hipótesis nula y se acepta la alternativa. Concluimos que la muestra NO sigue una distribución normal. |

**Resultado:** Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula: Concluimos que la muestra sigue una distribución normal.

Resultado para la muestra del algoritmo memético:

|   |  |
|---|--|
| Estadístico K-S   | 0.11256  |
| p-value   | 0.65025  |
| P-value > valor de significancia (0.05)   | P-value < valor de significancia (0.05)  |
| Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que la muestra sigue una distribución normal. | Debido a que el valor p-value es menor que la significancia de la prueba, Se rechaza la hipótesis nula y se acepta la alternativa. Concluimos que la muestra NO sigue una distribución normal. |

**Resultado:** Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que la muestra sigue una distribución normal.

### 10.3.1.3 Prueba F de Fisher

Esta prueba busca verificar si dos muestras tienen varianzas homogéneas.

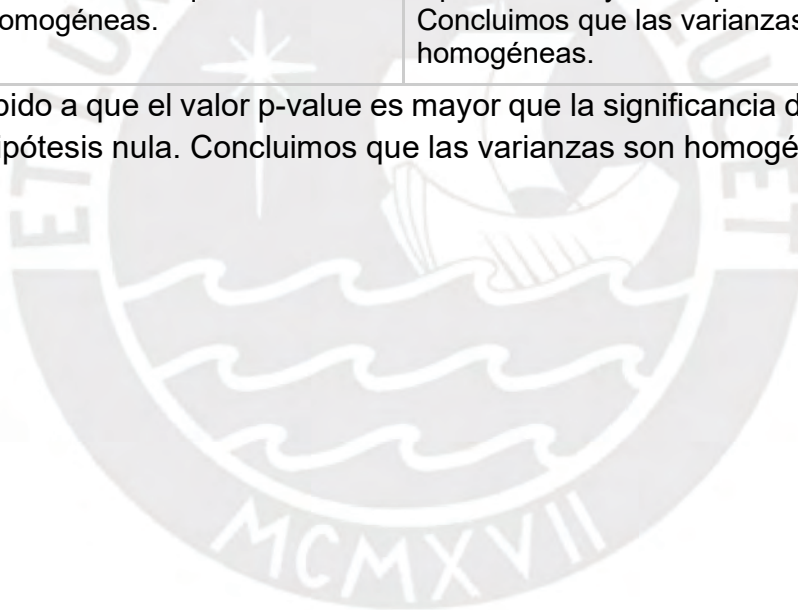
|                      |  |
|----------------------|--|
| Requerimiento previo | La muestra debe seguir una distribución normal |
|----------------------|--|

|                       |  |
|-----------------------|--|
| Objetivo de la prueba | Verificar si las varianzas de las muestras son homogéneas. |
| Hipótesis Nula        | Las varianzas son homogéneas                               |
| Hipótesis Alternativa | Las varianzas NO son homogéneas                            |

Resultado para la muestra del algoritmo genético y memético:

|   |  |
|---|--|
| Estadístico F   | 1.078512   |
| p-value   | 0.814620   |
| P-value > valor de significancia (0.05)   | P-value < valor de significancia (0.05)  |
| Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que las varianzas son homogéneas. | Debido a que el valor p-value es menor que la significancia de la prueba, Se rechaza la hipótesis nula y se acepta la alternativa. Concluimos que las varianzas NO son homogéneas. |

**Resultado:** Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que las varianzas son homogéneas.



#### 10.3.1.4 Prueba Z

Con la prueba Z se comparan finalmente los algoritmos.

|                       |  |
|-----------------------|--|
| Requerimiento previo  | La muestra debe seguir una distribución normal y las varianzas deben ser homogéneas  |
| Objetivo de la prueba | Verificar si las medias de los algoritmos son estadísticamente iguales o diferentes. |
| Hipótesis Nula        | Las medias son iguales.  |
| Hipótesis Alternativa | Las muestras NO son iguales  |

Resultados Finales para ambos algoritmos:

|   |  |
|---|--|
| Estadístico Z   | -0.0706354   |
| Estadístico S'  | 0.0391   |
| p-value   | 0.943688   |
| P-value > valor de significancia (0.05)   | P-value < valor de significancia (0.05)  |
| Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que las medias son iguales. | Debido a que el valor p-value es menor que la significancia de la prueba, Se rechaza la hipótesis nula y se acepta la alternativa. Concluimos que las medias NO son iguales. |

**Resultado:** Debido a que el valor p-value es mayor que la significancia de la prueba, NO se rechaza la hipótesis nula. Concluimos que las medias son iguales.

### 10.4 Conclusiones

**Conclusión:** “Se concluye que el algoritmo memético genera mejores resultados numéricos alcanzando un 23% de mejora en comparación con el algoritmo genético y que están a la par estadísticamente (Prueba Z, p-value 0.943688) con los resultados del algoritmo genético.”

# Capítulo 11. Conclusiones y Trabajos Futuros

## 11.1 Conclusiones

El objetivo de esta tesis es aportar una solución al problema del consumo de recursos en la producción industrial. En particular, el caso de la producción de barras de acero para hormigón. Desde una perspectiva de procesos y consumo de recursos, el caso se adapta al problema de corte de materiales. Así que este proyecto ha desarrollado una solución a dicho problema: el problema de cutting stock.

Inicialmente, se investigó en el estado del arte, y se encontraron diversas soluciones y variantes del problema. Sin embargo, existe la ausencia de variedad de soluciones unidimensionales, lo que ocasiona que se opte por proponer una nueva alternativa basada en los algoritmos meméticos.

Para comprobar la eficiencia de este nuevo enfoque, se planteó un experimento: Implementar un algoritmo memético que esté aplicado al caso del corte unidimensional y compararlo con el desempeño de un algoritmo genético adaptado.

Se aplicaron conocimientos de programación, algoritmia y estadística aplicada, entre otros, y se obtuvo un producto final en forma de aplicación de escritorio que procesa los pedidos de producción y devuelve una respuesta que describe la producción planificada para obtener los productos finales.

Tras la experimentación numérica, al estudiar los algoritmos genéticos y meméticos, se ha encontrado la siguiente información acerca de su eficacia: Se ha alcanzado una mejora en comparación con el algoritmo de la literatura de 23%.

|  | Genético | Memético |
|--|----------|----------|
| Tasa de Casamiento                     | 45%      |          |
| Tasa de mutación                       | 7%       |          |
| Parámetro Alpha                        |          | 60%      |
| Máximo de Generaciones sin Mejora      | 2        | 3        |
| Máximo de Generaciones                 | 600      | 300      |
| Mejora comparado con Población Inicial | 4.09%    | 5.05%    |



|                               |  |     |
|-------------------------------|--|-----|
| Mejora comparado con Genético |  | 23% |
|-------------------------------|--|-----|

Tabla 16: Resultados Finales (Autoría Propia)

Finalmente, respecto a los objetivos y resultados, se puede afirmar que se han cumplido los objetivos específicos, comprobando que: La función objetivo, las variables y las restricciones definidas representan el problema; que las estructuras de datos cumplen con expresar las propiedades estudiadas y el pseudocódigo del algoritmo resuelve el caso de estudio; que la programación refleja el pseudocódigo diseñado y adaptado, y el programa ha sido desarrollado con éxito; y que la calibración de los algoritmos fue exitosa y la experimentación numérica fue realizada.

En resumen, como resultado del desarrollo de los objetivos específicos, podemos concluir que se ha cumplido con el Objetivo General de Implementar un algoritmo memético para resolver el problema de corte de materiales aplicado a la producción de barras de acero para hormigón.

## 11.2 Conclusión del Objetivo General

La interpretación de esta información es que ambos algoritmos mejoran la población inicial en un porcentaje alrededor del 5%, principalmente debido a que se aplican operadores genéticos y de búsqueda local para aumentar el fitness de la población.

Además, concluimos que la etapa de búsqueda local es especialmente beneficiosa, debido a que acelera la evolución de las soluciones (23% de mejora) y lo hace sin ocasionar convergencia en la población (se mantiene normalidad de la muestra, prueba Kolmogorov-Smirnov) al funcionar en conjunto con los operadores globales.

La etapa voraz mejora cada solución individualmente, por lo que se evita promover una convergencia poblacional, pero funciona en conjunto con el resto de operadores para mantener un espacio de búsqueda variado. Se comprueba, entonces, que la mejor solución es alcanzada cuando se combinan operaciones globales con mejoras individuales basadas en conocimiento específico del problema.

Y, finalmente, a nivel de pruebas numéricas, las pruebas con 40 sets diferentes de datos de entrada y repetidas 100 veces cada una, se ha demostrado que una alternativa de solución para el problema del corte de materiales unidimensional basada en algoritmos memético genera resultados numéricamente mejores (23% de mejora contra el algoritmo genético, 5% de mejora contra la población inicial) y estadísticamente a la par (Prueba Z, p-value 0.943688) comparados con el algoritmo genético.

### 11.3 Trabajos Futuros

Se proponen los siguientes trabajos futuros que extienden esta tesis:

- Modificar el algoritmo genético para utilizar otras variantes de los operadores.
- Extender el algoritmo memético para considerar una mayor cantidad de materiales disponibles para la producción.
- Adaptar los algoritmos implementados en un sistema de información completo, que incluya una base de datos de materiales, productos y órdenes de pedido.
- Analizar la eficiencia y eficacia del algoritmo memético, utilizando otros enfoques para la búsqueda local.
- Explorar las variables calibradas en nuevos espacios.
- Analizar la eficiencia de los algoritmos presentados, evaluando el uso de recursos computacionales y de ejecución.



# Referencias

CAPAceros [Web]. Retrieved 11/14/2019 from:

[https://www.capacero.cl/cap\\_acero/productos/construccion/aceros-para-hormigon-armado/barras-de-refuerzo-hormigon/2018-06-28/161248.html](https://www.capacero.cl/cap_acero/productos/construccion/aceros-para-hormigon-armado/barras-de-refuerzo-hormigon/2018-06-28/161248.html)

Ministerio de Economía y Finanzas. Informe de Actualización de Proyecciones Macroeconómicas [PDF file]. Retrieved from: [https://www.mef.gob.pe/contenidos/pol\\_econ/marco\\_macro/IAPM\\_2018\\_2021.pdf](https://www.mef.gob.pe/contenidos/pol_econ/marco_macro/IAPM_2018_2021.pdf)

Marti, R., Pardalos, P., Resende, M. 2018. *Handbook of Heuristics*, Springer. ISBN 978-3-319-07123-7

Yang CH, Nygard K 1993 Effects of initial population in genetic search for time constrained traveling salesman problems. In: *Proceedings of the ACM computer science conference*, pp 378–383 <https://dl.acm.org/doi/10.1145/170791.170875>

Smith A, Coit D, Baeck T, Fogel D, Michalewicz Z 1997 Penalty functions. In: Bäck T, Fogel DB, Michalewicz Z (eds) *Handbook on evolutionary computation*. Oxford University Press, New York, pp C5.2:1–C5.2:6 [ISBN:978-0-7503-0392-7](https://doi.org/10.1002/9780470027318.ch52)

Yeniay Ö 2005 Penalty function methods for constrained optimization with genetic algorithms. *Math Comput Appl* 10(1):45–56

Baker J (1987) Adaptive selection methods for genetic algorithms. In: Grefenstette J (ed) *International conference on genetic algorithms applications and their application*. Erlbaum Associates, pp 14–21 <https://dl.acm.org/doi/10.5555/645511.657075>

Goldberg D, Korb B, Deb K (1990) Messy genetic algorithms: motivation, analysis, and first results. *Complex Syst* 3:493–530 [ISSN 0891-2513](https://doi.org/10.1002/9780470027318.ch52)

Goldberg D, Lingle R (1985) Alleles, Loci and the traveling salesman problem. In: *Proceedings of the international conference on genetic algorithms*, pp 154–159 <https://dl.acm.org/doi/10.5555/645511.657095>

Goldberg D (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, New York [doi:10.5555/534133](https://doi.org/10.5555/534133)

Davis L (1985) Adaptive algorithms to epistatic domains. In: *Proceedings of the international conference on artificial intelligence*, pp 162–164 <https://dl.acm.org/doi/10.5555/1625135.1625164>

Banzhaf W (1990) The “Molecular” traveling salesman. *Biol Cybern* 64:7–14 <https://doi.org/10.1007/BF00203625>

Kuo T, Hwang S (1996) A genetic algorithm with disruptive selection. *IEEE Trans Syst Man Cybern* 26(2):299–307 [DOI: 10.1109/3477.485880](https://doi.org/10.1109/3477.485880)

Hutter M, Legg S (2006) Fitness uniform optimization. *IEEE Trans Evol Comput* 10(5): 568–589 [DOI: 10.1109/TEVC.2005.863127](https://doi.org/10.1109/TEVC.2005.863127)

Sareni B, Krahenbuhl L (1998) Fitness sharing and Niching methods revisited. *IEEE Trans Evol Comput* 2(3):97–106 [DOI: 10.1109/4235.735432](https://doi.org/10.1109/4235.735432)

Eshelman L, Schaffer J (1993) Real-coded genetic algorithms and interval-schemata. In: Whitley LD (ed) Foundations of genetic algorithms 2. Morgan Kaufmann, San Mateo, pp 187–202 [doi:10.1016/B978-0-08-094832-4.50018-0](https://doi.org/10.1016/B978-0-08-094832-4.50018-0)

Kantorovich, L. V. . Mathematical Methods of Organizing and Planning Production. Management Science 1960. [Doi:10.1287/mnsc.6.4.366](https://doi.org/10.1287/mnsc.6.4.366)

Dyckhoff, H. A typology of cutting and packing problems. European Journal of Operation Research. 1990. [Doi:10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K)

Cook, S. 1971 The complexity of theorem-proving procedures. In Proceedings of the 3rd ACM Symposium on the Theory of Computing, ACM, New York. [doi:10.1145/800157.805047](https://doi.org/10.1145/800157.805047)

Leeuwen, Jan van, ed. 1998. Handbook of Theoretical Computer Science. Vol. A, Algorithms and complexity. Amsterdam: Elsevier. [ISBN 0262720140](https://www.elsevier.com/locate/97800262720140).

Levin, L. 1973 Universal'nyie perebornyie zadachi (Universal search problems: in Russian). Problemy Peredachi Informatsii. [ISSN05552923](https://doi.org/10.1016/0013-788X(73)90001-1)

Agrawal, M., Kayal, N., & Saxena, N. 2004. PRIMES is in P. Annals of Mathematics. [doi:10.4007/annals.2004.160.781](https://doi.org/10.4007/annals.2004.160.781)

Karp, R. M. 1972. Reducibility among Combinatorial Problems. Complexity of Computer Computations. R. Miller and J. Thatcher, Eds. Plenum Press [doi:10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9)

Fortnow, L.. The status of the P versus NP problem. Communications of the ACM (2009). [doi:10.1145/1562164.1562186](https://doi.org/10.1145/1562164.1562186)

El-Ghazali Talbi. 2009. Meta-Heuristics: From Design to Implementation, New Jersey: Wiley. [ISBN: 978-0-470-27858-1](https://www.wiley.com/9780470278581)

Brucker, P. 2007. Scheduling Algorithms 5° Edición, New York: Springer. [ISBN: 354069515X](https://www.springer.com/978354069515X)

Mellouli, A, Mellouli, A, Masmoudi, F. An Innovative Genetic Algorithm for Multi-Objective Optimization of Two-Dimensional Cutting-Stock Problem. Applied Artificial Intelligence, March 2019. Doi: [10.1080/08839514.2019.1583857](https://doi.org/10.1080/08839514.2019.1583857)

Monzon, D, Cueva, R, Tupia, M, Bruzza, M. Aoo search algorithm for 2d-cutting problem in decorative ceramic production lines with defects. 11th International Conference on Agents and Artificial Intelligence, January 2019. DOI: [10.5220/0007346705470553](https://doi.org/10.5220/0007346705470553)

Palomares, N. Implementación de un algoritmo memético para optimizar la carga de hornos para la producción de sanitarios. 2018.

