



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA INGENIERÍA EN SISTEMAS

Proyecto de Investigación previo
a la obtención del título de
Ingeniero en Sistemas.

Título del Proyecto de Investigación:

**“OPTIMIZACIÓN MULTI OBJETIVO BASADA EN MALLAS VARIABLES
(MOVMO) APLICADA A LA INFERENCIA DE ÁRBOLES FILOGENÉTICOS”**

Autor:

Carlos Alfredo Hidalgo Reyes

Director de Proyecto de Investigación:

Ing. Cristian Gabriel Zambrano Vega, MSc.

Quevedo – Los Ríos - Ecuador.

2017

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Carlos Alfredo Hidalgo Reyes**, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

f. _____

Carlos Alfredo Hidalgo Reyes

C.C. # 1205420498

CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN



El suscrito, **Ing. Cristian Gabriel Zambrano Vega, MSc.** Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante **Carlos Alfredo Hidalgo Reyes**, realizó el Proyecto de Investigación de grado titulado “**OPTIMIZACIÓN MULTIOBJETIVO BASADA EN MALLAS VARIABLES (MOVMO) APLICADA A LA INFERENCIA DE ÁRBOLES FILOGENÉTICOS**”, previo a la obtención del título de Ingeniero en Sistemas, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

.....

Ing. Cristian Gabriel Zambrano Vega, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO



Ing. Gabriel Zambrano Vega, MSc en calidad de Director de Proyecto de Investigación titulado **“OPTIMIZACIÓN MULTI OBJETIVO BASADA EN MALLAS VARIABLES (MOVMO) APLICADA A LA INFERENCIA DE ÁRBOLES FILOGENÉTICOS”**, me permito manifestar a usted y por intermedio del Consejo Académico lo siguiente:

Que, el señor **CARLOS ALFREDO HIDALGO REYES**, egresado de la Facultad de Ciencias de la Ingeniería, carrera Ingeniería en Sistemas, ha cumplido con las correcciones pertinentes, e ingresado su Proyecto de Investigación al sistema URKUND, tengo bien certificar la siguiente información sobre el informe del sistema anti plagio con un porcentaje de 3%.

URKUND	
Documento	Proyecto Investigativo PhylMOVMO.docx (D28288098)
Presentado	2017-05-16 1:39:00 (-05:00)
Presentado por	Cristian Gabriel Zambrano Vega (czambrano@uteq.edu.ec)
Recibido	cristianzambrano.uteq@analysis.orkund.com
Mensaje	Mostrar el mensaje completo
	3% de esta aprox. 52 páginas de documentos largos se componen de texto presente en 2 fuentes.

.....
Ing. Cristian Gabriel Zambrano Vega, MSc.
DIRECTOR DEL PROYECTO DE INVESTIGACIÓN



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERIA EN SISTEMAS

PROYECTO DE INVESTIGACIÓN

Título:

“OPTIMIZACIÓN MULTIOBJETIVO BASADA EN MALLAS VARIABLES
(MOVMO) APLICADA A LA INFERENCIA DE ÁRBOLES FILOGENÉTICOS”

Presentado al Consejo Académico como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado por:

PRESIDENTE DEL TRIBUNAL
PhD. Byron Wladimir Oviedo Bayas

MIEMBRO DEL TRIBUNAL
PhD. Amilkar Puris Cáceres

MIEMBRO DEL TRIBUNAL
PhD. Pavel Novoa Hernández

QUEVEDO – LOS RIOS – ECUADOR

2017

AGRADECIMIENTO

Para alcanzar un objetivo hay que ser determinado y constante, sin lugar a duda contar con el apoyo de quienes te rodean y forman parte de tu vida, por lo tanto, hoy les brindo mis más sinceros agradecimientos:

A mis padres, quienes me han brindado su amor incondicional, inculcando en mí un sinnúmero de valores y buenos principios, apoyándome en cada paso que eh dado durante todos mis años de estudio.

A mi hijo, mi alegría y mi motor aquel que me impulsa a seguir luchando día con día por superarme y culminar con éxito este proyecto, y más que nada doy gracias a Dios por haberme dado la dicha de ser su padre y darme sabiduría para sobrellevar cada obstáculo que se hizo presente a lo largo del camino.

Le agradezco a mi Director de Proyecto de Investigación, Ing. Cristian Zambrano Vega, MSc. un excelente docente que me ha brindado sus conocimientos, su tiempo y buenos consejos durante todo el proceso de desarrollo del proyecto, de la misma manera le agradezco a cada uno de los docentes que formaron parte de mi carrera universitaria, siendo base fundamental en la adquisición de mis conocimientos profesionales y valores éticos.

DEDICATORIA

A mis padres

A mi hijo

A mis hermanos

RESUMEN

Uno de los problemas con mayor relevancia en la Bioinformática y Biología Computacional es la búsqueda y reconstrucción del árbol filogenético más preciso que explica, lo más exactamente posible, las relaciones evolutivas entre especies de un determinado conjunto de datos. Se han empleado diferentes criterios para evaluar la exactitud de la hipótesis evolutiva con el fin de guiar un algoritmo de búsqueda hacia el mejor árbol. Sin embargo, estos criterios pueden conducir a filogenias distintas, que a menudo son conflictivas entre ellas. Por lo tanto, un enfoque multiobjetivo puede ser útil.

En este trabajo se presenta una adaptación filogenética de un algoritmo de Optimización Multiobjetivo basada en Mallas Variables (MOVMO) para la inferencia de filogenias, con la finalidad de abordar el problema de la inferencia filogenética según dos criterios de optimalidad: parsimonia máxima y máxima verosimilitud.

El objetivo de este enfoque es proponer una visión complementaria de la filogenética de acuerdo con los criterios de máxima parsimonia y máxima verosimilitud, con el fin de generar un conjunto diverso de topologías filogenéticas que representan un consenso entre ambos criterios de optimalidad. Los experimentos sobre cuatro conjuntos de datos de nucleótidos reales muestran que la propuesta puede lograr resultados prometedores, tanto en enfoques multiobjetivos como biológicos, con respecto a otras metaheurísticas multiobjetivos clásicas y recientes a partir del estado del arte.

ABSTRACT

One of the most relevant problems in Bioinformatics and Computational Biology is the search and reconstruction of the most accurate phylogenetic tree that explains, more exactly possible, the evolutionary relationships among species from a given dataset. Different criteria have been employed to evaluate the accuracy of evolutionary hypothesis in order to guide a search algorithm towards the best tree. However, these criteria may lead to distinct phylogenies, which are often conflicting among them. Therefore, a multiobjective approach can be useful.

In this work, we present a phylogenetic adaptation of the algorithm multiobjective optimization based variable meshes (MOVMO) for inferring phylogenies, with finality to tackle the phylogenetic inference problem according to two optimality criteria: maximum parsimony and maximum likelihood.

The aim of this approach is to propose a complementary view of phylogenetics according to the maximum parsimony and maximum likelihood criteria, in order to generate a set of trade-off phylogenetic topologies that represent a consensus between both optimality criteria. Experiments on four real nucleotide datasets show that our proposal can achieve promising results, under both multiobjective and biological approaches, with regard to other classical and recent multiobjective metaheuristics from the state-of-the-art.

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO I.....	3
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN	3
1.1. Problema de investigación	4
1.1.1. Planteamiento del problema	4
1.1.2. Formulación del problema	5
1.1.3. Sistematización del problema	5
1.2. Objetivos	5
1.2.1. Objetivo General	5
1.2.2. Objetivos Específicos.....	5
1.3. Justificación	6
CAPÍTULO II	7
FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	7
2.1. Marco conceptual	8
2.1.1. Inteligencia Artificial.....	8
2.1.2. Metaheurísticas.....	9
2.1.3. Clasificación de las metaheurísticas.....	9
2.1.4. Metaheurísticas para optimización multiobjetivo	18
2.1.5. Óptimo de Pareto.....	19
2.1.6. Dominancia de Pareto.....	20
2.1.7. Frente de Pareto.....	21
2.1.8. Objetivos en la resolución de MOPs	21
2.1.9. Evaluación estadística de resultados	22
2.1.10. Análisis estadístico de resultados	24
2.1.11. Filogenética	24
2.1.12. Árbol Filogenético	25

2.1.13.	Inferencia Filogenética	25
2.1.14.	Framework de optimización multiobjetivo (jMetalCpp)	29
2.1.15.	Framework bioinformático (BIO++)	29
2.1.16.	Framework Filogenético (PLL)	29
2.1.17.	Repositorio TreeBASE.....	30
2.1.18.	Optimización Multiobjetivo Basada en Mallas Variables (MOVMO)	31
2.1.19.	Algoritmo Evolutivo Multiobjetivo basado en la Descomposición (MOEA/D).....	31
2.1.20.	Algoritmo Genético de Ordenación No-dominada (NSGA-II)	32
2.1.21.	Algoritmo de Selección Multiobjetivo basado en Hipervolumen Dominado (SMS-EMOA) 32	
2.2.	Marco referencial	33
2.2.1.	Implementación de la principal referencia metaheurística de optimización multiobjetivo NSGA-II aplicada a la filogenética computacional.....	33
2.2.2.	Propuesta multiobjetivo basado en el algoritmo de luciérnaga (MO-FA) para inferir filogenias	35
2.2.3.	Adaptación multiobjetivo del algoritmo Artificial Bee Colony (MOABC) aplicada a la inferencia filogenética.....	36
2.2.4.	MO-Phylogenetics: Una Herramienta de Software de la Inferencia Filogenética con Metaheurísticas Evolutivos Multiobjetivo	37
CAPÍTULO III.....		39
MÉTODOLOGÍA DE LA INVESTIGACIÓN.....		39
3.1.	Tipo de investigación.....	40
3.2.	Métodos de investigación	40
3.3.	Población y Muestra.....	40
3.4.	Fuentes de recopilación de información.....	41
3.5.	Recursos humanos y materiales.....	41
3.5.1.	Recursos Humanos.....	41
3.5.2.	Recursos de Hardware	42
3.5.3.	Recursos de Software.....	42

3.5.4. Insumos/Materiales	43
CAPÍTULO IV	44
RESULTADOS Y DISCUSIÓN.....	44
4.1. Integración de frameworks del área de la bioinformática al algoritmo MOVMO	45
4.2. Adaptación Filogenética del algoritmo MOVMO	46
4.2.1. Representación de los individuos.....	46
4.2.2. Operador de Cruce	47
4.2.3. Distancia entre árboles filogenéticos	47
4.2.4. Técnica de Optimización Filogenética	48
4.2.5. Parámetros de Entradas y Salida.....	49
4.2.6. Generación de la malla inicial	50
4.2.7. Generación de nodos mediante búsqueda local.....	51
4.2.8. Generación de nodos mediante búsqueda global	52
4.2.9. Operador de recombinación	52
4.2.10. Evaluación y actualización de los archivos líder	53
4.2.11. Actualización dinámica de la población	54
4.3. Estudio experimental del algoritmo propuesto con el estado del arte	54
4.3.1. Análisis Multiobjetivos	55
4.3.2. Frente de Pareto de Referencia y Aproximaciones de los Frentes de Pareto.....	56
4.3.3. Análisis Biológico	59
4.3.4. Análisis de Pruebas Estadísticas	62
4.4. Discusión	67
CAPITULO V	68
CONCLUSIONES Y RECOMENDACIONES	68
6.1. Conclusiones.....	69
6.2. Recomendaciones	70
CAPITULO VI.....	71
BIBLIOGRAFÍA.....	71

6.1. Bibliografía	72
CAPITULO VII	79
ANEXOS.....	79

ÍNDICE DE ILUSTRACIONES

Ilustración 1.- Clasificación de la metaheurística.....	9
Ilustración 2.- Ejemplo del concepto de Dominancia de Pareto.....	20
Ilustración 3.- Ejemplo de mala convergencia (a) y diversidad (b) en Frente de Pareto.	22
Ilustración 4.- Estructura de un árbol bifurcado.	46
Ilustración 5.- Ejemplo del operador de cruce PDG.....	47
Ilustración 6.- Métrica de Robinson-Foulds	48
Ilustración 7.- Aproximaciones de Frentes de Pareto de los mejores valores de IHV obtenidos por todos los algoritmos (PhyloMOVMO, NSGAI, MOEA / D y SMSEMOA) a lo largo de las 10 secuencias independientes que resuelven cada conjunto de datos de nucleótidos.	57
Ilustración 8.- Aproximaciones de Frentes de Pareto de los mejores valores de IDG+ obtenidos por todos los algoritmos (PhyloMOVMO, NSGAI, MOEA / D y SMSEMOA) a lo largo de las 10 secuencias independientes que resuelven cada conjunto de datos de nucleótidos.	58
Ilustración 9.- Frentes de Pareto de Referencia y las mejores aproximaciones del Frente de Pareto obtenidos por todos los algoritmos en las 10 ejecuciones independientes resolviendo los problemas a) rbcL_55, b) mtDNA_186, c) RDPII_218 y d) ZILLA_500.	59
Ilustración 10.- Comparación de los árboles filogenéticos (Mejor Parsimonia (Izquierda), Mejor Verosimilitud (Derecha) del algoritmo PhyloMOVMO para el conjunto de datos rbcL_55.	60
Ilustración 11.- Comparación de los árboles filogenéticos (Mejor Parsimonia (Izquierda), Mejor Verosimilitud (Derecha) del algoritmo PhyloMOVMO para el conjunto de datos mtDNA_186.	61

ÍNDICE DE TABLAS

Tabla 1.- Conjunto de datos de nucleótidos.	41
Tabla 2.- Recursos Humanos.	41
Tabla 3.- Recursos de Hardware.	42
Tabla 4.- Recursos de Software.	42
Tabla 5.- Insumos/Materiales.	43
Tabla 6.- Mediana y Rangos Intercuartílicos de los valores del indicador IHV.	55
Tabla 7.- Mediana y Rangos Intercuartílicos de los valores del indicador IE+.	56
Tabla 8.- Mediana y Rangos Intercuartílicos de los valores del indicador IΔ.	56
Tabla 9.- Mediana y Rangos Intercuartílicos de los valores del indicador IDG+.	56
Tabla 10.- Mejores Puntuaciones (Máxima Parsimonia MP, Máxima Verosimilitud ML).	60
Tabla 11.- Resultado de la prueba de Friedman del indicador IHV.	63
Tabla 12.- Resultado de la prueba de Wilcoxon del indicador IHV para los conjuntos de datos rbcL_55, mtDNA_186, RDPII_218 y ZILLA_500.	64
Tabla 13.- Resultado de la prueba de Wilcoxon del indicador IE+ para los conjuntos de datos rbcL_55, mtDNA_186, RDPII_218 y ZILLA_500.	64
Tabla 14.- Resultado de la prueba de Wilcoxon del indicador IΔ para los conjuntos de datos rbcL_55, mtDNA_186, RDPII_218 y ZILLA_500.	64
Tabla 15.- Resultado de la prueba de Wilcoxon del indicador IIDG+ para los conjuntos de datos rbcL_55, mtDNA_186, RDPII_218 y ZILLA_500.	65
Tabla 16.- Resultado de la prueba de Wilcoxon del indicador IHV.	66
Tabla 17.- Resultado de la prueba de Wilcoxon del indicador IE+.	66
Tabla 18.- Resultado de la prueba de Wilcoxon del indicador IΔ.	66
Tabla 19.- Resultado de la prueba de Wilcoxon del indicador IIDG+.	67

ÍNDICE DE PSEUDOCÓDIGOS

Pseudocódigo 1.- Algoritmo NSGA-II aplicada a la filogenética.....	34
Pseudocódigo 2.- Algoritmo MOFA aplicada a la filogenética.	35
Pseudocódigo 3.- Algoritmo MOABC aplicada a la filogenética.....	37
Pseudocódigo 4.- Parámetros de E/S de MOVMO.....	49
Pseudocódigo 5.- Parámetros de E/S de PhyloMOVMO.....	50
Pseudocódigo 6.- Generación de la malla inicial en el algoritmo MOVMO.....	50
Pseudocódigo 7.- Generación de la malla inicial en el algoritmo PhyloMOVMO.....	51
Pseudocódigo 8.- Generación de nodos mediante búsqueda local en el algoritmo MOVMO.	51
Pseudocódigo 9.- Generación de nodos mediante búsqueda local en el algoritmo PhyloMOVMO.....	52
Pseudocódigo 11.- Generación de nodos mediante búsqueda global en el algoritmo PhyloMOVMO.....	52
Pseudocódigo 12.-Proceso de recombinación en el algoritmo MOVMO.	52
Pseudocódigo 13.-Proceso de recombinación en el algoritmo PhyloMOVMO.	53
Pseudocódigo 14.- Evaluación y actualización de los archivos líder en el algoritmo PhyloMOVMO.....	53
Pseudocódigo 15.- Actualización dinámica de la población en el algoritmo PhyloMOVMO.....	54

CÓDIGO DUBLIN

Título:	OPTIMIZACIÓN MULTI OBJETIVO BASADA EN MALLAS VARIABLES (MOVMO) APLICADA A LA INFERENCIA DE ÁRBOLES FILOGENÉTICOS				
Autor:	Hidalgo Reyes, Carlos Alfredo				
Palabras clave:	Inferencia Filogenética	Computación Evolutiva	Optimización Multiobjetivo	Bioinformática	Biología Computacional
Fecha de publicación:					
Editorial:					
Resumen:	<p>Resumen. - Uno de los problemas con mayor relevancia en la Bioinformática y Biología Computacional es la búsqueda y reconstrucción del árbol filogenético más preciso que explica, lo más exactamente posible, las relaciones evolutivas entre especies de un determinado conjunto de datos. Se han empleado diferentes criterios para evaluar la exactitud de la hipótesis evolutiva con el fin de guiar un algoritmo de búsqueda hacia el mejor árbol. Sin embargo, estos criterios pueden conducir a filogenias distintas, que a menudo son conflictivas entre ellas. Por lo tanto, un enfoque multiobjetivo puede ser útil. En este trabajo se presenta una adaptación filogenética del algoritmo de Optimización Multiobjetivo basada en Mallas Variables (MOVMO) para la inferencia de filogenias, con la finalidad de abordar el problema de la inferencia filogenética según dos criterios de optimalidad: parsimonia máxima y máxima verosimilitud (...).</p> <p>Abstract. - One of the most relevant problems in Bioinformatics and Computational Biology is the search and reconstruction of the most accurate phylogenetic tree that explains, more exactly possible, the evolutionary relationships among species from a given dataset. Different criteria have been employed to evaluate the accuracy of evolutionary hypothesis in order to guide a search algorithm towards the best tree. However, these criteria may lead to distinct phylogenies, which are often conflicting among them. Therefore, a multiobjective approach can be useful. In this work, we present a phylogenetic adaptation of the algorithm multiobjective optimization based variable meshes (MOVMO) for inferring phylogenies, with finality to tackle the phylogenetic inference problem according to two optimality criteria: maximum parsimony and maximum likelihood (...).</p>				
Descripción:					
URI:	<u>(en blanco hasta cuando se dispongan los repositorios)</u>				

INTRODUCCIÓN

La historia evolutiva de la humanidad y de todas las otras especies vivas y extintas en la tierra es una cuestión que ha estado preocupando a la humanidad durante siglos. Por lo tanto, la construcción de un "árbol de la vida" que comprende todos los organismos vivos y extintos en la tierra ha sido una idea fascinante y desafiante desde la aparición de la teoría evolutiva [1].

Típicamente, las relaciones evolutivas entre los organismos están representadas por un árbol evolutivo. La inferencia filogenética consiste en encontrar el mejor árbol que explique las relaciones genealógicas o la historia evolutiva de las especies a partir de secuencias moleculares (datos de ADN o proteína). Los datos utilizados en este análisis suelen provenir de secuencias alineadas de nucleótidos o aminoácidos llamada alineamiento múltiple de secuencias. [2] [3].

Varios campos científicos pueden beneficiarse gracias a las contribuciones de la filogenética, como la biología evolutiva, la fisiología, la ecología, la paleontología, la biomedicina, la química y otros [4]. Por todo esto, muchos científicos están de acuerdo en que la inferencia filogenética es uno de los temas de investigación más importantes en la Bioinformática.

Handl et al. [5] discutió las aplicaciones de la optimización multiobjetivo en varios problemas de bioinformática y biología computacional, en esta encuesta la inferencia filogenética es uno de los problemas centrales en esta área. Desafortunadamente, muchos problemas y algoritmos interesantes en la Bioinformática, tales como la inferencia de filogenias perfectas o la alineación óptima de la secuencia múltiple son NP-completos y de cómputo extremadamente intensivo. [5]

En la actualidad existen varias propuestas de metaheurísticas multiobjetivos orientadas a la exploración para ser aplicadas a la inferencia filogenética: MO-FA, un algoritmo inspirado por la bioluminiscencia de las luciérnagas [6], MOABC, un algoritmo inspirado en el comportamiento natural de las colonias de abejas [7], también existen dos técnicas basadas en algoritmos evolutivos: PhyloMOEA [8] primer algoritmo creado para resolver problemas de inferencia filogenética, MO-Phylogenetics [9], NSGAI [10], el paralelismo es otra de las técnicas que se han utilizado en la bioinformática: MOP [11].

En este trabajo se presenta una adaptación filogenética del algoritmo de Optimización Multiobjetivo basada en Mallas Variables [12] llamado PhyloMOVMO, para inferir árboles filogenéticos que optimizan dos criterios de optimalidad, simultáneamente: Máxima Parsimonia y Máxima Verosimilitud, con el objetivo de permitir a los biólogos deducir en una sola ejecución un conjunto diverso de topologías filogenéticas que representan un consenso entre diferentes puntos de ambos criterios de optimalidad.

Con el fin de evaluar el desempeño de la propuesta, se ha llevado a cabo experimentos en cuatro conjuntos de datos de nucleótidos (ADN), comparando los resultados multiobjetivos y biológicos con otras metaheurísticas multiobjetivos populares y recientes aplicando métricas de calidad multiobjetivo. PhyloMOVMO se ha implementado utilizando funcionalidades del framework de MO-Phylogenetics [13], una herramienta de software de inferencia filogenética con metaheurísticas evolutivas multiobjetivo. Los algoritmos a comparar, los problemas de pruebas, los archivos de configuraciones y parámetros se tomaron de este framework.

CAPÍTULO I
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1. Problema de investigación

1.1.1. Planteamiento del problema

En la bioinformática y la biología computacional uno de los problemas más importantes es la búsqueda y reconstrucción de árboles filogenéticos. La Inferencia filogenética es considerada como un problema de complejidad NP-completo por la dificultosa exploración del espacio de búsqueda cuyo tamaño incrementa exponencialmente por cada una de las especies en análisis.

Las metaheurísticas son métodos heurísticos de solución de problemas que adaptan mejoras en las operaciones de búsqueda local y estrategias de alto nivel pudiendo así abarcar búsquedas robustas de espacios de soluciones, sin embargo, no garantizan encontrar la solución óptima al problema, por esta razón se siguen desarrollando métodos que mediante la incorporación de nuevas estrategias mejoren las aproximaciones de Frente de Pareto. Los Frente de Pareto no solo nos permiten obtener una solución sino un conjunto de soluciones no-dominadas a un problema. [14]

La Optimización Basada en Mallas Variables (VMO), es una metaheurística poblacional que fue introducida en el ámbito de la optimización continua [15]. Su forma de exploración involucra dos operadores fundamentales, la expansión y contracción de una población de soluciones. Recientemente se ha presentado en la literatura una variante para resolver problemas multiobjetivo (MOVMO) [16], por lo cual se debe de realizar un análisis para determinar que tecnologías se deben incorporar para que pueda resolver problemas de inferencias filogenéticas desde una lógica multiobjetivo.

Diagnóstico

En la comunidad bioinformática los investigadores no cuentan con un método que aplique una metaheurística de Optimización Multiobjetivo basada en Mallas Variables con la finalidad de poder encontrar árboles filogenéticos inferidos aplicando al mismo tiempo los criterios de reconstrucción de árboles filogenéticos, máxima parsimonia y máxima verosimilitud.

1.1.2. Formulación del problema

¿Cómo obtener árboles filogenéticos inferidos mediante el algoritmo de Optimización Multiobjetivo basada en Mallas Variables (MOVMO)?

1.1.3. Sistematización del problema

- ¿Cómo aplicar el algoritmo MOVMO a la inferencia de árboles filogenético?
- ¿Cómo obtener soluciones filogenéticas del algoritmo MOVMO?
- ¿Cómo evaluar el rendimiento del algoritmo MOVMO resolviendo problemas filogenéticos?

1.2. Objetivos

1.2.1. Objetivo General

Realizar un estudio experimental para determinar cómo obtener árboles filogenéticos inferidos mediante el algoritmo de Optimización Multiobjetivo basada en Mallas Variables (MOVMO).

1.2.2. Objetivos Específicos

- Integrar al algoritmo MOVMO las funcionalidades de varios frameworks del área de la bioinformática.
- Adaptar el algoritmo Optimización Multiobjetivo basada en Mallas Variables (MOVMO) para resolver problemas de inferencia filogenética.
- Realizar experimentos del algoritmo implementado, basado en criterios de rendimiento y reconstrucción de árboles filogenético, mediante análisis multiobjetivo y biológico, respectivamente.

1.3. Justificación

La propuesta que se plantea en este tema de investigación se basa en la técnica de Optimización basada en Mallas Variables (VMO), la misma que ha sido extendida de mono-objetivo a multiobjetivos (MOVMO) obteniendo resultados considerables frente a otros algoritmos de referencia [17].

Por esta razón, y teniendo en cuenta que el problema de la inferencia filogenética es uno de los temas más tratados en el área de la biología por su utilidad en el análisis de la historia evolutiva y las relaciones genealógicas de las especies se presenta esta investigación para que se tome dicho algoritmo para ser adaptado a resolver problemas de inferencias filogenéticas.

Esta adaptación permitirá contribuir con un nuevo método para resolver problemas filogenéticos la misma que puede ser muy útil para la comunidad bioinformática, les proporcionará a los investigadores no solo una solución de árbol filogenético sino un conjunto de árboles filogenéticos optimizados mediante la máxima parsimonia y la máxima verosimilitud, brindándole así hipótesis alternativas útiles desde diferentes puntos de vistas biológicos.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

2.1. Marco conceptual

2.1.1. Inteligencia Artificial

Para abordar el concepto de inteligencia artificial, tal vez cabría plantearse primero la siguiente pregunta: “¿qué es la inteligencia?” Sin duda, se trata de una pregunta difícil cuya respuesta aún no ha sido resuelta totalmente, la cual sigue desconcertando tanto a los biólogos como a los psicólogos y filósofos de nuestra época. Por supuesto que el objetivo de este libro no es zanjar la discusión. Más bien la intención es presentar algunas ideas en torno a la noción de inteligencia que nos ayuden a identificar ciertas características distintivas de la denominada inteligencia artificial (IA). [18]

Se podría comenzar por destacar algunas propiedades generales que presenta la inteligencia humana, como por ejemplo la habilidad de enfrentar nuevas situaciones, la habilidad de resolver problemas, de responder preguntas, elaborar planes, etc. Desde sus inicios, el hombre se representó el mundo real mediante símbolos, los cuales constituyen la base del lenguaje humano. En este sentido, se podría considerar a la IA como un dialecto simbólico constituido por cadenas de caracteres que representan conceptos del mundo real. De hecho, los procesos simbólicos son una característica esencial de la IA. A partir de lo expuesto es posible formular una definición más aproximada de nuestro objeto de estudio: la IA es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas. [18]

El primer intento de definir la Inteligencia Artificial lo hizo el matemático Alan Turing, que es considerado como el padre de la computación. Este científico inglés es más conocido por su máquina de Turing: una máquina conceptual que utilizó para formalizar los conceptos del modelo computacional que seguimos utilizando hoy día. En concreto demostró que con las operaciones básicas que podía desarrollar su máquina podía codificarse cualquier algoritmo, y que toda máquina capaz de computar tendría las mismas operaciones básicas que su máquina o un súper conjunto de estas. [19]

2.1.2. Metaheurísticas

Las metaheurísticas son procedimientos generales que permiten generar soluciones aproximadas a problemas. Aunque se suele considerar como metaheurística a toda aquella generalización de una determinada heurística para cualquier tipo de problema, en la práctica es necesario analizar detalladamente el problema a resolver para determinar cuál de ellas es la que se presupone, y puede tener un mayor éxito en la búsqueda de soluciones, si bien no se pueda determinar con total seguridad si alcanzaremos dicho éxito. [20]

Una metaheurística es un método heurístico para resolver un tipo de problema computacional general, usando los parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera eficiente. Normalmente, estos procedimientos son heurísticos. [21]

2.1.3. Clasificación de las metaheurísticas

Hay diferentes formas de clasificar y describir las técnicas metaheurísticas [22]. Dependiendo de las características que se seleccionen se pueden obtener diferentes taxonomías basadas en la naturaleza y no basada en la naturaleza, con memoria o sin ella, con una o varias estructuras de vecindario, etc. Una de las clasificaciones más populares las divide en metaheurísticas basadas en trayectoria y basadas en población. Las primeras manipulan en cada paso un único elemento del espacio de búsqueda, mientras que las trabajan sobre un conjunto de ellos (población) [23].

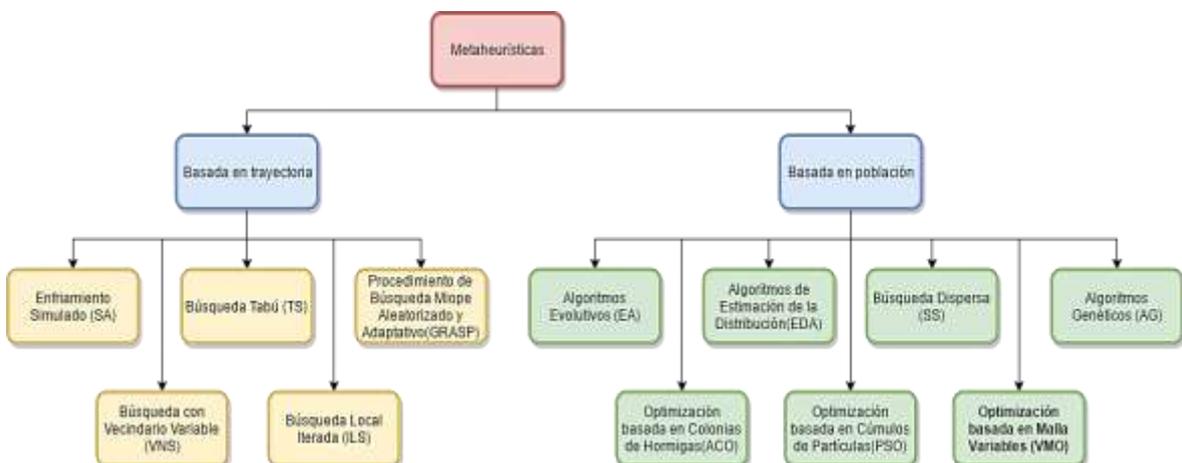


Ilustración 1.- Clasificación de la metaheurística.

2.1.3.1. Metaheurísticas basadas en trayectoria

La principal característica de estos métodos es que parten de una solución y, mediante la exploración del vecindario, van actualizando la solución actual, formando una trayectoria. La mayoría de estos algoritmos surgen como extensiones de los métodos simples de búsqueda local a los que se les añade algún mecanismo para escapar de los mínimos locales. Esto implica la necesidad de una condición de parada más elaborada que la de encontrar un mínimo local. Normalmente se termina la búsqueda cuando se alcanza un número máximo predefinido de iteraciones, se encuentra una solución con una calidad aceptable, o se detecta un estancamiento del proceso [23].

