# Manual de uso del simulador de reconexión de redes complejas

Este simulador permite hacer dos tipo de experimentos:

1. Formación de redes a partir de la reconexión de enlaces.
2. Degradación de redes a partir de la simulación secuencial de fallos (eliminación aleatoria de nodos) o de ataques (eliminación de nodos con el grado más alto).

Una vez que se han ejecutado los experimentos es posible calcular medidas estructurales de las redes obtenidas o degradadas.

## I. Instalación de dependencias e identificación de rutas

Los siguientes pasos se deben ejecutar por única ocasión cuando se usa el simulador por primera vez.

* 1. Crear un ambiente de Python en el que se ejecutará el simulador, se recomienda usar Anaconda.
  2. Dentro del ambiente creado, instalar los siguientes paquetes:
     + Networkx
     + Numpy
     + Matplotlib
     + Pandas
  3. Descargar la versión del simulador disponible en <https://github.com/daniaguirre/ComplexNetworks.git>
  4. Crear un **directorio en donde se guardarán los resultados de los experimentos**, se recomienda crear un directorio exclusivo para los resultados. Se debe considerar la ruta relativa al home del usuario. Por ejemplo, en la siguiente ruta la carpeta de Documents está dentro del home del usuario:

/Documents/Repositorios/ResultadosCN

* 1. Dentro del directorio de resultados, crear dos carpetas: una para los resultados de los experimentos de formación y otras para los resultados de degradación de las redes formadas, por ejemplo:

/Documents/Repositorios/ResultadosCN/Formación

/Documents/Repositorios/ResultadosCN/Degradación

* 1. Modificar el archivo **config.sh** para indicar la **ruta absoluta** la carpeta del ambiente de conda (línea 2) y de la **ruta relativa** (al home de usuario) de la carpeta de resultados (línea 4), por ejemplo:

/anaconda3/envs/networks/bin

/Documents/Repositorios/ResultadosCN

* 1. Modificar el archivo **config\_paths.py** para indicar las rutas relativas (al home de usuario) de las carpetas de resultados de formación y degradación (líneas 5 y 6).

## II. Experimentos de formación de redes

### Configuración de experimentos de formación de redes

Para configurar un experimento de formación, se deben considerar los siguientes parámetros de entrada del simulador:

* **Longitud del enlace dinámico.** D, D/2, D/4, D/8, D/16, etc.
* **Regla de reconexión.** R1, R2, R3
* **Topología inicial.** Malla, Anillo
* **Orden de la topología inicial.**
* **Algoritmo de ruteo.** Compass routing (CR), Shortest path (SP). Random walk (RW).
* **Número de ciclos de ejecución.**

A continuación, se indican los pasos para configurar un experimento con los siguientes parámetros de entrada:

* **Longitud del enlace dinámico:** D, D/2, D/4.
* **Regla de reconexión:** R1
* **Topología inicial:** Malla
* **Orden de la topología inicial:** 8 columnas x 8 renglones = 64
* **Algoritmo de ruteo:** Compass routing (CR).
* **Número de ciclos de ejecución:** 5.

### Pasos:

1. Crear el siguiente árbol de directorios dentro de la carpeta definida para los resultados de formación.

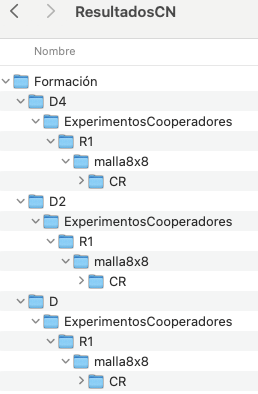


Figura 1. Estructura del directorio de resultados de Formación.

1. Modificar el archivo config.sh (líneas 9, 10, 11, 12 y 14) para indicar los siguientes datos del experimento:

#--------FORMACION-------------

ROUTING="CR" # Algoritmo de encaminamiento: "CR", "RW", "SP"

REGLA="1" # 1,2,3

CICLOS="5"

declare -a ARR=("D" "D2" "D4")

#--------RED-------------------

RED="malla8x8" #anillo

1. Modificar el archivo config.py (líneas 5, 6, 8, 9, 10, 11 y 13) para indicar los siguientes datos del experimento:

#--------MALLA-------------------

ROWS=8 #Filas de la malla

COLUMNS=8 #Columnas de la malla

#--------FORMACION-------------

RED="malla"

ROUTING="COMPASS-ROUTING" #"COMPASS-ROUTING","RANDOM-WALK","SHORTEST-PATH"

REGLA=1 #1,2,3

LONG\_ENLACE=1 #Divisor de la longitud de enlace dinámico: 1,2,4,etc.

#--------EJECUCION---------------

CICLOS=5 # Número de ciclos de simulación

Es importante notar que para la variable LONG\_ENLACE debemos configurar 3 longitudes (D, D/2 y D/4), esto se realizará en el paso 5.