Lopez, I.A., Mora, J, Santos, C, Gonzalez-Mendoza, M, Monziel, C. Solving binary cutting stock with matheuristics using particle swarm optimization and simulated annealing. Soft Computing, July 2017. DOI: [10.1007/s00500-017-2666-8](https://doi.org/10.1007/s00500-017-2666-8)

Evtimov, G, Fidanova, S. Ant Colony Optimization Algorithm for 1D Cutting Stock Problem. Advanced Computing in Industrial Mathematics, January 2018. DOI: [10.1007/978-3-319-65530-7\\_3](https://doi.org/10.1007/978-3-319-65530-7_3)

Fathy, M, Osama, M, El-Mahallawy. Evolutionary-based hybrid algorithm for 2D cutting stock problem. 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), December 2015. DOI:[10.1109/IntelCIS.2015.7397260](https://doi.org/10.1109/IntelCIS.2015.7397260)

Pelaez, J, Yanez, A, Santos, E, Moncayo, L. An evolutionary metaheuristic for the 2d guillotine cut problem. The 20th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2016, July 2016. [doi:10.1007/s10100-013-0300-0](https://doi.org/10.1007/s10100-013-0300-0)

Bonnevay, S, Aubertin, P, Gavin, G. A genetic algorithm to solve a real 2-D cutting stock problem with setup cost in the paper industry. 2015 Annual Conference on Genetic and Evolutionary Computation, July 2015. DOI:[10.1145/2739480.2754660](https://doi.org/10.1145/2739480.2754660)

Goncalvez, J. A hybrid biased random key genetic algorithm for a production and cutting problem. 15th IFAC Symposium on Information Control Problems in Manufacturing: INCOM 2015, December 2015. DOI:[10.1016/j.ifacol.2015.06.130](https://doi.org/10.1016/j.ifacol.2015.06.130)

Oracle (2020). Lenguaje de Programación Java [Web]. Retrieved from: <https://www.oracle.com/technetwork/es/java/index.html>

Apache (2020). Apache Netbeans IDE [Windows] [Web]. Retrieved from: <https://netbeans.apache.org>

Bleck, K. (2004). *Extreme Programming Explained: Embrace Change* (2nd ed.). Addison-Wesley. ISBN-10: [9780321278654](https://www.isbn-international.org/product/9780321278654)

Montgomery, D. (2013). *Applied Statistics and Probability for Engineers* (6th ed.). Wiley. ISBN-10: 1118539710

Moscato P, Tinetti F (1992) Blending heuristics with a population-based approach: a memetic algorithm for the traveling salesman problem. Report 92–12, Universidad Nacional de La Plata [doi:10.1.1.31.1439](https://doi.org/10.1.1.31.1439)

Krasnogor N, Blackburne B, Burke E, Hirst J (2002) Multimeme Algorithms for Protein Structure Prediction. *Lecture Notes in Computer Science*, vol 2439. Springer, Berlin, pp 769–778 [doi:10.1007/3-540-45712-7\\_74](https://doi.org/10.1007/3-540-45712-7_74)

Anexo A: Calibración de la Población Inicial

- Calibración del tamaño de la población inicial

Aquí se presentan los resultados de fitness de la población inicial generada para cada set de datos de entrada, con diferentes valores de tamaño de población.

| Set | Tamaño de la Población Inicial |          |          |          |          |          |          |          |          |          |
|-----|--------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|     | 50                             | 100      | 150      | 200      | 250      | 300      | 350      | 400      | 450      | 500      |
| 10  | 0.116269                       | 0.120196 | 0.120196 | 0.120196 | 0.141184 | 0.141184 | 0.143043 | 0.141184 | 0.141184 | 0.137443 |
| 20  | 0.220264                       | 0.219118 | 0.227074 | 0.225948 | 0.225948 | 0.225948 | 0.225948 | 0.225948 | 0.227074 | 0.227074 |
| 30  | 0.215348                       | 0.215348 | 0.215348 | 0.215348 | 0.215348 | 0.215348 | 0.221815 | 0.221815 | 0.215348 | 0.215348 |
| 40  | 0.140584                       | 0.146245 | 0.158004 | 0.159969 | 0.175572 | 0.157785 | 0.171356 | 0.170931 | 0.175363 | 0.173258 |
| 5   | 0.32759                        | 0.330315 | 0.328073 | 0.365832 | 0.352535 | 0.347422 | 0.352982 | 0.358887 | 0.365832 | 0.357421 |
| 60  | 0.299562                       | 0.349506 | 0.352287 | 0.368355 | 0.358594 | 0.352287 | 0.307930 | 0.349506 | 0.352287 | 0.356278 |
| 7   | 0.399870                       | 0.384066 | 0.411263 | 0.410256 | 0.412641 | 0.415006 | 0.401042 | 0.410256 | 0.410256 | 0.410256 |
| 80  | 0.293425                       | 0.289006 | 0.291571 | 0.291571 | 0.300745 | 0.293887 | 0.298701 | 0.289707 | 0.295958 | 0.293656 |
| 90  | 0.540386                       | 0.535076 | 0.536478 | 0.540386 | 0.536478 | 0.540386 | 0.540386 | 0.540386 | 0.544965 | 0.541135 |
| 100 | 0.164542                       | 0.158356 | 0.200999 | 0.174570 | 0.164542 | 0.198497 | 0.168533 | 0.197366 | 0.201123 | 0.222947 |
| 110 | 0.320918                       | 0.320918 | 0.320918 | 0.320918 | 0.320918 | 0.323988 | 0.320918 | 0.323988 | 0.323988 | 0.320918 |
| 120 | 0.250747                       | 0.269847 | 0.260421 | 0.260421 | 0.267178 | 0.267178 | 0.269847 | 0.269847 | 0.265837 | 0.269847 |
| 130 | 0.194279                       | 0.245033 | 0.250986 | 0.243028 | 0.254902 | 0.264326 | 0.260317 | 0.254707 | 0.256651 | 0.254902 |
| 140 | 0.292929                       | 0.313433 | 0.313433 | 0.313433 | 0.316770 | 0.313433 | 0.320623 | 0.324252 | 0.330544 | 0.339752 |
| 150 | 0.539489                       | 0.542543 | 0.549752 | 0.561596 | 0.556836 | 0.585708 | 0.561596 | 0.566349 | 0.559568 | 0.585708 |
| 160 | 0.432432                       | 0.431529 | 0.432432 | 0.432432 | 0.432432 | 0.432432 | 0.436019 | 0.43956  | 0.43956  | 0.495881 |
| 170 | 0.459603                       | 0.481707 | 0.464004 | 0.465409 | 0.474903 | 0.464004 | 0.466806 | 0.476252 | 0.474903 | 0.474903 |
| 18  | 0.344170                       | 0.331046 | 0.353591 | 0.356966 | 0.353055 | 0.348008 | 0.348008 | 0.362398 | 0.362398 | 0.36205  |
| 190 | 0.303136                       | 0.305976 | 0.317585 | 0.303136 | 0.324578 | 0.324578 | 0.317585 | 0.315609 | 0.321149 | 0.321149 |
| 200 | 0.249598                       | 0.188361 | 0.223373 | 0.224806 | 0.233157 | 0.227373 | 0.242970 | 0.234136 | 0.231754 | 0.241603 |
| 21  | 0.394370                       | 0.379695 | 0.418464 | 0.401170 | 0.398213 | 0.404097 | 0.404097 | 0.394370 | 0.418464 | 0.394369 |
| 22  | 0.16287                        | 0.167610 | 0.179229 | 0.188294 | 0.181514 | 0.181514 | 0.169726 | 0.181514 | 0.181514 | 0.174621 |
| 230 | 0.073826                       | 0.15660  | 0.134924 | 0.117096 | 0.148984 | 0.147059 | 0.134924 | 0.147059 | 0.15071  | 0.1566   |
| 240 | 0.202204                       | 0.219043 | 0.245582 | 0.215363 | 0.233425 | 0.226301 | 0.236764 | 0.219043 | 0.243869 | 0.238509 |
| 25  | 0.047367                       | 0.047367 | 0.047367 | 0.045395 | 0.047367 | 0.045395 | 0.045395 | 0.047367 | 0.045395 | 0.045395 |
| 260 | 0.363104                       | 0.364964 | 0.371387 | 0.375897 | 0.369565 | 0.371387 | 0.372294 | 0.384724 | 0.371387 | 0.382979 |
| 270 | 0.212429                       | 0.215251 | 0.212429 | 0.212429 | 0.218405 | 0.236806 | 0.218405 | 0.218386 | 0.218386 | 0.218386 |
| 280 | 0.061873                       | 0.085084 | 0.090104 | 0.071867 | 0.076868 | 0.085246 | 0.085084 | 0.085084 | 0.080336 | 0.096502 |
| 290 | 0.872014                       | 0.864636 | 0.869792 | 0.869792 | 0.869792 | 0.872014 | 0.869792 | 0.872014 | 0.872014 | 0.872014 |
| 300 | 0.496323                       | 0.458942 | 0.465468 | 0.477939 | 0.465342 | 0.464074 | 0.46280  | 0.481651 | 0.473562 | 0.469234 |

|   |
|---|
| 310.0819870.0819870.0819870.0819870.0929430.0929430.0929430.0929430.1193050.127284                          |
| 320.5367790.5930590.5956490.5948170.5948170.5956490.5948170.6061580.5913710.598451                          |
| 33 - - - - - - - - -0.0083 - -<br>0.0136690.0136690.0241920.0136690.0241920.0174630.013669 0.0136690.013669 |
| 340.3496650.3445030.3610320.3591950.3628570.3610320.3610320.3610320.3591950.373771                          |
| 350.2003820.2636990.2583650.2662120.2706920.2687070.2662120.2662120.2687070.268707                          |
| 360.1015990.1124530.0996940.0952160.097567 0.088740.1124540.1062240.1062240.106224                          |
| 370.2605710.2795510.2795510.2795510.2795510.2795510.2651270.2795510.2795510.265127                          |
| 380.2839810.2942580.2942580.3169720.3009480.2942580.2942580.2942580.2905330.304245                          |
| 390.123474 0.126620.1234740.1234740.1234740.1458210.1269680.1388120.162846 0.15781                          |
| 40 0.17530.1852010.190234 0.17530.1902340.1852010.1902340.1972190.2020940.197219                            |



Anexo B: Calibración del algoritmo genético

- Calibración de la probabilidad de crossover (tasa de recombinación)

| Set | 5%                       |              |                                    | 15%                      |              |                                    | 25%                      |              |                                    |
|-----|--------------------------|--------------|------------------------------------|--------------------------|--------------|------------------------------------|--------------------------|--------------|------------------------------------|
|     | Función Fitness          |              | Porcent<br>aje de<br>Mejora<br>(%) | Función Fitness          |              | Porcent<br>aje de<br>Mejora<br>(%) | Función Fitness          |              | Porcent<br>aje de<br>Mejora<br>(%) |
|     | Poblaci<br>ón<br>Inicial | Genétic<br>o |                                    | Poblaci<br>ón<br>Inicial | Genétic<br>o |                                    | Poblaci<br>ón<br>Inicial | Genétic<br>o |                                    |
| 1   | 0.120196                 | 0.118236     | -1.63%                             | 0.120196                 | 0.143043     | 19.01%                             | 0.120196                 | 0.137443     | 14.35%                             |
| 2   | 0.227074                 | 0.227074     | 0.00%                              | 0.227074                 | 0.225948     | -0.50%                             | 0.227074                 | 0.225948     | -0.50%                             |
| 3   | 0.215348                 | 0.215348     | 0.00%                              | 0.215348                 | 0.221815     | 3.00%                              | 0.215348                 | 0.215348     | 0.00%                              |
| 4   | 0.158004                 | 0.175363     | 10.99%                             | 0.158004                 | 0.162358     | 2.76%                              | 0.158004                 | 0.171356     | 8.45%                              |
| 5   | 0.328073                 | 0.352236     | 7.37%                              | 0.328073                 | 0.323381     | -1.43%                             | 0.328073                 | 0.322892     | -1.58%                             |
| 6   | 0.352287                 | 0.30793      | -12.59%                            | 0.352287                 | 0.34656      | -1.63%                             | 0.352287                 | 0.359001     | 1.91%                              |
| 7   | 0.411263                 | 0.410256     | -0.24%                             | 0.411263                 | 0.402339     | -2.17%                             | 0.411263                 | 0.410256     | -0.24%                             |
| 8   | 0.291571                 | 0.289006     | -0.88%                             | 0.291571                 | 0.291571     | 0.00%                              | 0.291571                 | 0.293887     | 0.79%                              |
| 9   | 0.536478                 | 0.536478     | 0.00%                              | 0.536478                 | 0.536478     | 0.00%                              | 0.536478                 | 0.552436     | 2.97%                              |
| 10  | 0.200999                 | 0.160892     | -19.95%                            | 0.200999                 | 0.17457      | -13.15%                            | 0.200999                 | 0.17457      | -13.15%                            |
| 11  | 0.320918                 | 0.320918     | 0.00%                              | 0.320918                 | 0.323988     | 0.96%                              | 0.320918                 | 0.324663     | 1.17%                              |
| 12  | 0.260421                 | 0.254924     | -2.11%                             | 0.260421                 | 0.271042     | 4.08%                              | 0.260421                 | 0.28033      | 7.64%                              |
| 13  | 0.250986                 | 0.250789     | -0.08%                             | 0.250986                 | 0.194279     | -22.59%                            | 0.250986                 | 0.250789     | -0.08%                             |
| 14  | 0.313433                 | 0.313433     | 0.00%                              | 0.313433                 | 0.313433     | 0.00%                              | 0.313433                 | 0.32299      | 3.05%                              |
| 15  | 0.549752                 | 0.566349     | 3.02%                              | 0.549752                 | 0.554764     | 0.91%                              | 0.549752                 | 0.547614     | -0.39%                             |
| 16  | 0.432432                 | 0.43956      | 1.65%                              | 0.432432                 | 0.432432     | 0.00%                              | 0.432432                 | 0.432432     | 0.00%                              |
| 17  | 0.464004                 | 0.465409     | 0.30%                              | 0.464004                 | 0.467919     | 0.84%                              | 0.464004                 | 0.474903     | 2.35%                              |
| 18  | 0.353591                 | 0.356966     | 0.95%                              | 0.353591                 | 0.348008     | -1.58%                             | 0.353591                 | 0.362398     | 2.49%                              |
| 19  | 0.317585                 | 0.321149     | 1.12%                              | 0.317585                 | 0.321149     | 1.12%                              | 0.317585                 | 0.317585     | 0.00%                              |
| 20  | 0.223373                 | 0.225949     | 1.15%                              | 0.223373                 | 0.224806     | 0.64%                              | 0.223373                 | 0.225949     | 1.15%                              |
| 21  | 0.418464                 | 0.394369     | -5.76%                             | 0.418464                 | 0.418464     | 0.00%                              | 0.418464                 | 0.418464     | 0.00%                              |
| 22  | 0.179229                 | 0.186046     | 3.80%                              | 0.179229                 | 0.167375     | -6.61%                             | 0.179229                 | 0.16761      | -6.48%                             |