2.1.3.1.1. Enfriamiento Simulado (SA)

El enfriamiento simulado o Simulated Annealing (SA) es una de las técnicas más antiguas entre las metaheurísticas y posiblemente es el primer algoritmo con una estrategia explícita para escapar de los mínimos locales. Los orígenes del algoritmo se encuentran en un mecanismo estadístico, denominado metropolis [24]. La idea del SA es simular el proceso de enfriamiento del metal y del cristal [23].

El SA fue inicialmente presentado en [25]. Para evitar quedar atrapado en un mínimo local, el algoritmo permite elegir con cierta probabilidad una solución cuyo valor de fitness sea peor que el de la solución actual. En cada iteración se elige, a partir de la solución actual s , una solución s' del vecindario $N(s)$. Si s' es mejor que s (es decir, tiene un mejor valor en la función de fitness), se sustituye s por s' como solución actual. Si la solución s' es peor, entonces es aceptada con una determinada probabilidad que depende de la temperatura actual T y de la diferencia de fitness entre ambas soluciones, $f(s') - f(s)$ (caso de minimización) [23].

2.1.3.1.2. Búsqueda Tabú (TS)

La búsqueda tabú o Tabu Search (TS) es una de las metaheurísticas que se han aplicado con más éxitos a la hora de resolver problemas de optimización combinatoria. Los fundamentos de este método fueron introducidos en [26], y están basados en las ideas formuladas en [27]. Un buen resumen de esta técnica y sus componentes se puede encontrar en [27] [23].

La idea básica de la búsqueda tabú es el uso explícito de un historial de la búsqueda (una memoria a corto plazo), tanto para escapar de los mínimos locales como para implementar su estrategia de exploración y evitar buscar varias veces en la misma región. Esta memoria a corto plazo se implementa con una lista tabú, donde se mantienen las soluciones visitadas más recientemente para excluirlas de los próximos movimientos. En cada iteración se elige la mejor solución entre las permitidas y la solución es añadida a la lista tabú [23].

Desde el punto de vista de la implementación, mantener una lista de soluciones completas no suele ser práctico debido a su ineficiencia. Por lo tanto, en general, se suelen almacenar los movimientos que ha llevado al algoritmo a generar esa solución o los componentes principales que definen la solución. En cualquier caso, los elementos de esta lista permiten filtrar el vecindario, generando un conjunto reducido de soluciones elegibles denominado $N_a(s)$ [23].

El almacenamiento de los movimientos en vez de las soluciones completas es bastante más eficiente, pero introduce una pérdida de información. Para evitar este problema, se define un criterio de aspiración que permite incluir una solución en $N_a(s)$ incluso si está prohibida debido a la lista tabú. El criterio de aspiración más ampliamente usado es permitir soluciones cuyo fitness sea mejor que el de la mejor solución encontrada hasta el momento [23].

2.1.3.1.3. Procedimiento de Búsqueda Miope Aleatorizado y Adaptativo (GRASP)

Greedy Randomized Adaptive Search Procedure (GRASP) [28] es una metaheurística simple que combina heurísticos constructivos con búsqueda local. GRASP es un procedimiento iterativo, compuesto de dos fases: primero la construcción de una solución y después un proceso de mejora. La solución mejorada es el resultado del proceso de búsqueda [23].

El mecanismo de construcción de soluciones es un heurístico constructivo aleatorio. Va añadiendo paso a paso diferentes componentes c a la solución parcial s_p , que inicialmente está vacía. Los componentes que se añaden en cada paso son elegidos aleatoriamente de una lista restringida de candidatos (RCL). Esta lista es un subconjunto de $N(s_p)$, el conjunto de componentes permitidos para la solución parcial s_p . Para generar esta lista, los componentes de la solución en $N(s_p)$ se ordenan de acuerdo a alguna función dependiente del problema (η) [23].

La lista RCL está compuesta por los α mejores componentes de ese conjunto. En el caso extremo de $\alpha = I$, siempre se añade el mejor componente encontrado de manera determinista, con lo que el método de construcción es equivalente a un algoritmo voraz. En el otro extremo, con $\alpha = |N(s_p)|$ el componente a añadir se elige de forma totalmente aleatoria de entre todos los disponibles. Por lo tanto, α es un parámetro clave que incluye en cómo se va a muestrear el espacio de búsqueda. La segunda fase del algoritmo consiste en aplicar un método de búsqueda local para mejorar la solución generada. Este mecanismo de mejora puede ser una técnica de mejora simple o algoritmos más complejos como SA o TS [23].

2.1.3.1.4. Búsqueda con Vecindario Variable (VNS)

La búsqueda con vecindario variable o Variable Neighborhood Search (VNS) es una metaheurística propuesta en [29] que aplica explícitamente una estrategia para cambiar entre diferentes vecindarios durante la búsqueda. Este algoritmo es muy general y con muchos grados de libertad a la hora de diseñar variaciones e instanciaciones particulares [23].

El primer paso a realizar es definir un conjunto de vecindarios. Esta elección puede hacerse de muchas formas: desde ser elegidos aleatoriamente hasta utilizar complejas ecuaciones deducidas del problema. Cada iteración consiste en tres fases: la elección del candidato, una fase de mejora y, finalmente, el movimiento. En la primera fase, se elige aleatoriamente un vecino s' de s usando el k -ésimo vecindario. Esta solución s' es utilizada como punto de partida de la búsqueda local de la segunda fase [23].

Cuando termina el proceso de mejora, se compara la nueva solución s'' con la original s . Si es mejor, s'' se convierte en la solución actual y se inicializa el contador de vecindarios ($k \leftarrow 1$); si no es mejor, se repite el proceso, pero utilizando el siguiente vecindario ($k \leftarrow k + 1$). La búsqueda local es el paso de intensificación del método y el cambio de vecindario puede considerarse como el paso de diversificación [23].

2.1.3.1.5. Búsqueda Local Iterada (ILS)

La búsqueda local iterada o Iterated Local Search (ILS) [30] [31] es una metaheurística basada en un concepto simple pero muy efectivo. En cada iteración, la solución actual es perturbada y, a esta nueva solución, se le aplica un método de búsqueda local para mejorarla.

El mínimo local obtenido por el método de mejora puede ser aceptado como nueva solución actual si pasa un test de aceptación [23].

La importancia del proceso de perturbación es obvia: si es demasiado pequeña puede que el algoritmo no sea capaz de escapar del mínimo local; por otro lado, si es demasiado grande, la perturbación puede hacer que el algoritmo sea como un método de búsqueda local con un reinicio aleatorio. Por lo tanto, el método de perturbación debe generar una nueva solución que sirva como inicio a la búsqueda local, pero que no debe estar muy lejos del actual para que no sea una solución aleatoria. El criterio de aceptación actúa como contra-balance, ya que filtra la aceptación de nuevas soluciones dependiendo de la historia de búsqueda y de las características del nuevo mínimo local [23].

2.1.3.2. Metaheurísticas basadas en población

Se caracterizan por la mejora continua de la solución actual mediante la inspección de un vecindario, mientras que las metaheurísticas basadas en población trabajan con un conjunto de soluciones que iterativamente son mejoradas a través de un proceso inteligente de exploración del espacio de búsqueda. [32]

2.1.3.2.1. Algoritmos Evolutivos (EA)

Los algoritmos evolutivos o Evolutionary Algorithms (EAs) están inspirados en la teoría de la evolución natural. Esta familia de técnicas sigue un proceso iterativo y estocástico que opera sobre una población de soluciones, denominadas en este contexto individuos. Inicialmente, la población es generada típicamente de forma aleatoria (quizás con ayuda de un heurístico de construcción) [23].

El esquema general de un algoritmo evolutivo comprende tres fases principales: selección, re-producción y reemplazo. El proceso completo es repetido hasta que se cumpla un cierto criterio de terminación (normalmente después de un número dado de iteraciones). En la fase de selección se escogen generalmente los individuos más aptos de la población actual para ser posteriormente recombinados en la fase de reproducción. Los individuos resultantes de la recombinación se alteran mediante un operador de mutación [23].

Finalmente, a partir de la población actual y/o los mejores individuos generados (de acuerdo a su valor de fitness) se forma la nueva población, dando paso a la siguiente generación del algoritmo [23].

2.1.3.2.2. Algoritmos de Estimación de la Distribución (EDA)

Los algoritmos de estimación de la distribución o Estimation of Distribution Algorithms (EDAs) [33] muestran un comportamiento similar a los algoritmos evolutivos presentados en la sección anterior y, de hecho, muchos autores consideran los EDAs como otro tipo de EA. Los EDAs operan sobre una población de soluciones tentativas como los algoritmos evolutivos, pero, a diferencia de estos últimos, que utilizan operadores de recombinación y mutación para mejorar las soluciones, los EDAs in eren la distribución de probabilidad del conjunto seleccionado y, a partir de ésta, generan nuevas soluciones que formarán parte de la población. [34]

Los modelos gráficos probabilísticos son herramientas comúnmente usadas en el contexto de los EDAs para representar eficientemente la distribución de probabilidad [23]. Algunos autores [35] [36] [37] han propuesto las redes bayesianas para representar la distribución de probabilidad en dominios discretos, mientras que las redes gaussianas se emplean usualmente en los dominios continuos [38] [23].

2.1.3.2.3. Búsqueda Dispersa (SS)

La búsqueda dispersa o Scatter Search (SS) [39] es una metaheurística cuyos principios fueron presentados en [27] [23] y que actualmente está recibiendo una gran atención por parte de la comunidad científica [40]. El algoritmo se basa en mantener un conjunto relativamente pequeño de soluciones tentativas (llamado conjunto de referencia o *RefSet*) que se caracteriza por contener soluciones de calidad y diversas (distantes en el espacio de búsqueda) [23].

Para la definición completa de SS hay que concretar cinco componentes: creación de la población inicial, generación del conjunto de referencia, generación de subconjuntos de soluciones, método de combinación de soluciones y método de mejora [23].

2.1.3.2.4. Algoritmos Genéticos (AG)

Un algoritmo genético (AG) es una técnica de búsqueda iterativa inspirada en los principios de *selección natural*. Los AG no buscan modelar la evolución biológica sino derivar estrategias de optimización. El concepto se basa en la generación de poblaciones de individuos mediante la reproducción de los padres. Durante el curso de la evolución, los genes con evolución lenta fueron reemplazados por genes con mejor estrategia evolutiva. Por lo tanto, se esperan estrategias altamente eficientes en la fauna y la flora modernas. Muchos problemas tienen funciones objetivo complejas y la optimización tiende a finalizar en mínimos/máximos locales. La idea de los AG es optimizar (hallar el máximo o mínimo) una función objetivo utilizando los principios de la selección natural sobre los parámetros de la función. [18]

2.1.3.2.4.1. Selección Natural

Mientras reflexionaba acerca de la selección artificial, Darwin leyó un ensayo del economista Thomas Malthus. El ensayo sugería que, de no controlarse la población humana, sobrepasaría con el tiempo a la producción de alimentos, ocasionando una fuerte competencia por la existencia. Darwin se dio cuenta de que las ideas de Malthus podían aplicarse al mundo natural. Darwin llamó a su teoría selección natural. Razonó que, dado el tiempo necesario, la selección natural podría modificar una población lo suficiente como para producir nuevas especies. [18]

Para las estructuras básicas de un algoritmo genético también es necesario conocer la transición de una generación a otra, la cual consta de cuatro elementos básicos:

- **Selección:** mecanismo de selección individual (cadena) para la reproducción acorde con la función de aptitud (valor de la función objetivo). Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no [18].

La idea básica de selección está asociada con la función de aptitud y el sistema original; para su implementación es comúnmente conocida como roulette-wheel (RWS); ésta utiliza una distribución de probabilidad, donde la probabilidad de selección de una cadena es directamente proporcional a su aptitud [18].

- **Cruzamiento:** método de fusión sobre la información genética de dos individuos; si la codificación es elegida apropiadamente, dos progenitores saludables producirán descendientes sanos. Es el principal operador genético; provee un mecanismo para heredar características a su descendencia; interviene en ambos progenitores [18].
- **Mutación:** en la evolución real, el material genético puede ser alterado en forma aleatoria debido a un error en la reproducción o la deformación de genes; un ejemplo es la radiación de los rayos gama. En los algoritmos genéticos, la mutación se realiza, con gran probabilidad, como una deformación aleatoria de las cadenas. Produce cambios incrementales al azar en la descendencia, efectuando cambios aleatorios en los valores del alelo en algunos genes. En el caso de cromosomas binarios le corresponde hacer los cambios de posiciones en cada bit. No afecta a toda la población, pero es probable que dañe a algunos. La mutación tiene el efecto de perturbar de manera segura a los cromosomas a fin de introducir nuevas características que no estaban presentes en ningún elemento de los progenitores [18].
- **Reemplazo:** procedimiento para calcular (crear) una nueva generación de la anterior y sus descendientes. Se crea un espacio a la descendencia en la población eliminando de ella a los padres [18].

2.1.3.2.5. Optimización basada en Colonias de Hormigas (ACO)

Los algoritmos de optimización basados en colonias de hormigas o Ant Colony Optimization (ACO) [41] [42] [23] están inspirados en el comportamiento de las hormigas reales cuando buscan comida. Este comportamiento es el siguiente: inicialmente, las hormigas exploran el área cercana a su nido de forma aleatoria. Tan pronto como una hormiga encuentra comida, la lleva al nido. Mientras que realiza este camino, la hormiga va depositando una sustancia química de-nominada feromona. Esta sustancia ayudará al resto de las hormigas a encontrar la comida [23].

La comunicación indirecta entre las hormigas mediante el rastro de feromona las capacita para encontrar el camino más corto entre el nido y la comida. Este comportamiento es el que intenta simular este método para resolver problemas de optimización. La técnica se basa en dos pasos principales: construcción de una solución basada en el comportamiento de una hormiga y actualización de los rastros de feromona artificiales [23].

El algoritmo no fija ninguna planificación o sincronización a priori entre las fases, pudiendo ser incluso realizadas simultáneamente [23].

2.1.3.2.6. Optimización basada en Cúmulos de Partículas (PSO)

Los algoritmos de optimización basados en cúmulos de partículas o Particle Swarm Optimization (PSO) están inspirados en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. El algoritmo PSO mantiene un conjunto de soluciones, también llamadas partículas, que son inicializadas aleatoriamente en el espacio de búsqueda. Cada partícula posee una posición y velocidad que cambia conforme avanza la búsqueda. En el movimiento de una partícula incluye su velocidad y las posiciones donde la propia partícula y las demás de su vecindario encontraron buenas soluciones. En el contexto de PSO, el *vecindario de una partícula* se define como un conjunto de partículas del cúmulo. El vecindario de una partícula puede ser global, en el cual todas las partículas del cúmulo se consideran vecinas, o local, en el que sólo las partículas más cercanas se consideran vecinas [23].

2.1.3.2.7. Optimización basada en Malla Variables (VMO)

VMO es un algoritmo desarrollado por A. Puris, es una metaheurística basada en población en el que esta se distribuye como una malla. Esta malla se compone de nodos P (n_1, n_2, \dots, n_p) que representan soluciones en el espacio de búsqueda [15].

Cada nodo se codifica como un vector de números de punto flotante M , $(v_1^i, v_2^i, \dots, v_M^i) = v_j^i, j = 1, \dots, M$ que representan la solución para el problema de optimización. En el proceso de búsqueda desarrollado por VMO, dos operaciones se ejecutan: los procesos de expansión y contracción [15].

El proceso de expansión genera nuevos nodos utilizando la población en la iteración actual, por medio de los pasos siguientes:

- Fase 1 (Generación de la población inicial): Se generan P soluciones factibles de manera aleatoria que representan los nodos de la malla inicial en el proceso de búsqueda del algoritmo [15].

- Fase 2 (Generación de nodos hacia los mejores vecinos): Para cada solución de la malla (n_i) se buscan los k nodos más cercanos. A continuación, se selecciona el vecino de mejor calidad (fitness) (n^*), y si n^* es mejor que n_i se genera un nuevo nodo entre ambos; caso contrario, no se genera una nueva solución [15].
- Fase 3 (Generación de nodos hacia la mejor solución): Se crea un nuevo nodo para cada n_i en dirección al nodo con mejor valor de fitness (n_g , óptimo global) de la malla actual [15].
- Fase 4 (Generación de nodos utilizando los nodos fronteros de la malla): Los nodos fronteras se definen como las soluciones que se encuentran más cerca y más lejos del centro de la población. En esta fase las nuevas soluciones se generan desplazando cada nodo frontera en función de un valor de desplazamiento w [15].
- Fase de Clearing: En esta fase, se seleccionan las soluciones más representativas entre la malla actual y los nuevos nodos generados en el proceso de expansión (Fase 2, 3, 4). Para ello, se evita seleccionar soluciones demasiado cercanas entre sí. Durante el clearing se define una distancia ξ como distancia mínima entre las soluciones, y se tiene que ignorar todo nodo que tenga una distancia inferior a un nodo con mejor o igual calidad. Dicha distancia se inicializa a un 25% del rango de búsqueda, y va reduciéndose durante la ejecución del algoritmo. [15]

2.1.4. Metaheurísticas para optimización multiobjetivo

La mayoría de los problemas de optimización del mundo real son de naturaleza multiobjetivo, lo que supone que hay que minimizar o maximizar varias funciones a la vez que están normalmente en conflicto entre sí (problemas multiobjetivo o MOPs, Multiobjective Optimization Problems). Debido a la falta de soluciones metodológicas adecuadas, los problemas multiobjetivo se han resuelto en el pasado como problemas mono-objetivo. Sin embargo, existen diferencias fundamentales en los principios de funcionamiento de los algoritmos para optimización mono y multiobjetivo [23].

Así, las técnicas utilizadas para resolver MOPs no se restringen normalmente a encontrar una solución única, sino un conjunto de soluciones de compromiso entre los múltiples objetivos contrapuestos, ya que no suele existir una solución que optimice simultáneamente todos los objetivos [23].

Se pueden distinguir, por tanto, dos etapas cuando se aborda este tipo de problemas: por un lado, la optimización de varias funciones objetivo involucradas y, por otro, el proceso de toma de decisiones sobre qué solución de compromiso es la más adecuada [14]. Atendiendo a cómo manejan estas dos etapas, las técnicas para resolver MOPs se pueden clasificar en [43] [23].

- A priori: las decisiones se toman antes de buscar soluciones [23].
- Progresivas: se integran la búsqueda de soluciones y la toma de decisiones [23].
- A posteriori: se busca antes de tomar decisiones [23].

Cada una de ellas tiene ciertas ventajas e inconvenientes que las hacen más adecuadas para determinados escenarios concretos [14] [44]. No obstante, en las dos primeras clases la búsqueda está muy influenciada por la decisión de un experto (decision maker) que determina la importancia de un objetivo sobre otro y que puede limitar arbitrariamente el espacio de búsqueda, impidiendo una resolución óptima del problema. En las técnicas *a posteriori*, por el contrario, se realiza una exploración lo más amplia posible para generar tantas soluciones de compromiso como sea posible. Es, entonces, cuando tiene lugar el proceso de toma de decisiones por parte del experto [23].

Precisamente por la aproximación que siguen, estas técnicas *a posteriori* están siendo muy utilizadas dentro del campo de las metaheurísticas y, particularmente, en el campo de la computación evolutiva [14] [44]. Más concretamente, los algoritmos más avanzados aplican técnicas basadas en el concepto de Optimalidad de Pareto [45]. Un MOP se puede definir como el problema de encontrar un vector de variables de decisión que satisface un conjunto de restricciones y que optimiza un conjunto de funciones objetivo. Estas funciones forman una descripción matemática de criterios de rendimiento que están normalmente en conflicto entre sí. Por tanto, el término “optimización” se refiere a la búsqueda de una solución tal que contenga valores aceptables para todas las funciones objetivo [46] [23].

2.1.5. Óptimo de Pareto

Un punto $x^{\rightarrow*} \in \Omega$ es un óptimo de Pareto si para cada $x^{\rightarrow} \in \Omega$ y $I = \{1, 2, \dots, k\}$, o bien $\forall_{i \in I} (f_i(x^{\rightarrow}) = f_i(x^{\rightarrow*}))$ o bien hay al menos un $i \in I \mid (f_i(x^{\rightarrow}) > f_i(x^{\rightarrow*}))$ [23].

Esta definición dice que $x^{\rightarrow*}$ es un óptimo de Pareto si no existe ningún vector factible x^{\rightarrow} que mejore algún criterio sin causar simultáneamente un empeoramiento en al menos otro criterio (asumiendo minimización). El concepto de Optimalidad de Pareto es integral tanto a la teoría como a la resolución de MOPs [47] [23].

2.1.6. Dominancia de Pareto

Un vector $\vec{u} = (u_1, \dots, u_k)$ se dice que domina a otro vector $\vec{v} = (v_1, \dots, v_k)$ (representado por $\vec{u} < \vec{v}$) si y sólo si \vec{u} es parcialmente menor que \vec{v} , es decir, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$ [23].

La relación de dominancia es la clave en este tipo de técnicas basadas en Optimalidad de Pareto, ya que nos va a permitir establecer una ordenación de las soluciones. De hecho, esta relación es una relación de orden parcial estricto, puesto que no es reflexiva, ni simétrica, ni anti simétrica, pero sí transitiva. Así, se han propuesto diferentes métodos en la literatura [14] [44] que, básicamente, transforman el vector de fitness en un valor único utilizando esta relación. Esta estrategia fue originalmente propuesta por Goldberg en [48] para guiar la población de un GA hacia el frente de Pareto de un MOP [23].

La idea básica consiste en encontrar las soluciones de la población que no están dominadas por ninguna otra. A estas soluciones se le asigna el mayor orden (las mejores en la ordenación establecida por la relación de dominancia). A continuación, se consideran las soluciones no-dominadas que quedan si se eliminan todas las anteriores, a las que se asigna el siguiente rango. El proceso continúa hasta que se le asigna un rango a todas las soluciones [23].

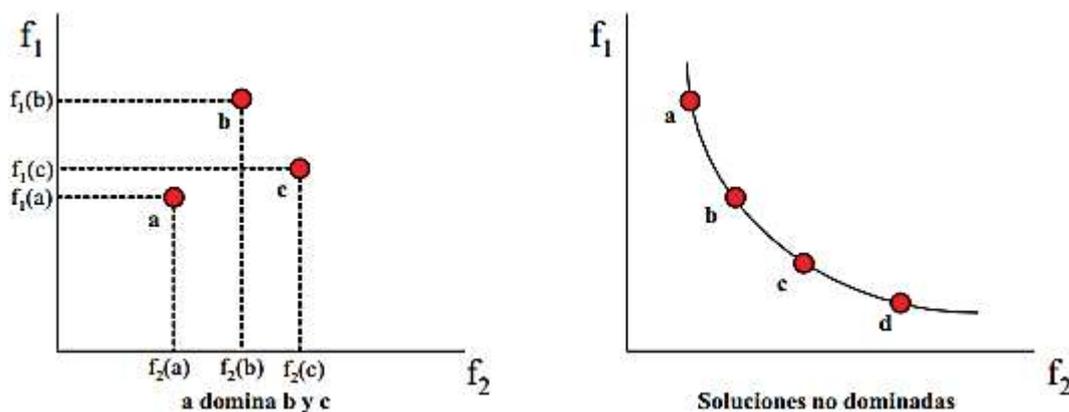


Ilustración 2.- Ejemplo del concepto de Dominancia de Pareto.

2.1.7. Frente de Pareto

Para un MOP dado $\vec{f}(\vec{x})$ y su conjunto óptimo de Pareto P^* , el frente de Pareto se define como $PF^* = \{\vec{f}(\vec{x}), \vec{x} \in P^*\}$ [23].

Es decir, el frente de Pareto está compuesto por los valores en el espacio de objetivos del conjunto óptimo de Pareto. En general, no es fácil encontrar una expresión analítica de la línea o superficie que contiene estos puntos. De hecho, en la mayoría de los casos es imposible [14] [23].

2.1.8. Objetivos en la resolución de MOPs

Dado un MOP, por tanto, se busca idealmente un número de soluciones no-dominadas que persiguen dos metas:

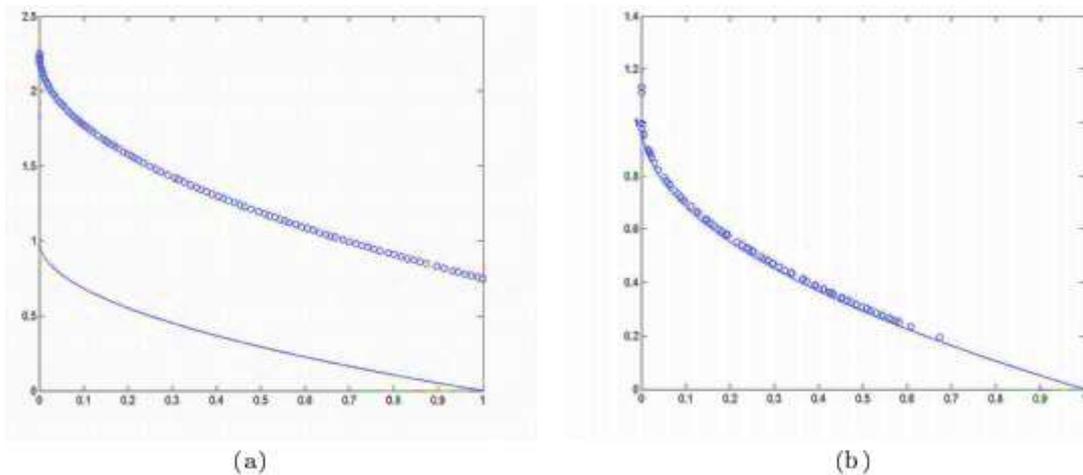
1. Encontrar un conjunto de soluciones lo más cercano posible al frente de Pareto óptimo [23].
2. Encontrar un conjunto de soluciones tan uniformemente diverso como sea posible [23].

Mientras que la primera meta, converger hacia la solución óptima, es obligatoria en toda tarea de optimización mono o multiobjetivo, la segunda es completamente específica para optimización multiobjetivo. Además de converger hacia el frente óptimo, las soluciones deben estar uniformemente repartidas a lo largo de todo el frente [23].

Sólo con un conjunto diverso de soluciones se puede asegurar, por una parte, un buen conjunto de soluciones de compromiso entre los diferentes objetivos para la posterior toma de decisiones por parte del experto y, por otra, que se ha realizado una buena exploración del espacio de búsqueda. La *Ilustración 3* muestra dos ejemplos de frentes que fallan, cada uno, en una de las metas anteriores [23].

En la parte (a) podemos ver una aproximación al frente en el que las soluciones no-dominadas se distribuyen perfectamente. No obstante, se trata de un MOP diseñado de forma que contiene múltiples frentes engañosos y, en realidad, las soluciones obtenidas no son Pareto-óptimas, aunque su diversidad es excelente [23].

Por el contrario, en la parte (b) de la misma figura, tenemos un conjunto de soluciones que han convergido hacia el frente de Pareto óptimo, pero, sin embargo, deja regiones de éste sin cubrir. Aunque ninguno de los dos casos es deseable, la primera situación es claramente peor: ninguna de las soluciones obtenidas es Pareto-óptima [23].



2.1.9. Evaluación estadística de resultados

Existen algunos trabajos que abordan el análisis teórico para un gran número de heurísticas y problemas [49]; [50], pero dada la dificultad que entraña este tipo de análisis teórico, tradicionalmente se analiza el comportamiento de los algoritmos mediante comparaciones empíricas. Para ello es necesario definir indicadores que permitan estas comparaciones. Podemos encontrar, en general, dos tipos diferentes de indicadores. Por un lado, tenemos aquellos que miden la calidad de las soluciones obtenidas. Por otro lado, están los indicadores que miden el rendimiento de los algoritmos y que hacen referencia a los tiempos de ejecución o a la cantidad de recursos computacionales utilizados [23].

2.1.9.1. Indicadores de calidad.

Estos indicadores son los más importantes a la hora de evaluar una metaheurística. Son distintos dependiendo de si se conoce o no la solución óptima del problema en cuestión (algo común para problema clásicos de la literatura, pero poco usual en problemas del mundo real) [23].

2.1.9.1.1. Indicadores para optimización multiobjetivo

En el campo multiobjetivo esto es un tema de investigación muy activo [51]; [52], ya que el resultado de estos algoritmos es un conjunto de soluciones no-dominadas y no una solución única. Hay que definir, por tanto, indicadores de calidad para aproximaciones al frente de Pareto. Hay normalmente dos aspectos a considerar para medir la calidad de un frente: convergencia y diversidad. La primera hace referencia a la distancia existente entre la aproximación y el frente de Pareto óptimo del problema, mientras que la segunda mide la uniformidad de la distribución de soluciones sobre el frente [23].

2.1.9.1.1.1. Hipervolumen (I_{HV})

El Hipervolumen (I_{HV}), diseñada para medir los aspectos de convergencia y diversidad en un frente de Pareto. Esta métrica calcula el volumen (en el espacio de objetivos) cubierto por miembros de un conjunto Q , de soluciones no-dominadas para problemas donde todos los objetivos han de ser minimizados. Los algoritmos que alcanzan mayores valores para I_{HV} son los mejores [53] [23].

2.1.9.1.1.2. Épsilon (I_{E+})

Mide la convergencia al determinar la distancia mínima (en cualquier objetivo) que habría que desplazar cada solución para ser no-dominada con respecto a otro frente (cuanto más pequeño mejor) [53] [23].

2.1.9.1.1.3. Spread (I_{Δ})

Dispersión (Spread (I_{Δ})) el cual mide la distribución de las soluciones y la distancia con los extremos del verdadero frente de Pareto, es decir esta medida toma el valor cero para una distribución ideal, cuando hay una dispersión perfecta de las soluciones del frente de Pareto [53] [23].

2.1.9.1.1.4. Distancia Generacional Invertida Plus (IGD+)

La idea básica en el IGD+ es calcular la distancia de cada punto de referencia a la región dominada por un conjunto de soluciones. Esta idea puede ser vista como la incorporación de la dominación de Pareto entre un punto de referencia y una solución en su cálculo de distancia. [54]

2.1.10. Análisis estadístico de resultados

2.1.10.1. Wilcoxon

El test de suma de rangos, es un test no paramétrico que permite hacer comparaciones dos a dos entre los resultados de pares de algoritmos para analizar la significancia estadística de los mismos. En todos los trabajos se ha aplicado un nivel de confianza del 95 % (un nivel de significancia del 5 % o p-value menor de 0.05), lo que implica que la probabilidad de que las diferencias observadas no se deban al azar es del 95 %. [34]

2.1.10.2. Friedman

La prueba de Friedman es una prueba no paramétrica para probar la diferencia entre varias muestras relacionadas. La prueba de Friedman es una alternativa para el análisis de medidas repetidas de las diferencias que se utiliza cuando el mismo parámetro se ha medido en condiciones diferentes sobre los mismos temas. La hipótesis nula para el test de Friedman es que no hay diferencias entre las variables. Si la probabilidad calculada es baja (P menor que el nivel de significación seleccionado) se rechaza la hipótesis nula y se puede concluir que al menos 2 de las variables son significativamente diferentes unos de otros. [55]

2.1.11. Filogenética

La sistemática filogenética, o simplemente "filogenética", no es simplemente otro enfoque de la sistemática. Es un acercamiento a la sistemática diseñada para estimar el patrón de descendencia filogenética que se necesita para deducir los procesos de evolución relacionados con el origen de las especies. Las clasificaciones que resultan del análisis filogenético son herramientas críticas para los estudios evolutivos. [56]

Sin embargo, la filogenética es también más que la ayuda de la evolución, ya que su filosofía subyacente proporciona una forma de ver la naturaleza, hacer preguntas y resolver problemas asociados con la evolución de los organismos. La sistemática filogenética es un enfoque de la sistemática que realiza un ordenamiento de la diversidad orgánica de tal manera que nuestras ideas sobre las relaciones evolutivas inferidas entre los organismos pueden ser científicamente discutidas y evaluadas. [56]

Las clasificaciones filogenéticas son lógicamente consistentes con el árbol filogenético propugnado por el investigador. Por lo tanto, son candidatos para ser clasificaciones naturales superiores a las alternativas que no son lógicamente coherente con la hipótesis del árbol filogenético. [12]

2.1.12. Árbol Filogenético

Una filogenia es un árbol que contiene nodos que están conectados por ramas. Cada rama representa la persistencia de un linaje genético a través del tiempo, y cada nodo representa el nacimiento de un nuevo linaje. Si el árbol representa la relación entre un grupo de especies, entonces los nodos representan eventos de especiación. En otros contextos, la interpretación podría ser diferente. [57]

Los árboles filogenéticos no se observan directamente y se deducen de la secuencia u otros datos. Los métodos de reconstrucción de la filogenia están basados en la distancia o en los caracteres. En los métodos de matriz de distancia, la distancia entre cada par de secuencias se calcula, y la matriz de distancia resultante se utiliza para la reconstrucción del árbol. Los métodos basados en caracteres incluyen métodos de máxima parsimonia, máxima verosimilitud e inferencia bayesiana [57]

2.1.13. Inferencia Filogenética

La inferencia filogenética se define como un proceso de análisis que busca describir las relaciones ancestro-descendiente que se han dado entre una serie de organismos a lo largo de la historia evolutiva. Toma como entrada n secuencias de caracteres de longitud N , pertenecientes a un determinado alfabeto α [58].

