

**Librería prototipo para el análisis de multifractalidad y robustez usando técnicas de
inteligencia artificial**

Carlos Andrés Delgado Saavedra, Ing

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Santiago de Cali
2018**

**Librería prototipo para el análisis de multifractalidad y robustez usando técnicas de
inteligencia artificial**

Carlos Andrés Delgado Saavedra, Ing

Código 1301662

carlos.andres.delgado@correounivalle.edu.co

**Documento presentado como requisito parcial para la obtención del
título en Magíster en Ingeniería énfasis en Ciencias de la Computación**

Director

Víctor Andrés Bucheli Guerrero, Ph.D.

victor.buchelli@correounivalle.edu.co

Universidad del Valle

Facultad de ingeniería

Escuela de ingeniería de sistemas y computación

Santiago de Cali

2018

Trabajo de grado presentado por
Carlos Andrés Delgado Saavedra, Ing.
Como requisito parcial para la obtención del título de Magíster en Ingeniería énfasis en Ciencias
de la Computación

Víctor Andrés Bucheli Guerrero, Ph.D.
Director

Oscar Bedoya Leyva, Ph.D.
Jurado

Angel de la Encarnación García Baños, Ph.D.
Jurado

Agradecimientos

En primer lugar, a mi familia por el apoyo incondicional en mis estudio de pregrado y posgrado.

A mis amigos, por su apoyo moral, en especial a Karen Martinez por mostrarme que con disciplina y constancia se pueden alcanzar las metas y a Joshua Triana por sus invaluable consejos.

Al profesor Angel García, a quien considero un modelo a seguir, por su apoyo académico y consejos durante todo este tiempo de estudio. Así mismo, al profesor Juan Francisco Díaz por sus enseñanzas durante mi trabajo de pregrado, en el cual adquirí la abstracción necesaria para enfrentar este proyecto.

Finalmente, a mi director Víctor Bucheli por su paciencia, por el tiempo dedicado y sus consejos, los cuales fueron muy importantes para la realización de este proyecto.

Tabla de Contenido

Resumen	XI
Introducción	XIII
1. Contexto y objetivos	1
1.1. Planteamiento del problema	1
1.2. Pregunta de investigación	2
1.3. Hipótesis	2
1.4. Justificación	2
1.5. Objetivos	2
1.5.1. Objetivo general	2
1.5.2. Objetivos específicos y resultados esperados	3
1.6. Alcances	3
1.6.1. Limites computacionales	3
1.6.2. Solución óptima	3
1.6.3. Librería prototipo	3
1.6.4. Validez de las pruebas	4
2. Revisión de literatura y Estado del arte	5
2.1. Marco conceptual redes complejas	5
2.1.1. Redes complejas	5
2.1.2. Glosario	7
2.2. Estado del arte de análisis fractal y multifractal de redes complejas	8
2.2.1. Análisis multifractal	8
2.2.2. Aplicaciones prácticas	9
2.2.3. Herramientas para el análisis fractal y multifractal	10
2.3. Estado del arte de análisis de robustez de redes complejas	11
2.3.1. Medida de la robustez	11
2.3.2. Aplicaciones del análisis de robustez	12
2.3.3. Herramientas para el análisis de robustez	13
2.4. Estado del arte en algoritmos de inteligencia artificial en redes complejas	14
2.4.1. Patrones en redes complejas	14
2.4.2. Algoritmos de inteligencia artificial aplicados en redes complejas	14
2.5. Resumen y conclusiones del capítulo	16

3. Generación de redes fractales	17
3.1. Redes fractales	17
3.2. Algoritmo para generar redes fractales	18
3.2.1. Red (2,2)-flower	18
3.2.2. Red (1,3)-flower	19
3.3. Resumen y conclusiones del capítulo	21
4. Análisis multifractal	22
4.1. Introducción	22
4.2. Estrategias de cubrimiento de cajas	22
4.3. Algoritmo SandBox	25
4.4. Análisis de los algoritmos	25
4.4.1. Redes libres de escala	26
4.4.2. Redes de mundo pequeño	28
4.4.3. Redes aleatorias	30
4.4.4. Red observada de Ecoli	32
4.4.5. Redes fractales	34
4.5. Estrategias evolutivas	36
4.5.1. El algoritmo	36
4.6. Estrategias Recocido Simulado	37
4.7. Estudio de los algoritmos de inteligencia artificial	38
4.7.1. Red libre de escala	38
4.7.2. Red de mundo pequeño	39
4.7.3. Red aleatoria	40
4.7.4. Red observada bacteria Ecoli	41
4.7.5. Red fractal	42
4.7.6. Análisis de los algoritmos	43
4.8. Análisis de los tiempos de ejecución	43
4.9. Resumen y conclusiones del capítulo	45
5. Análisis de robustez	46
5.1. Análisis de robustez	46
5.2. Estrategias de ataque	46
5.3. Pruebas y análisis de robustez en diferentes redes	47
5.3.1. Pruebas en red libre de escala	48
5.3.2. Pruebas red de mundo pequeño	49
5.3.3. Pruebas red aleatoria	50
5.3.4. Prueba en red observada Celengs	51
5.3.5. Prueba red fractal	52
5.4. Resumen y conclusiones del capítulo	53
6. Análisis de multifractalidad y robustez	55
6.1. Análisis multifractal	55
6.1.1. Relación entre multifractalidad y robustez	55
6.1.2. Discusión de resultados	68
6.2. Resumen y conclusiones del capítulo	69

7. Implementación de la librería	70
7.1. Desarrollo de la librería	70
7.1.1. Procesamiento de redes con SNAP	70
7.1.2. Arquitectura de la librería	71
7.2. Requerimientos de la librería	72
7.2.1. Dependencias de software	72
7.2.2. Recomendaciones de capacidad de cómputo	72
7.3. Funciones provistas	73
7.3.1. Análisis multifractal	73
7.3.2. Análisis de robustez	73
7.3.3. Algoritmos evolutivos	74
7.3.4. Algoritmos de recocido simulado	74
7.4. Documentación de la librería	74
7.4.1. Desarrollo de la documentación	74
7.4.2. Ejemplos de la documentación	75
7.5. Resumen y conclusiones del capítulo	76
8. Conclusiones y trabajos futuros	77
8.1. Conclusiones	77
8.2. Trabajos futuros	79
9. Bibliografía	80
Anexos	86
Anexo A: Descripción de redes de prueba	86
Anexo B: Descripción de protocolos de prueba	88
Anexo C: Resultados	89
Resultados análisis de multifractalidad	89
Resultados análisis de robustez	92

Lista de Figuras

2.1. Ejemplo de red de contactos sexuales. Tomado de [1]	6
3.1. Proceso iterativo de construcción de las $(x - y)$ -flowers. Tomado de [2]	17
3.2. Proceso iterativo de generación de redes (x, y) -flowers. Tomado de [2]	18
3.3. Matriz de adyacencia para la generación 0 de (2,2)-flower	18
3.4. Proceso para construir la generación 1 de (2,2)-flower. Observe que las aristas eliminadas son resaltadas en negrilla	19
3.5. Matriz de adyacencia para la generación 0 de (1,3)-flower	19
3.6. Proceso para construir la generación 1 de (1,3), flower. Observe que las aristas que se conservan son resaltadas en negrilla	20
4.1. Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB	26
4.2. Exponentes de masa para red libre de escala 4000 nodos	27
4.3. Dimensión fractal generalizada para red libre de escala 4000 nodos	27
4.4. Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB	28
4.5. Exponentes de masa para red de mundo pequeño con 5000 nodos y $p = 10\%$	29
4.6. Dimensión fractal generalizada para red de mundo pequeño con 5000 nodos y $p = 10\%$	29
4.7. Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB	30
4.8. Exponentes de masa para red aleatoria de 5620 nodos	31
4.9. Dimensión fractal generalizada para para red aleatoria de 5620 nodos	31
4.10. Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB	32
4.11. Exponentes de masa para red observada ecoli	33
4.12. Dimensión fractal generalizada para observada ecoli	33
4.13. Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB	34
4.14. Exponentes de masa para red fractal (2,2)-flower	35
4.15. Dimensión fractal generalizada para red fractal (2,2)-flower	35
4.16. Multifractalidad en red libre de escala con 4000 nodos	38
4.17. Evolución de la función de evaluación en red libre de escala con 4000 nodos	39
4.18. Multifractalidad en red de mundo pequeño de 5000 nodos con $p = 10\%$	39

4.19. Evolución de la función de evaluación en red de mundo pequeño de 5000 nodos con $p = 10\%$	40
4.20. Multifractalidad en red aleatoria con 5620 nodos	40
4.21. Evolución de la función de evaluación en red aleatoria con 5620 nodos	41
4.22. Multifractalidad en red observada de Ecoli	41
4.23. Evolución de la función de evaluación en red observada de Ecoli	42
4.24. Multifractalidad en red generada fractal (2,2)-flower	42
4.25. Evolución de la función de evaluación en red fractal (2,2)-flower	43
5.1. Medida componente de gigante	48
5.2. Medida componente de promedio de camino más cortos	48
5.3. Medida componente de gigante	49
5.4. Medida componente de promedio de camino más cortos	50
5.5. Medida componente de gigante	50
5.6. Medida componente de promedio de camino más cortos	51
5.7. Medida componente de gigante	51
5.8. Medida componente de promedio de camino más cortos	52
5.9. Medida componente de gigante	52
5.10. Medida componente de promedio de camino más cortos	53
6.1. Análisis de multifractalidad de red libre de escala a ataque aleatorio	56
6.2. Análisis de multifractalidad de red libre de escala a ataque por grado	56
6.3. Análisis de multifractalidad de red libre de escala a ataque por centralidad	57
6.4. Análisis de multifractalidad de red libre de escala a ataque por estrategia evolutiva	57
6.5. Análisis de multifractalidad de red libre de escala a ataque por estrategia recocida simulada	58
6.6. Análisis de multifractalidad de red de mundo pequeño ataque aleatorio	58
6.7. Análisis de multifractalidad de red de mundo pequeño a ataque por grado	59
6.8. Análisis de multifractalidad de red de mundo pequeño a ataque por centralidad	59
6.9. Análisis de multifractalidad de red de mundo pequeño a ataque por estrategia evolutiva	60
6.10. Análisis de multifractalidad de red de mundo pequeño a ataque por estrategia recocida simulada	60
6.11. Análisis de multifractalidad de red aleatoria a ataque aleatorio	61
6.12. Análisis de multifractalidad de red aleatoria a ataque por grado	61
6.13. Análisis de multifractalidad de red aleatoria a ataque por centralidad	62
6.14. Análisis de multifractalidad de red aleatoria a ataque por estrategia evolutiva	62
6.15. Análisis de multifractalidad de red aleatoria a ataque por estrategia recocida simulada	63
6.16. Análisis de multifractalidad de red observada de Celegens a ataque aleatorio	63
6.17. Análisis de multifractalidad de red observada de Celegens a ataque por grado	64
6.18. Análisis de multifractalidad de red observada de Celegens a ataque por centralidad	64
6.19. Análisis de multifractalidad de red observada de Celegens a ataque por estrategia evolutiva	65
6.20. Análisis de multifractalidad de red observada de Celegens a ataque por estrategia recocida simulada	65
6.21. Análisis de multifractalidad de red (1,3)-flower a ataque aleatorio	66

6.22. Análisis de multifractalidad de red (1,3)-flower a ataque por grado	66
6.23. Análisis de multifractalidad de red (1,3)-flower a ataque por centralidad	67
6.24. Análisis de multifractalidad de red (1,3)-flower a ataque por estrategia evolutiva	67
6.25. Análisis de multifractalidad de red (1,3)-flower a ataque por estrategia recocida simulada	68
7.1. Diagrama conceptual de L.A.M.F.R.I.A	71
7.2. Pagina inicial de la documentación de la librería	75
7.3. Información específica sobre una función de la librería	75
7.4. Resultado búsqueda información de la documentación en de la librería	76
9.1. Análisis multifractal red libre de escala 2000 nodos	89
9.2. Análisis multifractal red libre de escala 4000 nodos	89
9.3. Análisis multifractal red libre de escala 8000 nodos	89
9.4. Análisis multifractal Mundo pequeño $p=5\%$	89
9.5. Análisis multifractal Mundo pequeño $p=10\%$	90
9.6. Análisis multifractal Mundo pequeño $p=20\%$	90
9.7. Análisis multifractal red aleatoria 1991 nodos	90
9.8. Análisis multifractal red aleatoria 3737 nodos	90
9.11. Análisis multifractal red observada Cerevisiae	90
9.9. Análisis multifractal red observada ECOLI	91
9.10. Análisis multifractal red observada Celengs	91
9.12. Análisis multifractal red fractal (2,2)-flower	91
9.13. Análisis multifractal red observada (1,3)-flower	91
9.14. Análisis de robustez de red libre de escala de 2000 nodos por tamaño de componente gigante	92
9.15. Análisis de robustez de red libre de escala de 2000 nodos por diámetro de camino más corto	92
9.16. Análisis de robustez de red libre de escala de 4000 nodos por tamaño de componente gigante	92
9.17. Análisis de robustez de red libre de escala de 4000 nodos por diámetro de camino más corto	92
9.18. Análisis de robustez de red libre de escala de 8000 nodos por tamaño de componente gigante	92
9.19. Análisis de robustez de red libre de escala de 8000 nodos por diámetro de camino más corto	92
9.20. Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=10\%$ tamaño de componente gigante	93
9.21. Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=10\%$ por diámetro de camino más corto	93
9.22. Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=20\%$ tamaño de componente gigante	93
9.23. Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=20\%$ por diámetro de camino más corto	93
9.24. Análisis de robustez de red aleatoria con 1991 nodos por tamaño de componente gigante	93
9.25. Análisis de robustez de red aleatoria con 1991 nodos por diámetro de camino más corto	93

9.26. Análisis de robustez de red aleatoria con 3373 nodos por tamaño de componente gigante	94
9.27. Análisis de robustez de red aleatoria con 3373 nodos por diámetro de camino más corto	94
9.28. Análisis de robustez de red aleatoria con 5620 nodos por tamaño de componente gigante	94
9.29. Análisis de robustez de red aleatoria con 5620 nodos por diámetro de camino más corto	94
9.30. Análisis de robustez de red observada Ecoli por tamaño de componente gigante . . .	94
9.31. Análisis de robustez de red observada Ecoli por diámetro de camino más corto . . .	94
9.32. Análisis de robustez de red observada Celegens por tamaño de componente gigante .	95
9.33. Análisis de robustez de red observada Celegens por diámetro de camino más corto .	95
9.34. Análisis de robustez de red observada Cerevisae por tamaño de componente gigante	95
9.35. Análisis de robustez de red observada Cerevisae por diámetro de camino más corto .	95
9.36. Análisis de robustez de red fractal (2,2)-flower por tamaño de componente gigante .	95
9.37. Análisis de robustez de red fractal (2,2)-flower por diámetro de camino más corto . .	95
9.38. Análisis de robustez de red fractal (1,3)-flower por tamaño de componente gigante .	96
9.39. Análisis de robustez de red fractal (1,3)-flower por diámetro de camino más corto . .	96

Lista de tablas

1.1. Objetivos específicos	3
2.1. Herramientas para el análisis multifractal encontradas en la literatura	11
2.2. Herramientas para el análisis de robustez encontradas en la literatura	13
3.1. Propiedades estructurales de generaciones 0,1,2 y 3 de la red fractal (2,2)-flower . . .	19
3.2. Propiedades estructurales de generaciones 0,1,2 y 3 de la red fractal (1,3)-flower . . .	20
4.1. Comparación de los datos obtenidos con los del artículo de Wang[3]	28
4.2. Comparación de los datos obtenidos con los del artículo de Wang[3]	30
4.3. Comparación de los datos obtenidos con los del artículo de Wang[3]	32
4.4. Comparación de los datos obtenidos con los del artículo de Wang[3]	34
4.5. Comparación de los datos obtenidos de los diferentes métodos de análisis multifractal	36
4.6. Tiempos de ejecución para diferentes redes en segundos	44
9.1. Redes utilizadas en el proyecto	87

Resumen

Las redes complejas permiten observar diferentes sistemas a través de una representación basada en grafos. En la literatura se presentan múltiples métricas que permiten caracterizar estas redes, como lo son la distribución de grado, coeficiente de agrupamiento, coeficiente de agrupación, etc.

Dado que las redes complejas son muy variadas, el estudio de sus estructuras internas permite obtener más información que otras métricas que consideran la red como un todo. Este estudio puede ser realizado a través del análisis de multifractalidad, el cual permite estudiar estructuras que se repiten y que conforman la macro estructura de la red.

El análisis multifractal consiste en realizar el cálculo de las dimensiones fractales de una red, las cuales permiten observar cómo cambia su estructura interna a medida que la red es escalada. A partir de este análisis se puede determinar que una red es monofractal o que conserva su estructura interna a medida que se escala, lo que indica que la red tiende a estar compuesta por una sola estructura. Así mismo, se puede determinar que una red es multifractal, lo que indica que una red tiende a estar compuesta por múltiples estructuras.

En diferentes trabajos realizados en la literatura se han estudiado múltiples redes a través del análisis multifractal y su relación con las diferentes métricas que consideran la red como un todo. Sin embargo, esto no permite estudiar con mayor detalle las estructuras internas de una red, por lo que la propuesta de este trabajo es relacionarlo con el análisis de robustez para observar cómo comporta la estructura de una red ante perturbaciones.

En consecuencia, realizar un estudio sobre la relación entre multifractalidad y robustez busca aportar una herramienta en la caracterización de las redes complejas. Este estudio permite tener una imagen de cómo se comportan las estructuras internas de una red a medida que se pierden nodos y conexiones producto de perturbaciones en su estructura. Varios autores han indicado que las métricas más comunes en la literatura no permiten obtener esta caracterización.

Además, se evidencia en la literatura las técnicas de análisis multifractal basadas en cobertura de cajas son estrategias de búsqueda intensiva. lo que las convierte en computacionales intratables. Por esta razón, se proponen heurísticas para técnicas de inteligencia artificial, con el fin de optimizar la estrategia de búsqueda y así reducir los tiempos de cómputo. Para cumplir este objetivo, se propone una estrategia basada en la selección de los nodos que serán los centros de las cajas para el cubrimiento. Con esto, se diseñan un algoritmo evolutivo y un algoritmo de recocido simulado.

Metodológicamente, para el estudio de la relación entre la multifractalidad y robustez se realizan experimentos utilizando redes generadas libres de escala, de mundo pequeño, aleatorias y redes

reportadas en la literatura. Estos experimentos consisten en calcular las dimensiones fractales cada vez que la red pierde un porcentaje de sus nodos producto de una perturbación.

Los resultados encontrados permiten observar que las redes de mundo pequeño tienden a ser monofractales o de estructura homogénea, tal como se ha reportado en la literatura. Adicionalmente, esta estructura se conserva a medida que las redes de mundo pequeño son afectadas por ataques, tales como la pérdida de nodos. Así mismo, la relación entre las dimensiones fractales y la robustez, indican preliminarmente que la dimensión fractal se reduce a medida que la red pierde nodos. También, las pruebas indican que las estrategias de inteligencia artificial mejoran el tiempo requerido para el análisis multifractal, ya que permiten aplicar una estrategia para la ubicación de las cajas, pues los algoritmos presentes en la literatura seleccionan los centros de las cajas de forma repetitiva, de tal forma se aproxime al cubrimiento óptimo.

Introducción

Las redes complejas permiten modelar sistemas observados, como es el caso de los sistemas biológicos[4], en los cuales se pueden realizar modelos de interacción entre proteínas, de organismos en un ecosistema, mecanismos de comunicación celular, entre otros. Otro ejemplo son los sistemas de transporte se modelan a partir de las interacciones entre los vehículos, los pasajeros y el sistema vial de una ciudad[5]. Así mismo, los sistemas sociales se modelan a partir de las interacciones entre las personas, un caso de esto es la persuasión como un proceso en cascada[6]. Las redes también se utilizan en el estudio de transacciones e intercambio de bienes entre organizaciones[7], las redes de citaciones de artículos científicos[8], entre otras. En consecuencia, las redes complejas son una herramienta que proporciona un soporte matemático para el estudio de diversos problemas en diferentes áreas del conocimiento.

En las ciencias de la computación las redes complejas[9] son representadas como grafos, en los cuales los vértices o nodos representan una entidad en el modelo y las aristas las relaciones entre ellas. A partir de esta representación se pueden estudiar algunas propiedades estructurales de las redes, como es la distribución de grado que permite visualizar cómo están conectados los vértices en la red. Otras propiedades estructurales son las medidas de cercanía por centralidad y por intermediación que indican si un vértice hace parte de los caminos más cortos entre un par de vértices, así como la cercanía por vecindad, que se utiliza para medir que tan cerca está del centro de la red. Adicionalmente, una medida de gran importancia es la longitud de los caminos promedios entre un cada par de vértices, de esta forma diferentes medidas se propone para caracterizar elementos clave dentro de la topología de una red.

En la literatura, las redes complejas pueden ser clasificadas de acuerdo a sus propiedades estructurales. Las categorías más utilizadas son: redes libres de escala y redes de mundo pequeño. Las redes libres de escala según Barabasi[10] se caracterizan por tener una distribución de grado acorde a una ley de potencia, en la cual pocos vértices tienen un grado muy elevado (hubs) y un gran número de estos con un grado pequeño. Otra característica de las redes libres de escala es que el coeficiente de agrupación decrece a medida que el grado de los hubs se incrementa, esto implica que los diferentes componentes se encuentran conectados por los hubs y además estas redes presentan una gran robustez a fallas aleatorias. Por su parte, las redes de mundo pequeño según Amaral y otros[11] son aquellas que presentan las siguientes propiedades: primero, la distancia más corta promedio entre un par de vértices es proporcional al logaritmo natural del número de nodos y el coeficiente de agrupamiento es grande, lo que indica que los vértices de la red tienden a estar fuertemente conexos.

Sin embargo, dada la variedad de estructuras y formas de las redes complejas, estas medidas no

son suficientes para estudiar la estructura de redes complejas debido a que las consideran como un todo[12]. El análisis multifractal puede aportar más información[13], ya que permite estudiar la topología de las estructuras interna de una red. Por otro lado, la robustez[14], permite analizar cómo cambia la estructura interna de las redes complejas a medida se comporta a perturbaciones de su estructura.

Para la medida de robustez, se tienen algoritmos basados en cobertura por cajas[15], los cuales son utilizados en el procesamiento de imágenes para el análisis fractal, con la diferencia que en redes complejas se utiliza un sistema de referencia basado en el diámetro de la red. La idea, es observar que la dimensión fractal L_B es la relación entre el diámetro de las cajas R_B y número de cajas N_B , que está dado por $N_B = R_B^{L_B}$. Sin embargo, esta medida ha demostrado no ser suficiente para el estudio de las estructuras internas de las redes[3], por ello se han desarrollado variantes que permiten realizar un análisis multifractal, el cual arroja más información sobre las estructuras de la red a medida que la red es escalada por un factor q . En el caso de la robustez, el análisis consiste en estudiar las propiedades estructurales de las redes a medida que esta pierde nodos o aristas, seleccionados a partir de alguna estrategia de ataque.

El estudio de una relación entre la multifractalidad y robustez, promete proveer un marco de trabajo para el estudio de las redes complejas, ya que, permite estudiar cómo se comporta la estructura de la red a medida que esta se ve afectada por algún tipo de ataque. La metodología de este estudio consiste en observar cómo se comportan las dimensiones fractales de una red a medida que se pierden nodos y aristas bajo alguna estrategia de eliminación. Esto permite ver cómo varía la estructura interna de acuerdo al tipo de ataque que se somete a una red.

Dado que estos algoritmo son computacionalmente intratables al realizar una búsqueda intensiva y repetitiva de los centros y estructuras de las cajas, se propone una heurística para estrategias evolutivas y recocido simulado, con el fin de reducir el número de computaciones que se deben realizar.

Al realizar un conjunto de experimentos con diferentes redes complejas, se encuentra que la estructura de una red de mundo pequeño tiende a ser una sola o monofractal, ya que las curvas de dimensión fractal no presentan una variación significativa a diferencia de las redes libres de escala. Así mismo, en el caso de la relación entre multifractalidad y robustez, la estructura interna de las redes libres de escala no presenta variación ante ataques aleatorios a diferencia de ataques que afectan a los hubs; en las redes de mundo pequeño, se encuentra una variación insignificante en la dimensión fractal, ya que su estructura interna es compacta. A partir de estos dos resultados, se encuentra un campo de estudio para ampliar el significado de esta relación.

En el capítulo 1, se mostrará la definición del problema y objetivos de este proyecto. En el capítulo 2, se realiza una revisión de la literatura existente para el análisis de multifractalidad y robustez. En los capítulos 4 a 7, se muestra el proceso de análisis de multifractalidad y robustez en algunas redes complejas seleccionadas. Finalmente, en el capítulo 8 se presentan las conclusiones y los trabajos futuros.

Capítulo 1

Contexto y objetivos

1.1. Planteamiento del problema

En sistemas complejos reales, tales como, las redes de interacciones entre proteínas, redes de transporte y redes sociales, las mediciones de multifractalidad y robustez son de gran importancia. La multifractalidad, por su parte, permite identificar subestructuras que son significativas en una red[16] y, por otra parte, la robustez es una métrica para el análisis de tolerancia a fallos[17]. Además, estas medidas también sirven de apoyo a áreas como la astronomía donde eventos como el viento solar puede ser estudiado identificado estructuras fractales[18] o en economía, donde algunos estudios evidencian la existencia multifractalidad dentro del comportamiento de los mercados[19].

Por lo tanto, proveer una librería que permita realizar estas mediciones en diferentes entornos de trabajo cobra importancia, ya que estas son de gran apoyo para diferentes áreas del conocimiento. En la literatura, existen varias técnicas que mediante estrategias de exploración, permiten realizar estas mediciones. Dichas estrategias realizan una búsqueda intensiva en el espacio de soluciones de estos problemas que son NP-Hard, por lo que se requiere una gran capacidad de procesamiento y tiempo de ejecución, sobre todo en redes de gran tamaño.

Como consecuencia, es necesario buscar técnicas que permitan identificar patrones para el diseño de estrategias que reduzcan el espacio de búsqueda. Por lo anterior, se propone utilizar técnicas de inteligencia artificial, pues estas permiten el diseño de soluciones basadas en el aprendizaje automático e identificación automática de patrones, con el objetivo de explorar eficientemente el espacio de soluciones.

Por lo tanto, se busca aplicar técnicas basadas en inteligencia artificial, para la medición de la multifractalidad y robustez en redes complejas, ya que gran parte de las estrategias encontradas en la literatura son algorítmicas, basadas en la combinatoria de posibles soluciones. Las técnicas algorítmicas requieren realizar un gran número de cálculos dentro de las redes complejas para obtener las mediciones.

1.2. Pregunta de investigación

¿Cómo desarrollar un algoritmo que a través de patrones o regularidades identificadas automáticamente en el proceso de medición de multifractalidad y robustez en redes complejas, permitan llevar a cabo computaciones más económicas que las presentes en la literatura?

1.3. Hipótesis

Se pueden identificar patrones o regularidades asociados con los parámetros del análisis multifractal y de robustez en redes complejas, que permitan diseñar soluciones computacionalmente más económicas que los algoritmos disponibles en la literatura.

1.4. Justificación

Este trabajo busca integrar el área de inteligencia artificial (IA) con el estudio de redes complejas, las técnicas de inteligencia artificial ofrecen soluciones basadas en la identificación de patrones y aprendizaje de problemas, que permiten diseñar soluciones que realizan una búsqueda eficiente en problemas computacionalmente NP-Hard.

Los algoritmos disponibles en la literatura para la medición de multifractalidad y robustez, como el algoritmo de Box Covering, a medida que exploran la red, requieren la evaluación de un conjunto de soluciones, lo que implica realizar cálculos en un espacio combinatorio. Como consecuencia, no son prácticos para redes gran tamaño, por esta razón, en este trabajo se aborda el uso de técnicas de inteligencia artificial para encontrar patrones o reglas que permitan plantear técnicas para realizar estas mediciones más rápidamente.

La identificación de patrones y regularidades en la medición de parámetros en redes complejas, ofrece una experiencia para el diseño de algoritmos y estrategias para el estudio de características de diferentes sistemas complejos en diferentes áreas del conocimiento, como es el caso de las ciencias sociales, donde las poblaciones pueden ser caracterizadas o clasificadas como comunidades a través de una medida de modularidad, o en el estudio de redes de relaciones entre personas definidas por redes de conocidos o citas. Otro caso, es el estudio de sistemas urbanos, como son las redes de transporte, en donde se pueden identificar estructuras que ayuden a tomar decisiones para enfrentar problemas como los atascos de tráfico.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar una librería prototipo de medición de parámetros para el análisis de la multifractalidad y robustez en redes complejas utilizando técnicas de inteligencia artificial.

1.5.2. Objetivos específicos y resultados esperados

Objetivo específico	Sección del documento
Implementar dos algoritmos encontrados en la literatura para la medición de multifractalidad y robustez.	Capítulo 2, Capítulo 4 sección 4.2 y sección 4.3
Identificar dos técnicas de inteligencia artificial que se puedan utilizar para la medición de parámetros en el análisis de multifractalidad y robustez en redes complejas.	Capítulo 4 secciones 4.5 y 4.6
Desarrollar e implementar funciones que permitan realizar la medición de parámetros para el análisis de multifractalidad y robustez en redes complejas, utilizando las técnicas de inteligencia artificial elegidas.	Capítulo 4 secciones 4.5 y 4.6, y Capítulo 7
Experimentar con los algoritmos basados en inteligencia artificial y dos algoritmos encontrados en la literatura.	Capítulo 4 sección 4.7, Capítulo 6 y Capítulo 8
Análisis comparativo entre las soluciones basadas en inteligencia artificial y dos algoritmos encontrados en la literatura.	Capítulo 4 sección 4.7 y Capítulo 6

Tabla 1.1: Objetivos específicos

1.6. Alcances

1.6.1. Límites computacionales

Se debe tener en cuenta que los recursos computacionales son limitados y el problema a solucionar es NP-Hard, por lo que podrían existir limitaciones en tiempo y memoria para procesar algunas redes.

1.6.2. Solución óptima

Debido a que el uso de las técnicas de inteligencia artificial implica un proceso de aprendizaje de cómo medir parámetros, no se puede garantizar que la solución encontrada sea la óptima, debido a que este proceso depende en gran medida de los datos que se utilizaron para su entrenamiento.

1.6.3. Librería prototipo

La librería que se va a construir es únicamente para propósito validación de las técnicas aplicadas en esta tesis, por lo tanto, no representa un producto que se pueda utilizar comercialmente y sin conocimientos en redes complejas.

1.6.4. Validez de las pruebas

Se van a realizar pruebas en redes complejas representativas, para realizar el análisis de los algoritmos diseñados en este trabajo, sin embargo, debido a que pueden existir redes con características únicas, en las que las soluciones basadas en técnicas de inteligencia artificial podrían presentar resultados insatisfactorios, las pruebas sólo validarán el trabajo realizado en un conjunto seleccionado de redes complejas.

Capítulo 2

Revisión de literatura y Estado del arte

2.1. Marco conceptual redes complejas

2.1.1. Redes complejas

Una red o grafo[1] es un conjunto de elementos, vértices o nodos los cuales están conectados por aristas. Los sistemas reales pueden ser modelados como redes. Algunas redes reales son la Internet, redes sociales de conocidos, redes de estructura en organizaciones, redes de intercambio de bienes, redes neuronales, redes metabólicas, redes de citaciones y muchas otras.

El estudio de redes, se puede realizar usando la teoría de grafos. Los estudios típicos en redes se aplican en áreas como la sociología donde se pueden modelar redes a partir de los resultados de cuestionarios acerca de las relaciones interpersonales como es el caso de la figura 2.1, en estos modelos sociales se pueden encontrar resultados emergentes como la aparición de grupos de individuos que se caracterizan por sólo tener contactos sexuales entre ellos.

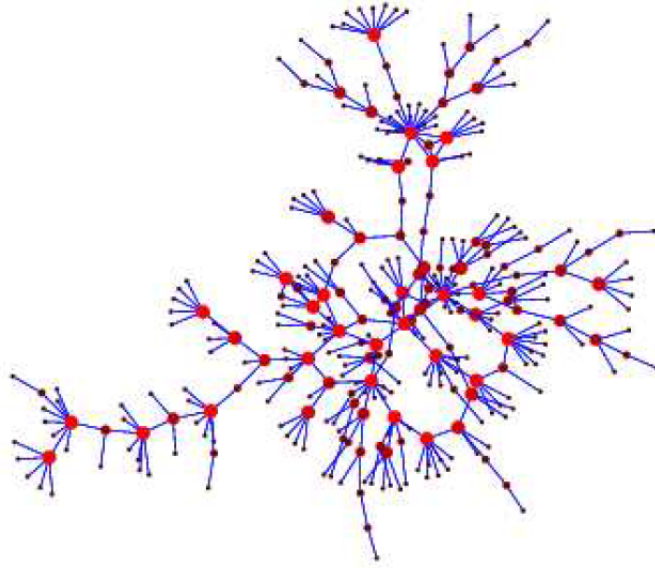


Figura 2.1: Ejemplo de red de contactos sexuales. Tomado de [1]

Definición formal

Una red o grafo se puede definir así:

$$G = (V, E) \quad (2.1)$$

Donde V es el conjunto de nodos y E es el conjunto de aristas. Una arista es un par de vértices $(i, j) \in V^2$, en el caso de los grafos dirigidos este par es ordenado. Los enlaces también pueden tener un peso asociado, el cual es un valor numérico. Las redes tienen las siguientes propiedades básicas:

1. Dos vértices distintos son adyacentes si y sólo si hay una arista que los conecta
2. Una arista es incidente a un vértice, si este hace parte de ella
3. El vecindario de un vértice son todos aquellos que se conectan directamente a través de una arista

Propiedades de redes complejas

Las propiedades de una red compleja son[20]:

- Distribución de grado: es la probabilidad que un nodo tenga un número determinado de conexiones
- Coeficiente de agrupación: es la probabilidad que tomando un número determinado de nodos se obtenga un grafo completo, el cual consiste en que todos los nodos se conectan entre sí.

- Longitud promedio de la red: Es el promedio de todas las distancias mínimas entre los vértices

Clasificación de las redes complejas

Las redes se clasifican de acuerdo a sus propiedades estructurales en:

1. Redes aleatorias, las cuales tienen una distribución de grado Poisson.
2. Redes libres de escala: las cuales tienen una distribución de grado de ley de potencia.
3. Redes de mundo pequeño: son aquellas redes cuyo coeficiente de agrupamiento es elevado y la distancia promedio entre los vértices es un valor proporcional al logaritmo del número de nodos.

Modelos de generación de redes complejas

Actualmente existen muchos modelos en la literatura, los más relevantes son:

1. Modelo de Erdos-Renyi[21]: es utilizado para generar redes aleatorias. El método consiste en conectar los nodos entre sí con igual probabilidad. La distribución de grado que se genera en estas redes es Poisson.
2. Modelo de Albert-Barabasi[22]: permite generar redes libres de escala, mediante un mecanismo de enlace preferencial, en el cual los nodos que tienen mayor grado son los que tienen mayor probabilidad de recibir nuevas conexiones, generando así una distribución de grado de ley de potencia.
3. Modelo Watts y Strogat[23]: el cual genera redes libres de escala mediante un mecanismo de colocar y quitar enlaces, de tal forma la distancia promedio entre cada par de nodos sea la menor posible.

2.1.2. Glosario

- **Diámetro de una red**[24]: Es el número de aristas de la distancia geodésica más larga entre un par de vértices de una red.
- **Distancia geodésica**[24]: Es el camino más corto entre un vértice y otro a través de una red. Es posible que exista más de un camino geodésico.
- **Distribución de grado**[9]: Es el recuento del número de vértices en un grafo que tienen un grado dado, tomando el número de vértices que tienen un grado dado este puede seguir una distribución de probabilidad.
- **Grado de un vértice**[24]: Es el número de conexiones asociadas a un vértice en un grafo.
- **Grafo**: Un grafo es un conjunto de vértices conectados con aristas
- **Hub**[9]: Vértice en una red libre de escala que tiene un mayor grado con respecto al resto.

- **Sistema complejo**[9]: Un sistema complejo es aquel que está compuesto de muchos componentes que interactúan entre sí. Se encuentran propiedades emergentes que no se pueden explicar sólo tomando en cuenta a sus componentes individuales, en otras palabras el todo no es la simple suma de sus partes.