|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 23 | 0.134924  | 0.144931  | 7.42%   | 0.134924  | 0.138679  | 2.78%   | 0.134924  | 0.123256  | -8.65%  |
| 24 | 0.245582  | 0.233425  | -4.95%  | 0.245582  | 0.220688  | -10.14% | 0.245582  | 0.233425  | -4.95%  |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.382979  | 3.12%   | 0.371387  | 0.384724  | 3.59%   | 0.371387  | 0.375897  | 1.21%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.215251  | 1.33%   | 0.212429  | 0.218386  | 2.80%   |
| 28 | 0.090104  | 0.080336  | -10.84% | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.085246  | -5.39%  |
| 29 | 0.869792  | 0.864636  | -0.59%  | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.490647  | 5.41%   | 0.465468  | 0.46786   | 0.51%   | 0.465468  | 0.469234  | 0.81%   |
| 31 | 0.081987  | 0.081987  | 0.00%   | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.092943  | 13.36%  |
| 32 | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.600406  | 0.80%   | 0.595649  | 0.596229  | 0.10%   |
| 33 | -0.024192 | -0.024192 | 0.00%   | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.01367  | -43.49% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.263699  | 2.06%   | 0.258365  | 0.263446  | 1.97%   | 0.258365  | 0.270692  | 4.77%   |
| 36 | 0.099694  | 0.103917  | 4.24%   | 0.099694  | 0.097567  | -2.13%  | 0.099694  | 0.106224  | 6.55%   |
| 37 | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.279551  | 0.00%   |
| 38 | 0.294258  | 0.293818  | -0.15%  | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.294258  | 0.00%   |
| 39 | 0.123474  | 0.123474  | 0.00%   | 0.123474  | 0.154419  | 25.06%  | 0.123474  | 0.154419  | 25.06%  |
| 40 | 0.190234  | 0.185201  | -2.65%  | 0.190234  | 0.190234  | 0.00%   | 0.190234  | 0.186759  | -1.83%  |

| Set | 35%               |          |                          | 45%               |          |                          | 55%               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.143043 | 19.01%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.225948 | -0.50%                   |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.221815 | 3.00%                    |
| 4   | 0.158004          | 0.171356 | 8.45%                    | 0.158004          | 0.181818 | 15.07%                   | 0.158004          | 0.173258 | 9.65%                    |
| 5   | 0.328073          | 0.331271 | 0.97%                    | 0.328073          | 0.352833 | 7.55%                    | 0.328073          | 0.331112 | 0.93%                    |
| 6   | 0.352287          | 0.352287 | 0.00%                    | 0.352287          | 0.352287 | 0.00%                    | 0.352287          | 0.353393 | 0.31%                    |
| 7   | 0.411263          | 0.398431 | -3.12%                   | 0.411263          | 0.449641 | 9.33%                    | 0.411263          | 0.39987  | -2.77%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 8  | 0.291571  | 0.291571  | 0.00%   | 0.291571  | 0.295269  | 1.27%   | 0.291571  | 0.291106  | -0.16%  |
| 9  | 0.536478  | 0.539007  | 0.47%   | 0.536478  | 0.552199  | 2.93%   | 0.536478  | 0.552199  | 2.93%   |
| 10 | 0.200999  | 0.160892  | -19.95% | 0.200999  | 0.201123  | 0.06%   | 0.200999  | 0.155933  | -22.42% |
| 11 | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.323988  | 0.96%   |
| 12 | 0.260421  | 0.271174  | 4.13%   | 0.260421  | 0.267178  | 2.59%   | 0.260421  | 0.281619  | 8.14%   |
| 13 | 0.250986  | 0.246829  | -1.66%  | 0.250986  | 0.252949  | 0.78%   | 0.250986  | 0.260509  | 3.79%   |
| 14 | 0.313433  | 0.330544  | 5.46%   | 0.313433  | 0.309309  | -1.32%  | 0.313433  | 0.34462   | 9.95%   |
| 15 | 0.549752  | 0.583792  | 6.19%   | 0.549752  | 0.564817  | 2.74%   | 0.549752  | 0.549752  | 0.00%   |
| 16 | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.489149  | 13.12%  |
| 17 | 0.464004  | 0.468196  | 0.90%   | 0.464004  | 0.47906   | 3.24%   | 0.464004  | 0.474903  | 2.35%   |
| 18 | 0.353591  | 0.356966  | 0.95%   | 0.353591  | 0.357143  | 1.00%   | 0.353591  | 0.356436  | 0.80%   |
| 19 | 0.317585  | 0.303136  | -4.55%  | 0.317585  | 0.315403  | -0.69%  | 0.317585  | 0.321149  | 1.12%   |
| 20 | 0.223373  | 0.238441  | 6.75%   | 0.223373  | 0.235947  | 5.63%   | 0.223373  | 0.24812   | 11.08%  |
| 21 | 0.418464  | 0.40117   | -4.13%  | 0.418464  | 0.418464  | 0.00%   | 0.418464  | 0.394369  | -5.76%  |
| 22 | 0.179229  | 0.167375  | -6.61%  | 0.179229  | 0.174621  | -2.57%  | 0.179229  | 0.179229  | 0.00%   |
| 23 | 0.134924  | 0.150901  | 11.84%  | 0.134924  | 0.1566    | 16.07%  | 0.134924  | 0.144931  | 7.42%   |
| 24 | 0.245582  | 0.228095  | -7.12%  | 0.245582  | 0.243697  | -0.77%  | 0.245582  | 0.226301  | -7.85%  |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.378571  | 1.93%   | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.369565  | -0.49%  |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.085246  | -5.39%  | 0.090104  | 0.080336  | -10.84% |
| 29 | 0.869792  | 0.864636  | -0.59%  | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.479263  | 2.96%   | 0.465468  | 0.475516  | 2.16%   | 0.465468  | 0.469857  | 0.94%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.084613  | 3.20%   | 0.081987  | 0.084613  | 3.20%   |
| 32 | 0.595649  | 0.594817  | -0.14%  | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.601134  | 0.92%   |
| 33 | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.017463 | -27.81% |
| 34 | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.373771  | 3.53%   | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.255282  | -1.19%  | 0.258365  | 0.260914  | 0.99%   | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.102022  | 2.34%   | 0.099694  | 0.099694  | 0.00%   | 0.099694  | 0.102022  | 2.34%   |

|    |          |          |        |          |          |        |          |          |        |
|----|----------|----------|--------|----------|----------|--------|----------|----------|--------|
| 37 | 0.279551 | 0.279551 | 0.00%  | 0.279551 | 0.265127 | -5.16% | 0.279551 | 0.279551 | 0.00%  |
| 38 | 0.294258 | 0.28705  | -2.45% | 0.294258 | 0.300948 | 2.27%  | 0.294258 | 0.294258 | 0.00%  |
| 39 | 0.123474 | 0.138812 | 12.42% | 0.123474 | 0.138815 | 12.42% | 0.123474 | 0.138812 | 12.42% |
| 40 | 0.190234 | 0.190234 | 0.00%  | 0.190234 | 0.205588 | 8.07%  | 0.190234 | 0.198907 | 4.56%  |

| Set | 65%               |          |                          | 75%               |          |                          | 85%               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.120196 | 0.00%                    | 0.120196          | 0.137443 | 14.35%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.225948 | -0.50%                   |
| 3   | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.173258 | 9.65%                    | 0.158004          | 0.157785 | -0.14%                   | 0.158004          | 0.170931 | 8.18%                    |
| 5   | 0.328073          | 0.3327   | 1.41%                    | 0.328073          | 0.352982 | 7.59%                    | 0.328073          | 0.357421 | 8.95%                    |
| 6   | 0.352287          | 0.361837 | 2.71%                    | 0.352287          | 0.382515 | 8.58%                    | 0.352287          | 0.362776 | 2.98%                    |
| 7   | 0.411263          | 0.404916 | -1.54%                   | 0.411263          | 0.408867 | -0.58%                   | 0.411263          | 0.410256 | -0.24%                   |
| 8   | 0.291571          | 0.291571 | 0.00%                    | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.536478 | 0.00%                    | 0.536478          | 0.540386 | 0.73%                    |
| 10  | 0.200999          | 0.15721  | -21.79%                  | 0.200999          | 0.155933 | -22.42%                  | 0.200999          | 0.158356 | -21.22%                  |
| 11  | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.323988 | 0.96%                    |
| 12  | 0.260421          | 0.269714 | 3.57%                    | 0.260421          | 0.267178 | 2.59%                    | 0.260421          | 0.265837 | 2.08%                    |
| 13  | 0.250986          | 0.258778 | 3.10%                    | 0.250986          | 0.245033 | -2.37%                   | 0.250986          | 0.262613 | 4.63%                    |
| 14  | 0.313433          | 0.32299  | 3.05%                    | 0.313433          | 0.32299  | 3.05%                    | 0.313433          | 0.324252 | 3.45%                    |
| 15  | 0.549752          | 0.564817 | 2.74%                    | 0.549752          | 0.554665 | 0.89%                    | 0.549752          | 0.556836 | 1.29%                    |
| 16  | 0.432432          | 0.432432 | 0.00%                    | 0.432432          | 0.436019 | 0.83%                    | 0.432432          | 0.432432 | 0.00%                    |
| 17  | 0.464004          | 0.47906  | 3.24%                    | 0.464004          | 0.472186 | 1.76%                    | 0.464004          | 0.481707 | 3.82%                    |
| 18  | 0.353591          | 0.356966 | 0.95%                    | 0.353591          | 0.362398 | 2.49%                    | 0.353591          | 0.356966 | 0.95%                    |
| 19  | 0.317585          | 0.321149 | 1.12%                    | 0.317585          | 0.321149 | 1.12%                    | 0.317585          | 0.317585 | 0.00%                    |
| 20  | 0.223373          | 0.241603 | 8.16%                    | 0.223373          | 0.245419 | 9.87%                    | 0.223373          | 0.238855 | 6.93%                    |
| 21  | 0.418464          | 0.405204 | -3.17%                   | 0.418464          | 0.40117  | -4.13%                   | 0.418464          | 0.410953 | -1.79%                   |

|    |           |           |         |           |           |        |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|--------|-----------|-----------|---------|
| 22 | 0.179229  | 0.179229  | 0.00%   | 0.179229  | 0.174389  | -2.70% | 0.179229  | 0.188294  | 5.06%   |
| 23 | 0.134924  | 0.1566    | 16.07%  | 0.134924  | 0.134924  | 0.00%  | 0.134924  | 0.123256  | -8.65%  |
| 24 | 0.245582  | 0.226301  | -7.85%  | 0.245582  | 0.243697  | -0.77% | 0.245582  | 0.233425  | -4.95%  |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.384724  | 3.59%   | 0.371387  | 0.382979  | 3.12%  | 0.371387  | 0.378571  | 1.93%   |
| 27 | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.218405  | 2.81%  | 0.212429  | 0.218386  | 2.80%   |
| 28 | 0.090104  | 0.075373  | -16.35% | 0.090104  | 0.085084  | -5.57% | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.872014  | 0.26%  | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.471716  | 1.34%   | 0.465468  | 0.479263  | 2.96%  | 0.465468  | 0.479862  | 3.09%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.081987  | 0.00%  | 0.081987  | 0.119171  | 45.35%  |
| 32 | 0.595649  | 0.593731  | -0.32%  | 0.595649  | 0.594817  | -0.14% | 0.595649  | 0.595649  | 0.00%   |
| 33 | -0.024192 | -0.024192 | 0.00%   | -0.024192 | -0.024192 | 0.00%  | -0.024192 | -0.013669 | -43.50% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.359195  | -0.51% | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.256055  | -0.89%  | 0.258365  | 0.266212  | 3.04%  | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.102022  | 2.34%   | 0.099694  | 0.106224  | 6.55%  | 0.099694  | 0.10831   | 8.64%   |
| 37 | 0.279551  | 0.327396  | 17.11%  | 0.279551  | 0.279551  | 0.00%  | 0.279551  | 0.279551  | 0.00%   |
| 38 | 0.294258  | 0.300948  | 2.27%   | 0.294258  | 0.294258  | 0.00%  | 0.294258  | 0.294258  | 0.00%   |
| 39 | 0.123474  | 0.138812  | 12.42%  | 0.123474  | 0.145533  | 17.87% | 0.123474  | 0.145821  | 18.10%  |
| 40 | 0.190234  | 0.205588  | 8.07%   | 0.190234  | 0.190234  | 0.00%  | 0.190234  | 0.190234  | 0.00%   |

