



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ  
MANUEL FÉLIX LÓPEZ**

**CARRERA DE INFORMÁTICA**

**TRABAJO DE TITULACIÓN PREVIA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERA EN INFORMÁTICA**

**MODALIDAD: SISTEMATIZACIÓN DE EXPERIENCIAS**

**TEMA:**

**REPRESENTACIONES CROMOSÓMICAS DEL ALGORITMO  
NSGA-II PARA EL PROBLEMA DE ÁRBOL DE EXPANSIÓN  
MÍNIMA MULTIOBJETIVO (MST-MO)**

**AUTORA:**

**PAOLA VICTORIA GUAMÁN BRAVO**

**TUTOR:**

**DR. INF. JORGE ANTONIO PÁRRAGA ÁLAVA**

**CALCETA, DICIEMBRE 2019**

## **DERECHOS DE AUTORÍA**

Paola Victoria Guamán Bravo, declaro bajo juramento que el trabajo aquí descrito 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.

A través de la presente declaración cedo los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, según lo establecido por la Ley de Propiedad Intelectual y su reglamento.

.....  
**PAOLA V. GUAMÁN BRAVO**

## CERTIFICACIÓN DEL TUTOR

Jorge Antonio Párraga Álava certifica haber tutelado el trabajo de titulación **REPRESENTACIONES CROMOSÓMICAS DEL ALGORITMO NSGA-II PARA EL PROBLEMA DE ÁRBOL DE EXPANSIÓN MÍNIMA MULTIOBJETIVO**, que ha sido desarrollado por Paola Victoria Guamán Bravo, previa la obtención del título de Ingeniera en Informática, de acuerdo al **REGLAMENTO DE UNIDAD DE TITULACIÓN ESPECIAL DE PROGRAMAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
**DR. INF. JORGE A. PÁRRAGA ÁLAVA**

## **APROBACIÓN DEL TRIBUNAL**

Los suscritos integrantes del tribunal correspondiente, declaran que han **APROBADO** el trabajo de titulación **REPRESENTACIONES CROMOSÓMICAS DEL ALGORITMO NSGA-II PARA EL PROBLEMA DE ÁRBOL DE EXPANSIÓN MÍNIMA MULTIOBJETIVO**, que ha sido propuesto, desarrollado y sustentado por Paola Victoria Guamán Bravo, previa la obtención del título de Ingeniera en Informática, de acuerdo con el **REGLAMENTO DE UNIDAD DE TITULACIÓN ESPECIAL DE PROGRAMAS DE GRADO** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

.....  
ING. ÁNGEL A. VELEZ MERO, MG.

**MIEMBRO**

.....  
ING. ALFONSO T. LOOR VERA, MG.

**MIEMBRO**

.....  
ING. LUIS C. CEDEÑO VALAREZO, MG.

**PRESIDENTE**

## **AGRADECIMIENTO**

A la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López que me dio la oportunidad de una educación superior de calidad y en la cual he forjado mis conocimientos profesionales día a día;

A mis docentes de la carrera de Informática, en especial a quienes han sido mis facilitadores y tutores, los ingenieros: Javier López, Jessica Morales, Joffre Moreira, y Jorge Párraga, por todo el apoyo y disposición de tiempo que me brindaron en cada una de las enseñanzas, consejos y valores impartidos a lo largo del proceso de mi formación como profesional junto con la realización de este trabajo, y

A todas las demás personas de la ESPAM MFL que de una u otra manera intervinieron en la colaboración y apoyo de la culminación del trabajo.

La autora.

## DEDICATORIA

A Dios, por la dicha de haberme permitido contar con cada una de las personas que han estado a mi lado durante el desarrollo y culminación de este trabajo y por permitirme al mismo tiempo seguir disfrutando de la vida que me ha dado, iluminándome siempre en cada paso que daba.

A mi mamá y mi hermana, por la paciencia y atención que han tenido conmigo durante el tiempo de realización de este trabajo de titulación y de los estudios superiores en general.

A mi papá, por todo el sacrificio que ha hecho para poder conseguir los aportes financieros que ayudaron a cumplir la meta propuesta con éxito.

A mi abuela materna, por el tiempo que disponía en escucharme y aconsejarme en medio de los obstáculos que se presentaban, brindándome su apoyo de cualquier forma en que le era posible.

A mis amistades más cercanas, por la confianza que han puesto en mí y por el apoyo moral durante cada etapa de mi formación profesional.

La autora.

## CONTENIDO GENERAL

CARÁTULA .....	i
DERECHOS DE AUTORÍA .....	ii
CERTIFICACIÓN DEL TUTOR .....	iii
APROBACIÓN DEL TRIBUNAL.....	iv
AGRADECIMIENTO.....	v
DEDICATORIA.....	vi
CONTENIDO GENERAL.....	vii
CONTENIDO DE CUADROS Y FIGURAS.....	ix
CONTENIDO DE GRÁFICOS .....	xii
CONTENIDO DE ALGORITMOS.....	xii
RESUMEN .....	xiii
PALABRAS CLAVE.....	xiii
ABSTRACT .....	xiv
KEYWORDS .....	xiv
CAPÍTULO I. ANTECEDENTES .....	1
1.1. DESCRIPCIÓN DE LA INSTITUCIÓN .....	1
1.2. DESCRIPCIÓN DE LA INTERVENCIÓN.....	2
1.3. OBJETIVOS .....	5
1.3.1. OBJETIVO GENERAL.....	5
1.3.2. OBJETIVOS ESPECÍFICOS .....	5
CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN .....	6
2.1. OBJETIVO 1: MATERIAL BIBLIOGRÁFICO.....	6
2.1.1. PROBLEMA A RESOLVER: MST-MO .....	6
2.1.2. ALTERNATIVA DE SOLUCIÓN: NSGA-II .....	7
2.2. OBJETIVO 2: IMPLEMENTACIÓN .....	8
2.2.1. PRÜFER NUMBER .....	9
2.2.2. LINK AND NODE BIASED ENCODING (LNB) .....	10
2.2.3. EDGE-SET ENCODING .....	11
2.3. OBJETIVO 3: PRUEBAS .....	12
2.4. OBJETIVO 4: MEJOR ALTERNATIVA.....	13
2.4.1. MÉTRICA DE DESEMPEÑO MULTIOBJETIVO .....	13

2.4.2. ANÁLISIS ESTADÍSTICOS .....	14
2.4.2.1. TEST DE FRIEDMAN .....	14
2.4.2.2. TEST DE NEMENYI.....	15
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA .....	16
3.1. OBJETIVO 1: MATERIAL BIBLIOGRÁFICO.....	16
3.1.1. INTRODUCCIÓN SOBRE LA TEORÍA DE GENES .....	16
3.1.2. MST Y MST-MO (EJEMPLOS).....	17
3.1.3. FUNCIONAMIENTO GENERAL DEL NSGA-II.....	18
3.1.3.1. FUNCIÓN OBJETIVO .....	20
3.1.3.2. OPERADORES EVOLUTIVOS.....	21
3.2. OBJETIVO 2: IMPLEMENTACIÓN .....	24
3.2.1. EJEMPLO CON PRÜFER NUMBER.....	25
3.2.1.1. FUNCIÓN OBJETIVO Y DOMINANCIA DE PARETO .....	30
3.2.1.2. OPERADORES DE SELECCIÓN, CRUCE Y MUTACIÓN ....	32
3.2.1.3. RESULTADOS EN LA SIGUIENTE GENERACIÓN .....	38
3.2.2. EJEMPLO CON LINK AND NODE BIASED ENCODING (LNB)...	41
3.2.2.1. PARÁMETROS PARA LA CONSTRUCCIÓN DE LOS	
ÁRBOLES.....	42
3.2.2.2. OPERADORES DE CRUZAMIENTO Y MUTACIÓN .....	46
3.2.3. EJEMPLO CON EDGE-SET ENCODING .....	47
3.3. OBJETIVO 3: PRUEBAS .....	53
3.4. OBJETIVO 4: MEJOR ALTERNATIVA.....	54
3.4.1. CÁLCULOS DE MÉTRICAS (HIPERVOLUMEN).....	56
3.4.2. ANÁLISIS ESTADÍSTICOS .....	56
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES .....	59
4.1. CONCLUSIONES.....	59
4.2. RECOMENDACIONES .....	60
BIBLIOGRAFÍA .....	62
ANEXOS .....	66
ANEXO 1. MEJORES MST-MO DE CADA REPRESENTACIÓN .....	67
ANEXO 2. HV COMPLETO DE CADA BASE DE DATOS.....	79

## CONTENIDO DE CUADROS Y FIGURAS

### CUADROS

Cuadro 3.1. Listado de los operadores evolutivos utilizados por cada representación en el NSGA-II.....	22
Cuadro 3.2. Datos de ejemplo para las variables establecidas.....	24
Cuadro 3.3. Valores de las funciones objetivo población Pt (1era generación – ejemplo con Prüfer number).....	31
Cuadro 3.4. Ranking Pareto de la población Pt (1era generación – ejemplo con Prüfer number). .....	31
Cuadro 3.5. Ranking Pareto de la población Rt (1era generación – ejemplo con Prüfer number). .....	38
Cuadro 3.6. Ranking Pareto de la población Rt (2da generación – ejemplo con Prüfer number). .....	40
Cuadro 3.7. Ranking Pareto de las mejores soluciones obtenidas (ejemplo con Prüfer number). .....	41
Cuadro 3.8. Genotipos para la población Pt (ejemplo con representación LNB). .....	42
Cuadro 3.9. Parámetros del NSGA-II .....	53
Cuadro 3.10. Resumen promedio del cálculo de HV en las representaciones por cada base de datos. ....	56
Cuadro 3.11. Estadística resumen de Arabidopsis.....	57
Cuadro 3.12. Estadística resumen de Cell cycle .....	57
Cuadro 3.13. Estadística resumen de Sporulation .....	57
Cuadro 3.14. Estadística resumen de Serum.....	57
Cuadro 3.15. Resultados en las diferencias significativas según post hoc test (Nemenyi).....	58
Cuadro 3.16. Hipervolumen completo de Arabidopsis .....	79
Cuadro 3.17. Hipervolumen completo de Cell cycle .....	79
Cuadro 3.18. Hipervolumen completo de Sporulation .....	80
Cuadro 3.19. Hipervolumen completo de Serum.....	80

## FIGURAS

Figura 2.1. Construcción del árbol de expansión a partir de la secuencia de Prüfer (2, 5, 6, 8, 2, 5). .....	9
Figura 2.2. Árbol de expansión con 12 nodos y un conjunto de bordes que lo representa. ....	11
Figura 2.3. Ejemplo para métrica HV. ....	13
Figura 2.4. Hipervolumen encerrado por las soluciones no dominadas. ....	14
Figura 3.1. Ejemplo de Árbol de expansión.....	17
Figura 3.2. Ejemplo de Árbol de Expansión Mínima (MST) a partir de un grafo conectado completo. ....	18
Figura 3.3. Comparación entre un MST (a) y un MST-MO (b). ....	18
Figura 3.4. Concepto de dominancia aplicado a un ejemplo de minimización. ....	19
Figura 3.5. Diagrama de selección de individuos en el NSGA-II.....	20
Figura 3.6. Matriz de Peso Distancia entre los nodos usando la ecuación 3.1. ....	25
Figura 3.7. Matriz de Peso Distancia entre los nodos usando la ecuación 3.2. ....	25
Figura 3.8. Población Pt (1era generación – ejemplo con Prüfer Number). ....	26
Figura 3.9. Unión de los primeros nodos del Individuo 1 (ejemplo con Prüfer number).....	26
Figura 3.10. Unión de los siguientes elementos del Individuo 1 (ejemplo con prüfer number).....	27
Figura 3.11. Individuo 1 (ejemplo con representación Prüfer number).....	27
Figura 3.12. Individuo 2 (ejemplo con representación Prüfer number).....	28
Figura 3.13. Individuo 3 (ejemplo con representación Prüfer number).....	28
Figura 3.14. Individuo 4 (ejemplo con representación Prüfer number).....	29
Figura 3.15. Individuo 5 (ejemplo con representación Prüfer number).....	29
Figura 3.16. Individuo 6 (ejemplo con representación Prüfer number).....	30
Figura 3.17. Puntos de corte para el cruzamiento en individuos de la representación Prüfer number.....	33
Figura 3.18. Proceso de Cruzamiento en los Individuos 4 y 5. ....	33
Figura 3.19. Población Qt (1era generación – ejemplo con Prüfer number). ...	34
Figura 3.20. Proceso de mutación en el Individuo 8.....	35
Figura 3.21. Individuo 7 (ejemplo con representación Prüfer number).....	35

Figura 3.22. Individuo 8 (ejemplo con representación Prüfer number).....	35
Figura 3.23. Individuo 9 (ejemplo con representación Prüfer number).....	36
Figura 3.24. Individuo 10 (ejemplo con representación Prüfer number).....	36
Figura 3.25. Individuo 11 (ejemplo con representación Prüfer number).....	36
Figura 3.26. Individuo 12 (ejemplo con representación Prüfer number).....	37
Figura 3.27. Individuos ganadores de la primera generación (ejemplo con Prüfer number).....	38
Figura 3.28. Población $R_t$ (2da generación – ejemplo con Prüfer number).....	39
Figura 3.29. Mejores soluciones obtenidas en dos generaciones (ejemplo con Prüfer number). .....	41
Figura 3.30. Pesos de nodo y enlace del Individuo 1 (ejemplo con la representación LNB). .....	42
Figura 3.31. Matrices de distancia modificadas del Individuo 1 (ejemplo con representación LNB). .....	43
Figura 3.32. Matriz de distancia modificada del Individuo 1 (ejemplo con representación LNB). .....	44
Figura 3.33. Individuo 1 (ejemplo con representación LNB).....	44
Figura 3.34. Individuo 2 (ejemplo con representación LNB).....	44
Figura 3.35. Individuo 3 (ejemplo con representación LNB).....	45
Figura 3.36. Individuo 4 (ejemplo con representación LNB).....	45
Figura 3.37. Individuo 5 (ejemplo con representación LNB).....	45
Figura 3.38. Individuo 6 (ejemplo con representación LNB).....	45
Figura 3.39. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 1.....	47
Figura 3.40. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.....	48
Figura 3.41. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.....	49
Figura 3.42. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.....	50
Figura 3.43. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.....	51
Figura 3.44. Vista de uno de los scripts principales elaborados en RStudio ....	52

## CONTENIDO DE GRÁFICOS

Gráfico 3.1. Fronteras de Pareto de la población Pt (1era Generación - ejemplo con Prüfer number). .....	32
Gráfico 3.2. Fronteras de Pareto de la población Rt (1era Generación - ejemplo con Prüfer Number). .....	37
Gráfico 3.3. Fronteras de Pareto de la población Rt (2da generación - ejemplo con Prüfer number). .....	39
Gráfico 3.4. Fronteras de Pareto de las mejores soluciones obtenidas (ejemplo con Prüfer number). .....	40
Gráfico 3.5. Mejores fronteras de Pareto aproximadas obtenidas en Arabidopsis Thaliana.....	54
Gráfico 3.6. Mejores fronteras de Pareto aproximadas obtenidas en Cell Cycle. ....	54
Gráfico 3.7. Mejores fronteras de Pareto aproximadas obtenidas en Sporulation. ....	55
Gráfico 3.8. Mejores fronteras de Pareto aproximadas obtenidas en Serum. ...	55

## CONTENIDO DE ALGORITMOS

ALGORITMO 1. Pseudocódigo del NSGA-II .....	7
ALGORITMO 2. Pseudocódigo de decodificación de Prüfer Number .....	9
ALGORITMO 3. Pseudocódigo del algoritmo de PRIM .....	10
ALGORITMO 4. Pseudocódigo del algoritmo de KruskalRST .....	11

## RESUMEN

La finalidad de este trabajo de titulación consistió en identificar la mejor alternativa de representación cromosómica que lograra mejorar el rendimiento en términos multiobjetivo de un algoritmo genético conocido como NSGA-II, el cual fue utilizado a la vez para abordar el problema del árbol de expansión mínima en su versión multiobjetivo. Esta identificación se logró a través de la evaluación exhaustiva de tres distintos tipos de representación cromosómica: Prüfer number, link and node biased encoding (LNB), y edge-set encoding. En unión a ello, se emplearon métodos de investigación científica (bibliográfico y analítico) para la búsqueda, interpretación y análisis de la información y de los resultados obtenidos en cada ejecución. Se empleó el software RStudio para la codificación y posterior ejecución de las representaciones, mismas que fueron aplicadas al NSGA-II junto con sus operadores evolutivos respectivos. En la realización de los experimentos computacionales se utilizaron datos de expresión génica, los cuales corresponden a cuatro bases de datos distintas disponibles en la web: arabidopsis, cell cycle, sporulation y serum. Una vez realizados dichos experimentos, se evaluó el rendimiento de las representaciones utilizando la métrica de desempeño multiobjetivo hipervolumen, con el fin de identificar las mejores fronteras de Pareto de cada representación. Luego del cálculo del hipervolumen se aplicaron pruebas estadísticas a sus datos por medio de los tests de Friedman y Nemenyi. Dichos cálculos de hipervolumen y test estadísticos permitieron determinar que la representación LNB fue la mejor alternativa para el mejoramiento del NSGA-II.

## PALABRAS CLAVE

NSGA-II, árbol de expansión mínima multiobjetivo, representaciones para algoritmos genéticos, prüfer number, link and node biased encoding, edge-set encoding.

## ABSTRACT

The purpose of this titration work was to identify the best alternative of chromosome representation that could improve the performance in multiobjective terms of a genetic algorithm known as NSGA-II, which was used at the same time to address the problem of the minimum spanning tree in its multiobjective version. This identification was achieved through the thorough evaluation of three different types of chromosome representation: Prüfer number, link and node biased encoding (LNB), and edge-set encoding. In conjunction with this, scientific research methods (bibliographic and analytical) were used to search, interpret and analyze the information and the results obtained in each execution. RStudio software was used for the coding and subsequent execution of the representations, which were applied to the NSGA-II together with their respective evolutionary operators. In the realization of the computational experiments, gene expression data were used, which correspond to four different databases available on the web: arabidopsis, cell cycle, sporulation and serum. Once these experiments were performed, the performance of the representations was evaluated using the hypervolume multiobjective performance metric, in order to identify the best Pareto boundaries of each representation. After the calculation of the hypervolume, statistical tests were applied to their data by means of the Friedman and Nemenyi tests. These hypervolume calculations and statistical tests allowed us to determine that the LNB representation was the best alternative for the improvement of NSGA-II.

## KEYWORDS

NSGA-II, multi-objective minimum spanning tree, representations for genetic algorithms, prüfer number, link and node biased encoding, edge-set encoding.

# CAPÍTULO I. ANTECEDENTES

## 1.1. DESCRIPCIÓN DE LA INSTITUCIÓN

La ESPAM MFL, mediante la Coordinación General de investigación realiza trabajos mancomunados por el aporte de la universidad a la ciencia. La planificación estratégica de la ESPAM MFL 2017-2021 plantea en el subsistema de investigación como objetivo estratégico “fortalecer el sistema de gestión de la investigación para que se contribuya al desarrollo de la zona 4 y el país”, y como objetivo específico establece “ejecutar investigaciones que contribuyan al desarrollo de la zona 4 y del país” (ESPAM MFL, 2018a).

Con los antecedentes previamente mencionados, la ESPAM MFL cuenta con un grupo de investigación denominado Sistemas Computacionales (SISCOM) que tiene como objetivo “contribuir al sector agro-productivo y de servicios con soluciones computacionales, que aporten al desarrollo tecnológico y al entorno social” para mejorar los procesos investigativos en esta área. Este grupo de investigación contempla principalmente los siguientes campos: Aprendizaje automático y asistido, visión por computador, minería de datos, robótica (Inteligencia artificial), optimización de procesos, sistemas distribuidos (ESPAM MFL, 2018a).

El grupo de investigación SISCOM indica en su misión “desarrollar investigaciones computacionales con pertinencia, compromiso ético y social, en procura del mejoramiento del sector agro-productivo y de servicios”. Además, tiene planteado como visión “ser un referente en investigación y desarrollo relacionados a sistemas y tecnologías computacionales que contribuyan al progreso agro-productivo y de servicio a nivel local o nacional” (ESPAM MFL, 2018a).

El grupo SISCOM se rige bajo el Reglamento para la conformación de Grupos de Investigación de la ESPAM MFL, el cual indica en su artículo primero “establecer las normas de cumplimiento para las líneas de investigación que presentan temas agrupados de estudios científicos que se fundamentan en

tradición investigativa, de donde se originan programas y proyectos cuyos resultados guarden afinidades entre sí” (ESPAM MFL, 2018b).

En contribución al desarrollo de la investigación científica que se efectúa en el grupo SISCOM, se pretende con este trabajo de titulación, dejar a disponibilidad del grupo los resultados que se obtengan de la evaluación de tres tipos de representaciones cromosómicas utilizadas en el Algoritmo Genético con Ordenamiento No-Dominado II o NSGA-II por sus siglas en inglés; con el propósito de mejorar el rendimiento del mismo en términos multiobjetivo para el problema del Árbol de Expansión Mínima Multiobjetivo (MST-MO).

## **1.2. DESCRIPCIÓN DE LA INTERVENCIÓN**

Muchos problemas de la vida real tienen varios objetivos en conflicto que se requieren optimizar al mismo tiempo, estos se denominan problemas multiobjetivo y la necesidad de resolverlos se presenta frecuentemente en cualquier actividad técnica o científica (Pandolfi *et al.*, 2017; UNED, 2015).

En el campo de la ingeniería es habitual considerar múltiples aspiraciones u objetivos enfrentados entre sí y la principal dificultad al considerar esta optimización multiobjetivo es que no existe una definición aceptada de óptimo o “la mejor” en este caso, más bien estos problemas admiten múltiples soluciones generalmente; por lo tanto, es difícil comparar una solución con otra, con lo que se hace necesario el estudio de técnicas de decisión basadas en un número finito de objetivos o criterios (López, 2015; UNED, 2015)

La investigación de este trabajo de titulación se lleva a cabo prestando especial atención en el área de la Bioinformática, que es el uso de técnicas computacionales, matemáticas y estadísticas para el análisis, interpretación y generación de datos biológicos (Olaya-Abril & Cejas-Molina, 2018). En la bioinformática se abordan diferentes problemas, pero este trabajo de titulación está orientado principalmente al problema del clustering de genes, el cual consiste en agrupar genes basados en su similitud de funciones biológicas. Para

resolver este problema existen algunos algoritmos que basan su funcionamiento en la teoría de grafos mediante el problema del MST (Párraga Álava, 2017).

El problema del MST consiste en encontrar un subgrafo que conecte todos los nodos al menor costo a partir de un grafo finito conectado. En este sentido, los nodos corresponden a los genes y las conexiones entre ellos representan la similitud, teniendo en cuenta características como expresión génica y funciones biológicas. Considerando que cada arista tiene asociado un costo, en una versión Multiobjetivo del problema MST (MST-MO) la complejidad se incrementa debido a los múltiples costos que surgen al representar conexiones con diversos valores (Montoya *et al.*, 2014; Párraga-Álava *et al.*, 2017; Simmons, 2015).

Una vez resuelto el problema del MST-MO, se podría identificar grupos de genes que estén asociados o compartan similares perfiles de expresión, situación que indica una alta probabilidad que dichos genes participen de una determinada función biológica.

Existen algunos enfoques que ofrecen soluciones aproximadas para el problema del MST-MO, entre ellos se destaca el uso de Algoritmos Evolutivos (EAs), debido a que utilizan mecanismos inspirados en la evolución natural para trabajar con una población de individuos que representen posibles soluciones para el problema (Jimenez & Chivata, 2017; Pandolfi *et al.*, 2017).

Con el fin de dar solución al problema del MST-MO se propone el uso de Algoritmos Genéticos (AGs). La razón de su uso se debe a su simplicidad y eficiencia para encontrar buenas soluciones en grandes y complejos espacios de búsqueda (García, 2018; López, 2015). El algoritmo que se utilizará para abordar el problema del MST-MO es el Algoritmo Genético con Ordenamiento No Dominado-II (NSGA-II), debido a su característica de producir y ofrecer una población de buenas soluciones aproximadas en un corto tiempo de cálculo y en una sola ejecución. Para su funcionamiento debe contar con los siguientes componentes: representación cromosómica, función objetivo y operadores evolutivos (selección, cruce y mutación) (Campos *et al.*, 2016; Fang *et al.*, 2018; López, 2015; Saborido, 2014; Zhang & Ma, 2015).

En términos simples, el NSGA-II comienza con una población inicial (representación cromosómica de individuos) a partir de la cual se genera una población descendiente. Dicho proceso es realizado mediante la cuantificación de una función objetivo que cada individuo de la población deberá tener para poder ser rankeados de acuerdo con un criterio conocido como dominancia de Pareto; en otras palabras, la idea es ir preservando las mejores soluciones y desechando aquellas que no tienen un rendimiento del todo adecuado. Posterior a ello, se hace uso de los operadores de selección, cruce y mutación; que son técnicas propias de los algoritmos genéticos (Meza *et al.*, 2016; Montoya *et al.*, 2014; Rojas *et al.*, 2016; Zhang & Ma, 2015).

Según Lima (2017) para un uso eficiente y exitoso de los algoritmos genéticos, no es suficiente simplemente aplicar un algoritmo genético estándar. Es necesario encontrar una representación adecuada para el problema y hacer uso de operadores de búsqueda apropiados que se ajusten bien a las propiedades de la representación.

Debido a que el NSGA-II requiere como entrada la representación cromosómica, en este trabajo de titulación se consideran tres tipos de representaciones del MST-MO, que luego son evaluadas con métricas de calidad apropiadas con el fin de obtener la mejor alternativa de representación que permita al NSGA-II resolver correctamente y con el más alto rendimiento el problema del MST-MO.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Evaluar representaciones cromosómicas del algoritmo NSGA-II para el problema de Árbol de Expansión Mínima Multiobjetivo (MST-MO) con el objetivo de identificar la mejor alternativa en términos de rendimiento del algoritmo.

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

- Analizar el material bibliográfico relacionado al problema a resolver y alternativas de solución.
- Implementar representaciones cromosómicas del problema del Árbol de Expansión Mínima Multiobjetivo (MST-MO).
- Verificar la implementación de las representaciones cromosómicas utilizando datos de expresión génica.
- Establecer la mejor alternativa de representación cromosómica implementada.

## **CAPÍTULO II. DESARROLLO METODOLÓGICO DE LA INTERVENCIÓN**

El proceso de intervención de este trabajo de titulación se realizó cumpliendo de forma secuencial con cada uno de los objetivos específicos planteados, utilizando a la vez los métodos de investigación científica descritos a continuación:

Rodríguez (2014) indica que el método bibliográfico es muy utilizado en investigaciones cuya finalidad sea obtener datos e información a partir de documentos escritos y no escritos. Por ello, la búsqueda se realizó en medios como libros, documentos científicos e información electrónica.

Asimismo, se utilizó el método analítico, el cual según Maya (2014) es útil cuando se llevan a cabo trabajos de investigación documental, que consiste en revisar en forma separada todo el material necesario para la investigación.

### **2.1. OBJETIVO 1: MATERIAL BIBLIOGRÁFICO**

En el alcance de este primer objetivo se realizó la redacción de la investigación en temas relacionados con la teoría de genes, el problema del MST-MO y el algoritmo NSGA-II, por lo que se empleó el método bibliográfico para la búsqueda de la información. El método analítico se utilizó en la comprensión de cada uno de los elementos que se necesitan para el funcionamiento del NSGA-II y en el entendimiento del problema MST-MO.

#### **2.1.1. PROBLEMA A RESOLVER: MST-MO**

Sea  $G = (V, E, C)$  un gráfico conectado y no dirigido, donde  $V = v_1, \dots, v_n$  es un conjunto finito de nodos y  $E = e_1, \dots, e_1$  es un conjunto finito de bordes. Cada borde vincula los nodos  $e = (i, j)$  tal que  $i \in V, j \in V, i \neq j$ .