2.2. Estado del arte de análisis fractal y multifractal de redes complejas

2.2.1. Análisis multifractal

Muchos estudios han demostrado que las redes complejas permiten modelar y caracterizar la dinámica de sistemas complejas presentes en la vida real[9]. Los nodos de una red representan los elementos o agentes y las aristas representan la relación entre estos. Las aristas tienen un significado dentro del sistema y su existencia se debe a una relación o una regla que rige los agentes. Las redes complejas pueden ser estudiadas a partir de sus propiedades estructurales, como lo son la distribución de grado, las medidas de centralidad, su diámetro, la existencia de comunidades dentro de la red, entre otras. Recientemente, las redes complejas han sido estudiadas desde sus propiedades geométricas, una de estas medidas es la fractalidad o autosimilaridad, la cual permite estudiar cómo es la estructura interna de una red compleja[25].

Para la medida de fractalidad, Song y otros[12] han propuesto un algoritmo generalizado de conteo de cajas. Este algoritmo es ampliamente utilizado en espacios geométricos, los cuales son transformados en espacios de acuerdo a la distancia más corta entre cada par de nodos, para así ser aplicado en redes complejas. Ellos encontraron que muchas redes complejas son auto-similares bajo ciertos valores de escala. En un trabajo posterior de los mismos autores[15], se proponen una serie de algoritmos para calcular la dimensión fractal, entre los cuales se tiene, una aproximación al problema de coloreo de grafos y un algoritmo de conteo basado en el solapamiento de las cajas.

Posteriormente, Kim y otros[26][27] proponen un algoritmo para la dimensión fractal en redes libres de escala basada en su esqueleto, el cual es obtenido a partir del árbol generador mínimo. Esta aproximación permite determinar la dimensión fractal más rápidamente que las anteriores estrategias, debido a que no se requiere explorar toda la red. A partir de esta idea, Zhou y otros[28] diseñan un algoritmo basado en el problema de cobertura de aristas, el cual es aplicado en redes complejas biológicas.

Según Lee y Jung[29] la dimensión fractal no es suficiente para describir las propiedades de autosimilaridad en redes complejas, se propone como herramienta el análisis multifractal, el cual además de considerar la dimensión fractal toma en cuenta la escalabilidad. Se encuentra preliminarmente que el análisis multifractal permite describir de mejor forma la distribución del coeficiente de agrupamiento en una red compleja.

Para el análisis multifractal se han propuesto algunos algoritmos para calcular los exponentes de masa[30], los cuales describen los cambios de densidad de la red compleja a medida que esta es escalada. Basado en los algoritmos de conteo de cajas para análisis fractal, se ha introducido una variante denominada algoritmo de conteo de cajas compacto[31]. Esta variante, toma como exponentes de masa el promedio del número de nodos elevado a un factor que representa la escala y

calcula la dimensión fractal a partir de una regresión lineal entre el logaritmo de los exponentes de masa y el logaritmo del tamaño de las cajas, sin embargo, debido al gran número de computaciones que se deben realizar, este algoritmo no es práctico para redes grandes.

Buscando reducir el número de computaciones en el análisis multifractal en redes complejas, Liu y otros[32] proponen una solución para redes complejas basado en el algoritmo propuesto para objetos geométricos de Tel y otros[33]. Esta solución, la cual denominan como caja de arena, busca reducir el número de computaciones al no tener que calcularse el número mínimo de cajas de un radio determinado que se requiere para cubrir la red, sin embargo, se requiere repetir el proceso varias veces para obtener una buena estimación de las dimensiones.

2.2.2. Aplicaciones prácticas

El análisis multifractal se aplica en diferentes áreas del conocimiento, a continuación algunos trabajos recientes por área:

- En biología, el análisis multifractal es aplicado para analizar relaciones en diferentes ámbitos[34], como son las relaciones en ecosistemas, entre proteínas[35], etc. Un trabajo destacado, es el estudio del genoma humano realizado por Moreno y otros[36], en el cual se caracteriza la estructura de los diferentes cromosomas, encontrando que se puede categorizar de acuerdo a baja, media o alta multifractalidad, la cual se ve traducida en términos de estabilidad genética, es decir resistencia a factores ambientales que puedan modificar el ADN. Con estudios de este tipo se puede caracterizar algunas particularidades genéticas.
- En economía, la variación de precios en los mercados se ha estudiado desde diferentes enfoques, pasando por análisis de promedios por ventajas o estadístico, sin embargo, estos métodos presentan problemas debido a que estas son series de tiempo[37]. Un enfoque que ha permitido describir el comportamiento de estas, ha sido el análisis multifractal, un ejemplo de ello son el análisis de la red de pagos de Estonia[38], en este artículo se encuentra que hay un patrón multifractal en esta red de pagos, por lo que toda la economía de este país, presenta un comportamiento multifractal. También, un hallazgo interesante, es que los exponentes de masa tienen un valor muy elevado, lo que indica que la red es bastante irregular. Finalmente, se concluye que este tipo de redes pueden ser estudiadas desde su esqueleto, lo que reduce considerablemente el número de computaciones que se requieren para obtener las dimensiones fractales.
- En la comunicación entre especies, se ha encontrado que el habla y los sonidos de algunos animales presenta patrones multifractales, un caso son los sonidos que emiten las aves migratorias[39]. El análisis multifractal permite concluir que los sonidos muestran comportamientos globales que varían de acuerdo al tono o la longitud del canto.
- El análisis multifractal se utiliza para estudiar la estructura de diferentes redes que modelan el tráfico en redes de telecomunicaciones[40], sistemas de tráfico[41] y redes sociales[42]. En estas redes reales, se ha encontrado patrones monofractales en redes que son de mundo pequeño y multifractales en las que presentan semejanzas con redes libres de escala.

2.2.3. Herramientas para el análisis fractal y multifractal

En la exploración del estado del arte no se evidencia la existencia de una herramienta comercial para usuario final para el análisis multifractal. Lo que se encuentra son implementaciones realizadas por los investigadores de este tema y un complemento para el software Matlab[43]. En la tabla 2.1 se pueden observar algunas de ellas:

Nombre	Descripción	Tipo de acceso	Lenguaje de programación	Funciones que provee
Multifractal.jl[44]	Librería para el análisis multifractal de series de tiempo	Paga ¹	Matlab	Determinar el espectro multifractal y dimensiones fractales.
Multifractal Tool[45]	Herramienta para el análisis multifractal de series de tiempo y de imágenes	Paga ¹	Matlab	Provee análisis multifractal utilizando el espectro calculado con la transformada Wavelet, así mismo provee la reconstrucción multifractal de series de tiempo e imágenes.
Multifractal estimation using a standard box-counting algorithm[46]	Esta librería permite realizar un análisis multifractal de un conjunto de imágenes	Libre	R y C++	Permite calcular los exponentes de masa T_q y dimensiones fractales D_q . Es trabajado en el artículo Multifractal analysis of spatial patterns in ecological communities[47].
Multifractal Analysis ToolBox[48]	Complemento de Simulink para el análisis multifractal de señales (series e imágenes)	Paga	Matlab	Provee la gráfica de dimensión fractal, el cual denominan espectro multifractal.
Multiscale Multifractal Analysis (MMA)[49]	Librería de análisis multifractal del Banco de Datos Physionet	Paga ¹	Matlab	Implementación del método propuesto en el artículo Multiscale multifractal analysis of heart rate variability recordings with a large number of occurrences of arrhythmia[50]

¹La librería es gratuita, pero Matlab[43] es un software pago

Nombre	Descripción	Tipo de acceso	Lenguaje de programación	Funciones que provee
MDFA in Python[51]	Modulo de Python para el análisis multifractal de series de tiempo e imágenes	Libre	Python	Esta es la implementación utilizada en el artículo Multifractal analysis for all. Frontiers in Physiology[52]
Implementation in Python of analysis multifractal[53]	Módulo en Python para análisis fractal de redes complejas	Libre	Python	Son las implementacion de los métodos propuestos en How to calculate the fractal dimension of a complex network- the box covering algorithm[15].

Tabla 2.1: Herramientas para el análisis multifractal encontradas en la literatura

Se puede evidenciar en la exploración realizada, que no se encuentra una herramienta para el análisis multifractal en redes complejas.

2.3. Estado del arte de análisis de robustez de redes complejas

2.3.1. Medida de la robustez

El objetivo del análisis de robustez, es el estudio del efecto en las propiedades estructurales ante la pérdida de nodos y aristas[54]. Este análisis depende de cada sistema que ha sido modelado con redes complejas, una instancia de esto es el efecto de la falla de enrutadores en el Internet.

Una descripción más precisa, este análisis permite observar si la red compleja continua funcionando a pesar de que algunos componentes individuales han fallado o se han degradado. El enfoque consiste en cómo la estructura de la red cambia a medida que esta pierde vértices y aristas. Varios trabajos relevantes abordan en este tema en sistemas , como es el caso de la resiliencia de Internet ante caídas de enlaces o fallas aleatorias[55] y de cómo fortalecer estos sistemas para conservar su funcionamiento en eventos de talla[56].

Mark Newman en su libro Networks: An introduction[57] realiza una profunda revisión en el capítulo 16, el cual trata sobre la percolación y resiliencia en redes complejas. Varias estrategias de estudio son la pérdida aleatoria de vértices y la pérdida de acuerdo al grado. Un enfoque de estudio es cómo se muestran las medidas estructurales dentro del componente gigante, el cual es el componente conexo más grande y a partir de allí se establece un análisis de la robustez de una red.

Para formalizar esta medida, inicialmente se tienen en cuenta que desde la década de 1980, se tienen algunas métricas de conectividad. Algunas de ellas son la conectividad algebraica[58], la superconectividad[59], conectividad condicional[60] y número isoperimétrico[61]. Sin embargo, estas

medidas de robustez basadas en la conectividad sólo consideran la estructura topológica de la red y no tienen en cuenta el nodo o enlace que falla. Para atacar este problema, Albert y otros[54] proponen considerar las propiedades estadísticas como marco de trabajo para el medida de robustez.

Siguiendo la idea de analizar las propiedades estadísticas de las redes complejas, Schneider y otros[14] proponen una métrica de robustez basada en el tamaño del componente gigante, de la siguiente forma:

$$R = \frac{1}{N} \sum_{i=1}^N \delta\left(\frac{i}{N}\right) \quad (2.2)$$

En el cual se tiene un factor de normalización que depende del número de nodos N , y $\delta(\frac{i}{N})$ es la medida de una propiedad estructural en el componente gigante a medida que se pierde un número determinado de nodos. Este índice es una aproximación ampliamente utilizada para la medición de la robustez de una red compleja. Sin embargo, ellos no proveen información en un algoritmo detallado para calcular esta métrica. Para resolver este problema, Li y otros[62] desarrollan un algoritmo que permite realizar este proceso.

2.3.2. Aplicaciones del análisis de robustez

Gran parte de las aplicaciones de análisis de robustez nacen a partir del trabajo de Schneider y otros[14], a continuación se mencionan algunos trabajos relevantes por área:

- En seguridad, el trabajo de Li y otros[63] abarca el problema del estudio de la entropía de la información a la cual se le aplica un patrón de ruido conocido y cómo este afecta su estructura la cual se modela a través de una red compleja. De esta forma, ellos proponen un índice de resistencia que da una idea de la medida de robustez de una red de comunicaciones.
- Para el suministro de energía, Pahwa y otros[64] estudian el estrés que son sometidas las redes de transmisión de energía a medida que aumenta la demanda y las caídas de enlaces. Ellos realizan un estudio con redes reales encontrando una directa relación entre la caída total del sistema y la medida de robustez de la red.
- En salud pública, un tema de gran interés es el estudio de la propagación de epidemias dentro de comunidad. En un estudio realizado por Chami y otros[65] se estudian las estrategias que pueden aplicar las organizaciones encargadas de velar por la salud pública. Algunas de estas estrategias consisten en aislar a los individuos enfermos o fomentar que estas personas eviten contacto con otros, sin embargo, ellos encuentran a partir de un caso de estudio que consiste en una red de amistades de comunidades rurales en Uganda que un enfoque más eficiente para controlar la propagación de epidemias es vacunar o aislar aleatoriamente a ciertos individuos que tienen contacto con muchas personas así no estén enfermos, como es el caso de profesores, funcionarios de gobierno entre otros. Al aplicar este enfoque en la comunidad de estudio, se encuentra que la efectividad de la propagación de enfermedades se reduce, ya que esto produce un mayor efecto en la red de propagación.

2.3.3. Herramientas para el análisis de robustez

Para el análisis de robustez, en la literatura explorada se encuentran las herramientas descritas en la tabla 2.2

Nombre	Descripción	Tipo de acceso	Lenguaje de programación	Funciones que provee
Attack Robustness and Centrality of Complex Networks[66]	Evaluación de redes complejas bajo diferentes estrategias de ataque	Libre	Python	Implementación de las estrategias de análisis de robustez del artículo Attack Robustness and Centrality of Complex Networks[67]. Se provee la medida R con relación al tamaño del componente gigante.
netjson-robustness-analyser[68]	Librería para análisis de robustez de grafos en formato NETJSON	Libre	Python	Esta librería sólo permite visualizar los grafos después de aplicar alguna estrategia de ataque.
Graph Theory Analysis of Brain MRI Data[69].	Librería para el análisis de redes que describen conexiones de neuronas en el cerebro.	Libre	Python	Provee las medidas estructurales de la red e incluye un análisis de robustez con respecto a la distancia promedio entre los vértices
Genonets[70]	Librería para el análisis de redes complejas	Libre	Python	Esta librería permite cargar, obtener medidas y realizar análisis de robustez de redes complejas en formato GML.
Transport Network Criticality Analysis[71]	Librería para el análisis de robustez de redes de transporte	Libre	Python	Esta librería permite realizar un análisis de robustez de redes complejas que se obtienen de un sistema de transporte. Esta librería es utilizada en la tesis de maestría Measuring freight transport network criticality: A case study in Bangladesh[72]

Tabla 2.2: Herramientas para el análisis de robustez encontradas en la literatura

En la exploración realizada se evidencia que hasta el momento no se encuentra evidencia de la existencia de una herramienta que realice una relación entre el análisis multifractal y de robustez.

2.4. Estado del arte en algoritmos de inteligencia artificial en redes complejas

2.4.1. Patrones en redes complejas

En las redes complejas se pueden identificar patrones estructurales, los cuales pueden ser utilizados para su estudio. Li y otros[73] realizan un evaluación de las redes complejas de gran escala utilizando herramientas de minería de datos y reconocimiento de patrones, a través de una conversión de las redes en estructuras de nubes de datos, que facilita el proceso de descubrimiento de patrones en las redes. Como resultado de este trabajo se encuentra que si se proyectan las redes en un espacio multidimensional de parámetros se pueden encontrar patrones.

Otro aporte relevante en fue la conferencia *Workshop on Structural, Syntactic, and Statistical Pattern Recognition*[74], realizada en el año 2010, donde se presentaron trabajos sobre la descripción estructural de redes complejas y el uso de estrategias de inteligencia artificial para su estudio. Un ejemplo de ello son el uso de redes Markovianas para el estudio de redes complejas enmarcados en el estudio de la categorización de material multimedia en el estándar MPEG-7. Otro campo de estudio de esta conferencia fue el estudio del aprendizaje estructural, con el uso de técnicas como los vectores de cuantización (LVQ) para la descripción estructura del grafos, los resultados indican que se pueden utilizar para la descripción de grafos.

Desde el enfoque del estudio de sistemas biológicos, Wuchty y otros[75] realizan un estudio de la aplicación de diferentes técnicas a sistemas complejos que representan interacciones entre proteínas. Se busca integrar las redes complejas, el análisis de imágenes, el procesamiento de señales y la inteligencia artificial, mediante el estudio de aplicación de estas técnicas a varios casos de estudio, uno de ellos, consiste en estudiar esencialidad de las proteínas, que consiste en construir una red compleja a partir de sus interacciones. El análisis de este caso indica que existen propiedades como la centralidad de nodos y propiedades de k -core en la red, las cuales permiten describir la topología de la red.

2.4.2. Algoritmos de inteligencia artificial aplicados en redes complejas

Los trabajos donde se han utilizado técnicas de inteligencia artificial para la medición de parámetros en redes complejas están basados principalmente en estrategias para optimizar algoritmos ya existentes, como es el caso del trabajo de Zhou y otros[28], en donde se explica la autosimilaridad en redes complejas celulares mediante un método de cobertura de aristas utilizando algoritmos de recocido simulado aplicado dentro del algoritmo cobertura de cajas. Esta técnica consiste en encontrar un número mínimo de N cajas de tamaño l que pueda cubrir las aristas de la red. Las pruebas se realizaron sobre 43 redes celulares, encontrando que estas redes presentan una distribución de ley de potencia, donde la dimensión fractal es $[2, 67 \pm 0, 15]$. Sin embargo, este resultado no puede concluir que exista una distribución de grado de ley de potencia en todas las redes celulares.

Para los algoritmos evolutivos, un elemento de la literatura clave es libro *Multiobjective Evolutionary Algorithms on Complex Networks* de Michell Kirley y Robert Stewart[76], en donde se realiza una introducción del uso de algoritmos evolutivos multiobjetivo en redes complejas. Este libro ofrece diferentes acercamientos teóricos para permitir mapear los nodos de una red dentro de los individuos de una población, crear reglas de selección y evolución de tal manera se pueda modelar una red compleja y así aplicar algoritmos para la medición de parámetros en ellas.

Con respecto a la aplicación de algoritmos de inteligencia artificial, se destaca el trabajo Chien[77], en el cual se utilizan estrategias de planificación basadas en algoritmos inteligencia artificial sobre redes que representan las interacciones entre diferentes componentes de software. Para esto, se utiliza algoritmo conocido como *Multimission VICAR Planner*(MVP), el cual se utiliza ampliamente en el procesamiento digital de imágenes. Su funcionamiento consiste en que a partir de un conjunto de imágenes de interés y la especificación de un estado deseado, realizar un conjunto de pasos para alcanzar dicho estado; en este algoritmo se construye un grafo que representa la estructura de almacenamiento de las imágenes. La aplicación MVP al problema de componentes de software permite estructurarlos considerando sus dependencias y categorías. Los resultados muestran una mejoría en el proceso de configuración y reconfiguración de estos módulos. Otro trabajo relevante, es el realizado por Jian Liua y Tingzhan Liub[78], dónde se utilizan una combinación entre el algoritmo de recocido simulado y el algoritmo K-Means para encontrar comunidades dentro de redes complejas. Las comunidades se pueden ver cómo grupos de nodos que comparten características similares representadas por el peso de las conexiones dentro de la red. En este trabajo, se encuentra que el uso de estas técnicas combinadas ayudan a encontrar más rápidamente las comunidades dentro de la red debido a que mejora la elección del centro de cada clúster en el procesos del algoritmo K-Means, debido al proceso de calentamiento o enfriamiento que provee el algoritmo de recocido simulado, y además, se pueden encontrar comunidades sin necesidad de conocimiento previo acerca de la estructura de las comunidades.

En el diseño heurísticas para aplicar algoritmos de inteligencia artificial en redes complejas, en la literatura se encuentran varios trabajos relevantes, uno de ellos es realizado por Chen y otros[79], el cual se propone una heurística para la detección de comunidades en redes complejas utilizando como base la existencia de relaciones cercanas entre los nodos de una misma comunidad, para calcular el centro de cada comunidad con base a las relaciones entre nodos y no directamente al promedio de los parámetros de cada nodo. Los resultados de este artículo demuestran a-priori que el uso de esta estrategia sobre cuatro redes de prueba, es más eficiente para la detección de comunidades en redes complejas que otras estrategias como es el caso del algoritmo K-Means, debido a que los centros de cada comunidad convergen más rápidamente.

2.5. Resumen y conclusiones del capítulo

En este capítulo se ha dado un marco referencial para entender el concepto de red compleja y así poder abordar el análisis de multifractalidad y robustez. También, se ha mostrado cómo se encuentra el estado del arte en estos temas. A continuación, algunos comentarios:

- El análisis de redes complejas se realiza con las herramientas que provee la teoría de grafos. Sus propiedades estructurales y la clasificación está basada con las que comúnmente se trabajan con grafos.
- En el estado del arte, el análisis fractal y multifractal es basado en métodos geométricos de cobertura de cajas, en los cuales se realiza un mapeo espacial de un sistema euclidiano a uno basado en la distancia más corta entre cada par de nodos.
- Así mismo, el análisis de robustez es el estudio de cómo cambian las propiedades estructurales de la red a medida que esta pierde nodos o conexiones, el significado de estos valores depende directamente de lo que se está modelando con la red.
- Finalmente, en la literatura se encuentra que la relación entre la medida de fractalidad y robustez no ha sido ampliamente explorada, por lo que realizar un estudio en esta materia permite generar un aporte teórico para la generación de nuevas métricas y algoritmos.
- Así mismo, en la literatura no se evidencia la existencia de una herramienta que realice un análisis de multifractalidad y robustez en redes complejas.

Capítulo 3

Generación de redes fractales

3.1. Redes fractales

Los fractales en naturaleza son estructuras que mantienen su forma a diferentes escalas. En el caso de las redes, existe una familia conocida como (x, y) -flowers[80], la cuales son conocidas como redes libres de escala jerárquicas. Rozenfeld y otros[81] han demostrado analíticamente que estas redes conservan sus propiedades analizando el escalado de los hubs y los nodos.

Un método de construcción iterativo de estas redes es propuesto por Lin y otros[2] en un estudio de árboles de cobertura mínima en redes complejas autosimilares.

Este método, consiste en generar una función $F(x, y)$ donde $x > 0$ y $y > 0$ que permite construir la flor dadas n generaciones. Para $n = 0$ la red que genera $F_0(x, y)$ siempre tiene dos vértices unidos por una arista. A medida que se itera $F_n(x, y)$ se deriva al reemplazar cada arista existente en F_{n-1} por caminos de largo x e y . El número de aristas del algoritmo generador siempre es $|E| = (x + y)^n$.

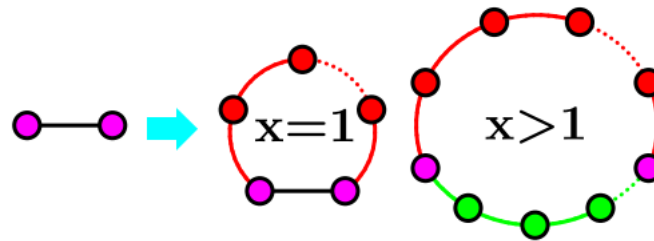


Figura 3.1: Proceso iterativo de construcción de las $(x - y)$ -flowers. Tomado de [2]

En la figura 3.1 el proceso iterativo consiste en reemplazar cada arista con un camino de tamaño $x, x \leq 1$ y $y, y \leq x \wedge y > 1$. Para $x = 1$ un par de nodos de la iteración anterior es directamente conectado con $y - 1$ nuevos nodos. Todos estos nuevos nodos y dos de los nodos de la iteración anterior forma un camino rojo de tamaño y . Para $x > 1$ cada arista antigua es reemplazada por dos caminos que consisten en $y - 1$ nodos.

Otra forma de construir las redes es ir formando la red con copias de ella misma. En la figura 3.2 se observa este proceso.

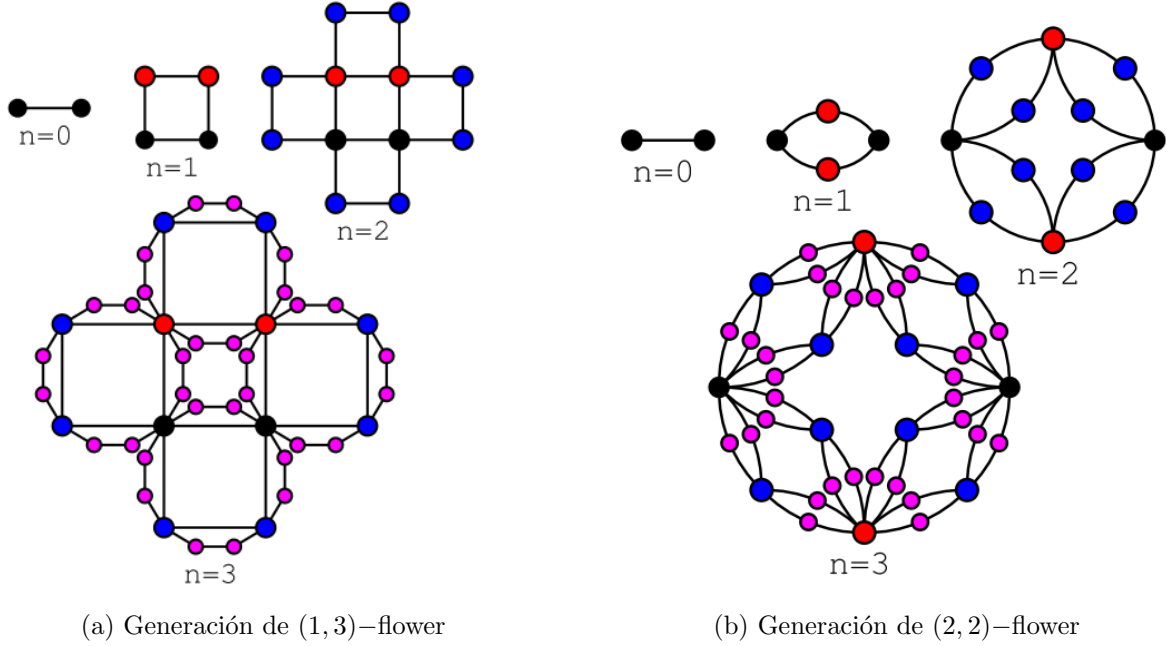


Figura 3.2: Proceso iterativo de generación de redes (x,y) -flowers. Tomado de [2]

3.2. Algoritmo para generar redes fractales

Para la generación de redes fractales se utiliza la matriz de adyacencia para realizar este proceso. La matriz de adyacencia M de un grafo o red $G(V, E)$ no dirigido es de tamaño $|V| \times |V|$. Para cada $(i, j) \in V^2$ se tiene que $M_{i,j} = 1$ si existe una arista entre i y j , en caso contrario $M_{i,j} = 0$. Dado que la matriz de adyacencia para grafos no dirigidos tiene simetría triangular superior, sólo se requiere recorrer las filas i y columnas j que cumplan $i < j$.

3.2.1. Red (2,2)-flower

Para la generación 0, se toma como caso base la siguiente matriz de adyacencia:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Figura 3.3: Matriz de adyacencia para la generación 0 de (2,2)-flower

Para cada generación, se recorre la matriz de tal forma que cuando se encuentra un 1 en una posición (i, j) cualesquiera, esta es cambiada por 0, es decir que se elimina la arista, es de anotar, que también se hace lo mismo con (j, i) dado que simétrica. Después se agregan dos filas llenas de ceros k y $k + 1$. Dado que la matriz es simétrica agregar una fila implica agregar una columna. En las siguientes posiciones se coloca 1:

- En las posiciones (i, k) , (k, i) , $(i, k + 1)$, $(k + 1, i)$, (j, k) , $(j, k + 1)$, $(j, k + 1)$ y $(j, k + 1)$,

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

(a) Generación 0

(b) Generación 1

Figura 3.4: Proceso para construir la generación 1 de (2,2)-flower. Observe que las aristas eliminadas son resaltadas en negrilla

En la figura 3.4 se muestra que cada arista en la generación anterior es eliminada.

En la tabla 3.1 se puede observar algunas propiedades estructurales de las primeras 4 generaciones de (2,2)-flower.

Generación	Número de vértices	Número de aristas	Grado promedio	Diámetro de la red	Longitud media de caminos
0	2	1	1	1	1
1	4	4	2	2	1.33
2	12	16	2.67	4	2.3
3	44	64	2.91	8	4.35

Tabla 3.1: Propiedades estructurales de generaciones 0,1,2 y 3 de la red fractal (2,2)-flower

3.2.2. Red (1,3)-flower

Para la generación 0, se toma como caso base la siguiente matriz de adyacencia:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Figura 3.5: Matriz de adyacencia para la generación 0 de (1,3)-flower

Para cada generación, se recorre la matriz de tal forma cuando se encuentra un 1 en una posición

(i, j) cualesquiera se deben crear dos vértices, los cuales se conectan entre sí, uno de los vértices se conecta con el vértice i y el otro con el vértice j .

Como se deben agregar dos vértices, se crean filas llenas de ceros k y $k + 1$. Dado que la matriz es cuadrada agregar una fila implica agregar una columna. Para crear el camino de longitud 1, en las siguientes posiciones se coloca 1:

- Para generar un camino de longitud 2, para las posiciones $(k, k + 1)$ y $(k + 1, k)$
- En las posiciones (i, k) , (k, i) , $(j, k + 1)$ y $(j, k + 1)$, para conectar los nodos antiguos con un camino de longitud 2 $(y - 1)$.

$$\begin{pmatrix} 0 & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & \mathbf{1} & 1 & 0 \\ \mathbf{1} & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

(a) Generación 0

(b) Generación 1

Figura 3.6: Proceso para construir la generación 1 de (1,3), flower. Observe que las aristas que se conservan son resaltadas en negrilla

El proceso mostrado en la figura 3.6 para las (1,3)-flowers, para cada arista se generan dos vértices conectados por una arista. Las dos aristas se conectan por sus extremos. A continuación se describen las propiedades estructurales de las primeras 4 generaciones de la (1,3)-flower

Generación	Número de vértices	Número de aristas	Grado promedio	Diámetro de la red	Longitud media de caminos
0	2	1	1	1	1
1	4	4	1	2	1.2
2	12	16	2.67	4	2.3
3	44	64	2.91	6	3.11

Tabla 3.2: Propiedades estructurales de generaciones 0,1,2 y 3 de la red fractal (1,3)-flower

3.3. Resumen y conclusiones del capítulo

Las redes fractales se pueden generar utilizando algoritmos y estas tienen propiedades de autosimilaridad a diferentes escalas.

El algoritmo propuesto se basa en las matrices de adyacencia, ya que esto permite representar las aristas directamente. El proceso consiste en encontrar una arista entre un par de vértices y aplicar alguna heurística de construcción, que puede implicar conservar o eliminar la arista, crear nuevos nodos y aristas bajo ciertas reglas de conexión con sus nodos.

Es importante analizar la complejidad computacional de generar una (x,y)-flower de cualquier generación n . La idea consiste en buscar dentro de la matriz de adyacencia una arista de la generación $n - 1$, generar 2 nodos por cada arista e interconectarlos con los nodos de esta arista. Esto implica recorrer la matriz de adyacencia de $n - 1$, lo que implica una complejidad cuadrática.

Es importante aclarar que sólo se estudiaron 2 casos en este capítulo del número de valores infinitos que pueden tomar x e y . En el caso de las redes estudiadas se encuentra que el número de vértices de la generación n consiste en sumar aproximadamente $n + 1$ veces el número de vértices de la generación anterior, por lo que, se plantea la siguiente ecuación de recurrencia:

$$T(n) = (n + 1)T(n - 1), T(0) = 2, n \geq 0 \quad (3.1)$$

La solución exacta de esta ecuación es $\Gamma(n + 1)$ ¹, debido a que $n \geq 0$, su solución es del orden $O(n!)$. Lo que indica que en una generación n hay $O(n!)$ vértices y el costo de recorrer la matriz de adyacencia será el cuadrado de este valor. Por lo tanto, se concluye que la complejidad computacional y espacial para generar una red (x,y)-flower es de $O(n!)$.

Debido al elevado costo computacional de memoria que se requiere para generar estas redes, se decide en las pruebas que se realizan en este proyecto trabajar con la séptima generación de (x,y)-flower.

¹Función Gamma

Capítulo 4

Análisis multifractal

4.1. Introducción

Desde el trabajo pionero de Song[12] se ha considerado que las redes complejas consisten en patrones auto-similares bajo transformaciones de escala. Se han proporcionado diferentes estrategias basadas en la cobertura de cajas para caracterizar la dimensión fractal en redes complejas. Sin embargo, la dimensión fractal no es suficiente para caracterizar las propiedades fractales de las redes complejas, por esta razón para el análisis multifractal se propone calcular la dimensión fractal a medida que se realiza un escalado de la red.

El análisis multifractal es útil para caracterizar la heterogeneidad espacial de las redes complejas, ya que considera que no sólo existe un patrón de estructuras dentro de las redes complejas.

Las diferentes estrategias permiten calcular los exponentes de masa τ_q , que representan cómo cambia la densidad de la red a medida que se escala por un factor real q y la pendiente de esta recta para un q dado viene siendo la dimensión fractal D_q , de esta forma se puede estudiar los cambios en la estructura de la red a medida que esta es escalada.

Si la función D_q es una línea recta, se dice que la red es monofractal, en caso contrario, es multifractal. Esta función tiene algunos valores clave[82]: D_0 es la dimensión fractal, D_1 es la dimensión informacional, que indica cómo se escala la información promedio que se necesita para identificar una caja ocupada y D_2 es la correlación, la cual es una medida que indica cuántos nodos se encuentran dentro de una estructura fractal.

4.2. Estrategias de cubrimiento de cajas

Este método es una herramienta para estimar la dimensión fractal de objetos en un espacio Euclidiano. Sin embargo, la métrica no es relevante para redes complejas, por esta razón se considera la longitud del camino más corto entre un par de nodos como métrica.

Para una red cualquiera, se toma N_B como el número de cajas de radio r_B que se necesitan para cubrir toda la red, por lo tanto la dimensión fractal d_B se caracteriza de la siguiente forma:

$$N_B \approx r_B^{d_B} \quad (4.1)$$

El algoritmo más común para obtener esta medida es:

1. Seleccione un nodo aleatoriamente como el centro de una caja
2. Busque los nodos a un radio r_B y que no han sido cubiertos por otra caja y asíguelos a la caja
3. Repita los pasos 1 y 2 hasta asignar los nodos a sus respectivas cajas

La dimensión fractal con la pendiente de la regresión lineal entre $\log(N_B)$ y $\log(d_B)$

De este algoritmo se tienen varias variantes que dependen de como se seleccionan los centros:

1. Selección aleatoria[27]: Seleccionando los centros de tal forma se obtenga el mínimo número de cajas.
2. Estrategia voraz[15]: Seleccionando los centros de acuerdo a un problema análogo de coloreo de grados
3. Estrategia quemada[15]: Consiste en permitir solapamientos entre las cajas y cubrir la red con estas.

Los autores de estas estrategias mediante pruebas en algunas redes, han encontrado que se obtiene una medida de dimensión fractal similar a la calculada analíticamente.

Box Covering Fixed Size (BCFS)

Uno de los algoritmos que se se ha propuesto, es el de *Box Covering Fixed Size*[82][38][83], en el cual se realiza una adaptación del problema de la partición[84] a redes complejas. Para esto se utiliza la siguiente ecuación:

$$Z_e(q) = \sum_{\mu(B) \neq 0} \mu(B)^q \quad (4.2)$$

Donde $q \in \mathbb{R}$ y la suma representa todas las diferentes cajas B no vacías, para un tamaño e definido y que cubren toda la red. Los exponentes de masa $\tau(q)$ están definidos por:

$$\tau(q) = \lim_{x \rightarrow \infty} \frac{\ln Z_e(q)}{\ln e} \quad (4.3)$$

Las dimensiones fractales generalizadas están definidas por:

$$D_q = \frac{\tau(q)}{q-1}, q \neq 1 \quad (4.4)$$

Y

$$D_1 = \lim_{x \rightarrow 0} \frac{Z_{1,e}}{\ln e} \quad (4.5)$$

Donde $Z_{1,e} = \sum_{\mu(B) \neq 0} \mu(B) \ln \mu(B)$

Los exponentes de masa se pueden obtener a través de una regresión lineal entre $\ln(Z_e(q))$ y $\ln e$, así mismo las dimensiones fractales

Debido a que la elección de los centros de las cajas es aleatorio, la medida puede ser imprecisa. Si se seleccionan los vértices con mayor grado(hubs) se puede cubrir la red de forma eficiente, pero si son seleccionados nodos con menor grado, pocos nodos pueden ser cubiertos. Para enfrentar este problema se propone seleccionar un conjunto T de permutaciones distintas de nodos y tomar un promedio. La sección de T, debe estar acorde al número de nodos.

El procedimiento para calcular las dimensiones fractales es:

1. Se generan matrices de adyacencia y distancias
2. En primer lugar, asegurarse que todos los nodos en la red no estén cubiertos por alguna caja
3. Se generan T permutaciones de vértices (con todos los existentes en la red), los cuales serán seleccionado como centros de cajas en el orden de aparición
4. Se selecciona el tamaño de una caja $r \in [1, d]$ donde d es el diámetro de la red.
5. Se selecciona el centro de la caja y se incluyen todos los nodos que están dentro de una distancia r y no han sido cubiertos
6. Se repite este proceso hasta que todos los vértices han sido cubiertos
7. Se toma como medida $\mu(B) = \frac{N_B}{N}$ donde N_B es el número de vértices en la red y N es el número total de vértices.
8. Se calcula la suma $Z_r(q) = \sum_{\mu(B) \neq 0} \mu(B)^q$
9. Se repiten los pasos 3-8 con cada uno de los conjuntos T. Se toman los promedios de $\overline{Z_r(q)} = \sum^t (Z_r(q)) / T$

Posteriormente se realiza una regresión lineal entre $\ln \overline{Z_r(q)}$ y $\ln \frac{r}{d}$. Se debe considera el caso especial cuando $q = 1$.