- Calibración de la probabilidad de mutación

| Set | 5%                |          |                          | 6%                |          |                          | 7%                |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.141184 | 17.46%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.171355 | 8.45%                    | 0.158004          | 0.173258 | 9.65%                    | 0.158004          | 0.175573 | 11.12%                   |
| 5   | 0.328073          | 0.332858 | 1.46%                    | 0.328073          | 0.33742  | 2.85%                    | 0.328073          | 0.356979 | 8.81%                    |
| 6   | 0.352287          | 0.347966 | -1.23%                   | 0.352287          | 0.356278 | 1.13%                    | 0.352287          | 0.347966 | -1.23%                   |
| 7   | 0.411263          | 0.414013 | 0.67%                    | 0.411263          | 0.448922 | 9.16%                    | 0.411263          | 0.410256 | -0.24%                   |
| 8   | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.289006 | -0.88%                   |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.547281 | 2.01%                    | 0.536478          | 0.540386 | 0.73%                    |
| 10  | 0.200999          | 0.168533 | -16.15%                  | 0.200999          | 0.17457  | -13.15%                  | 0.200999          | 0.160892 | -19.95%                  |
| 11  | 0.320918          | 0.320918 | 0.00%                    | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.323988 | 0.96%                    |
| 12  | 0.260421          | 0.254924 | -2.11%                   | 0.260421          | 0.252145 | -3.18%                   | 0.260421          | 0.267178 | 2.59%                    |
| 13  | 0.250986          | 0.250789 | -0.08%                   | 0.250986          | 0.260509 | 3.79%                    | 0.250986          | 0.250592 | -0.16%                   |
| 14  | 0.313433          | 0.313433 | 0.00%                    | 0.313433          | 0.32299  | 3.05%                    | 0.313433          | 0.324072 | 3.39%                    |
| 15  | 0.549752          | 0.561596 | 2.15%                    | 0.549752          | 0.549752 | 0.00%                    | 0.549752          | 0.549752 | 0.00%                    |
| 16  | 0.432432          | 0.43956  | 1.65%                    | 0.432432          | 0.432432 | 0.00%                    | 0.432432          | 0.432432 | 0.00%                    |
| 17  | 0.464004          | 0.47906  | 3.24%                    | 0.464004          | 0.474903 | 2.35%                    | 0.464004          | 0.474903 | 2.35%                    |
| 18  | 0.353591          | 0.353591 | 0.00%                    | 0.353591          | 0.353591 | 0.00%                    | 0.353591          | 0.353591 | 0.00%                    |
| 19  | 0.317585          | 0.321149 | 1.12%                    | 0.317585          | 0.324578 | 2.20%                    | 0.317585          | 0.303136 | -4.55%                   |
| 20  | 0.223373          | 0.224806 | 0.64%                    | 0.223373          | 0.233157 | 4.38%                    | 0.223373          | 0.224806 | 0.64%                    |
| 21  | 0.418464          | 0.404097 | -3.43%                   | 0.418464          | 0.404097 | -3.43%                   | 0.418464          | 0.432221 | 3.29%                    |
| 22  | 0.179229          | 0.181287 | 1.15%                    | 0.179229          | 0.186046 | 3.80%                    | 0.179229          | 0.181514 | 1.27%                    |
| 23  | 0.134924          | 0.144931 | 7.42%                    | 0.134924          | 0.131136 | -2.81%                   | 0.134924          | 0.136905 | 1.47%                    |
| 24  | 0.245582          | 0.226121 | -7.92%                   | 0.245582          | 0.229523 | -6.54%                   | 0.245582          | 0.226301 | -7.85%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.372294  | 0.24%   | 0.371387  | 0.364964  | -1.73%  | 0.371387  | 0.371387  | 0.00%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.080336  | -10.84% | 0.090104  | 0.094592  | 4.98%   | 0.090104  | 0.080336  | -10.84% |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.873096  | 0.38%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.467483  | 0.43%   | 0.465468  | 0.481532  | 3.45%   | 0.465468  | 0.464074  | -0.30%  |
| 31 | 0.081987  | 0.084613  | 3.20%   | 0.081987  | 0.081987  | 0.00%   | 0.081987  | 0.117961  | 43.88%  |
| 32 | 0.595649  | 0.593059  | -0.43%  | 0.595649  | 0.594817  | -0.14%  | 0.595649  | 0.602101  | 1.08%   |
| 33 | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.008299 | -65.70% | -0.024192 | -0.008299 | -65.70% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.263446  | 1.97%   | 0.258365  | 0.258109  | -0.10%  | 0.258365  | 0.258109  | -0.10%  |
| 36 | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.10831   | 8.64%   | 0.099694  | 0.101599  | 1.91%   |
| 37 | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.265127  | -5.16%  | 0.279551  | 0.328184  | 17.40%  |
| 38 | 0.294258  | 0.293818  | -0.15%  | 0.294258  | 0.292566  | -0.58%  | 0.294258  | 0.304245  | 3.39%   |
| 39 | 0.123474  | 0.133531  | 8.15%   | 0.123474  | 0.137045  | 10.99%  | 0.123474  | 0.137045  | 10.99%  |
| 40 | 0.190234  | 0.190234  | 0.00%   | 0.190234  | 0.190234  | 0.00%   | 0.190234  | 0.190234  | 0.00%   |

| Set | 8%                |          |                          | 9%                |          |                          | 10%               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.143043 | 19.01%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.225948 | -0.50%                   |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.166881 | 5.62%                    | 0.158004          | 0.160187 | 1.38%                    | 0.158004          | 0.175573 | 11.12%                   |
| 5   | 0.328073          | 0.363101 | 10.68%                   | 0.328073          | 0.352982 | 7.59%                    | 0.328073          | 0.352684 | 7.50%                    |
| 6   | 0.352287          | 0.379861 | 7.83%                    | 0.352287          | 0.357234 | 1.40%                    | 0.352287          | 0.349506 | -0.79%                   |
| 7   | 0.411263          | 0.410256 | -0.24%                   | 0.411263          | 0.410256 | -0.24%                   | 0.411263          | 0.398168 | -3.18%                   |
| 8   | 0.291571          | 0.289707 | -0.64%                   | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.295958 | 1.50%                    |
| 9   | 0.536478          | 0.539007 | 0.47%                    | 0.536478          | 0.533664 | -0.52%                   | 0.536478          | 0.540386 | 0.73%                    |

|    |           |           |          |           |           |         |           |           |         |
|----|-----------|-----------|----------|-----------|-----------|---------|-----------|-----------|---------|
| 10 | 0.200999  | 0.201123  | 0.06%    | 0.200999  | 0.158356  | -21.22% | 0.200999  | 0.17457   | -13.15% |
| 11 | 0.320918  | 0.323988  | 0.96%    | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.320918  | 0.00%   |
| 12 | 0.260421  | 0.269847  | 3.62%    | 0.260421  | 0.265837  | 2.08%   | 0.260421  | 0.261782  | 0.52%   |
| 13 | 0.250986  | 0.258778  | 3.10%    | 0.250986  | 0.249012  | -0.79%  | 0.250986  | 0.250986  | 0.00%   |
| 14 | 0.313433  | 0.313433  | 0.00%    | 0.313433  | 0.31306   | -0.12%  | 0.313433  | 0.324252  | 3.45%   |
| 15 | 0.549752  | 0.549752  | 0.00%    | 0.549752  | 0.556836  | 1.29%   | 0.549752  | 0.564817  | 2.74%   |
| 16 | 0.432432  | 0.43956   | 1.65%    | 0.432432  | 0.432432  | 0.00%   | 0.432432  | 0.43956   | 1.65%   |
| 17 | 0.464004  | 0.468196  | 0.90%    | 0.464004  | 0.47906   | 3.24%   | 0.464004  | 0.466806  | 0.60%   |
| 18 | 0.353591  | 0.357143  | 1.00%    | 0.353591  | 0.344354  | -2.61%  | 0.353591  | 0.362398  | 2.49%   |
| 19 | 0.317585  | 0.303136  | -4.55%   | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.321149  | 1.12%   |
| 20 | 0.223373  | 0.235947  | 5.63%    | 0.223373  | 0.228792  | 2.43%   | 0.223373  | 0.233157  | 4.38%   |
| 21 | 0.418464  | 0.39437   | -5.76%   | 0.418464  | 0.432221  | 3.29%   | 0.418464  | 0.404097  | -3.43%  |
| 22 | 0.179229  | 0.174621  | -2.57%   | 0.179229  | 0.192751  | 7.54%   | 0.179229  | 0.183787  | 2.54%   |
| 23 | 0.134924  | 0.147059  | 8.99%    | 0.134924  | 0.147059  | 8.99%   | 0.134924  | 0.136905  | 1.47%   |
| 24 | 0.245582  | 0.250675  | 2.07%    | 0.245582  | 0.250675  | 2.07%   | 0.245582  | 0.22087   | -10.06% |
| 25 | -0.047367 | -0.045395 | -4.16%   | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.378571  | 1.93%    | 0.371387  | 0.367733  | -0.98%  | 0.371387  | 0.375     | 0.97%   |
| 27 | 0.212429  | 0.218386  | 2.80%    | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.090104  | 0.00%    | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.085084  | -5.57%  |
| 29 | 0.869792  | 0.869792  | 0.00%    | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.869792  | 0.00%   |
| 30 | 0.465468  | 0.479143  | 2.94%    | 0.465468  | 0.475395  | 2.13%   | 0.465468  | 0.477939  | 2.68%   |
| 31 | 0.081987  | 0.081987  | 0.00%    | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.123313  | 50.41%  |
| 32 | 0.595649  | 0.595649  | 0.00%    | 0.595649  | 0.590522  | -0.86%  | 0.595649  | 0.595649  | 0.00%   |
| 33 | -0.024192 | 0.002308  | -109.54% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.008299 | -65.70% |
| 34 | 0.361032  | 0.347763  | -3.68%   | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.263699  | 2.06%    | 0.258365  | 0.258365  | 0.00%   | 0.258365  | 0.260914  | 0.99%   |
| 36 | 0.099694  | 0.103917  | 4.24%    | 0.099694  | 0.102022  | 2.34%   | 0.099694  | 0.102022  | 2.34%   |
| 37 | 0.279551  | 0.279551  | 0.00%    | 0.279551  | 0.265127  | -5.16%  | 0.279551  | 0.260571  | -6.79%  |
| 38 | 0.294258  | 0.293818  | -0.15%   | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.294258  | 0.00%   |

|    |          |          |       |          |          |       |          |          |        |
|----|----------|----------|-------|----------|----------|-------|----------|----------|--------|
| 39 | 0.123474 | 0.126968 | 2.83% | 0.123474 | 0.12662  | 2.55% | 0.123474 | 0.159496 | 29.17% |
| 40 | 0.190234 | 0.190234 | 0.00% | 0.190234 | 0.190234 | 0.00% | 0.190234 | 0.185201 | -2.65% |





- Calibración de la cantidad de generaciones sin mejora

| Set | 1                    |          |                                    | 2                    |          |                                    | 3                    |          |                                    |
|-----|----------------------|----------|------------------------------------|----------------------|----------|------------------------------------|----------------------|----------|------------------------------------|
|     | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) |
|     | Población<br>Inicial | Genético |                                    | Población<br>Inicial | Genético |                                    | Población<br>Inicial | Genético |                                    |
| 1   | 0.120196             | 0.141184 | 17.46%                             | 0.120196             | 0.137443 | 14.35%                             | 0.120196             | 0.144893 | 20.55%                             |
| 2   | 0.227074             | 0.227074 | 0.00%                              | 0.227074             | 0.227074 | 0.00%                              | 0.227074             | 0.227074 | 0.00%                              |
| 3   | 0.215348             | 0.215348 | 0.00%                              | 0.215348             | 0.215348 | 0.00%                              | 0.215348             | 0.215348 | 0.00%                              |
| 4   | 0.158004             | 0.15559  | -1.53%                             | 0.158004             | 0.175573 | 11.12%                             | 0.158004             | 0.179539 | 13.63%                             |
| 5   | 0.328073             | 0.358887 | 9.39%                              | 0.328073             | 0.363245 | 10.72%                             | 0.328073             | 0.366118 | 11.60%                             |
| 6   | 0.352287             | 0.35201  | -0.08%                             | 0.352287             | 0.352287 | 0.00%                              | 0.352287             | 0.369276 | 4.82%                              |
| 7   | 0.411263             | 0.407598 | -0.89%                             | 0.411263             | 0.410256 | -0.24%                             | 0.411263             | 0.443344 | 7.80%                              |
| 8   | 0.291571             | 0.289707 | -0.64%                             | 0.291571             | 0.298701 | 2.45%                              | 0.291571             | 0.298701 | 2.45%                              |
| 9   | 0.536478             | 0.540386 | 0.73%                              | 0.536478             | 0.547281 | 2.01%                              | 0.536478             | 0.658953 | 22.83%                             |
| 10  | 0.200999             | 0.171008 | -14.92%                            | 0.200999             | 0.210851 | 4.90%                              | 0.200999             | 0.213276 | 6.11%                              |
| 11  | 0.320918             | 0.320918 | 0.00%                              | 0.320918             | 0.323988 | 0.96%                              | 0.320918             | 0.320918 | 0.00%                              |
| 12  | 0.260421             | 0.261782 | 0.52%                              | 0.260421             | 0.267178 | 2.59%                              | 0.260421             | 0.269714 | 3.57%                              |
| 13  | 0.250986             | 0.249012 | -0.79%                             | 0.250986             | 0.264516 | 5.39%                              | 0.250986             | 0.254902 | 1.56%                              |
| 14  | 0.313433             | 0.324252 | 3.45%                              | 0.313433             | 0.324252 | 3.45%                              | 0.313433             | 0.330544 | 5.46%                              |
| 15  | 0.549752             | 0.564817 | 2.74%                              | 0.549752             | 0.585708 | 6.54%                              | 0.549752             | 0.561596 | 2.15%                              |
| 16  | 0.432432             | 0.432432 | 0.00%                              | 0.432432             | 0.432432 | 0.00%                              | 0.432432             | 0.43956  | 1.65%                              |
| 17  | 0.464004             | 0.472186 | 1.76%                              | 0.464004             | 0.47906  | 3.24%                              | 0.464004             | 0.468196 | 0.90%                              |
| 18  | 0.353591             | 0.356966 | 0.95%                              | 0.353591             | 0.356789 | 0.90%                              | 0.353591             | 0.362398 | 2.49%                              |
| 19  | 0.317585             | 0.321149 | 1.12%                              | 0.317585             | 0.324578 | 2.20%                              | 0.317585             | 0.324578 | 2.20%                              |
| 20  | 0.223373             | 0.25227  | 12.94%                             | 0.223373             | 0.234554 | 5.01%                              | 0.223373             | 0.258998 | 15.95%                             |
| 21  | 0.418464             | 0.39437  | -5.76%                             | 0.418464             | 0.408092 | -2.48%                             | 0.418464             | 0.39437  | -5.76%                             |
| 22  | 0.179229             | 0.179229 | 0.00%                              | 0.179229             | 0.16996  | -5.17%                             | 0.179229             | 0.190306 | 6.18%                              |
| 23  | 0.134924             | 0.150901 | 11.84%                             | 0.134924             | 0.134924 | 0.00%                              | 0.134924             | 0.150901 | 11.84%                             |
| 24  | 0.245582             | 0.236764 | -3.59%                             | 0.245582             | 0.236589 | -3.66%                             | 0.245582             | 0.243869 | -0.70%                             |

|    |           |           |        |           |           |         |           |           |         |
|----|-----------|-----------|--------|-----------|-----------|---------|-----------|-----------|---------|
| 25 | -0.047367 | -0.045395 | -4.16% | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.367733  | -0.98% | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.371387  | 0.00%   |
| 27 | 0.212429  | 0.218386  | 2.80%  | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.090104  | 0.00%  | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.869792  | 0.00%  | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.869792  | 0.00%   |
| 30 | 0.465468  | 0.471839  | 1.37%  | 0.465468  | 0.473439  | 1.71%   | 0.465468  | 0.471716  | 1.34%   |
| 31 | 0.081987  | 0.092943  | 13.36% | 0.081987  | 0.119171  | 45.35%  | 0.081987  | 0.121981  | 48.78%  |
| 32 | 0.595649  | 0.592133  | -0.59% | 0.595649  | 0.604338  | 1.46%   | 0.595649  | 0.610672  | 2.52%   |
| 33 | -0.024192 | -0.024192 | 0.00%  | -0.024192 | -0.008299 | -65.70% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.361032  | 0.00%  | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.362857  | 0.51%   |
| 35 | 0.258365  | 0.266212  | 3.04%  | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.256055  | -0.89%  |
| 36 | 0.099694  | 0.110179  | 10.52% | 0.099694  | 0.110179  | 10.52%  | 0.099694  | 0.102022  | 2.34%   |
| 37 | 0.279551  | 0.265127  | -5.16% | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.279551  | 0.00%   |
| 38 | 0.294258  | 0.294258  | 0.00%  | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.294258  | 0.00%   |
| 39 | 0.123474  | 0.126968  | 2.83%  | 0.123474  | 0.138815  | 12.42%  | 0.123474  | 0.159496  | 29.17%  |
| 40 | 0.190234  | 0.202094  | 6.23%  | 0.190234  | 0.198907  | 4.56%   | 0.190234  | 0.207076  | 8.85%   |