Un vector  $c_{ij} = (c_{ij}^1, \dots, c_{ij}^p)$  de  $p$  números reales positivos representa el costo de los bordes. Un árbol de gráfico  $G$  es un subgrafo  $T = (V, E_T, C_T)$  con  $E_T \subseteq E, C_T \subseteq C$ , de manera que  $T$  contiene todos los nodos en  $V$  y los conecta exactamente con  $n - 1$  bordes, de modo que no hay ciclos. Por lo tanto, el problema MST-MO se puede formular como:

$$\min f(T) = f_1(T), \dots, f_p(T), \text{ sujeto a } T \in \Omega \quad [2.1]$$

Donde  $f_k(T) = \sum_{(i,j) \in E_T} c_{ij}^k$  es el  $k$ -ésimo objetivo a minimizar y  $\Omega$  es el conjunto de todos los árboles de expansión del gráfico  $G$ . La imagen de una solución  $T \in \Omega$  es el punto  $z = f(T)$  en el espacio objetivo  $f(\Omega)$  (Párraga-Álava *et al.*, 2017)

### 2.1.2. ALTERNATIVA DE SOLUCIÓN: NSGA-II

La alternativa de solución presentada para el problema del MST-MO sigue los siguientes pasos:

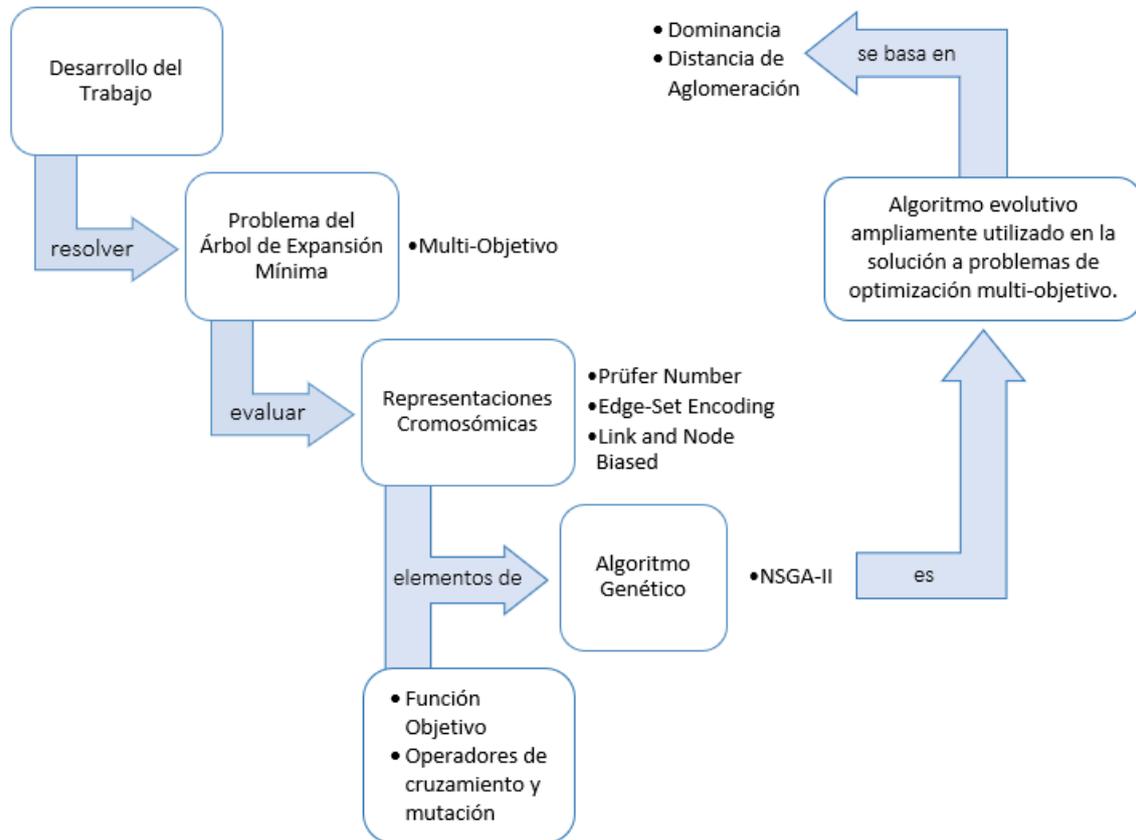
---

#### ALGORITMO 1. Pseudocódigo del NSGA-II

---

- 1:     **Inicializar una población:**
  - 2:             Generar aleatoriamente una población.
  - 3:             Evaluar la aptitud.
  - 4:             Asignar un nivel basado en la dominancia de Pareto – “ordenar”.
  - 5:             Generar una población siguiente:
  - 6:                 Selección mediante un torneo binario.
  - 7:                 Cruzamiento y mutación.
  - 8:     **Para  $i = 1$  hasta el número de generaciones hacer**
  - 9:             **Para la población padre e hijo hacer**
  - 10:                 Asignar un nivel basado en la dominancia de Pareto y ordenar.
  - 11:                 Generar el conjunto de frentes no dominados.
  - 12:                 Agregar soluciones a la siguiente generación, empezando por la primera jerárquica y utilizar el factor de agrupamiento (crowding) en cada frente.
  - 13:             **Fin Para**
  - 14:             Seleccionar los puntos en el frente más bajo y que estén fuera de la distancia del crowding.
  - 15:             Crear la siguiente generación:
  - 16:                 Seleccionar mediante un torneo binario.
  - 17:                 Cruzamiento y mutación.
  - 18:     **Fin Para**
-

Los pasos generales que se siguieron como parte de la intervención del trabajo de titulación se presentan en el siguiente diagrama:



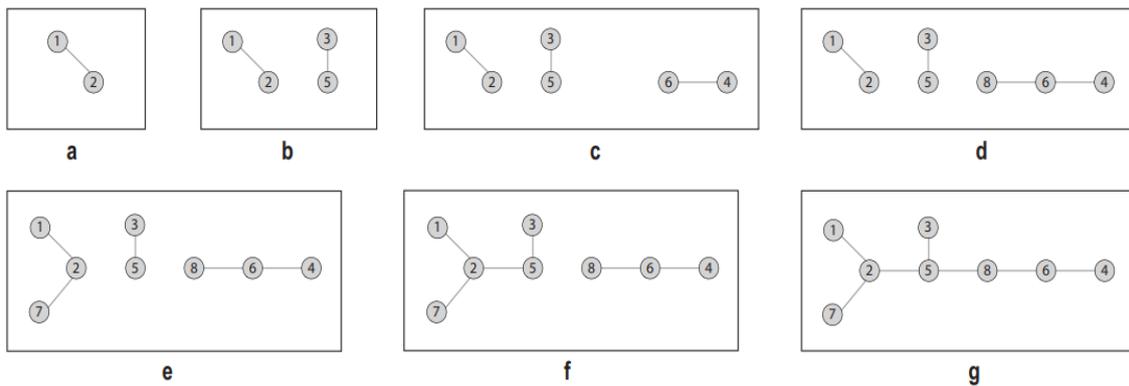
## 2.2. OBJETIVO 2: IMPLEMENTACIÓN

Se elaboraron algunos ejemplos donde se explica el funcionamiento de cada una de las representaciones aplicadas junto con sus operadores evolutivos respectivos al NSGA-II. Por lo cual se hizo uso del método bibliográfico para la búsqueda de información acerca de las representaciones y el método analítico utilizado en la comprensión del funcionamiento de las mismas.

Las tres representaciones del problema del Árbol de Expansión Mínima Multiobjetivo consideradas para ser implementadas en el algoritmo NSGA-II fueron: Prüfer number, link and node biased encoding y edge-set encoding, las cuales se describen a continuación:

### 2.2.1. PRÜFER NUMBER

Esta representación tiene una longitud de  $n - 2$  dígitos para representar un árbol con  $n$  nodos, donde cada dígito es un número entero entre 1 y  $n$ . La figura 2.1 muestra un ejemplo del proceso de esta codificación, en la cual la secuencia de Prüfer (2, 5, 6, 8, 2, 5) corresponde a un árbol de expansión de un grafo de ocho nodos (Párraga-Álava *et al.*, 2017).



**Figura 2.1.** Construcción del árbol de expansión a partir de la secuencia de Prüfer (2, 5, 6, 8, 2, 5).

**Fuente:** (Párraga-Álava *et al.*, 2017).

La construcción del árbol a partir del número Prüfer es un proceso conocido como decodificación y sigue los siguientes pasos:

---

#### ALGORITMO 2. Pseudocódigo de decodificación de Prüfer Number

---

- 1: **Inicializar** código Prüfer  $P$  con  $n - 2$  dígitos. Todos los números de nodo que no están incluidos en  $P$  se pueden usar para la construcción del árbol (son elegibles).
  - 2: **Para**  $i = 1$  hasta que no queden dígitos en  $P$  **hacer**
  - 3:     Sea  $x$  el nodo elegible con el valor más bajo no incluido en  $P$
  - 4:     Sea  $y$  el primer dígito a la izquierda de  $P$ .
  - 5:     Agregar el enlace  $(x, y)$  al árbol.
  - 6:     Designar  $x$  como valor no elegible
  - 7:     Quitar el primer dígito  $y$  de  $P$ .
  - 8:     **Si**  $y$  no aparece en ningún otro lugar del código Prüfer restante, **entonces**
  - 9:         Designar  $y$  como elegible.
  - 10: **Fin Para**
  - 11: **Si** no quedan más dígitos en  $P$ , **entonces**
  - 12:     Hay exactamente dos números,  $r$  y  $s$ , que son elegibles.
  - 12:     Agregar el enlace  $(r, s)$  al árbol.
-

### 2.2.2. LINK AND NODE BIASED ENCODING (LNB)

Según Rothlauf (2006) la idea básica de la codificación LNB es codificar la importancia de un nodo o enlace a partir de ponderaciones que se calculan tanto para los nodos como para los enlaces. Al aplicar esta idea a las codificaciones de árboles, la matriz de peso-distancia que define los pesos de distancia entre dos nodos cualquiera, está sesgada de acuerdo con la importancia de los nodos o enlaces.

Es por ello que el genotipo (individuo) de la codificación LNB posee pesos tanto para los nodos como para todos los enlaces posibles, utilizando un sesgo específico del nodo y otro del enlace. De acuerdo con esto, la representación LNB tiene por lo tanto una longitud de  $l = n(n + 1)/2$  elementos, siendo la función para la ponderación de nodos y enlaces la siguiente:

$$d'_{ij} = d_{ij} + P_1 b_{ij} d_{max} + P_2 (b_i + b_j) d_{max}, \quad [2.2]$$

Donde:

- $d'_{ij}$  corresponden a los pesos de distancia modificados:  $i, j \in \{0, \dots, n - 1\}$ .
- $b_i \in [0, 1]$  son los pesos de nodo y  $b_{ij} \in [0, 1]$  son los pesos de enlace.
- $d_{max}$  es el valor máximo de  $d_{ij}$
- $P_1$  es el sesgo específico del enlace y  $P_2$  el sesgo específico del nodo.

El algoritmo respectivo utilizado en esta representación para la construcción de los árboles se muestra en el siguiente pseudocódigo:

---

**ALGORITMO 3.** Pseudocódigo del algoritmo de PRIM

---

- 1: **Entrada:**  $G = (V, E)$
  - 2: Conjunto de vértices del árbol a iniciar  $V_t \leftarrow \{v_0\}$
  - 3: Salida del conjunto de aristas del árbol de expansión mínima  $E_t \leftarrow \emptyset$
  - 4: **Para**  $i \leftarrow 1$  hasta  $|V| - 1$  **hacer**
  - 5:       encontrar el peso mínimo del borde  $e^* = (v^*, u^*)$
  - 6:        $V_t \leftarrow V_t \cup \{u^*\}$  entre los bordes  $(v, u)$  tal que  $v$  está en  $V_t$  y  $u$  es en  $V - V_t$
  - 7:        $E_t \leftarrow E_t \cup \{e^*\}$
  - 8: **Fin Para**
  - 9: **Retornar** al paso 3 hasta su convergencia
-

### 2.2.3. EDGE-SET ENCODING

En esta técnica se representa directamente los árboles como conjuntos de bordes, los cuales se pueden implementar en una matriz cuyas entradas son los pares de nodos que definen cada borde. La figura 2.2 muestra un ejemplo de un árbol de expansión con 12 nodos y un conjunto de bordes que lo representa (Raidl & Julstrom, 2003).

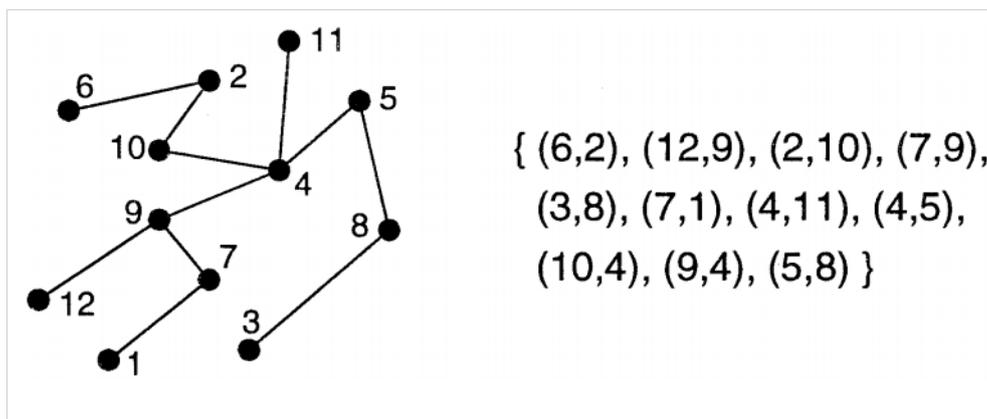


Figura 2.2. Árbol de expansión con 12 nodos y un conjunto de bordes que lo representa.

Fuente: (Raidl & Julstrom, 2003).

En esta representación se utiliza un algoritmo que genera árboles de expansión al azar para la población inicial de genotipos diversos que el NSGA-II requiere. Por lo tanto, el algoritmo que se utilizó fue el de KruskalRST, el cual representó también la base del operador de recombinación. La recombinación de dos árboles de expansión primarios combina sus conjuntos de bordes y deriva un nuevo árbol de expansión aleatorio de esta unión (Raidl & Julstrom, 2003).

El algoritmo utilizado para la inicialización de la población en esta representación se muestra en el siguiente pseudocódigo:

---

#### ALGORITMO 4. Pseudocódigo del algoritmo de KruskalRST

---

- 1: **Procedimiento** KruskalRST ( $V, E$ ):
  - 2:  $T \leftarrow \emptyset, A \leftarrow E$  //  $E$  es el conjunto de aristas disponibles  $e_{i,j}$
  - 3: **Mientras**  $|T| < |V| - 1$  **hacer**
  - 4:     elige una arista  $\{(uv)\} \in A$  al azar
  - 5:      $A \leftarrow A - \{(e_{u,v})\}$
  - 6:     **Si**  $u$  y  $v$  aún no están conectados en  $T$ , **entonces**
  - 7:          $T \leftarrow T \cup \{(e_{u,v})\}$
  - 8: **Retornar a**  $T$
-

La codificación de las representaciones se realizó utilizando bibliotecas del lenguaje de programación R versión 3.4.0. Empleando para este fin, el IDE RStudio en las versiones 1.1.463 y 1.2.1335, el cual está especialmente indicado para el análisis estadístico y gráfico (Pérez, 2015).

El método bibliográfico también fue aplicado en este objetivo para la búsqueda de documentación acerca de las funciones y librerías del lenguaje R que se utilizaron por cada representación.

### 2.3. OBJETIVO 3: PRUEBAS

Los valores para los enlaces entre cada par de nodos en las representaciones implementadas se obtuvieron a partir de las bases de datos que a continuación se describen, las cuales corresponden a datos de expresión génica diferenciados por distancia euclidiana y correlación de Pearson:

- **Arabidopsis Thaliana:** Corresponde a datos de expresión génica de una especie de planta nativa de Europa, Asia, y el noroeste de África. Muestra los valores de expresión génica de 138 genes sobre ocho puntos de tiempo (15 min, 30 min, 60 min, 90 min, 3 h, 6 h, 9 h, y 24 h) (Maulik *et al.*, 2008; Reymond *et al.*, 2000).
- **Yeast Cell Cycle:** Corresponde a datos del ciclo celular de la levadura. La matriz de expresión muestra la fluctuación de los niveles de expresión en 17 puntos de tiempo de 384 genes que son reguladores del ciclo celular (Cho *et al.*, 1998; Maulik *et al.*, 2008).
- **Yeast Cell Sporulation:** Esta base de datos también contiene información de levadura. Muestra los niveles de expresión de 474 genes que están diferencialmente expresados en 7 puntos de tiempo durante el proceso de esporulación de la levadura (Chu *et al.*, 1998; Maulik *et al.*, 2008).
- **Human Fibroblasts Serum:** Presenta datos de expresión génica de células fibroblastos humanas. En total se presentan 517 genes que están diferencialmente expresados considerando 13 puntos de tiempo (Iyer *et al.*, 1999; Maulik *et al.*, 2008).

## 2.4. OBJETIVO 4: MEJOR ALTERNATIVA

Para evaluar el rendimiento de las representaciones implementadas en el NSGA-II, se utilizaron los siguientes elementos luego de obtener las gráficas de las fronteras de Pareto de las representaciones por cada base de datos:

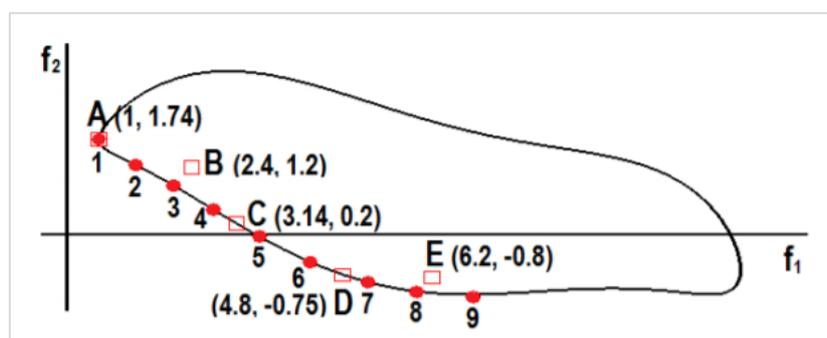
### 2.4.1. MÉTRICA DE DESEMPEÑO MULTI OBJETIVO

El indicador de Hipervolumen ( $HV$ ) se utilizó como métrica de desempeño multiobjetivo, el cual según Zitzler & Thiele (1999) es una métrica unaria que mide el volumen (área en este caso) en el espacio de la función objetivo cubierto por los miembros de un conjunto aproximado (frontera de Pareto ( $FP$ )).

El  $HV$  para cada solución  $C^i \in FP$  computa un área (en caso de dos objetivos)  $a_{C^i}$  respecto a un punto de referencia  $W$ . La union de todas las áreas  $a_{C^i}$  forma el hipervolumen de una frontera de Pareto aproximada ( $FP$ ):

$$HV(FP) = area \left( \bigcup_{i=1}^{|FP|} a_{C^i} \right) \quad [2.3]$$

Esta métrica requiere un punto de referencia correspondiente habitualmente a los peores valores posibles de cada función objetivo en donde los valores más altos de  $HV$  implican una mejor frontera de Pareto aproximada ( $FP$ ).



**Figura 2.3.** Ejemplo para métrica HV, donde las soluciones no dominadas se muestran como cuadrados, y las soluciones elegidas de la frontera de Pareto se muestran como círculos rellenos.

Fuente: (Fernández, 2019).

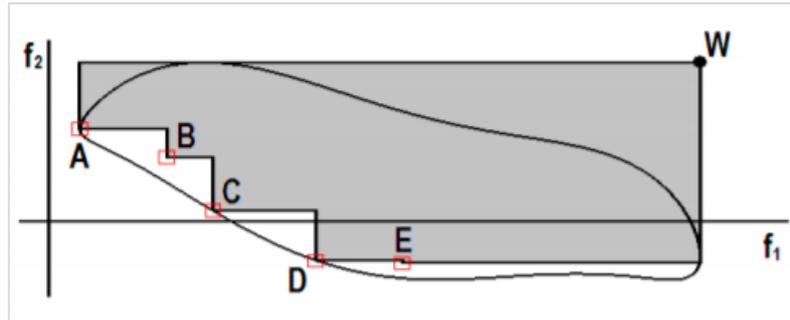


Figura 2.4. Hipervolumen encerrado por las soluciones no dominadas.

Fuente: (Fernández, 2019).

Tomando como ejemplo la región de color gris de la figura 2.4, la cual tiene un conjunto de soluciones no dominadas. Para resolver este tipo de problemas se debe construir hipercubos con cada una de las soluciones de la frontera de Pareto estableciendo un punto de referencia  $W$  y luego construyendo un vector con los peores puntos de cada solución, convirtiéndose así en el nuevo punto de referencia  $W$ , así sucesivamente con toda la región. Después se encuentra la unión de todos los hipercubos y se calcula su hipervolumen con la ecuación 2.3. A continuación, se muestra el proceso del cálculo del hipervolumen tomando como referencia el punto (11, 3) de la figura 2.3 para el valor de  $W$  (Fernández, 2019).

$$\begin{aligned}
 HV(FP) &= (11 - 6.2) \times (3 - (-0.8)) + (6.2 - 4.8) \times (3 - (-0.75)) + (4.8 - 3.14) \\
 &\quad \times (3 - 0.2) + (3.14 - 2.4) \times (3 - 1.2) + (2.4 - 1) \times (3 - 1.74) \\
 &= 31.23
 \end{aligned}$$

## 2.4.2. ANÁLISIS ESTADÍSTICOS

Para verificar si había diferencia significativa entre las representaciones se realizaron pruebas estadísticas por medio de los siguientes test:

### 2.4.2.1. TEST DE FRIEDMAN

La prueba de Friedman compara promedios poblacionales cuando se trabaja con muestras relacionadas, esta prueba es una alternativa de ANOVA de un factor

cuando los datos son dependientes, donde se toman dos variables (una dependiente, y otra independiente) para comparar sus medidas y concluir si en estas existen grupos similares o diferentes, puede considerarse como una comparación entre la mediana de varios grupos (Amat, 2016).

#### **2.4.2.2. TEST DE NEMENYI**

Luego de utilizar Friedman para probar las igualdades de medida entre los grupos, y haber determinado si existen diferencias, la prueba de Nemenyi permite identificar subconjuntos homogéneos de medidas que no se diferencian entre sí. La fórmula para el número máximo de comparaciones que puede hacer para N grupos es:  $(N * (N - 1))/2$ . Además, la fórmula para calcular la tasa de error para todo el conjunto de comparaciones es  $1 - (1 - \alpha)^C$ . Donde, alfa corresponde al nivel de significancia para una comparación única, y C es igual al número de comparaciones (Frost, 2019; Jakob, 2019).

El proceso de intervención de esta propuesta no demandó aportes económicos durante su ejecución, debido a que, ya se contaba con el equipo a utilizar para la exploración de las representaciones antes señaladas.

## CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA

### 3.1. OBJETIVO 1: MATERIAL BIBLIOGRÁFICO

#### 3.1.1. INTRODUCCIÓN SOBRE LA TEORÍA DE GENES

La **expresión génica** corresponde al proceso mediante el cual la información codificada en un gen se transcribe en uno o varios niveles de ARN funcionales, mediante la cuantificación de tales niveles. La cuantificación facilita la creación de perfiles de expresión de cada célula, permitiendo que la comparación entre ellos se convierta en una excelente herramienta para poder responder a un gran número de inquietudes biológicas (Wang, 2014).

Para poder efectuar el estudio de la expresión génica se han utilizado distintas tecnologías que han facilitado la obtención, almacenamiento y análisis de tal información biológica. Tal como es el caso de la tecnología basada en microarrays que obtiene los datos de la respuesta simultánea de miles de genes frente a una serie de condiciones de laboratorio (Mestizo, 2015).

El resultado final en el funcionamiento de los microarrays de expresión es una matriz en la que las filas corresponden a genes y las columnas corresponden a condiciones, muestras o puntos de tiempo. El contenido de la matriz es el nivel de expresión de cada gen bajo cada condición. Estos niveles pueden ser absolutos, relativos o normalizados (Párraga Álava, 2017; Pontes *et al.*, 2006).

Una de las tareas en el análisis de datos de expresión génica es la de descubrir grupos de genes que intervienen en una misma función celular o que están regulados de la misma manera. En este sentido, el **clustering** es una de las técnicas de clasificación no supervisada más importante cuyo objetivo es el de formar grupos o clases de datos llamados clústeres, de tal forma que los datos de un mismo grupo comparten una serie de características y similitudes mientras que los datos de grupos distintos tienen mayores diferencias (Pontes *et al.*, 2006).

De acuerdo con Párraga Álava (2017) para cuantificar la similitud entre los datos de un mismo grupo, es común el uso de medidas de distancia, ya que la distancia mide la diferencia entre dos puntos sobre la base de sus valores de atributos. Las medidas de distancia y/o similitud más utilizadas para realizar el clustering son: la distancia euclidiana y la correlación de Pearson.

### 3.1.2. MST Y MST-MO (EJEMPLOS)

Un árbol de expansión está compuesto por todos los vértices y algunas de las aristas de un grafo conexo no dirigido. Un grafo puede tener muchos árboles de expansión. En la siguiente figura se puede observar que el grafo dado posee 3 árboles de expansión:

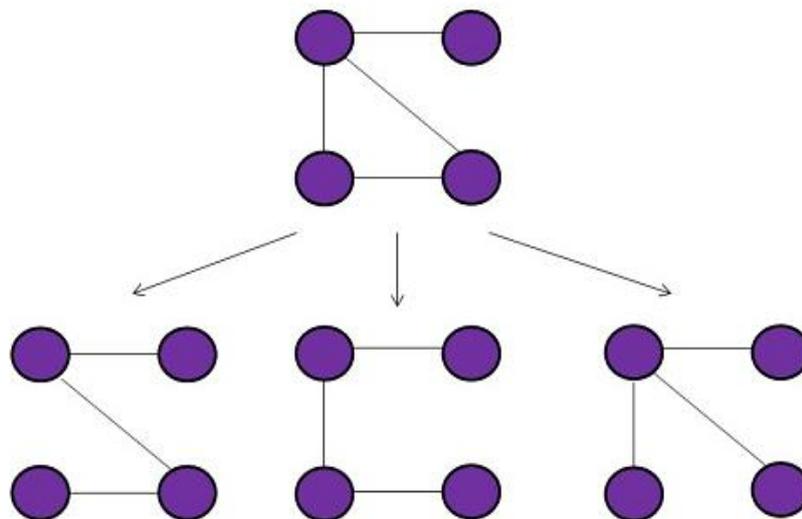
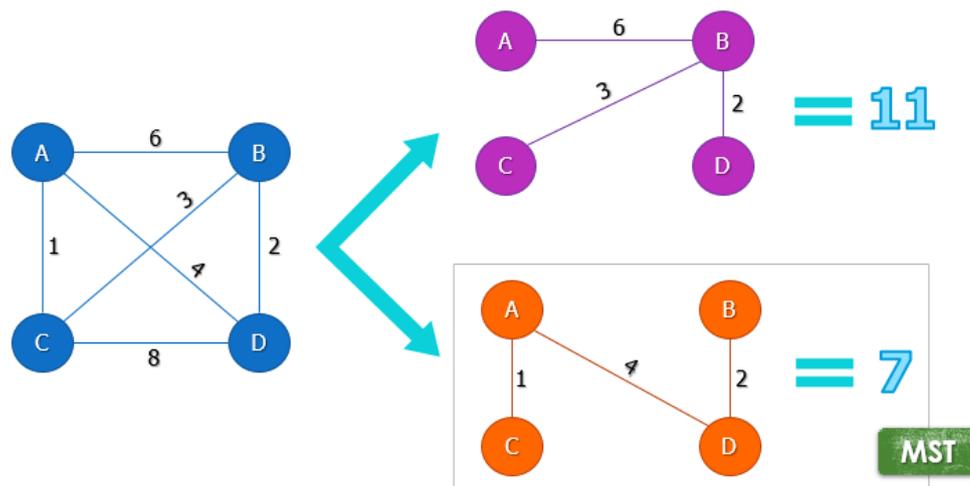


Figura 3.1. Ejemplo de Árbol de expansión.

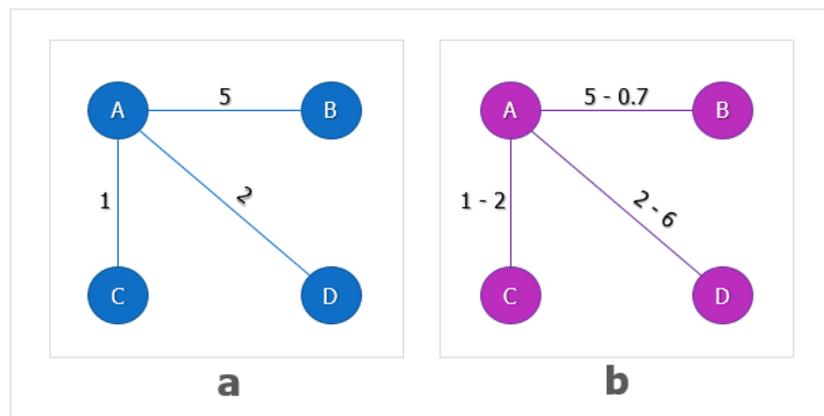
Fuente: (Arias, 2012).