Box Covering Compact (BCC)

Este algoritmo es propuesto por Li y otros[13] como una variación del BCFS. Consiste en buscar para cada tamaño posible de caja, el mínimo número de ellas posible para cubrir la red.

El procedimiento para obtener la dimensión fractal es:

1. Se generan las matrices de distancias y de adyacencia
2. Inicialmente todos los nodos de la red no están cubiertos por ninguna caja
3. Al igual que BCFS se generan T conjuntos de permutaciones de los nodos
4. Se selecciona el tamaño de la caja entre 1 y d , donde d es el diámetro de la red
5. Se selecciona un centro de acuerdo al conjunto T
6. Se cubren los nodos desde 1 hasta el radio seleccionado, si llegado un momento no se pueden agregar nodos, la caja es descartada. Esto garantiza que las cajas siempre tengan tamaño n
7. Estos pasos 3-5 se repiten hasta que la red es cubierta. De estos pasos para cada radio conserve la configuración que tiene el menor número de cajas.
8. Obtenga $Z_r(q)$ de la misma forma que en FSB
9. Realice el mismo proceso de regresión que en BCFS para obtener los exponentes de masa y la dimensión fractal

Los autores aseguran que este algoritmo permite estimar mejor la dimensión fractal D_0 , ya que al tener la configuración con el menor número de cajas posible, es la mejor forma de aproximar a los algoritmos de cálculo de dimensión fractal.

4.3. Algoritmo SandBox

Liu y otros[32] han realizado un estudio de los diferentes algoritmos de Boxcounting encontrando que estos requieren un gran tiempo de cómputo, ya que se realiza la búsqueda de una configuración óptima de centros para obtener una estructura de cubrimiento, en la cual cada caja tiene el mismo número de nodos o se obtiene el número mínimo de cajas necesario para el cubrimiento de la red.

Su propuesta conocida como algoritmo de caja de arena o SandBox, consiste en escoger un número de centros de cajas aleatorios y permitir que las cajas se solapen. Dado que se permite solapamientos, como medida se utilizará el promedio del número de nodos en cada caja y con este valor se realiza la medición.

El proceso de este algoritmo es el siguiente:

- Inicialmente, seleccione un porcentaje nodos como centros de las cajas.
- Seleccione el radio $r \in [1, d]$ de cada caja de arena que será usado para cubrir la red.
- Cuente el número de nodos en cada caja de arena de radio r , denote este número como $M(r)$.
- Para todos los centros elegidos calcule el promedio estadístico de $\overline{M(r)}(q-1)$

La regresión lineal se realiza entre $\ln(\overline{M(r)}(q-1))$ y $\ln(\frac{r}{d})$

Debido a que éste método elige centros aleatorios, se recomienda repetirlo varias veces para mejorar la precisión.

4.4. Análisis de los algoritmos

Para analizar los 3 algoritmos estudiados Box Counting Fixed Size (BCFS), Box Covering Compact (BCC) y SandBox(SB); se seleccionan algunas redes de prueba, las cuales pueden ser consultadas en el Anexo A de la página 86,

La configuración y protocolo de pruebas es explicado en el Anexo B de la página 88

Para este capítulo se han seleccionado las siguientes pruebas:

1. Red A: Red libre de escala generada 4000 nodos.
2. Red B: Red de mundo pequeño generada con probabilidad de reconexión de 10 %.
3. Red C: Red aleatoria generada con 5620
4. Red D: Red observada en la bacteria Ecoli.
5. Red E: Red fractal generada (2,2)-flower.

4.4.1. Redes libres de escala

Inicialmente, para cada estrategia se analiza la relación entre q y $\ln(Zr(q))$

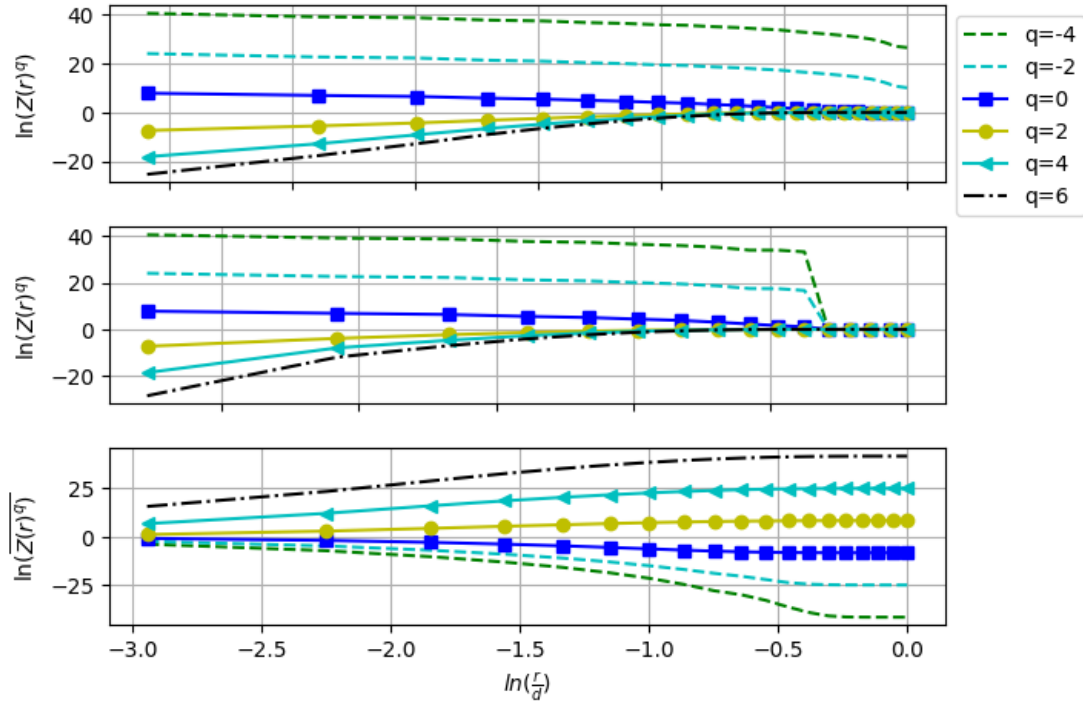


Figura 4.1: Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB

Cada valor de q tiene su función de la cual se obtiene su pendiente, la cual corresponde a sus exponentes de masa.

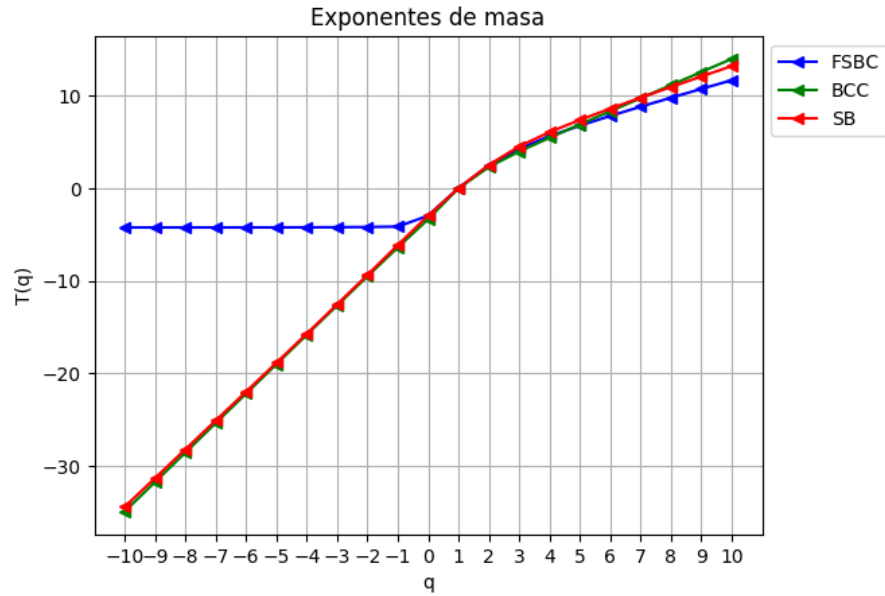


Figura 4.2: Exponentes de masa para red libre de escala 4000 nodos

Se observa que para el método BCFS los $q < 0$ no tienen significado, esto se debe a que en cada caja se toma la cantidad de nodos sobre el total, lo que da valores menores que 1, al elevarlos a un valor q se transforman en valores muy pequeños. Por esta razón, la dimensión se analiza para $q \geq 0$

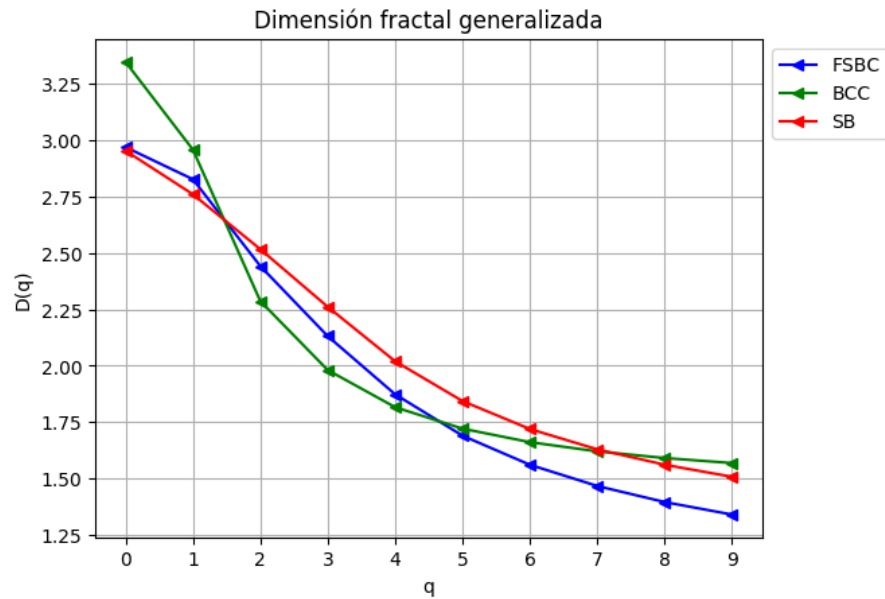


Figura 4.3: Dimensión fractal generalizada para red libre de escala 4000 nodos

De esta se puede obtener la siguiente información y compararla con el artículo de Wang[3]:

Dato	BCFS	BCC	SB	Artículo
Dimensión fractal	2.97	3.34	2.95	3.19
Dimensión fractal generalizada	2.82	2.95	2.75	3.15
Dimensión de correlación	2.44	2.28	2.51	2.81
Dimensión máxima	2.97	3.34	3.14	3.26
Dimensión mínima	1.29	1.55	1.46	1.75
Variación en la dimensión	1.68	1.79	1.68	1.51

Tabla 4.1: Comparación de los datos obtenidos con los del artículo de Wang[3]

Las diferencias en la medida de la dimensión fractal se deben a que el método es una aproximación y debe ejecutarse muchas veces, además que las redes del artículo y la generada son distintas.

Para evitar estas diferencias Li[13] recomienda que para el caso de $q = 0$ se tome directamente la dimensión fractal obtenida por algoritmo de Box Counting, que en este caso es 3.13.

En las gráficas se puede observar que las redes libres de escala tienden a ser multifractales, algo que es esperado ya que internamente estas redes tienen estructuras diferentes debido a la presencia de hubs altamente conectados.

4.4.2. Redes de mundo pequeño

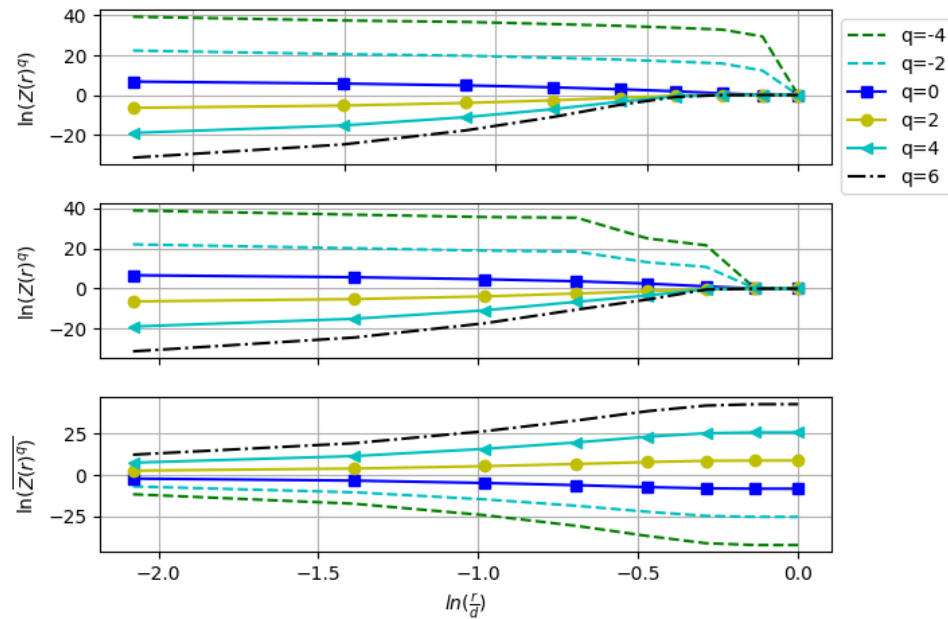


Figura 4.4: Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB

A continuación, los exponentes de masa:

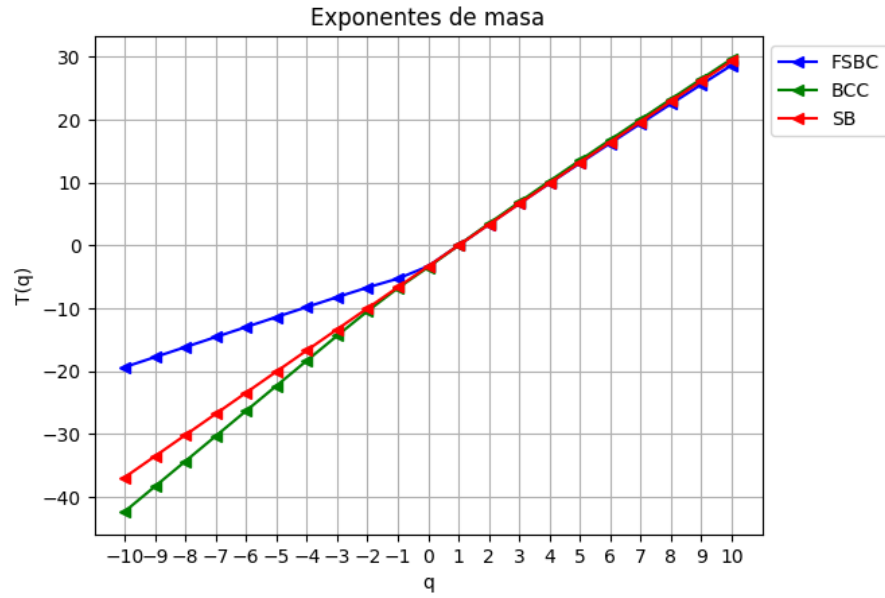


Figura 4.5: Exponentes de masa para red de mundo pequeño con 5000 nodos y $p = 10\%$

Finalmente, la dimensión fractal generalizada

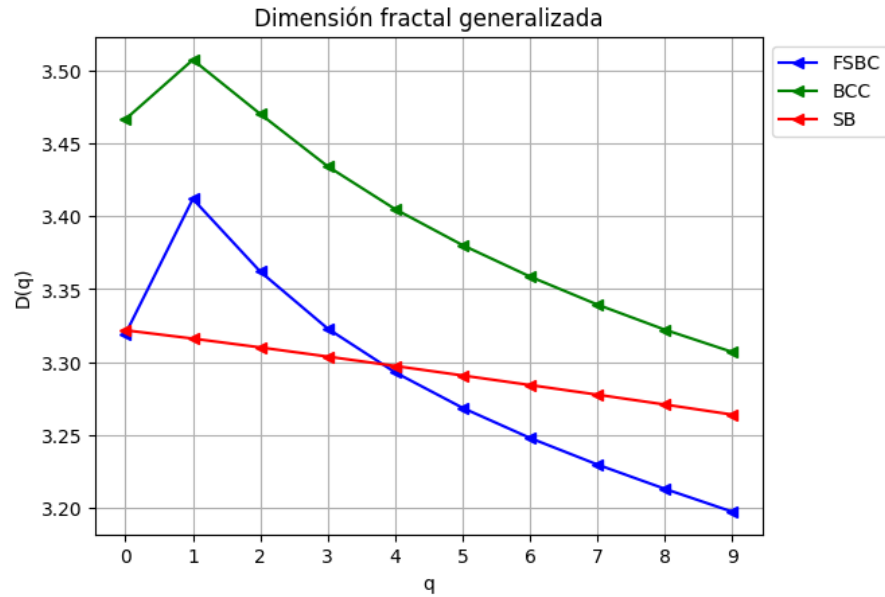


Figura 4.6: Dimensión fractal generalizada para red de mundo pequeño con 5000 nodos y $p = 10\%$

De acuerdo a los datos del artículo se encuentra:

Dato	BCFS	BCC	SB	Artículo
Dimensión fractal	3.31	3.46	3.32	2.79
Dimensión fractal generalizada	3.41	3.5	3.31	2.8
Dimensión de correlación	3.36	3.47	3.31	2.81
Dimensión máxima	3.41	3.84	3.35	2.81
Dimensión mínima	3.18	3.29	3.25	2.75
Variación en la dimensión	0.23	0.55	0.1	0.06

Tabla 4.2: Comparación de los datos obtenidos con los del artículo de Wang[3]

Los resultados varían ya que las redes del artículo y las utilizadas en estas pruebas son generadas, es decir que no son las mismas.

Se observa que la variación en la dimensión fractal es pequeña, de hecho menor que 1, por lo que, se puede concluir que las redes de mundo pequeño tienden a ser monofractales, es decir que se pueden considerar como una sola estructura. Es importante indicar que las mediciones de los diferentes algoritmos requieren un mayor número de iteraciones para mejorar su precisión.

La dimensión fractal de esta red es 3.47

4.4.3. Redes aleatorias

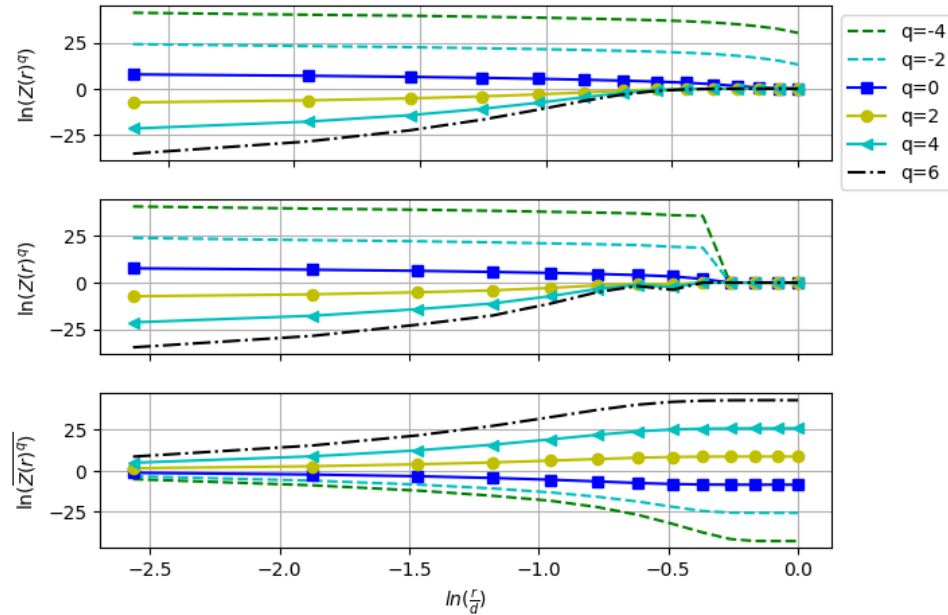


Figura 4.7: Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB

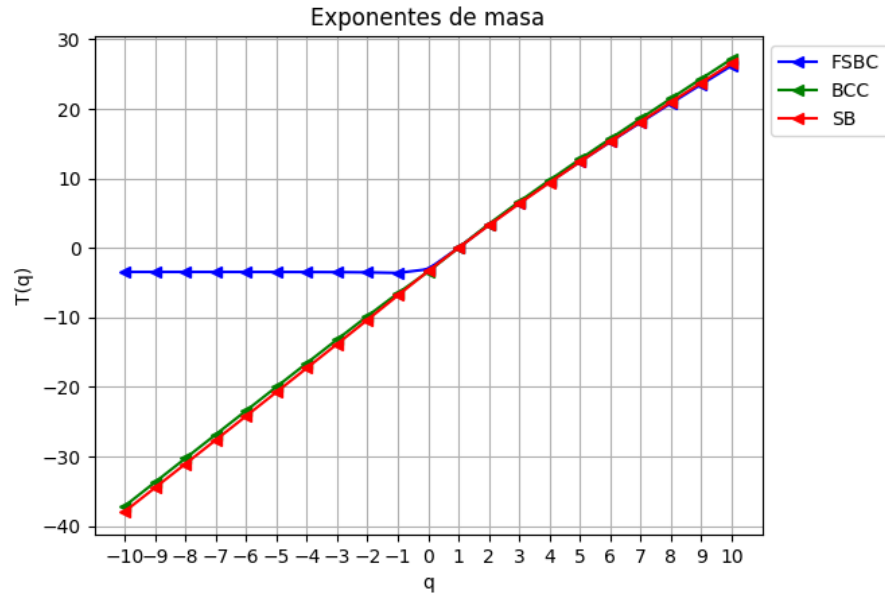


Figura 4.8: Exponentes de masa para red aleatoria de 5620 nodos

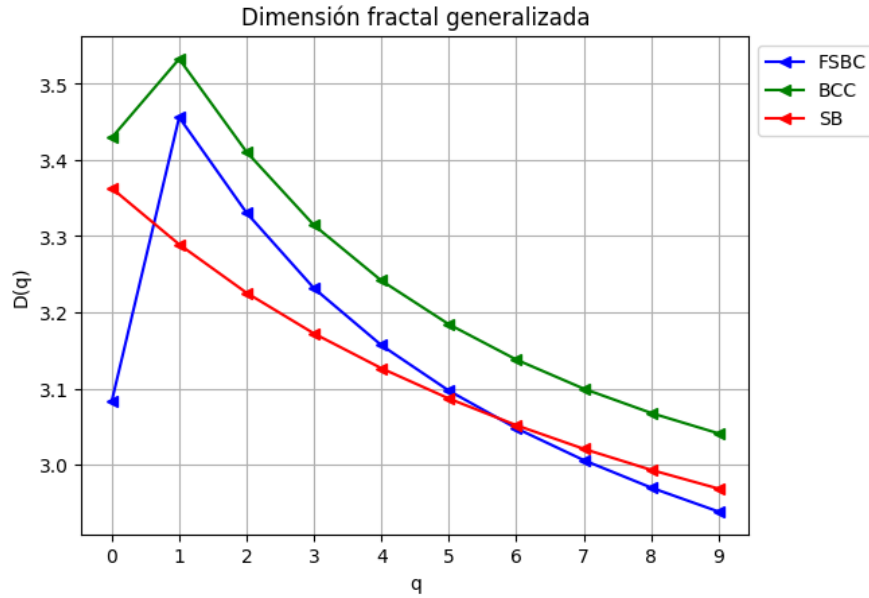


Figura 4.9: Dimensión fractal generalizada para para red aleatoria de 5620 nodos

La dimensión fractal de esta red es: 3.41

Los análisis realizados a las redes de prueba, permite concluir que el análisis multifractal no permite caracterizar redes aleatorias, ya que pueden tender a ser monofractal o multifractal. En este caso la red tiende a ser monofractal.

Dato	BCFS	BCC	SB	Artículo
Dimensión fractal	3.08	3.42	3.36	3.54
Dimensión fractal generalizada	3.46	3.53	3.28	3.56
Dimensión de correlación	3.33	3.41	3.22	3.52
Dimensión máxima	3.45	3.53	3.46	3.54
Dimensión mínima	3.08	3.01	2.94	3.61
Variación en la dimensión	0.37	0.52	0.52	0.33

Tabla 4.3: Comparación de los datos obtenidos con los del artículo de Wang[3]

4.4.4. Red observada de Ecoli

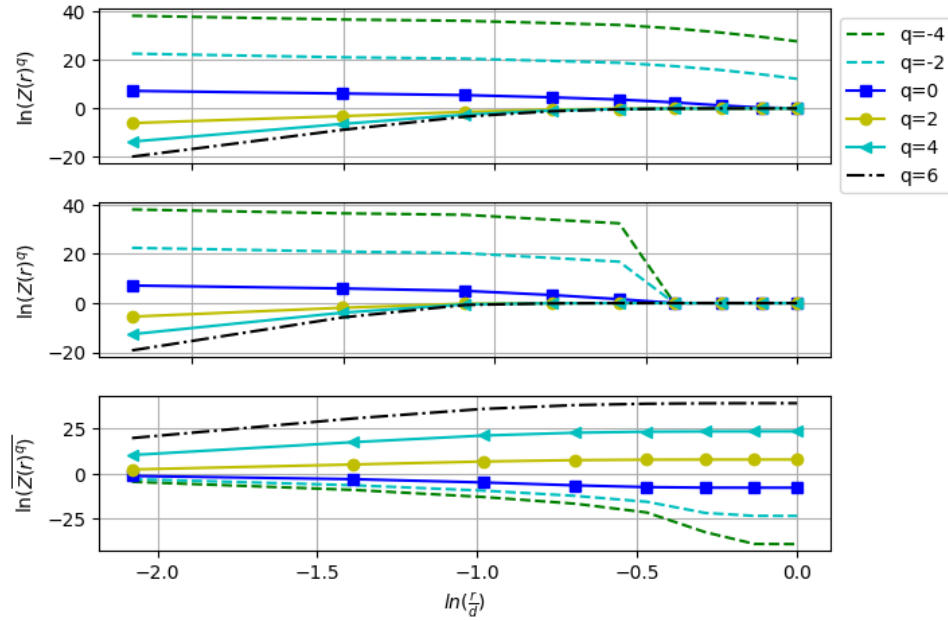


Figura 4.10: Regresión lineal entre $\ln(Z_r(q))$ y $\ln(\frac{r}{b})$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB

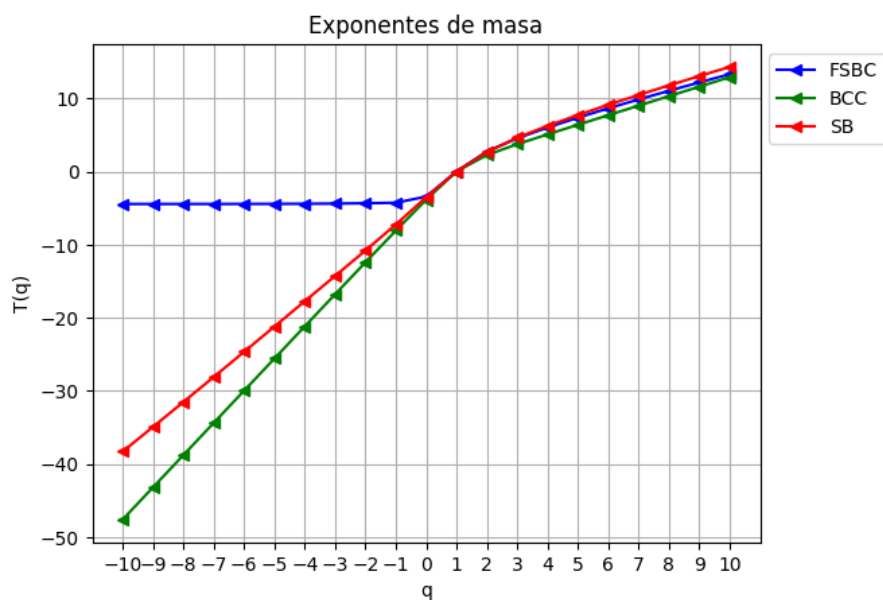


Figura 4.11: Exponentes de masa para red observada ecoli

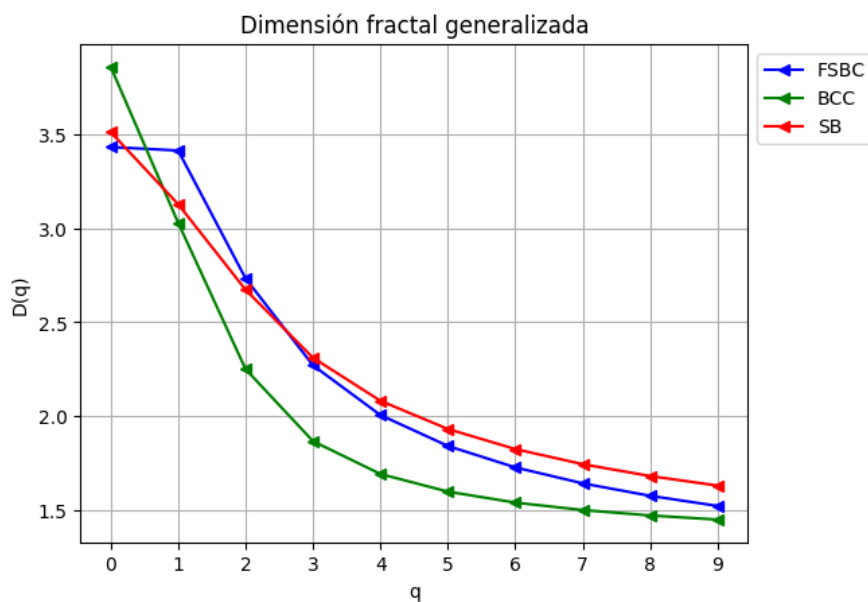


Figura 4.12: Dimensión fractal generalizada para observada ecoli

La dimensión fractal de esta red es 3.31

Dato	BCFS	BCC	SB	Artículo
Dimensión fractal	3.43	3.35	3.51	3.56
Dimensión fractal generalizada	3.41	3.03	3.12	3.45
Dimensión de correlación	2.73	2.25	2.67	2.9
Dimensión máxima	3.41	4.32	3.6	4.15
Dimensión mínima	1.47	1.43	1.5	2.1
Variación en la dimensión	1.94	2.91	2.1	2.05

Tabla 4.4: Comparación de los datos obtenidos con los del artículo de Wang[3]

4.4.5. Redes fractales

Utilizando el artículo de Liu y otros[32] se trabaja con la (2,2)-flower.

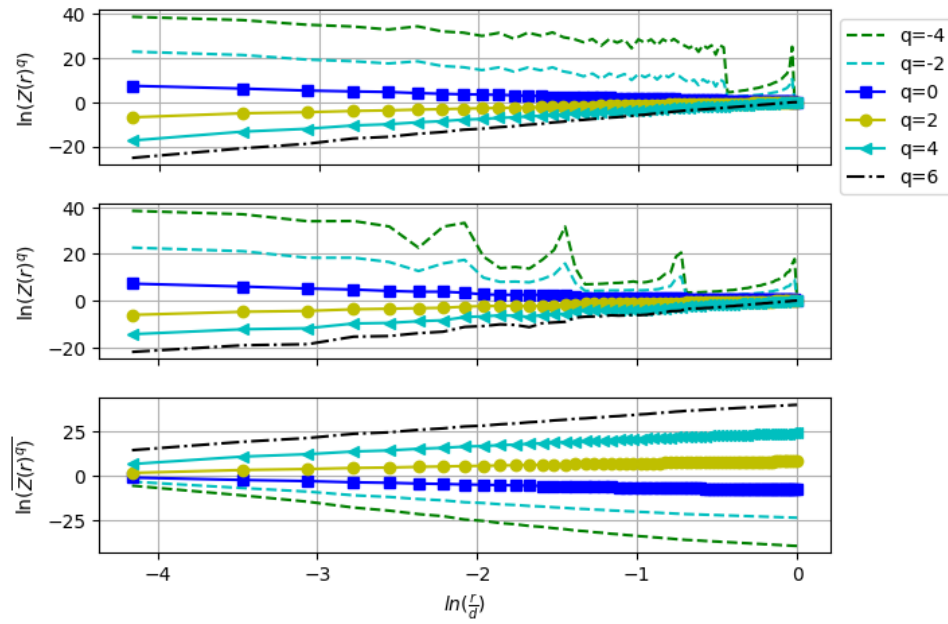


Figura 4.13: Regresión lineal entre $\ln(Z_r(q))$ y $\ln(r/b)$. De arriba hacia abajo, la primera figura corresponde al método BCFS, la segunda al método BCC y la tercera SB

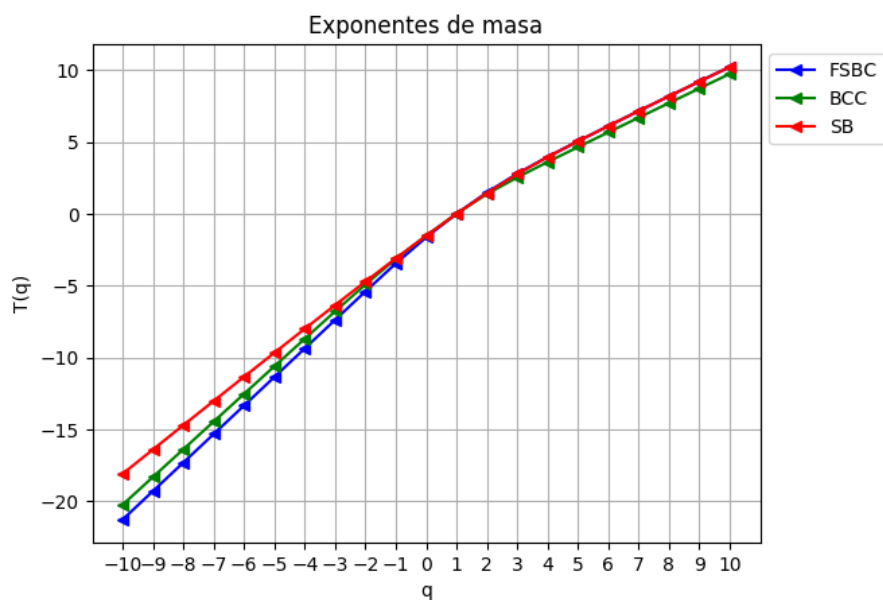


Figura 4.14: Exponentes de masa para red fractal (2,2)-flower

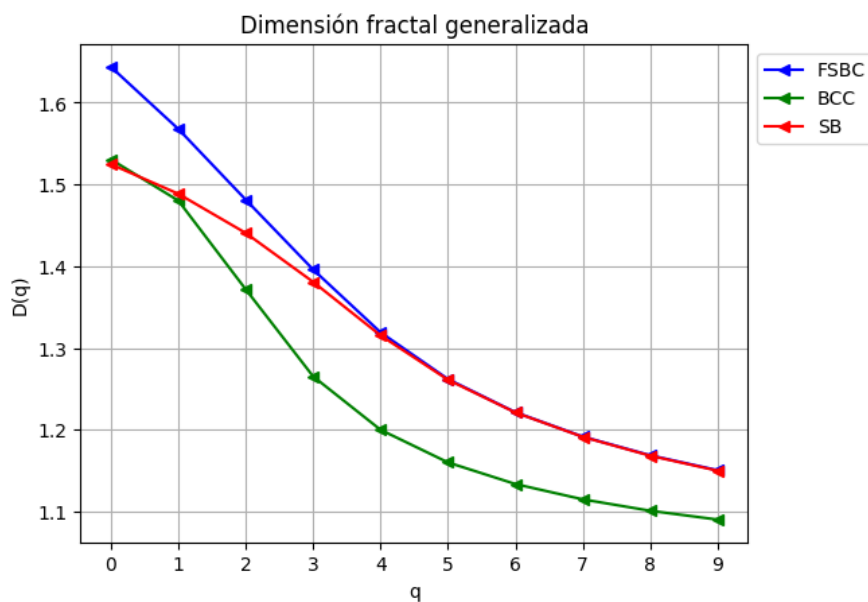


Figura 4.15: Dimensión fractal generalizada para red fractal (2,2)-flower

La dimensión fractal de esta red es 1.59

En el artículo sólo se realiza un análisis de gráficas, en el cual se indica que la red tiende a ser multifractal. Se indica que la estructura tiende a ser multifractal, ya que esta no es un fractal matemático, es decir que conserve su dimensión fractal sin importar la escala.

Los datos obtenidos de la red son.