En el caso de secuencias de ADN, el alfabeto está constituido por los valores $\alpha = \{A, C, G, T\}$ que representan, respectivamente, las bases adenina, citosina, guanina y timina. Es decir, las entradas son secuencias moleculares características de cada uno de los organismos a estudiar. La salida del proceso de inferencia recibe el nombre de árbol filogenético [58].

Un árbol filogenético es una estructura matemática que representa una hipótesis de cómo se ha producido la evolución de las especies de entrada, definiendo posibles ancestros a diferentes niveles de relaciones históricas. Esta estructura presenta una topología compuesta por:

- Hojas o nodos terminales. Son el resultado de la evolución, esto es, las especies de entrada [58].
- Nodos intermedios. Representan posibles ancestros de las especies consideradas [58].
- Ramas. Indican una relación ancestro-descendiente entre dos nodos del árbol. Las ramas pueden tener asociado un valor de longitud de rama, que generalmente indica periodos de tiempo evolutivo o bien el número de cambios a nivel molecular que se han producido al evolucionar a la especie descendiente [58].

Cuando la topología de un árbol presenta un nodo base a partir del cual se definen las relaciones de parentesco, se dice que este árbol es enraizado. Por su parte, si este nodo raíz no existe y no se muestra una direccionalidad clara en la hipótesis evolutiva, hablamos de árbol no enraizado [58].

El principal problema que presenta la mayoría de los métodos de reconstrucción filogenética es la exploración del espacio de búsqueda en busca del mejor árbol según el criterio adoptado [58]. Para el caso de árboles no enraizados, el número de topologías para un número n de especies viene dado por:

$$\frac{(2n - 5)!}{(n - 3)! 2^{n-3}} \quad (1)$$

Esta expresión implica que, para un número de especies mayor o igual que 10, sea inviable aplicar búsquedas exhaustivas. Esta es la razón por la que los métodos de reconstrucción basados en criterios de Optimalidad optan por usar técnicas basadas en heurísticas para realizar el proceso de inferencia en tiempos razonables. [58]

2.1.13.1. Análisis por máxima parsimonia

Los métodos de máxima parsimonia tienen por objeto encontrar el árbol que minimice el número de cambios que se precisan para explicar los datos [58].

Se prefiere aquel árbol cuya topología implique una menor cantidad de transformaciones a nivel molecular, constituyendo la descripción más simple de la historia evolutiva de los organismos de entrada. El valor de parsimonia para un árbol τ , construido a partir de un conjunto de n especies caracterizadas por una secuencia de N caracteres alineados, viene dado por la siguiente expresión [58]:

$$P(\tau) = \sum_{i=1}^N \sum_{(a,b) \in B(\tau)} C(a_i, b_i) \quad (2)$$

Donde (a, b) representa una rama que une a los nodos a y b en la topología del árbol τ , perteneciente al conjunto B de ramas de τ , siendo a_i y b_i los estados del carácter i para a y b respectivamente, y $C(a_i, b_i)$ el coste de evolucionar desde el estado a_i al b_i . En una aproximación por máxima parsimonia, se prefieren aquellos árboles que minimicen este valor, pues implica una serie de cambios evolutivos más simples [58].

Existen varios métodos para computar la parsimonia de un árbol en la literatura, siendo uno de los más usados el conocido como Algoritmo de Fitch. Para tratar la parsimonia de un árbol filogenético se puede hacer uso de distintos operadores de cambio topológico [58]. Los más empleados son los siguientes:

- Nearest Neighbour Interchange (NNI). Consiste en tomar una rama interna del árbol e intercambiar los nodos situados en los subárboles presentes en los extremos de dicha rama [58].
- Subtree Pruning and Regrafting (SPR). Separa un subárbol de la topología inicial y lo reinserta en todas las posiciones posibles, evaluando cada resultado y almacenando aquel que mejore el criterio que se haya considerado [58].
- Tree Bisection and Reconnection (TBR). Selecciona una rama del árbol y separa los subárboles. A continuación, crea nuevas topologías mediante la reconexión de los arboles usando las ramas internas de cada uno de ellos. Evalúa todos los subárboles y reconexiones posibles y almacena la mejor topología encontrada. [58]

2.1.13.2. Análisis por máxima verosimilitud

La verosimilitud es una función estadística que, aplicada a filogenética, indica la probabilidad de que la hipótesis evolutiva que plantea una topología de árbol filogenético diera lugar al conjunto de organismos observados en la entrada del procedimiento. El análisis por máxima verosimilitud busca encontrar aquel árbol que represente la evolución más probable de las especies de entrada [58].

A la hora de calcular la verosimilitud, es necesario considerar los siguientes parámetros:

1. La topología del árbol en cuestión.
2. La longitud de las ramas.
3. El modelo de evolución molecular.

Un modelo de evolución describe las probabilidades con que se puede producir un cambio que implique la sustitución de aminoácidos o nucleótidos entre generaciones, esto es, la probabilidad de que se dé la modificación de un carácter en las secuencias de los nodos del árbol. Existen multitud de modelos (JC69, F84, HKY85...) y de su elección dependerá en gran medida el valor de verosimilitud obtenido. Sea τ el árbol filogenético a evaluar, D el conjunto de secuencias de N caracteres de entrada, D_i los datos para el carácter i -ésimo de las secuencias alineadas y m el modelo de evolución elegido [58]. La verosimilitud viene dada por:

$$L[D, \tau, m] = Pr[D|r, m] = \prod_{i=1}^N \prod_{j=1}^B (r_j t_j)^{n_{ij}} \quad (3)$$

siendo B el total de ramas del árbol, r_i la probabilidad de cambio del carácter i , t_j la longitud de la rama j y n_{ij} el número de cambios de estado del carácter i en la rama j . La verosimilitud es un objetivo a maximizar, por lo que se prefieren aquellos árboles que obtengan mayores puntuaciones. [58]

2.1.14. Framework de optimización multiobjetivo (jMetalCpp)

El framework es la versión C++ del framework de optimización multiobjetivo jMetal, para la Optimización Mono y Multiobjetivo con metaheurísticas de acuerdo a las necesidades de la comunidad de investigación. Entre las metaheurísticas multiobjetivo cuenta con: NSGA-II, ssNSGA-II, GDE3, SMPSO, SMPSO_{hv}, OMOPSO, PAES, SMS-emoa, MOEA / D y MOCHC. [59]

2.1.15. Framework bioinformático (BIO++)

Bio++ es un framework bioinformático que provee un conjunto de librerías para el desarrollo de software para varias áreas de la Bioinformática: análisis de secuencias biológicas, inferencia filogenética, evolución molecular y genética poblacional. Entre sus librerías podemos citar a *Seq-Lib* (Sequence Library), la cual incluye funciones para manipular y analizar secuencias biológicas de ADN, ARN, proteínas y secuencias de codones. Su librería filogenética *PhylLib* (Phylogenetics Library) proporciona varias funcionalidades para la lectura, almacenamiento y manipulación de árboles filogenéticos, además de funciones para inferir filogenias bajos los métodos basados en caracteres como la máxima parsimonia, máxima verosimilitud y métodos basados en distancias como *Neighbour-joining*. [60]

2.1.16. Framework Filogenético (PLL)

La librería filogenética (PLL) es una librería altamente optimizada que provee funcionalidades para la estimación computacional de la máxima verosimilitud sobre árboles filogenéticos (PLF Phylogenetic Likelihood Function) proporcionando algunas características que apuntan a reducir los altos costos computacionales requeridos por esta función. Además, provee métodos para la optimización de las longitudes de ramas de los árboles filogenéticos y una función adaptada a un esquema de vectorización que permite una rápida y eficiente exploración del espacio de búsqueda de árboles filogenéticos mediante modificaciones topológicas y mejoramiento de las longitudes de las ramas afectas por los cambios, lo que permite inferir rápidamente filogenias bajo el criterio de la máxima verosimilitud. [61]

2.1.17. Repositorio TreeBASE

TreeBASE es una base de datos relacional diseñada para administrar y explorar información en relaciones filogenéticas. Incluye matrices de datos y árboles filogenéticos, junto con información sobre la publicación pertinente, taxonomía, caracteres morfológicos y basada en la secuencia y análisis publicados [62].

La base de datos está diseñada para permitir la recuperación de datos de diferentes estudios y árboles para que los resultados filogenéticos pueden reevaluarse, en comparación, combinados y reutilizados [62]. Las aplicaciones incluyen:

- Para buscar información sobre la filogenia de grupos particulares de interés. [62]
- Para obtener conjuntos de datos para estudios de evolución del carácter, incluyendo patrones generales a través de muchos grupos, tales como patrones de homoplasia.
- En estudios de biogeografía--recuperar árboles con representantes en particular áreas geográficas. [62]
- En estudios de coevolución--para recuperar información en filogenias anfitrión y parásito. [62]
- En estudios de congruencia y combinación de datos para recuperar todos filogenias moleculares y morfológicos para grupos particulares. [62]
- En estudios de métodos filogenéticos para recuperar toda parsimonia o reconstrucciones de máxima verosimilitud, por ejemplo, o para descargar conjuntos de datos de varios tipos de métodos de prueba. [62]
- Para enlazar juntos árboles de determinados grupos con filogenias más inclusivos.
- Para descubrir grupos escasamente - un recurso para los estudiantes, organismos de financiación, etcétera. [62]
- Para recuperar la información filogenética para el uso en biología de la conservación y la gestión de los recursos naturales. [62]

TreeBASE es un repositorio de información filogenética, árboles filogenéticos especialmente enviados por el usuario y los datos utilizados para generarlos. TreeBASE acepta todo tipo de datos filogenéticos (p. ej., árboles de especies, árboles de las poblaciones, de los genes) que representa a todos los taxa bióticos. [62]

2.1.18. Optimización Multiobjetivo Basada en Mallas Variables (MOVMO)

Una extensión VMO [15], denominado optimización multiobjetivos basada en mallas Variables (MOVMO). El nuevo algoritmo sigue en gran medida la estrategia de búsqueda VMO, incluyendo el reemplazo de población dinámica [16].

Porque MOVMO tiene como objetivo abordar los problemas de optimización multiobjetivo, varios fundamentos algorítmicos tradicionales en este campo se incorporan a su diseño, es decir, dominación de Pareto, estimación de densidad de espacio objetivo (mediante apretadura distancia) y elitismo en forma de un archivo externo que almacena el conjunto de soluciones descubiertas no-dominadas. [16]

Los cambios propuestos al clásico VMO ofrecido por MOVMO están dirigidos a abordar efectivamente a problemas de optimización multiobjetivo. Por ejemplo, MOVMO mantiene un archivo líder con un conjunto de soluciones bien distribuidas no-dominadas (global leaders) en vez del mejor global grabada por VMO. Otro ejemplo es la sustitución de la generación de malla de nodos T aumentada en VMO con un mecanismo de recambio nodo dinámico. La justificación para esta decisión fue motivada por el gran esfuerzo computacional necesario para fusionar y clasificar dos conjuntos (la malla actual de tamaño P y la malla expandida de nodo T) en un entorno multiobjetivo.

Por otra parte, la medida de distancia crowding aplicada a las soluciones no-dominadas en L desempeña el papel del operador de compensación adaptativo de VMO utilizado en la descomposición de las porciones exploradas en gran medida del espacio de decisión. [16]

2.1.19. Algoritmo Evolutivo Multiobjetivo basado en la Descomposición (MOEA/D)

Algoritmo evolutivo multiobjetivo basado en la descomposición, es descomponer la optimización multiobjetivo en varias optimizaciones de objetivos simples y al mismo tiempo evolucionarlas como comunidad, cada generación de comunidad de la evolución está compuesta por las soluciones óptimas actuales de cada sub objetivo. En MOEA / D, la optimización de cada objetivo secundario sólo necesita Información de su vecino. [63]

La relación de vecinos entre los objetivos sub es decidida por la distancia entre cada vector de peso de la función objetivo. Cuanto más cerca esté la distancia del vector de peso entre dos sub-objetivos, más aproximadas serán inevitablemente sus soluciones. [63]

2.1.20. Algoritmo Genético de Ordenación No-dominada (NSGA-II)

Es un algoritmo genético generacional basado en la creación de nuevos individuos a partir de la población original mediante la aplicación de los operadores genéticos de selección, cruce y mutación. Se aplica un procedimiento de ranqueo para promover la convergencia, mientras que un estimador de densidad (La distancia de Crowding) se utiliza para mejorar la diversidad del conjunto de soluciones encontradas. [53]

2.1.21. Algoritmo de Selección Multiobjetivo basado en Hipervolumen Dominado (SMS-EMOA)

El SMS-EMOA ha sido diseñado para cubrir un hipervolumen máximo con un número finito de puntos. El SMS-EMOA combina ideas tomadas de otras EMOA, como el bien establecido NSGA-II [10] y las estrategias de archivo presentado por Knowles [64] [65]. Se trata de un algoritmo de estado estacionario basado en dos pilares: 1) la clasificación no-dominada se utiliza como criterio de clasificación y 2) el hipervolumen se aplica como criterio de selección para descartar ese individuo, que contribuye con la menor hipervolumen a lo peor, alineado frente. [66]

2.2. Marco referencial

2.2.1. Implementación de la principal referencia metaheurística de optimización multiobjetivo NSGA-II aplicada a la filogenética computacional

Srinivas y Deb desarrollaron el Algoritmo Genético de Ordenación No-Dominada (Non-dominated Sorting Genetic Algorithm - NSGA) basándose en sistemas de clasificación de la población por niveles. A todos los individuos no-dominados se les asigna una categoría con un fitness proporcional al tamaño de la población. Para mantener la diversidad de la población, estos individuos son clasificados según este fitness y el resultado obtenido por un parámetro de distribución (parámetro de compartición) [10].

A continuación, una vez clasificado este conjunto de individuos, se elimina de la población y se repite el proceso con las soluciones restantes, esto se realiza hasta que toda la población está clasificada. Dado que las primeras soluciones son las de mejor calidad, de estas siempre se realizarán más copias, permitiéndose así una búsqueda más profunda de las regiones no-dominadas. Deb, presentó una versión mejorada del algoritmo denominada NSGA-II a través de un mecanismo de ordenación no-dominada de baja complejidad $O(MN^2)$, un operador de selección para combinar la población padre con la hija, y seleccionando los N mejores individuos teniendo en cuenta su calidad y distribución en el frente de Pareto. [10]

El proceso de adaptación del algoritmo NSGA-II [10] para ser aplicado a la inferencia de árboles filogenéticos fue el siguiente:

- *Representación de los individuos:* Todos los individuos son representados como árboles filogenéticos, basados en la clase *TreeTemplate*, los cuales están compuesto por un nodo padre y n nodos hijos, cada uno con su respectivo Nombre, ID, longitud de rama e ID del Nodo padre. [10]
- *La Población inicial:* Para definir el punto de partida del algoritmo genético se implementaron tres funcionalidades:
 - Aleatoria
 - Definidos por usuario
 - Árboles generados mediante el método Stepwise-Addition
- *Operador de Cruce:* Operador Prune-Delete-Graft (PDG) el cual ha demostrar generar mejores resultados sobre los diferentes criterios de optimización. [10]

- *Operadores de Mutación:* Se implementaron tres tipos de operadores de mutación de modificaciones topológicas: Nearest Neighbour Interchange (NNI), Tree Bisection and Reconnection (TBR) y Subtree Pruning and Regrafting (SPR). [10]
- *Funciones Objetivo:* Para estimar la máxima parsimonia se usaron las funcionalidades de la clase DRTreeParsimonyScore de la librería PhyLib de Bio++ y para la estimación de la máxima verosimilitud se emplearon las funcionalidades de la función PLF (Phylogenetic Likelihood Function) de la librería PLL bajo el modelo de sustitución GTR+ Γ . [10]
- *Búsquedas Locales para la exploración del espacio de árboles:* Se han incorporado dos técnicas de Búsqueda Local LS (Progressive Tree Neighbourhood (PPN) y pllRearrangeSearch de la librería PLL) específicas para la inferencia de árboles filogenéticos optimizando dos criterios de simultáneamente: la máxima parsimonia y máxima verosimilitud. [10]
- *Intervalo de Actualización de Parámetros:* Es por esto que el algoritmo incluye dos métodos de búsqueda de máximos y mínimos para la optimización de las longitudes de las ramas: NewtonRaphson y Gradient. [10]

```

P ← generarPoblacionInicial(NP)
P ← OptParametrosDelModeloSust(P)
P ← OptLongitudesRamas(P, BLP)
P ← evaluarPoblación(P, NP)
mientras no condición de terminación hacer
  para i = 1: NP todos individuos hacer
    ind1 ← funcionSeleccion(P, ECR)
    ind2 ← funcionSeleccion(P, ECR) // ind1 ≠ ind2
    Qi ← recombinacion(ind1, ind2, CR)
    Qi ← mutacion(Qi, PM, M)
    Qi ← OptFilogeneticaExhaustivaLS(Qi, LSP)
  fin para
  Q ← EvaluarPoblacion(Q, NP)
  R ← P ∪ Q
  R ← ordenacionNoDominada(R) // R = (F1, F2, ...)
  P ← ∅
  i ← 1
  mientras |P + Fi| < NP hacer
    P ← P ∪ Fi
    i ← i + 1
  fin mientras
  F1 ← OrdenacionPorDistanciaCrowding(Fi)
  P ← P ∪ Fi [1 : (NP - |P|)]

Si se cumple intervalo de actualización entonces
  P ← OptLongitudesRamosExa(P, BLP)
fin mientras

```

Pseudocódigo 1.- Algoritmo NSGA-II aplicada a la filogenética.

2.2.2. Propuesta multiobjetivo basado en el algoritmo de luciérnaga (MO-FA) para inferir filogenias

El Multiobjective Firefly Algorithm (MO-FA) es una adaptación multiobjetivo de un novedoso algoritmo de inteligencia de enjambre, el Firefly Algorithm (FA). El comportamiento de las luciérnagas se fundamenta en los conceptos de atracción y bioluminiscencia. Mediante la emisión de destellos de luz, las luciérnagas son capaces de atraer posibles parejas, que preferirán a aquellas luciérnagas que muestren los patrones más intensos de emisión de luz. [6]

Esta atracción se ve afectada por tres factores:

- La intensidad de la luz emitida
- La distancia entre luciérnagas
- El grado de absorción de luz por parte del entorno

```
Inicializar y evaluar la población ( $X$ , tamañoEnjambre)
para numIter = 1 to numGeneraciones hacer
  para  $r = 1$  to tamañoEnjambre hacer
    para  $s = 1$  to tamañoEnjambre hacer
      si  $X_s > X_r$  entonces
        Mover  $X_r$  hacia  $X_s$  ( $R_r \cdot M_1 X_s \cdot M_1 \beta_0 \gamma, \alpha$ )
      fin si
    fin para
    Aplicar un método de distancias para inferir el árbol  $X_r.T (X_r.M)$ 
    Evaluar la solución ( $X_r.T$ )
  fin para
  Actualizar el conjunto de Pareto con las soluciones más prometedoras ( $X$ )
fin para
```

Pseudocódigo 2.- Algoritmo MOFA aplicada a la filogenética.

La adaptación de este algoritmo al problema de la inferencia filogenética, se basada en el empleo de matrices de distancias para la representación del individuo, aplicando *métodos basados en distancias* para generar las topologías. [6]

MO-FA toma los siguientes parámetros de entrada:

1. *tamañoEnjambre* (tamaño de la población)
2. *numGeneraciones* (número de generaciones)
3. β_0 (factor de atracción)
4. γ (coeficiente de absorción)
5. α (factor de aleatoriedad).

2.2.3. Adaptación multiobjetivo del algoritmo Artificial Bee Colony (MOABC) aplicada a la inferencia filogenética

Artificial Bee Colony (ABC), es un algoritmo desarrollado por D. Karaboga inspirado en el comportamiento de las abejas de la miel a la hora de descubrir y explotar fuentes de alimento, pensado para la resolución de problemas de optimización. Forma parte de los algoritmos de inteligencia de enjambre (Swarm Intelligence) que basan su planteamiento en la definición de un conjunto de individuos, con un rol definido dentro de una sociedad, y que interactúan entre sí bajo una serie de reglas para solucionar un problema conjuntamente [7]. El algoritmo ABC se inspira en los roles que asumen tres tipos de abejas en el enjambre, a saber:

- Obreras. Se encargan de buscar y explotar fuentes de alimento. Pueden seleccionar de manera alternativa otras fuentes vecinas cercanas a la que están explotando en un momento dado [7].
- Observadoras. Las obreras, al volver al panal, comunican la calidad de las fuentes de comida que han encontrado (en términos de cercanía, cantidad de néctar...) a las abejas observadoras, expresando esta información mediante la ejecución de un baile. En virtud de los bailes realizados, la observadora decidirá cuál de las fuentes elegirá para su explotación [7].
- Exploradoras. Inspeccionan aleatoriamente el entorno en busca de nuevas fuentes de alimentos. Evitan que el enjambre se quede sin néctar cuando se agotan las fuentes descubiertas por los otros tipos de abejas [7].

La versión multiobjetivo del algoritmo ABC para la inferencia filogenética toma en cuenta dos objetivos conjuntos: Parsimonia y Verosimilitud. La evaluación de los árboles no puede llevarse a cabo de las maneras tradicionales pues son tan importantes las soluciones que optimicen la parsimonia como aquellas que hagan lo propio con la verosimilitud [7].

MOABC toma los siguientes parámetros de entrada:

1. *tamañoEnjambre*. Tamaño de la población [7].
2. *numIteraciones*. Número de iteraciones a realizar [7].
3. *límite*. Parámetro de control que evita el estancamiento de la población [7].
4. *mutación*. Ratio de mutación a aplicar sobre las soluciones para encontrar otras nuevas [7].

La salida del algoritmo estará formada por aquellos árboles filogenéticos que supongan un consenso entre los criterios de parsimonia y verosimilitud, constituyendo un conjunto de soluciones de Pareto no-dominadas [7].

```

inicializarYEvaluarPoblacion(tamañoEmjambre/2)
FrentePareto ← 0
i ← 0
mientras i < numIter hacer
    para j = 1 : tamañoEmjambre/2 hacer
        nuevaObrera ← generarVecina(C[j], mutación)
        si MOfitness(nuevaObrera) < MOfitness(C[j]) entonces
            C[j] ← nuevaObrera
        fin si
    fin para
    vectorProb ← calcularProbabilidadesSeleccion(C)
    para j = (tamañoEmjambre/2) + 1 : tamañoEmjambre hacer
        candidata ← seleccionObrera(vectorProb, C)
        nuevaCandidata ← generarVecina(candidata, mutación)
        si MOfitness(nuevaCandidata) <= MOfitness(candidata) entonces
            C[j] ← nuevaCandidata
        caso contrario
            C[j] ← candidata
        fin si
    fin para
    para j = 1 : tamañoEmjambre/2 hacer
        si C[j].iteraciones > limite entonces
            C[j] ← generateScoutBee()
        fin si
    fin para
    C ← ordenacionNoDominada(C)
    FrentePareto ← almacenarSoluciones(C, FrentePareto)
    i ← i + 1
fin mientras

```

Pseudocódigo 3.- Algoritmo MOABC aplicada a la filogenética.

2.2.4. MO-Phylogenetics: Una Herramienta de Software de la Inferencia Filogenética con Metaheurísticas Evolutivos Multiobjetivo

MO-Phylogenetics ha sido desarrollado integrando las funcionalidades de tres frameworks software multidisciplinares, jMetalCpp [59] un framework de optimización multiobjetivo, el conjunto de librerías bioinformáticas BIO++ [60] y la librería filogenética PLL (Phylogenetic Likelihood Library) [61] [13].

Entre la lista de los algoritmos que dispone en jMetalCpp y que han sido incluidos y adaptados en el software están: dos técnicas clásicas, pero aun comúnmente usadas, Nondominated Sorting Genetic Algorithm-II (NSGA-II) [67] y Pareto Archived Evolution Strategy (PAES) [68], y dos técnicas más modernas Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [69] y Multiobjective selection based on dominated hypervolume (SMS-EMOA) [70] [13].

Estos algoritmos comienzan por la inicialización de un conjunto de posibles soluciones que se desarrollaron de forma iterativa por los operadores de variación que aplican, tales como cruce o mutación [13].

Para optimizar los problemas de inferencia filogenética, un algoritmo evolutivo tiene que incorporar una estrategia adecuada para crear la población inicial y operadores variaciones específicas para la exploración del espacio árbol debe ser utilizado para hacer frente a los objetivos de máxima parsimonia y máxima verosimilitud [13].

CAPÍTULO III

MÉTODOLOGÍA DE LA INVESTIGACIÓN

3.1. Tipo de investigación

Exploratoria

Esta investigación permitirá introducir la propuesta dentro del problema de los árboles filogenéticos inferidos debido a que el algoritmo Optimización Multiobjetivo basada en Mallas Variables (MOVMO) no había sido estudiada dentro del campo de la Bioinformática.

3.2. Métodos de investigación

Método Comparativo

Este método nos permitirá comparar los resultados multiobjetivos y biológicos de la propuesta PhyloMOVMO con tres algoritmos multiobjetivos (NSGA-II, MOEA/D y SMSEMOA) representativos del estado de arte que infieren filogenias. El enfoque comúnmente adoptado es aplicar indicadores de calidad que den una medida sobre el grado de convergencia y/o diversidad de las aproximaciones de Frente de Pareto obtenidas, con la finalidad de establecer similitudes o diferencias y así poder sacar conclusiones acerca del rendimiento de la propuesta.

3.3. Población y Muestra

En este apartado se detalla los conjuntos de datos de nucleótidos (ADN) que se han seleccionado para la investigación bajo el confiable General Time Reversible (GTR + Γ) modelo evolutivo [71], estos conjuntos de datos se encuentran en GenBank [72], un repositorio muy conocido a nivel mundial y que aloja gran cantidad de base de datos de secuencia genética, de las cuales se escogieron cuatro tipos de secuencia de ADN las misma que se detallan en la siguiente tabla:

Tabla 1.- Conjunto de datos de nucleótidos.

Nombre	Descripción
rbcL_55	Comprende 55 secuencias (cada secuencia tiene 1314 sitios) del gen rbcL Cloroplasto de plantas verdes.
mtDNA_186	Contiene 186 secuencias de ADN mitocondrial humano (cada secuencia con 16608 sitios) obtenidos de la base de datos del genoma humano mitocondrial (MtDB).
RDPII_218	Comprende 218 secuencias procarióticas de ARN (cada secuencia contiene 4182 sitios) tomado del Proyecto de Base de Datos Ribosomal II.
ZILLA_500	Incluye 500 secuencias del gen rbcL (cada secuencia tiene 1428) de plaguicidas vegetales.

FUENTE: GENBANK

ELABORADO: AUTOR

3.4. Fuentes de recopilación de información

Fuente Secundaria

En el proyecto de investigación para la recopilación de información se utilizará esta técnica puesto que se obtendrá información de publicaciones de revistas científicas y libros digitales.

3.5. Recursos humanos y materiales

3.5.1. Recursos Humanos

Tabla 2.- Recursos Humanos.

Personal	Función
Carlos Alfredo Hidalgo Reyes	Desarrollador
Ing. MSc. Cristian Gabriel Zambrano Vega	Director de Proyecto de Investigación

FUENTE: CONOCIMIENTO PROPIO

ELABORADO: AUTOR

3.5.2. Recursos de Hardware

Tabla 3.- Recursos de Hardware.

Componente	Características técnicas
Servidor	HP ProLiant DL380 Gen9 E5-2650v3
	<ul style="list-style-type: none"> ✓ 2 procesadores Intel Xeon E5-2650v3 10-Core (2.30GHz) /25MB L3 Cache ✓ 32GB (2 x 16GB) DDR4 2133MHz RDIMM de Memoria RAM ✓ 1Gb Ethernet 4-port 331i Adapter ✓ FlexFabric 10Gb 2P 533FLR-T Adapter ✓ Smart Array P440ar/2G Controller Module / (RAID /1/1+0/5/5+0/6/6+0)/ (8) SFF ✓ SAS/SATA HDD bahías ✓ Unidad de SATA DVD-RW/ 6 PCIe slots ✓ 2 fuentes de 800W Flexible Slot Platinum Hot Plug Power Supply ✓ 6 hot plug fans, redundant/ Rack (2U)
	Disco Duro HP 300GB 12G SAS 10K 2.5in SC ENT HDD
Laptop	HP Pavilion g4
	<ul style="list-style-type: none"> ✓ Procesador: Intel Core i5-2450M ✓ Sistema operativo: Windows 10 Pro ✓ Memoria: 4 GB DDR3 1333Mhz ✓ Disco duro: 640 GB

FUENTE: CONOCIMIENTO PROPIO

ELABORADO: AUTOR

3.5.3. Recursos de Software

Tabla 4.- Recursos de Software.

Componente	Función
NetBeans IDE 8.0.2	Proceso de adaptación y ejecución del algoritmo.
Microsoft Office 2016	Desarrollo de documentos de entrega y documentación.
Matlab	Generación de plot (gráficos de Frentes de Pareto).
Latex	Visualización de resultados (Indicadores de Calidad)
MEGA7	Visualización de los árboles filogenéticos.

FUENTE: CONOCIMIENTO PROPIO

ELABORADO: AUTOR

3.5.4. Insumos/Materiales

Tabla 5.- Insumos/Materiales.

	Descripción
Internet	Acceso a internet para realizar búsquedas necesarias para el desarrollo de la investigación
Materiales de oficinas	<ul style="list-style-type: none">✓ Lápices y Lapiceros✓ Resma Hojas A4✓ Pen drivers✓ Impresora✓ Carpetas
Materiales de impresión	<ul style="list-style-type: none">✓ Copias✓ Anillados✓ Empastados

FUENTE: CONOCIMIENTO PROPIO

ELABORADO: AUTOR

CAPÍTULO IV
RESULTADOS Y DISCUSIÓN

En el presente capítulo se detalla la manera en que se realizó el estudio experimental para determinar cómo obtener árboles filogenéticos inferidos mediante el algoritmo MOVMO, se hizo la integración de varias funcionalidades de frameworks del área de la bioinformática, luego una adaptación del algoritmo a la inferencia de árboles filogenéticos y posteriormente la experimentación entre los resultados obtenidos por PhyloMOVMO con los algoritmos propuesto en el estado del arte.