El problema del Árbol de Expansión Mínima (MST) consiste en buscar un árbol que abarque todos los vértices del grafo, con suma de pesos de aristas mínimo, tal y como se aprecia en el ejemplo de la figura 3.2 en la cual el MST es el árbol con suma total de pesos igual a 7. Todo grafo tiene un bosque recubridor mínimo. El problema de hallar el MST de puede ser resuelto con varios algoritmos, los más conocidos son Kruskal y Prim. Para que sea un MST válido el número de aristas debe ser igual al número de vértices menos uno (Arias, 2012).



**Figura 3.2.** Ejemplo de Árbol de Expansión Mínima (MST) a partir de un grafo conectado completo.  
**Fuente:** (Elaboración propia, 2019).

Con el fin de entender de mejor manera el funcionamiento del problema del MST-MO se hizo una comparación a base de ejemplo entre el problema del MST simple y el problema de enfoque multiobjetivo (MST-MO):



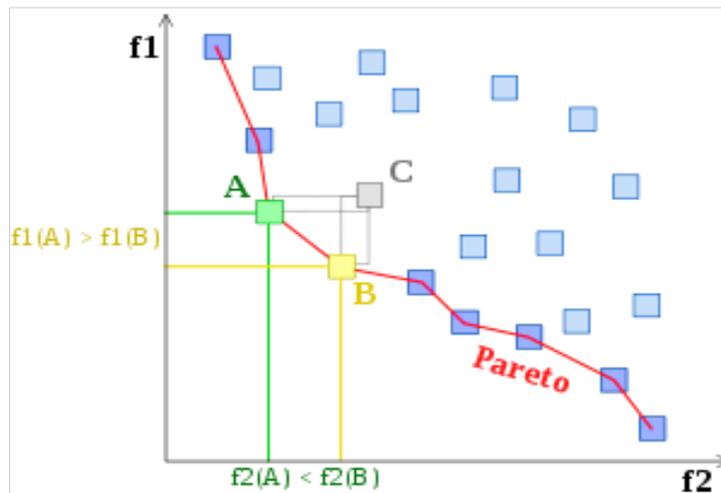
**Figura 3.3.** Comparación entre un MST (a) y un MST-MO (b).  
**Fuente:** (Elaboración propia, 2019).

### 3.1.3. FUNCIONAMIENTO GENERAL DEL NSGA-II

Los algoritmos genéticos pertenecen a las metaheurísticas poblaciones que pueden ser utilizadas para la solución de problemas mono u multiobjetivo. En el caso de problemas multiobjetivo se debe tener en cuenta el concepto de dominancia. Se dice que una solución  $x_1$  domina a otra solución  $x_2$  cuando:

- a) La solución  $x_1$  no es peor que  $x_2$  en ninguno de los objetivos y
- b) La solución  $x_1$  es estrictamente mejor que  $x_2$  en al menos un objetivo.

Si alguna de estas dos condiciones no se cumple, entonces  $x_1$  no domina a  $x_2$  (Rojas *et al.*, 2016).



**Figura 3.4.** Concepto de dominancia aplicado a un ejemplo de minimización. **A** y **B** son No-Dominadas entre sí, ya que ninguna domina a otra, pero ambas dominan a **C** y pertenecen a la frontera de Pareto óptimo.

**Fuente:** (SCI2S, 2014).

El concepto de dominancia visualizado como ejemplo en la figura anterior se utiliza para encontrar un conjunto de soluciones no-dominadas, denominadas ranking o frente de Pareto (FP). El NSGA-II pertenece a la gama de EAs que permite encontrar este set de soluciones. Dado que ninguna de las soluciones en el conjunto no-dominado es absolutamente mejor que las otras, entonces cualquiera de estas constituye una solución aceptable (Rojas *et al.*, 2016).

En el NSGA-II, a partir de una población de padres  $P_t$  ( $N$  individuos) generados aleatoriamente, se crea una población descendiente  $Q_t$  ( $N$  individuos). Las dos poblaciones constituyen el conjunto  $R_t$  de tamaño  $2N$  (Figura 3.5). Para crear la población  $Q_t$  primero se calcula la función objetivo de la población  $P_t$  y se hace su clasificación por dominancia; es decir, la población empieza a ser construida con el mejor frente no dominado ( $F_1$ ) o mejor ranking ( $R_1$ ), continuando con las soluciones del segundo frente ( $F_2$ ) y así sucesivamente tanto problemas de minimización como de maximización. Luego se aplican los operadores de selección, cruce y mutación (Rojas *et al.*, 2016).

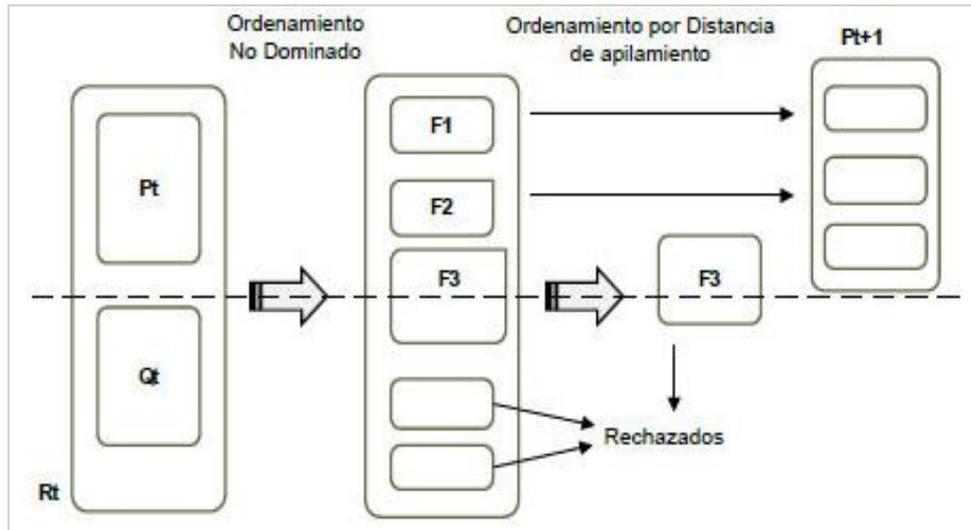


Figura 3.5. Diagrama de selección de individuos en el NSGA-II.  
Fuente: (Rojas *et al.*, 2016).

En las operaciones de selección se implementa elitismo, lo cual significa que la densidad de la población en la vecindad de un individuo se utiliza como medida secundaria de adaptación; es decir, si al ser comparados dos individuos, ambos forman parte del mismo frente, se prefiere entonces la solución con menos soluciones alrededor. Asegurando de esta forma, diversidad de la población en cada  $FP$  al obtener sólo los mejores individuos en cada generación (Rojas *et al.*, 2016).

Ya que  $R_t$  es de tamaño  $2N$  y tan solo se necesitan  $N$  individuos para la población descendiente, se deben descartar  $N$  configuraciones prefiriendo siempre los individuos pertenecientes a la mejor frontera de Pareto. Este proceso termina hasta cuando el algoritmo logra obtener las mejores soluciones o hasta que se alcance el número máximo de iteraciones establecidas (Rojas *et al.*, 2016).

### 3.1.3.1. FUNCIÓN OBJETIVO

Es un tipo especial de función que cuantifica la optimalidad de una solución. Se traduce en un cromosoma óptimo para que sus bases sean combinadas con cualquier otra técnica para la producción de una nueva generación que sea mejor a las anteriores (Jimenez & Chivata, 2017).

### 3.1.3.2. OPERADORES EVOLUTIVOS

Según Jimenez & Chivata (2017) dentro del algoritmo genético existen y pueden ser utilizadas diferentes técnicas matemáticas y computacionales, también llamadas operadores genéticos que son aplicadas para el paso de una generación a la siguiente. Algunos de estos operadores son:

- **Operador de selección:** Los operadores de selección son los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Se ha de otorgar un mayor número de oportunidad de reproducción a los individuos más aptos con respecto a la evolución y calificación inicial. Es importante resaltar que se debe dar una posibilidad de reproducirse a los individuos menos aptos ya que estos contienen material genético útil en el proceso de reproducción (Jimenez & Chivata, 2017).
- **Operador de cruzamiento:** Es una herramienta utilizada para introducir e intercambiar información entre cromosomas. La idea principal del cruce es tomar los individuos previamente seleccionados y obtener una descendencia que comparta sus genes. Al compartir las características de los individuos correctamente es posible que estos sean los causantes de proporcionar una solución aproximada (Jimenez & Chivata, 2017).
- **Operador de mutación:** La mutación tiene como objetivo lograr que alguno de los genes de un individuo varíe su valor de forma aleatoria. Tal y como sucede con los otros operadores, existen diversos operadores de mutación (Jimenez & Chivata, 2017; Sosa *et al.*, 2014).

En el cuadro siguiente se muestra y detalla los tipos de operadores respectivos que se utilizaron por cada representación:

**Cuadro 3.1.** Listado de los operadores evolutivos utilizados por cada representación en el NSGA-II

Operadores Evolutivos	Tipos de Operadores Evolutivos	Conceptos de los tipos de Operadores	Representación en las que se usaron los elementos
Inicialización	KruskalRST	El propósito de los algoritmos de inicialización es crear una solución inicial imparcial. Raidl y Julstrom (2003) citado por Rothlauf (2006) recomendaron el uso de KruskalRST que se basa en el algoritmo de Kruskal simple. En contraste con el algoritmo de Kruskal, KruskalRST (Random Spanning Tree) elige los bordes no de acuerdo con sus pesos de distancia correspondientes sino de manera aleatoria.	Edge-set encoding
Selección	Torneo Binario	En la técnica de selección binaria por torneo se comparan dos individuos que son escogidos al azar y en cada comparación se emplea un indicador que permite saber cuál está mejor adaptado (Zhang & Ma, 2015). Para identificar los individuos ganadores de cada torneo, se elige al individuo perteneciente al ranking de Pareto inferior y en el caso de un empate, se selecciona el que tiene la mayor distancia de amontonamiento (Párraga-Álava <i>et al.</i> , 2017).	Prüfer number, link and node biased encoding y edge-set encoding
	n puntos	Es una generalización del crossover de un solo punto. Los n puntos de cruce se identifican aleatoriamente, después de lo cual se dividen a lo largo de esos puntos y se reúnen, y posteriormente se alternan entre los padres (Lim <i>et al.</i> , 2017).	Prüfer number
Crossover	Aritmético	Los cromosomas que tienen un valor real o una representación de punto flotante se someten a un crossover aritmético. Este cruce crea un nuevo alelo en cada posición del gen de los descendientes. El valor de los nuevos alelos para los descendientes se encuentra entre los valores de los alelos principales y se calcula utilizando la siguiente ecuación para el primer descendiente: $Offspring1 = w * parent1 + (1 - w) * Parent2$ y la ecuación: $Offspring2 = (1 - w) * parent1 + w * Parent2$ para el segundo. Donde $w$ es el factor de peso constante que se utiliza para calcular nuevos valores (Furqan <i>et al.</i> , 2017)	Link and node biased encoding
	Cruce de KruskalRST*	Cuando se usa esta variante, en un primer paso todos los bordes $(E1 \cap E2)$ se incluyen en la descendencia $Toff$ . Luego se completa $Toff$ aplicando KruskalRST a los bordes restantes $(E1 \cup E2) \setminus (E1 \cap E2)$ (Rothlauf, 2006).	Edge-set encoding

Mutación	Aleatoria	En la mutación aleatoria controlada, se selecciona una posición aleatoria del cromosoma y el índice de nodo correspondiente se reemplaza por un índice de nodo elegido al azar (Párraga-Álava <i>et al.</i> , 2017).	Prüfer number
	Uniforme	Este operador de mutación reemplaza el genoma, ya sea inferior o superior con destino al azar. Esto puede ser usado para los genes enteros y de coma flotante. Por ejemplo, dada una población $p = \{x_1, \dots, x_m\}$ , el individuo mutado será: $p' = \{x_1, \dots, x_k^1, \dots, x_m\}$ donde $x_k^1$ es el individuo alterado con un valor entre rangos mínimos y máximos, usando una distribución uniforme (Jimenez & Chivata, 2017; Sosa <i>et al.</i> , 2014)	Link and node biased encoding
	Aleatoria	El operador de mutación reemplaza aleatoriamente un borde en el árbol de expansión. Este reemplazo se puede realizar de dos maneras diferentes. La primera variante del operador de mutación elige aleatoriamente un borde que no está presente en $T$ y lo incluye en $T$ . Luego, se elige y elimina aleatoriamente un borde del ciclo ("inserción antes de la eliminación"). La segunda variante primero elimina aleatoriamente un borde de $T$ y luego conecta los dos componentes conectados desunidos utilizando un borde aleatorio que no está presente en $T$ ("eliminación antes de la inserción") (Rothlauf, 2006). La variante que se escogió para ser utilizada en la mutación de la representación edge-set encoding es la de <b>eliminación antes de la inserción</b> , debido a una mejor comodidad en la codificación.	Edge-set encoding

Fuente: (Elaboración propia, 2019).

### 3.2. OBJETIVO 2: IMPLEMENTACIÓN

Los siguientes son ejemplos de las representaciones aplicadas al algoritmo NSGA-II junto con sus operadores evolutivos respectivos:

Para obtener los valores que tendrían los costos en las aristas al generar los grafos de estos ejemplos, se determinó un número de 3 variables por individuo con el fin de obtener las respectivas matrices de peso-distancia creadas a partir de los siguientes datos:

**Cuadro 3.2.** Datos de ejemplo para las variables establecidas

Población	Variable 1	Variable 2	Variable 3
	<i>Edad</i>	<i>Peso (Kg)</i>	<i>Estatura</i>
Individuo 1	23	65	1,70
Individuo 2	22	50	1,65
Individuo 3	21	70	1,80
Individuo 4	22	47	1,55
Individuo 5	19	68	1,72
Individuo 6	18	52	1,60

Fuente: (Elaboración propia, 2019).

En el cálculo de los valores correspondientes al costo del objetivo 1 y 2 o bien valores de las aristas en los grafos, se utilizaron las siguientes fórmulas de distancia euclidiana (ecuación 3.1) y de correlación de Pearson (ecuación 3.2) de las que se hace mención en la subsección 3.1.1 de este trabajo:

- **Distancia euclidiana:**

$$D_E(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad [3.1]$$

- **Correlación de Pearson:**

$$P(\vec{x}, \vec{y}) = \frac{(\sum_{i=1}^d x_i \cdot y_i) - \frac{(\sum_{i=1}^d x_i) \cdot (\sum_{i=1}^d y_i)}{d}}{\sqrt{\left(\sum_{i=1}^d (x_i)^2 - \frac{(\sum_{i=1}^d x_i)^2}{d}\right) \cdot \left(\sum_{i=1}^d (y_i)^2 - \frac{(\sum_{i=1}^d y_i)^2}{d}\right)}} \quad [3.2]$$

Donde  $d$  corresponde en este ejemplo al número de variables de cada individuo. Y al aplicar dichas fórmulas en los datos del cuadro 3.2, se obtuvieron las siguientes matrices de distancia y asociación entre cada uno de sus nodos:

Euclidiana						
Nodos	1	2	3	4	5	6
1		15,03	5,39	18,03	5,00	13,93
2	-		20,03	3,00	18,25	4,47
3	-	-		23,02	2,83	18,25
4	-	-	-		21,21	6,40
5	-	-	-	-		16,03
6	-	-	-	-	-	

**Figura 3.6.** Matriz de Peso Distancia entre los nodos usando la ecuación 3.1.  
Fuente: (Elaboración propia, 2019).

Pearson						
Nodos	1	2	3	4	5	6
1		1,00	1,00	0,99	1,00	1,00
2	-		0,99	1,00	0,98	0,99
3	-	-		0,98	1,00	1,00
4	-	-	-		0,98	0,99
5	-	-	-	-		1,00
6	-	-	-	-	-	

**Figura 3.7.** Matriz de Peso Distancia entre los nodos usando la ecuación 3.2.  
Fuente: (Elaboración propia, 2019).

### 3.2.1. EJEMPLO CON PRÜFER NUMBER

Se ha asignado un total de 6 individuos como parámetro para la población inicial de padres, que de acuerdo con el diagrama de selección de la figura 3.5, dicha población corresponde al conjunto  $Pt = 6$ .

Para generar los grafos de cada individuo con esta representación, cada cromosoma o individuo corresponde a un vector de longitud  $n - 2$  elementos, tal y como se aprecia a continuación:

Individuo 1	=	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	}	<b>Pt</b>
1	2	3	4					
Individuo 2	=	<table border="1"><tr><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	4	3	2	1		
4	3	2	1					
Individuo 3	=	<table border="1"><tr><td>2</td><td>1</td><td>4</td><td>3</td></tr></table>	2	1	4	3		
2	1	4	3					
Individuo 4	=	<table border="1"><tr><td>3</td><td>4</td><td>1</td><td>2</td></tr></table>	3	4	1	2		
3	4	1	2					
Individuo 5	=	<table border="1"><tr><td>1</td><td>3</td><td>4</td><td>2</td></tr></table>	1	3	4	2		
1	3	4	2					
Individuo 6	=	<table border="1"><tr><td>2</td><td>4</td><td>3</td><td>1</td></tr></table>	2	4	3	1		
2	4	3	1					

Figura 3.8. Población Pt (1era generación – ejemplo con Prüfer Number).

Fuente: (Elaboración propia, 2019).

La decodificación desde la secuencia de Prüfer de cada individuo a su árbol de expansión correspondiente se efectúa de la siguiente manera:

**Individuo 1:**

1	2	3	4
---	---	---	---

Se puede observar que la longitud de cada individuo es la misma con 4 elementos enteros por vector, a partir de una longitud total del código Prüfer de  $n = 6$  de acuerdo con  $(n - 2 = 4)$ . Con ello entonces se procedió a identificar al elemento más pequeño que no estuviera incluido en el código de Prüfer, en el caso de este primer individuo, el menor no incluido dentro del total de la longitud de enteros (del 1 al 6) es el 5 y al representarlos como dos conjuntos separados se obtuvo lo siguiente:

$\mathbf{a} = \{1, 2, 3, 4\}$  → Conjunto de los elementos iniciales del código

$\mathbf{b} = \{5\}$  → Conjunto de los menores no incluidos en el código

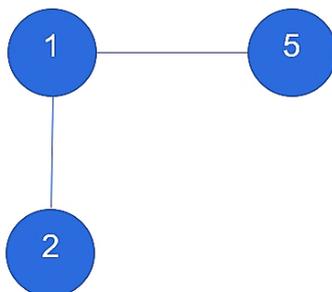


Figura 3.9. Unión de los primeros nodos del Individuo 1 (ejemplo con Prüfer number).

Fuente: (Elaboración propia, 2019).

Para la unión de los nodos se tomaba al primer elemento del conjunto  $\mathbf{a}$ , en este caso al 1, junto con el último elemento del conjunto  $\mathbf{b}$ , o sea el 5. Luego, para poder enlazar a los siguientes elementos se eliminaba el primero del conjunto  $\mathbf{a}$ , es decir, a partir de los conjuntos actualizados:  $\mathbf{a} = \{2, 3, 4\}$ ,  $\mathbf{b} = \{5\}$ , se volvía a identificar al menor no incluido (esta vez de ambos conjuntos). Con lo cual se

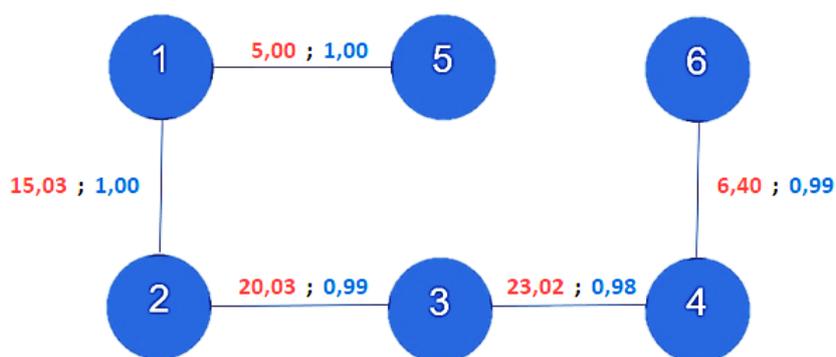
añadía el nuevo menor no incluido en el conjunto  $b$ , quedando como sigue:  $a = \{2, 3, 4\}$ ,  $b = \{5, 1\}$ . En la unión del siguiente nodo, el 2 del conjunto  $a$ , se enlazó con el elemento 1 del conjunto  $b$  que ya estaba en el grafo, quedando de la siguiente manera:



**Figura 3.10.** Unión de los siguientes elementos del Individuo 1 (ejemplo con Prüfer number).

**Fuente:** (Elaboración propia, 2019).

Este proceso se realizó hasta el momento en que el conjunto  $a$  quedara vacío. Al hacer esto, el conjunto  $b$  quedó al final con los siguientes elementos  $\{5, 1, 2, 3\}$ , haciendo entonces la última unión de los dos elementos que faltaban, es decir aquellos que no estaban incluidos en ninguno de los conjuntos, la unión entre el elemento 4 con el 6. Luego de tener generados los grafos, se asignaron los pesos correspondientes en cada arista, estos valores fueron tomados de las matrices de distancias ya realizadas (euclidiana y Pearson). Para este primer individuo, los pesos con su MST-MO respectivo fueron los siguientes:



**Figura 3.11.** Individuo 1 (ejemplo con representación Prüfer number).

**Fuente:** (Elaboración propia, 2019).

La generación de los siguientes grafos se realizó siguiendo el mismo proceso. Se muestra a continuación el resultado de cada individuo (Figura 3.12 a la 3.16):