Dato	BCFS	BCC	SB
Dimensión fractal	1.64	1.53	1.52
Dimensión fractal generalizada	1.56	1.48	1.48
Dimensión de correlación	1.48	1.37	1.44
Dimensión máxima	1.92	1.83	1.64
Dimensión mínima	1.13	1.83	1.13
Variación en la dimensión	0.79	0.74	0.51

Tabla 4.5: Comparación de los datos obtenidos de los diferentes métodos de análisis multifractal

4.5. Estrategias evolutivas

La idea de esta estrategia es buscar los centros de cajas directamente, de tal forma el algoritmo no deba realizar repeticiones

4.5.1. El algoritmo

El genotipo del algoritmo es un arreglo de enteros, el cual indica los nodos que van a ser centros de cajas. Este tiene tamaño entre el 40 % y 80 % del número total de nodos de la red.

$$[a_1, a_2, a_3, \dots, a_n] \quad (4.6)$$

El fenotipo es los centros de las cajas sobre los cuales se va calcular el algoritmo.

Población inicial

La población inicial son k individuos, cada uno de los cuales es una selección aleatoria de nodos sin repetir.

Función de evaluación

La evaluación de un individuo consiste en dos factores:

- El grado de cada uno de los centros, entre mayor sea, mayor puntaje tiene el individuo
- La distancia entre los centros, se busca que estén lo más alejados posibles

Un individuo es calificado de acuerdo a la suma de estos factores

Operadores

Para el cruce, se hace una selección por ruleta[85], el cruce entre dos individuos se genera de la siguiente forma:

1. Se toman los nodos de los dos padres y se incluyen en un conjunto
2. Se toman nodos aleatorios de este conjunto para generar un hijo

De esta forma, se cruzan los individuos para garantizar que se tengan hijos válidos. En cada generación se crea un número determinado de nuevos individuos, el cual es especificado como entrada al algoritmo.

Para la mutación, se selecciona un nodo de un individuo y se cambia por otro, conservando que no sea igual a otro nodo.

Cambio de población

Se eliminan los individuos con peor valor de función de evaluación, los cuales son reemplazados por los nuevos individuos creados en el cruce.

Criterio de parada

Se manejan dos criterios de parada, los cuales son entradas al algoritmo:

- Un número de iteraciones especificado por el usuario.
- Un criterio que consiste en observar si en un número determinado de iteraciones, no se presenta mejora en el mejor individuo.

4.6. Estrategias Recocido Simulado

Este algoritmo tiene la misma estrategia del algoritmo evolutivo, la diferencia radica en que iniciamos con un estado aleatorio y aceptamos el estado de acuerdo a la función de evaluación y a la temperatura del sistema.

El estado está descrito en términos de un conjunto de nodos de tamaño definido por el usuario, los cuales representan los centros de las cajas.

A diferencia del algoritmo evolutivo, en cada iteración comparamos el estado con un estado vecino. Este estado vecino se crea tomando aleatoriamente uno de los nodos del estado actual y cambiándolo por un nodo vecino en la red, conservando que los nodos no se repitan.

4.7. Estudio de los algoritmos de inteligencia artificial

Los algoritmos Genético y de Recocido simulado son comparados aplicando pruebas en las mismas redes seleccionadas de la sección 4.4

En el caso del algoritmo genético también es mostrada la evolución de la función de evaluación.

4.7.1. Red libre de escala

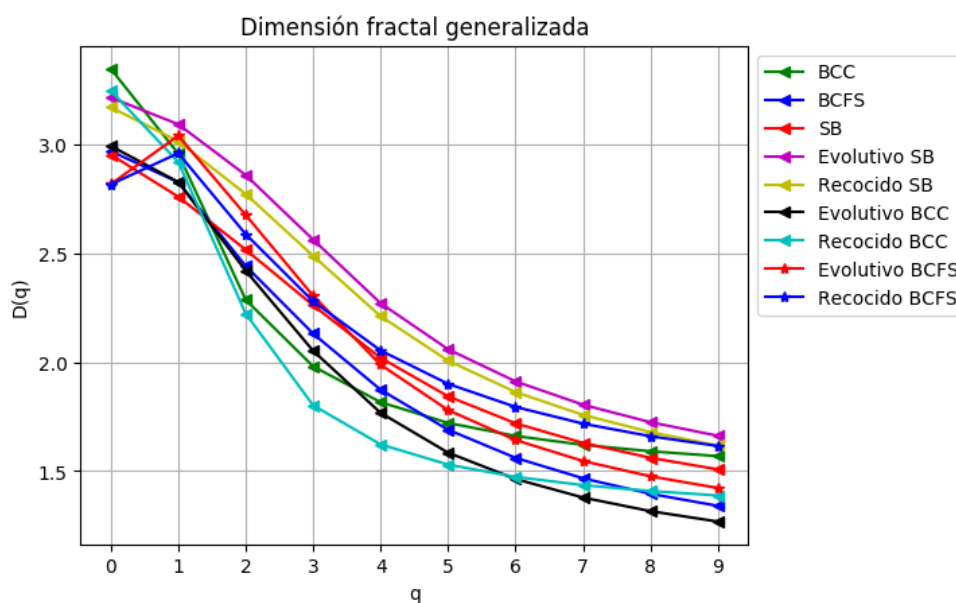


Figura 4.16: Multifractalidad en red libre de escala con 4000 nodos

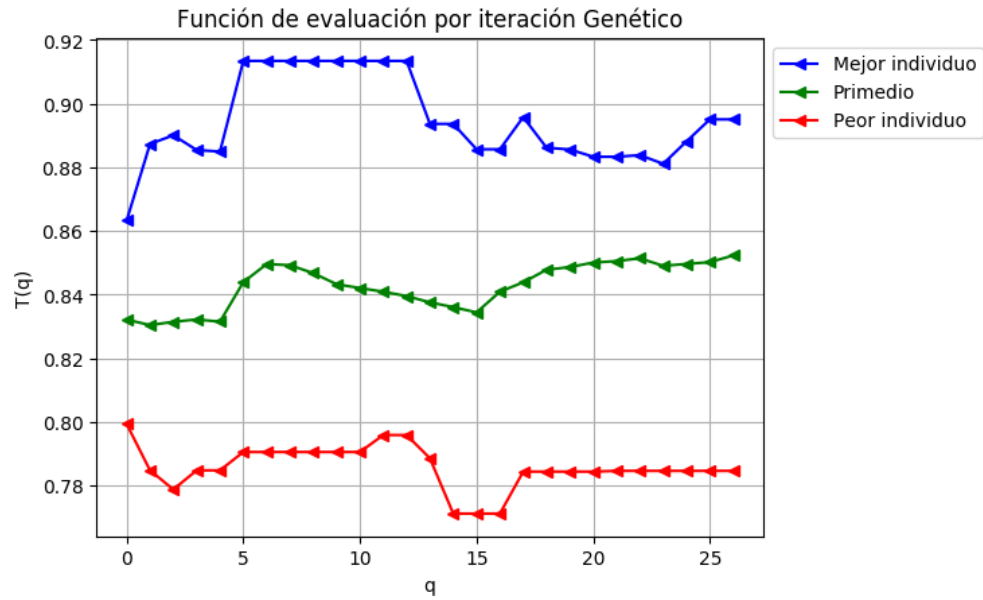


Figura 4.17: Evolución de la función de evaluación en red libre de escala con 4000 nodos

4.7.2. Red de mundo pequeño

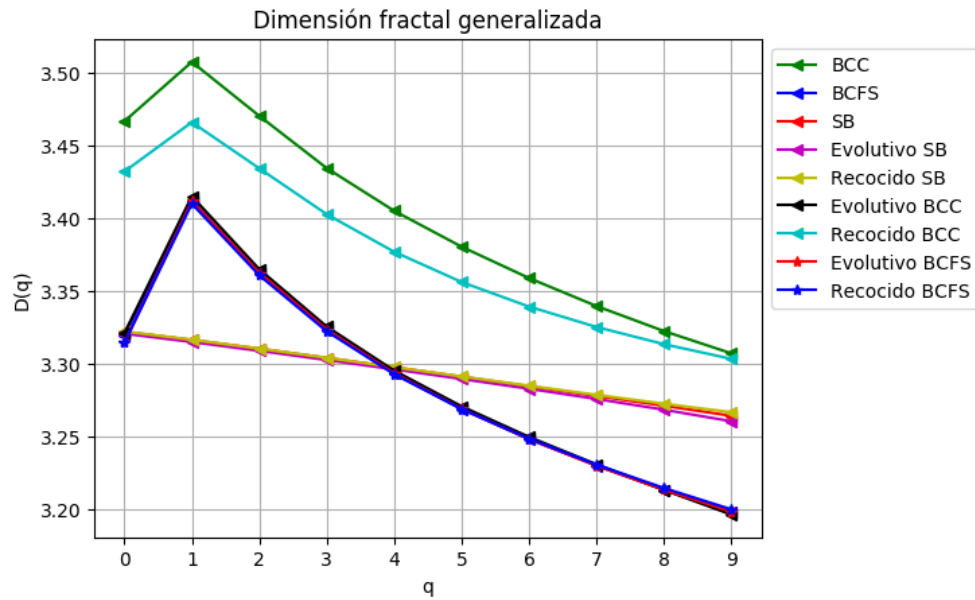


Figura 4.18: Multifractalidad en red de mundo pequeño de 5000 nodos con $p = 10\%$

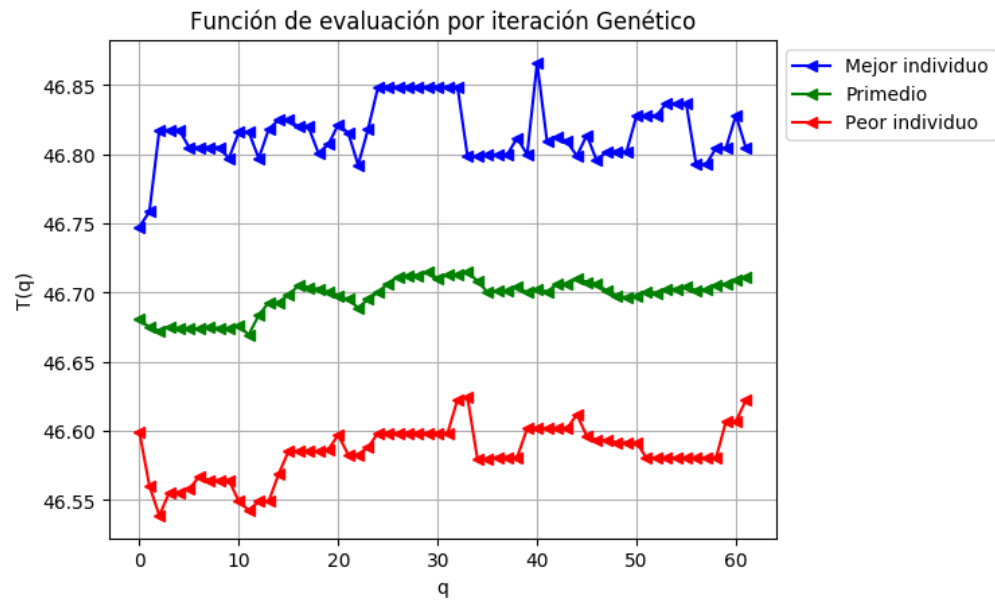


Figura 4.19: Evolución de la función de evaluación en red de mundo pequeño de 5000 nodos con $p = 10\%$

4.7.3. Red aleatoria

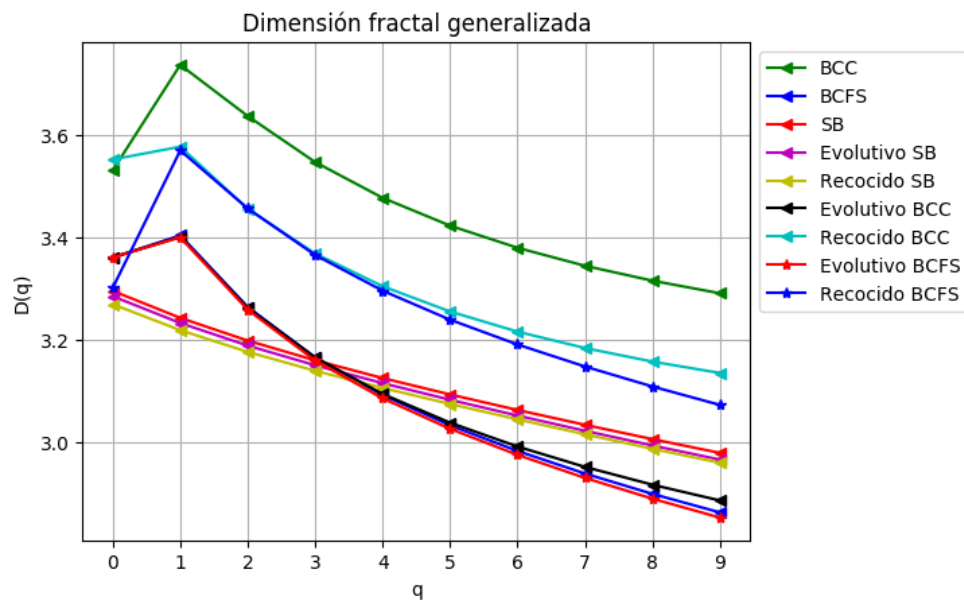


Figura 4.20: Multifractalidad en red aleatoria con 5620 nodos

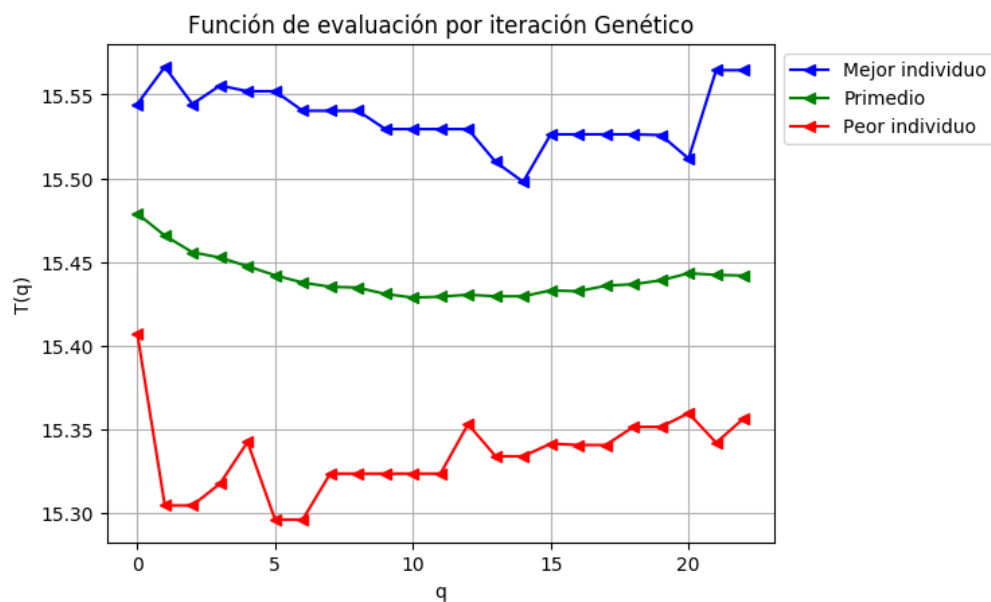


Figura 4.21: Evolución de la función de evaluación en red aleatoria con 5620 nodos

4.7.4. Red observada bacteria Ecoli

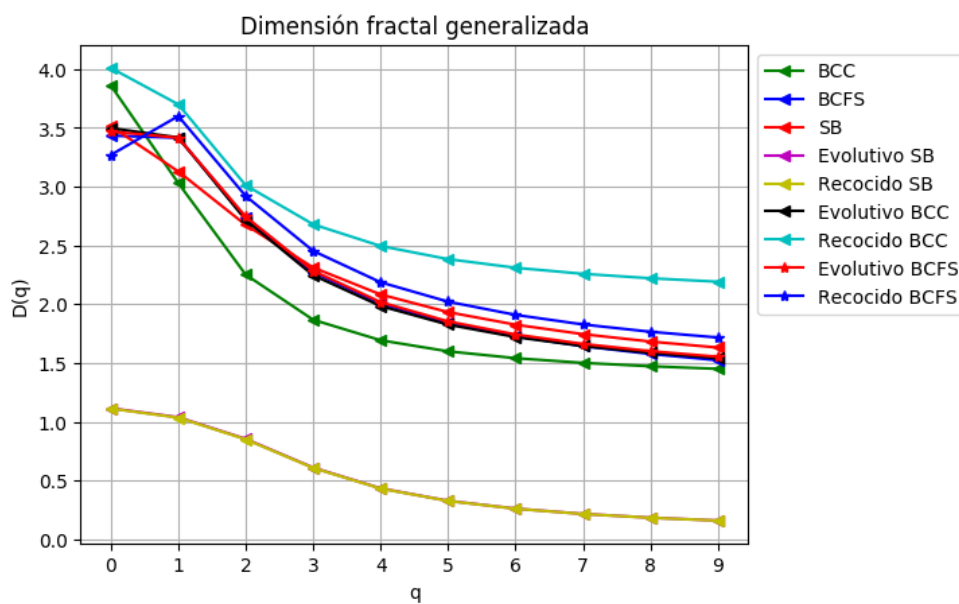


Figura 4.22: Multifractalidad en red observada de Ecoli

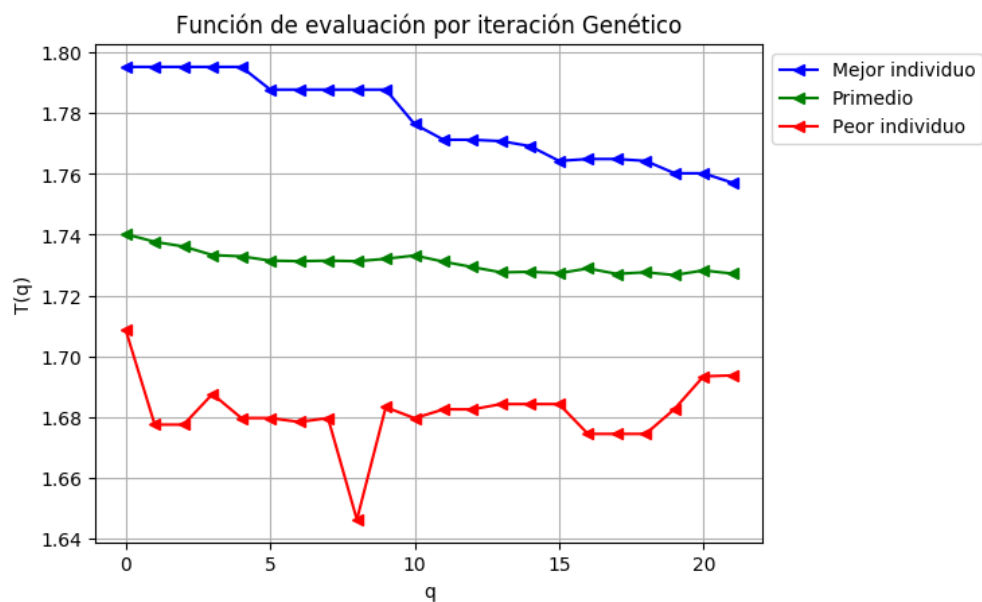


Figura 4.23: Evolución de la función de evaluación en red observada de Ecoli

4.7.5. Red fractal

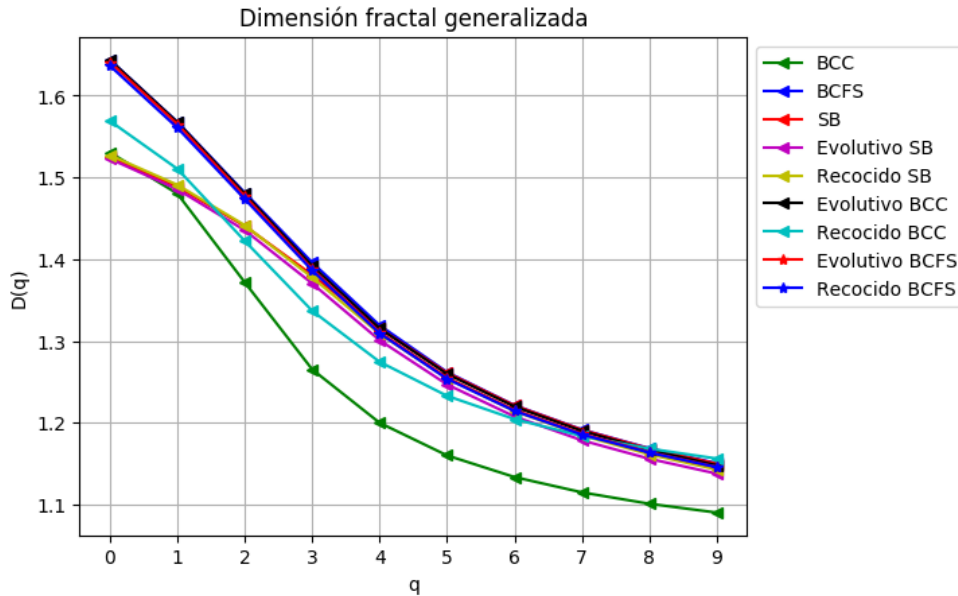


Figura 4.24: Multifractalidad en red generada fractal (2,2)-flower

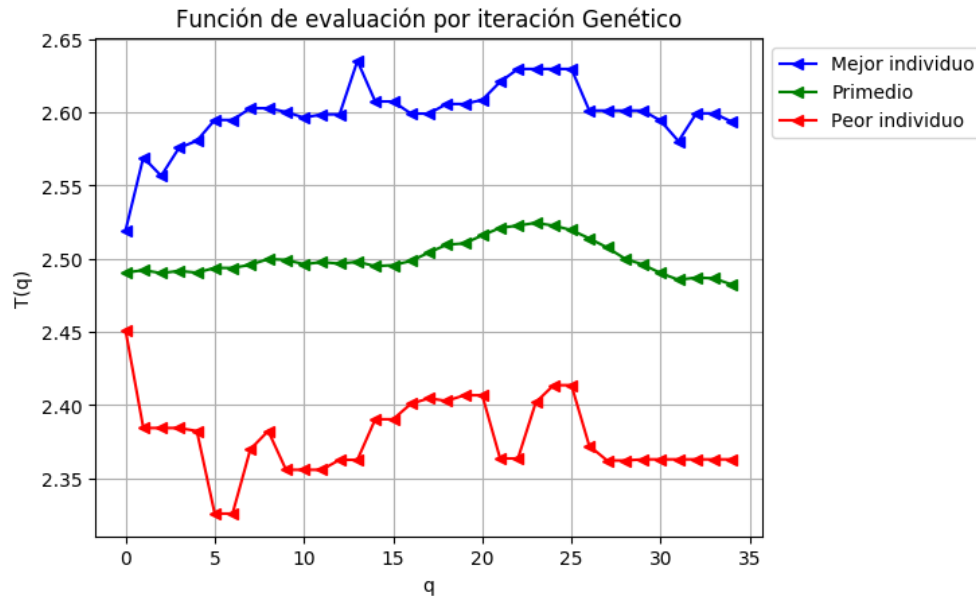


Figura 4.25: Evolución de la función de evaluación en red fractal (2,2)-flower

4.7.6. Análisis de los algoritmos

Se puede observar que:

1. No es sencillo estimar un número de iteraciones para los algoritmos de inteligencia artificial, en algunos casos la medida obtenida es similar a la obtenida con algoritmos deterministas, en otros no.
2. Dado que el tiempo requerido para la ejecución de los algoritmos es considerable (más de 100 horas) de acuerdo a tabla 4.6, se requiere un soporte teórico para conocer el número de iteraciones óptimo para cada estrategia.
3. Las gráficas de la evolución de la función de evaluación muestran que la heurística debe ser mejorada para puntar los candidatos a centros de cajas. Esto se debe que no hay una mejora significativa del promedio de la función de valuación, después de las primeras generaciones.

4.8. Análisis de los tiempos de ejecución

En la tabla 4.6 son analizados los tiempos de ejecución de cada uno de los algoritmos, de acuerdo al protocolo de pruebas especificado en el Anexo B de la página 88

Red	BCFS	BCC	SB	Genéti- co BCFS	Recocido BCFS	Genéti- co BCC	Recocido BCC	Genéti- co SB	Recocido SB
Red A	10788	10715	1193	7097	5092	6324	4995	4644	6653
Red B	84833	85043	16598	48511	38744	35022	48511	35067	48360
Red C	271062	380427	19535	16508	20410	27972	16508	24480	15692
Red D	106396	106751	60063	75580	61520	94662	57915	68997	60096
Red E	494803	394747	8430	13698	10614	31047	11136	14146	10448

Tabla 4.6: Tiempos de ejecución para diferentes redes en segundos

Se observa que en general los algoritmos BCFS y BCC tienen una mejora considerable cuando se les aplica estrategias de inteligencia artificial. Con respecto al algoritmo SB, se requiere realizar un estudio más amplio para llegar a una conclusión debido a que:

- El número de iteraciones de SB depende del número de nodos y del número de repeticiones del algoritmo.
- El número de iteraciones de los algoritmos de IA depende los parámetros suministrados por el usuario.

Lamentablemente, los algoritmos requieren gran capacidad de cómputo y tiempo de ejecución, por lo que se debe realizar en un estudio posterior.

4.9. Resumen y conclusiones del capítulo

En este capítulo, inicialmente se analizan 3 algoritmos presentes en la literatura: Box Counting Fixed Size (BCFS), Box Compact Counting (BCC) y SandBox (SB). Se encuentra que estos métodos se deben ejecutar un número no determinado de veces para hallar las dimensiones fractales, ya que dentro de sus pasos hay una selección aleatoria de centros de cajas.

Ninguno de los algoritmos estudiados garantiza un valor exacto de la dimensión fractal, por lo que se requiere realizar una búsqueda intensiva de los mejores centros para realizar el cómputo de las dimensiones fractales, en consecuencia, no son prácticos para redes con una gran cantidad de nodos. De hecho, en los artículos que son propuestos, el análisis es realizado con un conjunto selecto de redes de no más de 10000 nodos. Los algoritmos sólo proveen información de que tan heterogénea es la estructura interna de la red.

Estos algoritmos requieren un gran tiempo de cómputo, el cual depende del número de nodos, aristas y cómo esta conectada la red, por lo que proveer una medida de complejidad en términos de vértices o aristas no es práctica. En consecuencia, este estudio se realiza comparando el tiempo de cómputo en distintos ejemplos.

Como estrategia para reducir el tiempo de cómputo, se propone un algoritmo evolutivo y uno de recocido simulado, la idea es mediante un proceso distinto al de cubrimiento de cajas que debe estudiar toda la red elegir los centros de las cajas. La estrategia se basa en una heurística que consiste en buscar una configuración de nodos, los cuales tengan el mayor grado posible y se encuentren lo más alejados entre sí que sea posible.

Los resultados de la tabla 4.6, dan indicios que las estrategias de inteligencia artificial mejoran el tiempo de ejecución de los algoritmos BCFS y BCC: Para el algoritmo SB, debido a que depende de una selección aleatoria y del número de repeticiones que se le indiquen, no se puede concluir si las estrategias propuestas lo mejoran o no.

Capítulo 5

Análisis de robustez

5.1. Análisis de robustez

La medición de la robustez se utiliza el índice R presentado por Schneider[14] para calcular la robustez en la red. En el cual se toman dos medidas:

- El número de nodos con respecto al total inicial de nodos del componente gigante a medida que la red pierde nodos
- El promedio de los caminos más cortos del componente gigante a medida que la red pierde nodos

El algoritmo de robustez se describe de la siguiente forma:

1. Se calculan las medidas iniciales en la red sin quitar ningún nodo.
2. Se van quitando simultáneamente 10 % de los nodos de la red bajo alguna estrategia de ataque elegida.
3. El índice R de robustez es el promedio de la suma de las medidas que se toman cada vez que la red pierde un porcentaje de nodos simultáneamente.

Es de anotar que en este caso se utiliza la medida de ataque simultaneo y no secuencial debido a que se debe realizar el análisis multifractal cada vez y debido a su elevado costo computacional, no es factible realizar las pruebas repitiéndolo muchas veces.

5.2. Estrategias de ataque

Por grado

Se eliminan los nodos de acuerdo a su grado.

Por centralidad

Se eliminan los nodos de acuerdo a su centralidad. Con esto se busca destruir la capacidad de transmitir información de la red.

Ataque con estrategia evolutiva

La estrategia de ataque evolutivo consiste en seleccionar 10 % de los nodos, de acuerdo a la heurística definida en la sección 4.5. El objetivo es realizar un ataque seleccionando los centros de las cajas que se utilizan para el análisis multifractal.

Ataque con estrategia recocido simulado

Esta estrategia es seleccionar cada vez el 10 % de los nodos, en base a la estrategia definida en la sección 4.6.

El objetivo, al igual que en el caso del ataque genético es eliminar primero los nodos que son seleccionados como centros de las cajas en el análisis multifractal.

5.3. Pruebas y análisis de robustez en diferentes redes

Las redes que se van a estudiar son:

1. Red A: Red libre de escala 8000 nodos
2. Red B: Red de mundo pequeño 5000 nodos, probabilidad de reconexión 10 %
3. Red C: Red aleatoria 1991 nodos
4. Red D: Red obtenida de bacteria *C.elegans*
5. Red E: Red fractal (1,3)-flower

5.3.1. Pruebas en red libre de escala

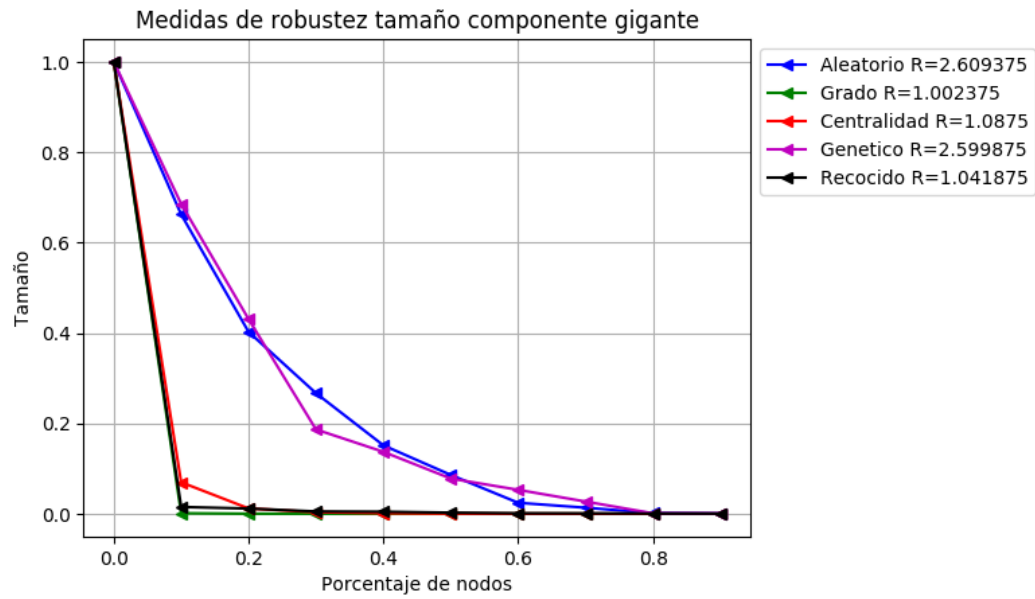


Figura 5.1: Medida componente de gigante

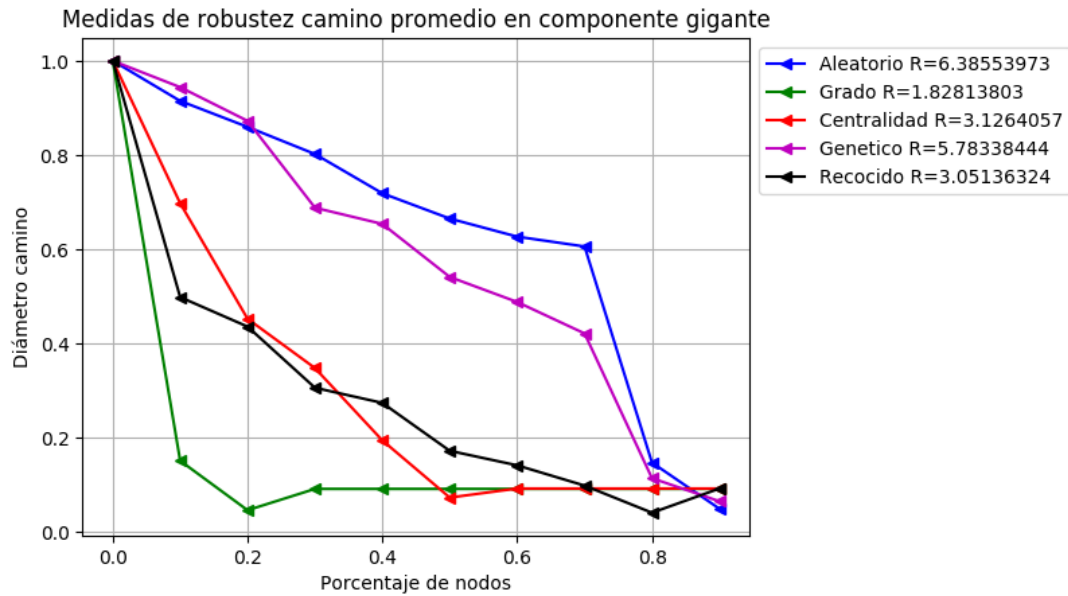


Figura 5.2: Medida componente de promedio de camino más cortos

En las redes libres de escala, las medidas indican que la red pierde sus propiedades estructurales más rápidamente con los ataques de grado y centralidad. Esto se debe a que la pérdida hubs, implica que

la red pierde capacidad de transmitir información y esto se refleja en que los caminos más cortos entre cualquier par de nodos, tienden a ser más largos.

Los ataques genético y recocido presentan una efectividad intermedia de afectación de las propiedades estructurales de la red, son mejores que el caso aleatorio, pero en las pruebas realizadas no presentan mejores resultados que los ataques por grado y centralidad.

5.3.2. Pruebas red de mundo pequeño

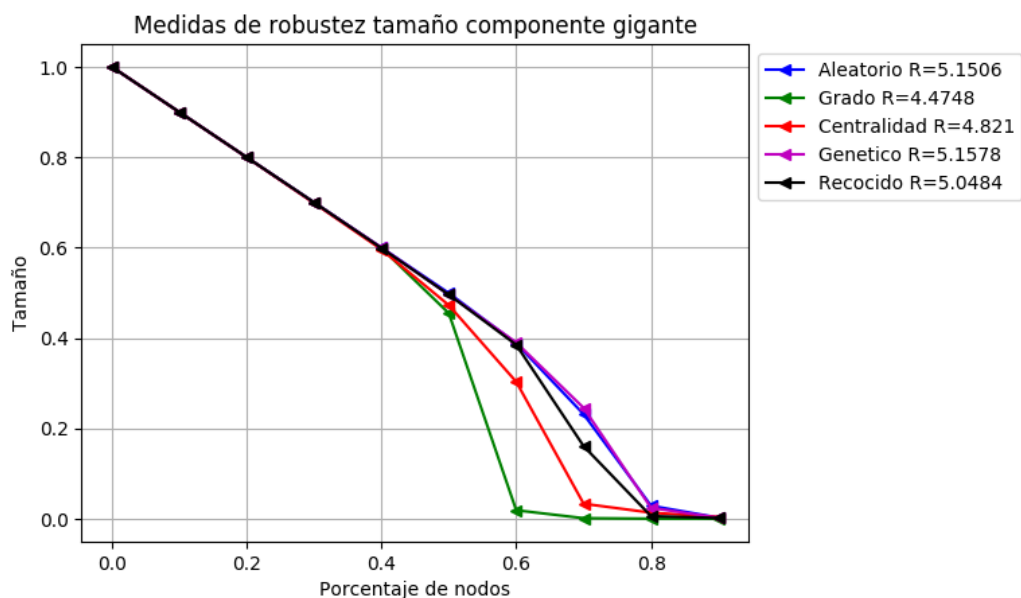


Figura 5.3: Medida componente de gigante

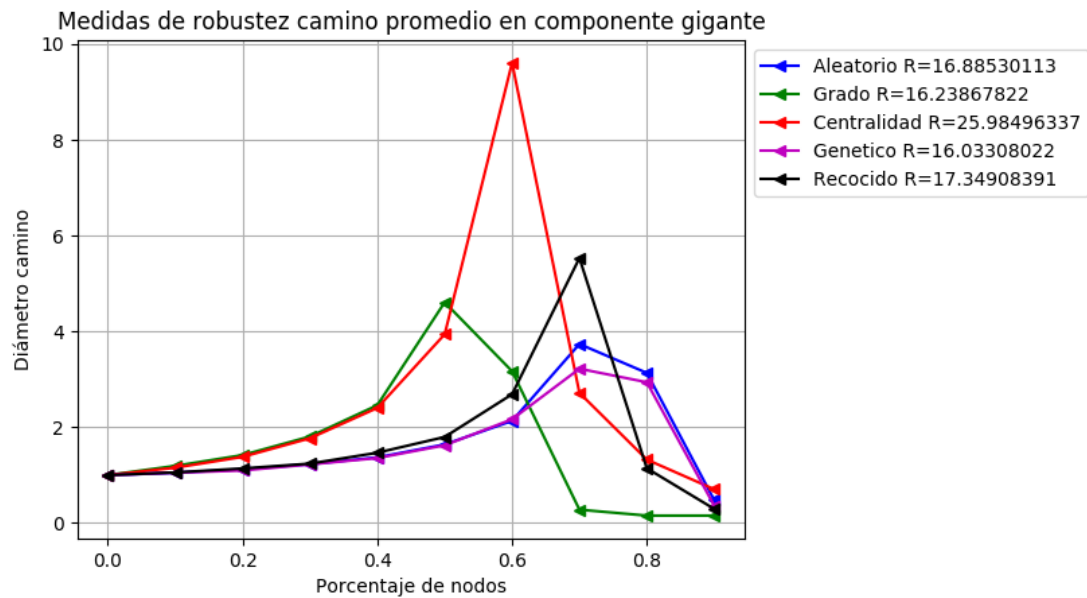


Figura 5.4: Medida componente de promedio de camino más cortos

En el caso de las redes de mundo pequeño, la medida que cobra importancia es el promedio de los caminos más cortos, ya que al destruir nodos por centralidad se pierde la propiedad de conservar un valor promedio pequeño de distancia más corta entre un par de nodos cualesquiera.