4.1. Integración de frameworks del área de la bioinformática al algoritmo MOVMO

PhyloMOVMO ha sido implementado utilizando funcionalidades del framework MO-Phylogenetics [13], como son los dos paquetes de filogenética, el conjunto de librerías bioinformáticas BIO++ [60] y las funciones de librería filogenética PLL (Phylogenetic Likelihood Library) [61].

- **BIO++:** Es un framework bioinformático que provee un conjunto de librerías para el desarrollo de software para varias áreas de la Bioinformática, de este framework se usó la librería *SeqLib* (Sequence Library) ya que nos permitirá manipular y analizar varios tipos de secuencias biológicas entre ellas la de ADN, asimismo para la lectura y manejo de las secuencias de entrada (alineamientos múltiples de secuencias), la cual provee soporte formatos *Phylip* y *Fasta*, también se utilizó la librería filogenética *PhylLib* (Phylogenetics Library) que proporciona varias funcionalidades para la lectura, almacenamiento y manipulación de árboles filogenéticos, permite evaluar el criterio de la Máxima Parsimonia, además también se basó en la clase *TreeTemplate* para la representación de los individuos como árboles filogenéticos y la implementación del operador genético de cruce.
- **PLL:** Es una librería altamente optimizada que provee funcionalidades para la estimación computacional de la máxima verosimilitud sobre árboles filogenéticos, de la cual se utilizó la función PLF (Phylogenetic Likelihood Function) para la estimación de la máxima verosimilitud sobre el modelo de sustitución de nucleótidos GTR+ Γ , se usó también el método numérico *Newton-Raphson* con el objetivo de mejorar la Verosimilitud en la optimización de las longitudes de rama, y la función *pllRearrangeSearch* para la implementación de una de las técnicas para explorar el espacio de búsqueda de los árboles del algoritmo.

4.2. Adaptación Filogenética del algoritmo MOVMO

El algoritmo MOVMO propone dos operaciones principales, expansión y contracción de una malla de soluciones, la cual se encarga de realizar la exploración del espacio de búsqueda. En el algoritmo PhyloMOVMO estas operaciones fueron resumidas utilizando el operador de cruce, la justificación de esta decisión fue tomada debido a que la versión MOVMO realiza un promedio entre los nodos mientras que para árboles filogenético lo más relacionado a un promedio entre árboles es un consenso, pero el problema de un árbol consenso se debe a que no calcula las distancias entre los árboles y solo calcula la bipartición es decir la distancia que hay entre las ramas.

A continuación, describiremos con más detalles la forma en que se ajustaron algunos de los elementos en el algoritmo MOVMO para poder ser aplicada al problema de la inferencia filogenética.

4.2.1. Representación de los individuos

Todos los individuos son representados como árboles filogenéticos, basados en la clase *TreeTemplate*, los cuales están compuesto por un nodo padre y n nodos hijos, cada uno con su respectivo Nombre, ID, Longitud de rama e ID del Nodo Padre, las hojas son nodos sin descendiente y la raíz está definida sin nodo padre, los nodos internos generalmente contendrán dos descendientes formando así la estructura de un árbol bifurcado, es decir que la clase *TreeTemplate* guarda acceso solo a un nodo y cada nodo tiene el nodo padre apuntando al padre y apuntando a sus nodos hijos.

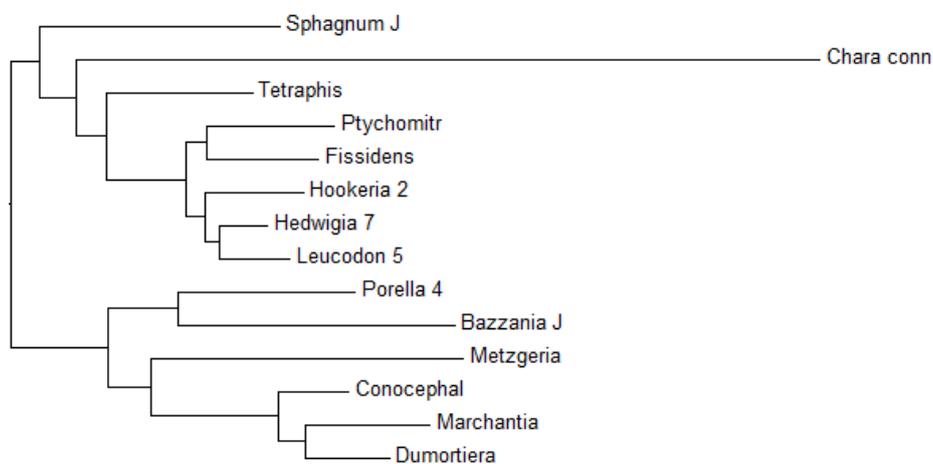


Ilustración 4.- Estructura de un árbol bifurcado.

4.2.2. Operador de Cruce

En el algoritmo propuesto se ha utilizado el operador de recombinación Prune-Delete-Graft (PDG) [73] disponible en MO-Phylogenetics. Este operador combina un subárbol de dos árboles padres y crea dos nuevos árboles descendientes. Dado los árboles T_1 y T_2 , este operador realiza los siguientes pasos:

1. Poda un subárbol s escogido al azar desde T_1 ;
2. Quita todas las hojas de T_2 que también están en s ;
3. Crea un nuevo descendiente T_1' insertando el subárbol s a un borde aleatorio de T_2 .

El segundo descendiente, denominado T_2' , se crea de manera similar: poda un subárbol de T_2 y lo inserta en T_1 . En resumen, este operador toma un subárbol aleatorio de uno de los árboles y lo inserta en el otro árbol en un punto de inserción seleccionado aleatoriamente, eliminando especies duplicadas.

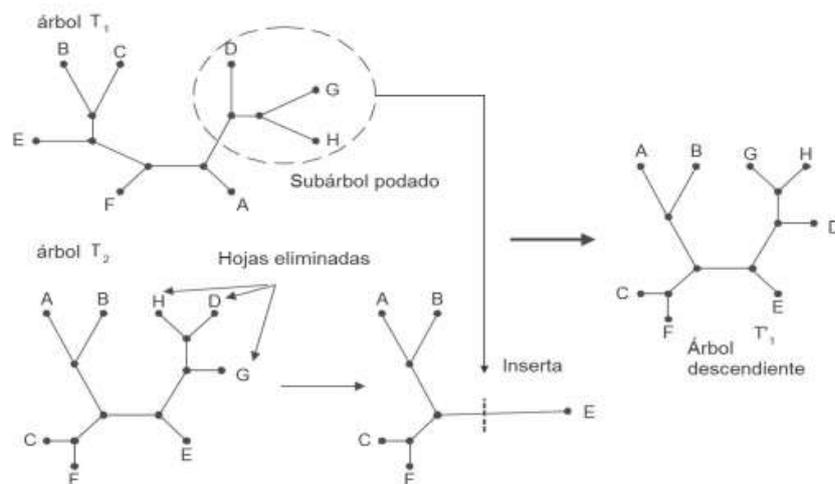


Ilustración 5.- Ejemplo del operador de cruce PDG.

4.2.3. Distancia entre árboles filogenéticos

Para calcular la distancia entre dos árboles filogenético se utilizó el método *robinsonFouldsDistance* propia del framework BIO++, este método utiliza la métrica Robinson-Foulds la cual mide la distancia entre árboles filogenético sin raíz. La distancia de Robinson-Foulds (RF) entre dos árboles es el número de biparticiones que difieren entre ellos, si el número de bipartición entre los árboles son diferente el valor que devuelve es 1.

Sea $\Sigma(T)$ el conjunto de biparticiones definidas por todos los bordes en el árbol T . La distancia RF entre los árboles T_1 y T_2 se define como:

$$d_{RF}(T_1, T_2) = \frac{|\Sigma(T_1) - \Sigma(T_2)| + |\Sigma(T_2) - \Sigma(T_1)|}{2} \quad (4)$$

En la siguiente ilustración se muestra la descripción general de la computación de la matriz de distancia RF. La colección de árboles se compone de cuatro filogenias: T_1 , T_2 , T_3 y T_4 . Biparticiones (o bordes internos) en un árbol se denominan B_i , donde i varía de 1 a 2.

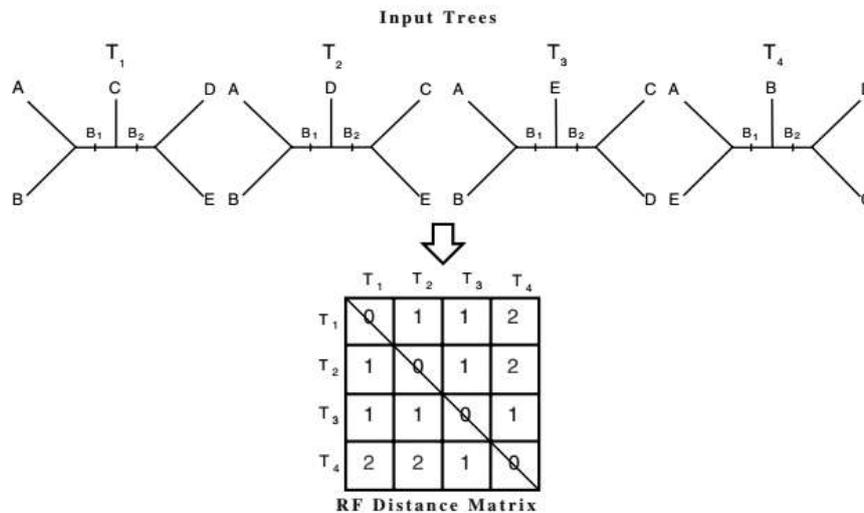


Ilustración 6.- Métrica de Robinson-Foulds

4.2.4. Técnica de Optimización Filogenética

Se han incorporado dos técnicas de Búsqueda Local LS (Local Search) específicas para la inferencia de árboles filogenéticos optimizando dos criterios de simultáneamente: la máxima parsimonia y máxima verosimilitud:

- La primera LS está basada en la alta relación que existe entre la parsimonia y la verosimilitud, en la que bajo una perspectiva teórica definida por [74] se establece que minimizar la parsimonia es equivalente a maximizar la verosimilitud bajo ciertos supuestos (ideal en nuestro enfoque), por lo que al igual a una de las técnicas de búsquedas implementadas en el software *PhyML* [75] ésta técnica explora el espacio de árboles encontrando soluciones multiobjetivo aplicando movimientos topológicos que minimizan la parsimonia y ajusta las longitudes solo de las ramas aplicando el método de optimización Newton-Raphson luego de cada cambio realizado.

Para la optimización de la parsimonia se emplea la técnica *Parametric Progressive Tree Neighbourhood (PPN)*, propuesta por [76], la cual está definida como un conjunto de movimientos topológicos SPR en los que la distancia de la rama del subárbol podado y la rama en la que se realizará el injerto está basada en un valor d , cuyo valor inicial es la distancia máxima que hay entre el nodo raíz y las hojas del árbol, permitiendo empezar con movimientos globales, reduciendo progresivamente su valor hasta llegar a un valor mínimo de 1, con el cual los movimientos SPR representan un movimiento local similar a los que se obtienen con la técnica NNI. Solo aquellos movimientos que mejoren la parsimonia son aplicados definitivamente sobre la topología con el fin de ir optimizándola.

- La segunda LS es una combinación parametrizada de dos técnicas enfocadas a mejorar los criterios de optimización por separado: para la parsimonia usa PPN [76], cuyo funcionamiento es el mismo definido en la LS anterior, y por el lado de la verosimilitud una técnica de exploración basada en reordenamiento de topologías de la librería *Phylogenetic Likelihood Library (PLL)* llamada *pllRearrangeSearch*, la cual realiza movimientos topológicos de tipo NNI o SPR desde un nodo específico con todos los demás nodos a su alrededor y dentro de un radio de cobertura. Gracias al esquema genérico de vectorización en el que está desarrollada PLL, se puede recalcular de forma rápida y óptima la verosimilitud de los árboles después de cada movimiento topológico probado.

4.2.5. Parámetros de Entradas y Salida

MOVMO

- La entrada P , k y C son como en VMO. Sin embargo, el parámetro T (tamaño de malla máximo después del procedimiento de expansión) en VMO ya no es necesario en esta formulación multiobjetivo. Un nuevo parámetro S , el tamaño máximo de los archivos líderes (Pareto) se introduce en su lugar. [16]
- La salida devuelve el archivo de líderes L como la aproximación del conjunto óptimo de Pareto encontrado. [16]

Input: mesh size P , neighborhood size k , number maximum of evaluations C , maximum archive size S

Result: An approximation L of the true Pareto set L^*

Pseudocódigo 4.- Parámetros de E/S de MOVMO.

PhyloMOVMO

- La entrada son los mismos del algoritmo MOVMO, el tamaño de malla P , tamaño de vecinos k , número máximo de evaluaciones C , tamaño máximo de archivo S . Se ha incluido el conjunto de datos de entrada para inferir filogenias: el alineamiento múltiple de secuencia con el conjunto de secuencias alineadas, los árboles filogenéticos inicial pre-computados en formato *Newick*, y los parámetros del modelo evolutivo para cada conjunto de datos, que puede ser computada utilizando jModelTest [77].
- La salida del algoritmo será un conjunto de soluciones de Pareto no-dominadas L que describe las diversas topologías filogenéticas de acuerdo con los criterios de máxima parsimonia y máxima verosimilitud.

Input: mesh size P , neighborhood size k , number maximum of evaluations C , maximum archive size S

Data: multiple sequence alignment, initial phylogenetic tree and evolutionary model parameters

Result: An approximation L of the true Pareto set L^*

Pseudocódigo 5.- Parámetros de E/S de PhyloMOVMO.

4.2.6. Generación de la malla inicial

MOVMO

El método comienza generando la malla inicial M_0 e inicializando el archivo de líderes L con todas las soluciones no-dominadas en M_0 (Lineas 1 y 2) Este archivo almacenará solamente soluciones no-dominadas, (la aproximación óptima de Pareto encontrada hasta ahora). [16]

1: Generate the initial mesh $M = \{n_i\}$ and calculate the fitness vector $F_i \forall i \in \{1, \dots, P\}$

2: $L =$ Initialize archive with each mesh node n_i

Pseudocódigo 6.- Generación de la malla inicial en el algoritmo MOVMO.

PhyloMOVMO

Comienza generando la malla inicial Pop_0 e inicializando el archivo de líderes L con todas las soluciones no-dominadas en Pop_0 (Líneas 1 y 2). Estas soluciones iniciales se asignan aleatoriamente a partir de un repositorio compuesto por filogenias generadas por un análisis bootstrap [78] y, con el objetivo de mejorar la probabilidad, todas las longitudes de rama de estos árboles se optimizan utilizando el método numérico *Newton-Raphson* proporcionado

por MO-Phylogenetics. Estos árboles representan los nodos de partida para llevar a cabo la reconstrucción de diversas topologías.

- 1: $Pop \leftarrow \text{Initialize_Evaluate_Population}(P)$
- 2: $L = \text{Initialize archive with each mesh node } n_i$

Pseudocódigo 7.- Generación de la malla inicial en el algoritmo PhyloMOVMO.

4.2.7. Generación de nodos mediante búsqueda local

MOVMO

Se selecciona el mejor nodo n_i^* entre los k vecinos más cercanos de n_i en el espacio variable de decisión (Línea 5) según el criterio de dominancia, si dos o más de estos k vecinos están mutuamente no-dominados, se escoge el más cercano a n_i . Si el óptimo local domina n_i , un nuevo nodo n_l es generado (Línea 7); De lo contrario n_i es el óptimo local en sí (Línea 9). Se mide la distancia euclídea entre los dos nodos de malla n_i y n_i^* en el espacio objetivo y crea una probabilidad en consecuencia. [16]

- 4: *foreach node n_i in the current mesh M do*
- 5: $n_i^* \leftarrow$ the best among the k neighbors of n_i
- 6: if $n_i^* < n_i$ then
- 7: $n_l \leftarrow$ generate a new node
- 8: else
- 9: $n_l \leftarrow n_i$

Pseudocódigo 8.- Generación de nodos mediante búsqueda local en el algoritmo MOVMO.

PhyloMOVMO

Se selecciona el mejor nodo n_i^* entre los k vecinos más cercanos de n_i en el espacio variable de decisión. La distancia entre los nodos (árboles filogenéticos) se calcula de acuerdo con la métrica de *Robinson-Foulds* y el mejor nodo se selecciona de acuerdo con el criterio de dominancia multiobjetivo (Línea 5). Si el óptimo local domina n_i , se genera un nuevo nodo n_l mediante la aplicación del operador *TreeCrossover* usando n_i^* y n_i (Línea 7); De lo contrario n_i es el óptimo local en sí (Línea 9).

```

4: foreach node  $n_i$  in the current mesh Pop do
5:    $n_i^* \leftarrow$  the best among the  $k$  neighbors of  $n_i$ 
6:   if  $n_i^* < n_i$  then
7:      $n_l \leftarrow$  PDGTreeCrossover( $n_i^*, n_i$ )
8:   else
9:      $n_l \leftarrow n_i$ 

```

Pseudocódigo 9.- Generación de nodos mediante búsqueda local en el algoritmo PhyloMOVMO.

4.2.8. Generación de nodos mediante búsqueda global

PhyloMOVMO

Un líder global n_g del archivo L se selecciona a través del operador binaryTournamentSelection (Línea 10). Dos soluciones no-dominadas de L son escogidas aleatoriamente y la que tiene mejor crowding distance en L es seleccionada. En el algoritmo MOVMO se realiza el mismo procedimiento

```

4: foreach node  $n_i$  in the current mesh Pop do
...
10:  $n_g \leftarrow$  Select a global leader from  $L$  (BinaryTournament_Selection)

```

Pseudocódigo 10.- Generación de nodos mediante búsqueda global en el algoritmo PhyloMOVMO.

4.2.9. Operador de recombinación

MOVMO

La solución elite n_g se cruza con el n_l óptimo local del paso precedente (Línea 11). Se conserva el mejor de los dos descendientes (en términos de dominio, crowding distance para romper los lazos). [16]

```

4: foreach node  $n_i$  in the current mesh Pop do
...
11:    $n_x \leftarrow$  Crossover( $n_l, n_g$ );

```

Pseudocódigo 11.-Proceso de recombinación en el algoritmo MOVMO.

PhyloMOVMO

Un operador *TreeCrossover* se aplica sobre el líder global n_g con el n_l local óptimo (Línea 11) para generar una nueva solución n_x , que contiene subárboles de ambas topologías (nodos de malla). Con el objetivo de mejorar n_x , se aplica un método de optimización filogenético (Línea 12), una búsqueda local proporcionada por MO-Phylogenetics [9] que combina dos técnicas altamente optimizadas para explorar el espacio de los árboles, *pllRearrangeSearch* [79] y *PPN* [76], para optimizar los criterios de verosimilitud y parsimonia, respectivamente.

```
4: foreach node  $n_i$  in the current mesh Pop do
  ...
11:   $n_x \leftarrow$  PDGTreeCrossover( $n_l, n_g$ )
12:   $n_x$  : PhylogeneticOptimization(PPN&PLL)
```

Pseudocódigo 12.-Proceso de recombinación en el algoritmo PhyloMOVMO.

4.2.10. Evaluación y actualización de los archivos líder

PhyloMOVMO

El nuevo nodo n_x se evalúa y se agrega al archivo de líderes L (Línea 13 y 14). En la adición de un nuevo nodo de malla n_x al fichero líder limitado L . Primero, todos los nodos en L que están dominados por la solución entrante se eliminan del archivo antes de la adición de n_x . Si el archivo alcanza su tamaño máximo, eliminamos el nodo con la menor crowding distance. Esto asegura que un conjunto bien extendido de soluciones no-dominadas se mantenga en L . En el algoritmo MOVMO se realiza el mismo procedimiento.

```
4: foreach node  $n_i$  in the current mesh Pop do
  ...
13:  evaluateFitness( $n_x$ )
14:  add  $n_x$  to the leaders archive L
    1: foreach  $n_j$  of L do
    2:   if  $n_x < n_j$  then
    3:      $L \leftarrow L - n_j$  /* remove  $n_j$  from the archive */
    4:   else if  $n_x = n_j \parallel n_j \leq n_x$  then
    5:     exit /* discard  $n_x$  */
    6:  $L \leftarrow L \cup n_x$ ; /* add  $n_x$  to the archive */
    7: if  $L.size() > L : maxSize()$  then
    8:   recompute crowding distances in L
    9:    $L \leftarrow L - \{ L.worstByCrowdingDistance \}$  /* remove most crowded solution */
```

Pseudocódigo 13.- Evaluación y actualización de los archivos líder en el algoritmo PhyloMOVMO.

4.2.11. Actualización dinámica de la población

PhyloMOVMO

Si n_i está dominado por n_x , es reemplazado por n_x en la malla actual P_{op} (Línea 16). Devuelve el archivo de líderes L como la aproximación del conjunto óptimo de Pareto encontrado. En el algoritmo MOVMO se realiza el mismo procedimiento.

```
4: foreach node  $n_i$  in the current mesh  $Pop$  do
  ...
15:   if  $n_x \leq n_i$  then
16:     Replace  $n_i$  with  $n_x$  in the current population  $Pop$ 
```

Pseudocódigo 14.- Actualización dinámica de la población en el algoritmo PhyloMOVMO.

4.3. Estudio experimental del algoritmo propuesto con el estado del arte

En la presente investigación se ha realizado un estudio experimental del algoritmo propuesto con los algoritmos descrito en el estado del arte tales como: MOEAD/D, NSGA-II y SMS-emoa, dichos algoritmos se encuentran en la herramienta de software MO-Phylogenetics. Con el objetivo de hacer una comparación equitativa, se ha configurado todos los algoritmos con la misma configuración de parámetros (ver Anexo 1), también se utilizó el mismo conjunto de datos de nucleótidos (ver apartado 3.3) y finalmente se realizaron la misma cantidad de ejecuciones (10 ejecuciones) de manera independiente, cada algoritmo resolviendo cada problema.

Este estudio experimental se basa en realizar un análisis tanto multiobjetivo como biológico. Para el análisis multiobjetivo se utilizaron indicadores de calidad en donde los indicadores de Hipervolumen y la Distancia Generacional Invertida Plus, aportan un único valor que tienen en cuenta tanto la convergencia como la diversidad de las aproximaciones de Frentes de Pareto, así como los indicadores de Épsilon y Dispersión, que se utilizan como complemento para medir el grado de convergencia y diversidad, respectivamente.

Mientras que para el análisis biológico se utilizó los Best Scores (o mejores puntuaciones), esto nos permite presentar el mejor puntaje en los dos criterios de reconstrucción para árboles filogenéticos como son: la máxima parsimonia y la máxima verosimilitud.

Con la finalidad de demostrar estadísticamente si el algoritmo propuesto es competitivo frente a los algoritmos de referencia se realizó un análisis estadístico utilizando las técnicas como: la prueba de Friedman y la prueba de Wilcoxon.

A continuación, se presenta de manera más detallada cada uno de los análisis que se realizaron tanto al algoritmo propuesto como a los algoritmos de referencias en cada uno de los problemas de pruebas.

4.3.1. Análisis Multiobjetivos

El enfoque comúnmente adoptado es aplicar indicadores de calidad que den una medida sobre el grado de convergencia y / o diversidad en los resultados de las aproximaciones de Frentes Pareto. Se han obtenido las medianas y rangos intercuartílicos (IQR) de los resultados de los indicadores de calidad como, Hipervolumen, Épsilon, Spread y Distancia Generacional Invertida Plus (ver tablas 6 al 9). En las tablas de resultados de los indicadores de calidad, los valores obtenidos son resaltados mediante un fondo de color que describen la calidad del resultado, los mejores de cada experimento son marcados con un fondo gris más oscuro y los segundos mejores resultados son marcados con tono de gris más claro. Hay que considerar que en el caso de los indicadores I_{HV} y I_{IDG+} los valores mayores son los mejores, y de forma contraria (los valores menores) para los indicadores I_{e+} y I_{Δ} .

En las tablas 6, 7 y 9 se presentan los resultados de I_{HV} , I_{e+} y I_{IDG+} respectivamente, en ellas se muestran que PhyloMOVMO obtiene los mejores valores medianas para todos los conjuntos de datos, excepto para la instancia rbcL_55, donde MOEA/D muestra un mejor rendimiento. Y para los resultados de I_{Δ} ocurre lo mismo, NSGA-II obtiene los mejores valores medianas para todos los conjuntos de datos, excepto para la instancia rbcL_55, donde SMSEMOA muestra un mejor rendimiento.

Tabla 6.- Mediana y Rangos Intercuartílicos de los valores del indicador I_{HV} .

Dataset	PhyloMOVMO	NSGAI	MOEAD	SMSEMOA
rbcL_55	6.34e - 01 _{1.7e - 01}	0.00e + 00 _{0.0e + 00}	6.83e - 01 _{5.5e - 02}	5.81e - 01 _{1.3e - 01}
mtDNA_186	3.07e - 01 _{1.3e - 01}	2.56e - 01 _{1.1e - 01}	2.75e - 01 _{1.6e - 01}	2.40e - 01 _{1.6e - 01}
RDPII_218	6.18e - 01 _{4.2e - 02}	6.08e - 01 _{5.5e - 02}	5.87e - 01 _{8.9e - 02}	5.99e - 01 _{4.1e - 02}
ZILLA_500	1.57e - 02 _{1.0e - 01}	0.00e + 00 _{1.1e - 02}	2.88e - 03 _{8.7e - 02}	0.00e + 00 _{3.1e - 02}

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 7.- Mediana y Rangos Intercuartílicos de los valores del indicador I_{E+} .

Dataset	PhyloMOVMO	NSGAI	MOEAD	SMSEMOA
rbcL_55	2.10e - 01 _{1.4e - 01}	1.01e + 00 _{5.0e - 01}	1.75e - 01 _{4.1e - 02}	2.50e - 01 _{8.5e - 02}
mtDNA_186	3.33e - 01 _{1.5e - 01}	3.38e - 01 _{1.1e - 01}	3.89e - 01 _{1.7e - 01}	4.44e - 01 _{2.1e - 01}
RDPII_218	1.18e - 01 _{3.6e - 02}	1.36e - 01 _{2.7e - 02}	1.59e - 01 _{5.6e - 02}	1.51e - 01 _{4.7e - 02}
ZILLA_500	7.68e - 01 _{3.8e - 01}	9.33e - 01 _{2.0e - 01}	8.13e - 01 _{3.7e - 01}	9.38e - 01 _{3.0e - 01}

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 8.- Mediana y Rangos Intercuartílicos de los valores del indicador I_{Δ} .

Dataset	PhyloMOVMO	NSGAI	MOEAD	SMSEMOA
rbcL_55	9.91e - 01 _{2.5e - 01}	1.01e + 00 _{3.4e - 01}	1.14e + 00 _{2.8e - 01}	8.76e - 01 _{3.8e - 01}
mtDNA_186	1.31e + 00 _{3.9e - 01}	7.97e - 01 _{5.0e - 01}	1.30e + 00 _{1.5e - 01}	1.13e + 00 _{7.7e - 01}
RDPII_218	8.89e - 01 _{1.8e - 01}	7.99e - 01 _{6.8e - 02}	1.13e + 00 _{9.4e - 02}	9.11e - 01 _{1.8e - 01}
ZILLA_500	1.09e + 00 _{1.6e - 01}	8.44e - 01 _{1.1e - 01}	1.16e + 00 _{8.4e - 02}	9.74e - 01 _{2.4e - 01}

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 9.- Mediana y Rangos Intercuartílicos de los valores del indicador $IDG+$.

Dataset	PhyloMOVMO	NSGAI	MOEAD	SMSEMOA
rbcL_55	1.10e - 01 _{1.2e - 01}	9.35e - 01 _{5.2e - 01}	8.71e - 02 _{4.0e - 02}	1.48e - 01 _{7.7e - 02}
mtDNA_186	1.90e - 01 _{1.2e - 01}	2.33e - 01 _{8.9e - 02}	2.26e - 01 _{2.0e - 01}	2.37e - 01 _{2.2e - 01}
RDPII_218	6.87e - 02 _{2.5e - 02}	7.55e - 02 _{3.5e - 02}	8.57e - 02 _{5.0e - 02}	7.77e - 02 _{3.2e - 02}
ZILLA_500	5.56e - 01 _{6.3e - 01}	8.25e - 01 _{3.2e - 01}	6.97e - 01 _{4.7e - 01}	5.99e - 01 _{2.0e - 01}

FUENTE: JMETALCPP

ELABORADO: AUTOR

4.3.2. Frente de Pareto de Referencia y Aproximaciones de los Frentes de Pareto

Al tratar con problemas de optimización en el mundo real, los frentes de Pareto no se conocen, por lo que se ha generado un frente de Pareto de referencia para cada conjunto de datos de nucleótidos el mismo que combina todas las soluciones no-dominadas en todas las ejecuciones de todos los algoritmos. Esta estrategia permite realizar una evaluación relativa del rendimiento de los algoritmos, ya que si el comportamiento de todas las técnicas comparadas es deficiente sabemos cuál de ellos rinde los mejores frentes, pero no sabemos si están cerca o lejos del verdadero frente de Pareto.

Para ilustrar gráficamente los resultados de los indicadores de calidad multiobjetivo (ver apartado 4.3.1), se ha apoyado proporcionando algunos ejemplos de las aproximaciones de Frentes de Pareto producidas por todos los algoritmos de los indicadores I_{HV} e I_{IDG^+} , considerando que ambas toman en cuenta la convergencia y diversidad de las aproximaciones de Frente de Pareto. Las ilustraciones 7 y 8 muestran el frente de referencia y las aproximaciones de Frentes de Pareto de cada algoritmo de cada conjunto de datos de nucleótidos, a partir de los mejores valores de los indicadores I_{HV} e I_{IDG^+} , respectivamente.