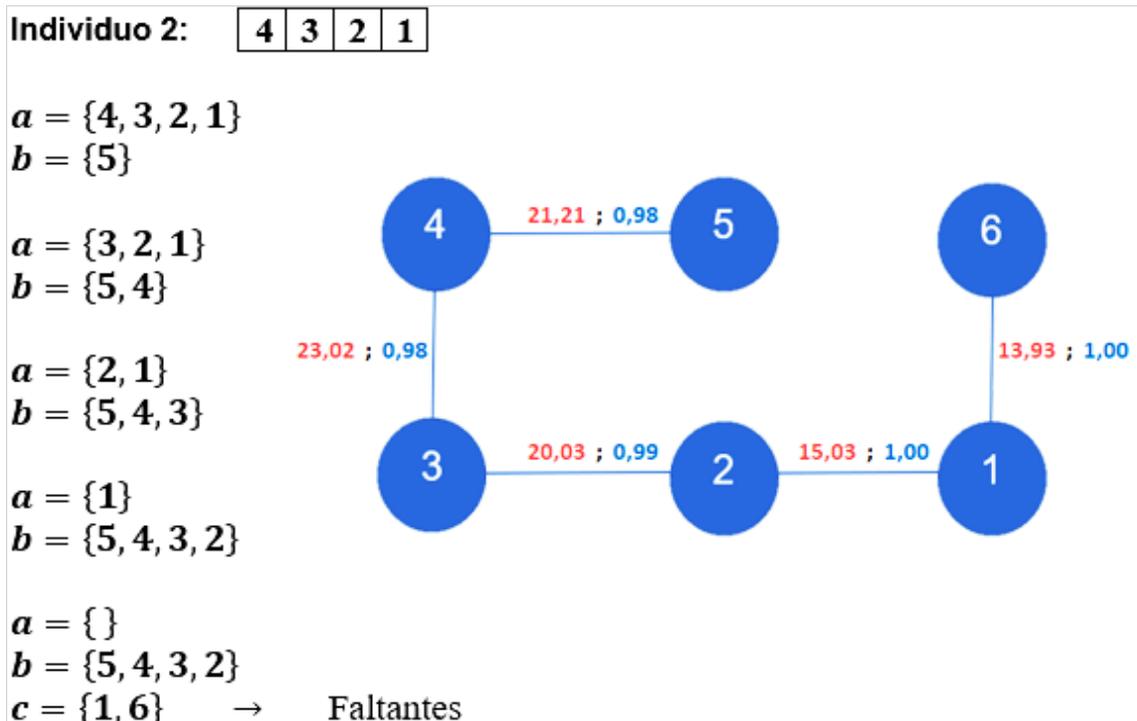


Figura 3.12. Individuo 2 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

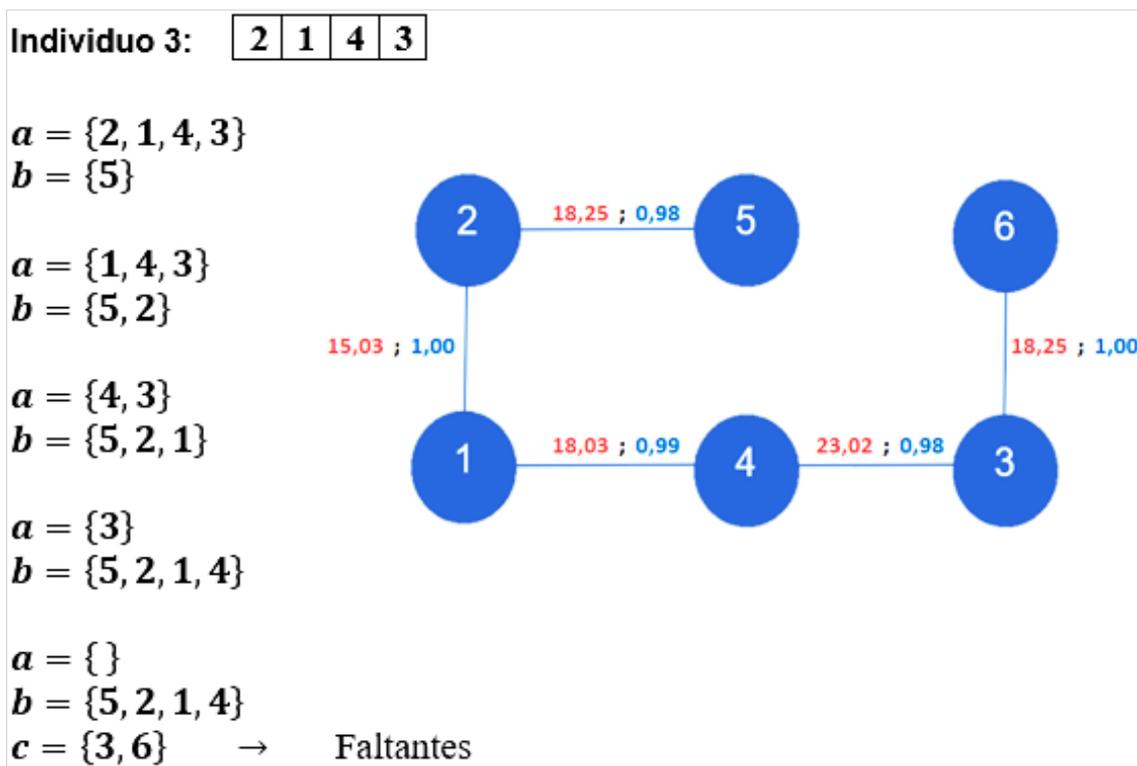


Figura 3.13. Individuo 3 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

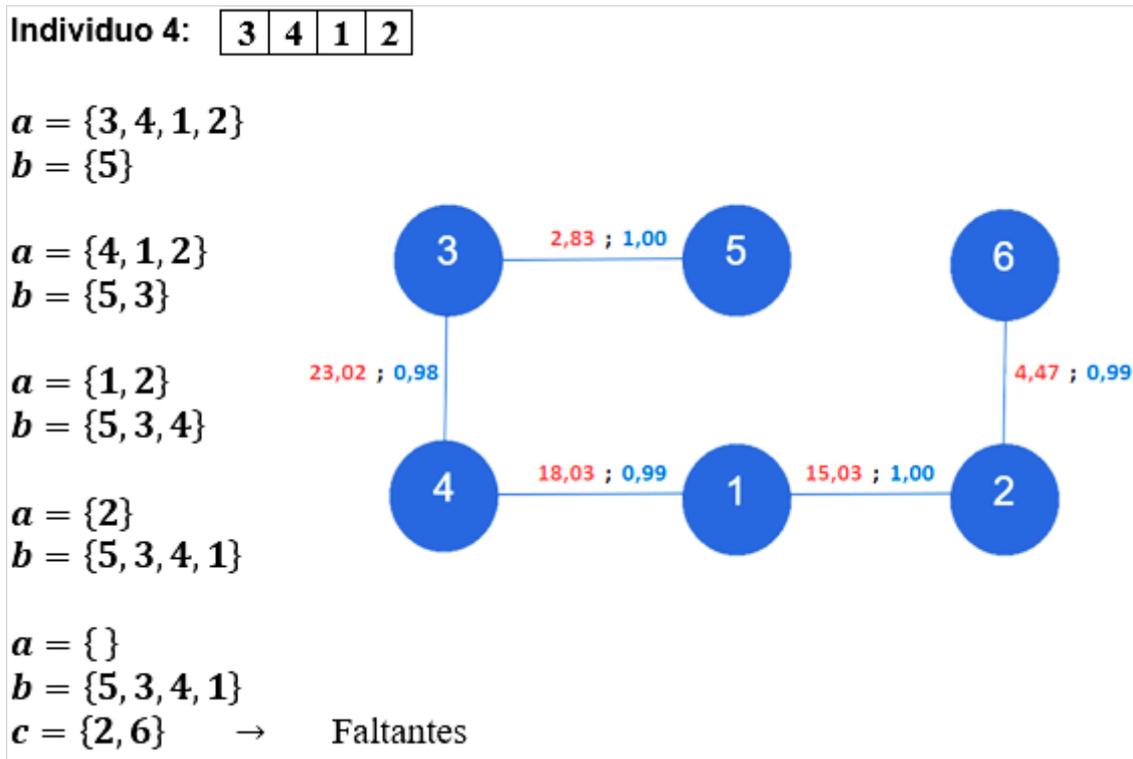


Figura 3.14. Individuo 4 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

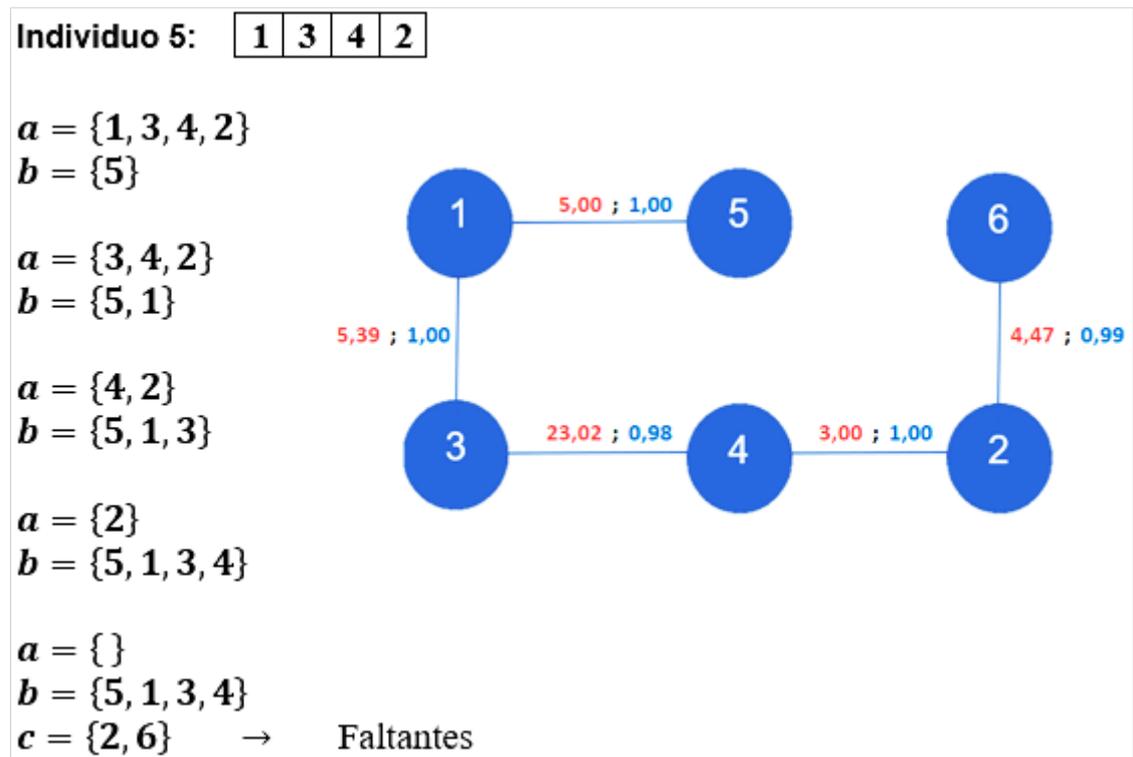


Figura 3.15. Individuo 5 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

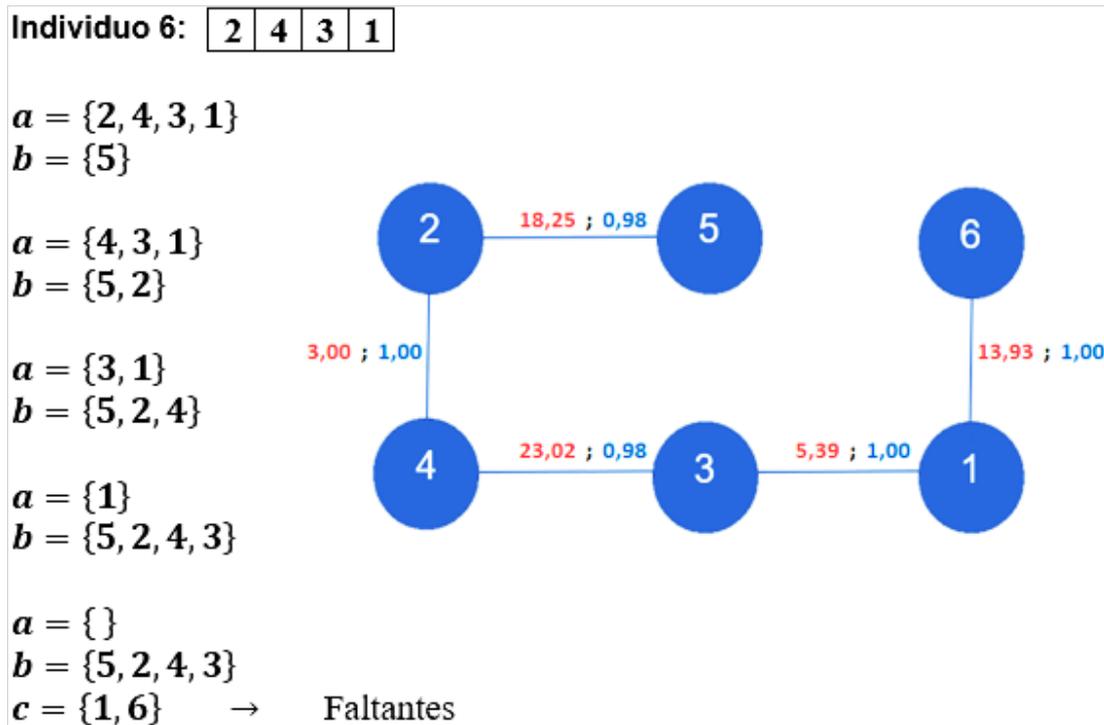


Figura 3.16. Individuo 6 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

### 3.2.1.1. FUNCIÓN OBJETIVO Y DOMINANCIA DE PARETO

Para determinar las funciones objetivo que tendría cada individuo se procedió a sumar cada uno de los pesos de las aristas. Por un lado, los pesos que se obtuvieron de la matriz euclidiana y por otro los que se obtuvieron de la matriz de Pearson tal y como se muestra a continuación:

**Sumatoria de la Función Objetivo 1 del individuo 1:**

$$\sum F1 = 5 + 15.03 + 20.03 + 23.02 + 6.40 = 69.48$$

**Sumatoria de la Función Objetivo 2 del individuo 1:**

$$\sum F2 = 1 + 1 + 0.99 + 0.98 + 0.99 = 4.95$$

La razón de sumar ambos objetivos por separado es debido a que se está trabajando con optimización multiobjetivo, en este caso con dos objetivos, los cuales se necesitan para ser clasificados luego con el ranking Pareto.

Por lo tanto, la sumatoria de todos los pesos por cada individuo fue la siguiente:

**Cuadro 3.3.** Valores de las funciones objetivo población Pt (1era generación – ejemplo con Prüfer number).

Población	F1	F2
Individuo 1	69,48	4,95
Individuo 2	93,22	4,94
Individuo 3	92,58	4,95
Individuo 4	63,38	4,96
Individuo 5	40,88	4,96
Individuo 6	63,59	4,96

**Fuente:** (Elaboración propia, 2019).

Para determinar la dominancia que tendría cada solución sobre las demás, se consideraron las dos reglas ya mencionadas en la figura 3.4 acerca de la dominancia de Pareto. Obteniendo entonces lo siguiente:

**Cuadro 3.4.** Ranking Pareto de la población Pt (1era generación – ejemplo con Prüfer number).

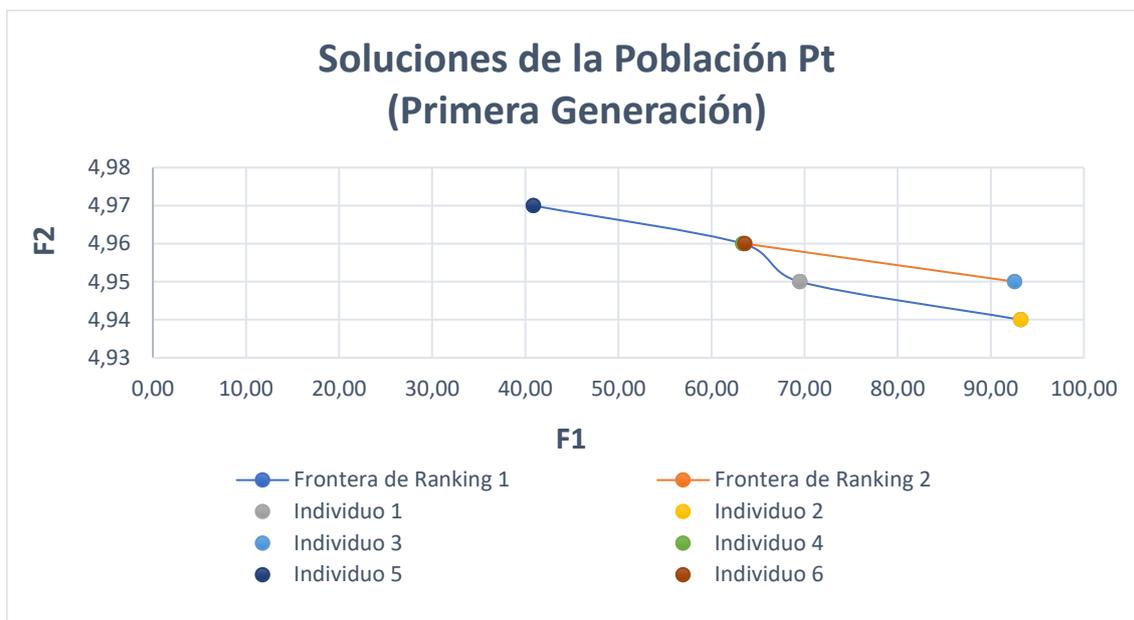
Población	F1	F2	Ranking Pareto
Individuo 1	69,48	4,95	1
Individuo 2	93,22	4,94	1
Individuo 4	63,38	4,96	1
Individuo 5	40,88	4,97	1
Individuo 3	92,58	4,95	2
Individuo 6	63,59	4,96	2

**Fuente:** (Elaboración propia, 2019).

Por ejemplo, al comparar a la solución o individuo 1 con el 2 se tiene: En la columna del primer objetivo (F1) el individuo 1 es mejor que el 2 debido a que el problema con el que se está trabajando es de minimización y el individuo 1 es el que tiene menor valor con 69.48. Sin embargo, al considerar la comparación con los valores del segundo objetivo (F2), se puede observar en el cuadro anterior que precisamente ese individuo 1 no es el menor o mejor. Debido a ello, en este caso se considera que dichas soluciones (1 y 2) fueron NO comparables entre sí.

Para saber a qué ranking pertenecería cada individuo, se debía comparar de la misma manera con el resto de individuos. Si en algún caso un individuo era dominado totalmente por otro, entonces se separaba en grupos, es decir a los dominados de los no dominados. Al separarse en grupos se volvían a comparar entre los mismos de cada grupo para buscar si entre ellos aparecían nuevamente

individuos dominados. La clasificación de rankings de 1 a n se daba de acuerdo al número de grupos que surgían, en este caso sólo fueron dos, mismos que se pueden visualizar en el gráfico 3.1, colocando en el primer ranking a los que no fueron comparables (no dominados) y en el segundo a los dominados.



**Gráfico 3.1.** Fronteras de Pareto de la población Pt (1era Generación - ejemplo con Prüfer number).

**Fuente:** (Elaboración propia, 2019).

### 3.2.1.2. OPERADORES DE SELECCIÓN, CRUCE Y MUTACIÓN

El siguiente paso fue proceder a la selección de los individuos para su posterior cruzamiento y mutación, es decir para generar la siguiente población. Al tener aquí más de un objetivo, quien cumple con el papel de “función objetivo” para cada individuo es el ranking Pareto.

Lo que se hizo en la selección binaria por torneo fue escoger de manera aleatoria a dos individuos de la población para poder ser comparados de acuerdo a su ranking. Siguiendo con el ejemplo, los individuos seleccionados para el primer torneo fueron el 4 y el 5, pero como los dos pertenecían al mismo ranking o frente de Pareto, se determinó como ganador al individuo 5 debido a que este individuo tenía menos soluciones alrededor que el individuo 4, de haber estado en diferentes rankings se hubiera preferido al de menor valor.

En el segundo torneo aleatoriamente se escogieron al individuo 4 y el 6, y así como en el criterio del torneo anterior, se estableció como ganador al 4.

Para el cruzamiento de los individuos se estableció una probabilidad  $p \leq 0.80$  y para la mutación  $p \leq 0.05$ . El operador de cruce se aplicó a cada par de padres en cada selección, mientras que el de mutación se aplicó a cada individuo al final de cada generación, es decir cuando ya la población ( $R_t$ ) estuviera completa; ambos generados de forma aleatoria.

En esta primera pareja ganadora de individuos se obtuvo una probabilidad de cruce de 0.04. Y ya que este valor estaba dentro de los parámetros establecidos, se procedió entonces a identificar los respectivos puntos de corte para el posterior cruzamiento en la pareja, quedando de la siguiente manera:

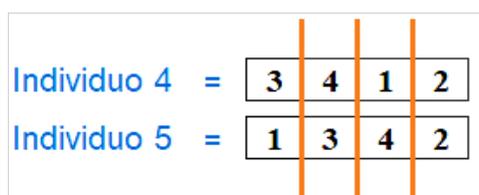


Figura 3.17. Puntos de corte para el cruzamiento en individuos de la representación Prüfer number.

Fuente: (Elaboración propia, 2019).

Para determinar qué elementos en los puntos de corte se iban a cruzar y cuáles no, se generaron cuatro números aleatorios (cantidad de acuerdo con la longitud del vector) entre los valores 0 y 1, tal como lo indica el tipo de cruzamiento n-puntos de esta representación. Permitiendo realizar el cruce entre los cortes sólo a aquellos elementos con  $p \geq 0.50$ , y como todos en este caso fueron mayores a 0.50, se realizó entonces el cruce de los elementos de la siguiente manera:

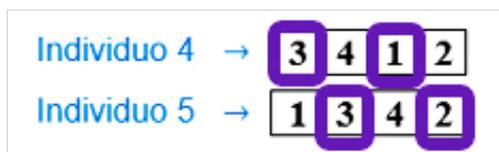


Figura 3.18. Proceso de Cruzamiento en los Individuos 4 y 5.

Fuente: (Elaboración propia, 2019).

Obteniendo a los siguientes individuos como los primeros descendientes: **Individuo 7** = {3, 3, 1, 2} e **Individuo 8** = {1, 4, 4, 2}, los cuales ahora serían

parte de la población  $Q_t$  (Figura 3.5). Luego, se generaron otros 4 hijos más (Figura 3.19) siguiendo el proceso ya descrito de selección y cruzamiento. Dado a que todos los valores para el cruce se establecieron menores a 0.80, a continuación, se muestra a los siguientes descendientes que se generaron para completar la población  $R_t$  de padres ( $P_t$ ) e hijos ( $Q_t$ ):

Individuo 9 =	2	2	4	4	}	<b>Q<sub>t</sub></b>
Individuo 10 =	1	1	3	3		
Individuo 11 =	1	3	4	1		
Individuo 12 =	4	3	2	2		

**Figura 3.19.** Población  $Q_t$  (1era generación – ejemplo con Prüfer number).  
Fuente: (Elaboración propia, 2019).

Cabe mencionar que cuando los valores para el cruce o mutación no se encuentran dentro de los parámetros iniciales establecidos, tan sólo se pasan los padres seleccionados tal y cómo están. Una vez completada la población  $R_t$ , se aplicó el operador de mutación en cada individuo, obteniendo lo siguiente:

- Individuo 1 = 0,68
- Individuo 2 = 0,08
- Individuo 3 = 0,68
- Individuo 4 = 0,95
- Individuo 5 = 0,24
- Individuo 6 = 0,26
- Individuo 7 = 0,78
- **Individuo 8 = 0,02**
- Individuo 9 = 0,89
- Individuo 10 = 0,89
- Individuo 11 = 0,96
- Individuo 12 = 0,35

En esta primera generación de individuos el único que sufrió mutación fue el octavo individuo. Dicha mutación es tan sólo el cambio de uno de los elementos del individuo por otro elemento generado aleatoriamente. El valor de reemplazo para este ejemplo debía estar entre el 1 y el 6 debido a que la longitud total de los individuos o del código Prüfer es  $n = 6$ .

La posición en que surgió el cambio también fue generada aleatoriamente sin sobrepasar la longitud del vector, tal y como se muestra a continuación:



Figura 3.20. Proceso de mutación en el Individuo 8.

Fuente: (Elaboración propia, 2019).

Luego de estos pasos, en este ejemplo la población de descendientes  $Q_t$  quedó con sus respectivos grafos de la siguiente forma:

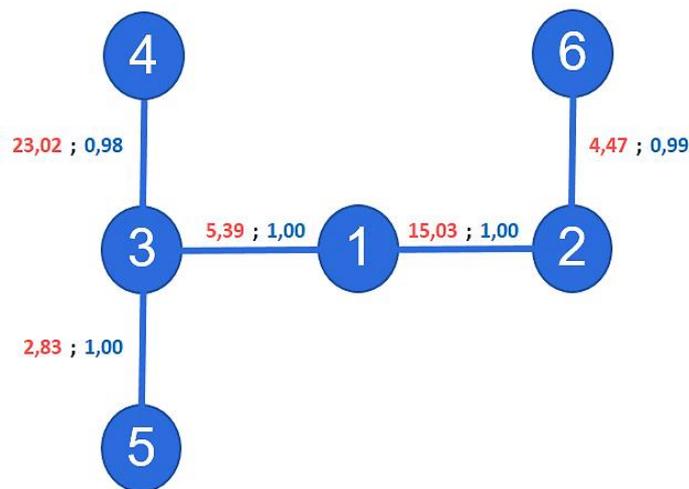


Figura 3.21. Individuo 7 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

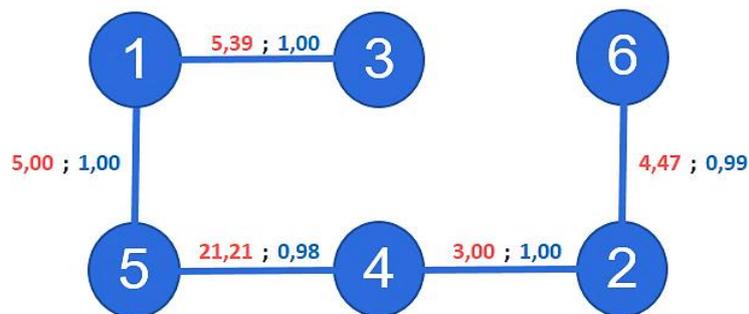


Figura 3.22. Individuo 8 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

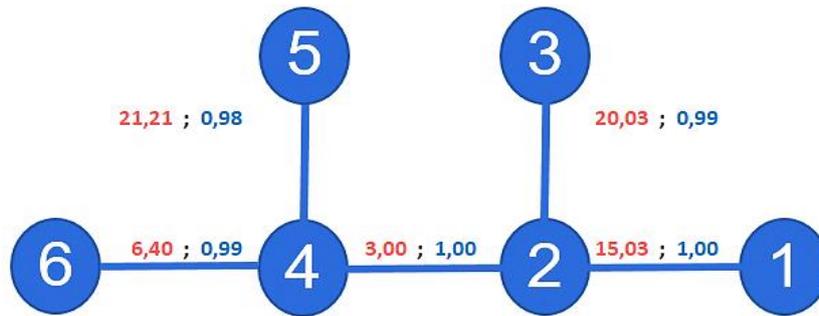


Figura 3.23. Individuo 9 (ejemplo con representación Prüfer number).  
Fuente: (Elaboración propia, 2019).

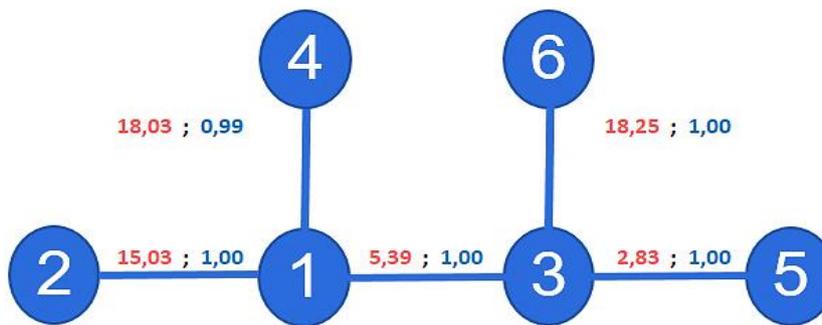


Figura 3.24. Individuo 10 (ejemplo con representación Prüfer number).  
Fuente: (Elaboración propia, 2019).

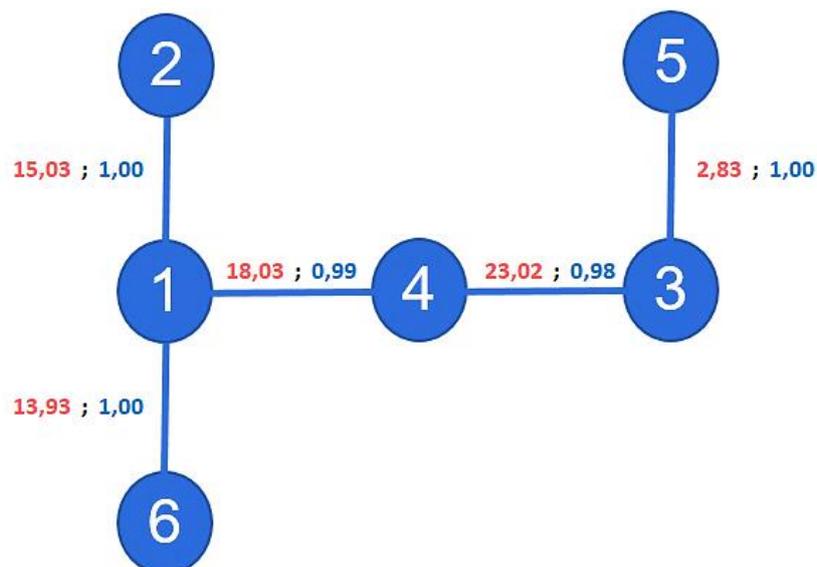
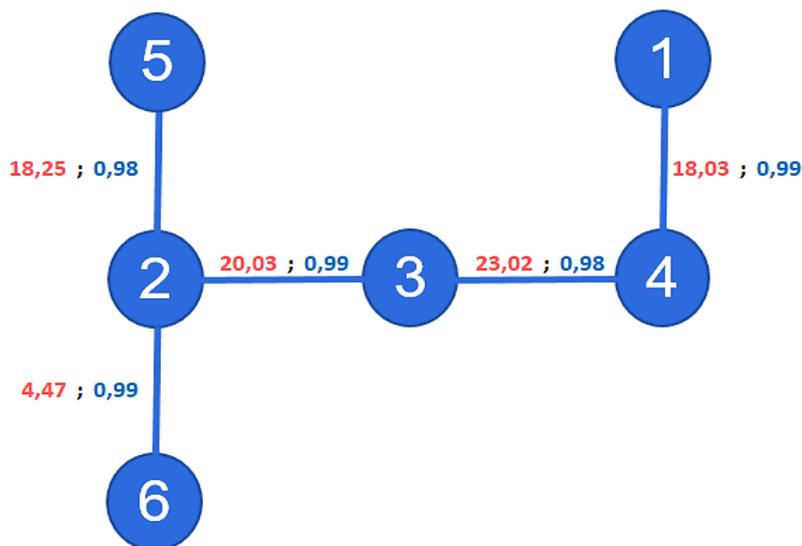


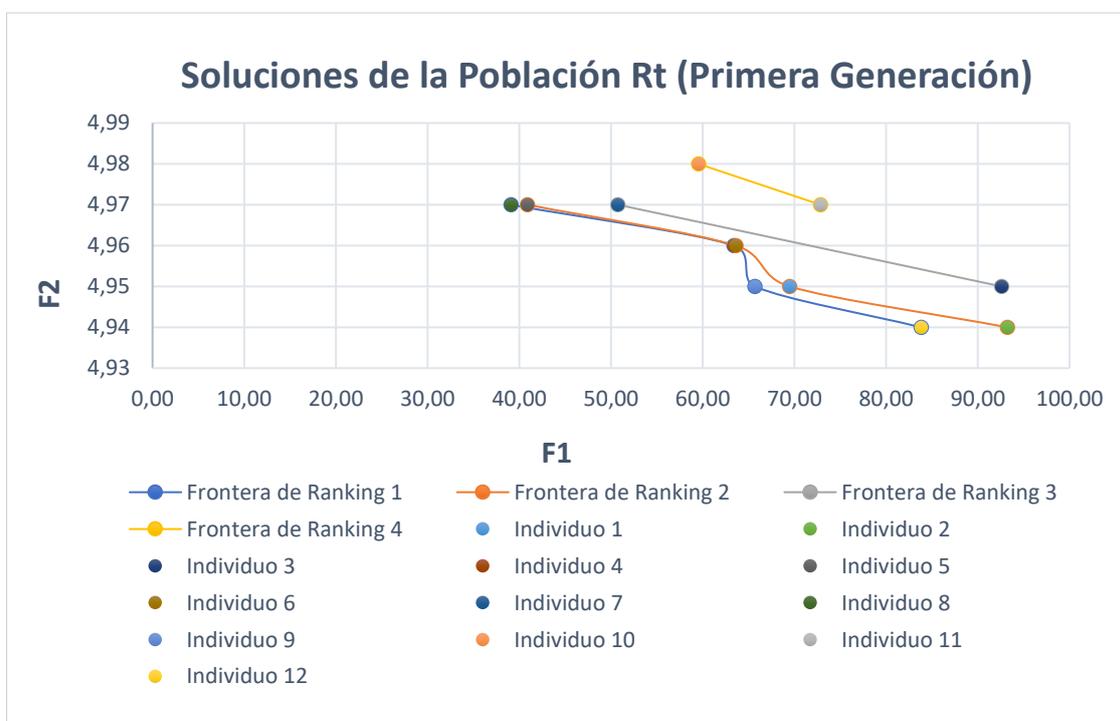
Figura 3.25. Individuo 11 (ejemplo con representación Prüfer number).  
Fuente: (Elaboración propia, 2019).



**Figura 3.26.** Individuo 12 (ejemplo con representación Prüfer number).

Fuente: (Elaboración propia, 2019).

El ranking Pareto de la población  $Rt$  junto con sus fronteras respectivas (en el que se incluye además el ranking de la población  $Pt$  ordenado y actualizado) es el siguiente:



**Gráfico 3.2.** Fronteras de Pareto de la población  $Rt$  (1era Generación - ejemplo con Prüfer Number).

Fuente: (Elaboración propia, 2019).