5.3.3. Pruebas red aleatoria

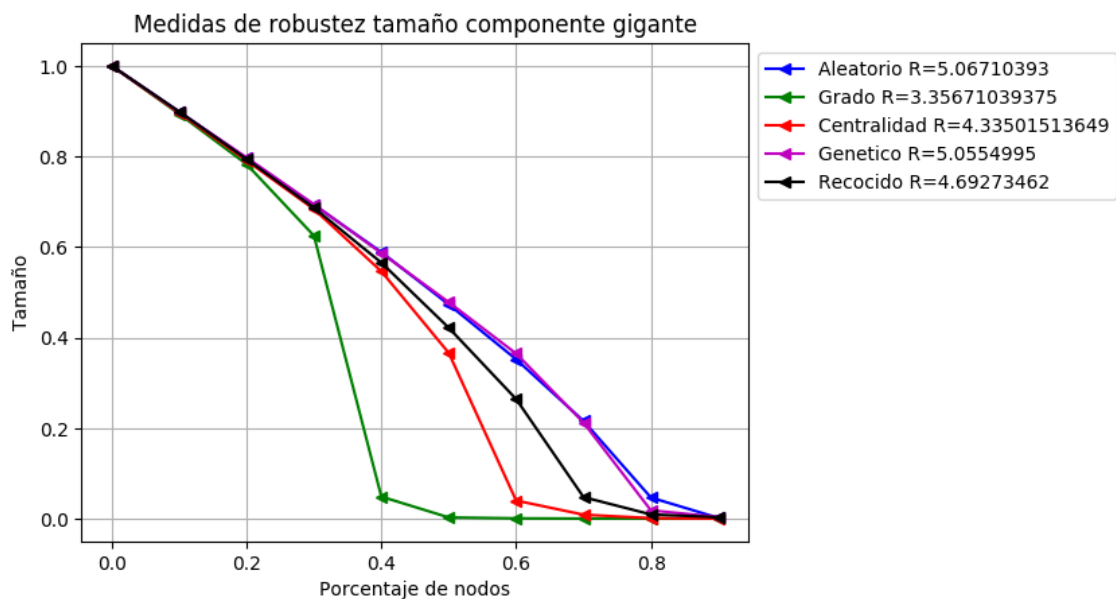


Figura 5.5: Medida componente de gigante

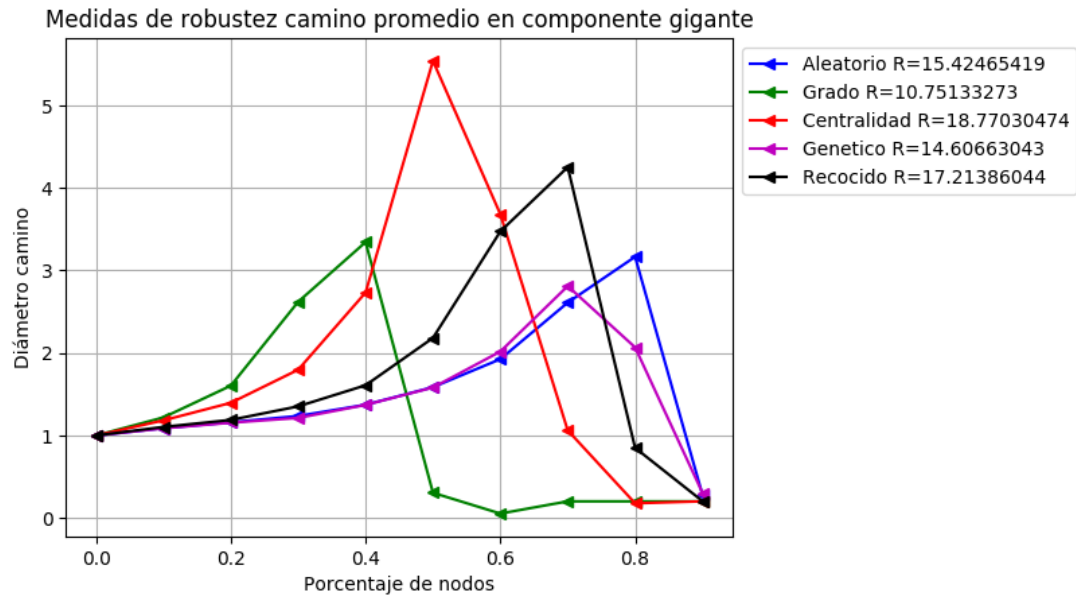


Figura 5.6: Medida componente de promedio de camino más cortos

En las redes aleatorias, el ataque por centralidad tiene el mismo efecto que las redes de mundo pequeño.

5.3.4. Prueba en red observada Celengs

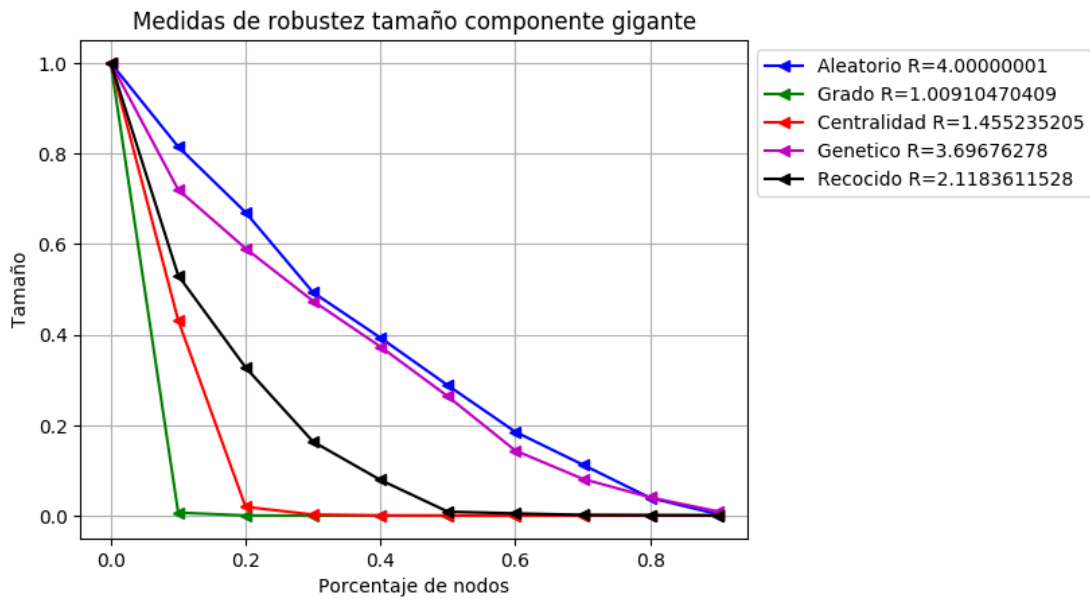


Figura 5.7: Medida componente de gigante

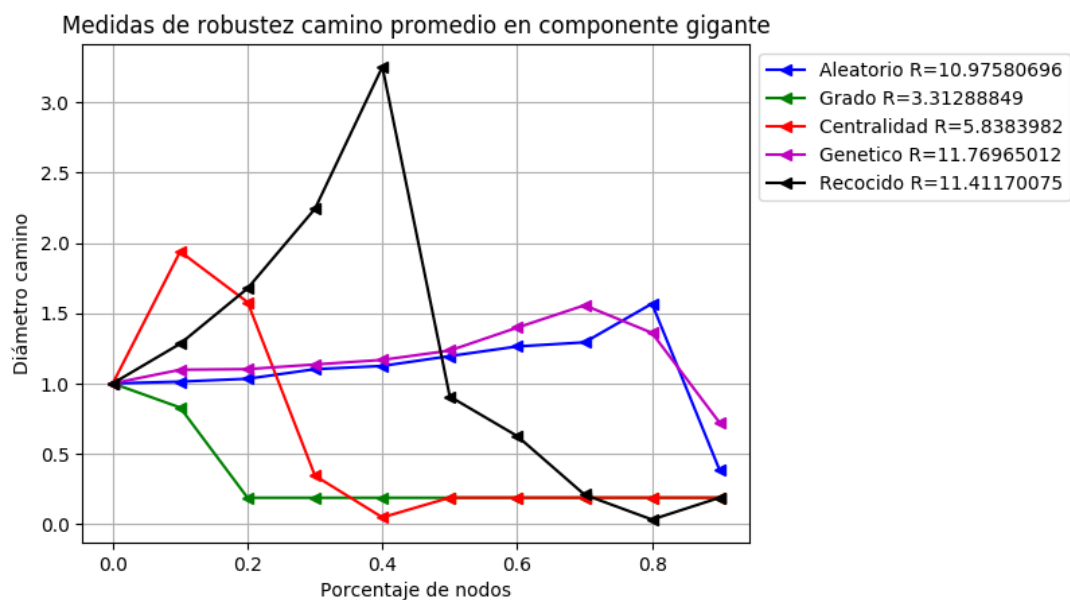


Figura 5.8: Medida componente de promedio de camino más cortos

Este resultado indica que al eliminar nodos de acuerdo a su grado, el tamaño del componente gigante se ve reducido considerablemente.

5.3.5. Prueba red fractal

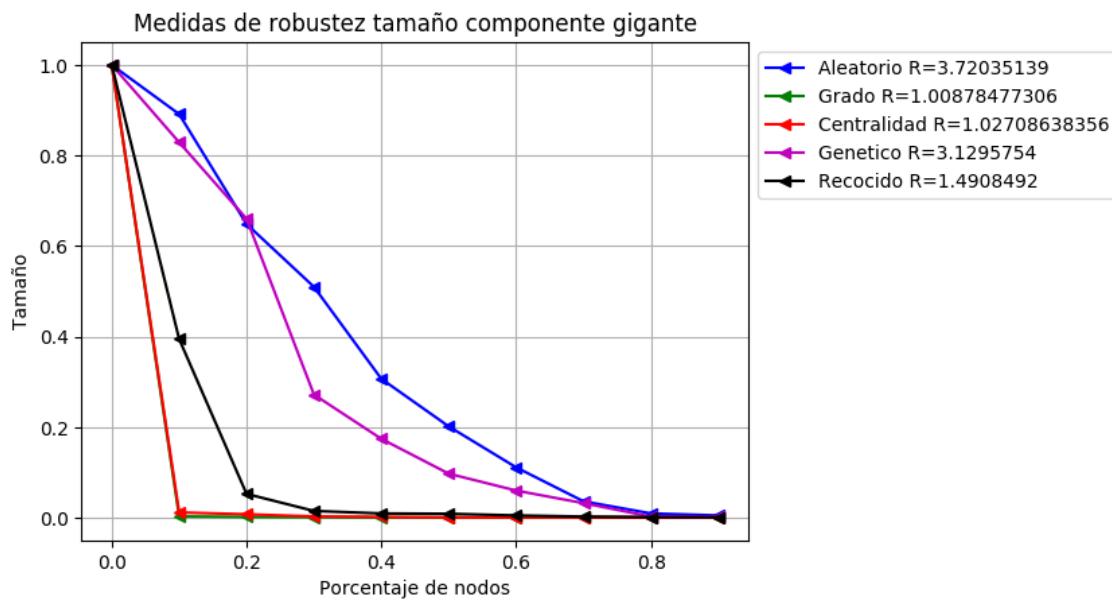


Figura 5.9: Medida componente de gigante

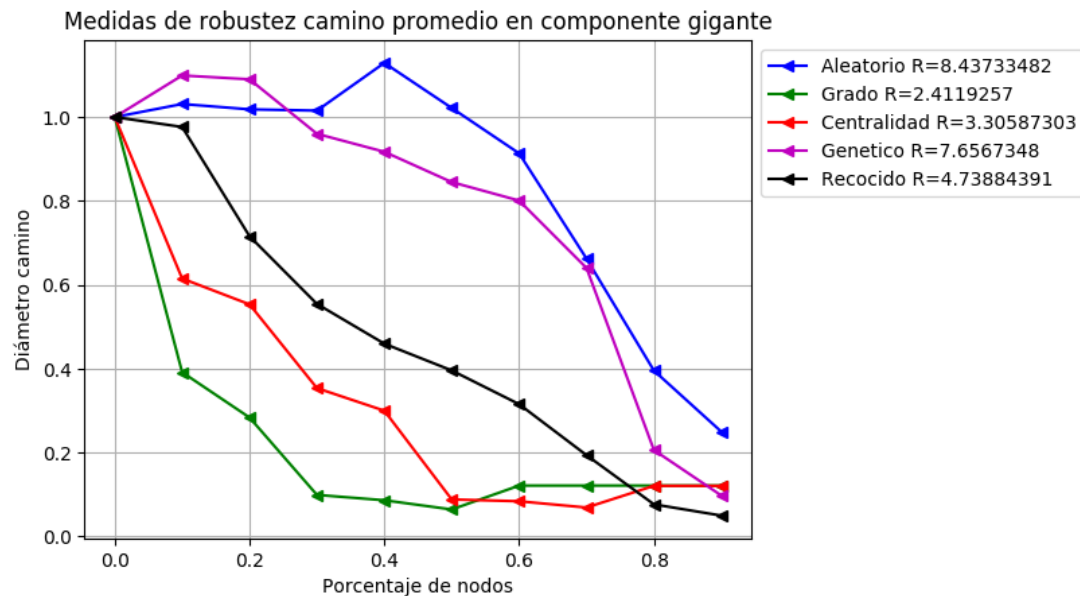


Figura 5.10: Medida componente de promedio de camino más cortos

En el caso de la red fractal, la estrategia de ataque más eficiente es la de centralidad. Esto se explica debido a que los fractales se generan iterativamente a partir de los nodos de las generaciones anteriores, los cuales pasan a ser parte de los caminos más cortos entre cualquier par de nodos.

5.4. Resumen y conclusiones del capítulo

La robustez es una medida que permite analizar cómo una red conserva su estructura a medida que pierde nodos

- Para las redes libres de escala, se encuentra que tienen una gran tolerancia a fallos aleatorios, sin embargo, si sus hubs se ven afectados, la red pierde rápidamente sus propiedades estructurales. En la mayoría de casos después de perder el 20 % de los nodos con mayor grado, es tal el efecto en la estructura de la red que las medidas muestran que hay pequeños grupos de nodos no conexos.
- Las redes de mundo pequeño conservan sus propiedades en la mayoría de ataques, excepto el de centralidad ya que su estructura está basada en que la distancia promedio entre cada par de nodos sea la menor posible, lo que implica que muchos caminos pasan por un número determinado de nodos.
- En redes aleatorias, las medidas robustez dependen de su estructura y puede variar de acuerdo cómo es la estructura interna de la red. En la mayoría de redes examinadas, el ataque por grado es el que más efecto produce en la red.

Considerando la eficiencia de daño de acuerdo a las dos medidas utilizadas, las estrategias evolutiva y de recocido simulado, presentan un comportamiento mejor que el caso aleatorio, pero peor que

los ataques por centralidad o grado. Esto se debe a que estas medidas están pensadas para destruir los centros de las cajas para el análisis multifractal.

Capítulo 6

Análisis de multifractalidad y robustez

6.1. Análisis multifractal

6.1.1. Relación entre multifractalidad y robustez

El análisis multifractal provee una herramienta para examinar la estructura de una red. Ahora, se va hacer una relación entre esa medida y algunas redes. Las estrategias de ataque fueron definidas en el capítulo 5.

Las redes de estudio son:

1. Red A: Red libre de escala 8000 nodos
2. Red B: Red de mundo pequeño 5000 nodos, probabilidad de reconexión 10 %
3. Red C: Red aleatoria 1991 nodos
4. Red D: Red observada en bacteria *C.elegans*
5. Red E: Red fractal (1,3)-flower

Debido a que se debe ejecutar el análisis de multifractalidad a medida que se pierde un porcentaje de nodos, se utiliza el algoritmo SandBox.

Red libre de escala

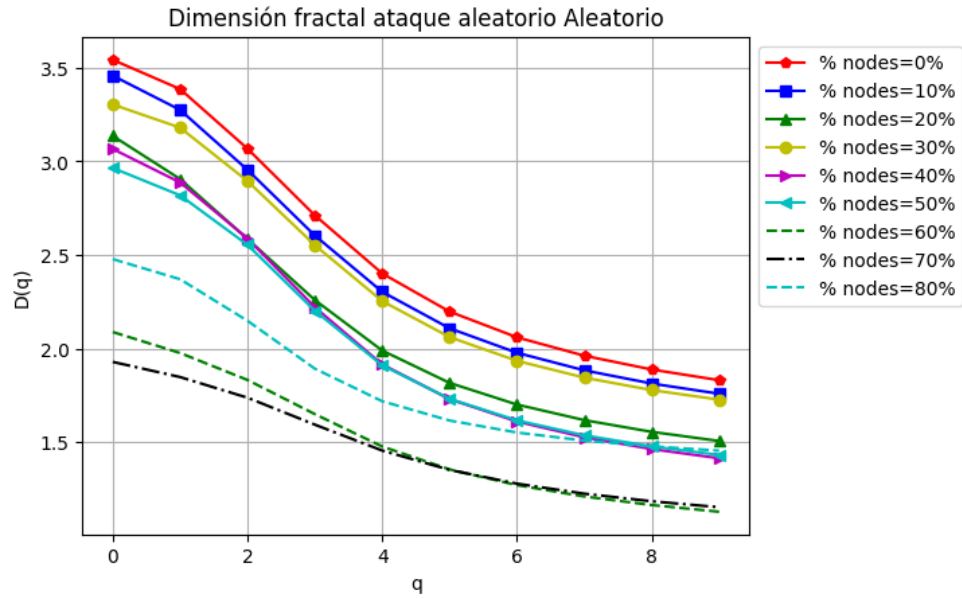


Figura 6.1: Análisis de multifractalidad de red libre de escala a ataque aleatorio

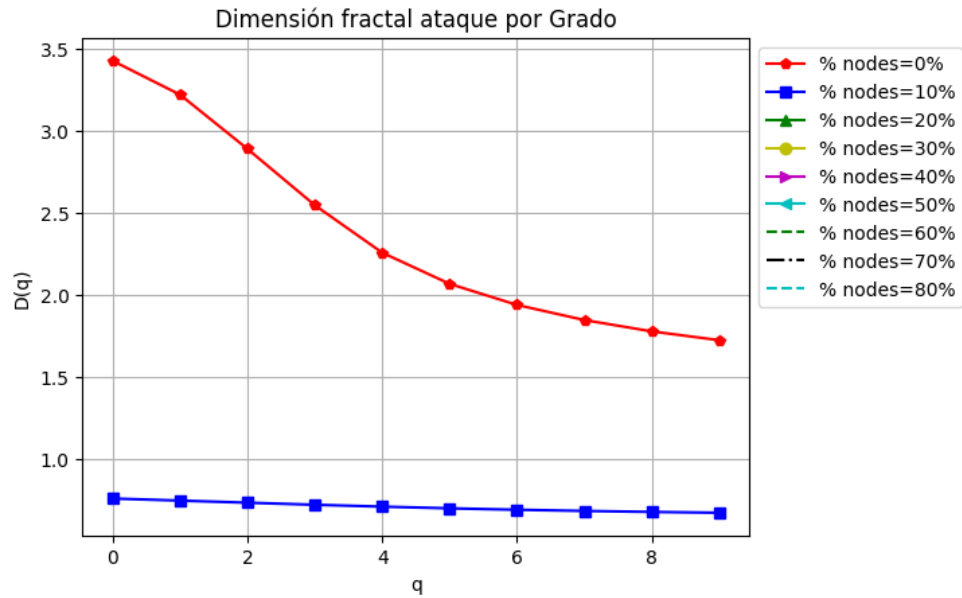


Figura 6.2: Análisis de multifractalidad de red libre de escala a ataque por grado

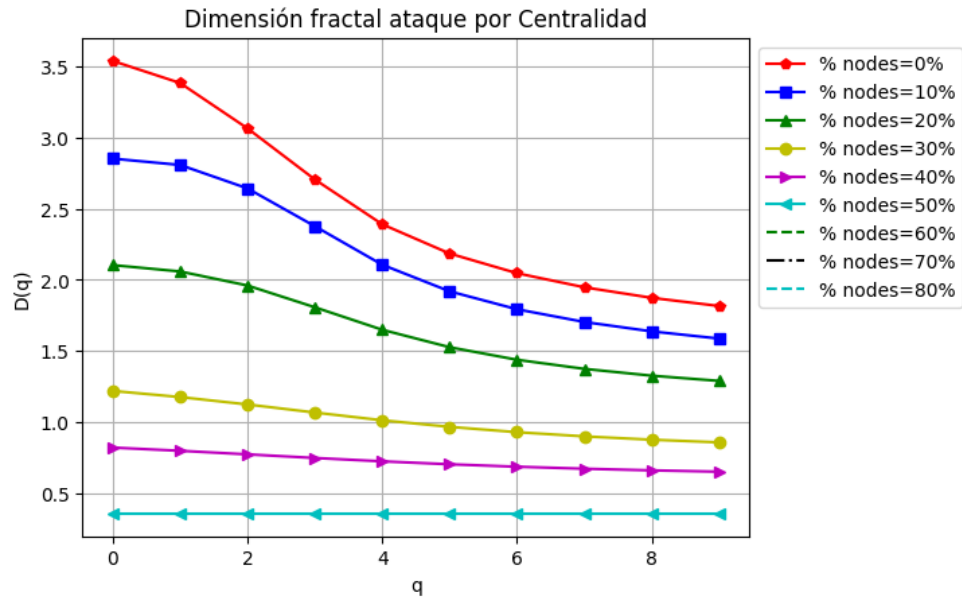


Figura 6.3: Análisis de multifractalidad de red libre de escala a ataque por centralidad

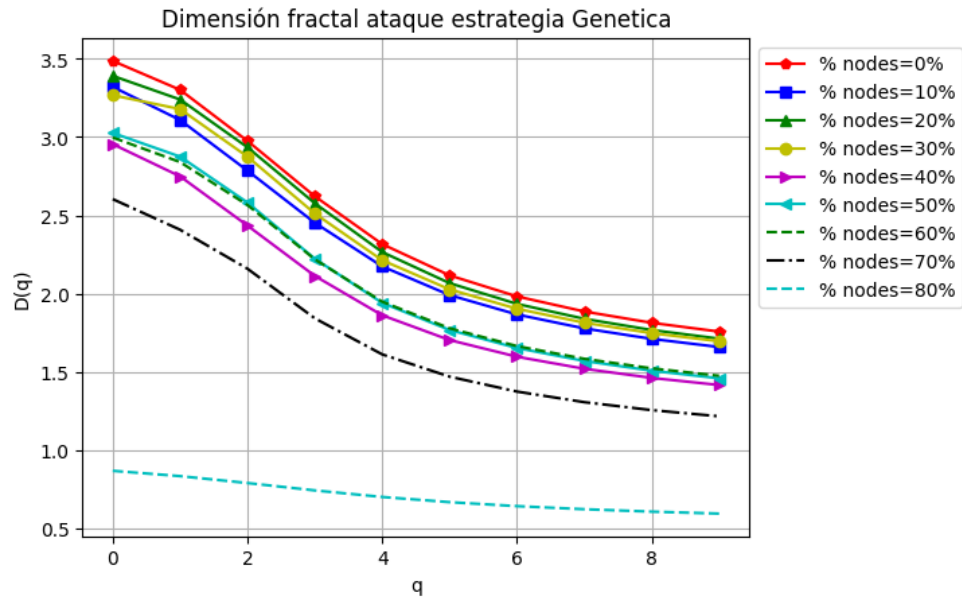


Figura 6.4: Análisis de multifractalidad de red libre de escala a ataque por estrategia evolutiva

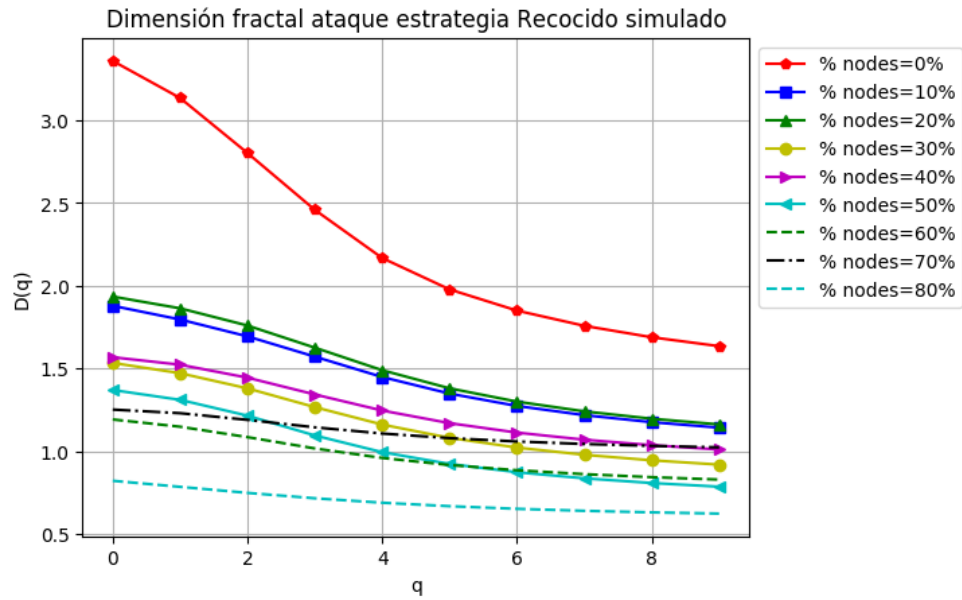


Figura 6.5: Análisis de multifractalidad de red libre de escala a ataque por estrategia recocida simulada

Red de mundo pequeño

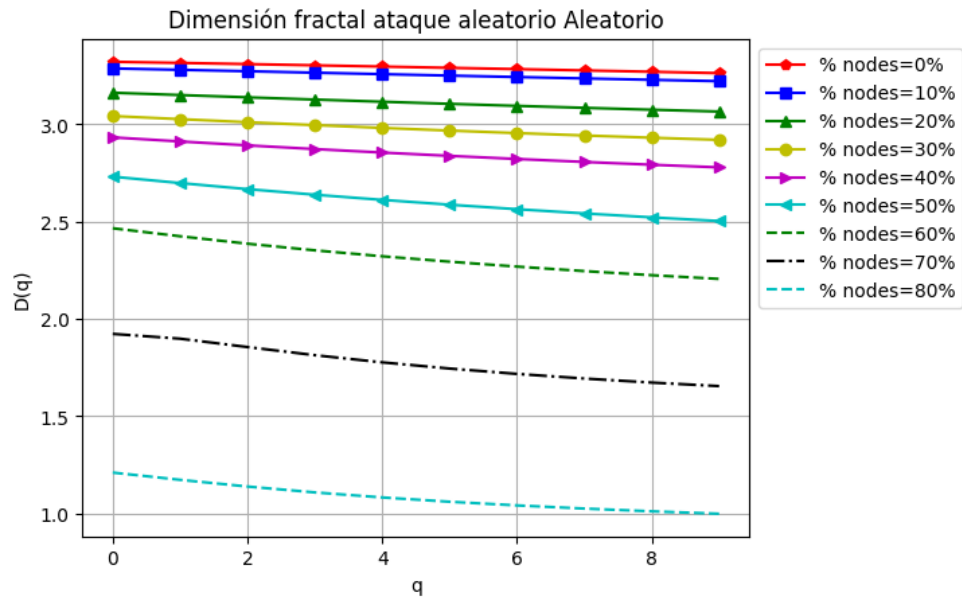


Figura 6.6: Análisis de multifractalidad de red de mundo pequeño ataque aleatorio

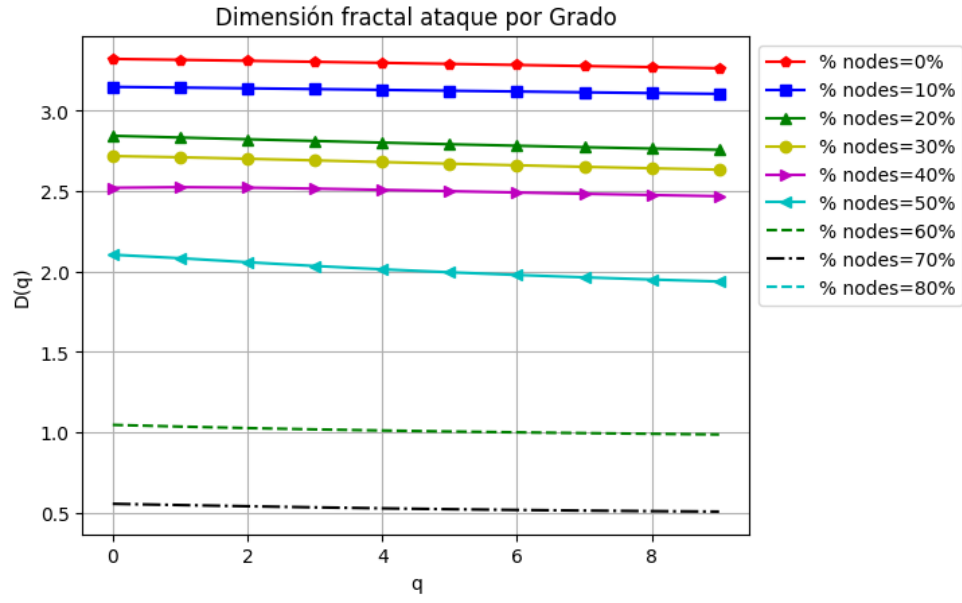


Figura 6.7: Análisis de multifractalidad de red de mundo pequeño a ataque por grado

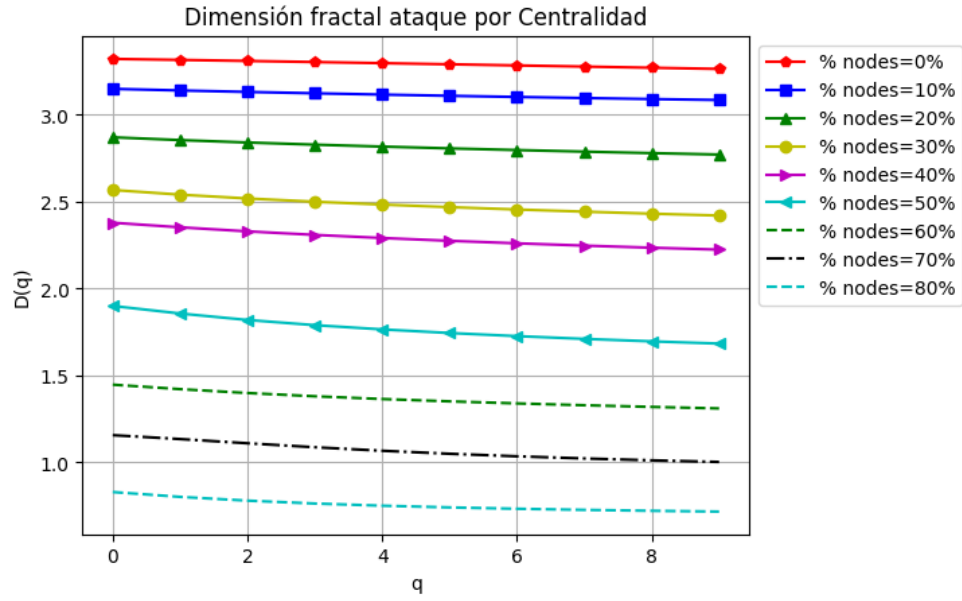


Figura 6.8: Análisis de multifractalidad de red de mundo pequeño a ataque por centralidad

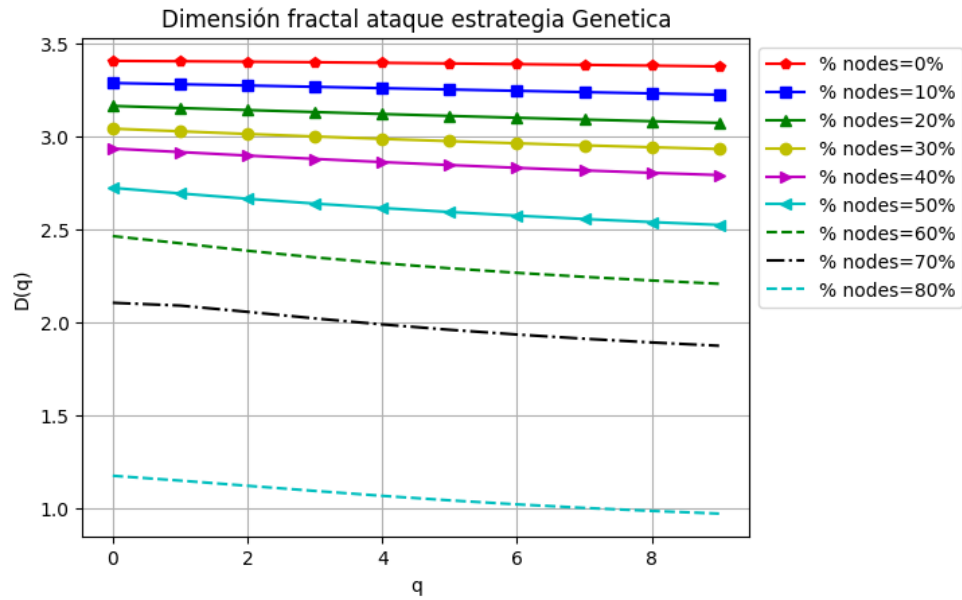


Figura 6.9: Análisis de multifractalidad de red de mundo pequeño a ataque por estrategia evolutiva

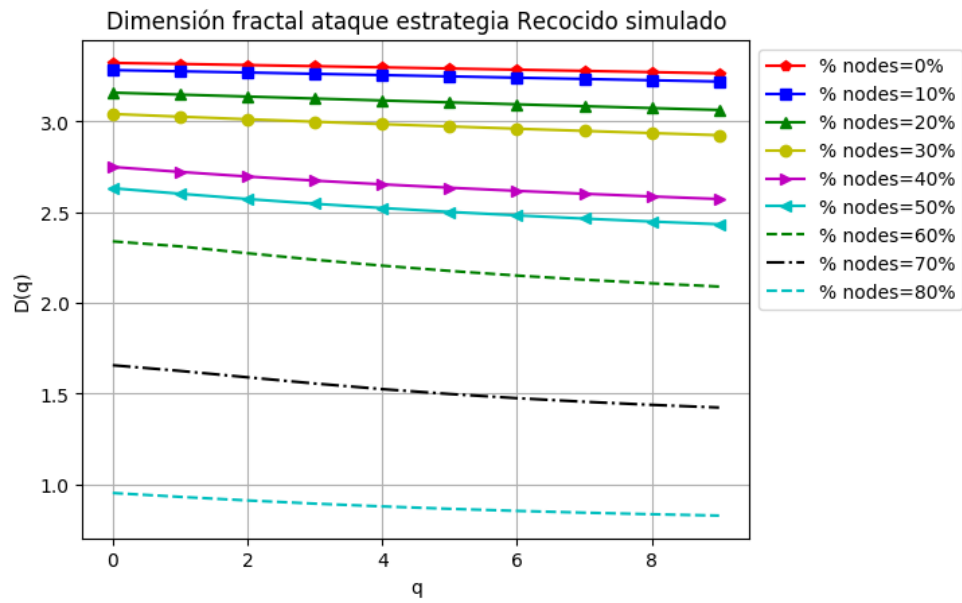


Figura 6.10: Análisis de multifractalidad de red de mundo pequeño a ataque por estrategia recocida simulada