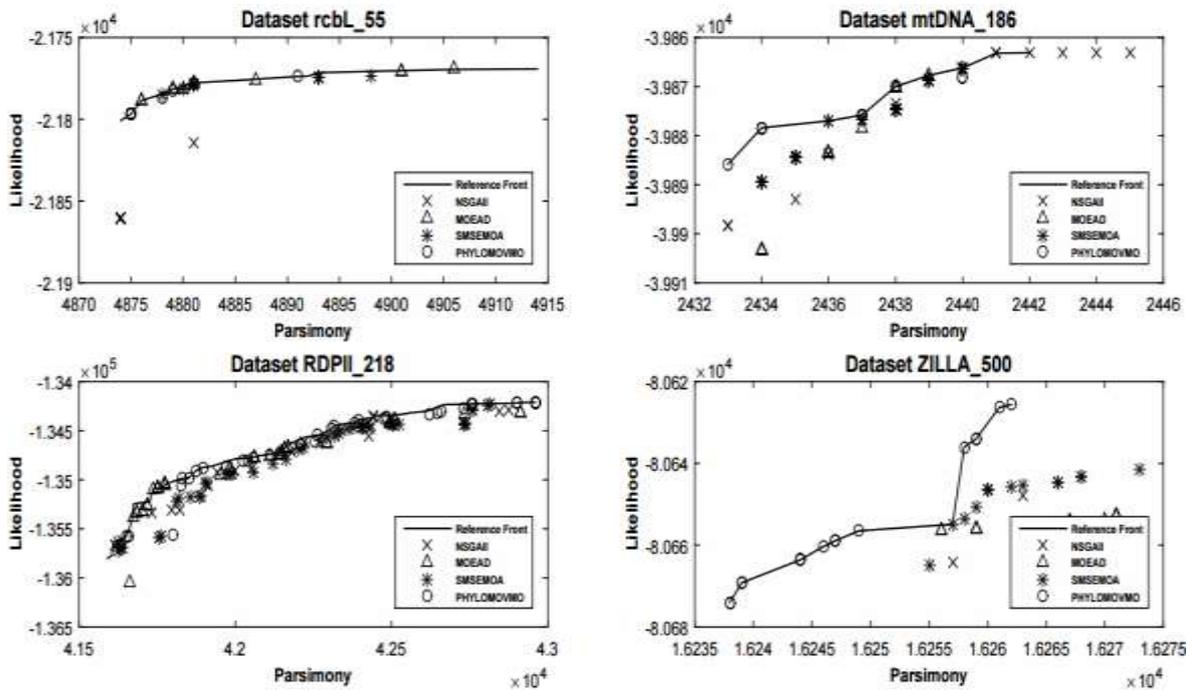


Ilustración 7.- Aproximaciones de Frentes de Pareto de los mejores valores de I_{HV} obtenidos por todos los algoritmos (PhyloMOVMO, NSGAI, MOEA / D y SMSEMOA) a lo largo de las 10 secuencias independientes que resuelven cada conjunto de datos de nucleótidos.

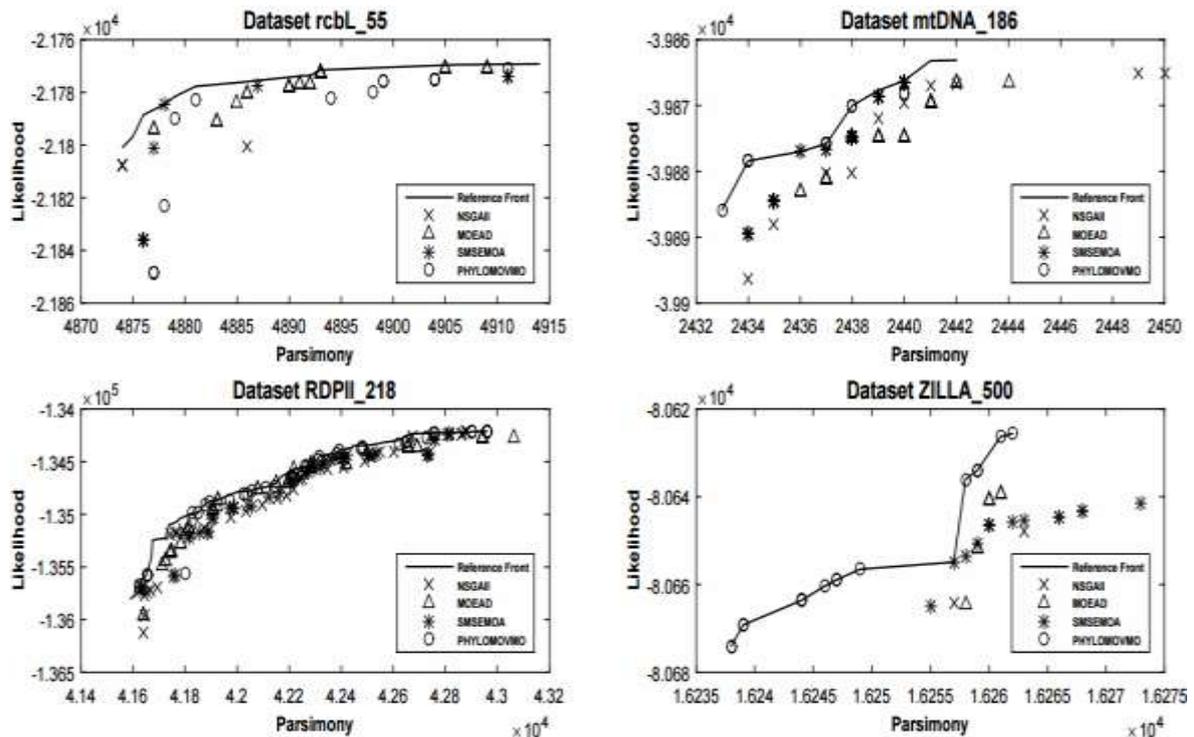


Ilustración 8.- Aproximaciones de Frentes de Pareto de los mejores valores de IDG+ obtenidos por todos los algoritmos (PhyloMOVMO, NSGAI, MOEA / D y SMSEMOA) a lo largo de las 10 secuencias independientes que resuelven cada conjunto de datos de nucleótidos.

Además, en la *Ilustración 9* se presenta el frente de Pareto de Referencia generado combinando todas las soluciones no dominadas calculadas en todas las ejecuciones de todos los algoritmos, y las mejores aproximaciones de Frente Pareto obtenidos por todos los algoritmos (PhyloMOVMO, NSGA-II, MOEA/D Y SMSEMOA) sobre 10 ejecuciones independientes que resuelven los conjuntos de datos de nucleótidos (ver apartado 3.3).

Podemos observar que todos los Frentes de Pareto de Referencia (*Ilustración 9*), están conformados en su mayoría por las soluciones no dominadas (filogenias) por las aproximaciones de Frente de Pareto de PhyloMOVMO, considerando sólo unas pocas soluciones de MOEA/D y SMSEMOA para los conjuntos de datos rbcL_55 y ZILLA_500, respectivamente. También, en la *Ilustración 9c* podemos observar un alto rendimiento competitivo que existe entre todos los algoritmos que resuelven el conjunto de datos de nucleótidos RDPII_218.

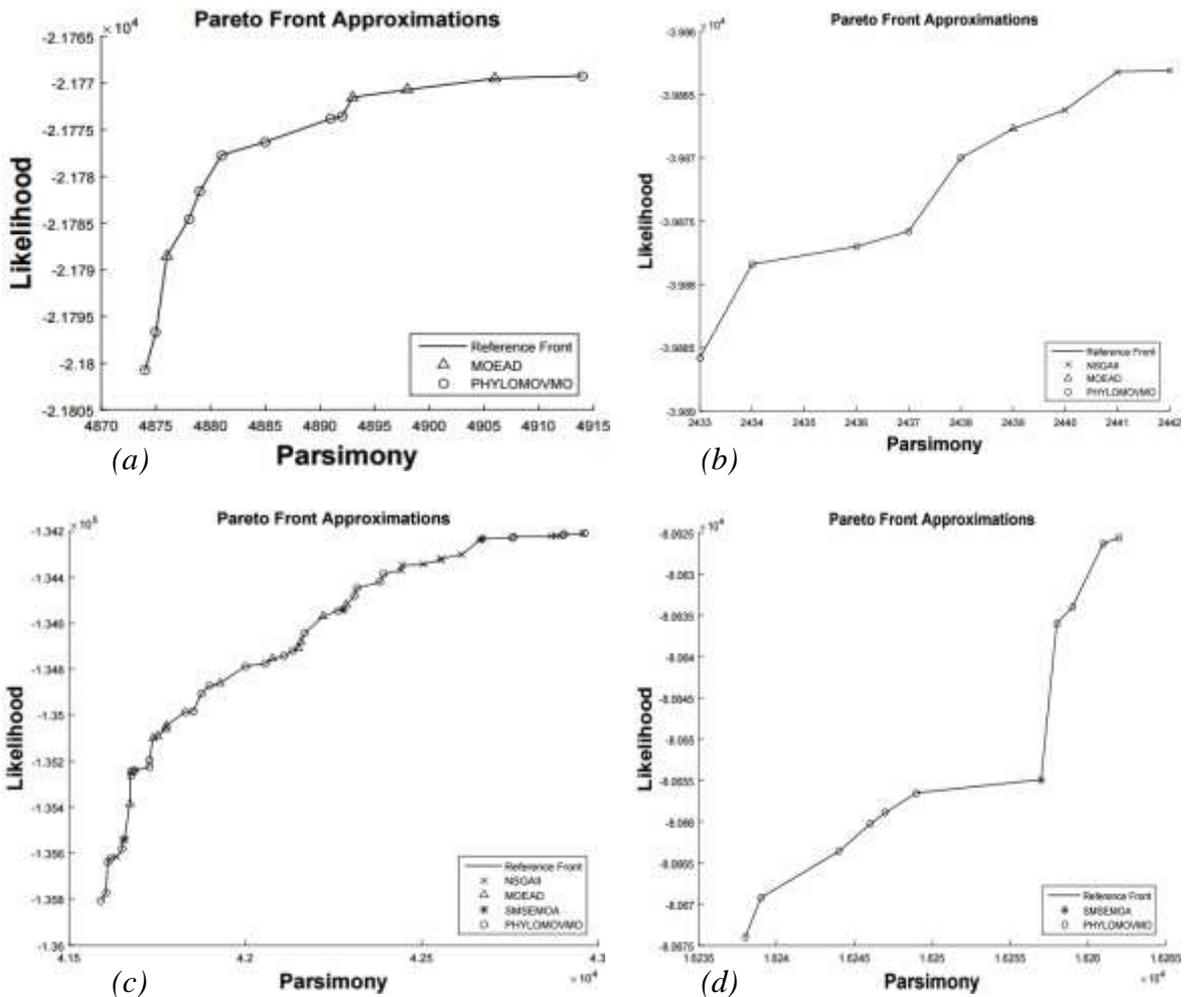


Ilustración 9.- Frentes de Pareto de Referencia y las mejores aproximaciones del Frente de Pareto obtenidos por todos los algoritmos en las 10 ejecuciones independientes resolviendo los problemas a) rbcL_55, b) mtDNA_186, c) RDPII_218 y d) ZILLA_500.

4.3.3. Análisis Biológico

En este apartado se presenta las mejores puntuaciones de máxima parsimonia y máxima verosimilitud (valor mínimo en MP, y valor máximo en ML) obtenidas por todos los algoritmos resolviendo los cuatro conjuntos de datos de nucleótidos, de acuerdo con las configuraciones definidas.

Podemos observar en la *Tabla 10* que nuestra propuesta logra una mejora significativa con respecto a las puntuaciones de parsimonia y verosimilitud reportadas por los otros algoritmos, excepto para los conjuntos de datos de nucleótidos rbcL_55 y mtDNA_186, donde todos los algoritmos generan las mismas puntuaciones de parsimonia y NSGA-II realiza una mejor puntuación de verosimilitud global de los algoritmos, respectivamente.

Tabla 10.- Mejores Puntuaciones (Máxima Parsimonia MP, Máxima Verosimilitud ML).

Instance	PhyloMOVMO		NSGA-II		MOEAD		SMSEMOA	
	MP	ML	MP	ML	MP	ML	MP	ML
rbcL_55	4874	-21769.2	4874	-21800.8	4874	-21769.5	4874	-21773.6
mtDNA_186	2433	-39865.5	2433	-39863.1	2434	-39866.6	2434	-39864.7
RDPII_218	41589	-134210	41613	-134211	41634	-134238	41618	-134224
ZILLA_500	16238	-80625.6	16251	-80630.9	16251	-80639.4	16250	-80628

FUENTE: JMETALCPP

ELABORADO: AUTOR

Además, en las Ilustraciones 10 y 11 se muestran una comparación de los árboles filogenéticos con mejor puntaje tanto en parsimonia (izquierda) como verosimilitud (derecha) del algoritmo PhyloMOVMO resolviendo los conjuntos de datos nucleótidos rbcL_55 y mtDNA_186, respectivamente.

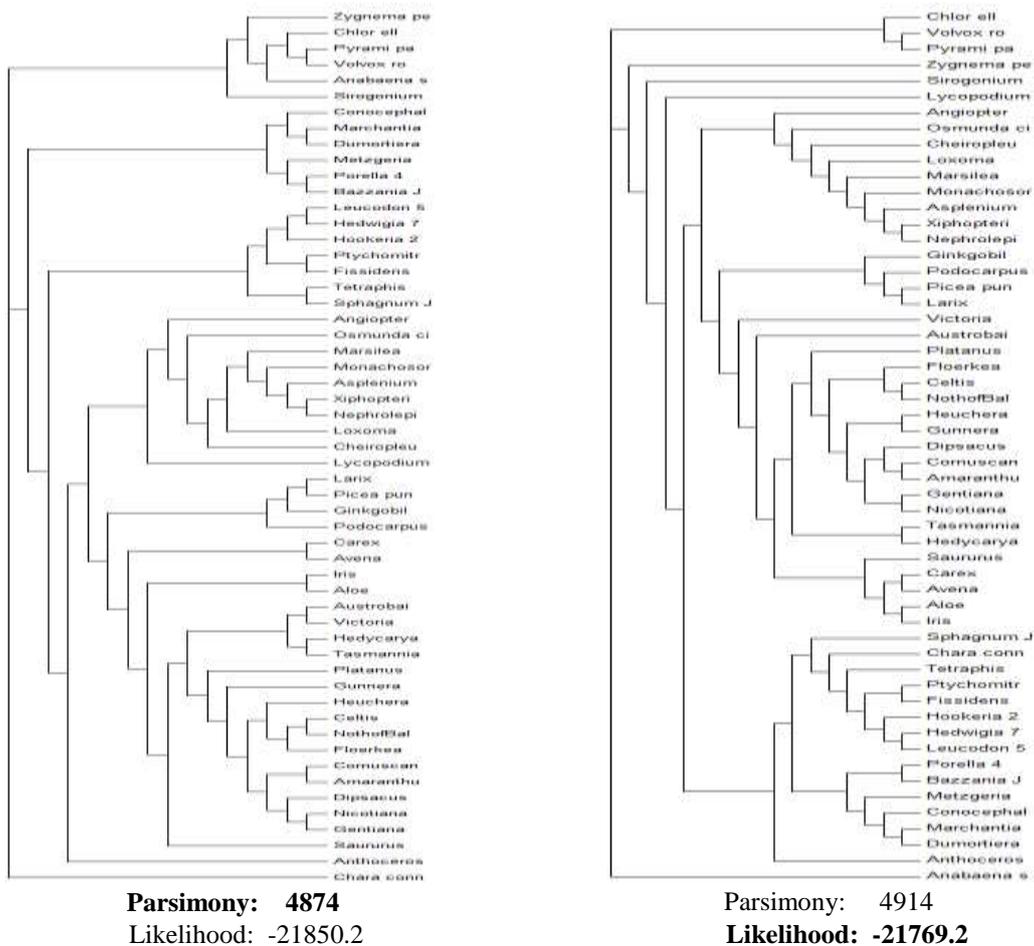
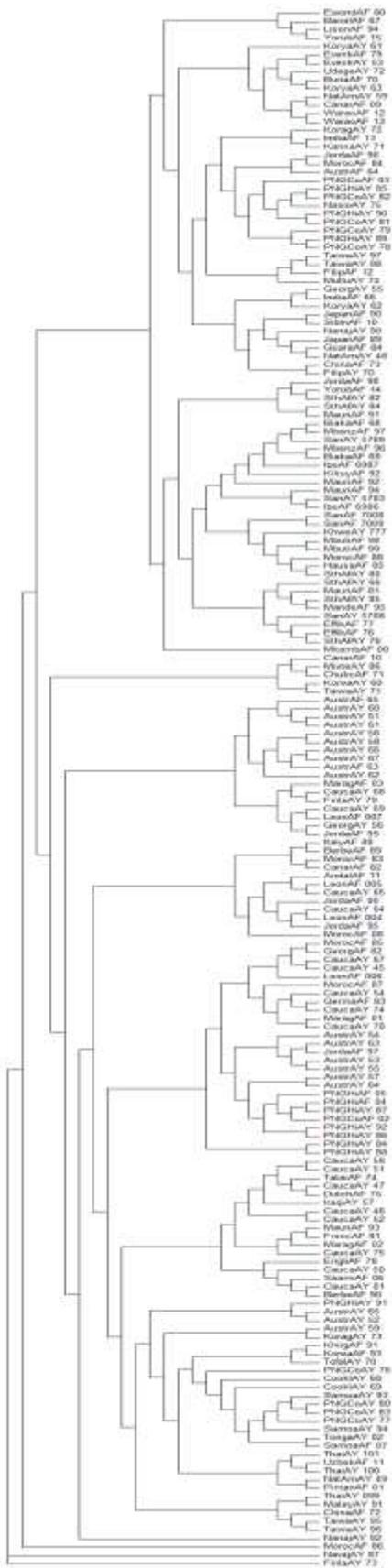
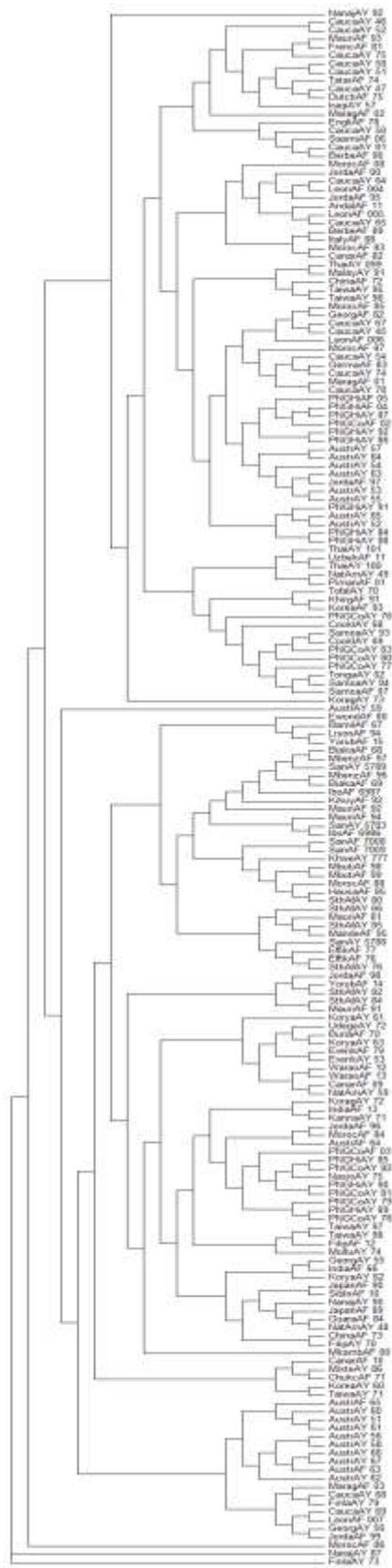


Ilustración 10.- Comparación de los árboles filogenéticos (Mejor Parsimonia (Izquierda), Mejor Verosimilitud (Derecha) del algoritmo PhyloMOVMO para el conjunto de datos rbcL_55.



Parsimony: 2433
Likelihood: -39897.3



Parsimony: 2445
Likelihood: -39865.5

Ilustración 11.- Comparación de los árboles filogenéticos (Mejor Parsimonia (Izquierda), Mejor Verosimilitud (Derecha) del algoritmo PhyloMOVMO para el conjunto de datos mtDNA_186.

4.3.4. Análisis de Pruebas Estadísticas

Mediante los resultados obtenidos en los análisis multiobjetivo y biológico se presume que nos encontramos frente a una superioridad del algoritmo PhyloMOVMO con los algoritmos de referencia, para validar dichos resultados se ha realizado un análisis estadístico utilizando técnicas no-paramétricas con el fin de determinar las diferencias significativas que existen entre los cuatros algoritmos, la prueba de Friedman es la primera prueba que se va aplicar para detectar diferencias significativas entre los algoritmos, si no existen se puede concluir que los algoritmos comparados tienen comportamientos parecidos, en caso que existan diferencias se procede aplicar la prueba de Wilcoxon según el caso. En la prueba de Wilcoxon se evaluarán los conjuntos de datos nucleótidos (ver apartado 3.3) de manera individual y en conjunto para cada algoritmo en cada de uno de los indicadores de calidad.

En la prueba individual los resultados se ilustrarán en una tabla con las siguientes simbologías: el caracter “-“ que indica que no hay diferencias significativas entre los algoritmos de la fila y columna, el símbolo “▲” significa que el algoritmo de la fila ha producido mejores resultados que el algoritmo de la columna con significancia estadística , y el símbolo “∇“ se utiliza cuando el algoritmo de la columna es estadísticamente mejor que el de la fila en el problema considerado [53] [23].

En la prueba en conjunto, si la hipótesis es rechazada se tendrá en cuenta los siguientes casos: En una comparación de tipo A vs B. Si se maximiza, el algoritmo A es mejor que el algoritmo B si los $R^+ > R^-$, en caso contrario el algoritmo B es mejor que el algoritmo A. Si se minimiza, el algoritmo A es mejor que el algoritmo B si los $R^- > R^+$, en caso contrario el algoritmo B es mejor que el algoritmo A. Los R^+ y R^- representan los valores acumulados de rangos positivos y negativos detectados por el test en el proceso de comparación.

Para realizar el análisis estadístico de las técnicas no-paramétricas se ha utilizado los resultados obtenidos por cada indicador en los cuatros algoritmos (ver Anexos 2 al 5). A continuación, se detallan los resultados de las técnicas utilizadas.

4.3.4.1. Prueba de Friedman

En la *Tabla 11* se puede visualizar que, los resultados en la prueba de Friedman (valor p) dio valores menores a 0.05 por tanto se rechaza la hipótesis nula y se comprueba que existen diferencias significativas entre los algoritmos que conforman los grupos para cada indicador, por lo que a continuación se realizará comparaciones por pares tomando como base el algoritmo propuesto, para dicha comparación se utilizará la prueba de Wilcoxon.

Tabla 11.- Resultado de la prueba de Friedman del indicador I_{HV} .

INDICADORES DE CALIDAD	VALOR DE LA DISTRIBUCIÓN	VALOR P	HIPÓTESIS
<i>Hipervolumen</i>	15,831	0,001	RECHAZADA
<i>Épsilon</i>	12,339	0,006	RECHAZADA
<i>Dispersión</i>	58,515	0,000	RECHAZADA
<i>Distancia Generacional Invertida Plus</i>	15,602	0,001	RECHAZADA

FUENTE: SPSS STATISTIC

ELABORADO: AUTOR

4.3.4.2. Prueba de Wilcoxon

En la prueba individual donde se evalúan de forma independiente los conjuntos de datos nucleótidos para cada algoritmo en cada de uno de los indicadores de calidad, los resultados obtenidos (ver tablas 12 al 15) confirman que PhyloMOVMO obtuvo un mejor rendimiento a un nivel de significancia de 95% (es decir, un nivel de significación del 5% o un p-value de 0.05) en los indicadores I_{HV} , I_{e+} y I_{IDG+} para los conjuntos de datos mtDNA_186, RDPII_218 y ZILLA_500, a excepción de la instancia rbcL_55 donde MOEAD reporta un mejor rendimiento general en los algoritmos. Además, estos resultados confirman el mejor rendimiento de NSGAI para los valores del indicador I_{Δ} , y aunque el SMSEMOA reporta el mejor rendimiento sobre la instancia de rbcL_55 en este indicador, la prueba de Wilcoxon indica que no son estadísticamente significativas con el resto de los algoritmos a excepción de MOEAD.

A continuación, se realizará una prueba en conjunto evaluando todos problemas de prueba para cada algoritmo en cada de uno de los indicadores de calidad.

Tabla 12.- Resultado de la prueba de Wilcoxon del indicador I_{HV} para los conjuntos de datos $rbcL_{55}$, $mtDNA_{186}$, $RDPII_{218}$ y $ZILLA_{500}$.

	NSGAI				MOEAD				SMSEMOA			
	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500
PhyloMOVMO	▲	▲	–	▲	▽	–	▲	▲	▲	▲	▲	▲
NSGAI					▽	–	▲	▽	▽	–	▲	–
MOEAD									▲	–	–	▲

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 13.- Resultado de la prueba de Wilcoxon del indicador I_{E+} para los conjuntos de datos $rbcL_{55}$, $mtDNA_{186}$, $RDPII_{218}$ y $ZILLA_{500}$.

	NSGAI				MOEAD				SMSEMOA			
	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500
PhyloMOVMO	▲	–	–	▲	▽	▲	▲	–	▲	▲	▲	▲
NSGAI					▽	▲	▲	▽	▽	▲	▲	–
MOEAD									▲	–	–	▲

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 14.- Resultado de la prueba de Wilcoxon del indicador I_{Δ} para los conjuntos de datos $rbcL_{55}$, $mtDNA_{186}$, $RDPII_{218}$ y $ZILLA_{500}$.

	NSGAI				MOEAD				SMSEMOA			
	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500
PhyloMOVMO	–	▽	▽	▽	–	–	▲	▲	–	–	–	–
NSGAI					–	▲	▲	▲	–	–	–	▲
MOEAD									▽	–	▽	▽

FUENTE: JMETALCPP

ELABORADO: AUTOR

Tabla 15.- Resultado de la prueba de Wilcoxon del indicador I_{IDG^+} para los conjuntos de datos $rbcL_55$, $mtDNA_186$, $RDPII_218$ y $ZILLA_500$.

	NSGAI				MOEAD				SMSEMOA			
	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500	rbcL_ 55	mtDNA_ 186	RDPII_ 218	ZILLA_ 500
PhyloMOVMO	▲	▲	–	▲	–	▲	▲	▲	▲	▲	▲	–
NSGAI					▽	▽	▲	–	▽	–	▲	–
MOEAD									▲	▲	–	–

FUENTE: JMETALCPP

ELABORADO: AUTOR

Para la prueba en conjunto, hay que considerar que en el caso de los indicadores I_{HV} y I_{IDG^+} se maximizan, y se minimizan para los indicadores I_{e^+} y I_{Δ} como ya se mencionó en el apartado 4.3.1. Esta información nos sirve para poder realizar la interpretación de los resultados mediante los rangos positivos y negativos.

En las tablas 16, 17, 18 y 19 se presentan los resultados de la prueba de Wilcoxon en los indicadores de calidad I_{HV} , I_{e^+} , I_{Δ} y I_{IDG^+} respectivamente, se puede apreciar comparaciones en donde el valor p es menor que el nivel de significancia determinado para el mismo ($\alpha = 0.05$) por lo que la hipótesis de igualdad es rechazada, es decir existe una diferencia significativa entre los dos algoritmos probados, para este caso específico necesitamos comparar los rangos positivos y negativos (ver apartado 4.3.4).

Para el caso en que se máxima, nos indica al algoritmo PhyloMOVMO como la mejor variante para el indicador de calidad de I_{HV} , y en el segundo indicador evaluado nos muestra que los algoritmos SMSEMOA y NSGAI superan al algoritmo PhyloMOVMO, mientras que para aquellos indicadores que se minimizan, nos indica que el indicador I_{e^+} el algoritmo PhyloMOVMO tiene mejor comportamiento frente a los otros algoritmos comparados, y en el indicador I_{Δ} el algoritmo propuesto solo supera al algoritmo MOEAD.

Tabla 16.- Resultado de la prueba de Wilcoxon del indicador I_{HV} .

COMPARACIÓN	R+	R-	VALOR P	HIPÓTESIS
<i>MOEAD vs PhyloMOVMO</i>	18	20	0,307	ACEPTADA
<i>SMSEMOA vs PhyloMOVMO</i>	11	26	0,003	RECHAZADA
<i>NSGAI vs PhyloMOVMO</i>	8	30	0,000	RECHAZADA

FUENTE: SPSS STATISTIC

ELABORADO: AUTOR

Tabla 17.- Resultado de la prueba de Wilcoxon del indicador I_{E+} .

COMPARACIÓN	R+	R-	VALOR P	HIPÓTESIS
<i>MOEAD vs PhyloMOVMO</i>	23	15	0,048	RECHAZADA
<i>SMSEMOA vs PhyloMOVMO</i>	28	11	0,001	RECHAZADA
<i>NSGAI vs PhyloMOVMO</i>	29	10	0,000	RECHAZADA

FUENTE: SPSS STATISTIC

ELABORADO: AUTOR

Tabla 18.- Resultado de la prueba de Wilcoxon del indicador I_{Δ} .

COMPARACIÓN	R+	R-	VALOR P	HIPÓTESIS
<i>MOEAD vs PhyloMOVMO</i>	32	8	0,002	RECHAZADA
<i>SMSEMOA vs PhyloMOVMO</i>	12	27	0,004	RECHAZADA
<i>NSGAI vs PhyloMOVMO</i>	7	33	0,000	RECHAZADA

FUENTE: SPSS STATISTIC

ELABORADO: AUTOR

Tabla 19.- Resultado de la prueba de Wilcoxon del indicador I_{DG+} .

COMPARACIÓN	R+	R-	VALOR P	HIPÓTESIS
<i>MOEAD vs PhyloMOVMO</i>	22	18	0,301	ACEPTADA
<i>SMSEMOA vs PhyloMOVMO</i>	25	14	0,049	RECHAZADA
<i>NSGAI vs PhyloMOVMO</i>	31	9	0,000	RECHAZADA

FUENTE: SPSS STATISTIC

ELABORADO: AUTOR

4.4. Discusión

PhyloMOVMO ofrece varias ventajas para aquellos investigadores que trabajan en análisis filogenéticos. No solo proporciona un árbol filogenético como resultado final, sino un conjunto de árboles filogenéticos optimizados con objetivos múltiples apoyados por criterios de parsimonia y de verosimilitud, cediendo a hipótesis alternativas que pueden ser útiles desde diferentes puntos de vista biológicos. Dentro de este conjunto de soluciones, los biólogos pueden encontrar los mejores árboles de parsimonia y verosimilitud (filogenias de alta calidad) representados por los puntos extremos de la aproximación de Frente de Pareto, en lugar de utilizar diferentes programas filogenéticos para cada criterio de optimización.

De hecho, los resultados experimentales indican que PhyloMOVMO proporciona resultados competitivos frente a otras herramientas filogenéticas.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

En el presente trabajo de proyecto de investigación se realizó una adaptación del algoritmo optimización multiobjetivo basada en mallas variables (MOVMO) para resolver problema de inferencia filogenética, del cual se puede concluir lo siguiente:

- Se realizó un nuevo enfoque apoyado en la técnica de optimización multiobjetivo basada en mallas variables para inferir filogenias, optimizando simultáneamente los dos criterios de reconstrucción de árboles filogenéticos como es la máxima parsimonia y la máxima verosimilitud.
- El estudio de otros algoritmos que resuelven problemas de inferencia filogenética sirvió como base para extraer funcionalidades de varios frameworks del área de la bioinformática útiles para el algoritmo propuesto.
- Se obtuvo una nueva versión del algoritmo MOVMO adaptada para resolver problemas de inferencias filogenética mediante el uso del operador de recombinación (cruce) propio para árboles filogenético.
- Los resultados obtenidos tanto multiobjetivo como biológico evaluando conjuntos de datos de secuencias nucleótidos (ADN). revelan que, PhyloMOVMO muestra un desempeño competitivo frente a los otros algoritmos de referencia.

6.2. Recomendaciones

Luego de la finalización del presente trabajo de investigación y de acuerdo con las experiencias obtenidas del mismo se recomienda:

- Utilizar los beneficios de la memoria compartida y los paradigmas de programación de memoria distribuida para acelerar el proceso de búsqueda y reconstrucción de los árboles filogenéticos, logrando así minimizar el tiempo para encontrar soluciones aceptables.
- Realizar un estudio descriptivo de los diferentes frameworks que existen en el área de la bioinformática, con el objetivo de buscar más variedad en las funcionalidades que requieren los algoritmos para inferir filogenias.
- Ampliar el estudio del operador de recombinación (cruce) con la finalidad de mejorar la funcionalidad del mismo, debido a que dicho operador es base en el algoritmo propuesto.
- Probar con otros tipos conjuntos de datos tales como las secuencias de aminoácidos (Proteínas) para ello se recomienda agregar nuevos modelos evolutivos que soporten dichos conjuntos de datos.