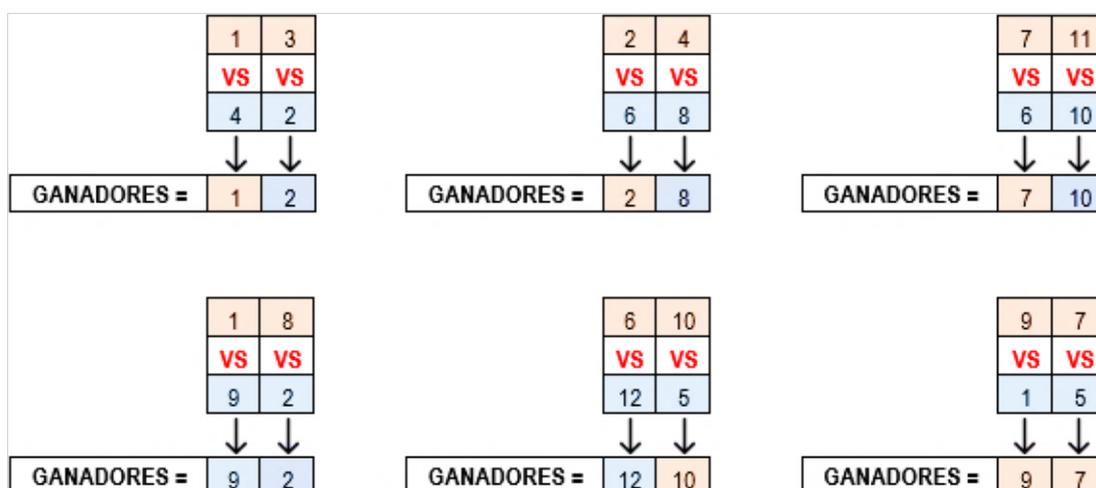
**Cuadro 3.5.** Ranking Pareto de la población  $R_t$  (1era generación – ejemplo con Prüfer number).

Población	F1	F2	Ranking Pareto
Individuo 4	63,38	4,96	1
Individuo 8	39,07	4,97	1
Individuo 9	65,67	4,95	1
Individuo 12	83,80	4,94	1
Individuo 1	69,48	4,95	2
Individuo 2	93,22	4,94	2
Individuo 5	40,88	4,97	2
Individuo 6	63,59	4,96	2
Individuo 3	92,58	4,95	3
Individuo 7	50,74	4,97	3
Individuo 10	59,53	4,98	4
Individuo 11	72,84	4,97	4

Fuente: (Elaboración propia, 2019).

### 3.2.1.3. RESULTADOS EN LA SIGUIENTE GENERACIÓN

Se muestran a continuación los individuos ganadores en la selección aleatoria de la población  $R_t$  (1era generación) para su posterior cruzamiento, y los descendientes obtenidos en esta 2da generación (Figura 3.28):



**Figura 3.27.** Individuos ganadores de la primera generación (ejemplo con Prüfer number).

Fuente: (Elaboración propia, 2019).

La probabilidad de cruce para cada par de padres ganadores fue la siguiente:

- 1er Cruce = 0,05
- 2do Cruce = 0,44
- 3er Cruce = 0,71
- 4to Cruce = 0,64
- 5to Cruce = 0,19
- 6to Cruce = 0,17

La nueva población  $R_t$  fue la siguiente:

Individuo 13 =	<table border="1"><tr><td>1</td><td>3</td><td>3</td><td>1</td></tr></table>	1	3	3	1	Individuo 19 =	<table border="1"><tr><td>2</td><td>3</td><td>4</td><td>1</td></tr></table>	2	3	4	1
1	3	3	1								
2	3	4	1								
Individuo 14 =	<table border="1"><tr><td>4</td><td>2</td><td>2</td><td>4</td></tr></table>	4	2	2	4	Individuo 20 =	<table border="1"><tr><td>4</td><td>2</td><td>2</td><td>4</td></tr></table>	4	2	2	4
4	2	2	4								
4	2	2	4								
Individuo 15 =	<table border="1"><tr><td>4</td><td>5</td><td>2</td><td>2</td></tr></table>	4	5	2	2	Individuo 21 =	<table border="1"><tr><td>4</td><td>1</td><td>2</td><td>3</td></tr></table>	4	1	2	3
4	5	2	2								
4	1	2	3								
Individuo 16 =	<table border="1"><tr><td>1</td><td>3</td><td>4</td><td>1</td></tr></table>	1	3	4	1	Individuo 22 =	<table border="1"><tr><td>1</td><td>3</td><td>3</td><td>2</td></tr></table>	1	3	3	2
1	3	4	1								
1	3	3	2								
Individuo 17 =	<table border="1"><tr><td>3</td><td>1</td><td>1</td><td>3</td></tr></table>	3	1	1	3	Individuo 23 =	<table border="1"><tr><td>2</td><td>3</td><td>4</td><td>2</td></tr></table>	2	3	4	2
3	1	1	3								
2	3	4	2								
Individuo 18 =	<table border="1"><tr><td>1</td><td>3</td><td>3</td><td>2</td></tr></table>	1	3	3	2	Individuo 24 =	<table border="1"><tr><td>3</td><td>2</td><td>1</td><td>4</td></tr></table>	3	2	1	4
1	3	3	2								
3	2	1	4								

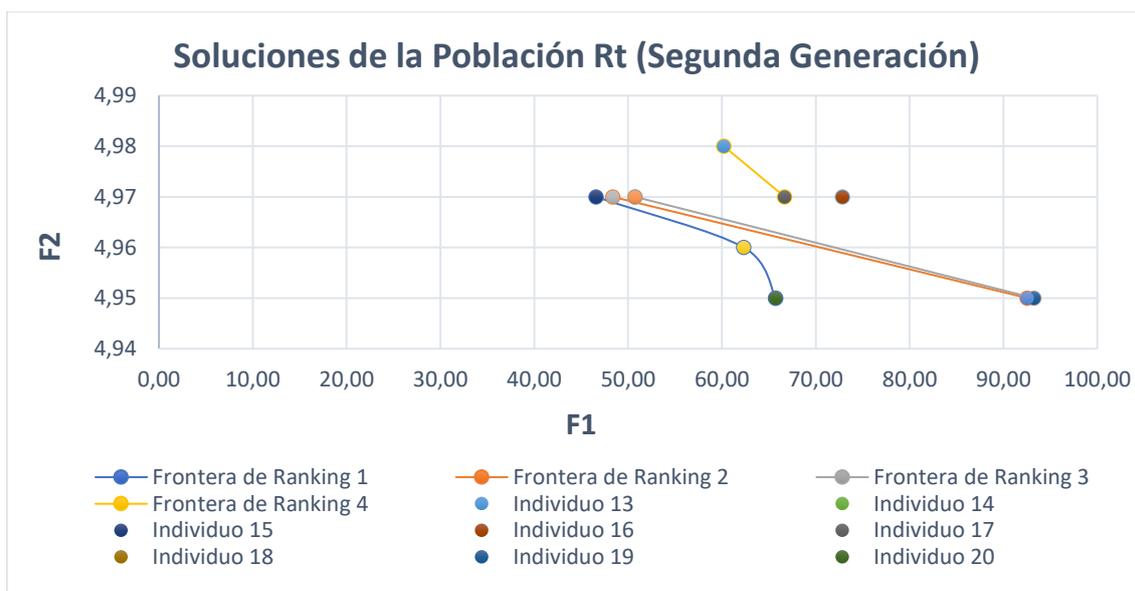
**Figura 3.28.** Población  $R_t$  (2da generación – ejemplo con Prüfer number).

Fuente: (Elaboración propia, 2019).

Dependiendo del ranking se podían ir eliminando soluciones de la anterior generación. La probabilidad de mutación para cada individuo fue la siguiente:

- Individuo 1 = 0,74
- Individuo 2 = 0,72
- Individuo 3 = 0,46
- Individuo 4 = 0,46
- Individuo 5 = 0,61
- Individuo 6 = 0,12
- Individuo 7 = 0,73
- Individuo 8 = 0,99
- Individuo 9 = 0,11
- Individuo 10 = 0,69
- Individuo 11 = 0,37
- Individuo 12 = 0,27

Y los resultados del ranking Pareto junto con sus fronteras fueron:



**Gráfico 3.3.** Fronteras de Pareto de la población  $R_t$  (2da generación - ejemplo con Prüfer number).

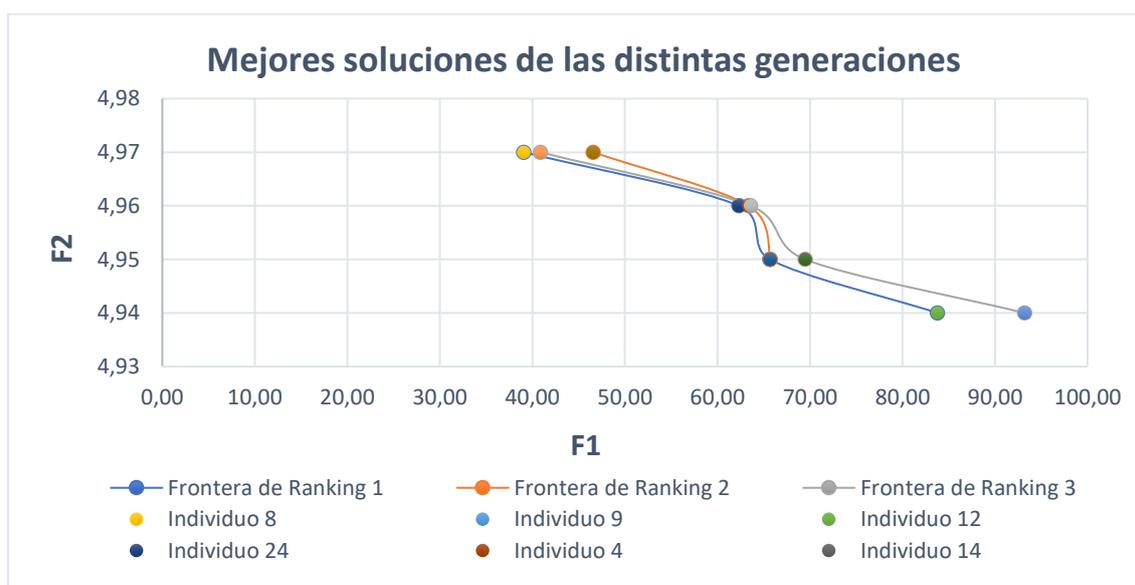
Fuente: (Elaboración propia, 2019).

**Cuadro 3.6.** Ranking Pareto de la población Rt (2da generación – ejemplo con Prüfer number).

Población	F1	F2	Ranking Pareto
Individuo 14	65,71	4,95	1
Individuo 15	46,58	4,97	1
Individuo 20	65,71	4,95	1
Individuo 24	62,32	4,96	1
Individuo 21	92,55	4,95	2
Individuo 23	48,36	4,97	2
Individuo 18	50,74	4,97	3
Individuo 19	93,25	4,95	3
Individuo 22	50,74	4,97	3
Individuo 13	60,20	4,98	4
Individuo 17	66,69	4,97	4
Individuo 16	72,84	4,97	5

Fuente: (Elaboración propia, 2019).

La idea en seguir obteniendo generaciones, es poder ir seleccionando sólo a los mejores individuos de cada generación, reemplazando a las peores soluciones encontradas por las mejores, teniendo en cuenta que no se excedan el número establecido de la población total. Para efectos del ejemplo en esta ocasión se ha asumido que este ha sido el ciclo total de generaciones para proceder a realizar la selección de las mejores soluciones obtenidas hasta el momento, comparando individuos de ambas generaciones y descartando aquellas que pertenecían al peor ranking. El resultado final entre ambas generaciones fue el siguiente:

**Gráfico 3.4.** Fronteras de Pareto de las mejores soluciones obtenidas (ejemplo con Prüfer number).

Fuente: (Elaboración propia, 2019).

**Cuadro 3.7.** Ranking Pareto de las mejores soluciones obtenidas (ejemplo con Prüfer number).

Población	F1	F2	Ranking Pareto
Individuo 8	39,07	4,97	1
Individuo 9	65,67	4,95	1
Individuo 12	83,80	4,94	1
Individuo 24	62,32	4,96	1

**Fuente:** (Elaboración propia, 2019).

Tal y como se muestra el cuadro y gráfico anterior, las soluciones han sido seleccionadas tanto de la población original de padres en la primera generación, como también de la población de descendientes de la segunda generación. Esto con el fin de asegurar que se obtuvieran sólo las mejores soluciones de cada ciclo (soluciones con ranking 1) sin haber excedido el número total de la población  $Rt = 12$  individuos. Teniendo como mejores soluciones hasta este punto a los siguientes individuos:

Individuo 8	=	1	5	4	2
Individuo 9	=	2	2	4	4
Individuo 12	=	4	3	2	2
Individuo 24	=	3	2	1	4

**Figura 3.29.** Mejores soluciones obtenidas en dos generaciones (ejemplo con Prüfer number).

**Fuente:** (Elaboración propia, 2019).

Si bien en la figura anterior se aprecia que no se excedió del total de la población  $Rt$  con tamaño  $2N$ , tampoco se completó las  $N$  configuraciones que se debería tener en el resultado final, en casos como este lo que se realiza es generar las soluciones o individuos faltantes (2 faltantes en este caso) de forma aleatoria.

### 3.2.2. EJEMPLO CON LINK AND NODE BIASED ENCODING (LNB)

En esta representación, los parámetros propios del NSGA-II como tamaño de la población, tipo de selección de los individuos, probabilidad de cruce, probabilidad de mutación y clasificación de acuerdo al ranking Pareto fueron los mismos que se establecieron en la representación Prüfer number. Por lo tanto, se procede a detallar de manera más directa el funcionamiento de la representación.

### 3.2.2.1. PARÁMETROS PARA LA CONSTRUCCIÓN DE LOS ÁRBOLES

En el siguiente cuadro se muestran los genotipos aleatorios que se usaron de ejemplo para los pesos de los nodos y de enlaces:

**Cuadro 3.8.** Genotipos para la población Pt (ejemplo con representación LNB).

Población Pt	Pesos de Nodo	Pesos de enlace
Individuo 1	0.7, 0.5, 0.2, 0.8	0.1, 0.6, 0.2, 0.1, 0.9, 0.3
Individuo 2	0.96, 0.17, 0.82, 0.71	0.69, 0.47, 0.76, 0.42, 0.16, 0.53
Individuo 3	0.35, 0.82, 0.59, 0.01	0.89, 0.04, 0.16, 0.8, 0.01, 0.81
Individuo 4	0.01, 0.33, 0.23, 0.43	0.4, 0.68, 0.52, 0.15, 0.98, 0.48
Individuo 5	0.17, 0.16, 0.54, 0.75	0.55, 0.08, 0.34, 0.47, 0.92, 0.55
Individuo 6	0.94, 0.9, 0.03, 0.55	0.05, 0.11, 0.73, 0.69, 0.45, 0.28

**Fuente:** (Elaboración propia, 2019).

Se estableció un total de 4 nodos por individuo con  $P_1 = 1$  y  $P_2 = 1$ . Los valores correspondientes a los pesos de nodos y enlaces de cada individuo fueron colocados en una matriz de pesos  $b_{ij}$  tal y como se puede apreciar en la siguiente imagen con los datos del individuo 1:

		0,70	0,50	0,20	0,80	
		1	2	3	4	
b <sub>ij</sub>	1		0,10	0,60	0,20	0,70
	2	0,10		0,10	0,90	0,50
	3	0,60	0,10		0,30	0,20
	4	0,20	0,90	0,30		0,80

**Figura 3.30.** Pesos de nodo y enlace del Individuo 1 (ejemplo con la representación LNB).

**Fuente:** (Elaboración propia, 2019).

Debido a que en la función para ponderaciones de nodo y enlace se necesitaban datos de una matriz de distancia, se tomaron los valores establecidos en la matriz euclidiana (Figura 3.6) y Pearson (Figura 3.7) respectivamente.

Luego se reemplazaron en la ecuación 2.2 los pesos del individuo 1 y los valores de las matrices de distancia originales (Figura 3.6 y 3.7), realizando primero el proceso con los datos de la matriz euclidiana:

$$d'_{1,2} = 15.03 + (0.10 * 23.02) + (0.70 + 0.50) * 23.02 = 44.96$$

$$d'_{1,3} = 5.39 + (0.60 * 23.02) + (0.70 + 0.20) * 23.02 = 39.92$$

$$d'_{1,4} = 18.03 + (0.20 * 23.02) + (0.70 + 0.80) * 23.02 = 57.16$$

$$d'_{2,3} = 20.03 + (0.10 * 23.02) + (0.50 + 0.20) * 23.02 = 38.45$$

$$d'_{2,4} = 3.00 + (0.90 * 23.02) + (0.50 + 0.80) * 23.02 = 53.64$$

$$d'_{3,4} = 23.02 + (0.30 * 23.02) + (0.20 + 0.80) * 23.02 = 52.95$$

Y luego con los datos de la matriz de Pearson:

$$d'_{1,2} = 1.00 + (0.10 * 1.00) + (0.70 + 0.50) * 1.00 = 2.29$$

$$d'_{1,3} = 1.00 + (0.60 * 1.00) + (0.70 + 0.20) * 1.00 = 2.50$$

$$d'_{1,4} = 0.99 + (0.20 * 1.00) + (0.70 + 0.80) * 1.00 = 2.69$$

$$d'_{2,3} = 0.99 + (0.10 * 1.00) + (0.50 + 0.20) * 1.00 = 1.79$$

$$d'_{2,4} = 1.00 + (0.90 * 1.00) + (0.50 + 0.80) * 1.00 = 3.20$$

$$d'_{3,4} = 0.98 + (0.30 * 1.00) + (0.20 + 0.80) * 1.00 = 2.28$$

Los valores resultantes en dichas operaciones permitieron obtener dos matrices de distancia modificadas. Sin embargo, para poder generar el grafo respectivo de este individuo se tenía que descartar una de las dos matrices obtenidas, debido a que la codificación LNB utiliza para la construcción de sus árboles al algoritmo de Prim, el cual para la unión de los elementos de cada individuo necesita sólo de una sola matriz y al tener dos se generarían dos grafos, lo cual sería algo erróneo ya que estos datos pertenecían sólo a un individuo.

		EUCLIDIANA:						PEARSON:			
		1	2	3	4			1	2	3	4
d'ij	1		44,96	39,92	57,16		1		2,29	2,50	2,69
	2	44,96		38,45	53,64		2	2,29		1,79	3,20
	3	39,92	38,45		52,95		3	2,50	1,79		2,28
	4	57,16	53,64	52,95			4	2,69	3,20	2,28	

**Figura 3.31.** Matrices de distancia modificadas del Individuo 1 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).

Por lo tanto, para facilitar la construcción de los grafos, se elaboró una tercera matriz a partir de las ya existentes, la cual serviría como punto de partida para la unión de los nodos utilizando el algoritmo de Prim. En dicha matriz se compararon los valores de distancia entre los nodos de las matrices de la figura 3.31, escogiendo solamente al menor en cada comparación:

	1	2	3	4
1		2,29	2,50	2,69
2	2,29		1,79	3,20
3	2,50	1,79		2,28
4	2,69	3,20	2,28	

**Figura 3.32.** Matriz de distancia modificada del Individuo 1 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).

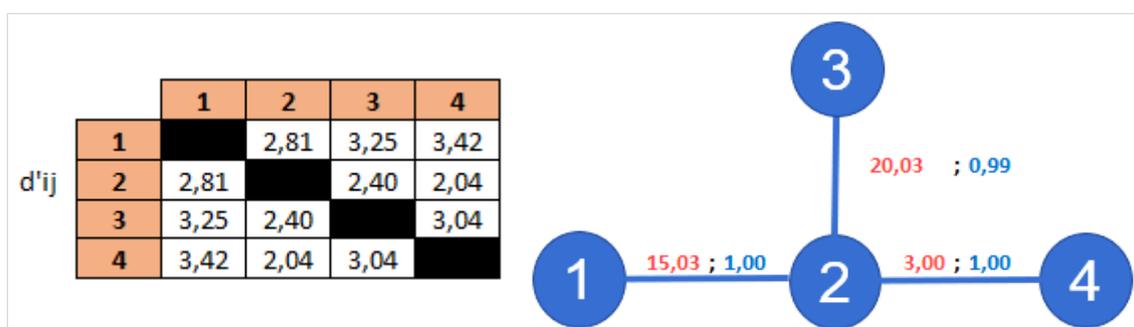
En la matriz resultante (Figura 3.32) cada fila y columna representaba un nodo, y para la construcción del árbol, se realizaba la unión entre un número de fila escogido al azar y un número de columna con el menor valor en esa fila. Por ejemplo, en la primera fila que representa al nodo 1, el valor menor fue de 2.29 y ese está en la columna 2; efectuando la primera unión de nodos entre el número 1 y el 2. En las siguientes uniones ya no se escogía un número (nodo) al azar, sino que se verificaba en las filas de los nodos que ya estaban añadidos, seleccionando al que contenía menor valor en sus columnas. Así sucesivamente hasta completar  $n-1$  uniones. El grafo resultante de este proceso fue el siguiente:



**Figura 3.33.** Individuo 1 (ejemplo con representación LNB).

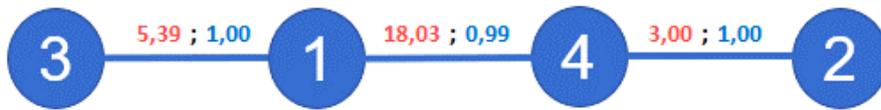
**Fuente:** (Elaboración propia, 2019).

A partir de aquí se muestran los MST-MO de esta representación desde el individuo 2 al individuo 6 (Figuras 3.34 a la 3.38), los cuales se obtuvieron realizando el mismo proceso que se hizo con el individuo 1.



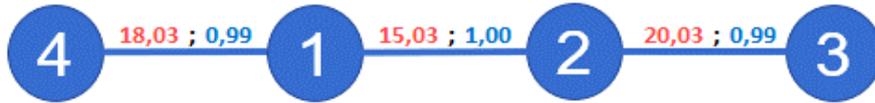
**Figura 3.34.** Individuo 2 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).



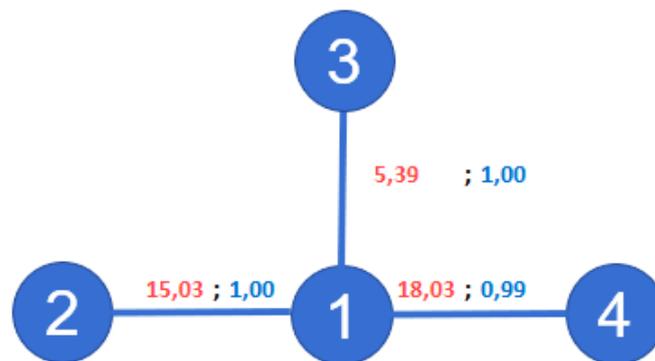
**Figura 3.35.** Individuo 3 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).



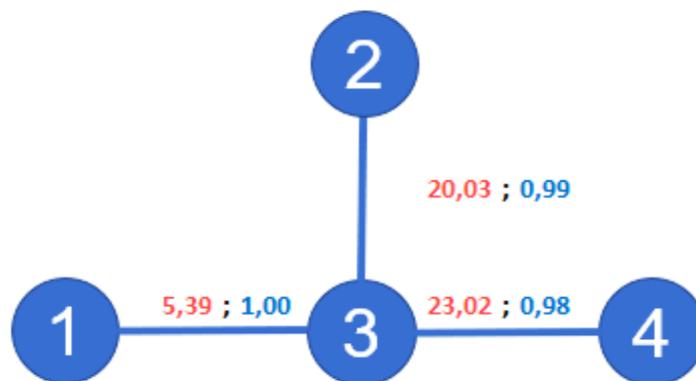
**Figura 3.36.** Individuo 4 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).



**Figura 3.37.** Individuo 5 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).



**Figura 3.38.** Individuo 6 (ejemplo con representación LNB).

**Fuente:** (Elaboración propia, 2019).

### 3.2.2.2. OPERADORES DE CRUZAMIENTO Y MUTACIÓN

La selección de individuos para la segunda generación en esta representación se realiza tal y como se hace en la representación Prüfer number, es decir de acuerdo a su ranking Pareto. Sin embargo, el operador de cruzamiento que se aplica en cada pareja de individuos luego de ser seleccionados se calcula siguiendo las ecuaciones de crossover aritmético presentadas en el cuadro 3.1, las cuales son precisamente indicadas para trabajar con genotipos que están compuestos por valores reales, tal y como es el caso de la representación LNB.

En estas ecuaciones el factor de peso constante  $w$  que permite calcular nuevos valores se utiliza para medir el porcentaje de los padres que va a tener cada hijo. Por ejemplo, al asumir un cruzamiento entre los valores de la matriz resultante del individuo 1 (Figura 3.32) con los valores de la matriz resultante del individuo 2 (Figura 3.34) y estableciendo un valor  $w = 70\%$  para esta pareja, las ecuaciones quedarían de la siguiente manera y los nuevos valores generados corresponderían a matrices resultantes para la unión de nodos en los hijos:

$$\begin{aligned}
 \text{Offspring1} &= \mathbf{0.70} * (2.29, 2.5, 2.69, 1.79, 3.2, 2.28) + \\
 &\quad (1 - \mathbf{0.70}) * (2.81, 3.25, 3.42, 2.4, 2.04, 3.04) \\
 &= (1.60, 1.75, 1.88, 1.25, 2.24, 1.6) \\
 &\quad + (0.84, 0.98, 1.03, 0.72, 0.61, 0.91) \\
 &= \mathbf{2.44, 2.63, 2.91, 1.97, 2.85, 2.51}
 \end{aligned}$$

$$\begin{aligned}
 \text{Offspring2} &= (1 - \mathbf{0.70}) * (2.29, 2.5, 2.69, 1.79, 3.2, 2.28) + \\
 &\quad \mathbf{0.70} * (2.81, 3.25, 3.42, 2.4, 2.04, 3.04) \\
 &= (0.69, 0.75, 0.81, 0.54, 0.96, 0.68) \\
 &\quad + (1.97, 2.28, 2.39, 1.68, 1.43, 2.13) \\
 &= \mathbf{2.66, 3.03, 3.2, 2.22, 2.39, 2.81}
 \end{aligned}$$

El tipo de mutación que se utilizó aquí es la mutación uniforme, en la cual un mismo individuo puede sufrir cambio en más de una posición de sus valores a alterar (en su matriz resultante en este caso), y no en una sola posición como en Prüfer number. Además, que los valores a reemplazar en dichas posiciones seleccionadas al azar, son generados aleatoriamente estableciendo un límite máximo y mínimo usando distribución uniforme, dichos límites se establecieron con valores entre cero y uno en esta representación.

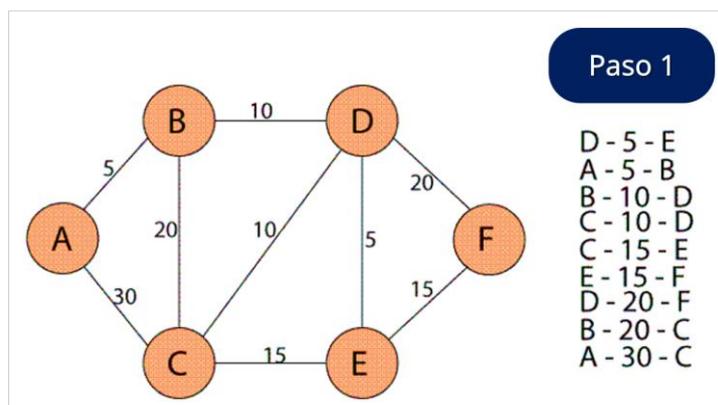
### 3.2.3. EJEMPLO CON EDGE-SET ENCODING

Los parámetros propios del NSGA-II para este ejemplo fueron los mismos establecidos en los ejemplos de las dos representaciones anteriores. Por ello, se explica de manera más directa el funcionamiento de los elementos que se necesitan en esta representación:

A diferencia de las representaciones anteriores para generar los individuos iniciales de la población  $P_t$ , edge-set encoding utiliza una variación de uno de los algoritmos de enfoque mono-objetivo más utilizados (algoritmo de Kruskal) este algoritmo funciona de la siguiente manera:

1. Primero se generan individuos correspondientes a grafos conectados completos y se ordenan las aristas del grafo por su peso de menor a mayor. Cabe recalcar que los datos trabajados en estos ejemplos fueron generados aleatoriamente o calculados a partir de los datos del cuadro 3.2; sin embargo, para las pruebas realizadas con el software RStudio se utilizaron datos reales de expresión génica disponibles en la web.
2. Se comienza con la arista de menor peso y si los vértices que contienen dicha arista no están en el mismo componente conexo, entonces se unen para formar un solo componente conexo, es decir Kruskal intenta unir cada arista hasta  $n-1$  veces siempre y cuando no se forme un ciclo.