Red aleatoria

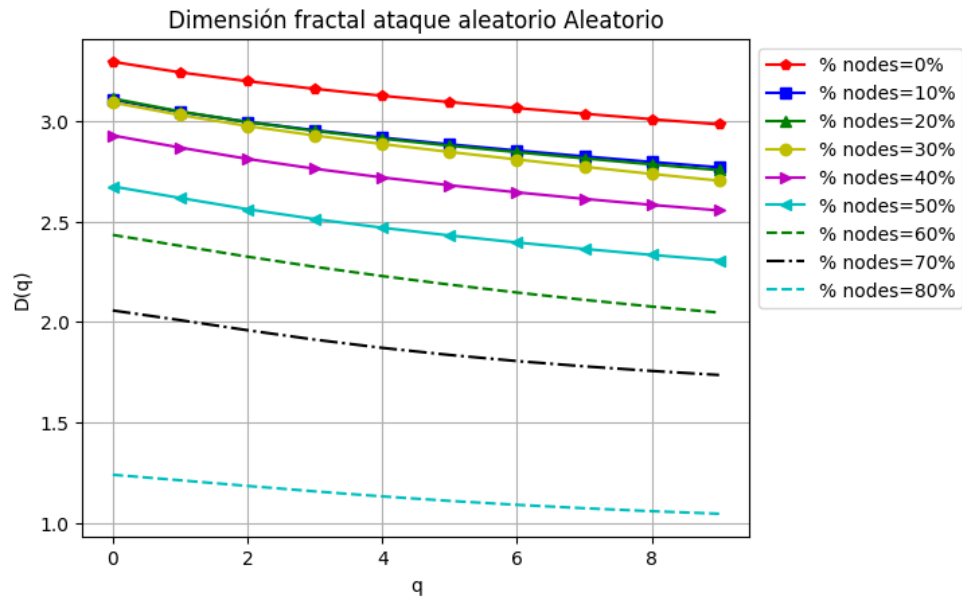


Figura 6.11: Análisis de multifractalidad de red aleatoria a ataque aleatorio

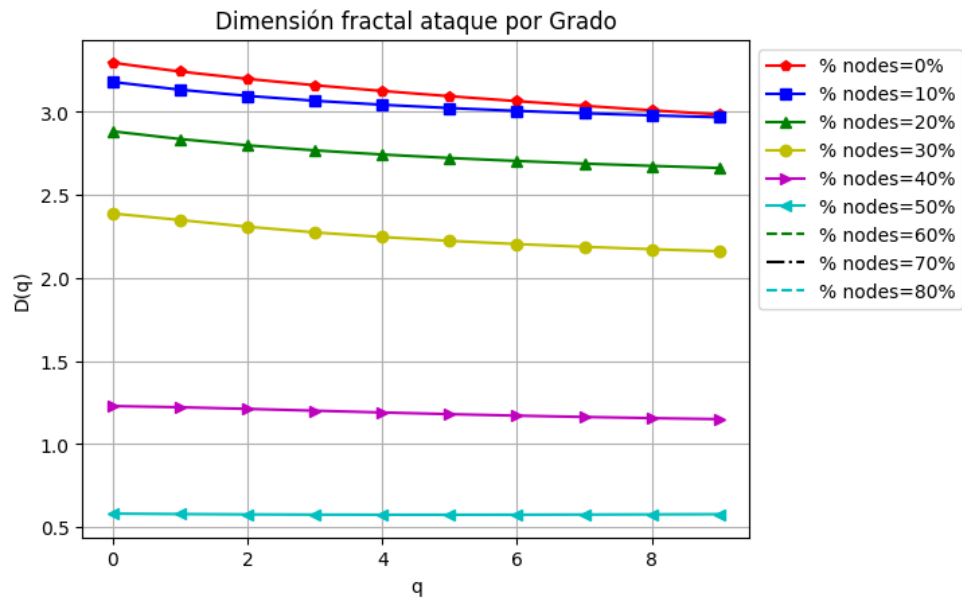


Figura 6.12: Análisis de multifractalidad de red aleatoria a ataque por grado

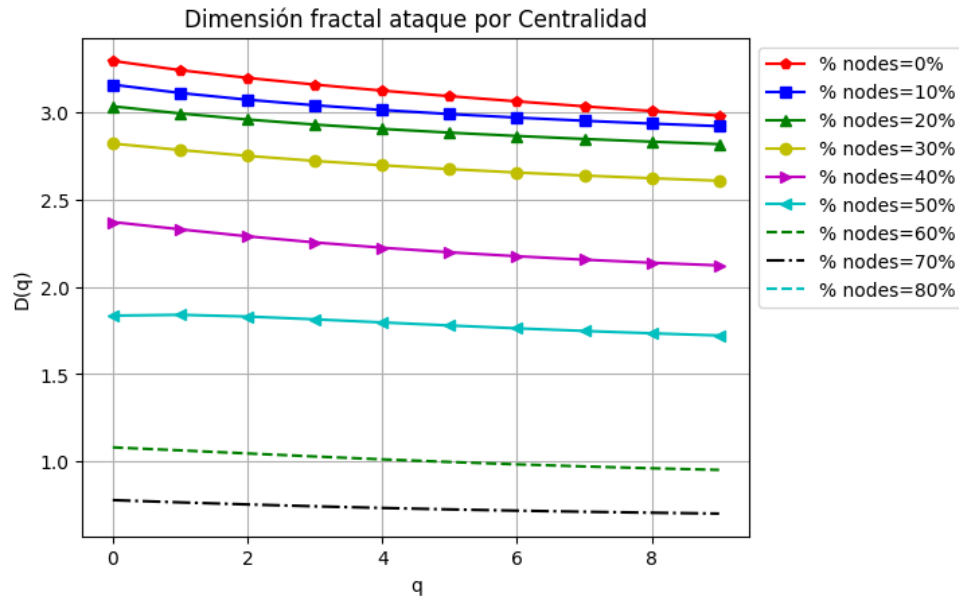


Figura 6.13: Análisis de multifractalidad de red aleatoria a ataque por centralidad

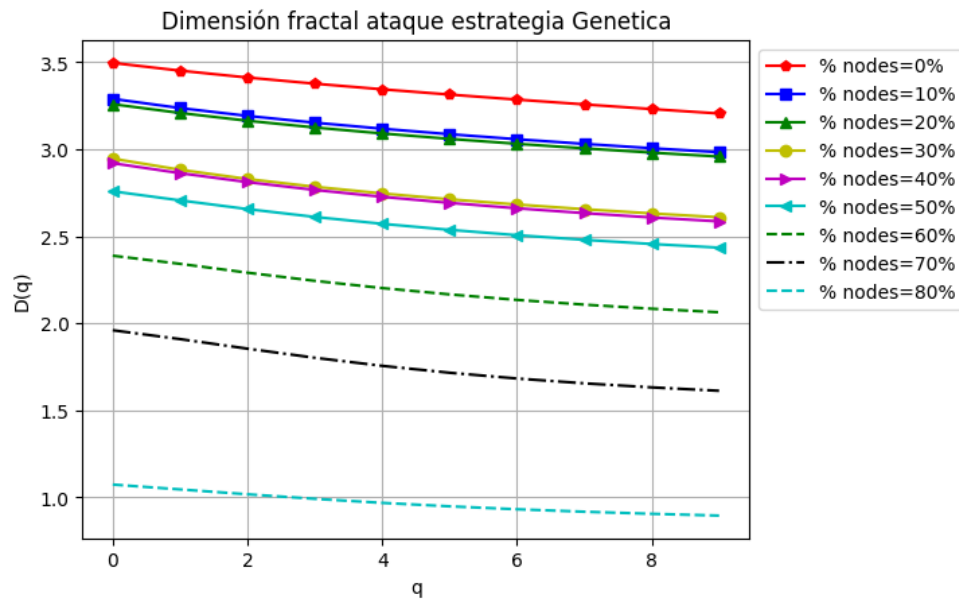


Figura 6.14: Análisis de multifractalidad de red aleatoria a ataque por estrategia evolutiva

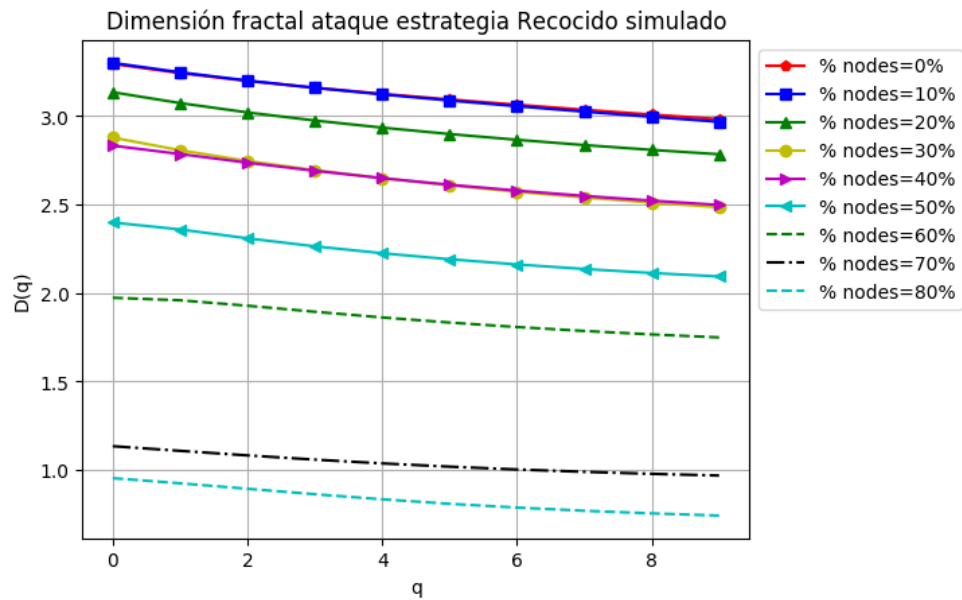


Figura 6.15: Análisis de multifractalidad de red aleatoria a ataque por estrategia recocida simulada

Red real

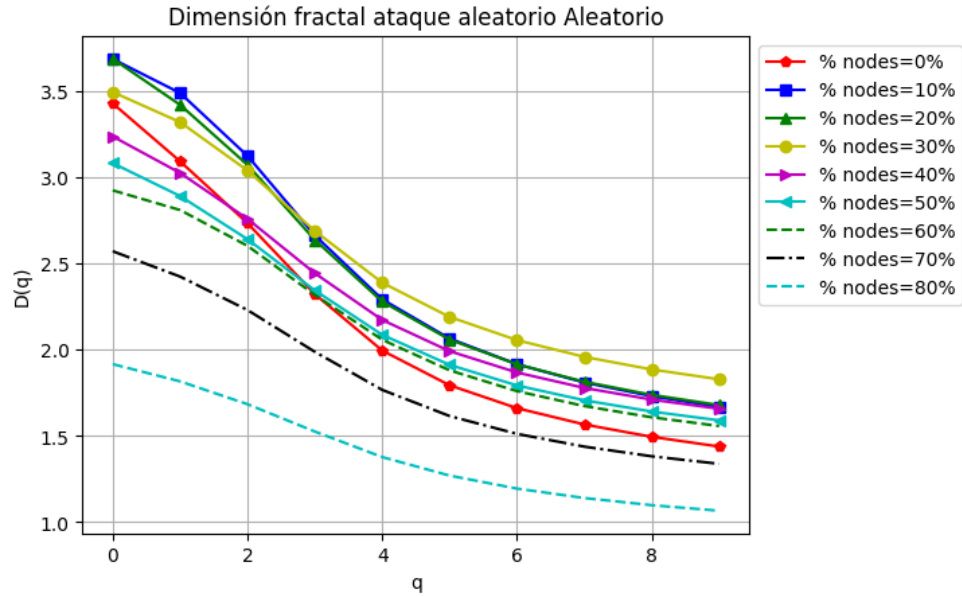


Figura 6.16: Análisis de multifractalidad de red observada de Celegens a ataque aleatorio

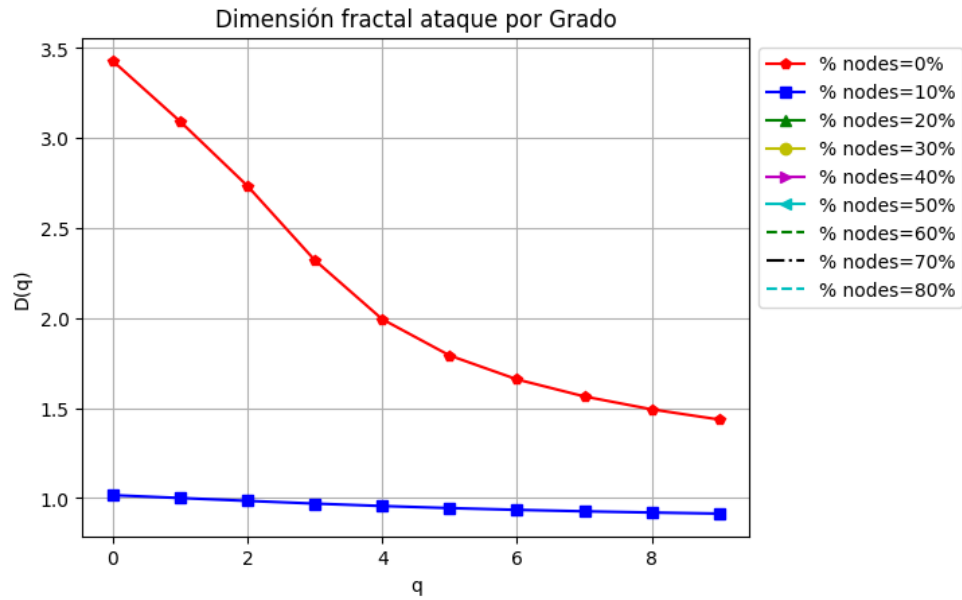


Figura 6.17: Análisis de multifractalidad de red observada de Celegens a ataque por grado

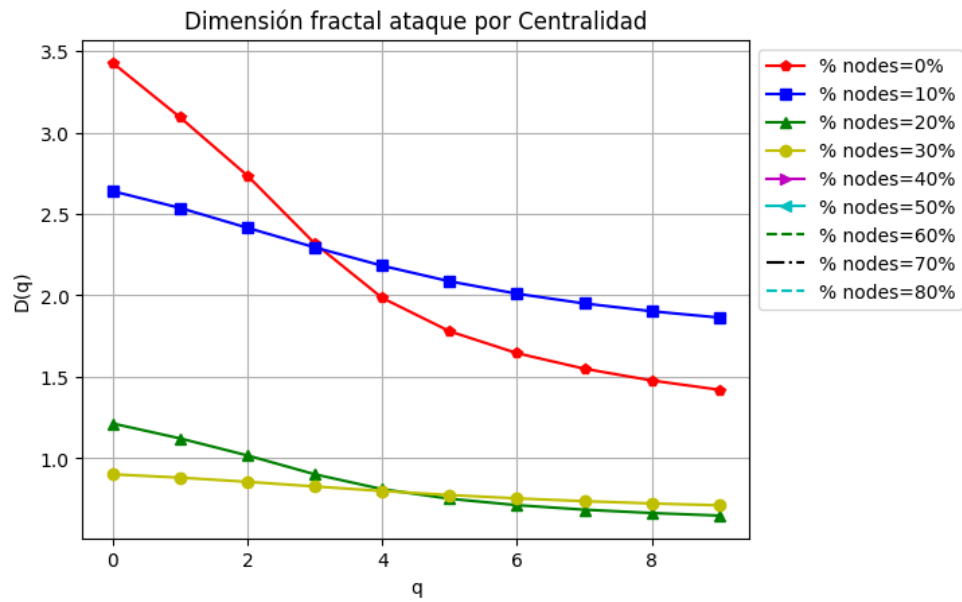


Figura 6.18: Análisis de multifractalidad de red observada de Celegens a ataque por centralidad

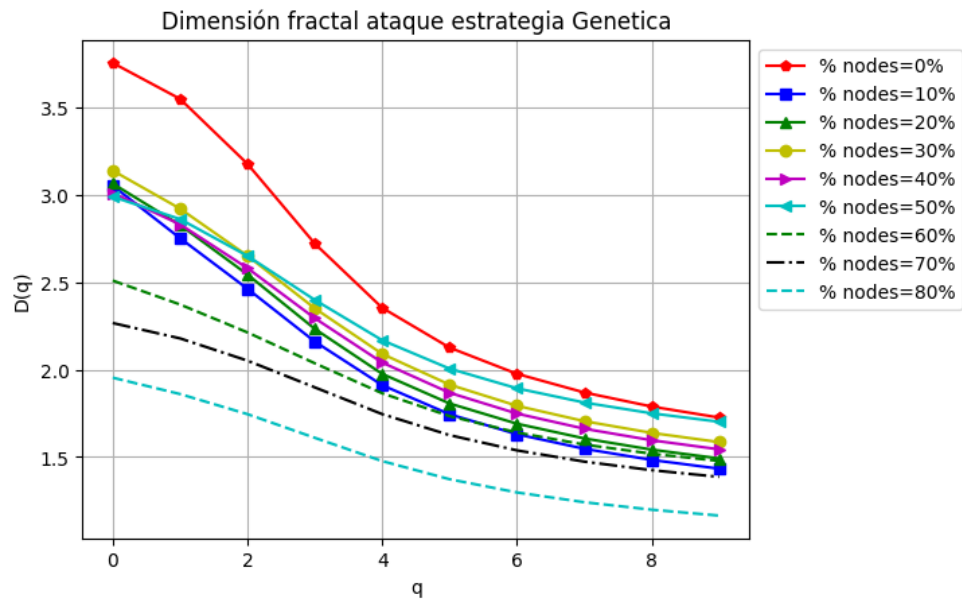


Figura 6.19: Análisis de multifractalidad de red observada de Celegens a ataque por estrategia evolutiva

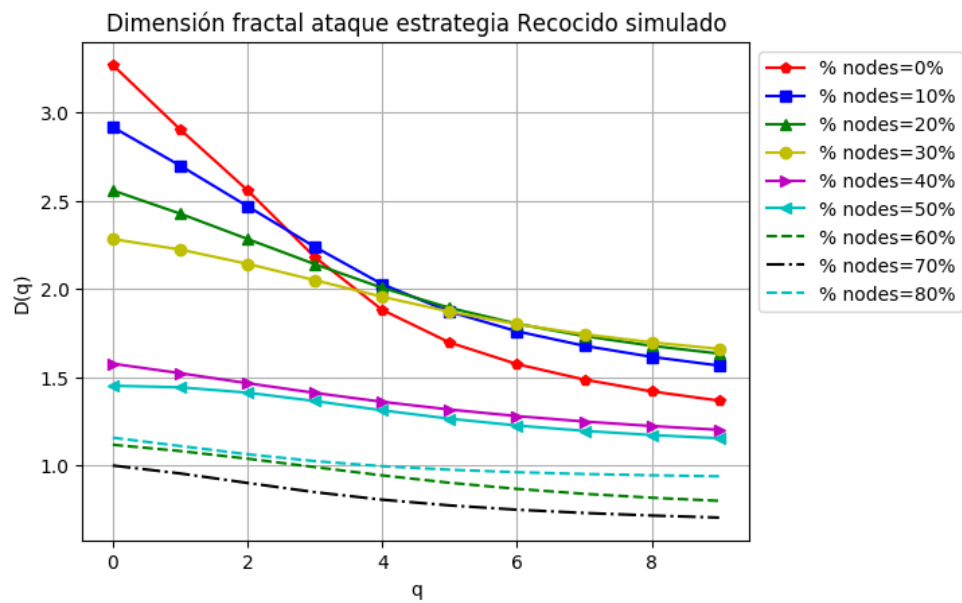


Figura 6.20: Análisis de multifractalidad de red observada de Celegens a ataque por estrategia recocida simulada

Red fractal

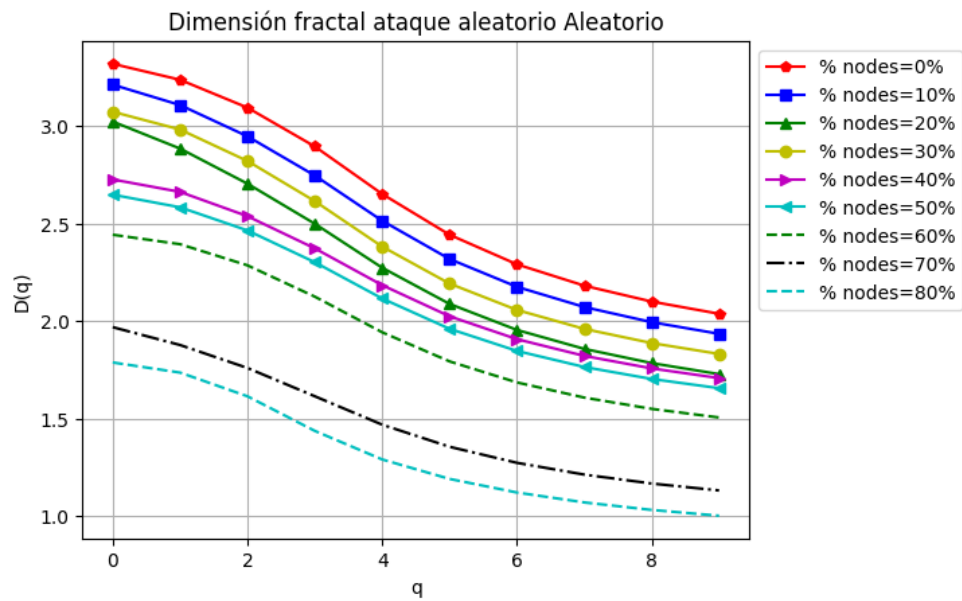


Figura 6.21: Análisis de multifractalidad de red (1,3)-flower a ataque aleatorio

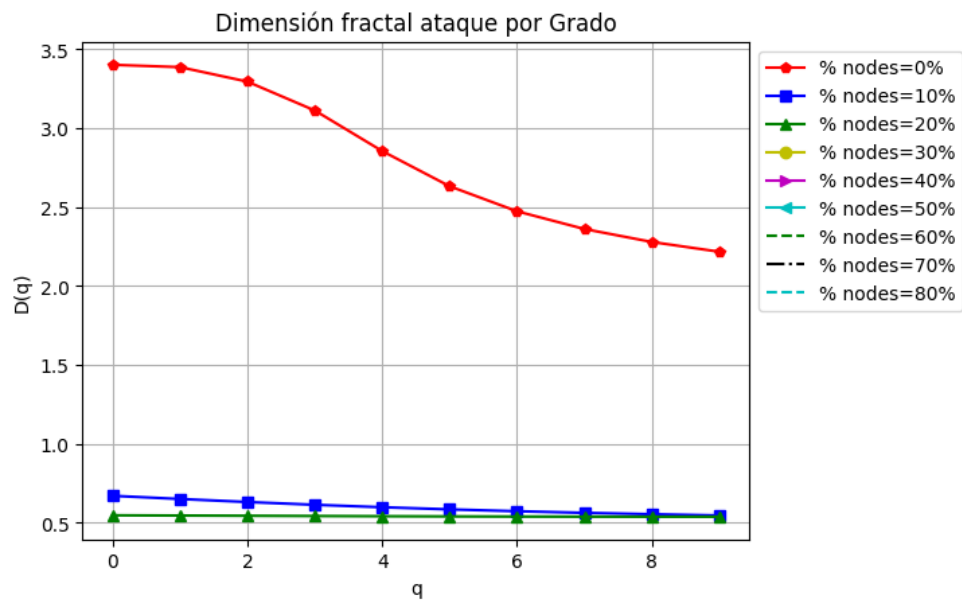


Figura 6.22: Análisis de multifractalidad de red (1,3)-flower a ataque por grado

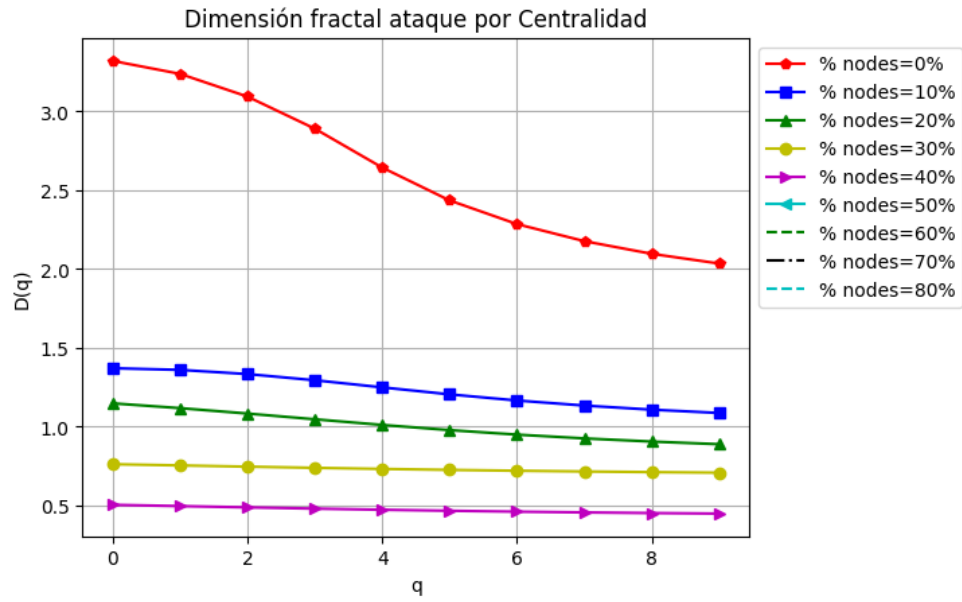


Figura 6.23: Análisis de multifractalidad de red (1,3)-flower a ataque por centralidad

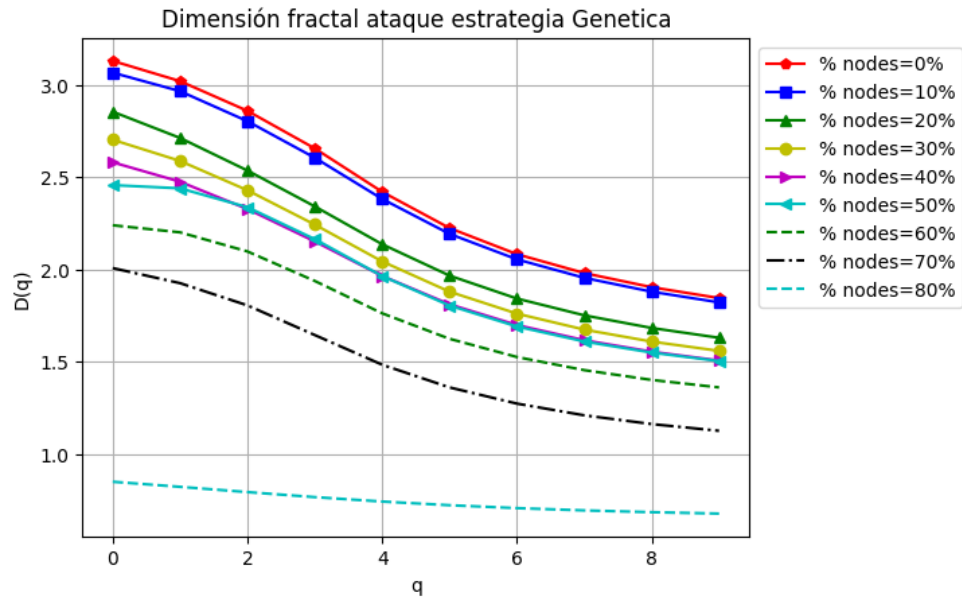


Figura 6.24: Análisis de multifractalidad de red (1,3)-flower a ataque por estrategia evolutiva

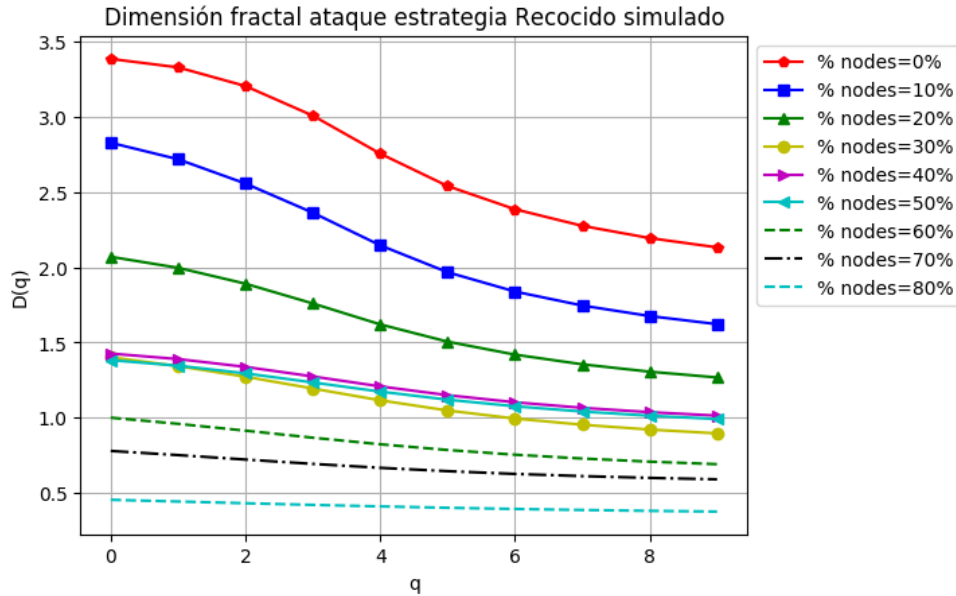


Figura 6.25: Análisis de multifractalidad de red (1,3)-flower a ataque por estrategia recocida simulada

6.1.2. Discusión de resultados

La relación entre multifractalidad y robustez está dada en cómo se comporta la dimensión fractal de la red a medida que pierde nodos. Se encuentra que entre más elevada sea la dimensión fractal la red es más robusta, esto se evidencia al cruzar la información, por ejemplo entre la variación del componente gigante 5.3 y de camino más cortos 5.4 sin importar la estrategia de ataque con respecto a la dimensión fractal estudiadas en la sección 6.1.1.

Así mismo, se encuentra que realizar el análisis de multifractalidad y robustez permite concluir que redes que presentan un gran número de componentes son las que mejor se adaptan ante fallas aleatorias, ya que las curvas de dimensión fractal presentan poca variación si se pierde un porcentaje pequeño de nodos (10 % o 20 %). En el caso de las redes que tienden a ser monofractales, son las que mejor se adaptan ante cualquier tipo de ataque, ya que el análisis multifractal permite intuir que su estructura interna es compacta, es decir toleran fallos de cualquier tipo.

La estrategia que más efectos produce son las de ataque por grado y centralidad, ya que estas tienden a desintegrar la red. El efecto es tan fuerte, que eliminados un 20 % a 30 % de los nodos, no es posible aplicar los algoritmos de análisis multifractal en algunas redes complejas, ello se evidencia, por ejemplo, en la figura 6.2.

Las estrategias de ataque por algoritmos de Inteligencia Artificial, no tienen un efecto tan evidente como lo tienen la centralidad o el grado. Sin embargo, muestran mayor efecto en las redes que las estrategia aleatoria.

Con respecto a las redes de mundo pequeño, se observa que estas conservan su estructura a medida que pierde nodos, lo que muestra que su estructura interna permanece intacta ante diferentes

estrategias de ataque.

Para las redes aleatorias, libres de escala, observadas y fractales, se encuentra que pierden su estructura a medida que pierden nodos, pasando de ser objetos multifractales a monofractales. Esto se debe a que estas presentan en mayor o menor medida hubs, los cuales estructuralmente son los centros de estructuras internas, que a medida que se pierden la red se va tornando más uniforme.

En todos los casos, la dimensión fractal tiende hacia 0 a medida que la red pierde nodos, lo que indica que el diámetro de las estructuras internas de la red tiende a cero.

6.2. Resumen y conclusiones del capítulo

En este capítulo se ha obtenido un resultado interesante de este trabajo, al observar cómo varía la estructura de una red a medida que pierde nodos. En todos los casos la dimensión fractal tiende a 1 a medida que se pierden nodos, lo que es un indicador que la red pierde sus estructuras internas.

Así mismo, por tipo de red se encuentra:

1. En las redes libres de escala, el ataque por grado es el que mayor tiene efecto, en la mayor parte de las pruebas con sólo perder 20 % de los nodos de mayor grado, la red pasa a ser monofractal
2. En las redes de mundo pequeño, la estructura de la red se ve mayormente afectada por la pérdida de los nodos con mayor centralidad, esto se debe a que son fundamentales para conservar que la distancias promedio entre los nodos sea baja

Se encuentra que en general, la estrategia más efectiva para afectar la estructura de las redes es el ataque por grado. Las redes más sensibles a este tipo de ataque son las redes libres de escala.

Capítulo 7

Implementación de la librería

7.1. Desarrollo de la librería

La librería es nombrada como **L.A.M.F.R.I.A** que significa: Librería para el análisis de multifractalidad y robustez con algoritmos de inteligencia artificial.

7.1.1. Procesamiento de redes con SNAP

Este trabajo es desarrollado con el apoyo de la librería SNAP de la Universidad de Standford <https://snap.stanford.edu>, la cual está escrita en C++ y se encuentra optimizada para redes de gran tamaño, gracias al uso de una representación basada en matrices dispersas.

Para la implementación de los algoritmos se utiliza el lenguaje de Programación Python y el entorno que ofrece esta librería para este lenguaje. La elección del lenguaje se debe a que actualmente es tendencia para el desarrollo de aplicaciones Inteligencia Artificial, el cual es el enfoque de este proyecto. Esto permite a futuro los algoritmos utilizados puedan ser integradas con mayor facilidad con otras estrategias de inteligencia artificial.

Selección de SNAP frente a otras librerías

En el mercado se destacan algunas librerías para el procesamiento de redes complejas como:

- NetworkX: Es una librería de Python comúnmente utilizadas, pero no es recomendada con redes de más de 100000 nodos
- IGraph: Utilizada en Python y R, está enfocada a estudio de propiedades y representación gráfica que a procesamiento de redes de gran tamaño.
- sigma.js, escrita en JavaScript, sin embargo, el lenguaje no es recomendado para procesamiento si no para visualización.

La selección de SNAP se debe a lo siguiente:

1. Está escrita en C++, lo que permite a la librería administrar directamente la memoria.
2. En Python se utiliza una interfaz a las funciones escritas en C++, por lo que el código se ejecuta directamente en la máquina, evitando el paso por el interpretador del lenguaje.
3. Se utilizan matrices dispersas para la representación de las redes, lo que permite reducir la complejidad computacional de operaciones y reducir el espacio requerido en memoria.
4. La librería está diseñada para el procesamiento de redes a gran escala y ha sido ampliamente probada en diferentes problemas y escenarios.
5. Se cuenta con una gran documentación y un foro en el cual la comunidad de desarrolladores da solución a problemas que se puedan presentar.
6. Es una librería independiente, lo que permite exportarla fácilmente a diferentes entornos computacionales.

7.1.2. Arquitectura de la librería

La librería desarrollada es un conjunto de funciones que corren sobre SNAP. En la figura 7.1 se provee un diagrama conceptual de la librería.