| Set | 4                 |          |                          | 5                 |          |                          | 6                 |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.143043 | 19.01%                   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.137443 | 14.35%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.221815 | 3.00%                    |
| 4   | 0.158004          | 0.175572 | 11.12%                   | 0.158004          | 0.181818 | 15.07%                   | 0.158004          | 0.175572 | 11.12%                   |
| 5   | 0.328073          | 0.362812 | 10.59%                   | 0.328073          | 0.352684 | 7.50%                    | 0.328073          | 0.362378 | 10.46%                   |
| 6   | 0.352287          | 0.35477  | 0.70%                    | 0.352287          | 0.375    | 6.45%                    | 0.352287          | 0.352287 | 0.00%                    |
| 7   | 0.411263          | 0.401042 | -2.49%                   | 0.411263          | 0.444688 | 8.13%                    | 0.411263          | 0.442731 | 7.65%                    |
| 8   | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.300745 | 3.15%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.547402 | 2.04%                    | 0.536478          | 0.544965 | 1.58%                    |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 10 | 0.200999  | 0.213276  | 6.11%   | 0.200999  | 0.207185  | 3.08%   | 0.200999  | 0.198497  | -1.24%  |
| 11 | 0.320918  | 0.327731  | 2.12%   | 0.320918  | 0.327731  | 2.12%   | 0.320918  | 0.323988  | 0.96%   |
| 12 | 0.260421  | 0.28033   | 7.64%   | 0.260421  | 0.269847  | 3.62%   | 0.260421  | 0.267178  | 2.59%   |
| 13 | 0.250986  | 0.268293  | 6.90%   | 0.250986  | 0.254902  | 1.56%   | 0.250986  | 0.262613  | 4.63%   |
| 14 | 0.313433  | 0.320075  | 2.12%   | 0.313433  | 0.330544  | 5.46%   | 0.313433  | 0.330544  | 5.46%   |
| 15 | 0.549752  | 0.561596  | 2.15%   | 0.549752  | 0.564365  | 2.66%   | 0.549752  | 0.561596  | 2.15%   |
| 16 | 0.432432  | 0.490849  | 13.51%  | 0.432432  | 0.490849  | 13.51%  | 0.432432  | 0.432432  | 0.00%   |
| 17 | 0.464004  | 0.47906   | 3.24%   | 0.464004  | 0.483021  | 4.10%   | 0.464004  | 0.474903  | 2.35%   |
| 18 | 0.353591  | 0.362398  | 2.49%   | 0.353591  | 0.362398  | 2.49%   | 0.353591  | 0.362398  | 2.49%   |
| 19 | 0.317585  | 0.321149  | 1.12%   | 0.317585  | 0.321149  | 1.12%   | 0.317585  | 0.317585  | 0.00%   |
| 20 | 0.223373  | 0.238855  | 6.93%   | 0.223373  | 0.257688  | 15.36%  | 0.223373  | 0.24406   | 9.26%   |
| 21 | 0.418464  | 0.418464  | 0.00%   | 0.418464  | 0.438596  | 4.81%   | 0.418464  | 0.431953  | 3.22%   |
| 22 | 0.179229  | 0.192751  | 7.54%   | 0.179229  | 0.183787  | 2.54%   | 0.179229  | 0.188294  | 5.06%   |
| 23 | 0.134924  | 0.150901  | 11.84%  | 0.134924  | 0.15071   | 11.70%  | 0.134924  | 0.1566    | 16.07%  |
| 24 | 0.245582  | 0.233425  | -4.95%  | 0.245582  | 0.228095  | -7.12%  | 0.245582  | 0.243869  | -0.70%  |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.384724  | 3.59%   | 0.371387  | 0.375897  | 1.21%   | 0.371387  | 0.378571  | 1.93%   |
| 27 | 0.212429  | 0.223088  | 5.02%   | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.223088  | 5.02%   |
| 28 | 0.090104  | 0.094592  | 4.98%   | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.096502  | 7.10%   |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.479742  | 3.07%   | 0.465468  | 0.475516  | 2.16%   | 0.465468  | 0.481651  | 3.48%   |
| 31 | 0.081987  | 0.127284  | 55.25%  | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.092943  | 13.36%  |
| 32 | 0.595649  | 0.606158  | 1.76%   | 0.595649  | 0.600406  | 0.80%   | 0.595649  | 0.600972  | 0.89%   |
| 33 | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.008299 | -65.70% |
| 34 | 0.361032  | 0.373771  | 3.53%   | 0.361032  | 0.362857  | 0.51%   | 0.361032  | 0.362857  | 0.51%   |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.278039  | 7.61%   | 0.258365  | 0.266212  | 3.04%   |
| 36 | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.110179  | 10.52%  |
| 37 | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.327396  | 17.11%  |
| 38 | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.304245  | 3.39%   |

|    |          |          |        |          |          |        |          |          |        |
|----|----------|----------|--------|----------|----------|--------|----------|----------|--------|
| 39 | 0.123474 | 0.159496 | 29.17% | 0.123474 | 0.138812 | 12.42% | 0.123474 | 0.145821 | 18.10% |
| 40 | 0.190234 | 0.202094 | 6.23%  | 0.190234 | 0.202094 | 6.23%  | 0.190234 | 0.198907 | 4.56%  |



- Calibración de la cantidad máxima de generaciones

| Set | 100               |          |                          | 200               |          |                          | 300               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.144893 | 20.55%                   | 0.120196          | 0.141184 | 17.46%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.173258 | 9.65%                    | 0.158004          | 0.18797  | 18.97%                   | 0.158004          | 0.18797  | 18.97%                   |
| 5   | 0.328073          | 0.352684 | 7.50%                    | 0.328073          | 0.35874  | 9.35%                    | 0.328073          | 0.352982 | 7.59%                    |
| 6   | 0.352287          | 0.35873  | 1.83%                    | 0.352287          | 0.365578 | 3.77%                    | 0.352287          | 0.369669 | 4.93%                    |
| 7   | 0.411263          | 0.410256 | -0.24%                   | 0.411263          | 0.435394 | 5.87%                    | 0.411263          | 0.435394 | 5.87%                    |
| 8   | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298246 | 2.29%                    | 0.291571          | 0.298701 | 2.45%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.548491 | 2.24%                    |
| 10  | 0.200999          | 0.201123 | 0.06%                    | 0.200999          | 0.171008 | -14.92%                  | 0.200999          | 0.17457  | -13.15%                  |
| 11  | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.330771 | 3.07%                    | 0.320918          | 0.323988 | 0.96%                    |
| 12  | 0.260421          | 0.28546  | 9.61%                    | 0.260421          | 0.269847 | 3.62%                    | 0.260421          | 0.267178 | 2.59%                    |
| 13  | 0.250986          | 0.264516 | 5.39%                    | 0.250986          | 0.268293 | 6.90%                    | 0.250986          | 0.268293 | 6.90%                    |
| 14  | 0.313433          | 0.31677  | 1.06%                    | 0.313433          | 0.330544 | 5.46%                    | 0.313433          | 0.330544 | 5.46%                    |
| 15  | 0.549752          | 0.561596 | 2.15%                    | 0.549752          | 0.561596 | 2.15%                    | 0.549752          | 0.564365 | 2.66%                    |
| 16  | 0.432432          | 0.432432 | 0.00%                    | 0.432432          | 0.43956  | 1.65%                    | 0.432432          | 0.436019 | 0.83%                    |
| 17  | 0.464004          | 0.472186 | 1.76%                    | 0.464004          | 0.474903 | 2.35%                    | 0.464004          | 0.476252 | 2.64%                    |
| 18  | 0.353591          | 0.346369 | -2.04%                   | 0.353591          | 0.356966 | 0.95%                    | 0.353591          | 0.362398 | 2.49%                    |
| 19  | 0.317585          | 0.324578 | 2.20%                    | 0.317585          | 0.324578 | 2.20%                    | 0.317585          | 0.324578 | 2.20%                    |
| 20  | 0.223373          | 0.241603 | 8.16%                    | 0.223373          | 0.238855 | 6.93%                    | 0.223373          | 0.250936 | 12.34%                   |
| 21  | 0.418464          | 0.404097 | -3.43%                   | 0.418464          | 0.404097 | -3.43%                   | 0.418464          | 0.410953 | -1.79%                   |
| 22  | 0.179229          | 0.186046 | 3.80%                    | 0.179229          | 0.186046 | 3.80%                    | 0.179229          | 0.190306 | 6.18%                    |
| 23  | 0.134924          | 0.147059 | 8.99%                    | 0.134924          | 0.150901 | 11.84%                   | 0.134924          | 0.1566   | 16.07%                   |
| 24  | 0.245582          | 0.243869 | -0.70%                   | 0.245582          | 0.233425 | -4.95%                   | 0.245582          | 0.233425 | -4.95%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.382979  | 3.12%   | 0.371387  | 0.375897  | 1.21%   | 0.371387  | 0.375897  | 1.21%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.080336  | -10.84% | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.477939  | 2.68%   | 0.465468  | 0.474174  | 1.87%   | 0.465468  | 0.469234  | 0.81%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.104829  | 27.86%  | 0.081987  | 0.104829  | 27.86%  |
| 32 | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.596229  | 0.10%   |
| 33 | -0.024192 | -0.008299 | -65.70% | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.013669 | -43.50% |
| 34 | 0.361032  | 0.362857  | 0.51%   | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.263699  | 2.06%   | 0.258365  | 0.270692  | 4.77%   |
| 36 | 0.099694  | 0.112453  | 12.80%  | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.103917  | 4.24%   |
| 37 | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.279551  | 0.00%   |
| 38 | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.316972  | 7.72%   | 0.294258  | 0.304245  | 3.39%   |
| 39 | 0.123474  | 0.159496  | 29.17%  | 0.123474  | 0.138812  | 12.42%  | 0.123474  | 0.138812  | 12.42%  |
| 40 | 0.190234  | 0.205588  | 8.07%   | 0.190234  | 0.197219  | 3.67%   | 0.190234  | 0.205588  | 8.07%   |

| Set | 400               |          |                          | 500               |          |                          | 600               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Genético |                          | Población Inicial | Genético |                          | Población Inicial | Genético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.143043 | 19.01%                   | 0.120196          | 0.144893 | 20.55%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.223684 | -1.49%                   |
| 3   | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.170931 | 8.18%                    | 0.158004          | 0.181818 | 15.07%                   | 0.158004          | 0.174943 | 10.72%                   |
| 5   | 0.328073          | 0.353874 | 7.86%                    | 0.328073          | 0.352833 | 7.55%                    | 0.328073          | 0.352982 | 7.59%                    |
| 6   | 0.352287          | 0.364247 | 3.39%                    | 0.352287          | 0.368224 | 4.52%                    | 0.352287          | 0.361837 | 2.71%                    |
| 7   | 0.411263          | 0.435394 | 5.87%                    | 0.411263          | 0.416367 | 1.24%                    | 0.411263          | 0.435394 | 5.87%                    |
| 8   | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.293887 | 0.79%                    | 0.291571          | 0.293887 | 0.79%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.557128 | 3.85%                    | 0.536478          | 0.658952 | 22.83%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 10 | 0.200999  | 0.213276  | 6.11%   | 0.200999  | 0.210972  | 4.96%   | 0.200999  | 0.171008  | -14.92% |
| 11 | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.324663  | 1.17%   | 0.320918  | 0.323988  | 0.96%   |
| 12 | 0.260421  | 0.267178  | 2.59%   | 0.260421  | 0.265837  | 2.08%   | 0.260421  | 0.269847  | 3.62%   |
| 13 | 0.250986  | 0.256845  | 2.33%   | 0.250986  | 0.258778  | 3.10%   | 0.250986  | 0.264516  | 5.39%   |
| 14 | 0.313433  | 0.324252  | 3.45%   | 0.313433  | 0.316955  | 1.12%   | 0.313433  | 0.330544  | 5.46%   |
| 15 | 0.549752  | 0.566349  | 3.02%   | 0.549752  | 0.564817  | 2.74%   | 0.549752  | 0.564817  | 2.74%   |
| 16 | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.432432  | 0.00%   | 0.432432  | 0.43956   | 1.65%   |
| 17 | 0.464004  | 0.476252  | 2.64%   | 0.464004  | 0.47906   | 3.24%   | 0.464004  | 0.47906   | 3.24%   |
| 18 | 0.353591  | 0.353591  | 0.00%   | 0.353591  | 0.353591  | 0.00%   | 0.353591  | 0.353413  | -0.05%  |
| 19 | 0.317585  | 0.321149  | 1.12%   | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.321149  | 1.12%   |
| 20 | 0.223373  | 0.241603  | 8.16%   | 0.223373  | 0.240232  | 7.55%   | 0.223373  | 0.245419  | 9.87%   |
| 21 | 0.418464  | 0.441341  | 5.47%   | 0.418464  | 0.40117   | -4.13%  | 0.418464  | 0.404097  | -3.43%  |
| 22 | 0.179229  | 0.186046  | 3.80%   | 0.179229  | 0.186046  | 3.80%   | 0.179229  | 0.181514  | 1.27%   |
| 23 | 0.134924  | 0.15071   | 11.70%  | 0.134924  | 0.1566    | 16.07%  | 0.134924  | 0.1566    | 16.07%  |
| 24 | 0.245582  | 0.236764  | -3.59%  | 0.245582  | 0.245582  | 0.00%   | 0.245582  | 0.233425  | -4.95%  |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.375     | 0.97%   | 0.371387  | 0.372294  | 0.24%   | 0.371387  | 0.378571  | 1.93%   |
| 27 | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.223088  | 5.02%   |
| 28 | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.46786   | 0.51%   | 0.465468  | 0.481651  | 3.48%   | 0.465468  | 0.479263  | 2.96%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.127284  | 55.25%  |
| 32 | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.595649  | 0.00%   | 0.595649  | 0.595649  | 0.00%   |
| 33 | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.013669 | -43.50% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.373771  | 3.53%   | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.270692  | 4.77%   | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.103917  | 4.24%   |
| 37 | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.327396  | 17.11%  | 0.279551  | 0.328184  | 17.40%  |
| 38 | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.300948  | 2.27%   | 0.294258  | 0.300948  | 2.27%   |

|    |          |          |        |          |          |        |          |          |       |
|----|----------|----------|--------|----------|----------|--------|----------|----------|-------|
| 39 | 0.123474 | 0.138812 | 12.42% | 0.123474 | 0.145821 | 18.10% | 0.123474 | 0.126968 | 2.83% |
| 40 | 0.190234 | 0.198907 | 4.56%  | 0.190234 | 0.190234 | 0.00%  | 0.190234 | 0.202094 | 6.23% |