Los pasos gráficos para el funcionamiento de este algoritmo se muestran en las siguientes figuras:



**Figura 3.39.** Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 1.

**Fuente:** (Arias, 2012).

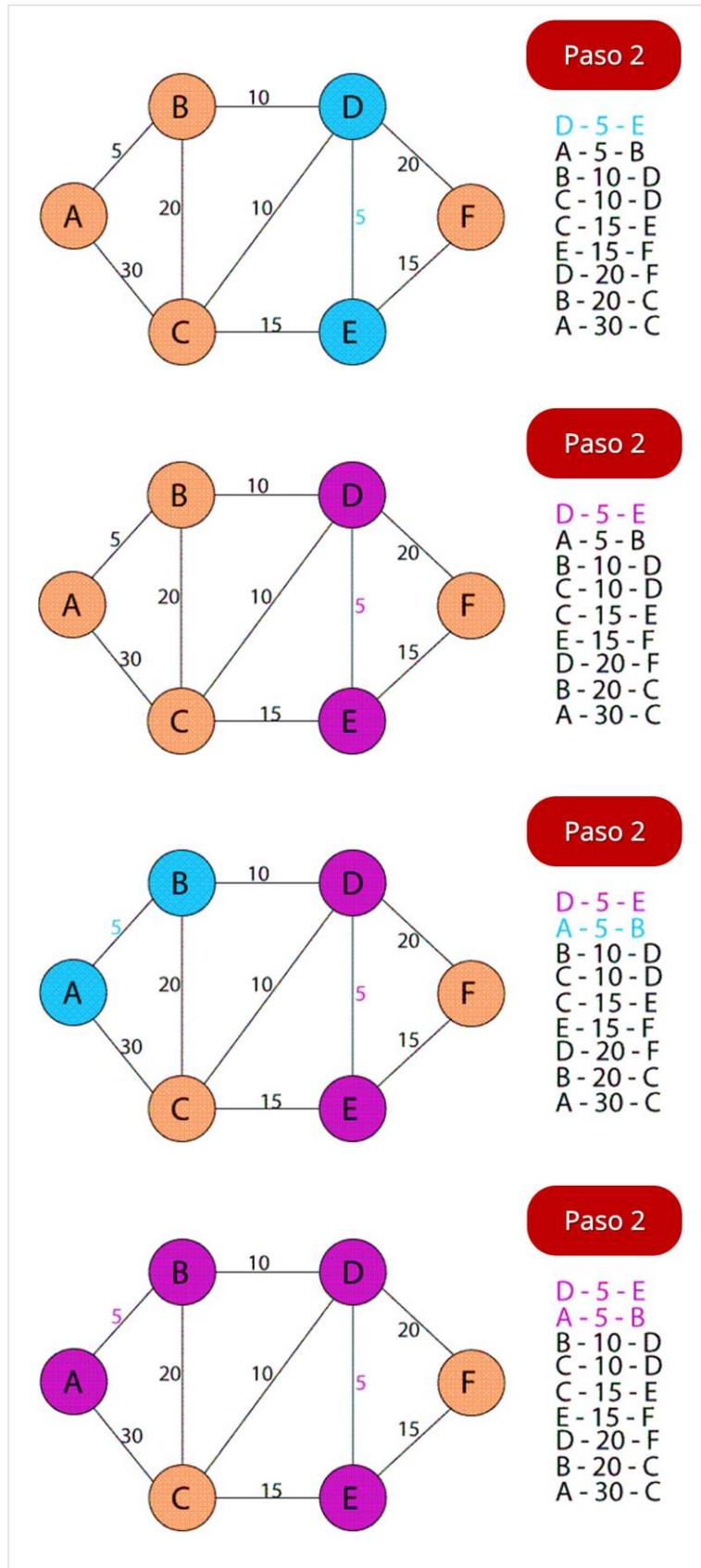


Figura 3.40. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.

Fuente: (Arias, 2012).

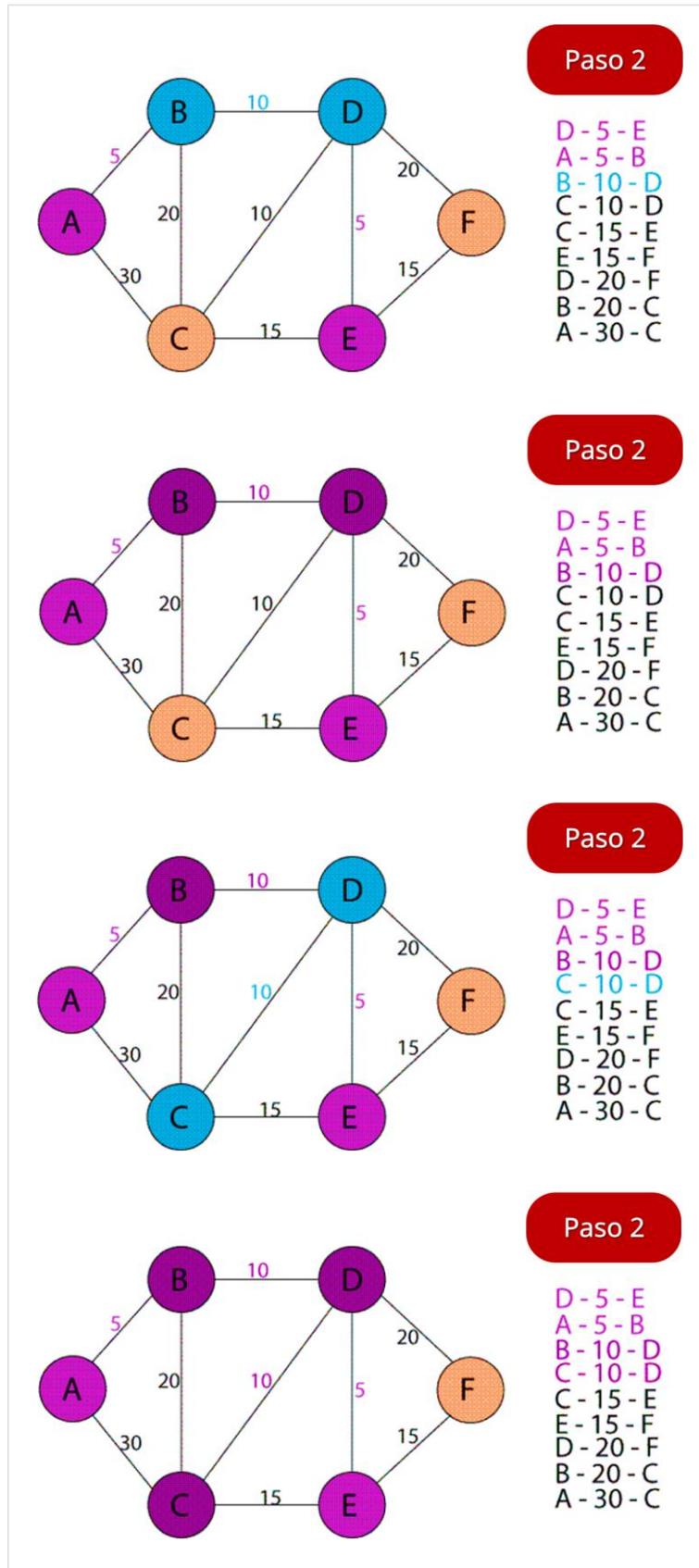


Figura 3.41. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.

Fuente: (Arias, 2012).

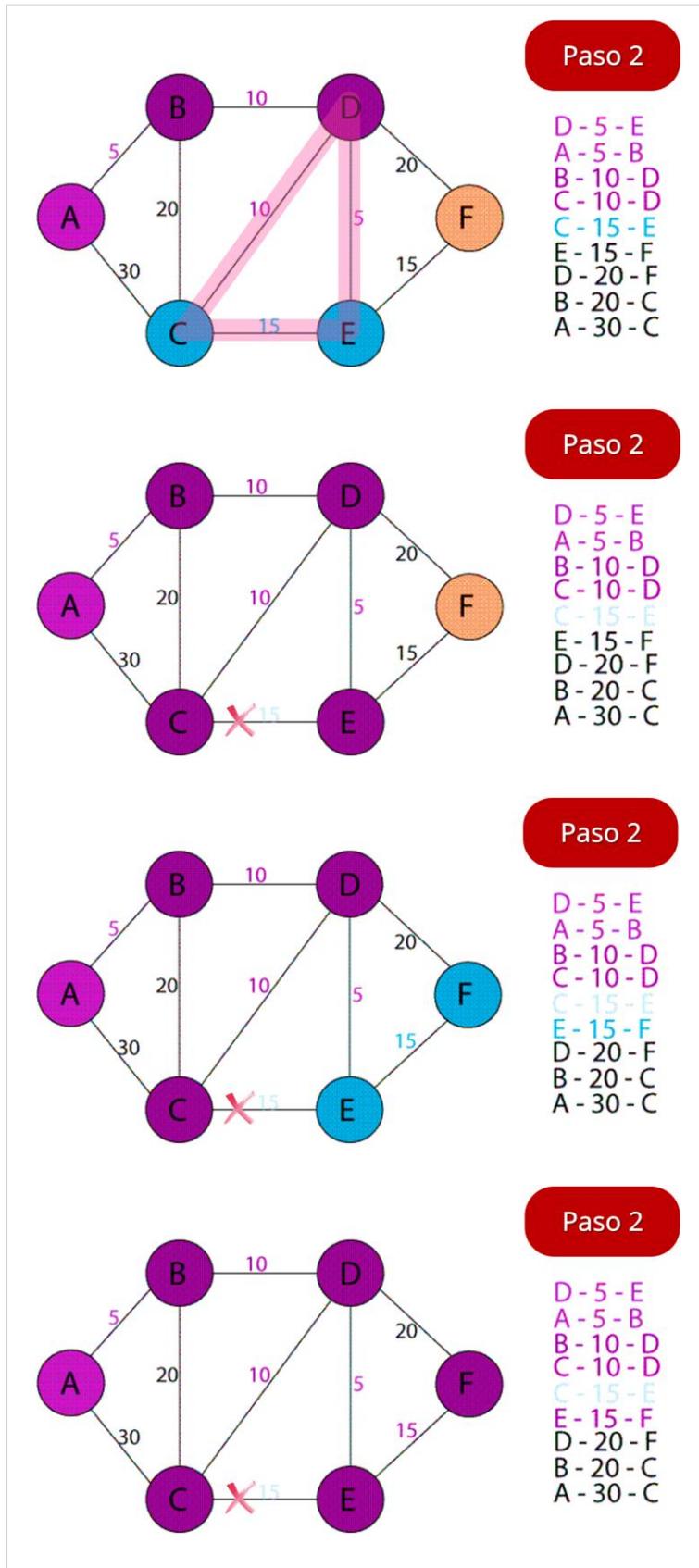
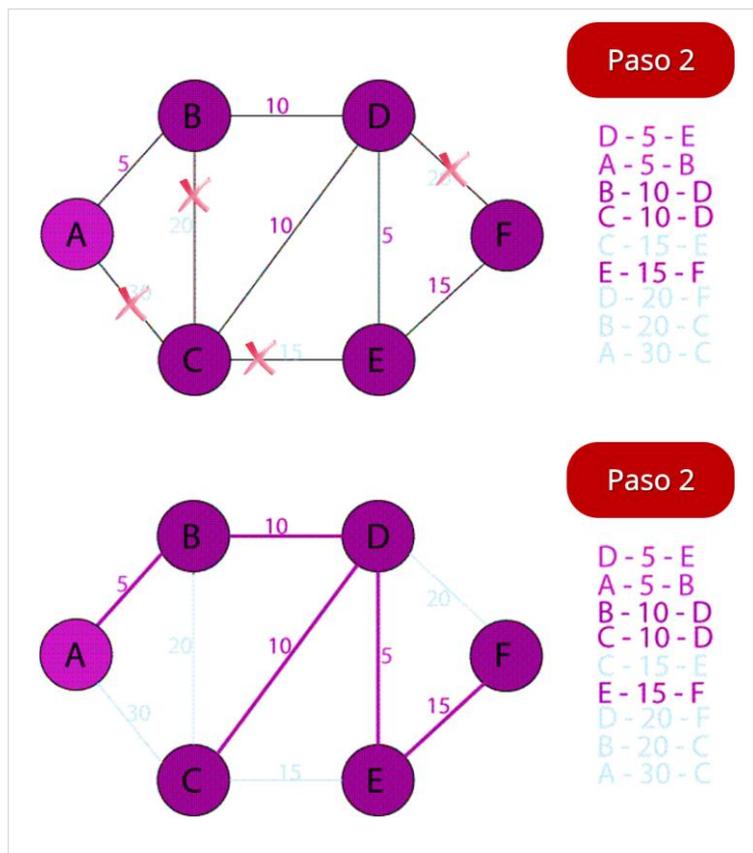


Figura 3.42. Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.

Fuente: (Arias, 2012).



**Figura 3.43.** Proceso de elaboración de MST a partir de un grafo conectado completo. Paso 2.  
Fuente: (Arias, 2012).

Este último grafo generado representa al MST que se obtuvo de un grafo conectado completo, y el proceso descrito corresponde al algoritmo de Kruskal. Sin embargo, el algoritmo utilizado aquí en esta representación no fue el de Kruskal original sino una variación de este; el de KruskalRST. La diferencia con el original es que éste último no ordenaba los bordes de acuerdo a sus pesos en las aristas como el de Kruskal original, sino que los elige al azar. Este mismo algoritmo (KruskalRST) se utilizó para el cruzamiento en los individuos. Y en la mutación, se eliminaba el primer borde en la lista que se realizaba y luego se añadía el último encontrado.

En la siguiente figura se muestra la vista de un script correspondiente a la implementación de una de las representaciones aplicadas al NSGA-II en RStudio, utilizando bibliotecas de R versión 3.4.0:

The screenshot displays the RStudio interface with a script editor on the left and the Environment pane on the right. The script in the editor is a main.R file that defines several functions and a main execution block. The functions include:

- `funcion_grafo_completo`: Generates a complete graph instance.
- `funcion_pesos_edge`: Generates edge weights.
- `funcion_objetivos_edge`: Calculates edge objectives.
- `funcion_ranking_crowding`: Calculates non-dominated crowding.
- `funcion_crossover_edge`: Generates crossover parameters.
- `funcion_mutacion_edge`: Generates mutation parameters.
- `funcion_busqueda_local`: Performs local search.

The main execution block (lines 14-51) creates a dataset, sets up parallel processing with `makeCluster` and `registerDoParallel`, and runs a `foreach` loop to generate solutions iteratively. It includes progress reporting and file saving.

The Environment pane on the right shows the Global Environment with various objects:

- `c1`: List of 4
- `elementos`: num [1, 1+]: 310 228 1 Inf
- `individuo_mst`: 1 obs. of 1004 variables
- `instancia`: 125250 obs. of 4 variables
- `it_completa`: Large matrix (375754 elements, 2.9 Mb)
- `iteraciones_pa...`: List of 0
- `poblacion_fina...`: Large list (2 elements, 303.9 Mb)
- `range_a`: Large matrix (125250 elements, 978.7...)
- `range_b`: Large matrix (125250 elements, 978.7...)
- `soluciones_ite...`: Large list (2 elements, 361.2 Mb)
- `soluciones_par...`: Large list (2 elements, 22.9 Mb)
- `soluciones_par...`: List of 2

The Values pane shows the following attributes:

- `crossRate`: 0.8
- `directorio`: "Resultados\_base\_Edge"
- `g1`: int [1:500]: 1 2 3 4 5 6 7 8 9 10 ...
- `g2`: int [1:500]: 125251 125252 125253 125254...
- `g3`: int [1:4]: 375751 375752 375753 375754
- `it`: 14L
- `itera`: 2
- `longitud_pesos`: 125250
- `m`: 500
- `maxIteration`: 14
- `minIteration`: 13
- `mutaRate`: 0.05
- `n`: 501L
- `nucleos`: 4
- `num_col_ed`: 375750
- `numberGenerati...`: 100
- `objDim`: 2

Figura 3.44. Vista de uno de los scripts principales elaborados en RStudio  
Fuente: (Elaboración propia, 2019).

### 3.3. OBJETIVO 3: PRUEBAS

Para los experimentos computacionales se utilizaron las representaciones y operadores evolutivos respectivos presentados en el cuadro 3.1. junto con los parámetros establecidos en el cuadro 3.9 aplicados al algoritmo NSGA-II en el software para análisis estadísticos RStudio.

**Cuadro 3.9.** Parámetros del NSGA-II

Configuración	Magnitud
Número de objetivos:	2
Tamaño de la población:	50
Tamaño del torneo:	2
Probabilidad de cruce:	0,80
Probabilidad de la mutación:	0,05
Número de generaciones:	100
Número de iteraciones:	20

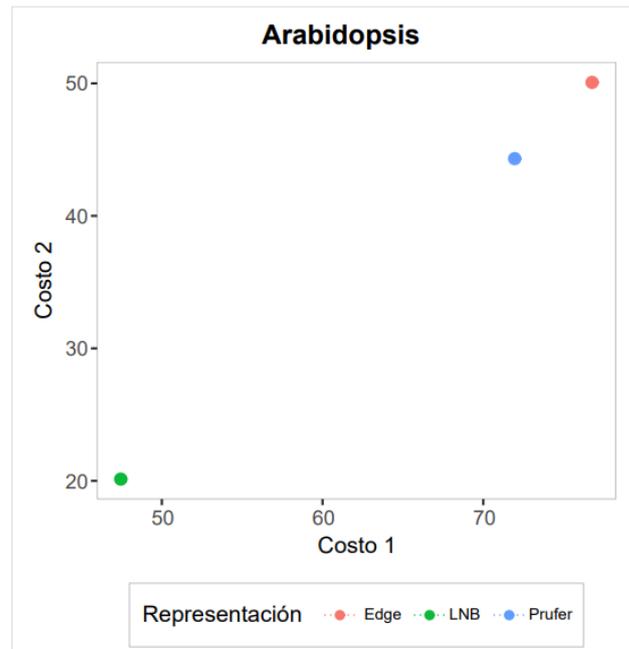
**Fuente:** (Elaboración propia, 2019).

Por cada representación implementada se llevaron a cabo cuatro conjuntos de ejecuciones correspondientes al número de bases de datos utilizadas (arabidopsis, cell cycle, sporulation y serum). Los valores de estas bases de datos corresponden a datos de expresión génica, los cuales se pueden representar como Gráficos Completos (CG) con 133, 384, 472 y 501 nodos, por lo que se eliminó de las matrices originales los elementos (genes) duplicados o con valores de expresión faltantes. Por lo tanto, cada CG tiene 8.778, 73.536, 111.156 y 125.250 bordes respectivamente, los valores de dichos bordes se utilizaron como costos de los objetivos  $c1_{i,j}$  y  $c2_{i,j}$ .

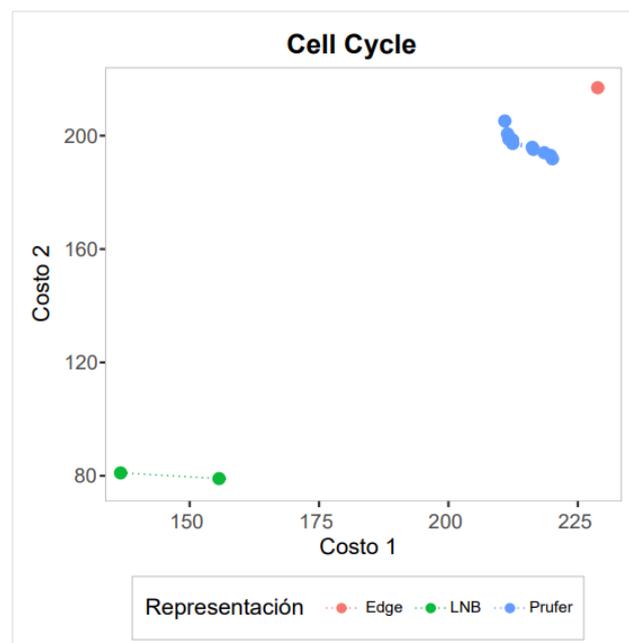
Se aplicó clustering a los datos de expresión génica mencionados, utilizando para ello la medida de distancia euclidiana y la correlación de Pearson (ecuación 3.1 y 3.2), estos datos posteriormente fueron normalizados y transformados para disminuir las variaciones que tendrían entre sí al trabajar con ellos y para hacer que los cálculos posteriores sean más sencillos. Todas estas configuraciones fueron establecidas de acuerdo a trabajos similares mencionados en las referencias bibliográficas de esta investigación, debido a los buenos resultados que se demostraron con ellas.

### 3.4. OBJETIVO 4: MEJOR ALTERNATIVA

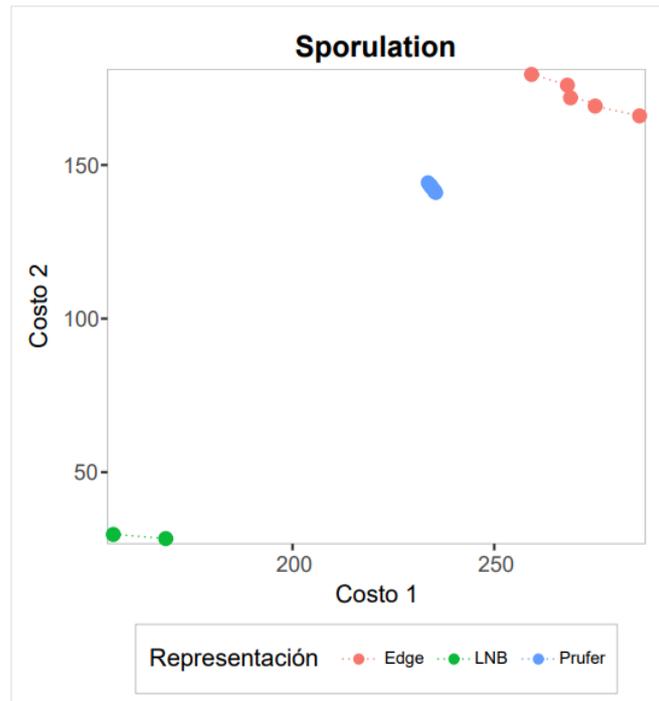
Las mejores fronteras de Pareto aproximadas de cada representación, obtenidas en cada base de datos para casos bi-objetivos según HV fueron las siguientes:



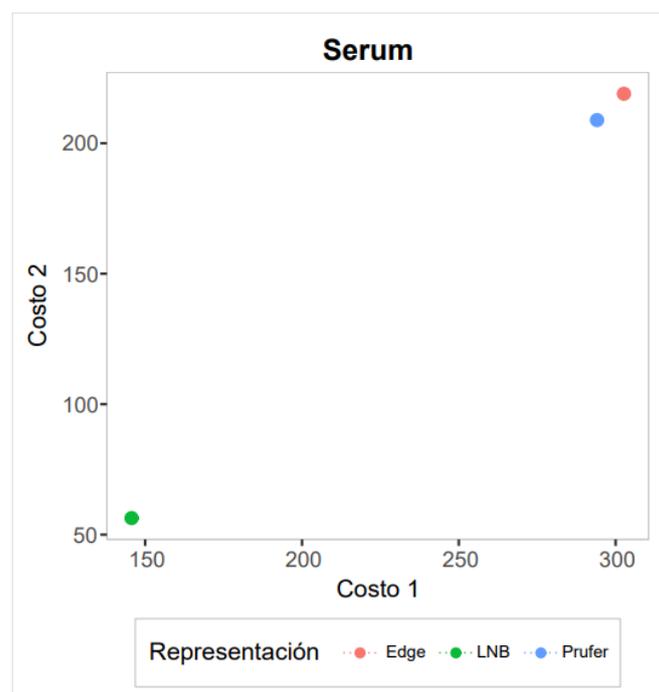
**Gráfico 3.5.** Mejores fronteras de Pareto aproximadas obtenidas en Arabidopsis Thaliana.  
**Fuente:** (Elaboración propia, 2019).



**Gráfico 3.6.** Mejores fronteras de Pareto aproximadas obtenidas en Cell Cycle.  
**Fuente:** (Elaboración propia, 2019).



**Gráfico 3.7.** Mejores fronteras de Pareto aproximadas obtenidas en Sporulation.  
**Fuente:** (Elaboración propia, 2019).



**Gráfico 3.8.** Mejores fronteras de Pareto aproximadas obtenidas en Serum.  
**Fuente:** (Elaboración propia, 2019).

### 3.4.1. CÁLCULOS DE MÉTRICAS (HIPERVOLUMEN)

Los mejores MST-MO generados de las representaciones en cada base de datos se presentan en el Anexo 1, para el efecto se utilizó el IDE RStudio.

En el cuadro 3.10 se muestran los valores medios de hipervolumen obtenidos de los cuadros presentados en el Anexo 2, en los cuales se realizaron 20 ejecuciones consecutivas de las distintas representaciones por cada base de datos. En el cálculo de dichos valores se utilizó (1, 1) como punto de referencia normalizado, es decir, todos los valores de las funciones objetivo estuvieron dentro del rango [0 – 1] en los bi-objetivos. Y aquellos que obtuvieron un mayor valor de hipervolumen significaron mejores soluciones desde un punto de vista multiobjetivo. De forma similar, en los gráficos 3.5 al 3.8 se muestran los mejores frentes de Pareto de cada representación de acuerdo con el hipervolumen en donde las mejores soluciones no dominadas fueron aquellas que estuvieron más cerca de los valores mínimos en cada objetivo de los gráficos.

**Cuadro 3.10.** Resumen promedio del cálculo de HV en las representaciones por cada base de datos.

BASES DE DATOS	REPRESENTACIONES		
	<i>Prüfer Number</i>	<i>LNB encoding</i>	<i>Edge-set encoding</i>
<i>Arabidopsis</i>	0,0545938086612448	0,956520665992264	0,0045575269805201
<i>Cell Cycle</i>	0,0493150479810103	0,937933768628762	0,0084032015966041
<i>Sporulation</i>	0,0788078099334588	0,957823492260757	0,0158584957274975
<i>Serum</i>	0,0075246661970557	0,984912302776144	0,0011056384537812

Fuente: (Elaboración propia, 2019).

### 3.4.2. ANÁLISIS ESTADÍSTICOS

Por medio del test de Friedman se realizó la comprobación-rechazo de las siguientes Hipótesis:

- H0: El rendimiento de las representaciones cromosómicas es similar.
- H1: El rendimiento de las representaciones cromosómicas es diferente.

Se determinó un nivel de confianza del 95%, el cual según Cardenas (2015) es el más utilizado es este tipo de análisis. De acuerdo con esto, cuando la significación sea menor a 0,05 es que las dos variables están relacionadas.

Los resultados obtenidos en las representaciones por cada base de datos para esta prueba estadística de diferencias significativas fueron los siguientes:

**Cuadro 3.11.** Estadística resumen de Arabidopsis

Representación	hv_media	hv_mediana	hv_sd	hv_min	hv_max
Edge	0.00455752698 052005	0.00267900666 007515	0.00505416830 181569	0.00078696112 9335193	0.01992372937 02616
LNB	0.95652066599 2264	0.96069738088 0718	0.02316172419 55204	0.89900945363 2752	1
Prüfer	0.05459380866 12447	0.05408615256 32346	0.01447661929 46859	0.02971249456 47199	0.07824733860 7772

Fuente: (Elaboración propia, 2019).

**Cuadro 3.12.** Estadística resumen de Cell cycle

Representación	hv_media	hv_mediana	hv_sd	hv_min	hv_max
Edge	0.00840320159 660413	0.00787220036 51577	0.00225283820 524331	0.00411562277 252953	0.01256279175 68969
LNB	0.93793376862 8762	0.93682475226 5658	0.02302634292 71872	0.89584007197 5036	0.98504838478 5442
Prüfer	0.04931504798 10103	0.04717008293 25684	0.01296165363 29977	0.02636690125 03811	0.07239193077 42371

Fuente: (Elaboración propia, 2019).

**Cuadro 3.13.** Estadística resumen de Sporulation

Representación	hv_media	hv_mediana	hv_sd	hv_min	hv_max
Edge	0.01585849572 74975	0.01576476226 22192	0.00386064901 891062	0.00973243946 991808	0.02182866617 08118
LNB	0.95782349226 0757	0.95943840112 6032	0.02016471840 33212	0.92443733467 545	0.99463706801 253
Prüfer	0.07880780993 34588	0.07812383335 02551	0.02211930428 45166	0.03491033524 64848	0.11631421687 2299

Fuente: (Elaboración propia, 2019).

**Cuadro 3.14.** Estadística resumen de Serum

Representación	hv_media	hv_mediana	hv_sd	hv_min	hv_max
Edge	0.01585849572 74975	0.01576476226 22192	0.00386064901 891062	0.00973243946 991808	0.02182866617 08118
LNB	0.95782349226 0757	0.95943840112 6032	0.02016471840 33212	0.92443733467 545	0.99463706801 253
Prüfer	0.07880780993 34588	0.07812383335 02551	0.02211930428 45166	0.03491033524 64848	0.11631421687 2299

Fuente: (Elaboración propia, 2019).

En donde, a partir de los resultados obtenidos en el hipervolumen se calculó: media, mediana, desviación estándar, mínimo y máximo de dichos datos.