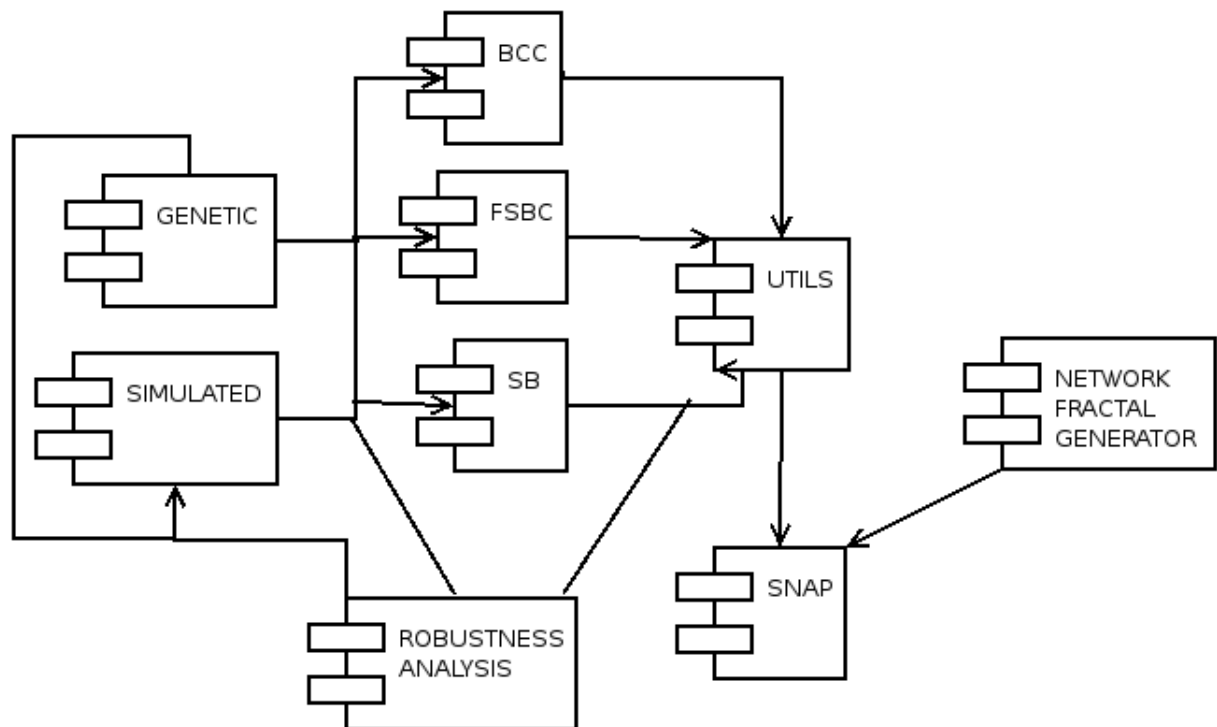


Figura 7.1: Diagrama conceptual de L.A.M.F.R.I.A

- SNAP: Provee las funciones de carga y búsqueda en redes complejas

- UTILS: Provee funciones generales para los algoritmos, como es el caso de cálculo de matrices de adyacencia o conectividad. Así mismo funciones para la carga y guardado de archivos
- BCC: Algoritmo de Box Compact Counting
- FSBC: Algoritmo de Fixed Size Box Counting
- DB: Algoritmo de SandBox
- Network Fractal Generator: Ofrece funciones para la creación de (2,2)-flower y (1,3)-flower de cualquier generación
- Genetic y Simulated ofrecen las estrategias de IA. Estas hacen parte de la ejecución de los algoritmos ya que calculan los centros de cajas y se ejecuta el algoritmo con esta configuración
- Robustness Analysis: Realiza el análisis de robustez, permite utilizar todos los algoritmos para el análisis de multifractalidad a medida que se pierden nodos. En las pruebas solo se usó el algoritmo SandBox por cuestiones de tiempo.

7.2. Requerimientos de la librería

7.2.1. Dependencias de software

Se requiere Python 2.7 y las siguientes librerías:

1. Matplotlib ≥ 2.0
2. Numpy ≥ 1.12
3. Snap ≥ 4.0 (Esta viene incluida en la librería y no se requiere instalar)

7.2.2. Recomendaciones de capacidad de cómputo

Se recomienda tener memoria suficiente para aplicar los algoritmos en redes grandes, ya que para optimizar se generaron matrices de distancias y estas pueden ocupar bastante espacio. Así mismo, en el proceso iterativo de los algoritmos de análisis multifractal requieren una gran cantidad de memoria por los cálculos que se realizan, se recomienda para redes de menos de 10000 nodos la siguiente configuración

1. Memoria: Al menos 4GB
2. Procesadores. Al menos 2 procesadores

Para redes más grandes, se requiere mayor capacidad, sin embargo, como no se realizaron pruebas con redes de más de 10000 nodos, no se especifica una configuración determinada.

7.3. Funciones provistas

7.3.1. Análisis multifractal

Las funciones provistas para el análisis multifractal son:

```
FSBCAlgorithm.FSBCAlgorithm.FSBCAlgorithm(g, minq, maxq, percentNodesT, centerNodes=
    array([], dtype=float64))
```

```
BCAlgorithm.BCAlgorithm.BCAlgorithm(g, minq, maxq, percentNodesT, centerNodes=array
    ([], dtype=float64))
```

```
SBCAlgorithm.SBCAlgorithm.SBCAlgorithm(g, minq, maxq, percentSandBox, repetitions,
    centerNodes=array([], dtype=float64))
```

Estas funciones permiten ingresar la configuración de los diferentes algoritmos.

- maxq y minq, son los valores de q. El rango debe tener 0 incluido
- g es el grafo en formato SNAP
- percentNodesT, percentSandBoxes, son el número de conjuntos que serán seleccionados como centros
- repetitions, para SandBox, al ser un algoritmo de elección aleatoria de centros debe repetirse y tomar promedios de los resultados
- centernodes, utilizados por las estrategias de Inteligencia Artificial que proveen al algoritmo centros ya calculados, por defecto es vacío

7.3.2. Análisis de robustez

```
robustness.robustness.robustness_analysis(graph, typeRemoval, minq, maxq,
    percentSandBox, repetitions, temperature=0, sizePopulation=0, iterationsGenetic
    =0, percentCrossOver=0, percentMutation=0, degreeOfBoring=0)
```

- maxq y minq, son los valores de q. El rango debe tener 0 incluido
- typeRemoval, es el tipo de ataque, las opciones son:
 1. Random: para aleatorio
 2. Degree: para grado
 3. Centrality: para centralidad
 4. Genetic: para genético
 5. Simulated: para recocido simulado

- percentSandBox, parámetro del algoritmo SandBox
- repetitions, parámetro del algoritmo SanBox
- temperature, parámetro del algoritmo de recocido simulado
- sizePopulation, iterationsGenetic, percentCrossOver, percentMutation, degreeOfBoring. Parámetros del algoritmo evolutivo

7.3.3. Algoritmos evolutivos

```
Genetic.Genetic.Genetic(g, minq, maxq, sizePopulation, iterations, percentCrossOver,
    percentMutation, degreeOfBoring, typeAlgorithm)
```

- percentCorssOver: Porcentaje de cruce
- percentMutacion: Porcentaje de mutación
- degreeOfBoring: Número de generaciones en las cuales se detiene el algoritmo si el mejor individuo no mejora
- typeAlgorithm. Tipo de algoritmo que se aplica el análisis multifractal, puede ser:
 1. SB para SandBox
 2. BC Para Box Compact Counting
 3. FSBC para Fixed Size Box Counting

7.3.4. Algoritmos de recocido simulado

```
SimulatedAnnealing.SimulatedAnnealing.SA(g, minq, maxq, percentNodes, sizePopulation,
    Kmax, typeAlgorithm)
```

Los parámetros de este algoritmo han sido explicados anteriormente. El funcionamiento es similar al evolutivo, ya que se calculan los centros de las cajas y estas son pasadas a alguno de los algoritmos de análisis multifractal.

7.4. Documentación de la librería

7.4.1. Desarrollo de la documentación

Para esto se utiliza el paquete sphinx de Python <http://www.sphinx-doc.org> y el tema Read The Docs <https://readthedocs.org>.

La selección de estas herramienta se debe a que permite generar documentación automáticamente a través de comentarios en el código, lo que facilita en gran medida esta tarea

7.4.2. Ejemplos de la documentación

La pagina inicial de la librería puede verse en la figura 7.2

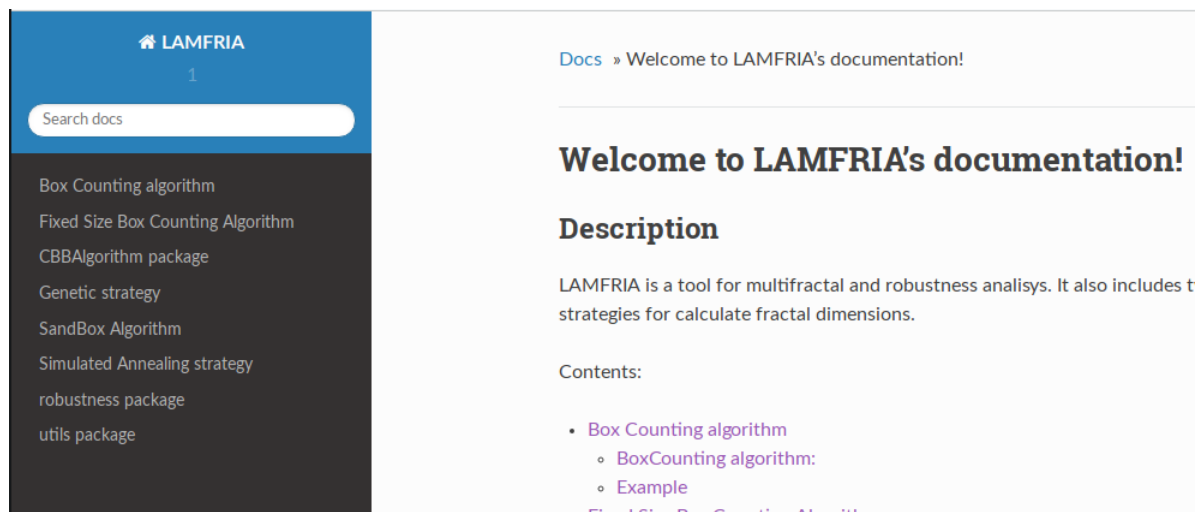


Figura 7.2: Pagina inicial de la documentación de la librería

Al hacer clic en alguna de las funciones del menú izquierdo, se puede ampliar la información:

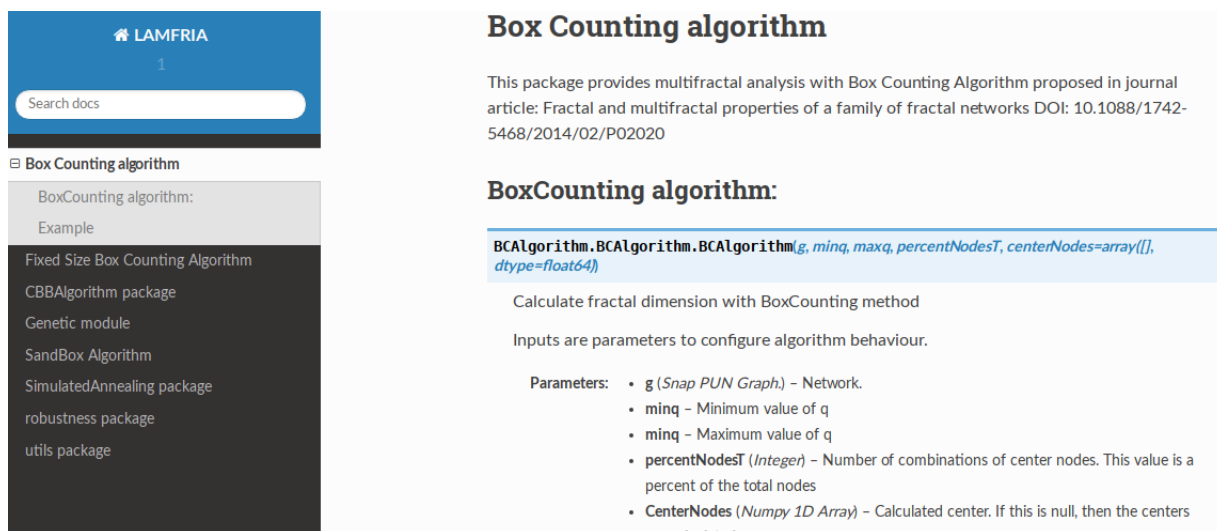


Figura 7.3: Información específica sobre una función de la librería

Así mismo, se puede hacer búsqueda de información dentro de la documentación, en este caso la palabra *genetic*

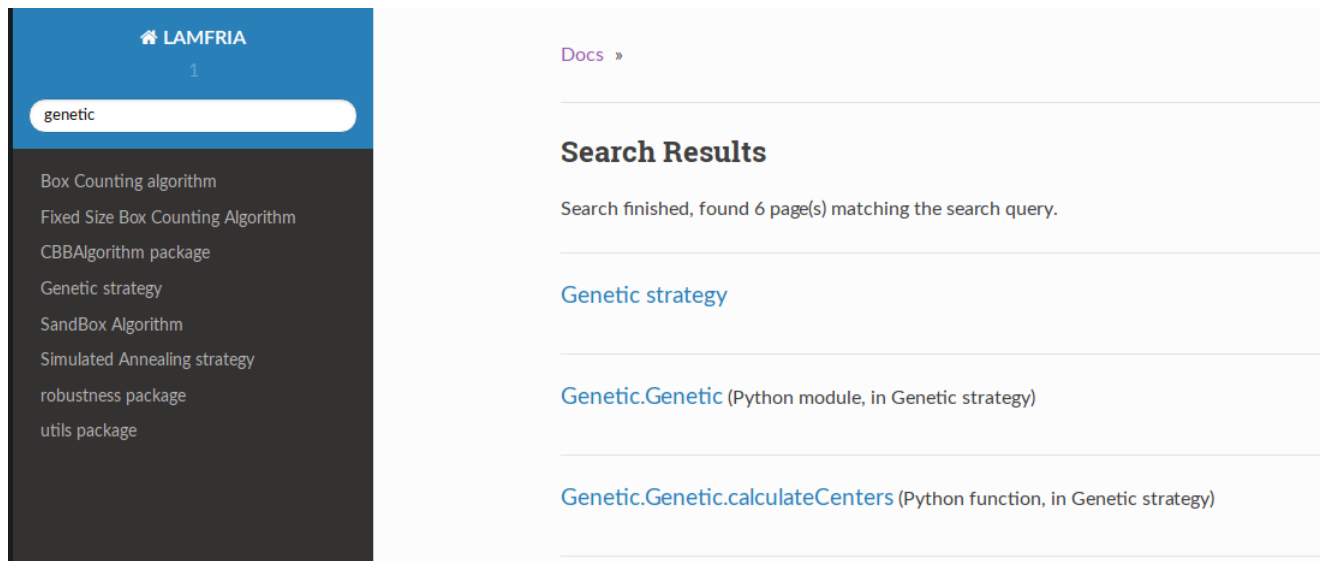


Figura 7.4: Resultado búsqueda información de la documentación en de la librería

7.5. Resumen y conclusiones del capítulo

La selección de SNAP se debe a que utiliza una representación interna de matrices dispersas, la cual permite realizar operaciones sobre las redes con menor costo computacional que otras opciones disponibles. Además al estar escrita en C++, elimina las limitaciones que tienen las librerías que requieren ser gestionadas por un interpretador.

Sin embargo, los algoritmos de análisis multifractal requieren una gran capacidad de memoria y cómputo, por lo que para redes grandes se debe estudiar la posibilidad de trabajar en equipos con gran cantidad memoria y procesadores. Así mismo, en la posibilidad de distribuir sus tareas.

La documentación provista por la librería Sphinx permite a un usuario de librería explorar las diferentes funciones que se proveen y realizar búsqueda. Así mismo, la flexibilidad del generador de documentación permite ejemplos de uso de la librería.

Capítulo 8

Conclusiones y trabajos futuros

8.1. Conclusiones

1. La topología de una red puede describirse con base a la diferencia entre la máxima dimensión fractal y la mínima dimensión fractal. Si esta medida crece indica la existencia de diferentes componentes en la red interconectados entre sí por un conjunto de nodos clave. Por el contrario, si esta disminuye, la red tiende a tener una estructura compacta o monofractal.
2. Las redes de mundo pequeño muestran una tendencia a ser monofractales y conservan su estructura a media que son atacadas, esto es debido a que son compactas. En consecuencia, se puede utilizar el análisis de multifractal y de robustez para caracterizar estas redes.
3. Las relación entre multifractalidad y robustez está dada que a medida una red pierde nodos, la dimensión fractal de las estructuras presentes en la red va tendiendo a 0 y a una característica monofractal. Esto indica una descomposición de las estructuras de la red; siendo el caso más relevante el de las redes libres de escala a medida que pierden los nodos con mayor grado.
4. Las diferentes estrategias de análisis multifractal muestran variaciones en el cálculo de las dimensiones fractales, esto se debe a que los centros se establecen aleatoriamente, lo que puede producir diferentes resultado de acuerdo a la red. Sin embargo, este análisis permite caracterizar las redes de acuerdo a su estructura, ya que entre más pronunciada sea la curva de la dimensión fractal, más diferencias estructurales existen dentro de la red.
5. Se encuentra que realizar el análisis de multifractalidad y robustez permite concluir que redes que presentan un gran número de componentes son las más que se adaptan ante fallas aleatorias, ya que las curvas de dimensión fractal presentan poca variación si se pierde un porcentaje pequeño de nodos (10 % o 20 %) aleatoriamente. En el caso de las redes que tienden a ser monofractales, lo que indica que se adaptan ante cualquier tipo de ataque, en otras palabras, toleran fallos de cualquier tipo, sin embargo, se debe realizar un protocolo de pruebas más rigurosas para dar un soporte a este resultado.
6. En general, las estrategias de inteligencia artificial permiten estimar centros de cajas de las redes, lo que permite mejorar el tiempo de cómputo de los algoritmos BCC y BCFS, sin embargo, se encuentra problemas de precisión, ya que la elección de los centros de las cajas es

un proceso de búsqueda que depende del número de nodos y la forma en que se interconectan. Al aumentar el número de iteraciones de los algoritmos se mejora la precisión, sin embargo, los cálculos podrían tomar una gran cantidad de tiempo.

7. Las estrategias de inteligencia artificial dan un indicio de que se obtienen mejores tiempos de ejecución de los algoritmos BCC y BCFS en el caso del análisis multifractal. Sin embargo, para el caso de SB los resultados no son concluyentes, debido a que las redes estudiadas no son lo suficientemente grandes para llegar a un resultado concreto.
8. Para el análisis de la robustez, los algoritmos de IA no mostraron ser una estrategia de ataque que produzca más efectos en la estructura de la red que las basadas en el grado y la centralidad de los nodos. Esto se debe a que en las redes estudiadas los nodos que tienen mayor centralidad o grado, son claves para la transmisión de información dentro de la red.
9. El uso de librerías con representaciones comprimidas de las redes, provee un marco de trabajo para el desarrollo de algoritmos para el análisis de multifractalidad, ya que al reducirse la complejidad espacial y computacional se realizan las búsquedas dentro de las redes con mayor rapidez que con otras opciones disponibles.
10. La herramienta desarrollada en este trabajo ofrece la posibilidad de realizar un análisis de multifractalidad y robustez de cualquier red compleja. Frente a las herramientas disponibles en la literatura se destaca que es libre, construida en un lenguaje flexible y común. Así mismo, cuenta con la optimización de algoritmos de inteligencia artificial, lo que permite realizar menos cálculos que los métodos de cubrimiento de cajas BC y BCFS.
11. Las heurísticas desarrolladas en este trabajo deben ser mejoradas, sin embargo, se muestra que es posible diseñarlas para trabajar con redes complejas. La evolución de la función de evaluación muestra que es necesario realizar un proceso más amplio de exploración de opciones para mejorar la convergencia de los algoritmos de inteligencia artificial.

8.2. Trabajos futuros

- Debe establecerse un método analítico para validar el análisis multifractal de los diferentes métodos, como se vio en las diferentes pruebas los resultados pueden variar significativamente
- El algoritmo de SandBox propuesto por Liu[32], es el que mejor resultados presenta en las pruebas realizadas, sin embargo, no se puede asegurar se tenga un mejor rendimiento que las técnicas de Inteligencia Artificial propuestas, por lo que se requiere un estudio a profundidad de este tema.
- Se debe proponer una estrategia para generar una heurística que permita seleccionar los centros de las cajas, ya que la propuesta de este trabajo depende si la red tiene estructura libre de escala, ya que se buscan nodos altamente conectados.
- A partir del trabajo realizado, una idea interesante a explorar son los generadores de redes monofractales y robustas, los cuales deben conservar su estructura ante diferentes estrategias de ataque. Esto permite generar un marco de trabajo para el diseño de redes con gran tolerancia a fallos.
- En el caso de las redes libres de escala debe realizarse un estudio variando el término de la ley de potencia, que dado el gran tiempo que se requiere para este análisis, se debe desarrollar en trabajos posteriores.
- Las pruebas de este trabajo deben ejecutarse en un entorno con mayor capacidad de cómputo, ya que los resultados encontrados en algunos casos no corresponden a lo esperado, ya que los algoritmos requieren iterar una gran cantidad de veces para obtener resultados cercanos a los algoritmos presentes en la literatura.
- Se deben proponer estrategias de distribución de los cálculos, ya que el análisis multifractal depende de la configuración de cajas a un radio dado r , este valor puede ser distribuido, ya que no hay dependencias entre ellos. Así mismo, para la robustez se puede probar con diferentes porcentajes de pérdidas de nodos y realizar el respectivo análisis multifractal.

Capítulo 9

Bibliografía

- [1] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Review*, vol. 45, pp. 167–256, jan 2003.
- [2] Y. Lin, B. Wu, Z. Zhang, and G. Chen, “Counting spanning trees in self-similar networks by evaluating determinants,” *Journal of Mathematical Physics*, vol. 52, p. 113303, nov 2011.
- [3] D.-L. Wang, Z.-G. Yu, and V. Anh, “Multifractal analysis of complex networks,” *Chinese Physics B*, vol. 21, p. 080504, aug 2012.
- [4] L. da F. Costa, F. A. Rodrigues, and A. S. Cristino, “Complex networks: the key to systems biology,” *Genetics and Molecular Biology*, vol. 31, no. 3, pp. 591–601, 2008.
- [5] X. Wu, H. Dong, C. K. Tse, I. W. Ho, and F. C. Lau, “Analysis of metro network performance from a complex network perspective,” *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 553–563, feb 2018.
- [6] W.-M. Huang, L.-J. Zhang, X.-J. Xu, and X. Fu, “Contagion on complex networks with persuasion,” *Scientific Reports*, vol. 6, mar 2016.
- [7] S. Heymann and B. L. Grand, “Visual analysis of complex networks for business intelligence with gephi,” in *2013 17th International Conference on Information Visualisation*, pp. 307–312, July 2013.
- [8] Z. Qing and Y. Hong, “Citegraph: A citation network system for medline articles and analysis,” *Studies in Health Technology and Informatics*, vol. 192, no. MEDINFO 2013, p. 832–836, 2013.
- [9] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.
- [10] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of Modern Physics*, vol. 74, pp. 47–97, jan 2002.
- [11] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, “Classes of small-world networks,” *Proceedings of the National Academy of Sciences*, vol. 97, pp. 11149–11152, sep 2000.
- [12] C. Song, S. Havlin, and H. A. Makse, “Self-similarity of complex networks,” *Nature*, vol. 433, pp. 392–395, jan 2005.

- [13] B.-G. Li, Z.-G. Yu, and Y. Zhou, “Fractal and multifractal properties of a family of fractal networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, p. P02020, feb 2014.
- [14] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, “Mitigation of malicious attacks on networks,” *Proceedings of the National Academy of Sciences*, vol. 108, pp. 3838–3841, feb 2011.
- [15] C. Song, L. K. Gallos, S. Havlin, and H. A. Makse, “How to calculate the fractal dimension of a complex network: the box covering algorithm,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2007, pp. P03006–P03006, mar 2007.
- [16] F. Shuhei and Y. Kousuke, “Multifractality of complex networks,” *Phys Rev E*, vol. 036118, p. 24, 2011.
- [17] J. Martin-Hernandez, *Measuring Robustness*. Doctoral thesis, Technische Universiteit Delft Printed, 2013.
- [18] W. M. Macek, “Multifractality and intermittency in the solar wind,” *Nonlinear Processes in Geophysics*, vol. 14, pp. 695–700, 2007.
- [19] P. Caraiiani, “Evidence of Multifractality from Emerging European Stock Markets.,” *PLoS ONE*, vol. 7, no. 7, pp. 1–9, 2012.
- [20] A.-L. Barabasi, *Linked the new science of networks*. Cambridge, Mass.: Perseus Pub., 2002.
- [21] P. Erdős and A. Rényi, “On random graphs, I,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.
- [22] A.-L. Barabási, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, oct 1999.
- [23] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, pp. 440–442, jun 1998.
- [24] K. H. Rosen, *Discrete Mathematics and Its Applications*. McGraw-Hill Higher Education, 5th ed., 2002.
- [25] E. Estrada, *The Structure of Complex Networks*. Oxford University Press, oct 2011.
- [26] J. S. Kim, K.-I. Goh, G. Salvi, E. Oh, B. Kahng, and D. Kim, “Fractality in complex networks: Critical and supercritical skeletons,” *Physical Review E*, vol. 75, jan 2007.
- [27] J. S. Kim, K.-I. Goh, B. Kahng, and D. Kim, “A box-covering algorithm for fractal scaling in scale-free networks,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, p. 026116, jun 2007.
- [28] W.-X. Zhou, Z.-Q. Jiang, and D. Sornette, “Exploring self-similarity of complex cellular networks: The edge-covering method with simulated annealing and log-periodic sampling,” *Physica A: Statistical Mechanics and its Applications*, vol. 375, pp. 741–752, mar 2007.
- [29] C.-Y. Lee and S. Jung, “Statistical self-similar properties of complex networks,” *Physical Review E*, vol. 73, jun 2006.

- [30] J.-L. Liu, Z.-G. Yu, and V. Anh, “Topological properties and fractal analysis of a recurrence network constructed from fractional brownian motions,” *Physical Review E*, vol. 89, mar 2014.
- [31] S. Furuya and K. Yakubo, “Multifractality of complex networks,” *Physical Review E*, vol. 84, sep 2011.
- [32] J. L. Liu, Z. G. Yu, and V. Anh, “Determination of multifractal dimensions of complex networks by means of the sandbox algorithm,” *Chaos*, vol. 25, no. 2, 2015.
- [33] T. Tél, Á. Fülöp, and T. Vicsek, “Determination of fractal dimensions for geometrical multifractals,” *Physica A: Statistical Mechanics and its Applications*, vol. 159, pp. 155–166, aug 1989.
- [34] M. Barat, Z. Joveini, and J. Sadri, “Application of Fractal Theory on Motifs Counting in Biological Networks,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5963, no. c, pp. 1–12, 2016.
- [35] D. Wang and Y. Wang, “Multifractal analysis on gene and ppi networks,” *Nature phenomena and innovative engineering*, vol. 18, no. 8, pp. 353–357, 2014.
- [36] P. A. Moreno, P. E. Vélez, E. Martínez, L. E. Garreta, N. Díaz, S. Amador, I. Tischer, J. M. Gutiérrez, A. K. Naik, F. Tobar, and F. García, “The human genome: A multifractal analysis,” *BMC Genomics*, vol. 12, no. 1, p. 506, 2011.
- [37] F. M. Siokis, “European economies in crisis: A multifractal analysis of disruptive economic events and the effects of financial assistance,” *Physica A: Statistical Mechanics and its Applications*, vol. 395, pp. 283–292, feb 2014.
- [38] S. Rendón de la Torre, J. Kalda, R. Kitt, and J. Engelbrecht, “Fractal and multifractal analysis of complex networks: Estonian network of payments,” *European Physical Journal B*, vol. 90, no. 12, pp. 1–13, 2017.
- [39] T. C. Roeske, D. Kelty-Stephen, and S. Wallot, “Multifractal analysis reveals music-like dynamic structure in songbird rhythms,” *Scientific Reports*, vol. 8, no. 1, p. 4570, 2018.
- [40] T. D. Dang, S. Molnár, and I. Maricza, “Performance analysis of multifractal network traffic,” *European Transactions on Telecommunications*, vol. 15, pp. 63–78, mar 2004.
- [41] R. Vojak, M. Danech-Pajouh, and J. L. Véhel, “Multifractal description of road traffic structure,” *IFAC Proceedings Volumes*, vol. 27, pp. 877–882, aug 1994.
- [42] W. Zheng, Z. Zhang, and Y. Deng, “Multifractal analysis of mobile social networks,” *EPL (Europhysics Letters)*, vol. 119, no. 6, p. 66006, 2017.
- [43] M. Inc, “Matlab,” 2018. <https://es.mathworks.com/products/matlab.html>, Consultado 2018-05-27.
- [44] C. Santana, “Multifractal.jl,” 2018. <https://github.com/cndesantana/Multifractal.jl>, Consultado 2018-05-27.
- [45] J. Grazzini, “Gjacopo/multifractal: Final release of ”multifractal” software tools,” 2017.

- [46] L. A. Saravia, “Multifractal estimation using a standard box-counting algorithm,” 2018. <https://github.com/laravia/mfsba>, Consultado 2018-05-27.
- [47] L. A. Saravia, “mfSBA: Multifractal analysis of spatial patterns in ecological communities,” *F1000Research*, apr 2014.
- [48] M. Inc, “Multifractal analysis toolbox,” 2018. <https://es.mathworks.com/products/matlab.html>, Consultado 2018-05-27.
- [49] G. L. H. J. I. P. M. R. M. J. M. G. P. C.-K. S. H. Goldberger AL, Amaral LAN, “Multiscale multifractal analysis (mma),” 2015. <https://physionet.org/physiotools/mma/>, Consultado 2018-05-27.
- [50] J. Gierałowski, J. J. Żebrowski, and R. Baranowski, “Multiscale multifractal analysis of heart rate variability recordings with a large number of occurrences of arrhythmia,” *Physical Review E*, vol. 85, feb 2012.
- [51] P. Jurica, “Introduction to mdfa in python,” 2018. <http://bsp.brain.riken.jp/juricap/mdfa/mdfaintro.html>, Consultado 2018-05-27.
- [52] P. Jurica, “Multifractal analysis for all,” *Frontiers in Physiology*, vol. 6, feb 2015.
- [53] H. Makse, “Implementation in python of analysis multifractal,” 2007. <http://www-levich.engr.cuny.cuny.edu/webpage/hmakse/software-and-data/>, Consultado 2018-05-27.
- [54] R. Albert, H. Jeong, and A.-L. Barabási, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, pp. 378–382, jul 2000.
- [55] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, “Resilience of the internet to random breakdowns,” *Physical Review Letters*, vol. 85, pp. 4626–4628, nov 2000.
- [56] R. Cohen, S. Havlin, and D. ben Avraham, “Efficient immunization strategies for computer networks and populations,” *Physical Review Letters*, vol. 91, dec 2003.
- [57] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.
- [58] M. Fiedler, “Algebraic Connectivity of Graphs,” *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.
- [59] D. Bauer, F. Boesch, C. Suffel, and R. Tindell, “Combinatorial optimization problems in the analysis and design of probabilistic networks,” *Networks*, vol. 15, no. 2, pp. 257–271, 1985.
- [60] F. Harary, “Conditional connectivity,” *Networks*, vol. 13, no. 3, pp. 347–357, 1983.
- [61] B. Mohar, “Isoperimetric numbers of graphs,” *Journal of Combinatorial Theory, Series B*, vol. 47, pp. 274–291, dec 1989.
- [62] R.-H. Li, J. X. Yu, X. Huang, H. Cheng, and Z. Shang, “Measuring robustness of complex networks under MVC attack,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM Press, 2012.
- [63] A. Li, Q. Hu, J. Liu, and Y. Pan, “Resistance and security index of networks: Structural information perspective of network security,” *Scientific Reports*, vol. 6, jun 2016.

- [64] S. Pahwa, C. Scoglio, and A. Scala, “Abruptness of cascade failures in power grids,” *Scientific Reports*, vol. 4, jan 2014.
- [65] G. F. Chami, S. E. Ahnert, N. B. Kabatereine, and E. M. Tukahebwa, “Social network fragmentation and community health,” *Proceedings of the National Academy of Sciences*, vol. 114, pp. E7425–E7431, jul 2017.
- [66] S. Iyer, “Attack robustness and centrality of complex networks,” 2016. <https://github.com/swamiiyer/robustness>, Consultado 2018-05-27.
- [67] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, “Attack robustness and centrality of complex networks,” *PLoS ONE*, vol. 8, p. e59613, apr 2013.
- [68] L. Maccari, “A simple library to manipulate netjson graphs in order to make some robustness analysis,” 2018. https://github.com/netCommonsEU/netjson_robustness, Consultado 2018-05-27.
- [69] C. G. Watson, “Graph theory analysis of brain mri data,” 2015. <https://github.com/cran/brainGraph>, Consultado 2018-05-27.
- [70] F. Khalid, “Python framework for creating and analyzing genotype networks from data,” 2016. <https://github.com/fkhalid/genonets>, Consultado 2018-05-27.
- [71] B. A. Jafino, “Transport network criticality analysis,” 2017. <https://github.com/bramkaarga/transcrit>, Consultado 2018-05-27.
- [72] V. Nannen, “Measuring freight transport network criticality,” Master’s thesis, Delft University of Technology, Netherlands, 2017.
- [73] W. Li, Y. Xu, J. Yang, and Z. Tang, “Finding structural patterns in complex networks,” in *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, IEEE, oct 2012.
- [74] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 4–37, Jan. 2000.
- [75] S. Wuchty and E. Almaas, “Peeling the yeast protein network,” *PROTEOMICS*, vol. 5, pp. 444–449, feb 2005.
- [76] M. Kirley and R. Stewart, “Multiobjective evolutionary algorithms on complex networks,” in *Evolutionary Multi-Criterion Optimization* (S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, eds.), (Berlin, Heidelberg), pp. 81–95, Springer Berlin Heidelberg, 2007.
- [77] S. Chien, F. Fisher, H. Mortensen, E. Lo, R. Greeley, A. Govindjee, T. Estlin, and X. Wang, “Using artificial intelligence planning techniques to automatically reconfigure software modules,” in *Tasks and Methods in Applied Artificial Intelligence*, pp. 415–426, Springer Berlin Heidelberg, 1998.
- [78] J. Liu and T. Liu, “Detecting community structure in complex networks using simulated annealing with -means algorithms,” *Physica A: Statistical Mechanics and its Applications*, vol. 389, pp. 2300–2309, jun 2010.

- [79] D. Chen, Y. Fu, and M. Shang, “A fast and efficient heuristic algorithm for detecting community structures in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 388, pp. 2741–2749, jul 2009.
- [80] H. D. Rozenfeld and D. ben Avraham, “Percolation in hierarchical scale-free nets,” *Physical Review E*, vol. 75, jun 2007.
- [81] H. D. Rozenfeld, S. Havlin, and D. ben Avraham, “Fractal and transfractal recursive scale-free nets,” *New Journal of Physics*, vol. 9, pp. 175–175, jun 2007.
- [82] T. C. Halsey, M. H. Jensen, L. P. Kadanoff, I. Procaccia, and B. I. Shraiman, “Fractal measures and their singularities: The characterization of strange sets,” *Physical Review A*, vol. 33, pp. 1141–1151, feb 1986.
- [83] Z.-G. Yu, V. Anh, and K.-S. Lau, “Multifractal and correlation analyses of protein sequences from complete genomes,” *Physical Review E*, vol. 68, aug 2003.
- [84] A. Percus, *Computational Complexity and Statistical Physics (Santa Fe Institute Studies on the Sciences of Complexity)*. Oxford University Press, 2006.
- [85] M. Galipienso, M. Quevedo, O. Pardo, F. Ruiz, and M. Ortega, *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Ediciones Paraninfo. S.A., 2003.
- [86] TyersLab.com, “Biogrid file repository,” 2018. <http://thebiogrid.org/download.php>, Consultado 2018-05-12.
- [87] UCLA, “Database of interacting proteins,” 2018. <http://dip.doe-mbi.ucla.edu/dip/Download.cgi>, Consultado 2018-05-12.
- [88] N. R. for Network Biology, “Cytoscape: Network data integration, analysis, and visualization in a box,” 2018. <http://www.cytoscape.org>, Consultado 2018-05-12.
- [89] Gephi.org, “The open graph viz platform,” 2018. <https://gephi.org>, Consultado 2018-05-12.

Anexos

Anexo A: Descripción de redes de prueba

Para este trabajo se han utilizado las redes de prueba del artículo de Wang[35]. Estas son descritas a continuación en la tabla 9.1.