## Anexo C: Calibración del algoritmo memético

- Calibración de la probabilidad de búsqueda local (alpha)

| Set | 0%                |          |                          | 20%               |          |                          | 40%               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Memético |                          | Población Inicial | Memético |                          | Población Inicial | Memético |                          |
| 1   | 0.120196          | 0.13913  | 15.75%                   | 0.120196          | 0.144893 | 20.55%                   | 0.120196          | 0.120196 | 0.00%                    |
| 2   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.223684 | -1.49%                   | 0.227074          | 0.225948 | -0.50%                   |
| 3   | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.179539 | 13.63%                   | 0.158004          | 0.173258 | 9.65%                    | 0.158004          | 0.173259 | 9.65%                    |
| 5   | 0.328073          | 0.331271 | 0.97%                    | 0.328073          | 0.352982 | 7.59%                    | 0.328073          | 0.352982 | 7.59%                    |
| 6   | 0.352287          | 0.359137 | 1.94%                    | 0.352287          | 0.312321 | -11.34%                  | 0.352287          | 0.35477  | 0.70%                    |
| 7   | 0.411263          | 0.414013 | 0.67%                    | 0.411263          | 0.418947 | 1.87%                    | 0.411263          | 0.459402 | 11.71%                   |
| 8   | 0.291571          | 0.286186 | -1.85%                   | 0.291571          | 0.289006 | -0.88%                   | 0.291571          | 0.298701 | 2.45%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.530164 | -1.18%                   | 0.536478          | 0.540386 | 0.73%                    |
| 10  | 0.200999          | 0.200999 | 0.00%                    | 0.200999          | 0.210972 | 4.96%                    | 0.200999          | 0.216766 | 7.84%                    |
| 11  | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.320918 | 0.00%                    | 0.320918          | 0.323988 | 0.96%                    |
| 12  | 0.260421          | 0.267178 | 2.59%                    | 0.260421          | 0.267178 | 2.59%                    | 0.260421          | 0.250607 | -3.77%                   |
| 13  | 0.250986          | 0.258778 | 3.10%                    | 0.250986          | 0.205267 | -18.22%                  | 0.250986          | 0.264326 | 5.32%                    |
| 14  | 0.313433          | 0.31306  | -0.12%                   | 0.313433          | 0.313433 | 0.00%                    | 0.313433          | 0.344799 | 10.01%                   |
| 15  | 0.549752          | 0.554764 | 0.91%                    | 0.549752          | 0.549752 | 0.00%                    | 0.549752          | 0.561596 | 2.15%                    |
| 16  | 0.432432          | 0.432432 | 0.00%                    | 0.432432          | 0.43956  | 1.65%                    | 0.432432          | 0.432432 | 0.00%                    |
| 17  | 0.464004          | 0.481707 | 3.82%                    | 0.464004          | 0.465409 | 0.30%                    | 0.464004          | 0.47906  | 3.24%                    |
| 18  | 0.353591          | 0.357143 | 1.00%                    | 0.353591          | 0.347462 | -1.73%                   | 0.353591          | 0.338609 | -4.24%                   |
| 19  | 0.317585          | 0.321149 | 1.12%                    | 0.317585          | 0.324578 | 2.20%                    | 0.317585          | 0.324578 | 2.20%                    |
| 20  | 0.223373          | 0.24297  | 8.77%                    | 0.223373          | 0.234554 | 5.01%                    | 0.223373          | 0.24297  | 8.77%                    |
| 21  | 0.418464          | 0.40117  | -4.13%                   | 0.418464          | 0.405204 | -3.17%                   | 0.418464          | 0.432221 | 3.29%                    |
| 22  | 0.179229          | 0.16996  | -5.17%                   | 0.179229          | 0.199346 | 11.22%                   | 0.179229          | 0.186046 | 3.80%                    |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 23 | 0.134924  | 0.134924  | 0.00%   | 0.134924  | 0.117096  | -13.21% | 0.134924  | 0.150901  | 11.84%  |
| 24 | 0.245582  | 0.226121  | -7.92%  | 0.245582  | 0.22087   | -10.06% | 0.245582  | 0.236764  | -3.59%  |
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.372294  | 0.24%   | 0.371387  | 0.375     | 0.97%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.223088  | 5.02%   |
| 28 | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.864636  | -0.59%  | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.475516  | 2.16%   | 0.465468  | 0.474174  | 1.87%   | 0.465468  | 0.464074  | -0.30%  |
| 31 | 0.081987  | 0.121981  | 48.78%  | 0.081987  | 0.117961  | 43.88%  | 0.081987  | 0.084613  | 3.20%   |
| 32 | 0.595649  | 0.606158  | 1.76%   | 0.595649  | 0.597959  | 0.39%   | 0.595649  | 0.598696  | 0.51%   |
| 33 | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.347762  | -3.68%  | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.260914  | 0.99%   | 0.258365  | 0.266212  | 3.04%   | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.10831   | 8.64%   | 0.099694  | 0.103917  | 4.24%   | 0.099694  | 0.106224  | 6.55%   |
| 37 | 0.279551  | 0.265127  | -5.16%  | 0.279551  | 0.279551  | 0.00%   | 0.279551  | 0.327396  | 17.11%  |
| 38 | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.294258  | 0.00%   | 0.294258  | 0.294258  | 0.00%   |
| 39 | 0.123474  | 0.145533  | 17.87%  | 0.123474  | 0.145533  | 17.87%  | 0.123474  | 0.126968  | 2.83%   |
| 40 | 0.190234  | 0.205588  | 8.07%   | 0.190234  | 0.197219  | 3.67%   | 0.190234  | 0.190234  | 0.00%   |

| Set | 60%               |          |                          | 80%               |          |                          | 100%              |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Memético |                          | Población Inicial | Memético |                          | Población Inicial | Memético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.137443 | 14.35%                   | 0.120196          | 0.137443 | 14.35%                   |
| 2   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.225948 | -0.50%                   |
| 3   | 0.215348          | 0.221815 | 3.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.166881 | 5.62%                    | 0.158004          | 0.175572 | 11.12%                   | 0.158004          | 0.18797  | 18.97%                   |
| 5   | 0.328073          | 0.365975 | 11.55%                   | 0.328073          | 0.366118 | 11.60%                   | 0.328073          | 0.366118 | 11.60%                   |
| 6   | 0.352287          | 0.352287 | 0.00%                    | 0.352287          | 0.347966 | -1.23%                   | 0.352287          | 0.345148 | -2.03%                   |
| 7   | 0.411263          | 0.44988  | 9.39%                    | 0.411263          | 0.456848 | 11.08%                   | 0.411263          | 0.452144 | 9.94%                    |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 8  | 0.291571  | 0.293656  | 0.72%   | 0.291571  | 0.293887  | 0.79%   | 0.291571  | 0.293887  | 0.79%   |
| 9  | 0.536478  | 0.539007  | 0.47%   | 0.536478  | 0.539007  | 0.47%   | 0.536478  | 0.540386  | 0.73%   |
| 10 | 0.200999  | 0.213276  | 6.11%   | 0.200999  | 0.222947  | 10.92%  | 0.200999  | 0.228683  | 13.77%  |
| 11 | 0.320918  | 0.320918  | 0.00%   | 0.320918  | 0.320918  | 0.00%   | 0.320918  | 0.323988  | 0.96%   |
| 12 | 0.260421  | 0.281619  | 8.14%   | 0.260421  | 0.261782  | 0.52%   | 0.260421  | 0.260421  | 0.00%   |
| 13 | 0.250986  | 0.250789  | -0.08%  | 0.250986  | 0.250592  | -0.16%  | 0.250986  | 0.250789  | -0.08%  |
| 14 | 0.313433  | 0.294282  | -6.11%  | 0.313433  | 0.309498  | -1.26%  | 0.313433  | 0.313433  | 0.00%   |
| 15 | 0.549752  | 0.554665  | 0.89%   | 0.549752  | 0.556836  | 1.29%   | 0.549752  | 0.561596  | 2.15%   |
| 16 | 0.432432  | 0.491694  | 13.70%  | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.432432  | 0.00%   |
| 17 | 0.464004  | 0.468196  | 0.90%   | 0.464004  | 0.474903  | 2.35%   | 0.464004  | 0.481707  | 3.82%   |
| 18 | 0.353591  | 0.344354  | -2.61%  | 0.353591  | 0.342142  | -3.24%  | 0.353591  | 0.347826  | -1.63%  |
| 19 | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.321149  | 1.12%   |
| 20 | 0.223373  | 0.24406   | 9.26%   | 0.223373  | 0.250936  | 12.34%  | 0.223373  | 0.24406   | 9.26%   |
| 21 | 0.418464  | 0.433026  | 3.48%   | 0.418464  | 0.433026  | 3.48%   | 0.418464  | 0.431953  | 3.22%   |
| 22 | 0.179229  | 0.188294  | 5.06%   | 0.179229  | 0.181514  | 1.27%   | 0.179229  | 0.174621  | -2.57%  |
| 23 | 0.134924  | 0.147059  | 8.99%   | 0.134924  | 0.147059  | 8.99%   | 0.134924  | 0.150901  | 11.84%  |
| 24 | 0.245582  | 0.233425  | -4.95%  | 0.245582  | 0.233425  | -4.95%  | 0.245582  | 0.250675  | 2.07%   |
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.382979  | 3.12%   | 0.371387  | 0.364964  | -1.73%  |
| 27 | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218405  | 2.81%   |
| 28 | 0.090104  | 0.085084  | -5.57%  | 0.090104  | 0.083452  | -7.38%  | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.864636  | -0.59%  | 0.869792  | 0.869792  | 0.00%   | 0.869792  | 0.864636  | -0.59%  |
| 30 | 0.465468  | 0.46786   | 0.51%   | 0.465468  | 0.46786   | 0.51%   | 0.465468  | 0.46786   | 0.51%   |
| 31 | 0.081987  | 0.123313  | 50.41%  | 0.081987  | 0.104829  | 27.86%  | 0.081987  | 0.092943  | 13.36%  |
| 32 | 0.595649  | 0.600972  | 0.89%   | 0.595649  | 0.606158  | 1.76%   | 0.595649  | 0.609049  | 2.25%   |
| 33 | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.344503  | -4.58%  |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.266212  | 3.04%   | 0.258365  | 0.266212  | 3.04%   |
| 36 | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.093283  | -6.43%  |

|    |          |          |        |          |          |        |          |          |        |
|----|----------|----------|--------|----------|----------|--------|----------|----------|--------|
| 37 | 0.279551 | 0.331781 | 18.68% | 0.279551 | 0.337947 | 20.89% | 0.279551 | 0.337947 | 20.89% |
| 38 | 0.294258 | 0.294258 | 0.00%  | 0.294258 | 0.294258 | 0.00%  | 0.294258 | 0.304245 | 3.39%  |
| 39 | 0.123474 | 0.159496 | 29.17% | 0.123474 | 0.154419 | 25.06% | 0.123474 | 0.157641 | 27.67% |
| 40 | 0.190234 | 0.190234 | 0.00%  | 0.190234 | 0.186759 | -1.83% | 0.190234 | 0.190234 | 0.00%  |



- Calibración de la cantidad de generaciones sin mejora

| Set | 1                    |          |                                    | 2                    |          |                                    | 3                    |          |                                    |
|-----|----------------------|----------|------------------------------------|----------------------|----------|------------------------------------|----------------------|----------|------------------------------------|
|     | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) | Función Fitness      |          | Porcentaj<br>e de<br>Mejora<br>(%) |
|     | Población<br>Inicial | Memético |                                    | Población<br>Inicial | Memético |                                    | Población<br>Inicial | Memético |                                    |
| 1   | 0.120196             | 0.141184 | 17.46%                             | 0.120196             | 0.141184 | 17.46%                             | 0.120196             | 0.189    | 57.24%                             |
| 2   | 0.227074             | 0.227074 | 0.00%                              | 0.227074             | 0.227074 | 0.00%                              | 0.227074             | 0.227074 | 0.00%                              |
| 3   | 0.215348             | 0.215348 | 0.00%                              | 0.215348             | 0.215348 | 0.00%                              | 0.215348             | 0.215348 | 0.00%                              |
| 4   | 0.158004             | 0.171355 | 8.45%                              | 0.158004             | 0.157785 | -0.14%                             | 0.158004             | 0.171356 | 8.45%                              |
| 5   | 0.328073             | 0.363245 | 10.72%                             | 0.328073             | 0.365832 | 11.51%                             | 0.328073             | 0.363245 | 10.72%                             |
| 6   | 0.352287             | 0.299416 | -15.01%                            | 0.352287             | 0.347966 | -1.23%                             | 0.352287             | 0.315264 | -10.51%                            |
| 7   | 0.411263             | 0.448561 | 9.07%                              | 0.411263             | 0.459402 | 11.71%                             | 0.411263             | 0.450836 | 9.62%                              |
| 8   | 0.291571             | 0.295958 | 1.50%                              | 0.291571             | 0.298701 | 2.45%                              | 0.291571             | 0.300065 | 2.91%                              |
| 9   | 0.536478             | 0.535076 | -0.26%                             | 0.536478             | 0.658328 | 22.71%                             | 0.536478             | 0.528723 | -1.45%                             |
| 10  | 0.200999             | 0.219155 | 9.03%                              | 0.200999             | 0.219155 | 9.03%                              | 0.200999             | 0.224125 | 11.51%                             |
| 11  | 0.320918             | 0.323988 | 0.96%                              | 0.320918             | 0.323988 | 0.96%                              | 0.320918             | 0.323988 | 0.96%                              |
| 12  | 0.260421             | 0.267178 | 2.59%                              | 0.260421             | 0.267178 | 2.59%                              | 0.260421             | 0.269714 | 3.57%                              |
| 13  | 0.250986             | 0.250986 | 0.00%                              | 0.250986             | 0.258778 | 3.10%                              | 0.250986             | 0.258778 | 3.10%                              |
| 14  | 0.313433             | 0.31677  | 1.06%                              | 0.313433             | 0.313433 | 0.00%                              | 0.313433             | 0.31677  | 1.06%                              |
| 15  | 0.549752             | 0.554764 | 0.91%                              | 0.549752             | 0.556836 | 1.29%                              | 0.549752             | 0.559568 | 1.79%                              |
| 16  | 0.432432             | 0.432432 | 0.00%                              | 0.432432             | 0.490849 | 13.51%                             | 0.432432             | 0.43956  | 1.65%                              |
| 17  | 0.464004             | 0.476252 | 2.64%                              | 0.464004             | 0.47906  | 3.24%                              | 0.464004             | 0.47906  | 3.24%                              |
| 18  | 0.353591             | 0.353591 | 0.00%                              | 0.353591             | 0.362398 | 2.49%                              | 0.353591             | 0.34417  | -2.66%                             |
| 19  | 0.317585             | 0.321149 | 1.12%                              | 0.317585             | 0.324578 | 2.20%                              | 0.317585             | 0.321149 | 1.12%                              |
| 20  | 0.223373             | 0.250936 | 12.34%                             | 0.223373             | 0.25227  | 12.94%                             | 0.223373             | 0.249598 | 11.74%                             |
| 21  | 0.418464             | 0.432221 | 3.29%                              | 0.418464             | 0.438596 | 4.81%                              | 0.418464             | 0.438596 | 4.81%                              |
| 22  | 0.179229             | 0.186046 | 3.80%                              | 0.179229             | 0.210102 | 17.23%                             | 0.179229             | 0.188294 | 5.06%                              |
| 23  | 0.134924             | 0.150901 | 11.84%                             | 0.134924             | 0.148984 | 10.42%                             | 0.134924             | 0.1566   | 16.07%                             |
| 24  | 0.245582             | 0.233425 | -4.95%                             | 0.245582             | 0.238509 | -2.88%                             | 0.245582             | 0.243697 | -0.77%                             |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.382979  | 3.12%   | 0.371387  | 0.372294  | 0.24%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.218386  | 2.80%   |
| 28 | 0.090104  | 0.085246  | -5.39%  | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.873096  | 0.38%   |
| 30 | 0.465468  | 0.465468  | 0.00%   | 0.465468  | 0.475395  | 2.13%   | 0.465468  | 0.470354  | 1.05%   |
| 31 | 0.081987  | 0.119305  | 45.52%  | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.127284  | 55.25%  |
| 32 | 0.595649  | 0.60835   | 2.13%   | 0.595649  | 0.667149  | 12.00%  | 0.595649  | 0.664583  | 11.57%  |
| 33 | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.013669 | -43.50% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.345849  | -4.21%  | 0.361032  | 0.362857  | 0.51%   |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.266212  | 3.04%   | 0.258365  | 0.270692  | 4.77%   |
| 36 | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.112453  | 12.80%  |
| 37 | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.337947  | 20.89%  |
| 38 | 0.294258  | 0.300948  | 2.27%   | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.304245  | 3.39%   |
| 39 | 0.123474  | 0.138815  | 12.42%  | 0.123474  | 0.162846  | 31.89%  | 0.123474  | 0.162846  | 31.89%  |
| 40 | 0.190234  | 0.190234  | 0.00%   | 0.190234  | 0.205588  | 8.07%   | 0.190234  | 0.202094  | 6.23%   |