Como el valor de  $p$  fue de  $0.01832 < 0.05$  en dichos resultados, se rechaza  $H_0$ , es decir que los datos presentan suficiente evidencia para indicar que existe diferencia significativa en el rendimiento HV de las representaciones cromosómicas. Por lo tanto, para determinar en cuál par de representación cromosómica existía diferencia significativa se aplicó el test de Nemenyi.

Por medio del test de Nemenyi se comprobó la diferencia significativa entre las representaciones LNB y edge, para lo cual en el experimento se incluyó a los tres pares de comparaciones mostradas en el cuadro siguiente:

**Cuadro 3.15.** Resultados en las diferencias significativas según post hoc test (Nemenyi)

Pares de Representaciones	Resultado		¿Existe diferencia significativa?
Prüfer – LNB	0.333	→	NO
Prüfer – Edge	0.333	→	NO
LNB – Edge	0.013	→	SI

**Fuente:** (Elaboración propia, 2019).

## CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

### 4.1. CONCLUSIONES

- En este trabajo de titulación se identificó a LNB encoding como la mejor alternativa de representación cromosómica que mejorara el rendimiento del algoritmo NSGA-II en términos multiobjetivo para el problema del MST-MO.
- Por medio del método bibliográfico se permitió recabar la información necesaria sobre la problemática expuesta y las alternativas que se recomiendan utilizar en investigaciones previas para su solución.
- La elaboración de informes acerca de la comprensión de cada una de las temáticas involucradas en este trabajo de investigación, facilitó la posterior codificación de las representaciones (Prüfer number, link and node biased encoding y edge-set encoding) en el software RStudio. Con esto también, se permitió contribuir a los futuros investigadores que deseen continuar con la exploración y mejora del algoritmo NSGA-II, debido a que no en todas las representaciones existen estudios actuales suficientes como fue el caso de la representación link and node biased encoding (LNB).
- Con el fin de trabajar con datos reales en los experimentos computacionales, se utilizaron cuatro bases de datos distintas (arabidopsis, cell cycle, sporulation y serum), mismas que se encuentran disponibles en la web y que corresponden a datos de expresión génica. Los parámetros o instancias de prueba fueron implementados de acuerdo con configuraciones similares utilizadas en algunas de las fuentes bibliográficas previamente consultadas en este trabajo de titulación. Empleando para el efecto, librerías del lenguaje de programación R versión 3.4.0 a través del IDE RStudio versiones 1.1.463 y 1.2.1335, por motivo de que las ejecuciones fueron realizadas en diferentes máquinas con distintas versiones del software.
- Para determinar cuál de las representaciones sería la alternativa que tuviese mejor rendimiento en términos multiobjetivo, se utilizó el indicador de hipervolumen que es una métrica de desempeño multiobjetivo muy utilizada en este tipo de evaluaciones. A partir del cálculo de esta métrica se pudo determinar que la representación LNB fue la mejor alternativa, debido a que

en esta representación se obtuvieron los valores más altos desde el punto de referencia establecido  $(1, 1)$ ; los cuales se pueden visualizar en el cuadro 3.10. Además, de que gráficamente también se demuestran las mejores fronteras de Pareto de las representaciones por cada base de datos en los gráficos 3.5 al 3.8; donde claramente se observa que LNB posee las fronteras de Pareto con las soluciones más cercanas al punto  $(0, 0)$  que es considerado como el punto óptimo para problemas de minimización. Del mismo modo, los análisis estadísticos por medio de los test de Friedman y Nemenyi demostraron que la mejor representación cromosómica fue la de LNB debido a que tuvo diferencia significativa con la representación edge-set encoding en los valores promedios obtenidos del hipervolumen.

## 4.2. RECOMENDACIONES

- A los futuros investigadores que documenten sus investigaciones para una mejor comprensión de la información en trabajos similares.
- Se recomienda utilizar computadores de gama media o alta con un sistema operativo basado en Linux cuando se trabaje con bastantes datos en RStudio, ya que estos SO se distinguen de otros por su estabilidad al momento de ejecutar gran cantidad de procesos al mismo tiempo. También se sugiere que los ordenadores posean las siguientes características principales mínimas: procesador i5 7ma generación o superior, RAM 8GB o más.
- A futuros investigadores, cuando trabajen con grandes cantidades de datos se les recomienda hacer los experimentos computacionales respectivos conforme vayan realizando el código fuente y obteniendo las bases de datos con las que vayan a trabajar, es decir se recomienda ejecutar los códigos lo más pronto posible para así evitar posibles atrasos en los proyectos. Además, se recomienda normalizar las bases de datos para facilitar su posterior manipulación.
- En este trabajo de titulación sólo fue necesario utilizar una métrica de desempeño multiobjetivo debido a que no se presentaron cruces entre las fronteras de Pareto resultantes de cada representación, pero si éste hubiese

siendo el caso, se habría visto en la necesidad de utilizar otras métricas adicionales a la de Hipervolumen para evaluar las representaciones. Por lo tanto, si éste llega a ser el caso de los proyectos futuros, se recomienda utilizar las métricas de Cobertura y Distancia Generacional.

## BIBLIOGRAFÍA

- Amat, J. (2016). RPubS - Test de Friedman. Retrieved August 16, 2019, from [https://rpubs.com/Joaquin\\_AR/219732](https://rpubs.com/Joaquin_AR/219732)
- Arias, J. (2012). Árbol de Expansión Mínima: Algoritmo de Kruskal. Retrieved June 5, 2019, from <https://jariasf.wordpress.com/2012/04/19/arbore-expansion-minima-algoritmo-de-kruskal/>
- Campos, G., Dugardin, F., Yalaoui, F., & Kelly, R. (2016). A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*, 49(12), 1272–1277. <https://doi.org/10.1016/J.IFACOL.2016.07.690>
- Cardenas, J. (2015). Qué es ANOVA de un factor y cómo analizarla fácilmente. Blog de Sociología. Retrieved August 16, 2019, from <http://networkianos.com/anova-de-un-factor-que-es-como-analizar/>
- Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., ... Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1), 65–73. [https://doi.org/10.1016/S1097-2765\(00\)80114-8](https://doi.org/10.1016/S1097-2765(00)80114-8)
- Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., & Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, 282(5389), 699–705. <https://doi.org/10.1126/science.282.5389.699>
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí). (2018a). *Grupo de Investigación Sistemas Computacionales (SISCOM)* (pp. 1–2). pp. 1–2. Calceta-Manabí, EC.
- ESPAM MFL (Escuela Superior Politécnica Agropecuaria de Manabí). (2018b). *Reglamento para la conformación de Grupos de Investigación* (pp. 2–3). pp. 2–3. Calceta-Manabí, EC.
- Fang, X., Wang, W., He, L., Huang, Z., Liu, Y., & Zhang, L. (2018). Research on Improved NSGA-II Algorithm and Its Application in Emergency Management. *Mathematical Problems in Engineering*, 2018, 1–13. <https://doi.org/10.1155/2018/1306341>
- Fernández, M. (2019). *Optimización de rutas de vehículos con entrega y recolección con múltiples objetivos* (Universidad Nacional Autónoma de México). Retrieved from <http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/16621>
- Frost, J. (2019). Using Post Hoc Tests with ANOVA - Statistics By Jim. Retrieved August 15, 2019, from <https://statisticsbyjim.com/anova/post-hoc-tests-anova/#comments>

- Furqan, M., Hartono, Ongko, E., & Ikhsan, M. (2017). Performance of Arithmetic Crossover and Heuristic Crossover in Genetic Algorithm Based on Alpha Parameter. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(5), 31–36. <https://doi.org/10.9790/0661-1905013136>
- García, E. (2018). *Algoritmo evolutivo paralelo para transformaciones de espacio en problemas de optimización*. (Instituto Politécnico Nacional). Retrieved from <https://tesis.ipn.mx/bitstream/handle/123456789/24967/T1895.pdf?sequence=1&isAllowed=y>
- Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C. F., ... Brown, P. O. (1999). The transcriptional program in the response of human fibroblasts to serum. *Science*, 283(5398), 83–87. <https://doi.org/10.1126/science.283.5398.83>
- Jakob, R. (2019). How to correctly apply the Nemenyi post-hoc test after the Friedman test. Retrieved August 16, 2019, from <https://stats.stackexchange.com/questions/73376/how-to-correctly-apply-the-nemenyi-post-hoc-test-after-the-friedman-test>
- Jimenez, D., & Chivata, O. (2017). *Diseño de un Algoritmo Evolutivo Bioinspirado el cual permita sintonizar un controlador PID con el fin de controlar la variable de proceso "Flujo" en un sistema de control de proceso AMATROL T5552 ubicado en el Laboratorio de electrónica de la Facultad T* (Universidad Distrital Francisco José de Caldas). Retrieved from <http://repository.udistrital.edu.co/handle/11349/6003>
- Lim, S., Sultan, A., Sulaiman, N., Mustapha, A., & Leong, K. (2017). Crossover and Mutation Operators of Genetic Algorithms. *International Journal of Machine Learning and Computing*, 7(1), 12. <https://doi.org/10.18178/ijmlc.2017.7.1.611>
- Lima, R. (2017). *Operadores de recombinação baseados em permutação para representações de grafos* (Universidade Federal de Goiás). Retrieved from <https://repositorio.bc.ufg.br/tede/handle/tede/7770>
- López, J. (2015). Optimización multi-objetivo. In *Portal de Libros de la Universidad Nacional de La Plata*. Retrieved from <https://libros.unlp.edu.ar/index.php/unlp/catalog/book/152>
- Maulik, U., Mukhopadhyay, A., & Bandyopadhyay, S. (2008). Combining Pareto-Optimal Clusters using Supervised Learning for Identifying Co-expressed Genes. Retrieved February 12, 2019, from <http://anirbanmukhopadhyay.50webs.com/data.html>
- Maya, E. (2014). *Métodos y técnicas de investigación* (Universidad Nacional Autónoma de México., Ed.). Retrieved from [http://arquitectura.unam.mx/uploads/8/1/1/0/8110907/metodos\\_y\\_tecnicas.pdf](http://arquitectura.unam.mx/uploads/8/1/1/0/8110907/metodos_y_tecnicas.pdf)

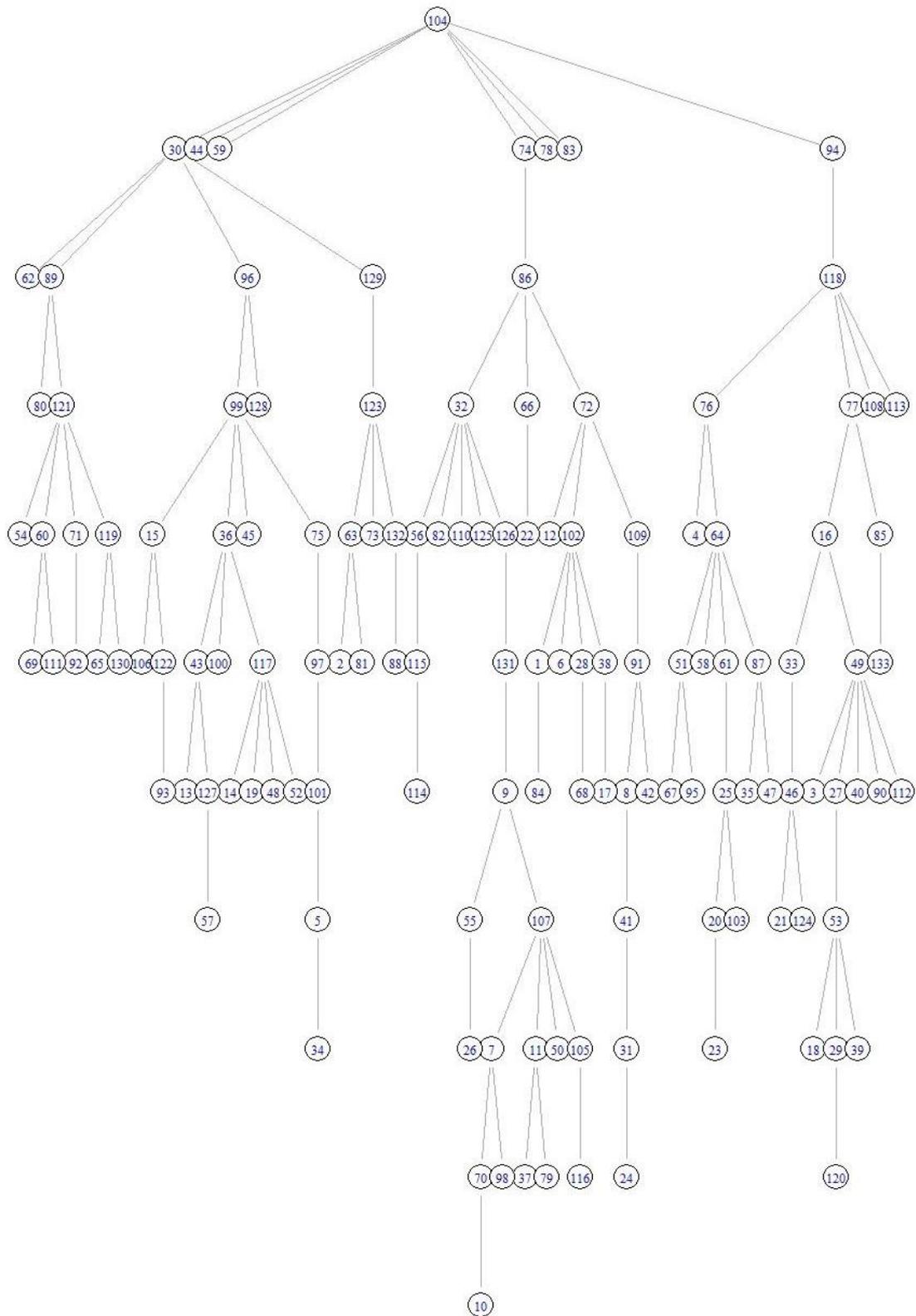
- Mestizo, S. (2015). *Árboles de decisión y redes bayesianas para el análisis de genes involucrados en la enfermedad de Alzheimer* (Universidad Veracruzana). Retrieved from <https://cdigital.uv.mx/bitstream/handle/123456789/42521/MestizoGutierrezSonia.pdf?sequence=1&isAllowed=y>
- Meza, J., Cueva, J., & Espitia, H. (2016). Revisión sobre algoritmos de optimización multi-objetivo genéticos y basados en enjambres de partículas. *Redes de Ingeniería*, 6(2), 54. <https://doi.org/10.14483/udistrital.jour.redes.2015.2.a06>
- Montoya, D., Ramirez, J., & Zuluaga, J. (2014). Multi-objective optimization for reconfiguration and capacitor allocation in distribution systems. *2014 North American Power Symposium (NAPS)*, 1–6. <https://doi.org/10.1109/NAPS.2014.6965364>
- Olaya-Abril, A., & Cejas-Molina, M. (2018). Bioinformática como recurso educativo: Proyecto de ingeniería genética. *EDMETIC, Revista de Educación Mediática y TIC*, 7(71), 2254–2259. <https://doi.org/10.21071/edmetic.v7i1.10027>
- Pandolfi, D., Villagra, A., & Molina, D. (2017). *Inteligencia computacional aplicada a la optimización multiojetivo de problemas de scheduling con restricciones*. 91–94. Retrieved from <http://sedici.unlp.edu.ar/handle/10915/61503>
- Párraga-Álava, J., Dorn, M., & Inostroza-Ponta, M. (2017). Using local search strategies to improve the performance of NSGA-II for the Multi-Criteria Minimum Spanning Tree problem. *2017 IEEE Congress on Evolutionary Computation (CEC)*, 1119–1126. <https://doi.org/10.1109/CEC.2017.7969432>
- Párraga Álava, J. A. (2017). *Clustering difuso multi-objetivo de genes basado en información biológica externa y datos de expresión génica*. Chile/Universidad de Santiago de Chile/2017.
- Pérez, C. (2015). *R. Lenguaje de Programación y análisis estadístico de datos* (GARCETA GRUPO EDITORIAL, Ed.). Retrieved from <https://www.casadellibro.com/libro-r-lenguaje-de-programacion-y-analisis-estadistico-de-datos/9788416228126/2632614>
- Pontes, B., Rodríguez, D., Díaz, N., & Giráldez, R. (2006). *Análisis de datos de expresión genética*. Retrieved from [http://www.lsi.us.es/~bepontes/papers/JA06\\_ExpGen.pdf](http://www.lsi.us.es/~bepontes/papers/JA06_ExpGen.pdf)
- Raidl, G. R., & Julstrom, B. A. (2003). Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, 7(3), 225–239. <https://doi.org/10.1109/TEVC.2002.807275>
- Reymond, P., Weber, H., Damond, M., & Farmer, E. E. (2000). Differential gene expression in response to mechanical wounding and insect feeding in Arabidopsis. *Plant Cell*, 12(5), 707–719. <https://doi.org/10.1105/tpc.12.5.707>

- Rodríguez, A. (2014). El método bibliográfico. Retrieved July 18, 2018, from <http://roa.ult.edu.cu/handle/123456789/2352>
- Rojas, L., López, J., & Muñoz, N. (2016). Asignación Óptima de Presupuesto para Mejoramiento de la Calidad del Servicio en Sistemas de Distribución usando NSGA-II y un Algoritmo Memético. *Información Tecnológica*, 27(1), 115–126. <https://doi.org/10.4067/S0718-07642016000100013>
- Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms* (2nd ed.). Retrieved from <http://carlosreynoso.com.ar/archivos/rothlauf.pdf>
- Saborido, R. (2014). *Optimización multi-objetivo basada en preferencias para la planificación de proyectos software* (Universidad de Málaga - Escuela Técnica Superior de Ingeniería Informática). Retrieved from <https://riuma.uma.es/xmlui/handle/10630/15247>
- SCI2S (Soft Computing and Intelligent Information Systems). (2014). BIOINFORMÁTICA. Retrieved January 29, 2019, from <https://sci2s.ugr.es/sites/default/files/files/Teaching/GraduatesCourses/Bioinformatica/Tema 12 - MOEAS.pdf>
- Simmons, O. (2015). Minimum Spanning Tree Problem. Retrieved January 28, 2019, from <https://slideplayer.com/slide/8807967/>
- Sosa, H., Villagra, S., & Villagra, N. (2014). Operadores de mutación en algoritmos genéticos celulares aplicados a problemas continuos. *Informe Científico Técnico UNPA, ISSN-e 1852-4516, Vol. 6, N.º. 2, 2014, Págs. 141-157, 6(2), 141–157.* Retrieved from <https://dialnet.unirioja.es/servlet/articulo?codigo=5123621>
- UNED (Universidad Nacional de Educación a Distancia). (2015). *Optimización Multiobjetivo.* Retrieved from [http://portal.uned.es/EadmonGuiasWeb/htdocs/abrir\\_fichero/abrir\\_fichero.jsp?idGuia=80603](http://portal.uned.es/EadmonGuiasWeb/htdocs/abrir_fichero/abrir_fichero.jsp?idGuia=80603)
- Wang, Y. (2014). *Fuzzy Clustering Models for Gene Expression Data Analysis* (University of the Northumbria). Retrieved from [http://nrl.northumbria.ac.uk/21438/1/wang.yu\\_phd.pdf](http://nrl.northumbria.ac.uk/21438/1/wang.yu_phd.pdf)
- Zhang, C., & Ma, X. (2015). NSGA-II Algorithm with a Local Search Strategy for Multiobjective Optimal Design of Dry-Type Air-Core Reactor. *Mathematical Problems in Engineering*, 2015, 1–9. <https://doi.org/10.1155/2015/839035>
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271. <https://doi.org/10.1109/4235.797969>

# **ANEXOS**

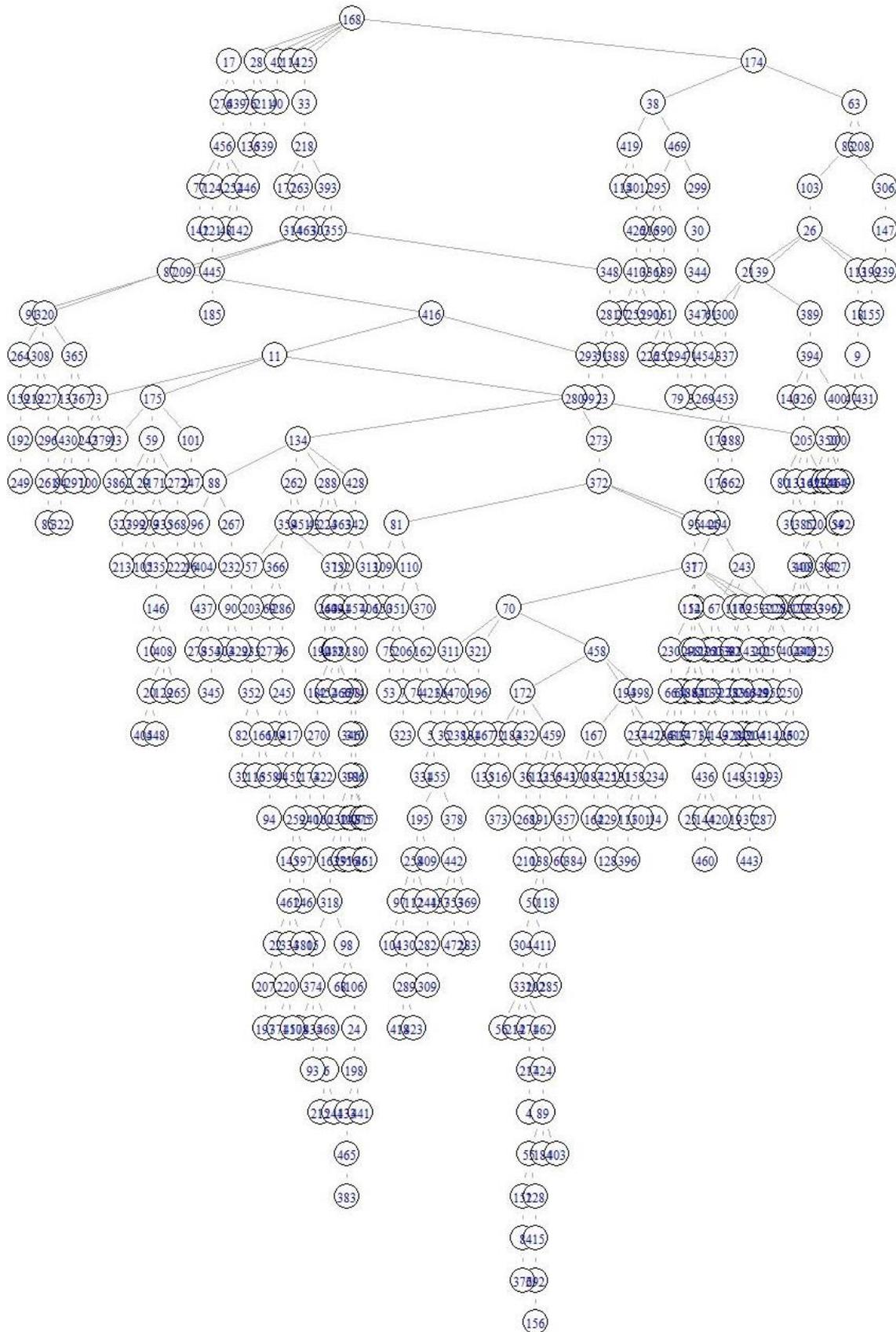
## ANEXO 1. MEJORES MST-MO DE CADA REPRESENTACIÓN