CAPITULO VI
BIBLIOGRAFÍA

6.1. Bibliografía

- [1] A. Stamatakis, «Distributed and Parallel Algorithms and Systems for Inference of Huge Phylogenetic Trees based on the Distributed and Parallel Algorithms and Systems for Inference of Huge Phylogenetic Trees based on the Maximum Likelihood Method.,» *Ph.D. thesis, Technische University at Munchen, Germany, 2004.*
- [2] C. Zambrano-Vega, A. Nebro, J. Durillo, J. Garca-Nieto y A.-M. J.F, «Multiple sequence alignment with multiobjective metaheuristics a comparative study.,» *International Journal of Intelligent Systems, 2017.*
- [3] C. Zambrano-Vega, A. Nebro, J. García-Nieto y J. Aldana-Montes, «Comparing multi-objective metaheuristics for solving a three-objective formulation of multiple sequence alignment.,» *Progress in Artificial Intelligence*, pp. 1-16, 2017.
- [4] S. V.-R. M. Santander-Jiménez, «Applying a multiobjective metaheuristic inspired by honey bees to phylogenetic inference.,» *Biosystems*, vol. 1, nº 114, pp. 39-55, 2013.
- [5] J. Handl, D. Kell y J. Knowles, «Multiobjective optimization in bioinformatics and computational biology.,» *IEEE/ACM transactions on computational biology and bioinformatics / IEEE*, p. 279–292, 2007.
- [6] S. S. Jiménez y M. A. V. Rodríguez, «A Multiobjective Proposal Based on the Firefly Algorithm for Inferring Phylogenies,» *Springer Berlin Heidelberg*, vol. 7833, nº 7833, pp. 141-152, 2013.
- [7] S. S. Jiménez, M. A. V. Rodríguez, J. A. G. Pulido y J. M. S. Pérez, «Inferring Phylogenetic Trees Using a Multiobjective Artificial Bee Colony Algorithm,» *Springer Berlin Heidelberg*, vol. 7246, nº 7246, pp. 144-155, 2012.
- [8] W. Cancino y A. C. B. Delbem, «A Multi-objective Evolutionary Approach for Phylogenetic Inference,» *Springer-Verlag Berlin Heidelberg*, vol. 4403, nº 4403, p. 428–442, 2007.
- [9] C. Zambrano-Vega, A. J. Nebro y J. A. Montes, «MO-Phylogenetics: A phylogenetic inference software tool with multi-objective evolutionary metaheuristics,» *Methods in Ecology and Evolution*, vol. 7, nº 7, pp. 800-805, 2016.
- [10] C. Zambrano-Vega, A. J. Nebro y J. F. Aldana, «Implementación de la principal referencia metaheurística de Optimización Multiobjetivo NSGA-II aplicada a la Filogenética Computacional,» *Revista Publicando*, vol. 2, nº 5, pp. 1-20, 2015.
- [11] W. Cancino, L. Jourdan, E.-G. Talbi y A. C. B. Delbem, «Parallel Multi-Objective Approaches for Inferring Phylogenies,» *Springer Berlin Heidelberg*, vol. 6023, nº 23, pp. 26-37, 2010.

- [12] Y. Salgueiro, J. Toro, R. Bello y R. Falcon, «Multiobjective variable mesh optimization.,» *Annals of Operations Research*, pp. 1-25, 2016.
- [13] C. Zambrano-Vega, A. J. Nebro y J. A. Montes, «MO-Phylogenetics: A phylogenetic inference software tool with multi-objective evolutionary metaheuristics,» *Methods in Ecology and Evolution*, vol. 7, nº 7, pp. 800-805, 2016.
- [14] C. A. Coello, G. B. Lamont y D. A. V. Veldhuizen, «Evolutionary Algorithms for Solving Multi-Objective Problems 2nd edn,» *Genetic and Evolutionary Computation Series, Springer*, 2007.
- [15] A. Puris-Cáceres, R. Bello y D. Molina, «Variable mesh optimization for continuous optimization problems,» *Springer-Verlag*, vol. 20, nº 20, p. 511–525, 2011.
- [16] Y. Salgueiro, J. L. Toro, R. Bello y R. Falcon, «Multiobjective variable mesh optimization,» *Springer*, vol. 21, nº 21, pp. 1-25, 2016.
- [17] S.-. S. Yamisleydi, L. T.-P. Jorge y B.-P. Rafael, «Evaluación del desempeño de la metaheurística MOVMO en funciones de prueba con restricciones.,» *Revista Cubana de Ciencias Informáticas*, vol. 10, nº 1, pp. 182-193, 2016.
- [18] P. P. Cruz, *Inteligencia Artificial con aplicaciones a la ingeniería*, Primera ed., México: Alfaomega Grupo Editor, 2010, p. 376.
- [19] A. G. Serrano, *Inteligencia Artificial: Fundamentos, prácticas y aplicaciones*, Madrid: Grupo RC, 2012.
- [20] N. Alancay, S. Villagra y A. Villagra, «Metaheurísticas de trayectoria y poblacional aplicadas a problemas de optimización combinatoria.,» *UNPA*, vol. 2, nº 2, p. 19, 2014.
- [21] G. Brassard y P. Bratley, *Fundamentos de algoritmia*, Madrid: Prentice Hall, 1998, p. 579.
- [22] C. Blum y A. Roli, «Metaheuristics in combinatorial optimization: Overview and conceptual comparison.,» *ACM Computing Surveys*, vol. 35, nº 3, pp. 268-308, 2003.
- [23] F. Luna-Valero, «Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones.,» pp. 1-213, 2008.
- [24] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller y E. Teller, «Equation of state calculations by fast computing machines,» *Journal of Chemical Physics*, vol. 21, nº 21, pp. 1087-1092, 1953.
- [25] S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi, «Optimization by simulated annealing,» *Science*, vol. 220, nº 220, pp. 671-680, 1983.
- [26] F. Glover, «Future paths for integer programming and links to artificial intelligence,» *Computers & Operations Research*, vol. 13, nº 13, pp. 533-549, 1986.

- [27] F. Glover, «Heuristics for integer programming using surrogate constraints,» *Decision Sciences*, vol. 8, nº 8, pp. 156-166, 1977.
- [28] T. Feo y M. Resende, «Greedy randomized adaptive search procedures,» *Journal of Global Optimization*, vol. 6, nº 6, pp. 109-133, 1999.
- [29] N. Mladenovic y P. Hansen, «Variable neighborhood search,» *Com Oper Res*, vol. 24, nº 24, pp. 1097-1100, 1997.
- [30] H. R. Lourenço, O. Martin y T. Stützle, «Handbook of Metaheuristics, Chapter Iterated local search,» *Kluwer Academic Publishers*, vol. 0, nº 0, pp. 321-353, 2002.
- [31] T. Stützle, «Local search algorithms for combinatorial problems analysis, algorithms and new applications.,» *Tech. rep., DISKI Dissertationen zur Künstliken Intelligenz. Sankt Augustin, Germany*, 1999.
- [32] J. L. Laura Lanzarini, «Metaheurísticas poblacionales aplicadas a la resolución de problemas complejos.,» *UNLP*, vol. 2, nº 2, pp. 98-102, 2009.
- [33] H. Mühlenbein, «The equation for response to selection and its use for prediction,» *Evolutionary Computation*, vol. 5, nº 5, pp. 303-346, 1998.
- [34] C. Zambrano-Vega, «Metaheurísticas de Optimización Multiobjetivo aplicadas a la Inferencia Filogenética y al Alineamiento Múltiple de Secuencias,» *PhD Tesis, Universidad de Málaga, Lenguajes y Ciencias de la Computación*, pp. 1-180, 2017.
- [35] M. Pelikan, D. E. Goldberg y E. Cantú-Paz, «BOA: The Bayesian optimization algorithm,» *Proceedings of the Genetic and Evolutionary Computation Conference GEC*, vol. 1, pp. 525-532, 1999.
- [36] P. Larrañaga, R. Etxeberria, J. A. Lozano y J. M. Peña, «Optimization by learning and simulation of Bayesian and Gaussian networks.,» *Tech. Rep. KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country*, 1999.
- [37] M. Soto, A. Ochoa, S. Acid y L. M. De Campos, «Introducing the polytree approximation of distribution algorithm. In: Second Symposium on Artificial Intelligence,» *Adaptive Systems*, pp. 360-367, 1999.
- [38] J. Whittaker, «Graphical models in applied multivariate statistics,» *John Wiley & Sons, Inc*, 1990.
- [39] F. Glover, «A template for Scatter Search and Path Relinking. In: et al. J.-K. H.,» *Artificial Evolution, Springer*, pp. 13-54, 1998.
- [40] M. Laguna y R. Martí, «Scatter Search,» *Methodology and Implementations in C. Kluwer*, 2003.

- [41] M. Dorigo, «Optimization, learning and natural algorithms,» *PhD thesis, Dipartimento di Elettronica, Politecnico di Milano*, 1992.
- [42] M. Dorigo y T. Stützle, «Handbook of Metaheuristics,» *International Series In Operations Research and Management Science, vol 57, Kluwer Academic Publisher, chap The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*, pp. 251-285, 2003.
- [43] J. L. Cohon y D. H. Marks, «A review and evaluation of multiobjective programming techniques,» *Water Resources Research*, vol. 2, nº 11, pp. 208-220, 1975.
- [44] K. Deb, «Multi-Objective Optimization Using Evolutionary Algorithms,» *John Wiley & Sons*, 2001.
- [45] V. Pareto, «Cours D'Economie Politique, vol I and II,» *F. Rouge, Lausanne*, 1896.
- [46] A. Osyczka, «Multicriteria optimization for engineering design,» *Gero J. S. (ed) Design Optimization, Academic Press*, pp. 193-227, 1895.
- [47] V. D. A. Van, «Multiobjective evolutionary algorithms: Classifications, analyses, and new innovation,» *PhD thesis, DTIC Document, Wright-Patterson, AFB, OH*, 1999.
- [48] D. E. Goldberg, «Genetic Algorithms in Search, Optimization and Machine Learning,» *Addison-Wesley*, 1989.
- [49] R. L. Graham, « Bounds on multiprocessor timing anomalies.,» *SIAM Journal of Applied Mathematics*, vol. 17, pp. 416-429, 1969.
- [50] R. M. Karp, «Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane.,» *Mathematics of Operations Research*, vol. 2, pp. 209-224, 1977.
- [51] J. D. Knowles, L. Thiele y E. Zitzler, « A tutorial on the performance assessment of stochastic multiobjective optimizers.,» *Tech. Rep. TIK-Report 214, Computer Engineering and Networks Laboratory, ETHC Zurich*, 2006.
- [52] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca y F. V. G. Da, «Performance assessment of multiobjective optimizers: an analysis and review.,» *IEEE Transactions on Evolutionary Computation*, vol. 7, nº 2, pp. 114-132, 2003.
- [53] C. Zambrano-Vega, M. Cardenas-Zea y R. Aguirre-Pérez, «Un enfoque Multi-Objetivo a la optimizacion del Alineamiento Multiple de Secuencias (MSA),» *LATIN AMERICAN JOURNAL OF COMPUTING - LAJC*, pp. 1-9.
- [54] I. Hisao, M. Hiroyuki y N. Yusuke, «Sensitivity of Performance Evaluation Results by Inverted Generational Distance to Reference Points,» *Inverted Generational Distance to Reference Points*, pp. 1107-1114, 2016.

- [55] K. Ji-Hyon, «MedCalc Software BVBA,» [En línea]. Available: https://www.medcalc.org/manual/friedman_test.php.
- [56] E. O. W. & B. S. Lieberman, *Phylogenetics: Theory and Practice of Phylogenetic Systematics*, Second Edition, New Jersey: Wiley-Blackwell, 2011.
- [57] Y. Ziheng y R. Bruce, «Molecular phylogenetics: principles and practice,» *NATURE REVIEWS GENETICS*, vol. 13, nº 13, pp. 303-314, 2012.
- [58] S. S. Jiménez, M. A. V. Rodríguez, J. A. G. Pulido y J. M. S. Pérez, «Una adaptación multiobjetivo y paralela del algoritmo Artificial Bee Colony aplicada a la inferencia filogenética,» *Spring*, vol. 4, nº 4, pp. 1-8, 2013.
- [59] E. López-Camacho, G. M. J. García, A. J. Nebro y J. F. Aldana-Montes, «MetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework,» *JBioinformatics (Oxford, England)*, 2014.
- [60] J. Dutheil, S. Gaillard, E. Bazin, S. Glémin, V. Ranwez, N. Galtier y K. Belkhir, «Bio++: A set C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics,» *BMC bioinformatics*, pp. 7-188, 2006.
- [61] T. Flouri, F. Izquierdo-Carrasco, D. Darriba, A. Aberer, L.-T. Nguyen, B. Minh, H. A. Von y A. Stamatakis, «The Phylogenetic Likelihood Library,» *Systematic Biology*, vol. 2, nº 64, pp. 356-362, 2015.
- [62] A. Jon, «TreeBASE: A Database of Phylogenetic Knowledge,» The Phyloinformatics Research Foundation, Inc., [En línea]. Available: <https://treebase.org/treebase-web/home.html>.
- [63] Z. Jian-Qiu, X. Feng y F. Xian-Wen, «Decomposition of Multi-Objective Evolutionary Algorithm based on Estimation of Distribution,» *Natural Sciences Publishing Cor, Applied Mathematics & Information Sciences*, vol. 8, nº 1, pp. 249-254, 2014.
- [64] K. J.D., C. D.W. y F. M., «Bounded Archiving using the Lebesgue Measure,» *IEEE Press, Piscataway, NJ, Congress on Evolutionary Computation (CEC'2003)*, vol. 4, p. 2490–2497, 2003.
- [65] K. J. y C. D., «Properties of an adaptive archiving algorithm for storing nondominated vectors,» *IEEE Transactions on Evolutionary Computation*, vol. 7, nº 2, p. 100–116, 2003.
- [66] B. Nicola, N. Boris y E. Michael, «SMS-EMOA: Multiobjective selection based on dominated hypervolume,» *Science Direct*, p. 1653–1669, 2007.

- [67] K. Deb, A. Pratap, S. Agarwal y T. Meyarivan, «A fast and elitist multiobjective genetic algorithm: NSGA-II,» *IEEE Transactions on Evolutionary Computation*, vol. 2, nº 6, pp. 182-197, 2002b.
- [68] J. Knowles y D. Corne, «The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation,» *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol 1, pp. 105 Vol. 1*, 1999b.
- [69] Q. Zhang y H. Li, «MOEA/D: A multiobjective evolutionary algorithm based on decomposition,» *IEEE Trans Evolutionary Computation*, vol. 6, nº 11, pp. 712-731, 2007.
- [70] N. Beume, B. Naujoks y M. Emmerich, «Sms-emoa: Multiobjective selection based on dominated hypervolume.,» *European Journal of Operational Research* , vol. 3, nº 181, pp. 1653-1669, 2007.
- [71] C. Lanave, G. Preparata, C. Sacone y G. Serio, «A new method for calculating evolutionary substitution rates.,» *Journal of molecular evolution*, vol. 1, nº 20, p. 86–93, 1984.
- [72] NCBI, «GenBank,» NCBI, 15 11 2016. [En línea]. Available: <https://www.ncbi.nlm.nih.gov/genbank/>.
- [73] C. Cotta y P. Moscato, «Inferring phylogenetic trees using evolutionary algorithms,» *International Conference on Parallel Problem Solving from Nature*, pp. 720-729, 2002.
- [74] S. M. Tuffley C., «Links between maximum likelihood and maximum parsimony under a simple model of site substitution.,» *Bulletin of Mathematical Biology*, pp. 581-607, 1997.
- [75] S. Guindon, J. Dufayard, V. Lefort, M. Anisimova, W. Hordijk y O. Gascuel, «New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0.,» *Systematic Biology*, vol. 3, nº 59, p. 307–321, 2010.
- [76] A. Goeffon, J. Richer y J. Hao, «Progressive tree neighborhood applied to the maximum parsimony problem.,» *IEEE/ACM transactions on computational biology and bioinformatics / IEEE*, p. 136–145, 2008.
- [77] D. Darriba, G. Taboada, R. Doallo y D. Posada, «jModelTest 2: more models, new heuristics and parallel computing.,» *Nature methods*, p. 772–772, 2012.
- [78] J. Felsenstein, «Inferring Phylogenies,» *Palgrave Macmillan*, 2004.
- [79] T. Flouri, F. Izquierdo-Carrasco, D. Darriba, A. Aberer, L. Nguyen, B. Minh, A. Von Haeseler y A. Stamatakis, «The Phylogenetic Likelihood Library,» *Systematic Biology*, vol. 2, nº 64, p. 356–362, 2015.

- [80] S. S. Jiménez, M. A. V. Rodríguez, J. A. G. Pulido y J. M. S. Pérez, «Inferring Phylogenetic Trees Using a Multiobjective Artificial Bee Colony Algorithm,» *Springer Berlin Heidelberg*, vol. 7246, nº 7246, pp. 144-155, 2012.
- [81] P. P. Cruz, *Inteligencia Artificial con aplicaciones a la ingeniería*, Primera ed., México: Alfaomega Grupo Editor, 2010, p. 376.
- [82] J. Kennedy, «Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance,» *Proceedings of IEEE Congress on Evolutionary Computation (CEC 1999)*, pp. 1931-1938, 1999.
- [83] C. Zambrano-Vega, A. J. Nebro, I. Navas-Delgado, J. Garcia-Nieto y J. Aldana-Montes, «MORPHY: Multi-Objective Phylogenetic Inference of Protein Coded Sequences,» *Submitted*, p. Submitted, Submitted.
- [84] P. W. H., S. A. Teukolsky, W. T. Vetterling y B. P. Flannery, «Numerical Recipes,» *The Art of Scientific Computing. Second Edition. Cambridge University Press*, 1992.
- [85] W. Cancino y A. Delbem, «A multi-criterion evolutionary approach applied to phylogenetic reconstruction.,» *Korosec, P. (ed.) New Achievements in Evolutionary Computation, chap. 06. InTech, Rijeka*, 2010.
- [86] Biopp, «bpp-phyl,» Biopp, [En línea]. Available: http://biopp.univ-montp2.fr/apidoc/bpp-phyl/html/classbpp_1_1TreeTemplate.html#details.

CAPITULO VII

ANEXOS

ANEXO 1. Lista de parámetros de los algoritmos

Parámetros	rbcL_55	mtDNA_186	RDPII_218	ZILLA_500	Descripción
Parámetros de la Metaheurísticos					
experimentid	55	186	218	500	Experiment ID, los archivos resultados serán nombrados con este ID
Datapath*			data		Nombre del directorio con los datos
populationsize			100		Tamaño de la población
maxevaluations	2500	1500	1000	2000	Número de evaluaciones
neighborhoodSizeValue			4		Parámetro PhyloMOVMO
archiveSizeValue			100		Parámetro PhyloMOVMO
betaValue			0		Parámetro PhyloMOVMO
offset			100		Parámetro SMSEMOA
intervalupdateparameters	0	200	100	200	#0 No aplica, cada 500 evaluaciones se actualizan Modelo y Newton
printtrace			true		Imprime los tiempos transcurridos y los scores obtenidos durante las estrategias de optimización aplicadas
printbestscores			true		Imprime los mejores scores encontrados durante la ejecución del algoritmo.
Operadores Genéticos					
selection.method			binarytournament		Operadores de Selección disponibles:binarytournament y randomselection (solo para SMSEMOA)
crossover.method			PDG		Operador de cruce, disponible solo el operador PDG (Prune-Delete-Graf)
crossover.probability	80%	80%	0%	0%	Probabilidad de ejecución del operador de cruce
crossover.offspringsize			2		Número de descendiente generado por el cruce, (solo 1 para MOEAD)
mutation.method			NNI		Operadores de mutación topológica
mutation.probability			20%		Probabilidad de ejecución del operador de mutación

Parámetros Filogenéticos					
alphabet			DNA		Para secuencia de ADN
input.sequence.file*			Requerido		El fichero con el alineamiento múltiple de secuencias.
input.sequence.format			Phylip		El formato del MSA. Opciones: split=spaces, type=extended o classic, order=sequential, interleaved
input.sequence.print			false		Imprime las secuencias leídas
partitionmodelfilepll*			Requerido		Nombre del fichero con información del Modelo Evolutivo, define esquema de particionamiento del MSA y la asignación de un modelo de sustitución para cada partición definida. Basado en el formato RAxML
init.population	stepwise	user	user	user	Método para generar o crear los árboles filogenéticos iniciales, los valores son: user, random o stepwise
init.population.tree.file			Requerido		Nombre del fichero con el conjunto de árboles filogenéticos definidos por el usuario para la generación de la población inicial. Si los árboles filogenéticos iniciales son definidos por el usuario, este parámetro es obligatorio (bootstrap)
init.population.tree.format			Newick		Formato de los árboles filogenéticos: Newick
Parámetros del Modelo Evolutivo					
model			GTR+G		Modelo Evolutivo, sólo GTR+G
model.frequencies			user		Las frecuencias de las bases pueden ser definida por el usuario o estimada empíricamente por el software: none o user

rate distribution.ncat	4				Número de categorías, 4 por defecto
Optimización Filogenética					
opt.method	h2	h2	h2	h2	h1 o h2
opt.method.perc	50%	50%	50%	50%	Si la estrategia h2 es seleccionada, representa al porcentaje de aplicar una de las siguientes técnicas pllRearrangeSearch o PPN
opt.ppn.numiterations	100	1000	1000	1000	Número de iteraciones de la técnica PPN
opt.ppn.maxsprbestmoves	20	100	100	100	Máximo número de mejores movimientos a ser aplicados en la técnica PPN
opt.pll.percnodes	0.1	0.3	0.4	0.4	Porcentaje de nodos a ser usados como 'nodo raíz' por la función
opt.pll.mintranv-maxtranv	1-20	1-20	1-30	1-30	Define el tamaño del anillo en el que los movimientos topológicos tienen lugar
opt.pll.newton3sprbranch	true	false	false	False	Optimización de las longitudes de las ramas afectadas por los movimientos topológicos
Optimización de las longitudes de las ramas					
bl_opt.starting	true	true	true	true	Optimización de los árboles de la población inicial
bl_opt	false	false	false	false	Optimización de los árboles generados durante la exploración
bl_opt.final	true	true	true	true	Optimización a árboles obtenidos al final del algoritmo
model_opt	false	false	false	false	Optimización del modelo evolutivo

ANEXO 2. Resultados obtenidos de los algoritmos en el indicador I_{HV}

MOEAD	PhyloMOVMO	NSGAI	SMSEMOA
0,637420886	0,590427215	0	0,697310127
0,662974684	0,723022152	0	0,567879747
0,725474684	0,723971519	0	0,593670886
0,756170886	0,554588608	0	0,625791139
0,657832278	0,646360759	0	0,626582278
0,628797468	0,55039557	0	0,285917722
0,693670886	0,624683544	0	0,554746835
0,685917722	0,642721519	0	0,508781646
0,679588608	0,721993671	0	0,476661392
0,701582278	0,510917722	0	0,617800633
0,307391092	0,392070485	0,310817425	0,420949584
0,276064611	0,266764562	0,29319628	0,049926579
0,049926579	0,341164953	0,348507097	0,049926579
0,239353891	0,48556045	0,239843368	0,239843368
0,274106706	0,332843857	0,239843368	0,239843368
0,389133627	0,420949584	0,271169848	0,389133627
0,307880568	0,27361723	0,129221733	0,27361723
0,318159569	0,199706314	0,192853647	0,239843368
0,06020558	0,28144885	0,205580029	0,174743025
0,174743025	0,277043563	0,326970142	0,174743025
0,52217534	0,608793926	0,554958956	0,623931916
0,593799685	0,686415602	0,643470019	0,603193618
0,576424218	0,602588043	0,619665515	0,61145033
0,608214363	0,604109965	0,608578986	0,580728953
0,650375484	0,636664038	0,601358184	0,594734743
0,580282188	0,629856232	0,576608126	0,595010377
0,541043979	0,564759039	0,648439653	0,612993701
0,663099847	0,635846262	0,619862657	0,542958817
0,64444341	0,626280286	0,607714662	0,638573856
0,562466344	0,569868316	0,538942495	0,55862891
0,091167355	0,354597107	0,031508264	0
0	0,014376722	0	0
0	0,031766529	0	0,116821625
0,005767906	0,009297521	0,041752755	0
0	0,175103306	0	0,044249311
0,085743802	0,080061983	0	0,001463499
0,060433884	0,017045455	0	0
0	0	0,004476584	0
0	0	0	0
0,091167355	0	0	0,027117769

ANEXO 3. Resultados obtenidos de los algoritmos en el indicador I_{E^+}

MOEAD	PhyloMOVMO	NSGAI	SMSEMOA
0,175	0,202531646	2,626582278	0,139240506
0,175	0,14556962	1,680379747	0,256329114
0,1	0,14556962	1	0,237341772
0,151898734	0,3	1	0,300632911
0,224683544	0,25	1	0,205696203
0,275	0,313291139	1	0,425
0,167721519	0,180379747	1,025	0,243670886
0,14556962	0,21835443	1	0,306962025
0,180379747	0,1	1,436708861	0,275
0,17721519	0,275316456	1,417721519	0,221518987
0,333333333	0,224669604	0,444444444	0,222222222
0,333333333	0,36123348	0,333333333	0,731277533
0,731277533	0,31277533	0,321585903	0,731277533
0,444444444	0,220264317	0,444444444	0,444444444
0,333333333	0,333333333	0,444444444	0,444444444
0,333333333	0,222222222	0,333333333	0,333333333
0,333333333	0,333333333	0,49339207	0,333333333
0,444444444	0,431718062	0,286343612	0,444444444
0,585903084	0,436123348	0,343612335	0,475770925
0,475770925	0,333333333	0,333333333	0,475770925
0,196115288	0,143481428	0,152255639	0,156591406
0,159774436	0,110706482	0,097744361	0,115076475
0,157319738	0,098370927	0,121631464	0,142753095
0,159774436	0,118718135	0,142024763	0,149308084
0,109649123	0,107142857	0,141296431	0,152221413
0,154761905	0,11716792	0,14420976	0,113619811
0,19227968	0,134741442	0,126729789	0,15726817
0,115914787	0,115288221	0,130371449	0,219227968
0,104151493	0,154135338	0,112163146	0,101966497
0,162907268	0,178571429	0,184837093	0,17481203
0,75	0,033057851	0,619834711	0,916666667
1,166666667	0,625	0,832644628	0,623966942
0,791666667	0,833333333	0,958333333	0,708333333
0,698347107	0,828512397	0,791666667	1
1,083333333	0,497933884	1,25	0,75
0,833333333	0,601239669	0,948347107	0,958333333
0,666666667	0,708333333	1,208333333	0,958333333
1,208333333	1,039256198	0,958333333	1
1	0,950413223	0,916666667	1
0,75	0,950413223	0,916666667	0,694214876

ANEXO 4. Resultados obtenidos de los algoritmos en el indicador I_{Δ}

MOEAD	PhyloMOVMO	NSGAI	SMSEMOA
1,216136036	0,989117522	0,968099493	1,047237441
1,380161868	0,915075472	0,783878428	0,871415505
1,137615847	1,110696764	1,017441561	0,954227995
0,96974263	0,993222218	0,831550942	0,880396106
1,095580234	0,669354883	1	1,073786027
1,218604229	1,040838577	1,186275693	0,99666675
0,93714835	0,83900425	0,865023151	0,657368889
0,926838457	1,361223409	1,068640927	0,85080983
1,144370957	0,953490403	1,220484562	0,453827832
1,295162522	1,254157295	1,235742037	0,558186427
1,37199232	1,546780375	0,733569302	1,610125613
1,380851651	1,508218427	0,748799496	1,08800942
1,08800942	1,480546708	0,84591924	1,08800942
1,375845341	1,21680192	0,518308057	0,518308057
1,301138293	1,110548272	0,518308057	0,518308057
1,226956808	1,610125613	0,479650399	1,226956808
1,326771061	1,179360272	0,955141675	1,179360272
1,226868373	1,399275751	1,22728448	0,518308057
1,224767115	1,117917737	0,848053526	1,292968628
1,292968628	1,127307953	1,271701775	1,292968628
1,056108387	0,787220735	0,956395957	0,751602972
1,148858263	0,793909647	0,731768854	0,698014171
1,100854811	0,863103795	0,786148959	0,81721034
1,171619035	0,787736315	0,769612406	0,891083266
1,234574232	0,91496632	0,802087699	0,949088326
1,113430599	0,93408528	0,833921703	0,863686886
1,179484086	0,860434645	0,795771812	0,979830424
1,258123444	0,956700196	0,843032565	1,001109202
1,094847438	1,005010994	0,762039984	0,931248255
1,115398596	1,005492743	0,810964064	1,052936215
1,098706729	0,988163255	0,837964157	0,743806536
1,173332604	1,326151435	0,804517892	1,117337207
1,16880571	0,923230519	0,892380394	0,889184444
1,161041076	1,115703213	0,753387102	0,914408258
1,210878644	1,178885966	0,812195764	1,085213439
1,082185144	1,005894364	0,716832222	0,864403283
1,155812114	1,073748907	0,915907951	0,929404563
1,148400305	0,823606496	0,892133564	1,121385582
1,260406083	1,098594351	0,85043596	1,121385582
1,098706729	1,098594351	0,977348133	1,018973903

ANEXO 5. Resultados obtenidos de los algoritmos en el indicador I_{IDG^+}

MOEAD	PhyloMOVMO	NSGAI	SMSEMOA
0,106983921	0,134998374	2,298213414	0,07216502
0,091150638	0,047346641	1,417308439	0,152264383
0,054716205	0,048831548	0,673539435	0,144261845
0,042024144	0,168724062	0,681102165	0,130525803
0,103743167	0,094814995	0,697921745	0,112511883
0,106221208	0,172574852	0,820107712	0,343419823
0,082090859	0,123736894	1,04945019	0,162540167
0,087408036	0,095307055	0,713594947	0,189024137
0,086846615	0,045029219	1,14288382	0,211957618
0,067997113	0,187118857	1,094662022	0,119700573
0,208262423	0,143542917	0,19747227	0,103008167
0,216225555	0,2451968	0,187567333	0,603466483
0,603466483	0,186274532	0,138876232	0,603466483
0,298496172	0,07586882	0,236575689	0,236575689
0,227184202	0,173901564	0,236575689	0,236575689
0,151827505	0,103008167	0,229582069	0,151827505
0,223863108	0,234695685	0,367307845	0,234695685
0,203257541	0,303183749	0,27846827	0,236575689
0,504481545	0,275832025	0,298728045	0,371735171
0,371735171	0,194046323	0,208443085	0,371735171
0,124434766	0,073835007	0,106538659	0,064404497
0,082427497	0,026023633	0,052204695	0,075174966
0,09326633	0,079774712	0,064414485	0,065825277
0,070020702	0,073090869	0,076914818	0,094646159
0,050960674	0,060975459	0,080912384	0,080285829
0,088885448	0,059495406	0,093421206	0,0852382
0,107767935	0,099595384	0,050292699	0,072429216
0,042464749	0,056831846	0,069192677	0,112344374
0,055197692	0,064316048	0,074180569	0,053587309
0,103077868	0,096067664	0,115299841	0,105399532
0,469808958	0,003005259	0,506701677	0,735070791
0,98025917	0,553143167	0,879086654	0,558458808
0,814931396	0,544306548	0,97501274	0,38137923
0,579822395	0,5856684	0,465767752	0,569953515
1,078246211	0,289449829	1,44538479	0,543102477
0,409561655	0,396743861	0,815696454	0,627946156
0,478970985	0,558415036	0,83526266	0,698835138
0,844763111	1,15898228	0,670203249	0,796961889
0,921653691	1,003824194	0,938438082	0,796961889
0,469808958	1,003824194	0,78555605	0,556882977

ANEXO 6. PhyloMOVMO

Requerimientos

La adaptación de MOVMO [17] a la inferencia filogenética se ha desarrollado en la máquina Unix (CentOS 7) utilizando el compilador g++ (4.4.7). La utilidad make se ha utilizado para compilar el paquete de software.