| Set | 4                 |          |                          | 5                 |          |                          | 6                 |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Memético |                          | Población Inicial | Memético |                          | Población Inicial | Memético |                          |
| 1   | 0.120196          | 0.144893 | 20.55%                   | 0.120196          | 0.144893 | 20.55%                   | 0.120196          | 0.141184 | 17.46%                   |
| 2   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.221815 | 3.00%                    |
| 4   | 0.158004          | 0.148713 | -5.88%                   | 0.158004          | 0.175573 | 11.12%                   | 0.158004          | 0.173258 | 9.65%                    |
| 5   | 0.328073          | 0.366118 | 11.60%                   | 0.328073          | 0.366118 | 11.60%                   | 0.328073          | 0.366118 | 11.60%                   |
| 6   | 0.352287          | 0.352287 | 0.00%                    | 0.352287          | 0.352287 | 0.00%                    | 0.352287          | 0.353531 | 0.35%                    |
| 7   | 0.411263          | 0.448922 | 9.16%                    | 0.411263          | 0.452144 | 9.94%                    | 0.411263          | 0.459402 | 11.71%                   |
| 8   | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.658641 | 22.77%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 10 | 0.200999  | 0.228683  | 13.77%  | 0.200999  | 0.228683  | 13.77%  | 0.200999  | 0.228683  | 13.77%  |
| 11 | 0.320918  | 0.324663  | 1.17%   | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.327731  | 2.12%   |
| 12 | 0.260421  | 0.269847  | 3.62%   | 0.260421  | 0.281619  | 8.14%   | 0.260421  | 0.281619  | 8.14%   |
| 13 | 0.250986  | 0.258778  | 3.10%   | 0.250986  | 0.254902  | 1.56%   | 0.250986  | 0.260509  | 3.79%   |
| 14 | 0.313433  | 0.308743  | -1.50%  | 0.313433  | 0.313433  | 0.00%   | 0.313433  | 0.313433  | 0.00%   |
| 15 | 0.549752  | 0.556836  | 1.29%   | 0.549752  | 0.556836  | 1.29%   | 0.549752  | 0.559568  | 1.79%   |
| 16 | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.490849  | 13.51%  |
| 17 | 0.464004  | 0.483021  | 4.10%   | 0.464004  | 0.483021  | 4.10%   | 0.464004  | 0.483021  | 4.10%   |
| 18 | 0.353591  | 0.348008  | -1.58%  | 0.353591  | 0.335982  | -4.98%  | 0.353591  | 0.346369  | -2.04%  |
| 19 | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.324578  | 2.20%   |
| 20 | 0.223373  | 0.245419  | 9.87%   | 0.223373  | 0.25227   | 12.94%  | 0.223373  | 0.256374  | 14.77%  |
| 21 | 0.418464  | 0.435825  | 4.15%   | 0.418464  | 0.445087  | 6.36%   | 0.418464  | 0.438596  | 4.81%   |
| 22 | 0.179229  | 0.188294  | 5.06%   | 0.179229  | 0.188294  | 5.06%   | 0.179229  | 0.210102  | 17.23%  |
| 23 | 0.134924  | 0.150901  | 11.84%  | 0.134924  | 0.1566    | 16.07%  | 0.134924  | 0.1566    | 16.07%  |
| 24 | 0.245582  | 0.243869  | -0.70%  | 0.245582  | 0.243869  | -0.70%  | 0.245582  | 0.250675  | 2.07%   |
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.384724  | 3.59%   | 0.371387  | 0.372294  | 0.24%   | 0.371387  | 0.384724  | 3.59%   |
| 27 | 0.212429  | 0.223088  | 5.02%   | 0.212429  | 0.223088  | 5.02%   | 0.212429  | 0.223088  | 5.02%   |
| 28 | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.096502  | 7.10%   |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.479263  | 2.96%   | 0.465468  | 0.485896  | 4.39%   | 0.465468  | 0.483429  | 3.86%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.12978   | 58.29%  |
| 32 | 0.595649  | 0.607413  | 1.97%   | 0.595649  | 0.669201  | 12.35%  | 0.595649  | 0.612281  | 2.79%   |
| 33 | -0.024192 | -0.008299 | -65.70% | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.359195  | -0.51%  | 0.361032  | 0.361032  | 0.00%   |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.273403  | 5.82%   | 0.258365  | 0.285477  | 10.49%  |
| 36 | 0.099694  | 0.106224  | 6.55%   | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.112454  | 12.80%  |
| 37 | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.337947  | 20.89%  |
| 38 | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.304245  | 3.39%   |

|    |          |          |        |          |          |        |          |          |        |
|----|----------|----------|--------|----------|----------|--------|----------|----------|--------|
| 39 | 0.123474 | 0.195705 | 58.50% | 0.123474 | 0.162846 | 31.89% | 0.123474 | 0.162846 | 31.89% |
| 40 | 0.190234 | 0.205588 | 8.07%  | 0.190234 | 0.205588 | 8.07%  | 0.190234 | 0.202094 | 6.23%  |





- Calibración de la cantidad máxima de generaciones

| Set | 100               |          |                          | 200               |          |                          | 300               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Memético |                          | Población Inicial | Memético |                          | Población Inicial | Memético |                          |
| 1   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.143043 | 19.01%                   |
| 2   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.221815 | 3.00%                    |
| 4   | 0.158004          | 0.170931 | 8.18%                    | 0.158004          | 0.162141 | 2.62%                    | 0.158004          | 0.160187 | 1.38%                    |
| 5   | 0.328073          | 0.366118 | 11.60%                   | 0.328073          | 0.363245 | 10.72%                   | 0.328073          | 0.365975 | 11.55%                   |
| 6   | 0.352287          | 0.357643 | 1.52%                    | 0.352287          | 0.349506 | -0.79%                   | 0.352287          | 0.352287 | 0.00%                    |
| 7   | 0.411263          | 0.449521 | 9.30%                    | 0.411263          | 0.451906 | 9.88%                    | 0.411263          | 0.453446 | 10.26%                   |
| 8   | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    |
| 9   | 0.536478          | 0.536478 | 0.00%                    | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.535076 | -0.26%                   |
| 10  | 0.200999          | 0.225299 | 12.09%                   | 0.200999          | 0.217006 | 7.96%                    | 0.200999          | 0.237982 | 18.40%                   |
| 11  | 0.320918          | 0.323988 | 0.96%                    | 0.320918          | 0.330771 | 3.07%                    | 0.320918          | 0.320918 | 0.00%                    |
| 12  | 0.260421          | 0.269847 | 3.62%                    | 0.260421          | 0.269847 | 3.62%                    | 0.260421          | 0.269714 | 3.57%                    |
| 13  | 0.250986          | 0.245033 | -2.37%                   | 0.250986          | 0.264326 | 5.32%                    | 0.250986          | 0.254902 | 1.56%                    |
| 14  | 0.313433          | 0.313433 | 0.00%                    | 0.313433          | 0.313433 | 0.00%                    | 0.313433          | 0.320623 | 2.29%                    |
| 15  | 0.549752          | 0.554665 | 0.89%                    | 0.549752          | 0.564817 | 2.74%                    | 0.549752          | 0.557522 | 1.41%                    |
| 16  | 0.432432          | 0.43956  | 1.65%                    | 0.432432          | 0.432432 | 0.00%                    | 0.432432          | 0.43956  | 1.65%                    |
| 17  | 0.464004          | 0.476252 | 2.64%                    | 0.464004          | 0.481707 | 3.82%                    | 0.464004          | 0.483021 | 4.10%                    |
| 18  | 0.353591          | 0.347644 | -1.68%                   | 0.353591          | 0.336923 | -4.71%                   | 0.353591          | 0.347644 | -1.68%                   |
| 19  | 0.317585          | 0.321149 | 1.12%                    | 0.317585          | 0.324578 | 2.20%                    | 0.317585          | 0.324578 | 2.20%                    |
| 20  | 0.223373          | 0.24297  | 8.77%                    | 0.223373          | 0.245419 | 9.87%                    | 0.223373          | 0.24406  | 9.26%                    |
| 21  | 0.418464          | 0.441341 | 5.47%                    | 0.418464          | 0.447768 | 7.00%                    | 0.418464          | 0.447768 | 7.00%                    |
| 22  | 0.179229          | 0.188294 | 5.06%                    | 0.179229          | 0.210102 | 17.23%                   | 0.179229          | 0.210102 | 17.23%                   |
| 23  | 0.134924          | 0.1566   | 16.07%                   | 0.134924          | 0.150901 | 11.84%                   | 0.134924          | 0.150901 | 11.84%                   |
| 24  | 0.245582          | 0.233425 | -4.95%                   | 0.245582          | 0.243869 | -0.70%                   | 0.245582          | 0.236764 | -3.59%                   |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 25 | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   |
| 26 | 0.371387  | 0.384724  | 3.59%   | 0.371387  | 0.369565  | -0.49%  | 0.371387  | 0.369565  | -0.49%  |
| 27 | 0.212429  | 0.218405  | 2.81%   | 0.212429  | 0.239789  | 12.88%  | 0.212429  | 0.223088  | 5.02%   |
| 28 | 0.090104  | 0.098087  | 8.86%   | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.132301  | 46.83%  |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.476124  | 2.29%   | 0.465468  | 0.47794   | 2.68%   | 0.465468  | 0.487179  | 4.66%   |
| 31 | 0.081987  | 0.081987  | 0.00%   | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.119305  | 45.52%  |
| 32 | 0.595649  | 0.610672  | 2.52%   | 0.595649  | 0.606158  | 1.76%   | 0.595649  | 0.60835   | 2.13%   |
| 33 | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.362857  | 0.51%   | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.373771  | 3.53%   |
| 35 | 0.258365  | 0.273403  | 5.82%   | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.10831   | 8.64%   | 0.099694  | 0.112454  | 12.80%  |
| 37 | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.331781  | 18.68%  | 0.279551  | 0.337947  | 20.89%  |
| 38 | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.304245  | 3.39%   | 0.294258  | 0.300948  | 2.27%   |
| 39 | 0.123474  | 0.159496  | 29.17%  | 0.123474  | 0.159496  | 29.17%  | 0.123474  | 0.157641  | 27.67%  |
| 40 | 0.190234  | 0.205588  | 8.07%   | 0.190234  | 0.198907  | 4.56%   | 0.190234  | 0.198907  | 4.56%   |

| Set | 400               |          |                          | 500               |          |                          | 600               |          |                          |
|-----|-------------------|----------|--------------------------|-------------------|----------|--------------------------|-------------------|----------|--------------------------|
|     | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) | Función Fitness   |          | Porcentaje de Mejora (%) |
|     | Población Inicial | Memético |                          | Población Inicial | Memético |                          | Población Inicial | Memético |                          |
| 1   | 0.120196          | 0.143043 | 19.01%                   | 0.120196          | 0.141184 | 17.46%                   | 0.120196          | 0.141184 | 17.46%                   |
| 2   | 0.227074          | 0.225948 | -0.50%                   | 0.227074          | 0.227074 | 0.00%                    | 0.227074          | 0.227074 | 0.00%                    |
| 3   | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    | 0.215348          | 0.215348 | 0.00%                    |
| 4   | 0.158004          | 0.166667 | 5.48%                    | 0.158004          | 0.169018 | 6.97%                    | 0.158004          | 0.153384 | -2.92%                   |
| 5   | 0.328073          | 0.363245 | 10.72%                   | 0.328073          | 0.365975 | 11.55%                   | 0.328073          | 0.363245 | 10.72%                   |
| 6   | 0.352287          | 0.347966 | -1.23%                   | 0.352287          | 0.316657 | -10.11%                  | 0.352287          | 0.34656  | -1.63%                   |
| 7   | 0.411263          | 0.448922 | 9.16%                    | 0.411263          | 0.450955 | 9.65%                    | 0.411263          | 0.453446 | 10.26%                   |
| 8   | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.298701 | 2.45%                    | 0.291571          | 0.293887 | 0.79%                    |
| 9   | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.540386 | 0.73%                    | 0.536478          | 0.536478 | 0.00%                    |