Representación: Edge – BD: Arabidopsis  
 Costo 1: 76.1 - Costo 2: 49.6

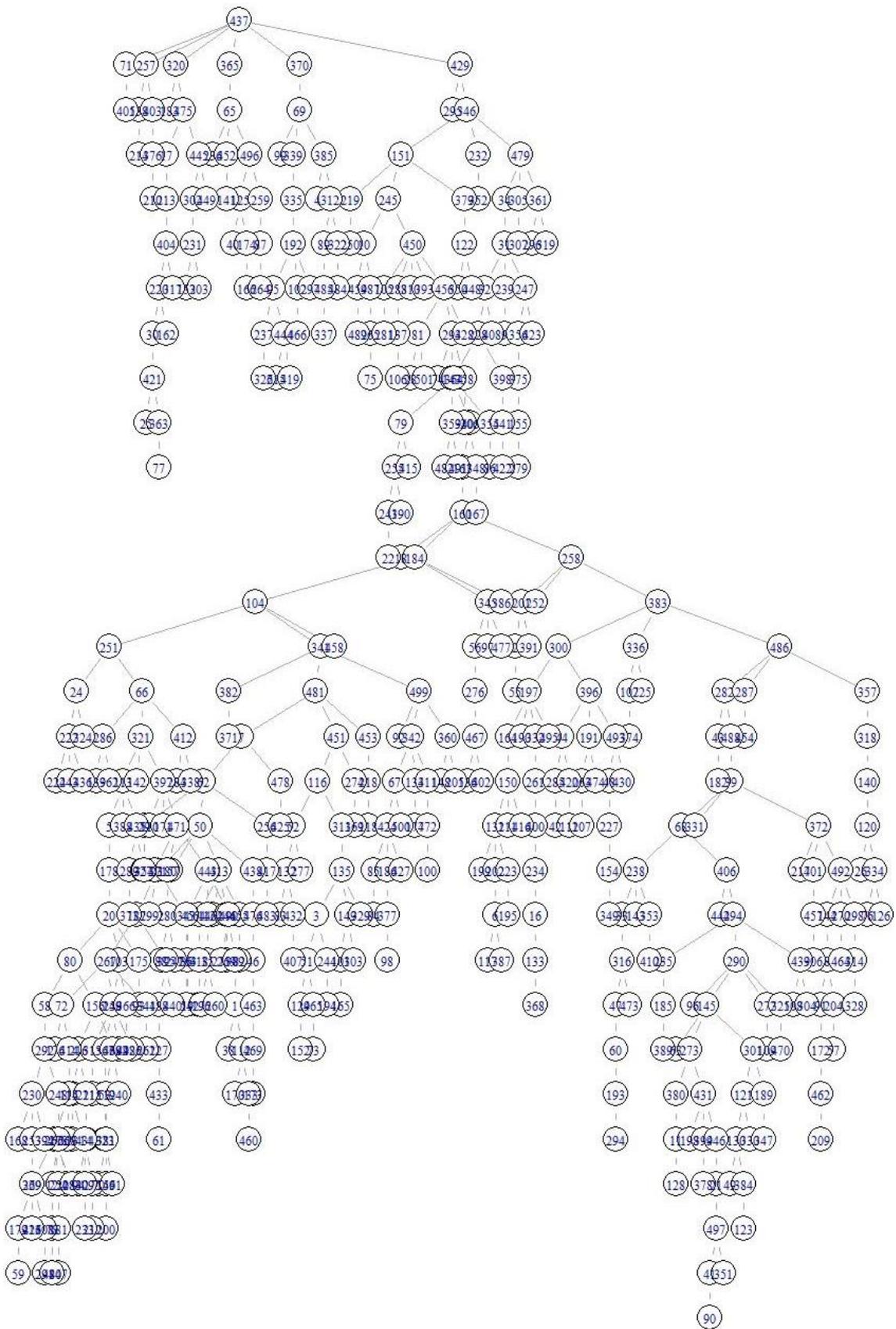




**Representación: Edge – BD: Sporulation**  
**Costo 1: 269.4 - Costo 2: 169.2**

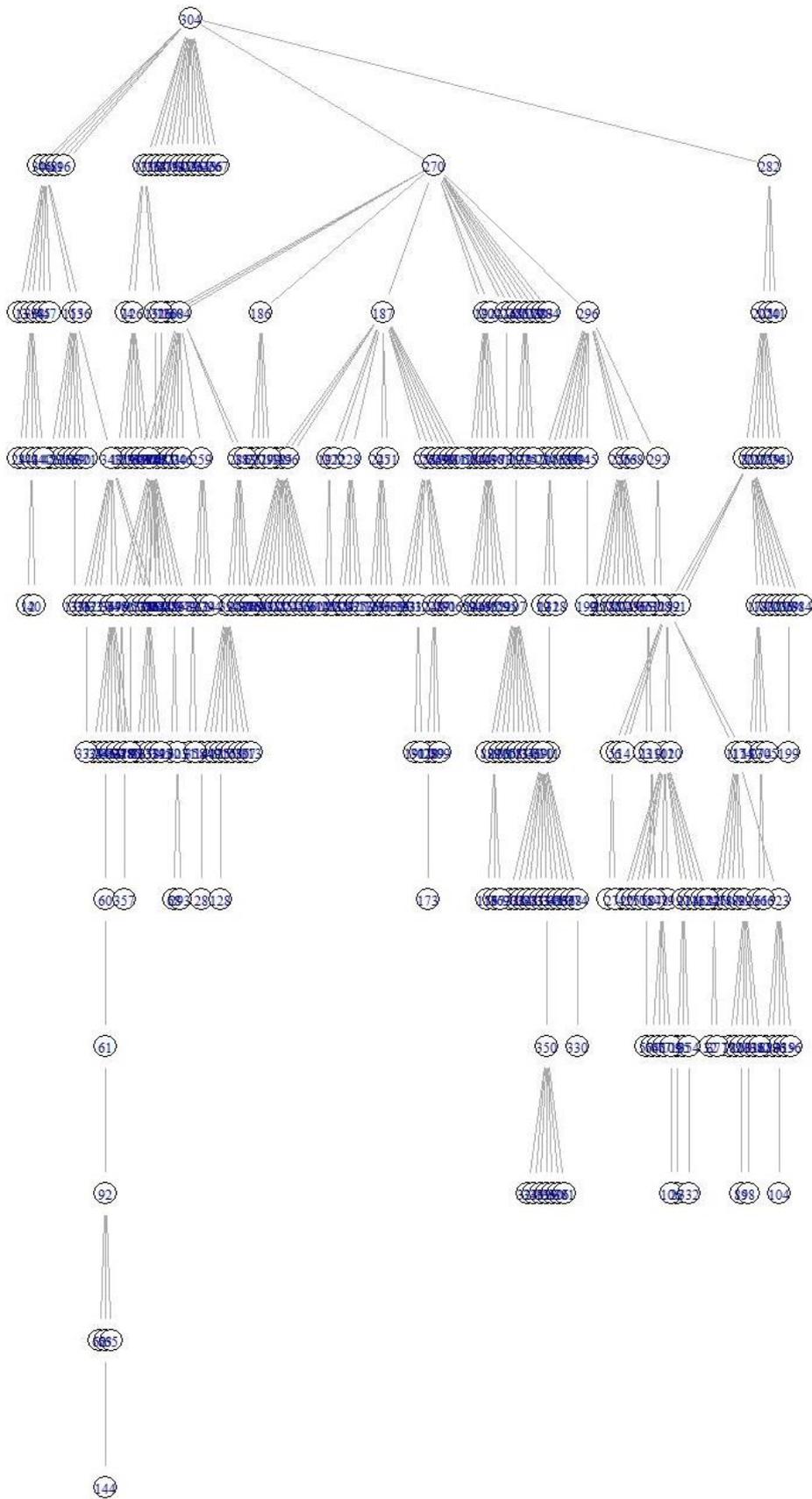


**Representación: Edge – BD: Serum**  
**Costo 1: 302.7 - Costo 2: 219**



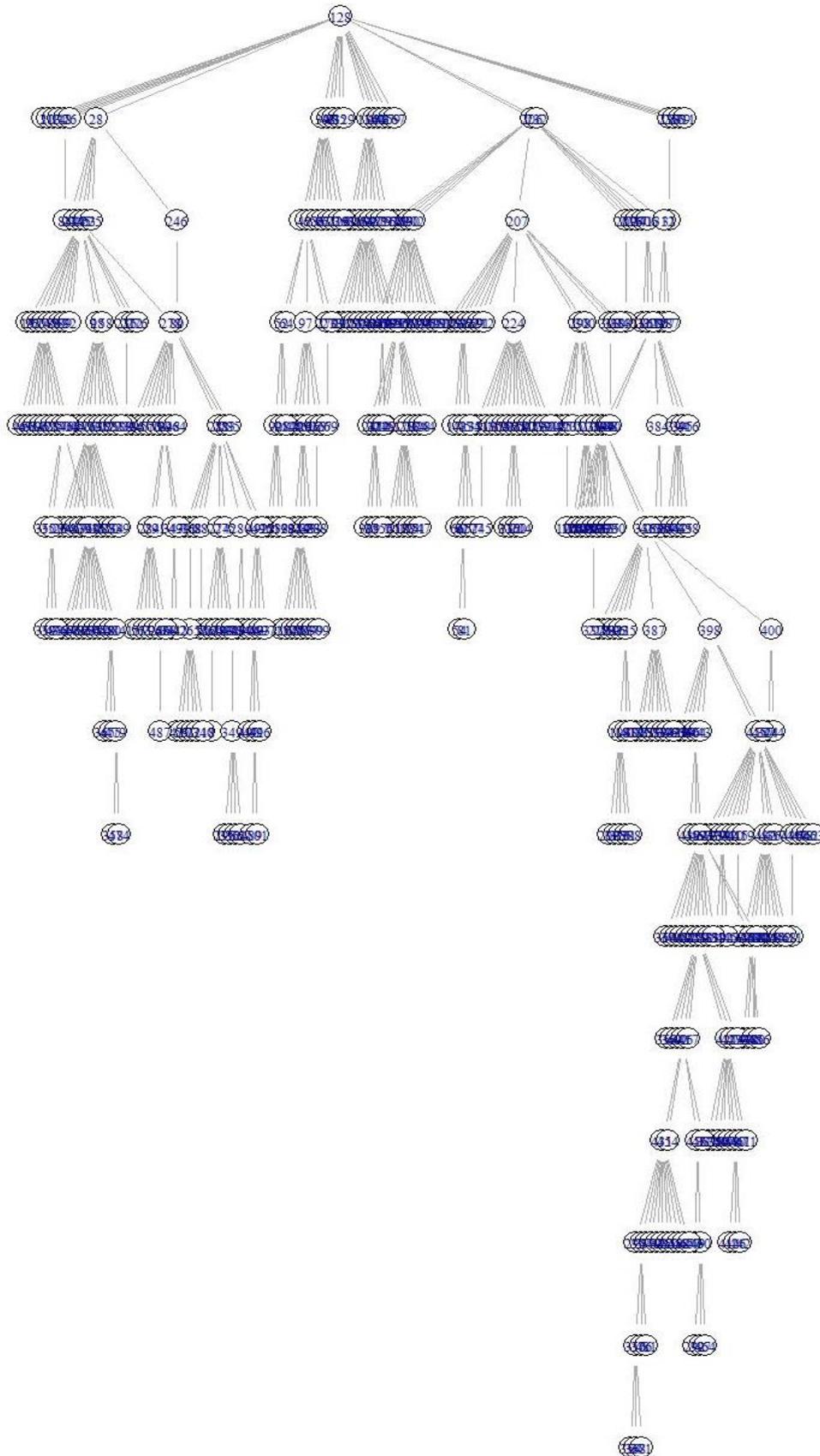


**Representación: LNB – BD: Cell cycle**  
**Costo 1: 156.9 - Costo 2: 78.2**

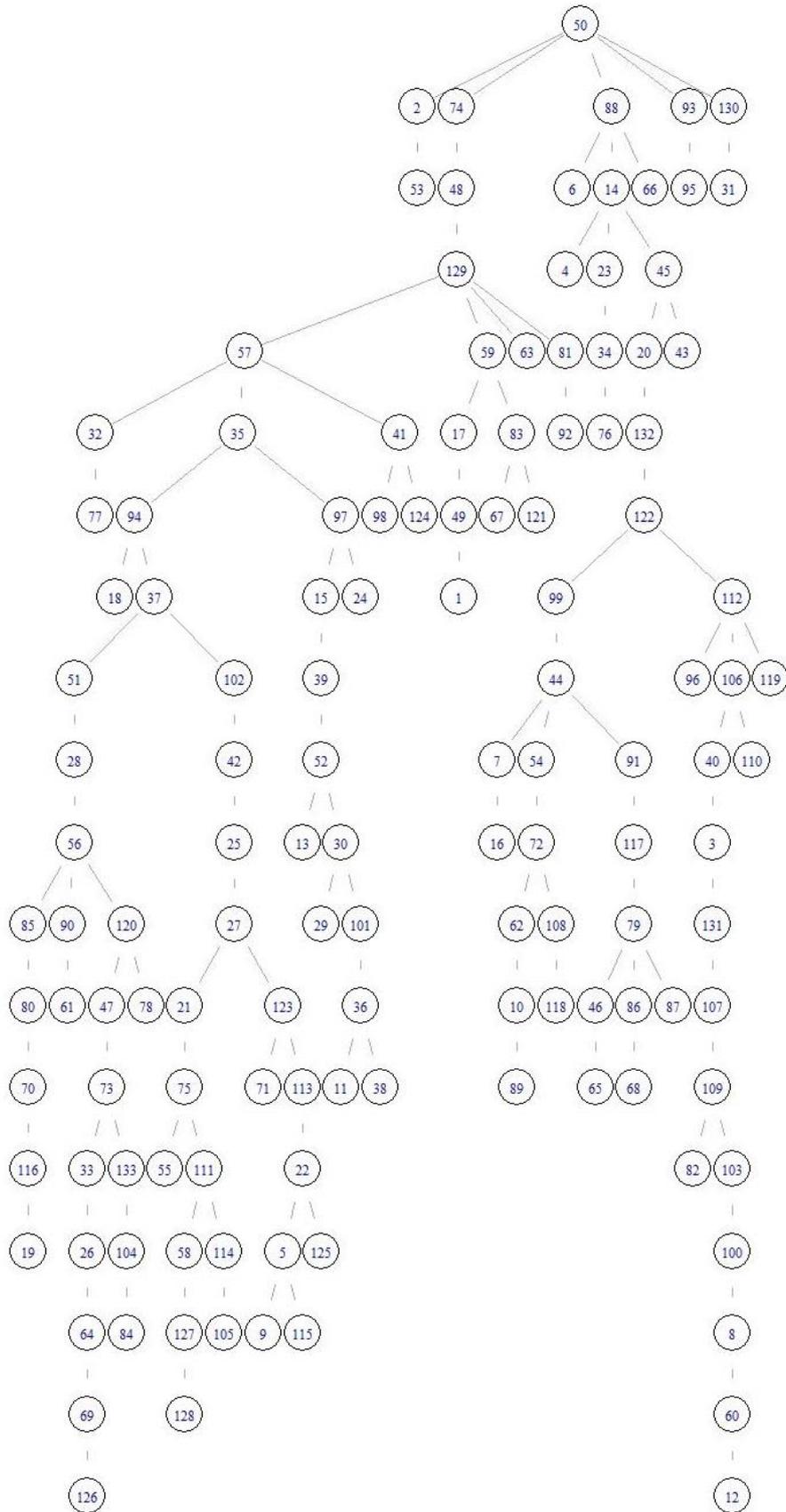




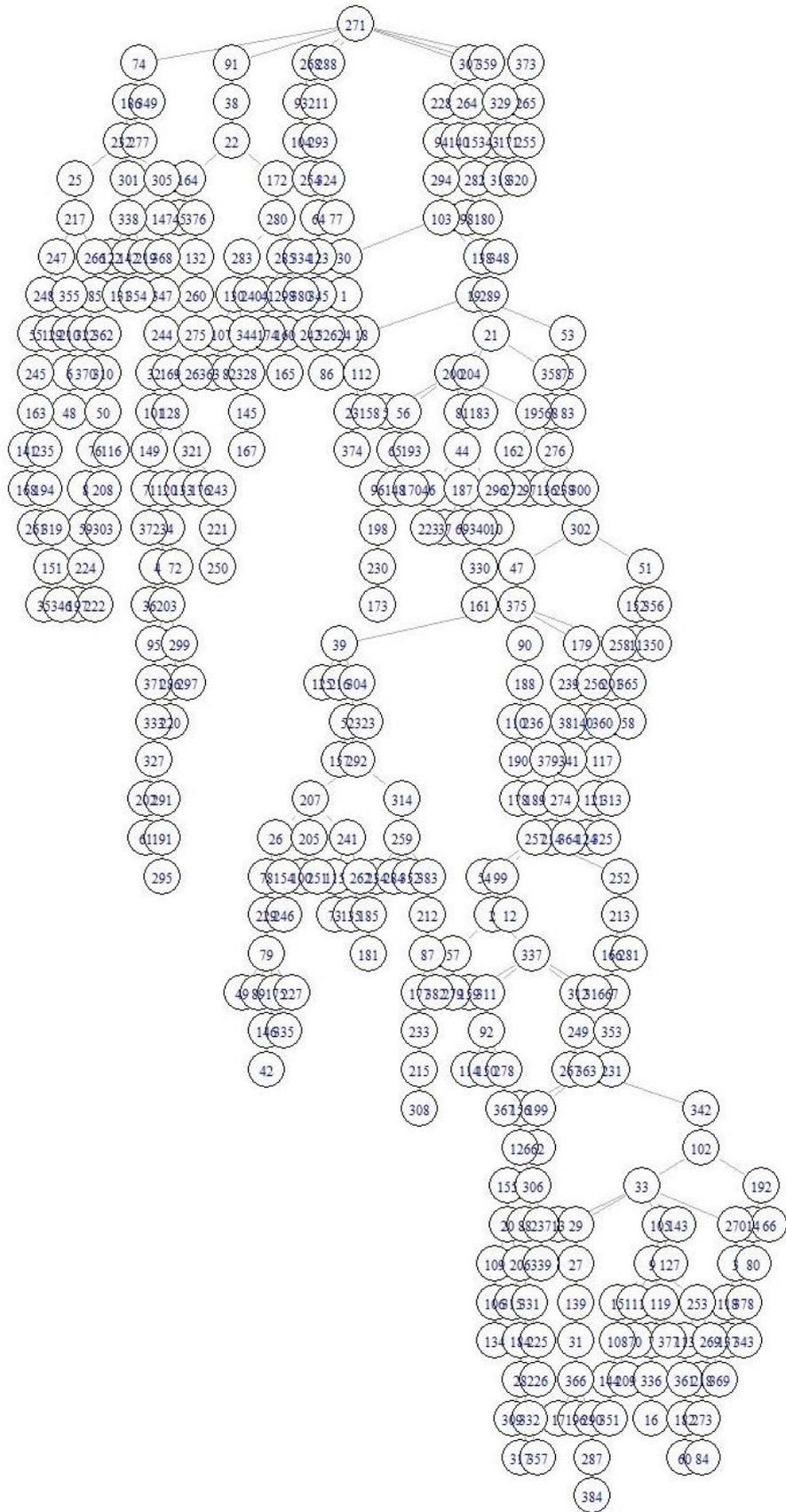
**Representación: LNB – BD: Serum**  
**Costo 1: 145.7 - Costo 2: 56.4**



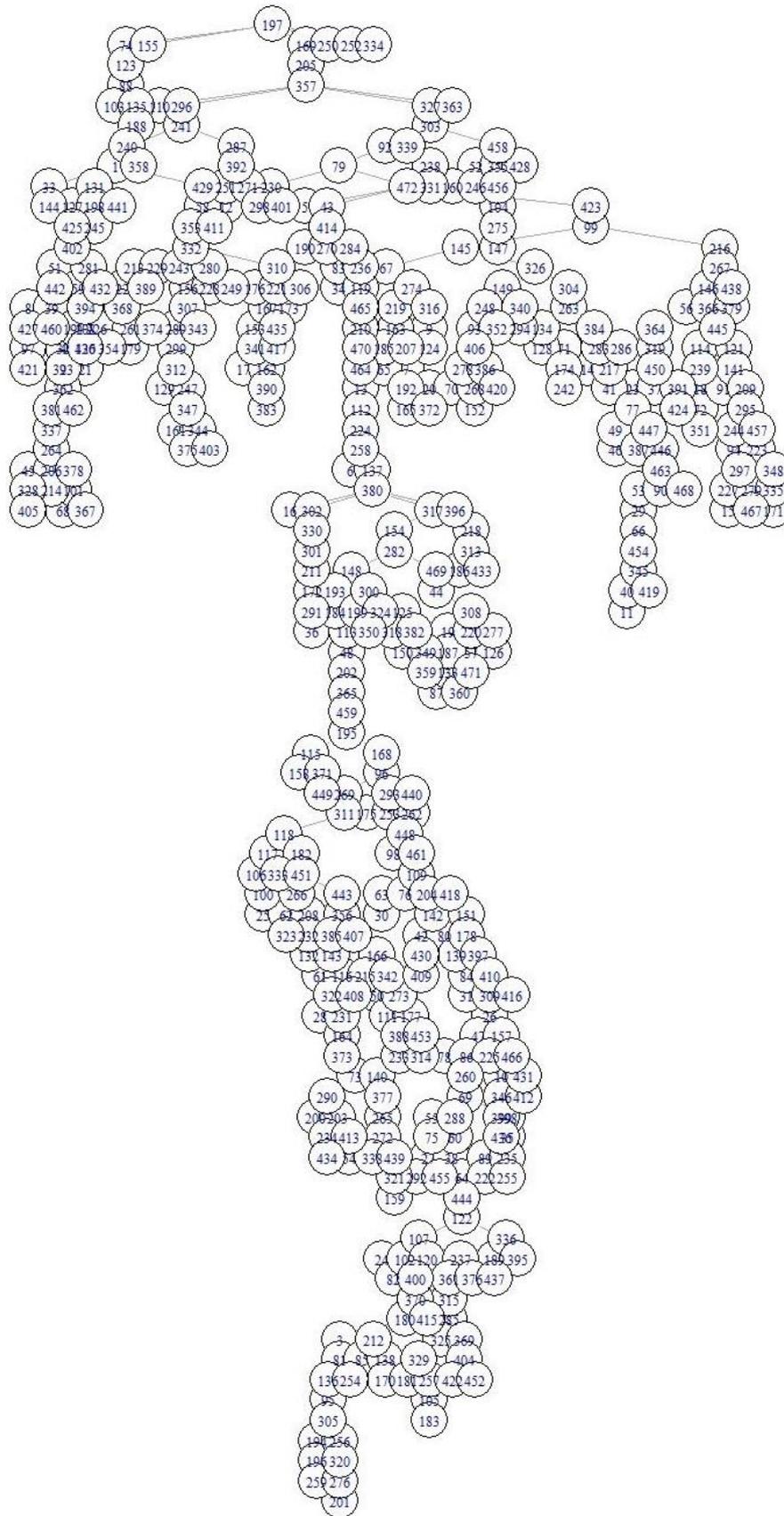
**Representación: Prüfer Number – BD: Arabidopsis**  
**Costo 1: 72 - Costo 2: 44.3**



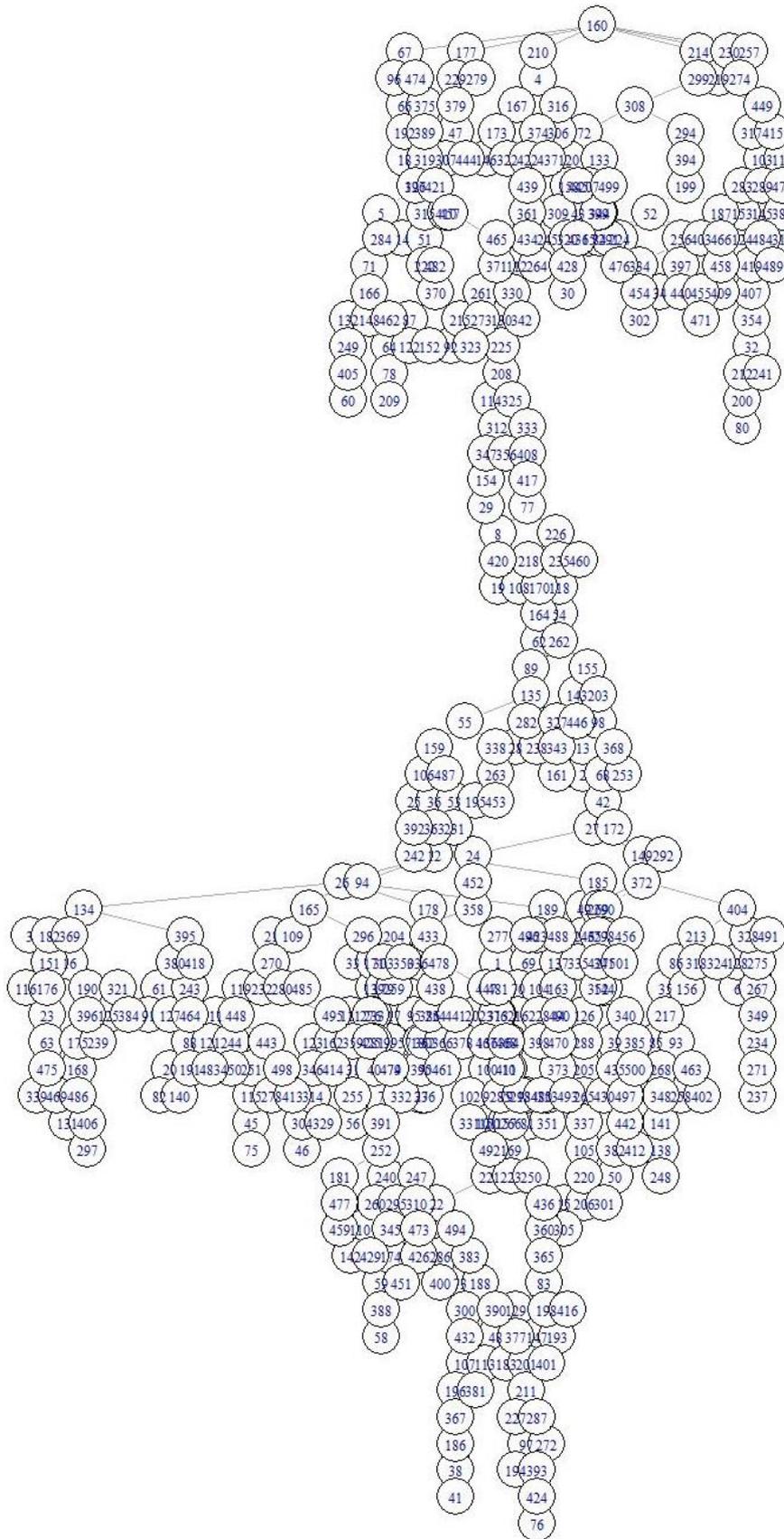
**Representación: Prüfer Number – BD: Cell Cycle**  
**Costo 1: 226 - Costo 2: 211.1**



**Representación: Prüfer Number – BD: Sporulation**  
**Costo 1: 269.2 - Costo 2: 199.2**



**Representación: Prüfer Number – BD: Serum**  
**Costo 1: 294.1 - Costo 2: 208.9**



## ANEXO 2. HV COMPLETO DE CADA BASE DE DATOS

**Cuadro 3.16.** Hipervolumen completo de Arabidopsis

ITERACIÓN	VALORES HV DE LAS REPRESENTACIONES		
	<i>Prüfer</i>	<i>LNB</i>	<i>Edge</i>
1	0,0555431707538925	0,899009453632752	0,000799431268573055
2	0,0682092280056378	1	0,00111793125294291
3	0,0383946971668106	0,927854670686287	0,00124947775038285
4	0,078247338607772	0,963105430169142	0,00411772608880829
5	0,0493034855214259	0,946836513728086	0,0153700933065181
6	0,0409377440520326	0,958289331592295	0,00134910171465429
7	0,057067123986436	0,946832515510804	0,00318045331144871
8	0,0744952495649453	0,938651323228852	0,000786961129335193
9	0,0345921050677923	0,969699822453435	0,00682092977669497
10	0,0386607301952065	0,970544200674121	0,00156814866977484
11	0,0699484822093966	0,974975649420899	0,00643484737247595
12	0,0563993620109798	0,971663323569843	0,00231139354170067
13	0,0474727831762307	0,935647665262563	0,00666170116077315
14	0,0736536317342247	0,978297419382425	0,00688698211488348
15	0,0297124945647199	0,934954291025383	0,0199237293702616
16	0,0450397242289228	0,964411081789289	0,000891479813820219
17	0,0729370110846146	0,977419065666826	0,000791898218391311
18	0,0545765431310554	0,980525616600617	0,00559734892753443
19	0,0535957619954138	0,945022518943127	0,00304661977844964
20	0,0530895061673852	0,946673426508536	0,0022442850429773

Fuente: (Elaboración propia, 2019).

**Cuadro 3.17.** Hipervolumen completo de Cell cycle

ITERACIÓN	VALORES HV DE LAS REPRESENTACIONES		
	<i>Prüfer</i>	<i>LNB</i>	<i>Edge</i>
1	0,0263669012503811	0,917657268650559	0,00638860724452141
2	0,0362843178854165	0,944692742409351	0,0125627917568969
3	0,0318389748651853	0,895840071975036	0,00715910142561613
4	0,0581886521762857	0,952225549720191	0,00974231612280405
5	0,0540831039484553	0,936832973689619	0,00411562277252953
6	0,0477882726062541	0,9433574733105	0,00710189280144398
7	0,044447768614727	0,936816530841697	0,0122987772516874
8	0,0462593179038432	0,969963588933021	0,0114799502140186
9	0,0707172996471175	0,916343399924282	0,0069026662520537
10	0,053355746888677	0,914001467704905	0,00576259825548198
11	0,0292200821218005	0,931083605028212	0,0109824241898418
12	0,071037180306131	0,911002152894329	0,00915781663228906
13	0,0723919307742371	0,964262067651491	0,00890978377913451
14	0,0547162672097293	0,943051154106586	0,00769762622426038
15	0,0449858378178025	0,985048384785442	0,00841128916249829
16	0,0474518064248195	0,9752257706503	0,010195666500823
17	0,0451725637227392	0,926046106331737	0,0071706882595233
18	0,0468883594403173	0,927253821964799	0,00767770611775816
19	0,0464121532831896	0,925160661366874	0,00804677450605501
20	0,0586944227330969	0,942810580636314	0,00629993246284549

Fuente: (Elaboración propia, 2019).

**Cuadro 3.18.** Hipervolumen completo de Sporulation

ITERACIÓN	VALORES HV DE LAS REPRESENTACIONES		
	<i>Prüfer</i>	<i>LNB</i>	<i>Edge</i>
1	0,066870524592029	0,927997906942251	0,0107694818513778
2	0,095270651697844	0,981746183753532	0,0140810990317673
3	0,069974974266621	0,99463706801253	0,018294431011256
4	0,0763479341514574	0,931972138834652	0,0144206601498514
5	0,0965582543695889	0,97349185665937	0,0216074650286625
6	0,0728729473779762	0,973042291739399	0,0158110997337118
7	0,10552936027505	0,973731553629986	0,00973243946991808
8	0,0756500914491053	0,941428759047206	0,0115197350722563
9	0,0798997325490529	0,959310533437272	0,0174230445385718
10	0,0361324539606554	0,92443733467545	0,015986335324559
11	0,0695093458515142	0,929997982049667	0,0157184247907266
12	0,0987390157159848	0,937139340638401	0,0183739234037403
13	0,0924961257555687	0,972690581650645	0,0129660796438205
14	0,0685335944777737	0,970483366141042	0,0136978249173485
15	0,116314216872299	0,958354323238502	0,0197705497754377
16	0,0441004031073604	0,972897342215054	0,02182715234343
17	0,0946557214613009	0,959566268814791	0,0195920900849225
18	0,0955529061185957	0,971739523455683	0,0130202142416936
19	0,0862376093729142	0,954151871320049	0,0218286661708118
20	0,0349103352464848	0,94765361895965	0,010729197966087

Fuente: (Elaboración propia, 2019).

**Cuadro 3.19.** Hipervolumen completo de Serum

ITERACIÓN	VALORES HV DE LAS REPRESENTACIONES		
	<i>Prüfer</i>	<i>LNB</i>	<i>Edge</i>
1	0,0120674892415259	0,983707564675148	0,00117189421149399
2	0,00603927954239832	0,988694750871047	0
3	0,00943222108810682	0,982307633534601	0,000157398826327731
4	0,00539152491125389	0,982293205364636	0,00124620626937259
5	0,0121186054656908	0,974755877214112	0,00103922342468314
6	0,0126697661199724	0,988147210684199	0,000141939430957018
7	0,00658349810390748	0,999411873685661	0,00206725733657296
8	0,00427044410015003	0,977937050524368	0,000111885231067708
9	0,00725360102665162	0,974210348740777	0,00182761463322773
10	0,00437845526011885	0,985160749115467	0,000309278526341712
11	0,00357948564610004	0,997646879280928	0,00238465724298254
12	0,0106739077720186	0,963523065083062	0,00252481131111705
13	0,00762060001640166	0,970076625327299	0,00331413638871646
14	0,00732792915335887	0,980387596085993	0,0000645847003896953
15	0,00590876124289876	0,994492763593557	0,00163128828502842
16	0,00836241547980962	0,987266149241991	0,000361324779201473
17	0,00865810511964181	0,988637117059413	0,00151255542453339
18	0,00313940397510938	0,987496091288175	0,00111291485212561
19	0,00772750801777407	0,995867988064271	0,000137036135239641
20	0,00729032265822517	0,996225516088183	0,00099676206624571

Fuente: (Elaboración propia, 2019).