PhyloMOVMO requiere instalar dos frameworks:

1. Bio ++: un conjunto de bibliotecas de c++ para bioinformática, incluyendo análisis de secuencias, filogenéticos, evolución molecular y genética poblacional. Página de Inicio: <http://biopp.Univ-montp2.fr/>
2. PLL- Librería filogenética verosimilitud: una biblioteca de software altamente optimizada y paralelizada para facilitar el desarrollo de nuevas herramientas de software que se ocupan de la inferencia filogenética. Página de Inicio: Librería filogenética verosimilitud <http://www.libpll.org/>

Instalación

Copie en la ubicación donde desea instalar PhyloMOVMO. A continuación, ejecute el `install.sh` con el siguiente comando:

```
% sh install.sh
```

Esto instala todas las bibliotecas necesarias: Bio ++ (BPP-Core, BPP-SEQ y BPP-Phyl) y PLL. Agregue la ruta de las nuevas bibliotecas instaladas en el archivo `.bashrc`

Para bio ++:

```
export CPATH=$CPATH: PhyloMOVMO _path/lib/Bpp/include
export LIBRARY_PATH=$LIBRARY_PATH: PhyloMOVMO _path/lib/Bpp/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: PhyloMOVMO _path/lib/Bpp/lib
```

Para PLL:

```
export CPATH=$CPATH: PhyloMOVMO _path/lib/Pll/src/include
export LIBRARY_PATH=$LIBRARY_PATH: PhyloMOVMO _path/lib/Pll/src/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: PhyloMOVMO _path/lib/Pll/src/lib
```

Ejecución

El main binary se encuentra en la subcarpeta ' bin '. Introduzca esta carpeta para ejecutar PhyloMOVMO.

```
% cd bin
% ./ PhyloMOVMO param= parametersfile
```

El parameterfile es el nombre de archivo que tiene todos los parámetros requeridos para ejecutar el programa. Por ejemplo, para ejecutar el programa utilizando uno de los problemas de prueba ZILLA_500, escriba el siguiente comando:

```
%./ PhyloMOVMO param = parameters /params500.txt
```

Estructura

- PhyloMOVMO
 - Lib
 - bpp: código fuente de las bibliotecas bio ++: bpp-core, bpp-seq y bpp-phyl.
 - pll: código fuente de la biblioteca de probabilidad filogenética.
 - libjmetal: biblioteca jMetalCpp.
 - src: código fuente de PhyloMOVMO basado en el Framework jMetalCpp.
 - bin: el main binary del software.
 - data
 - sequences: archivo de secuencias.
 - model: archivo de modelo de partición utilizado por PLL.
 - inputusertrees: entrada de árboles filogenéticos utilizados como repositorio para generar la población inicial.
 - parameters: archivo de parámetros que contiene todos los parámetros necesarios para personalizar el software.

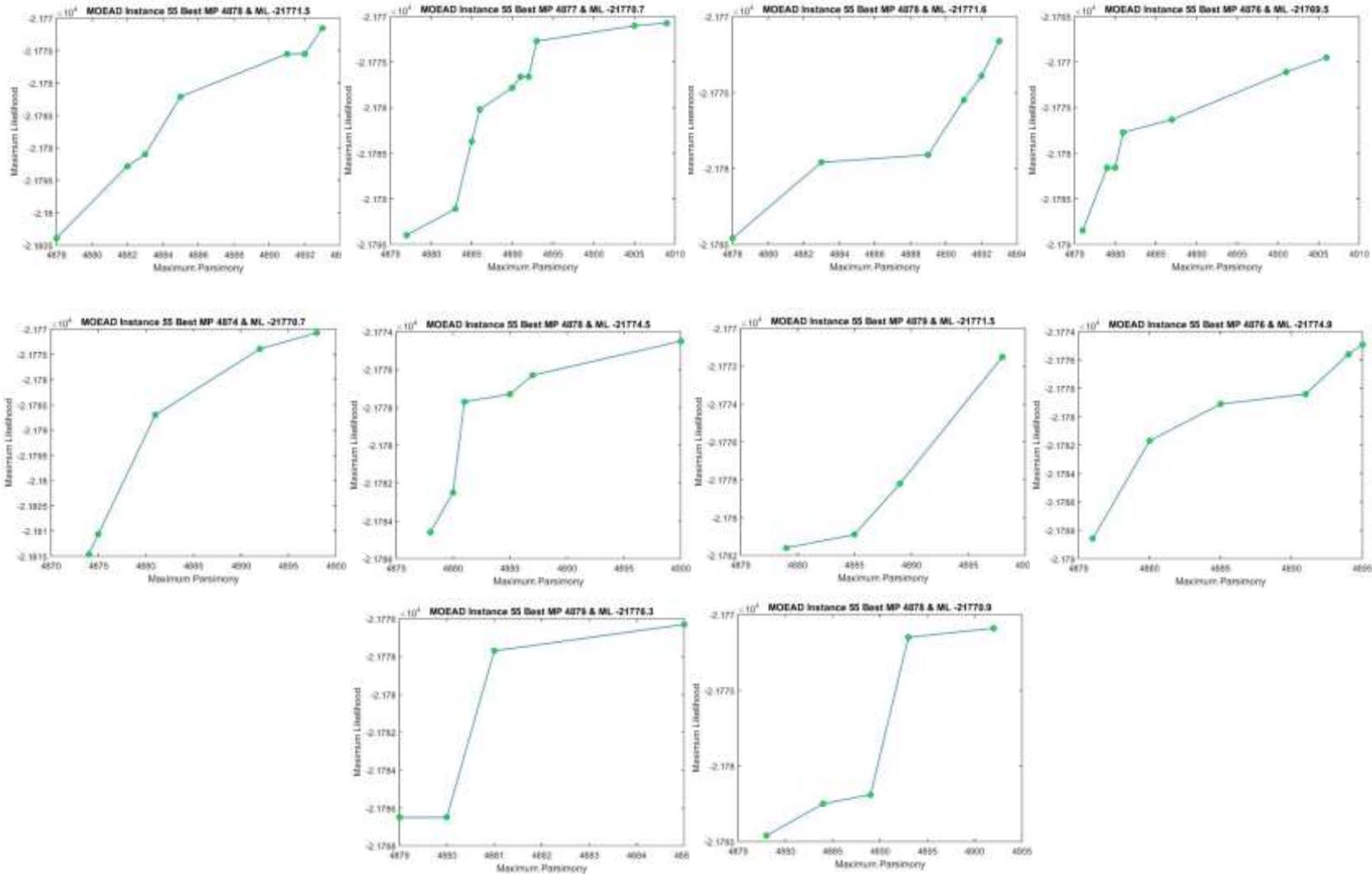
Formato de Resultados

Los resultados de PhyloMOVMO se basan en el formato de salida de la herramienta de software MO-Phylogenetics [13], crea dos archivos llamados FUN+ExperimentID y VAR+ExperimentID. El archivo FUN contiene los datos de la aproximación frontal de Pareto y el archivo VAR contiene los árboles filogenéticos optimizados en formato Newick.

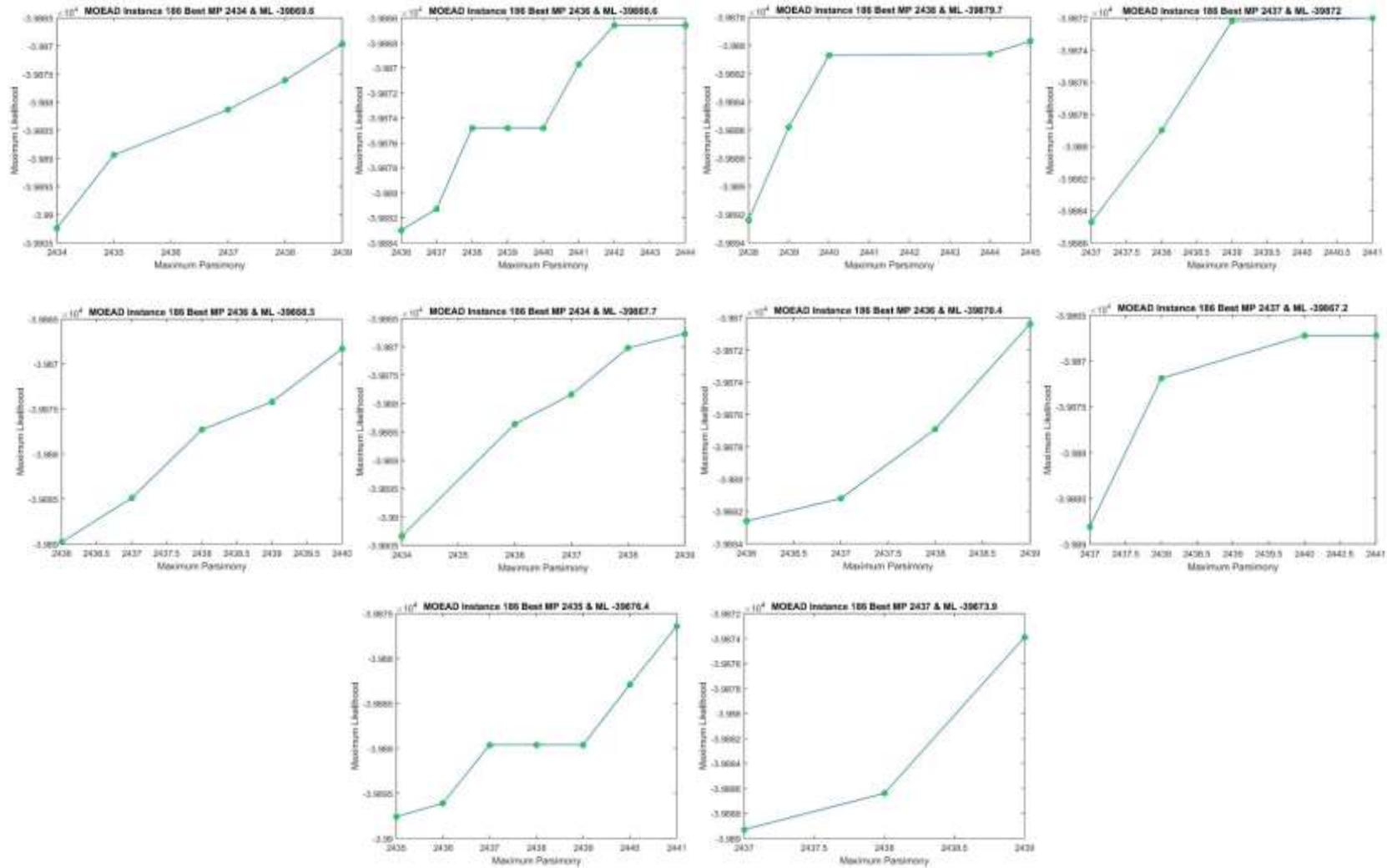
ANEXO 7. Matriz de Relación

Problema	Objetivos	Resultados
¿Cómo obtener árboles filogenéticos inferidos mediante la metaheurística de optimización multiobjetivo basada en mallas variables?	Realizar un estudio experimental para determinar cómo obtener árboles filogenéticos inferidos mediante la metaheurística de optimización multiobjetivo basada en mallas variables.	<ul style="list-style-type: none"> ▪ Árboles filogenéticos inferidos empleando dos criterios de reconstrucción filogenética la Máxima Parsimonia (MP) y la Máxima Verosimilitud (MV) simultáneamente. ▪ Aproximaciones de Frentes de Pareto compuesto por un conjunto de soluciones no-dominadas entre ellas, las mismas que ofrecerán diversos grupos de hipótesis filogenéticas útiles desde varios enfoques biológicos.
¿Cómo aplicar el algoritmo MOVMO a la inferencia de árboles filogenético?	Integrar al algoritmo MOVMO las funcionalidades de varios frameworks del área de la bioinformática.	<ul style="list-style-type: none"> ▪ Representación de árboles basado a la Inferencia Filogenética.
¿Cómo obtener soluciones filogenéticas del algoritmo MOVMO?	Adaptar el algoritmo optimización multiobjetivo basada en mallas variables (MOVMO) para resolver problema de inferencia filogenética.	<ul style="list-style-type: none"> ▪ Mejora en la exploración del espacio de búsqueda de árboles filogenéticos.
¿Cómo evaluar el rendimiento del algoritmo MOVMO resolviendo problemas filogenéticos?	Realizar experimentos del algoritmo implementado, basado en criterios de rendimiento y reconstrucción de árboles, mediante análisis multiobjetivo y biológico, respectivamente.	<ul style="list-style-type: none"> ▪ Demostración del rendimiento competitivo frente a resultados de otros algoritmos.

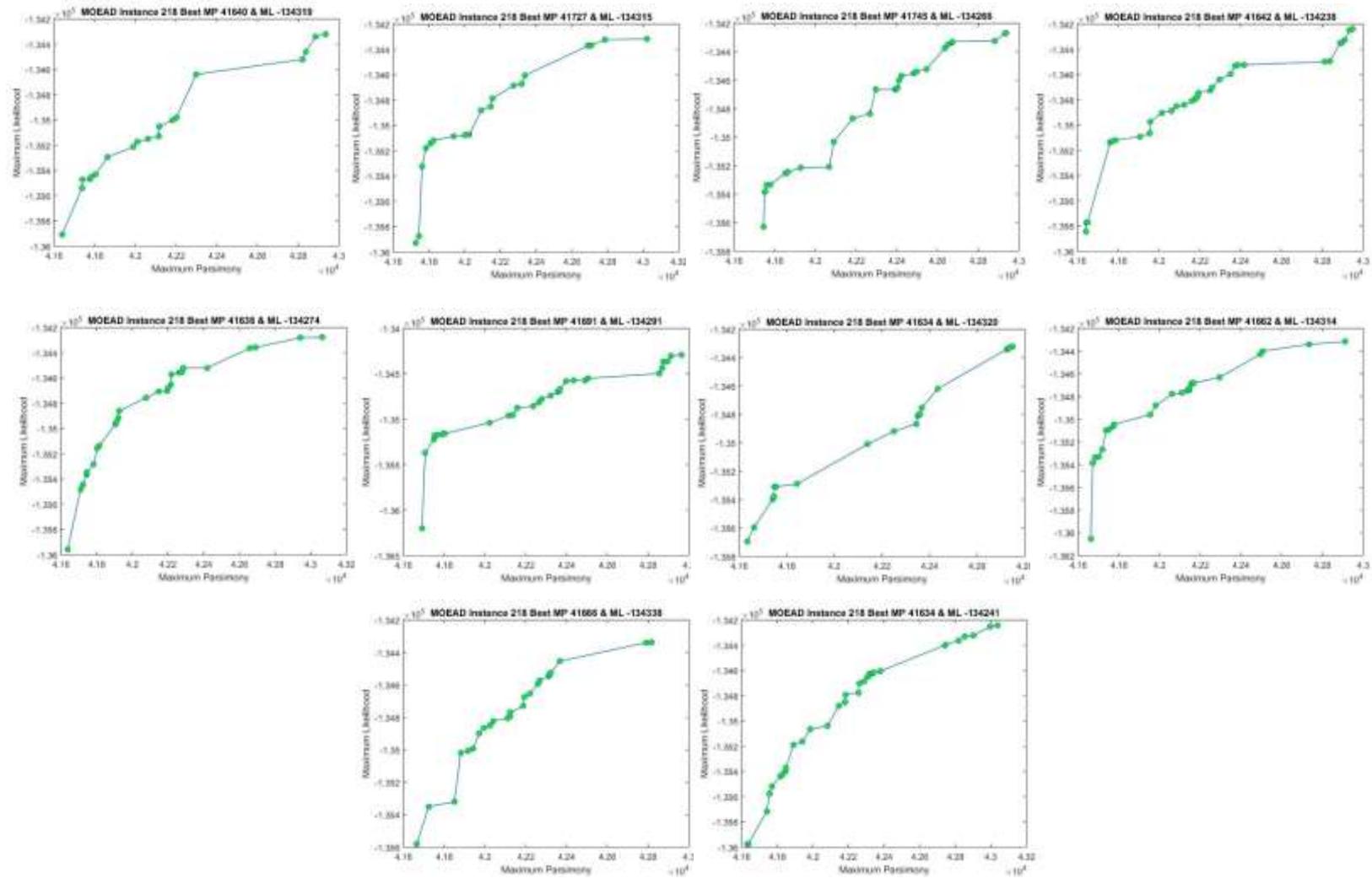
ANEXO 8. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo MOEA/D, para el problema rbcl_55



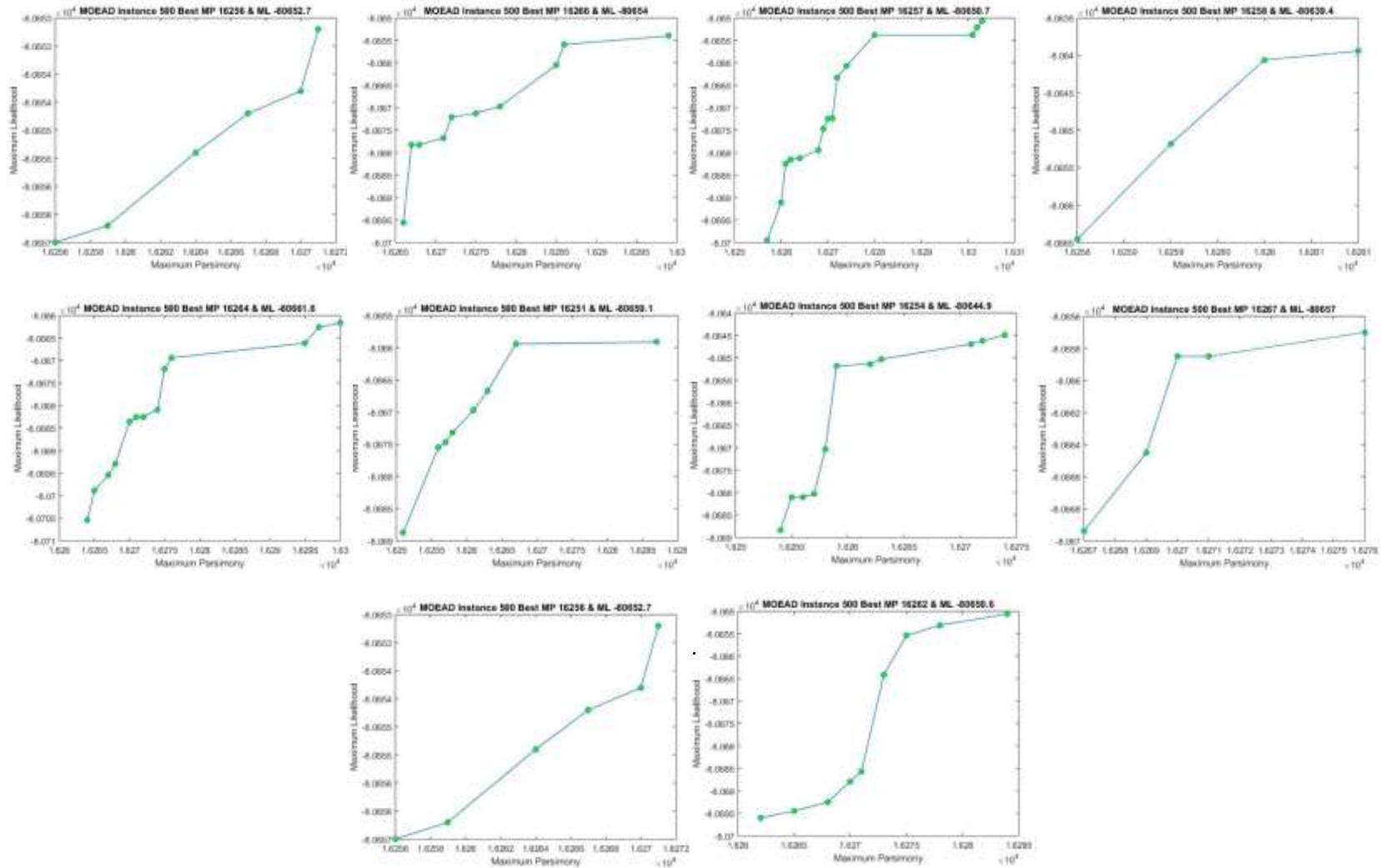
ANEXO 9. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo MOEA/D, para el problema mtDNA_186



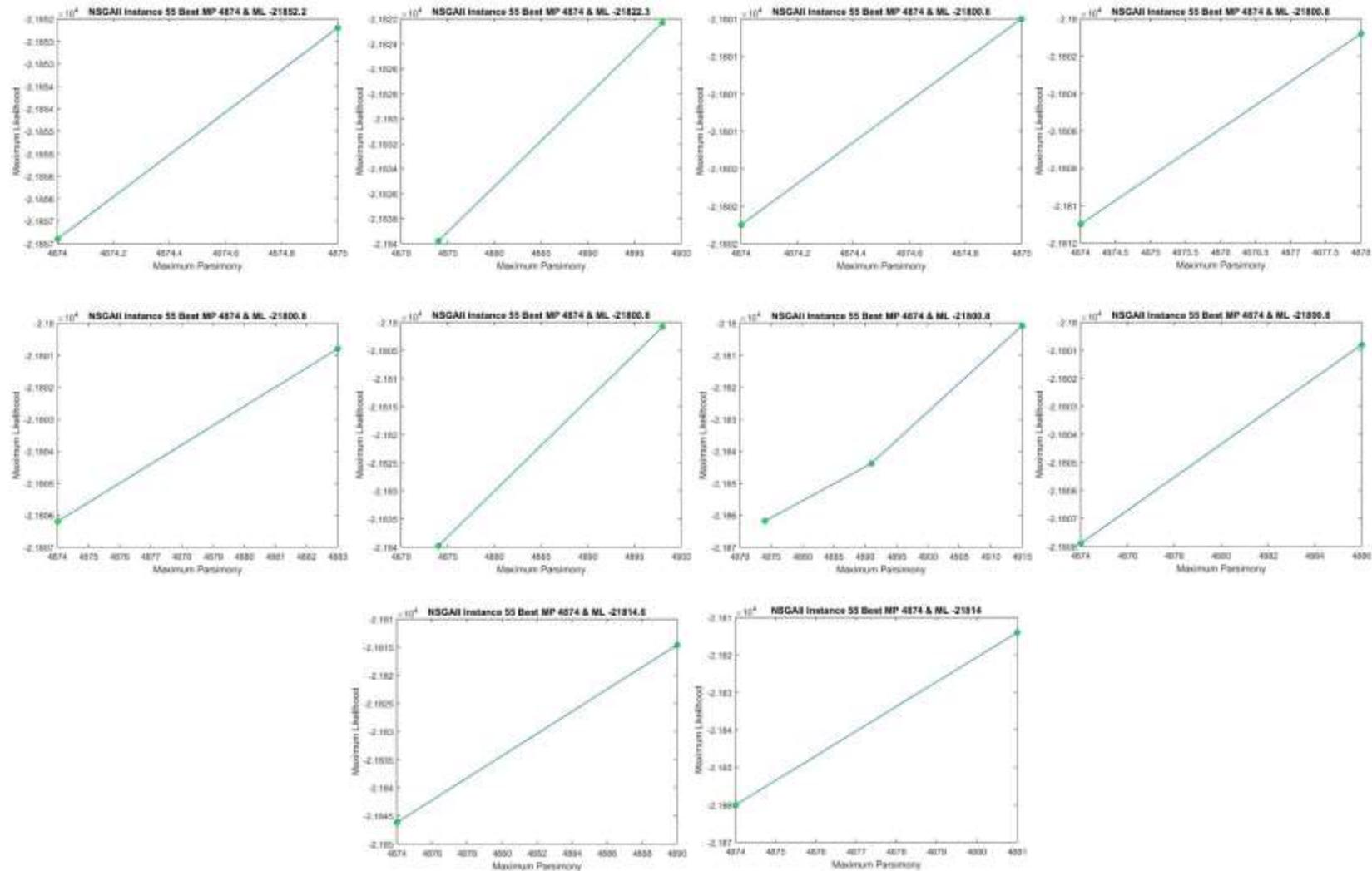
ANEXO 10. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo MOEA/D, para el problema RDPII_218



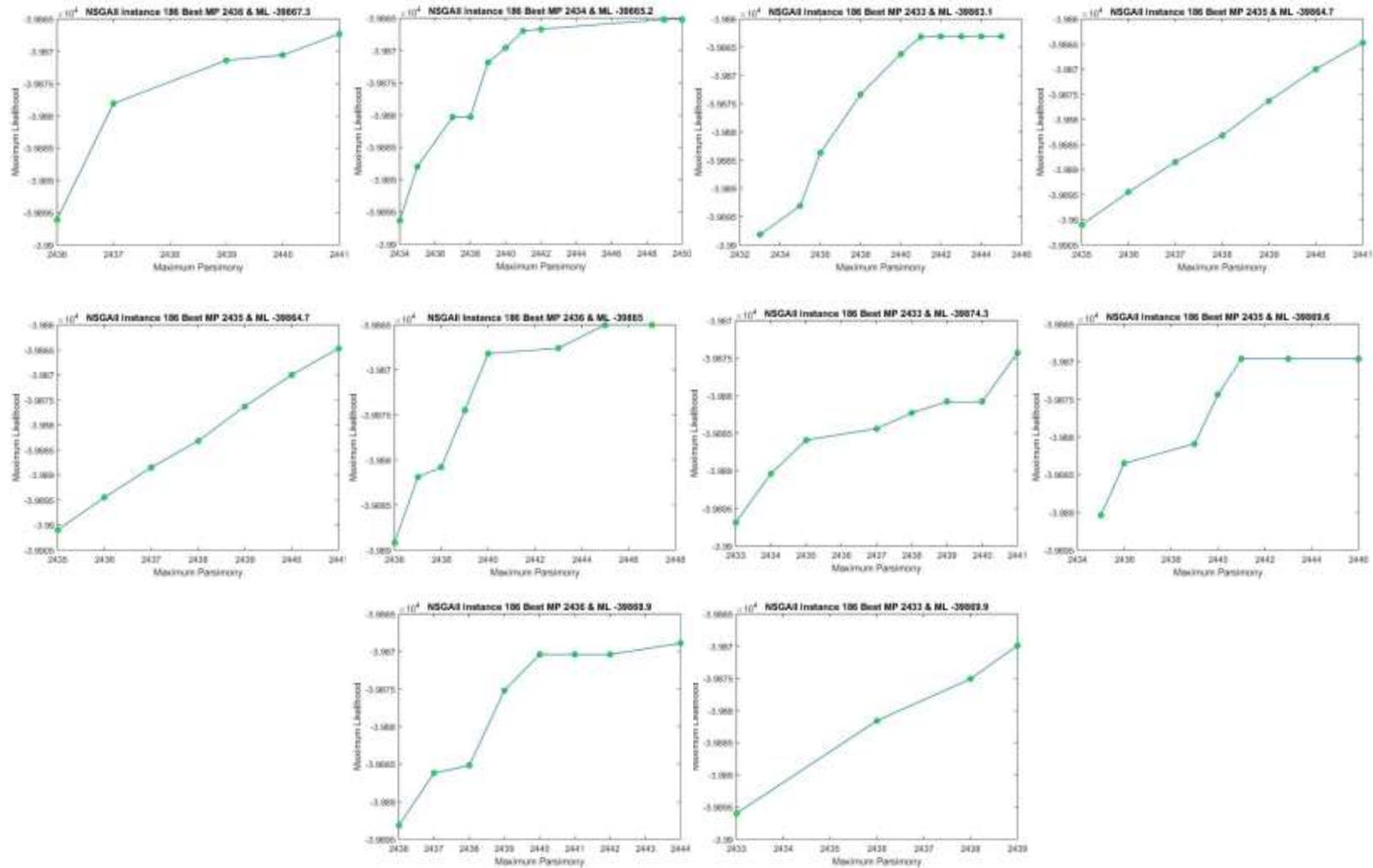
ANEXO 11. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo MOEA/D, para el problema ZILLA_500



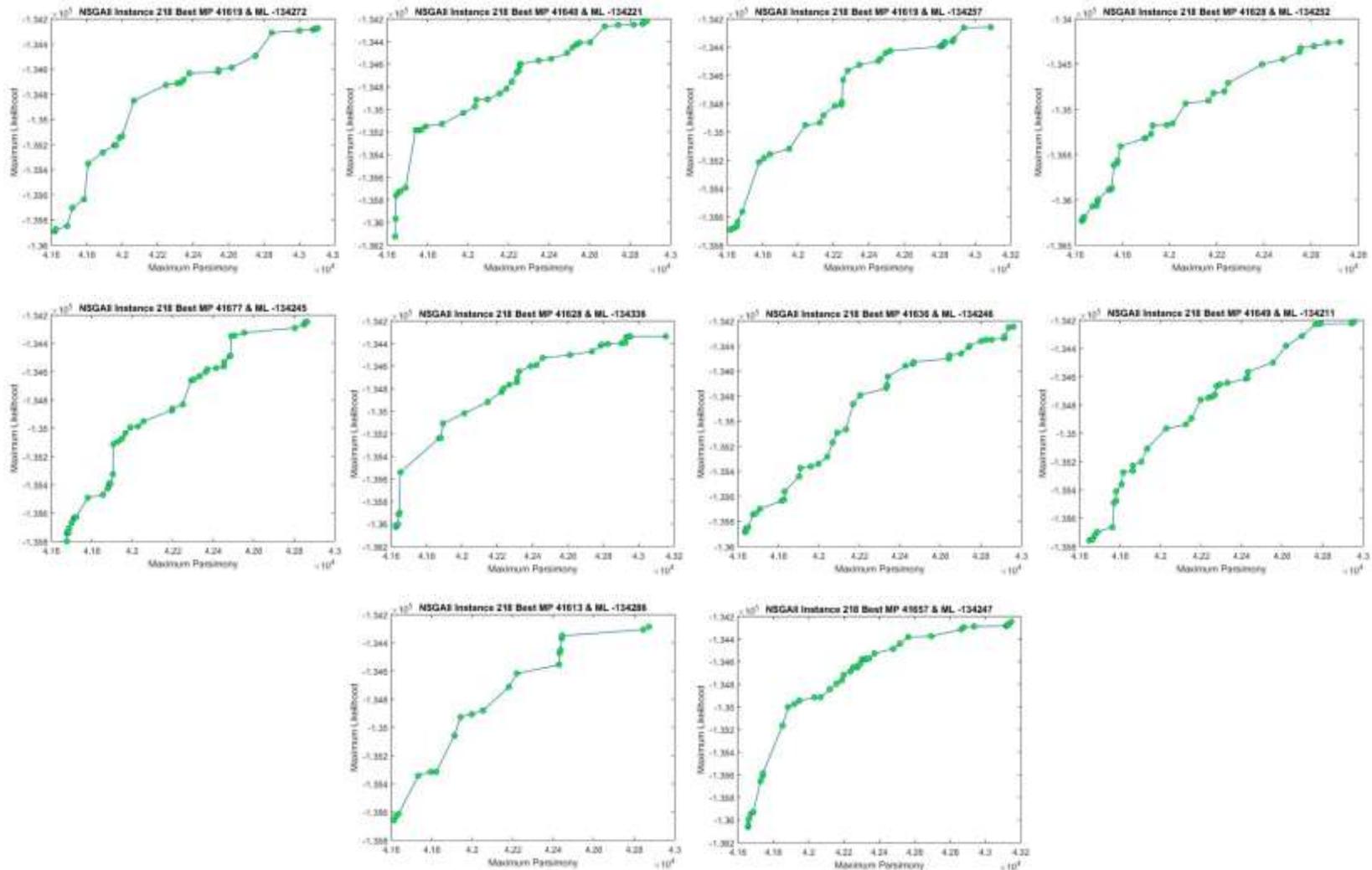
ANEXO 12. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo NSGA-II, para el problema rbcL_55