Tipo	Número de vértices	Número de aristas	Diámetro	Diámetro camino promedio más corto	Descripción
Libre de escala	2000	1999	19	10.14	Generada con modelo Barabasi-Albert. Parámetro de ley de potencia 1.94.
Libre de escala	4000	3999	22	11.91	Generada con modelo Barabasi-Albert. Parámetro de ley de potencia 2.04.
Libre de escala	8000	7999	23	10.95	Generada con modelo Barabasi-Albert. Parámetro de ley de potencia 1.98.
Mundo pequeño	5000	24997	11	7.87	Generada con modelo Watts-Strogatz, con probabilidad de reconexión del 5 %
Mundo pequeño	5000	25000	9	6.4	Generada con modelo Watts-Strogatz, con probabilidad de reconexión del 10 %
Mundo pequeño	5000	24996	7	5.43	Generada con modelo Watts-Strogatz, con probabilidad de reconexión del 20 %

Tipo	Número de vértices	Número de aristas	Diámetro	Diámetro camino promedio más corto	Descripción
Aleatoria	1991	5939	8	4.95	Generada con modelo Erdos-Renyi
Aleatoria	3373	5978	8	7.69	Generada con modelo Erdos-Renyi
Aleatoria	5620	8804	14	8.84	Generada con modelo Erdos-Renyi
Biológica observada	3954	7810	12	5.07	Red de iteraciones de proteínas de la bacteria intestinal <i>C.elegans</i> obtenida desde Biogrid[86]
Biológica observada	5176	22624	9	4.75	Red de de iteraciones de proteínas del hongo <i>cerevisiae</i> obtenida desde DIP [87]
Biológica observada	2940	11627	10	4.93	Red de las iteraciones de proteínas de <i>Ecoli</i> obtenida desde DIP[87]
Fractal	2732	4096	64	57.15	Séptima generación de (2,2)-flower
Fractal	2732	4096	12	7.87	Séptima generación de (1,3)-flower

Tabla 9.1: Redes utilizadas en el proyecto

Las redes reales fueron procesadas con el software Cytoscape[88] para generar la red asociada no dirigida y posteriormente con Gephi[89] para transformarlas en formato PAJEK.

Anexo B: Descripción de protocolos de prueba

Las pruebas son ejecutadas bajo la siguiente configuración:

1. Algoritmo SandBox: 40 % de los nodos como caja y, 250 repeticiones
2. Algoritmo Box Counting Fixed Size: Número de permutaciones 200 % del número de nodos
3. Algoritmo Box Compact Counting: Número de permutaciones 200 % del número de nodos
4. Estrategias genéticas:
 - Porcentaje de nueva población 30 %
 - Tamaño población: 200
 - Número máximo de iteraciones: 250
 - Grado de aburrimiento: 20 iteraciones
 - Probabilidad de mutación: 5 %
5. Estrategia recocido simulado: Temperatura inicial: 3000, porcentaje de nodos 40 %

Las pruebas son realizadas en la infraestructura de Google Cloud `cloud.google.com`, con la siguiente configuración:

1. Memoria: 4.75GB
2. Disco duro: 8GB
3. Sistema operativo: Debian 9
4. 2 procesadores Intel Xeon 2.3GH

Anexo C: Resultados

Resultados análisis de multifractalidad

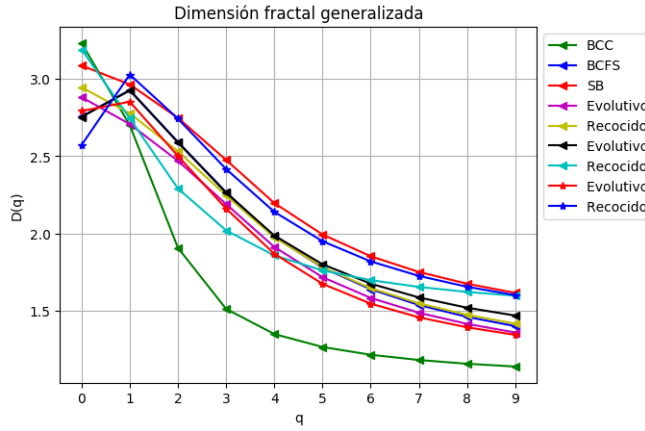


Figura 9.1: Análisis multifractal red libre de escala 2000 nodos

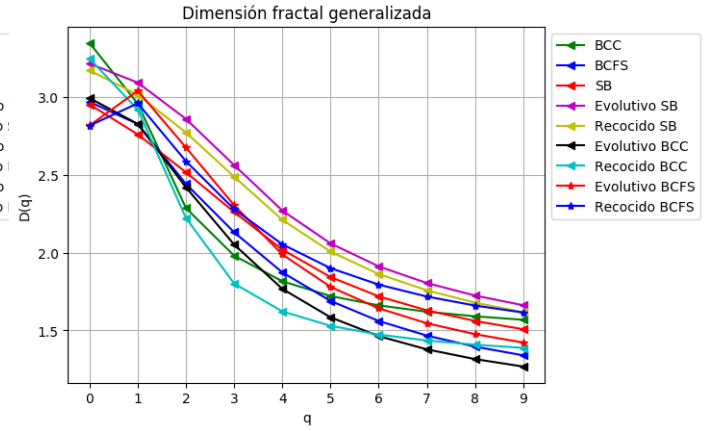


Figura 9.2: Análisis multifractal red libre de escala 4000 nodos

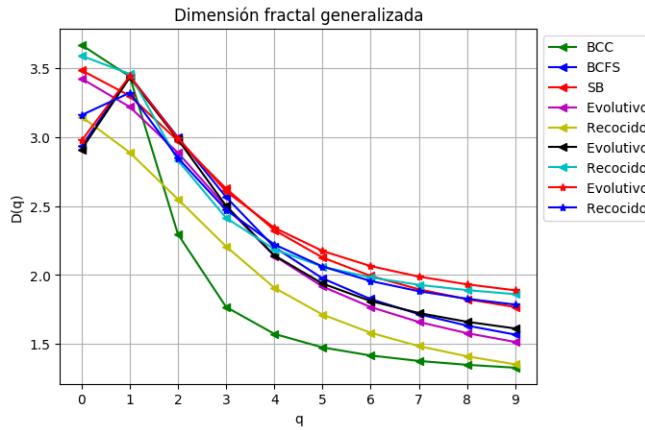


Figura 9.3: Análisis multifractal red libre de escala 8000 nodos

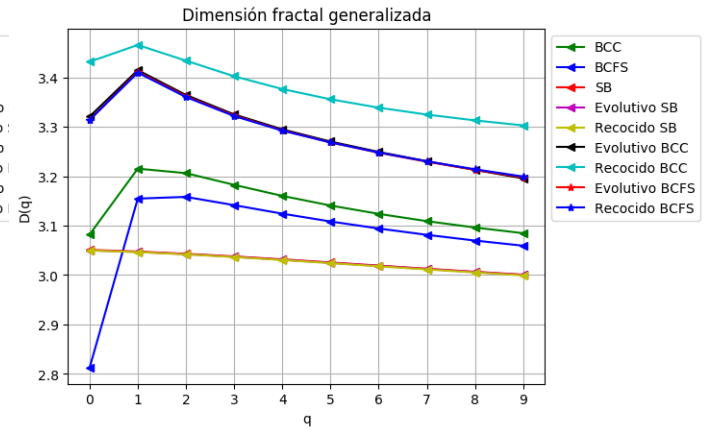


Figura 9.4: Análisis multifractal Mundo pequeño $p=5\%$

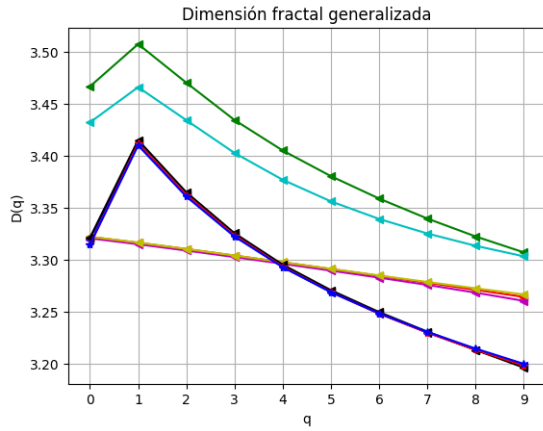


Figura 9.5: Análisis multifractal Mundo pequeño $p=10\%$

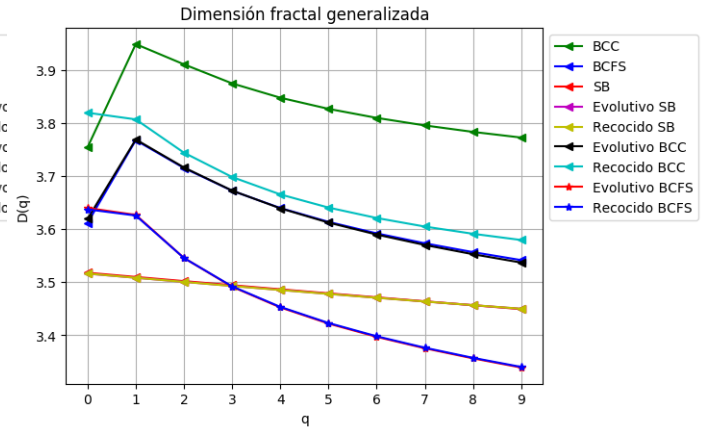


Figura 9.6: Análisis multifractal Mundo pequeño $p=20\%$

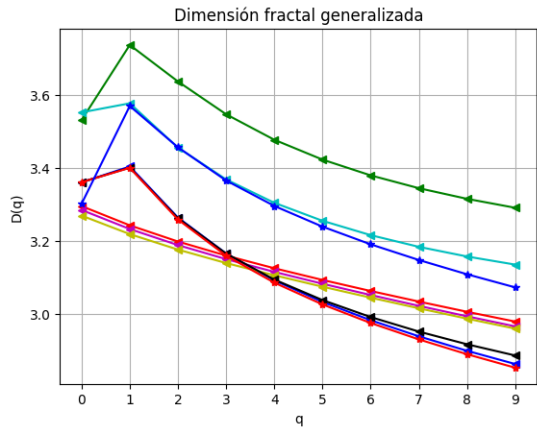


Figura 9.7: Análisis multifractal red aleatoria 1991 nodos

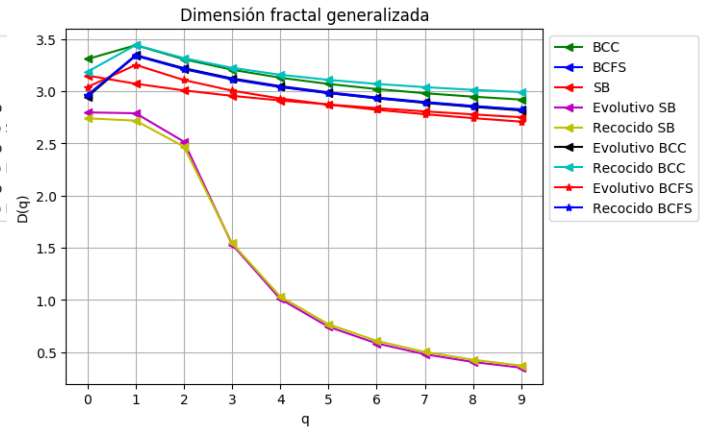


Figura 9.8: Análisis multifractal red aleatoria 3737 nodos

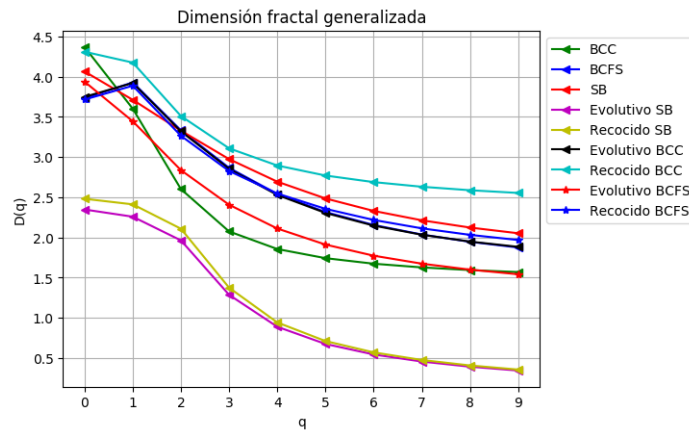


Figura 9.11: Análisis multifractal red observada *Cerevisiae*

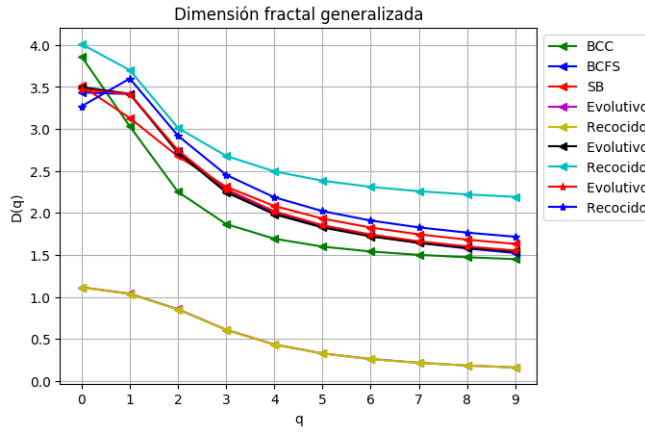


Figura 9.9: Análisis multifractal red observada ECOLI

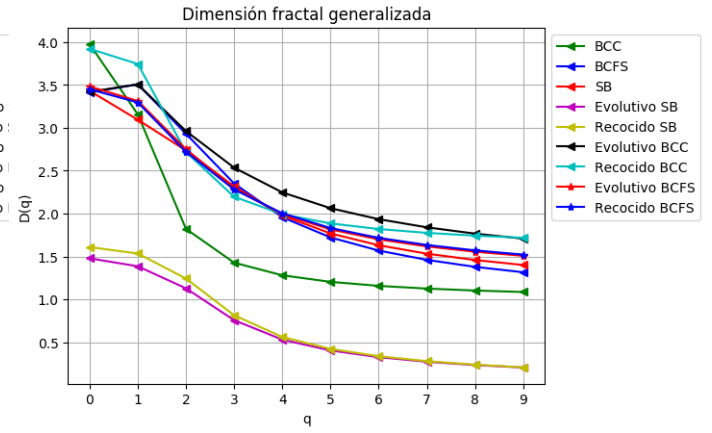


Figura 9.10: Análisis multifractal red observada Celengs

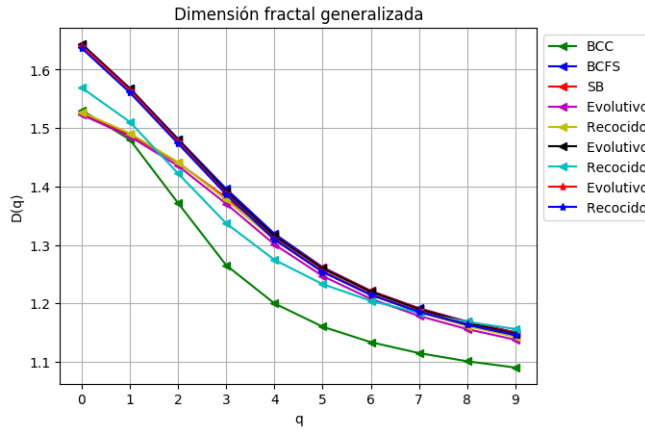


Figura 9.12: Análisis multifractal red fractal (2,2)-flower

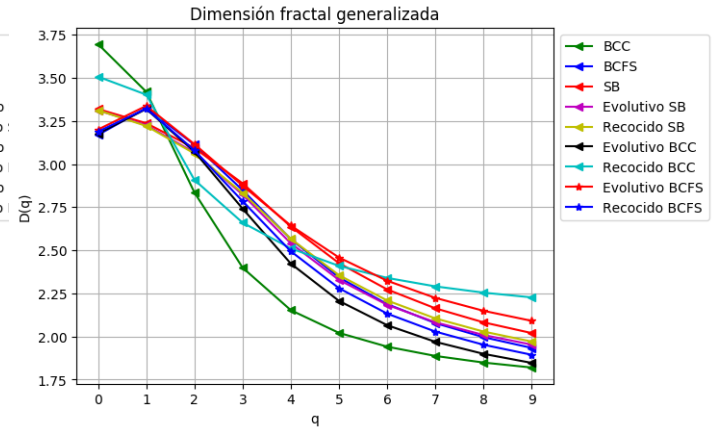


Figura 9.13: Análisis multifractal red observada (1,3)-flower

Resultados análisis de robustez

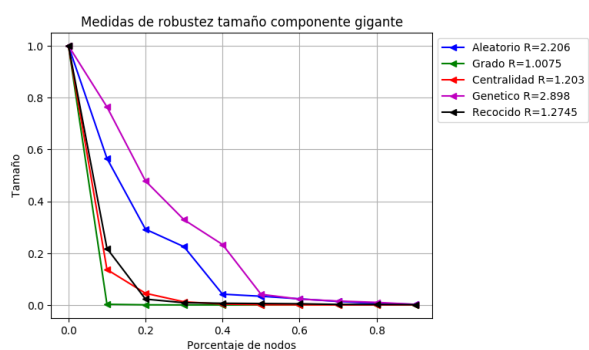


Figura 9.14: Análisis de robustez de red libre de escala de 2000 nodos por tamaño de componente gigante

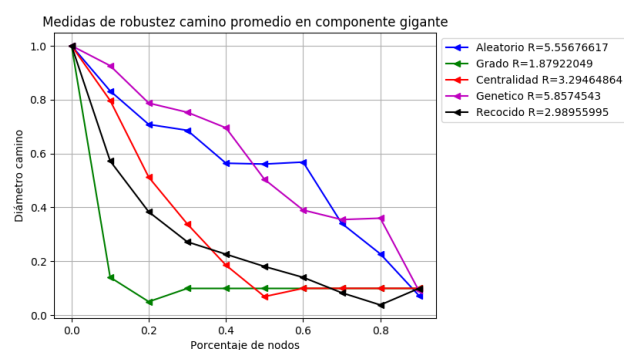


Figura 9.15: Análisis de robustez de red libre de escala de 2000 nodos por diámetro de camino más corto

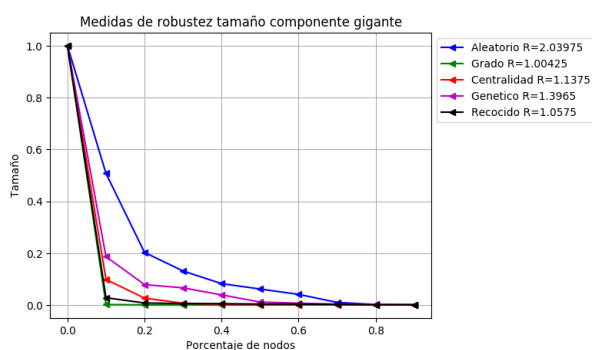


Figura 9.16: Análisis de robustez de red libre de escala de 4000 nodos por tamaño de componente gigante

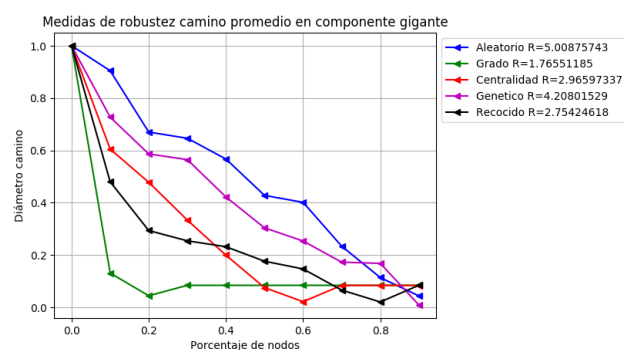


Figura 9.17: Análisis de robustez de red libre de escala de 4000 nodos por diámetro de camino más corto

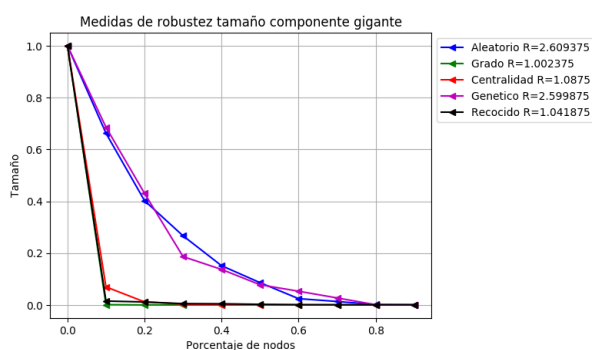


Figura 9.18: Análisis de robustez de red libre de escala de 8000 nodos por tamaño de componente gigante

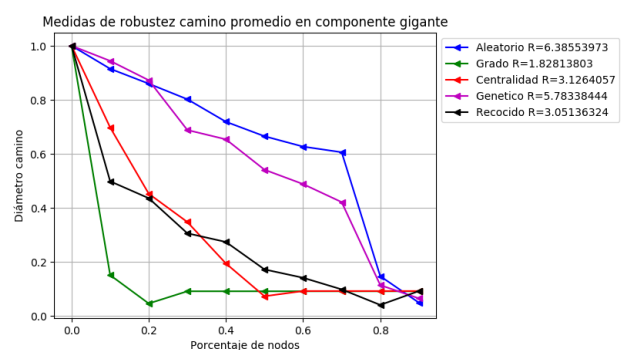


Figura 9.19: Análisis de robustez de red libre de escala de 8000 nodos por diámetro de camino más corto

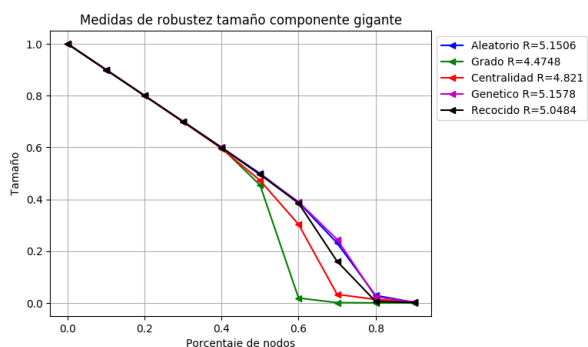


Figura 9.20: Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=10\%$ tamaño de componente gigante

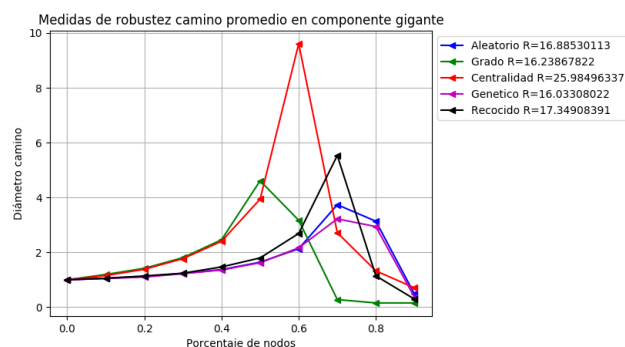


Figura 9.21: Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=10\%$ por diámetro de camino más corto

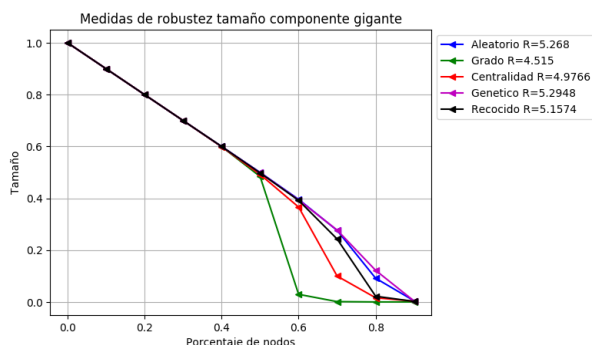


Figura 9.22: Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=20\%$ tamaño de componente gigante

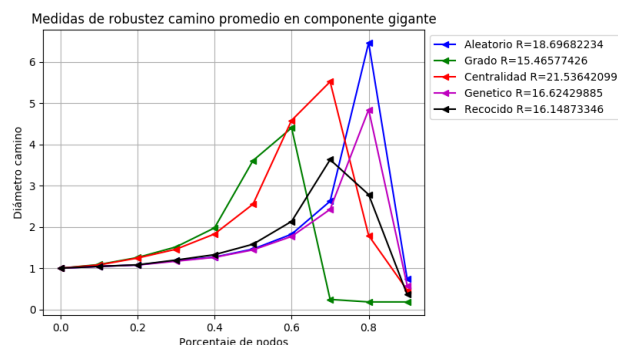


Figura 9.23: Análisis de robustez de red de mundo pequeño de 5000 nodos con $p=20\%$ por diámetro de camino más corto

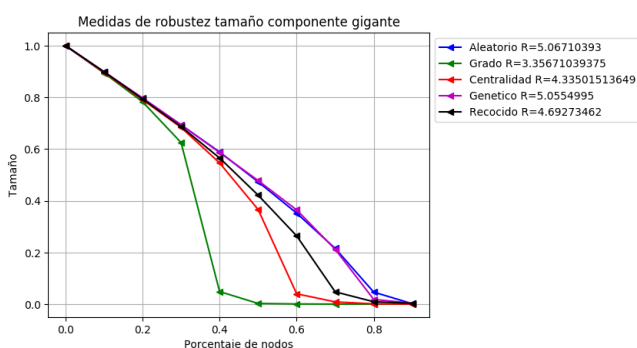


Figura 9.24: Análisis de robustez de red aleatoria con 1991 nodos por tamaño de componente gigante

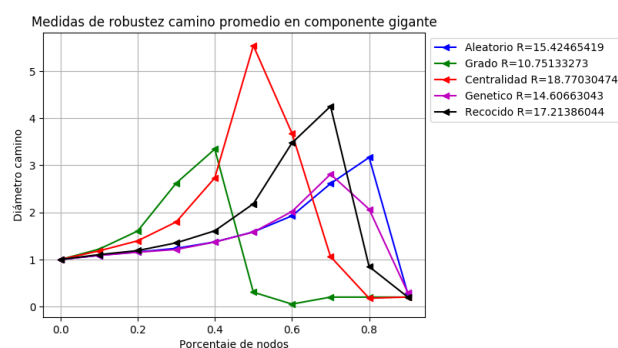


Figura 9.25: Análisis de robustez de red aleatoria con 1991 nodos por diámetro de camino más corto

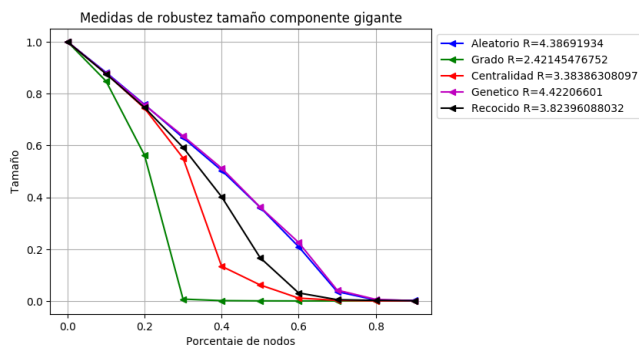


Figura 9.26: Análisis de robustez de red aleatoria con 3373 nodos por tamaño de componente gigante

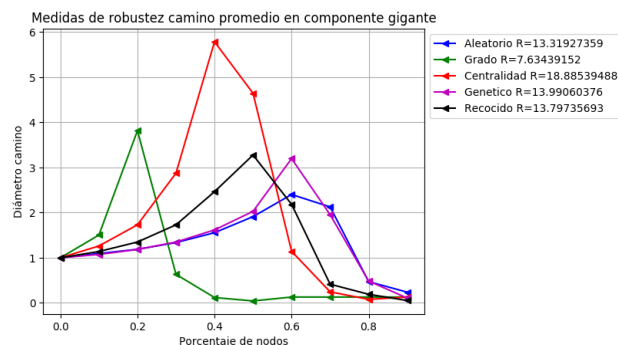


Figura 9.27: Análisis de robustez de red aleatoria con 3373 nodos por diámetro de camino más corto

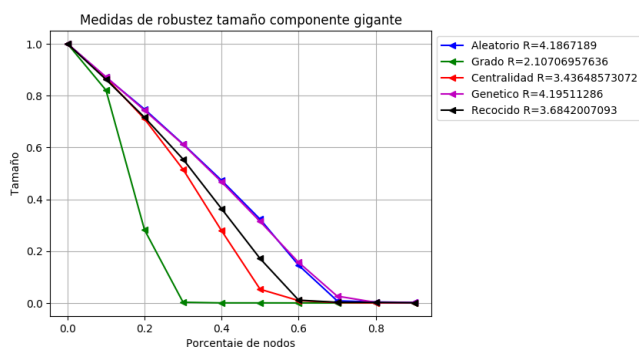


Figura 9.28: Análisis de robustez de red aleatoria con 5620 nodos por tamaño de componente gigante

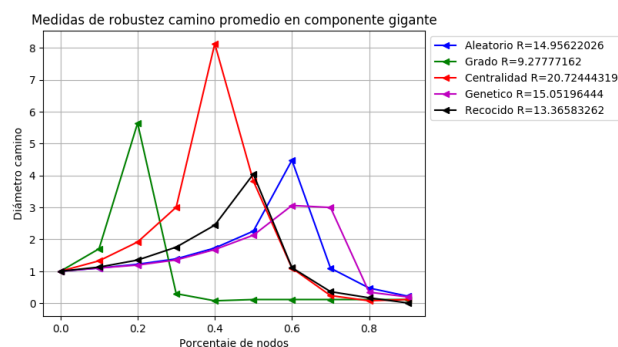


Figura 9.29: Análisis de robustez de red aleatoria con 5620 nodos por diámetro de camino más corto

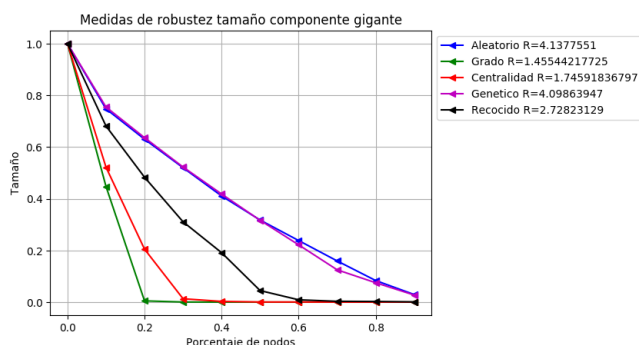


Figura 9.30: Análisis de robustez de red observada Ecoli por tamaño de componente gigante

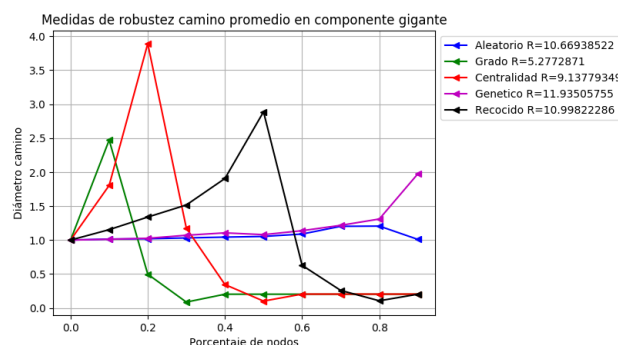


Figura 9.31: Análisis de robustez de red observada Ecoli por diámetro de camino más corto

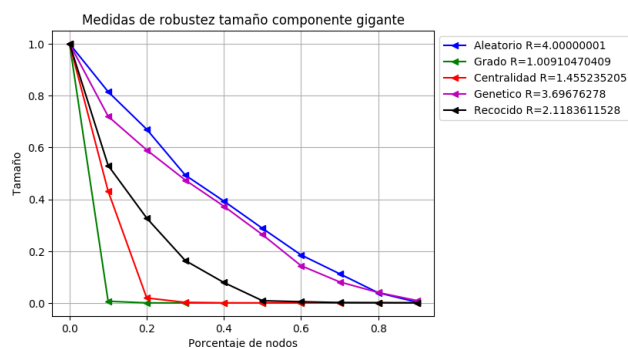


Figura 9.32: Análisis de robustez de red observada Celegens por tamaño de componente gigante

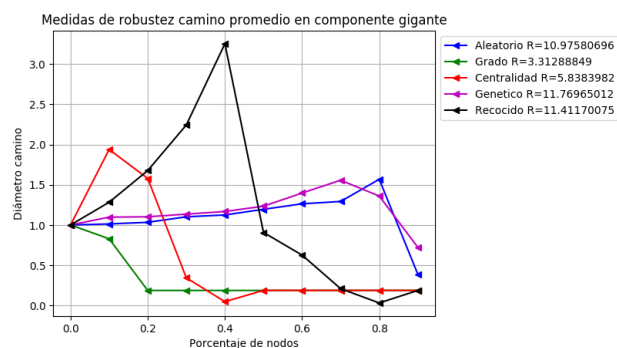


Figura 9.33: Análisis de robustez de red observada Celegens por diámetro de camino más corto

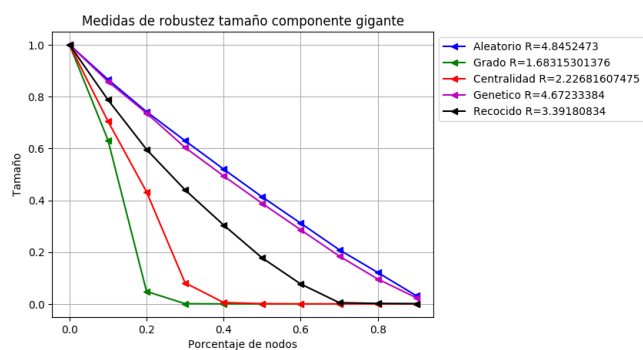


Figura 9.34: Análisis de robustez de red observada Cerevisae por tamaño de componente gigante

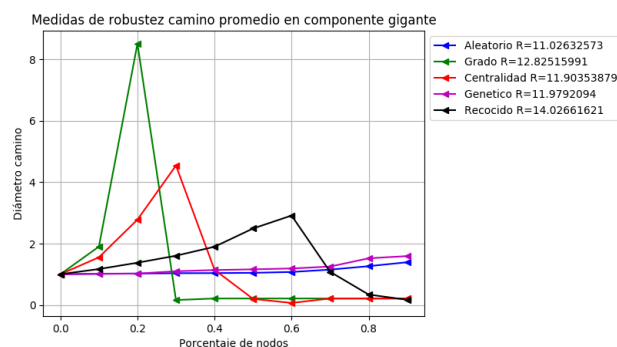


Figura 9.35: Análisis de robustez de red observada Cerevisae por diámetro de camino más corto

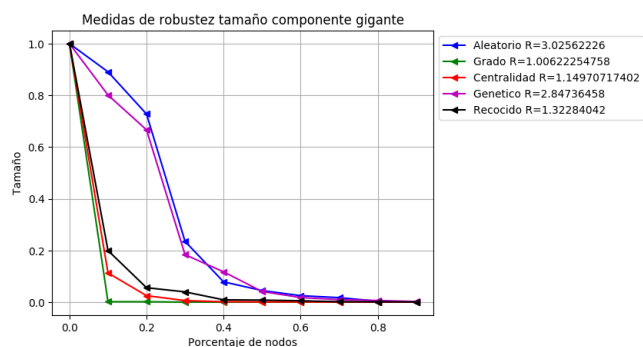


Figura 9.36: Análisis de robustez de red fractal (2,2)-flower por tamaño de componente gigante

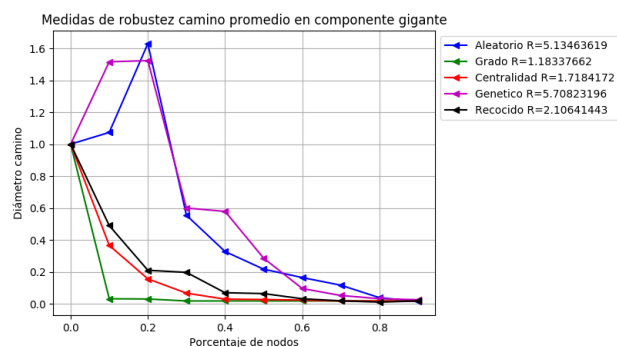


Figura 9.37: Análisis de robustez de red fractal (2,2)-flower por diámetro de camino más corto

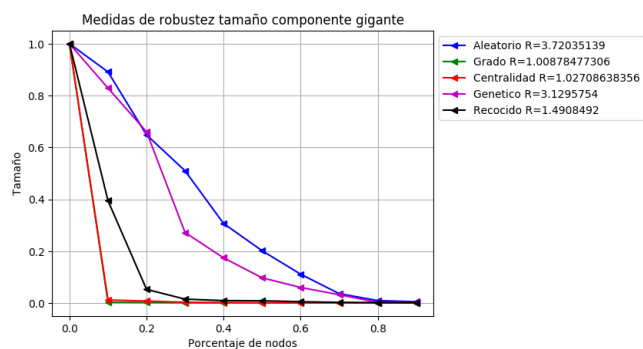


Figura 9.38: Análisis de robustez de red fractal (1,3)-flower por tamaño de componente gigante

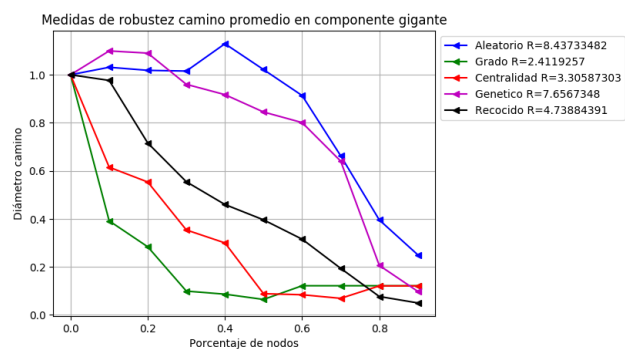


Figura 9.39: Análisis de robustez de red fractal (1,3)-flower por diámetro de camino más corto