|    |           |           |         |           |           |         |           |           |         |
|----|-----------|-----------|---------|-----------|-----------|---------|-----------|-----------|---------|
| 10 | 0.200999  | 0.226352  | 12.61%  | 0.200999  | 0.228683  | 13.77%  | 0.200999  | 0.224125  | 11.51%  |
| 11 | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.323988  | 0.96%   | 0.320918  | 0.323988  | 0.96%   |
| 12 | 0.260421  | 0.267178  | 2.59%   | 0.260421  | 0.28033   | 7.64%   | 0.260421  | 0.269847  | 3.62%   |
| 13 | 0.250986  | 0.256845  | 2.33%   | 0.250986  | 0.260509  | 3.79%   | 0.250986  | 0.258778  | 3.10%   |
| 14 | 0.313433  | 0.316401  | 0.95%   | 0.313433  | 0.32299   | 3.05%   | 0.313433  | 0.320623  | 2.29%   |
| 15 | 0.549752  | 0.549752  | 0.00%   | 0.549752  | 0.556836  | 1.29%   | 0.549752  | 0.547614  | -0.39%  |
| 16 | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.43956   | 1.65%   | 0.432432  | 0.43956   | 1.65%   |
| 17 | 0.464004  | 0.47906   | 3.24%   | 0.464004  | 0.467919  | 0.84%   | 0.464004  | 0.476252  | 2.64%   |
| 18 | 0.353591  | 0.340659  | -3.66%  | 0.353591  | 0.342142  | -3.24%  | 0.353591  | 0.338609  | -4.24%  |
| 19 | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.324578  | 2.20%   | 0.317585  | 0.324578  | 2.20%   |
| 20 | 0.223373  | 0.250535  | 12.16%  | 0.223373  | 0.245419  | 9.87%   | 0.223373  | 0.245419  | 9.87%   |
| 21 | 0.418464  | 0.445087  | 6.36%   | 0.418464  | 0.435825  | 4.15%   | 0.418464  | 0.432221  | 3.29%   |
| 22 | 0.179229  | 0.207974  | 16.04%  | 0.179229  | 0.186046  | 3.80%   | 0.179229  | 0.186046  | 3.80%   |
| 23 | 0.134924  | 0.150901  | 11.84%  | 0.134924  | 0.1566    | 16.07%  | 0.134924  | 0.148984  | 10.42%  |
| 24 | 0.245582  | 0.250675  | 2.07%   | 0.245582  | 0.236764  | -3.59%  | 0.245582  | 0.250675  | 2.07%   |
| 25 | -0.047367 | -0.045395 | -4.16%  | -0.047367 | -0.047367 | 0.00%   | -0.047367 | -0.045395 | -4.16%  |
| 26 | 0.371387  | 0.371387  | 0.00%   | 0.371387  | 0.378571  | 1.93%   | 0.371387  | 0.375897  | 1.21%   |
| 27 | 0.212429  | 0.218386  | 2.80%   | 0.212429  | 0.239789  | 12.88%  | 0.212429  | 0.239789  | 12.88%  |
| 28 | 0.090104  | 0.090104  | 0.00%   | 0.090104  | 0.094592  | 4.98%   | 0.090104  | 0.090104  | 0.00%   |
| 29 | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   | 0.869792  | 0.872014  | 0.26%   |
| 30 | 0.465468  | 0.479742  | 3.07%   | 0.465468  | 0.485194  | 4.24%   | 0.465468  | 0.47794   | 2.68%   |
| 31 | 0.081987  | 0.092943  | 13.36%  | 0.081987  | 0.117961  | 43.88%  | 0.081987  | 0.123313  | 50.41%  |
| 32 | 0.595649  | 0.663109  | 11.33%  | 0.595649  | 0.60835   | 2.13%   | 0.595649  | 0.606158  | 1.76%   |
| 33 | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.0083   | -65.69% | -0.024192 | -0.0083   | -65.69% |
| 34 | 0.361032  | 0.345849  | -4.21%  | 0.361032  | 0.361032  | 0.00%   | 0.361032  | 0.362857  | 0.51%   |
| 35 | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.268707  | 4.00%   | 0.258365  | 0.268707  | 4.00%   |
| 36 | 0.099694  | 0.112453  | 12.80%  | 0.099694  | 0.112454  | 12.80%  | 0.099694  | 0.103917  | 4.24%   |
| 37 | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.337947  | 20.89%  | 0.279551  | 0.333643  | 19.35%  |
| 38 | 0.294258  | 0.316971  | 7.72%   | 0.294258  | 0.316972  | 7.72%   | 0.294258  | 0.304245  | 3.39%   |

|    |          |          |        |          |          |        |          |          |        |
|----|----------|----------|--------|----------|----------|--------|----------|----------|--------|
| 39 | 0.123474 | 0.162846 | 31.89% | 0.123474 | 0.154419 | 25.06% | 0.123474 | 0.159496 | 29.17% |
| 40 | 0.190234 | 0.202094 | 6.23%  | 0.190234 | 0.202094 | 6.23%  | 0.190234 | 0.202094 | 6.23%  |



Anexo D: Experimentación Numérica

| Set | Fitness Población Inicial | Fitness Algoritmo Genético | Fitness Algoritmo Memético |
|-----|---------------------------|----------------------------|----------------------------|
| 1   | 0.120196                  | 0.144893                   | 0.143043                   |
| 2   | 0.227074                  | 0.223684                   | 0.227074                   |
| 3   | 0.215348                  | 0.215348                   | 0.221815                   |
| 4   | 0.158004                  | 0.174943                   | 0.160187                   |
| 5   | 0.328073                  | 0.352982                   | 0.365975                   |
| 6   | 0.352287                  | 0.361837                   | 0.352287                   |
| 7   | 0.411263                  | 0.435394                   | 0.453446                   |
| 8   | 0.291571                  | 0.293887                   | 0.298701                   |
| 9   | 0.536478                  | 0.658952                   | 0.535076                   |
| 10  | 0.200999                  | 0.171008                   | 0.237982                   |
| 11  | 0.320918                  | 0.323988                   | 0.320918                   |
| 12  | 0.260421                  | 0.269847                   | 0.269714                   |
| 13  | 0.250986                  | 0.264516                   | 0.254902                   |
| 14  | 0.313433                  | 0.330544                   | 0.320623                   |
| 15  | 0.549752                  | 0.564817                   | 0.557522                   |
| 16  | 0.432432                  | 0.43956                    | 0.43956                    |
| 17  | 0.464004                  | 0.47906                    | 0.483021                   |
| 18  | 0.353591                  | 0.353413                   | 0.347644                   |
| 19  | 0.317585                  | 0.321149                   | 0.324578                   |
| 20  | 0.223373                  | 0.245419                   | 0.24406                    |
| 21  | 0.418464                  | 0.404097                   | 0.447768                   |
| 22  | 0.179229                  | 0.181514                   | 0.210102                   |
| 23  | 0.134924                  | 0.1566                     | 0.150901                   |
| 24  | 0.245582                  | 0.233425                   | 0.236764                   |
| 25  | -0.047367                 | -0.045395                  | -0.047367                  |
| 26  | 0.371387                  | 0.378571                   | 0.369565                   |
| 27  | 0.212429                  | 0.223088                   | 0.223088                   |
| 28  | 0.090104                  | 0.090104                   | 0.132301                   |
| 29  | 0.869792                  | 0.872014                   | 0.872014                   |
| 30  | 0.465468                  | 0.479263                   | 0.487179                   |

|    |           |           |          |
|----|-----------|-----------|----------|
| 31 | 0.081987  | 0.127284  | 0.119305 |
| 32 | 0.595649  | 0.595649  | 0.60835  |
| 33 | -0.024192 | -0.013669 | -0.0083  |
| 34 | 0.361032  | 0.361032  | 0.373771 |
| 35 | 0.258365  | 0.268707  | 0.268707 |
| 36 | 0.099694  | 0.103917  | 0.112454 |
| 37 | 0.279551  | 0.328184  | 0.337947 |
| 38 | 0.294258  | 0.300948  | 0.300948 |
| 39 | 0.123474  | 0.126968  | 0.157641 |
| 40 | 0.190234  | 0.202094  | 0.198907 |



# 1 Alcance y Limitaciones

En esta sección, se presentan tanto el alcance del proyecto, como las limitaciones y los riesgos encontrados.

## 1.1 Alcance

El presente proyecto comprende el desarrollo de una aplicación para resolver el problema del corte de materiales en su variante unidimensional. Esta solución es desarrollada en el lenguaje Java y está disponible en forma de aplicación de escritorio.

Se abarcan las tareas realizadas por un algoritmo memético, adaptando su funcionamiento al problema en la variante de una dimensión. Las funcionalidades se limitarán a aquellas pertenecientes a la definición en la literatura del problema, debido a que está bien definido y sirve como un framework para el desarrollo de la solución.

Acerca de la comparación con un algoritmo genético, para la evaluación de la eficiencia del algoritmo se utilizarán pruebas numéricas, las cuales comparan el desempeño de la solución contra el desempeño de un algoritmo genético. Se sigue el diseño de prueba Z para la comparación numérica.

Conforme al cronograma del proyecto y el diseño del mismo, el desarrollo considera la definición de las funciones objetivo, variables y restricción del problema, la elaboración de un algoritmo como propuesta de solución, la adaptación de un algoritmo de la literatura para realizar una comparación de eficiencia, experimentación numérica y un reporte de resultados y conclusiones.

## 1.2 Limitaciones

Los obstáculos y restricciones identificados, que podrían limitar o dificultar el éxito del proyecto son:

- Se limita el tiempo de ejecución y de experimentación: Debido a la cantidad de variables y factores del problema. Cantidad de factores que afectan el rendimiento del algoritmo, Cantidad de datos que deben ser soportados por la aplicación, Cantidad de resultados obtenidos por los algoritmos.

## 2 Riesgos

Estos son los riesgos identificados, la probabilidad de su ocurrencia, el impacto sobre el proyecto que causarían y la contingencia propuesta.

Riesgos y Contingencias  
(Autoría Propia)

| Nº | Riesgo   | Probabilidad | Impacto | Contingencia   |
|----|--|--------------|---------|--|
| 1  | Pérdida del progreso alcanzado   | 0.1          | 0.9     | Tener un repositorio en línea donde versionar la información del proyecto regularmente   |
| 2  | Pérdida física de los equipos informáticos usados para el desarrollo del proyecto  | 0.1          | 0.8     | Tener un fondo económico para la adquisición de equipos nuevos.  |
| 3  | No se cumpla con alcanzar los resultados esperados en el tiempo planificado  | 0.4          | 0.3     | Definir el cronograma de avances del proyecto con holgura, tomando en cuenta los tiempos de las tareas a realizar y contando con un espacio temporal de respaldo.        |
| 4  | Error al analizar los pasos realizadas por los algoritmos de corte de materiales, que cause la redefinición de la aplicación | 0.2          | 0.6     | Programar un prototipo con los pasos realizados y probar su funcionamiento comparando el algoritmo genético usado como caso de estudio. Intentar cubrir todos los pasos. |
| 5  | Cambios en los términos de uso y las licencias de las herramientas utilizadas  | 0.1          | 0.7     | Tener herramientas de reemplazo  |

## 3 Justificación y Viabilidad



## 3.1 Justificación

El objetivo de este proyecto es de brindar una herramienta informática que minimice el desperdicio de materiales en la manufactura de barras de acero. Con esto, se espera beneficiar a la industria a través de una reducción de costos, debido al aumento en la eficiencia.

Como se ha observado en la revisión del estado del arte, una gran cantidad de variantes de algoritmos de corte de materiales han sido explorados por diversos investigadores. Sin embargo, las investigaciones se han centrado en presentar algoritmos arquetípicos que son aplicados a diferentes problemas, sin un mayor grado de especialización, que incluya, por ejemplo, fases de mejora de soluciones en base al conocimiento del caso específico.

Este proyecto permitirá aplicar con efectividad algoritmos de corte de materiales, balanceando el uso de operaciones de generación de soluciones con operadores específicos al problema de corte de barras de acero. Este proyecto utiliza los llamados Algoritmos Meméticos.

Por lo que el desarrollo de este proyecto se justifica en la implementación de una solución especializada para resolver el problema de corte de materiales en la producción de barras de acero para hormigón.

El producto final es una aplicación de escritorio y la documentación de los pasos seguidos para su obtención.

También se provee una solución que puede ser tomada de referencia en proyectos futuros de desarrollo, implementación o adaptación.

## 3.2 Análisis de Viabilidad

### 3.2.1 Viabilidad Técnica

A lo largo de la educación superior y la experiencia laboral se ha hecho uso de las diversas herramientas, métodos y metodologías utilizadas en este proyecto.

El uso de los entornos de desarrollo y los lenguajes de programación es habitual y las pruebas científicas han sido desarrolladas en proyectos afines tanto académicos como profesionales. La metodología de Extreme Programming ha sido estudiada por el autor de este proyecto y también se cuenta con experiencia en el desarrollo de software.

### 3.2.2 Viabilidad Económica

Para el desarrollo del proyecto se hace uso de un computador de escritorio estándar, una licencia gratuita de los entornos de desarrollo y herramientas de uso libre. Además, el proyecto presentado cuenta con un fondo de dinero extra.

#### Viabilidad Económica (Autoría Propia)

| Herramienta   | Tipo  | Costo (USD) | Costo Total (USD) |
|---|---|-------------|-------------------|
| Tiempo de programación                              | Servicio (6 meses de programación)                  | 300         | 1800              |
| Computador  | Producto  | 300         | 300               |
| Software 1: Visual Studio 2015 (Versión gratuita)   | Licencia de Software (gratuita)                     | 0           | 0                 |
| Microsoft Excel (Licencia de la PUCP)               | Licencia de software (a cuenta de la PUCP)          | 0           | 0                 |
| Librería en línea: Scopus (Licencia de la PUCP)     | Licencia de Librería en línea (a cuenta de la PUCP) | 0           | 0                 |
| Conexión a internet (Desde computadores de la PUCP) | Servicio de internet (a cuenta de la PUCP)          | 0           | 0                 |
|   |   | Total       | 2100              |
|   |   | Fondo Extra | 400               |

### 3.2.3 Viabilidad Temporal

Este proyecto se desarrolla a lo largo de dos ciclos académicos semestrales. El primer ciclo es usado para el planteamiento del proyecto, documentación y aprobación del tema. El segundo ciclo empieza con la implementación del sistema y la entrega de los avances desarrollados. El cronograma que se sigue es el siguiente

#### Cronograma del proyecto (Autoría Propia)

| Actividad | Inicio | Fin |
|-----------|--------|-----|
|-----------|--------|-----|

|  |            |            |
|--|------------|------------|
| Planificación del Proyecto   |            |            |
| Definición del proyecto de fin de carrera  | 18/01/2020 | 25/01/2020 |
| Investigación y definición de métodos y herramientas a utilizar                  | 26/01/2020 | 10/02/2020 |
| Análisis y Diseño  |            |            |
| Investigación de actividades realizadas por el algoritmo de corte de materiales. | 11/02/2020 | 19/02/2020 |
| Identificación de los pasos que pueden ser adaptados a un algoritmo memético     | 20/02/2020 | 3/03/2020  |
| Elección del algoritmo a adaptar   | 4/03/2020  | 8/03/2020  |
| Planteamiento del algoritmo  | 9/03/2020  | 16/03/2020 |
| Documentación del planteamiento del sistema                                      | 17/03/2020 | 18/03/2020 |
| Implementación   |            |            |
| Primera versión  | 19/03/2020 | 25/03/2020 |
| Evaluación de la primera versión   | 26/03/2020 | 30/03/2020 |
| Documentación sobre la primera versión   | 31/03/2020 | 1/04/2020  |
| Segunda versión  | 2/04/2020  | 9/04/2020  |
| Evaluación de la segunda versión   | 10/04/2020 | 13/04/2020 |
| Documentación sobre la segunda versión   | 14/04/2020 | 15/04/2020 |
| Tercera versión  | 16/04/2020 | 23/04/2020 |
| Evaluación de la tercera versión   | 24/04/2020 | 27/04/2020 |
| Documentación sobre la tercera versión   | 28/04/2020 | 1/05/2020  |
| Independización de la aplicación   | 2/05/2020  | 7/05/2020  |
| Cuarta versión   | 8/05/2020  | 9/05/2020  |
| Evaluación de la cuarta versión  | 10/05/2020 | 13/05/2020 |
| Documentación sobre la cuarta versión  | 14/05/2020 | 15/05/2020 |
| Quinta versión   | 16/05/2020 | 23/05/2020 |

|   |            |            |
|---|------------|------------|
| Evaluación de la cuarta versión         | 24/05/2020 | 27/05/2020 |
| Documentación sobre la cuarta versión   | 28/05/2020 | 29/05/2020 |
| Evaluación                              | 30/05/2020 | 31/05/2020 |
| Elaboración de pruebas de la aplicación | 1/06/2020  | 3/06/2020  |
| Evaluación de la aplicación             | 4/06/2020  | 7/06/2020  |
| Documentación final                     | 8/06/2020  | 13/06/2020 |