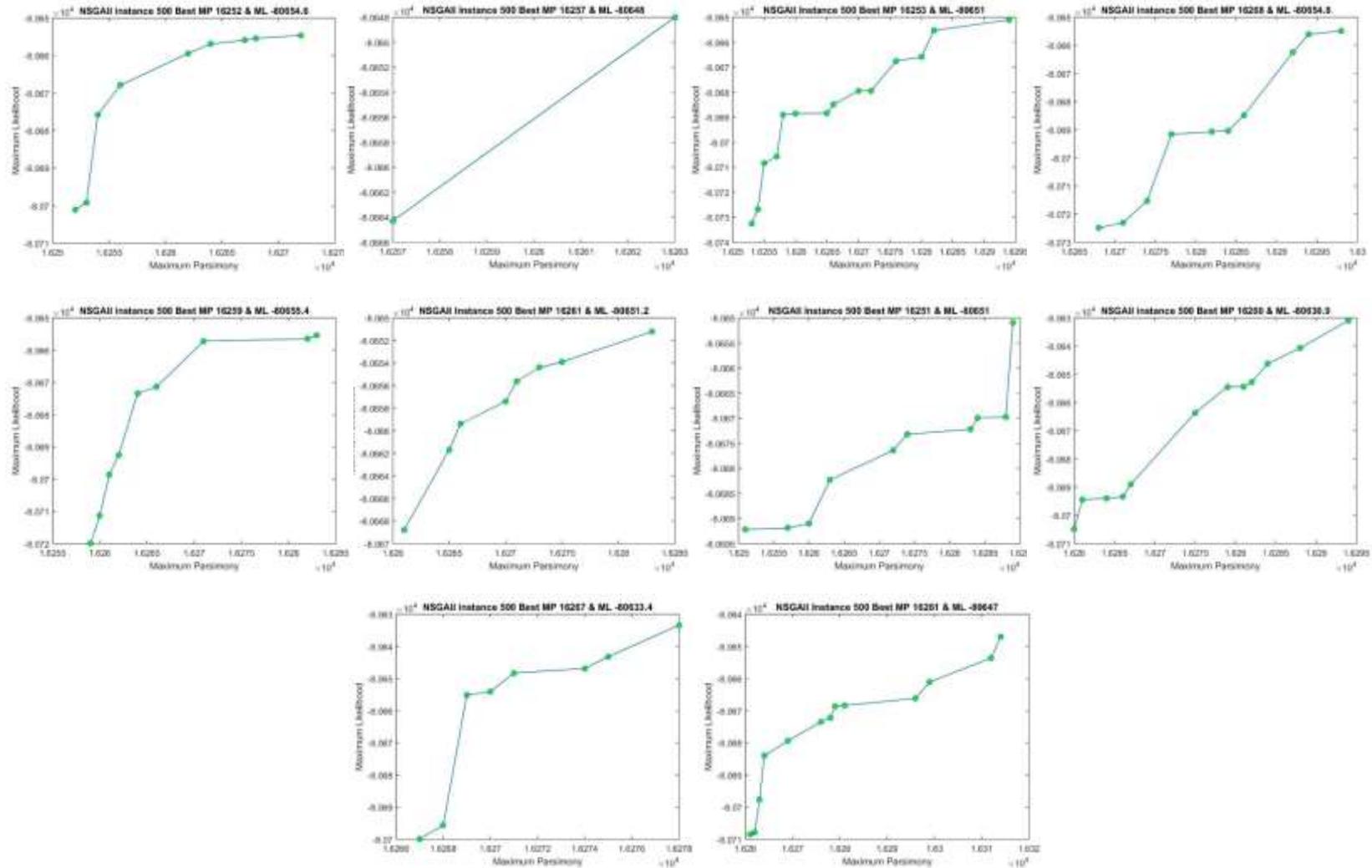
ANEXO 13. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo NSGA-II, para el problema mtDNA_186



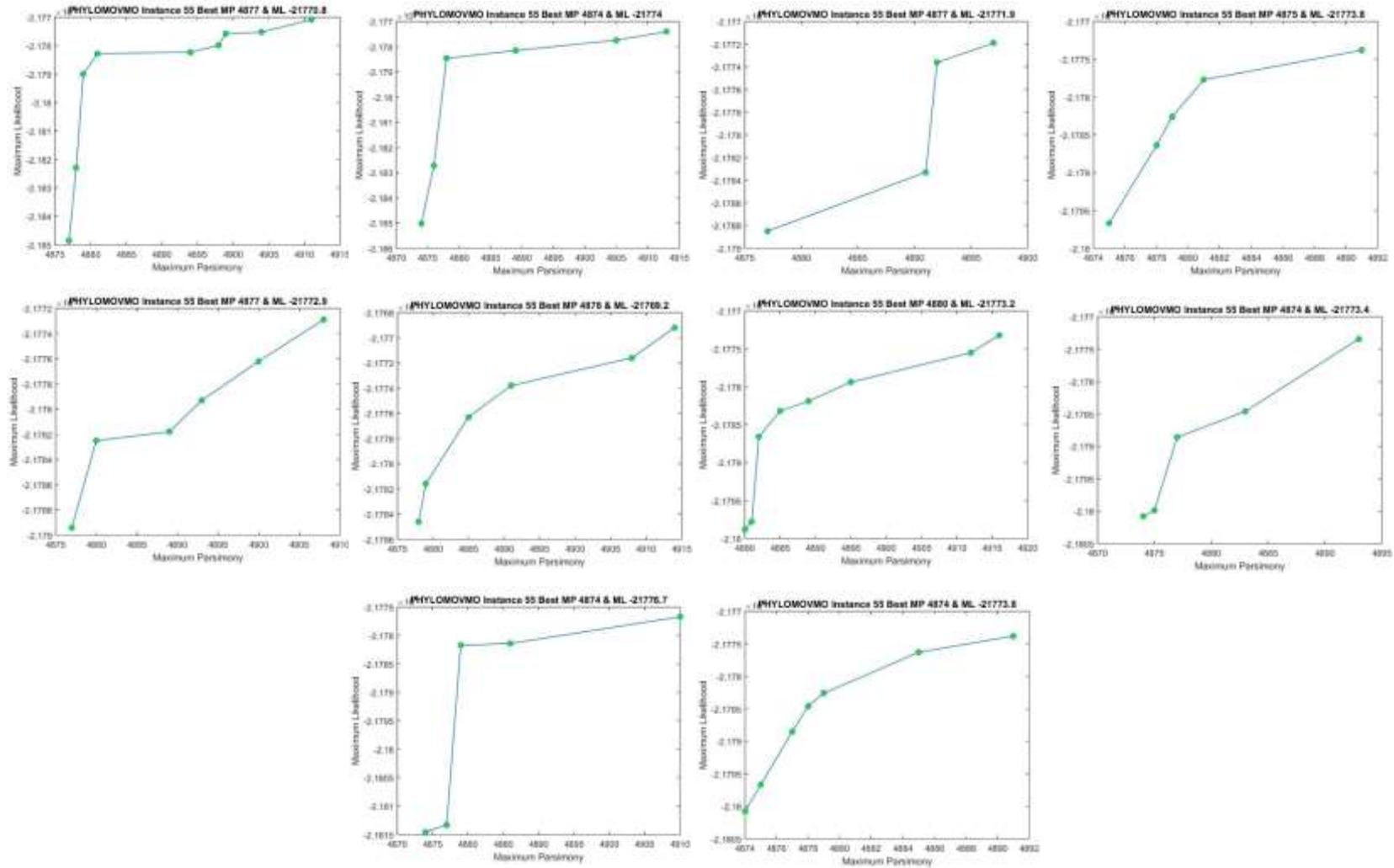
ANEXO 14. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo NSGA-II, para el problema RDPII_218



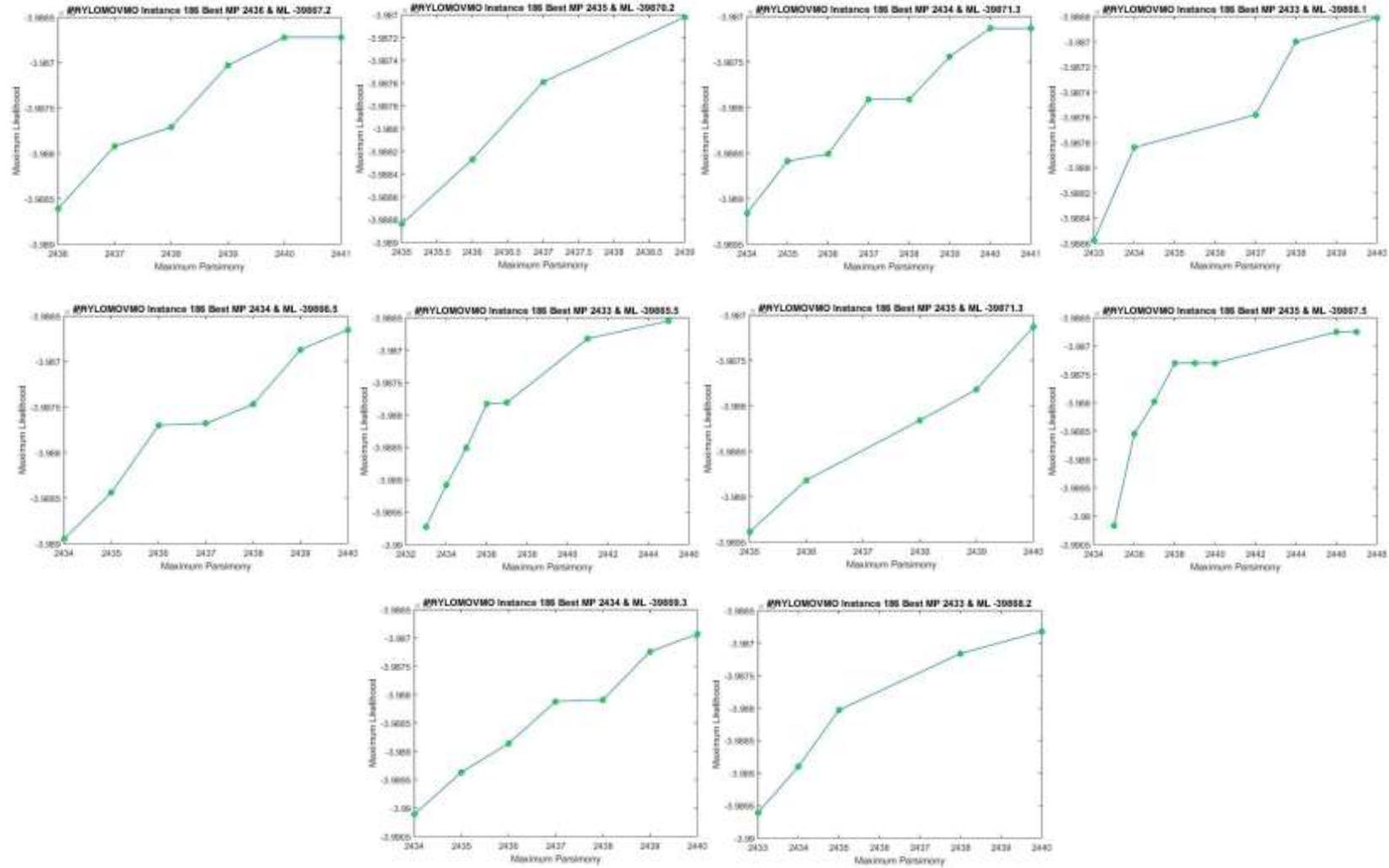
ANEXO 15. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo NSGA-II, para el problema ZILLA_500



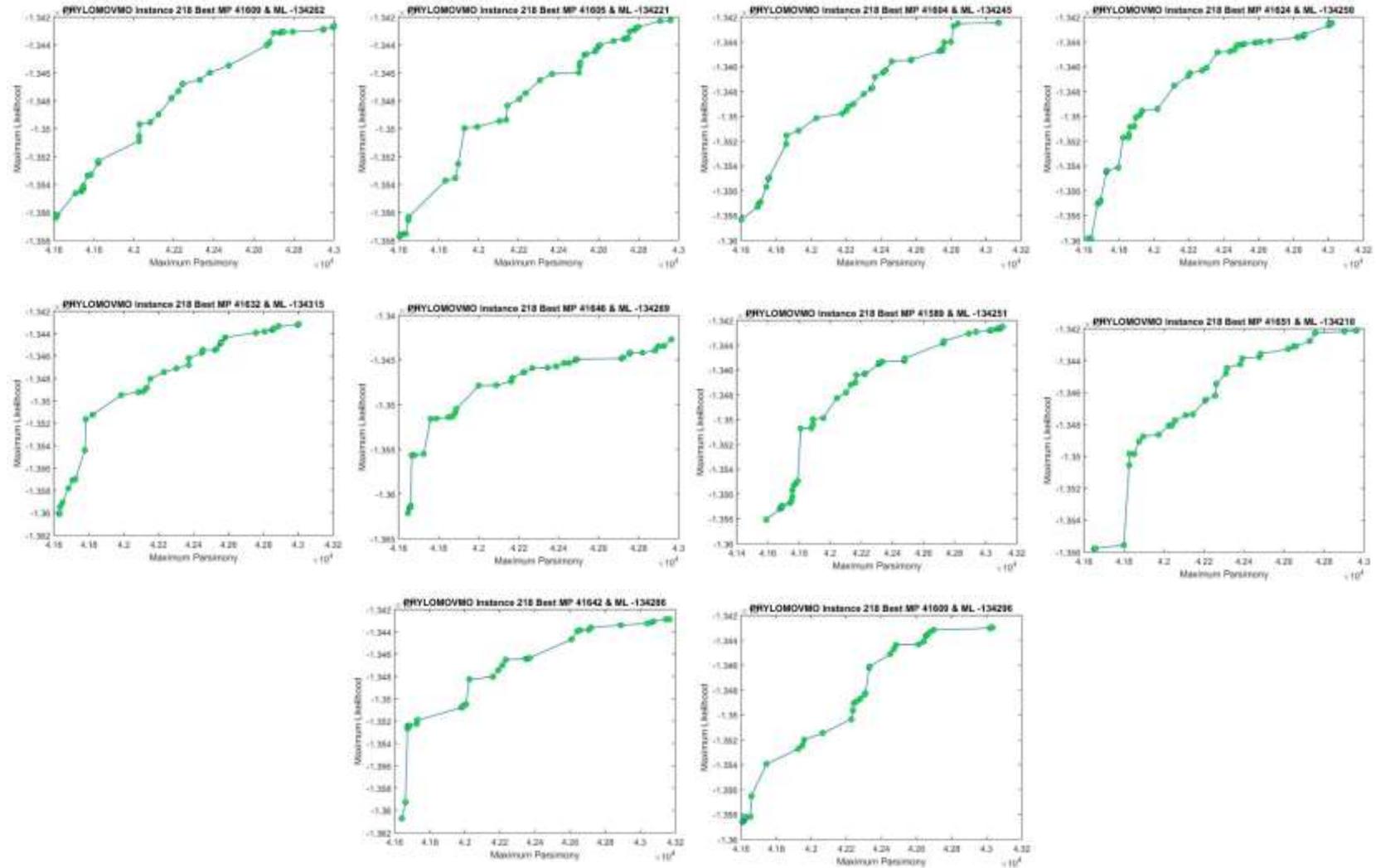
ANEXO 16. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo PhyloMOVMO, para el problema rbcL_55



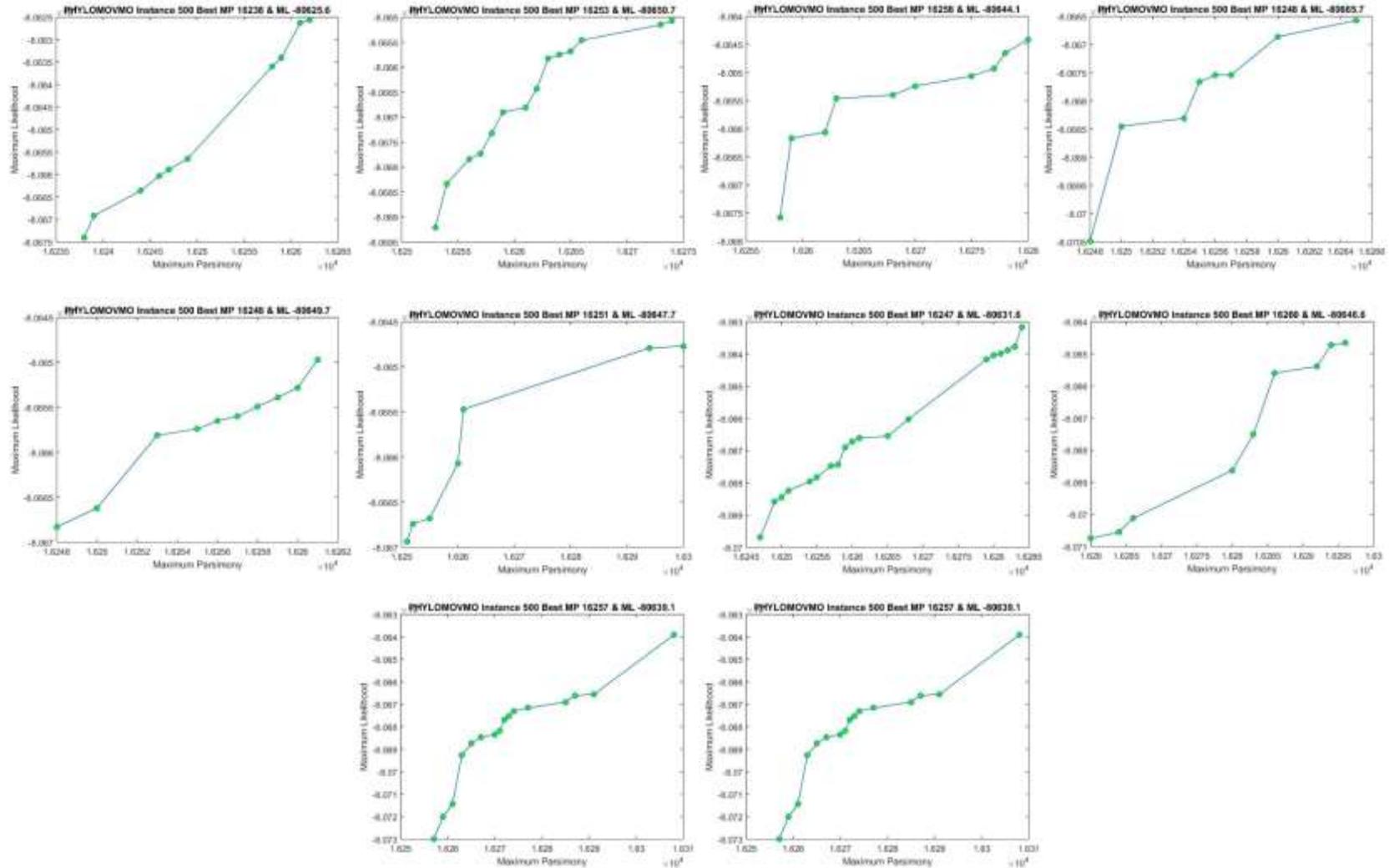
ANEXO 17. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo PhyloMOVMO, para el problema mtDNA_186



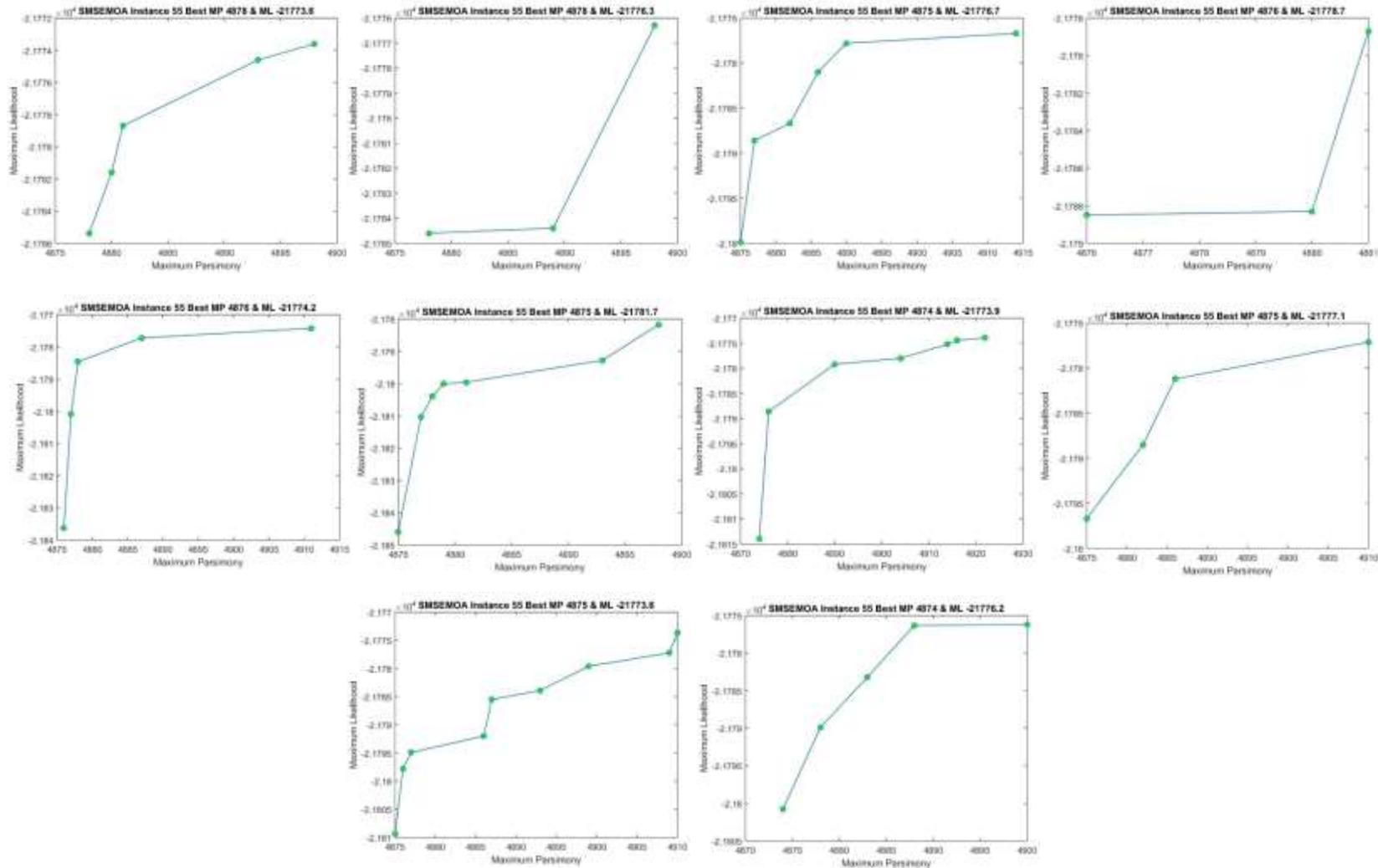
ANEXO 18. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo PhyloMOVMO, para el problema RDPII_218



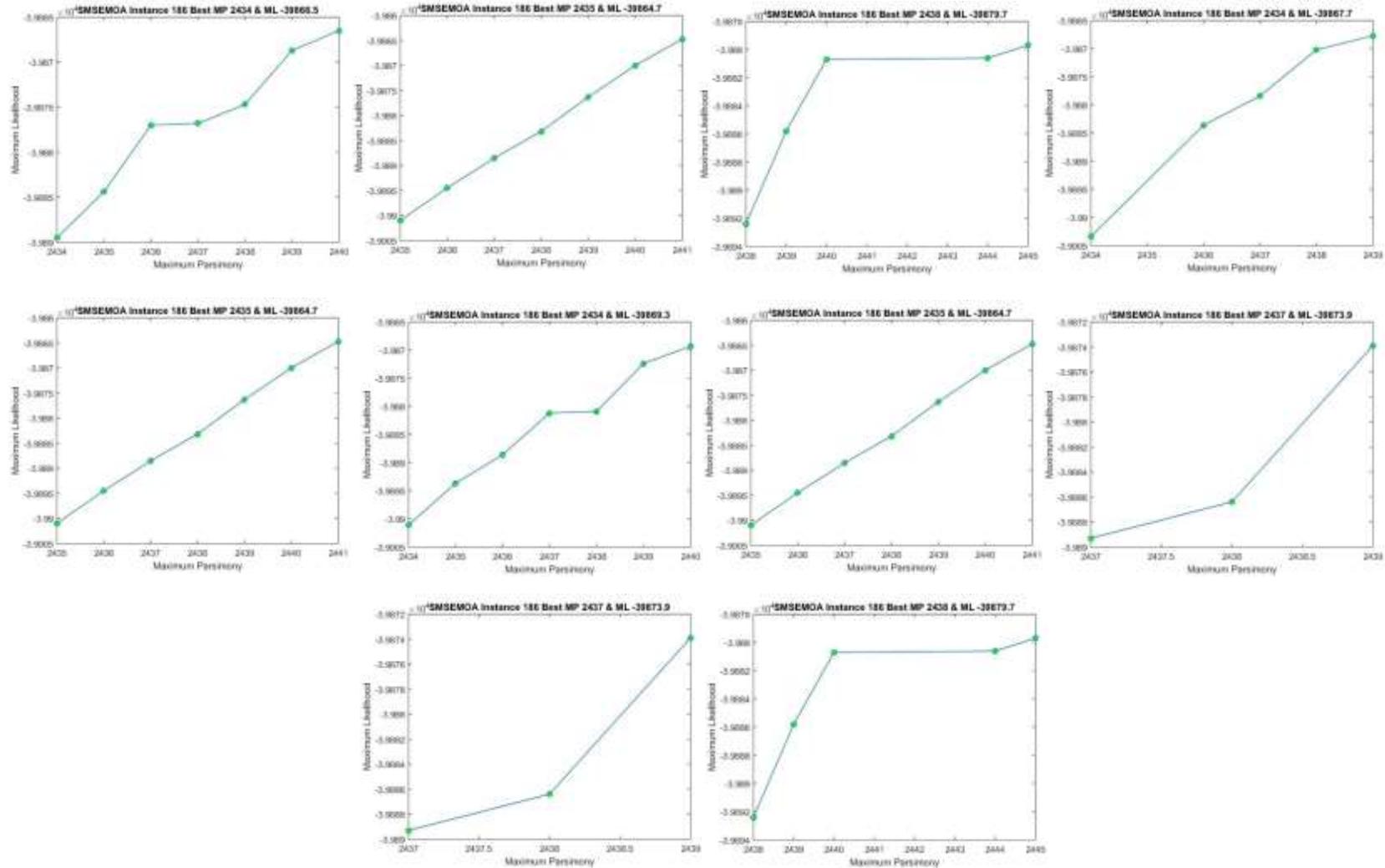
ANEXO 19. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo PhyloMOVMO, para el problema ZILLA_500



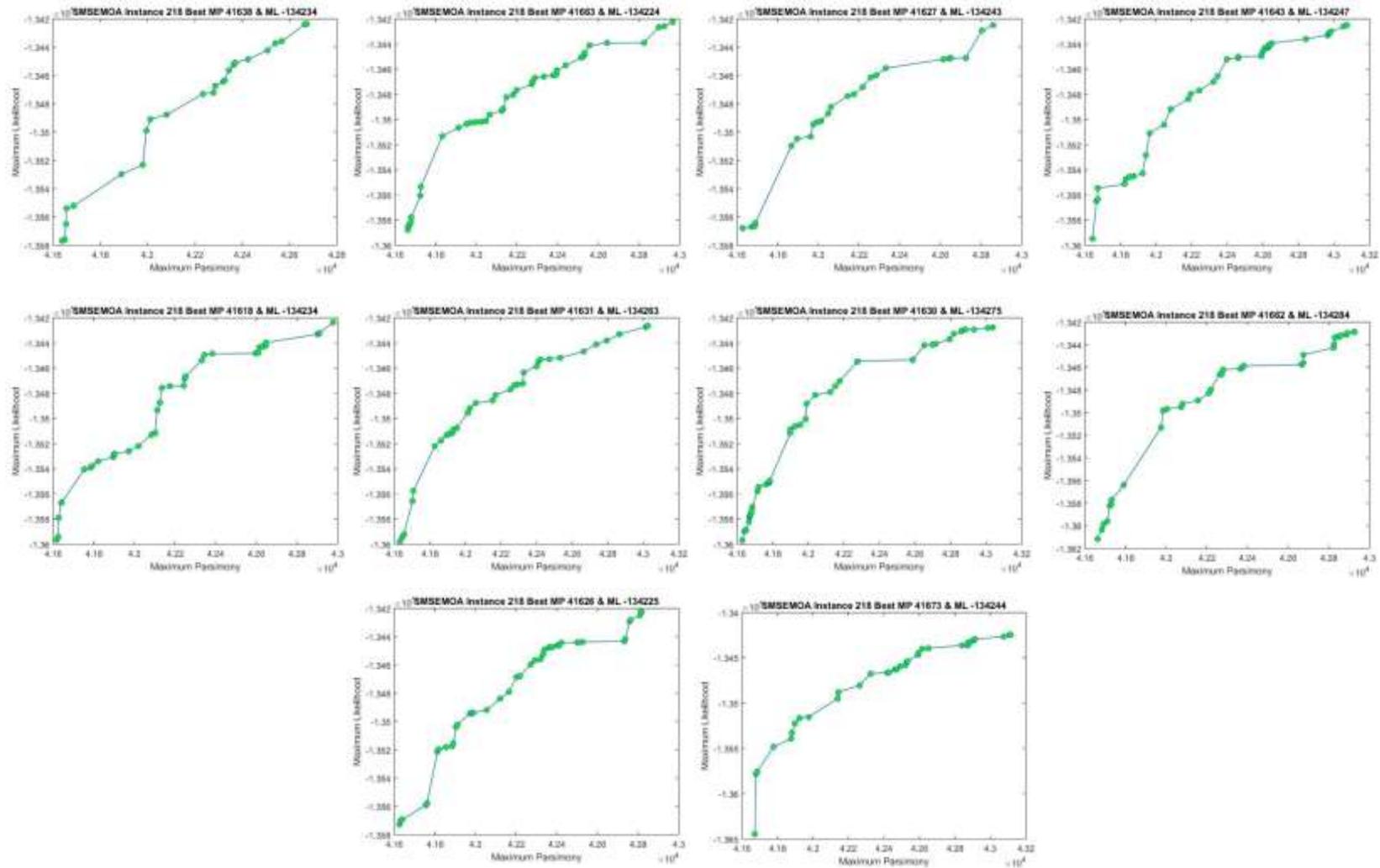
ANEXO 20. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo SMS-EMOA, para el problema rbcL_55



ANEXO 21. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo SMS-EMOA, para el problema mtDNA_186



ANEXO 22. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo SMS-EMOA, para el problema RDPII_218



ANEXO 23. Aproximaciones de Frentes de Pareto generado en base a las 10 ejecuciones independientes del Algoritmo SMS-EMOA, para el problema ZILLA_500