1. Ejecutar el script **0creaCopiaSimulador.py**, el cual hará una copia de los scripts del simulador a los directorios hoja (CR) de la Figura 1. Después de ejecutar el script, cada directorio debe tener lo siguientes archivos

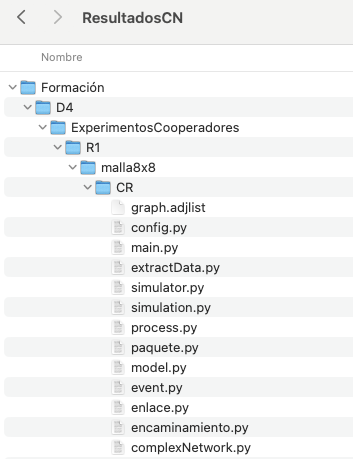


Figura 2. Archivos contenidos en las directorios hoja del directorio de resultados de Formación.

1. En cada directorio hoja (CR), modificar la línea 11 del archivo **config.py** para longitud del enlace dinámico correspondiente, de la siguiente manera

* En el archivo D/ExperimentosCooperadores/R1/malla8x8/CR/config.py colocar:

LONG\_ENLACE=1

* En el archivo D2/ExperimentosCooperadores/R1/malla8x8/CR/config.py colocar:

LONG\_ENLACE=2

* En el archivo D4/ExperimentosCooperadores/R1/malla8x8/CR/config.py colocar:

LONG\_ENLACE=4

1. Dentro del directorio ResultadosCN, copiar los archivos **formacion.sh** y **config.sh**. Estos scripts automatizan la ejecución de los experimentos de formación definidos en este ejemplo.

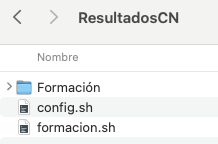


Figura 3. Ubicación de los scripts formacion.sh y config.sh.

1. Ejecutar el script **formacion.sh**.

chmod +x **formacion.sh**

./**formacion.sh 1**

Pasar como argumento 1 si se trabaja con malla y 3 con anillo.

1. Para cada configuración se realizarán 10 ejecuciones del experimento. De modo que al terminar la ejecución del script **formacion.sh**, dentro de cada directorio CR, habrá 10 carpetas cuyo nombre corresponde al número de experimento. En cada una de estas carpetas, se encontrarán los siguientes archivos:
   * **Listas de adyacencia** (archivos de extensión *adjlist*) de las redes obtenidas en cada ciclo de ejecución.
   * **Distribuciones de grados** (archivos de nombre *hist\_test*) de las redes obtenidas en cada ciclo de ejecución.
   * Reporte de **medidas estructurales** (coeficiente promedio de agrupamiento, longitud promedio de la trayectoria más corta, diámetro) de las redes obtenidas en cada ciclo de ejecución (archivo datos-salida).

### Generación de imágenes de las redes obtenidas y gráficas de sus medidas estructurales

A continuación, se indican la lista de archivos para obtener distintos reportes de medidas estructurales e imágenes de las redes generadas.

Antes de ejecutar los archivos se deben ejecutar los experimentos de formación y configurar el archivo config\_7GraficasFormacion.py con información de los experimentos ejecutados, por ejemplo:

#PARA AGENTES COOPERADORES:

TAM\_ENLACES=["D","D2","D4"]

REGLAS=["R1"]

TOPOLOGIAS=["malla8x8"]

ALG\_ENCAMINAMIENTO=["CR"]

#PARA AGENTES SEMI-COOPERADORES:

TAM\_ENLACES\_SC=["D","D2","D4","D8","D16"]

REGLAS\_SC=[]#["R1","R2","R3"]

TOPOLOGIAS\_SC=["anillo1500","malla50x50"]

CONEXIONES\_SC=["n4","n16","n64"]

* **1creaPromediosFormacion.py**

Para cada configuración de experimento (p.e. ResultadosCN/Formación/D2/ExperimentosCooperadores/R1/malla8x8/CR) crea el archivo datos-promedio.csv que contiene los promedios y la desviación estándar para medidas estructurales de todas las ejecuciones del experimento.

Los datos se muestran en una tabla con las siguientes columnas:

* + Ciclo
  + Promedio del coeficiente de agrupamiento
  + Promedio de la trayectoria más corta
  + Diámetro
  + Desviación estándar del promedio del coeficiente de agrupamiento
  + Desviación estándar del promedio de la trayectoria más corta
  + Desviación estándar del diámetro
* **2creaPromediosDistGradosFormacion.py**

Para cada configuración de experimento (p.e. ResultadosCN/Formación/D2/ExperimentosCooperadores/R1/malla8x8/CR) crea el archivo datos-promedio\_grados.csv que contiene los promedios de la distribución de grados de todas las ejecuciones del experimento.

Los datos se muestran en una tabla con las siguientes columnas:

* + grado
  + prom\_nodos
* **3creaImagenesPNG.py**

Para cada ejecución de experimento (p.e. ResultadosCN/Formación/D2/ExperimentosCooperadores/R1/malla8x8/CR/5) crea imágenes de la red final y su distribución de grados promedio (estos datos se obtienen de datos-promedio\_grados.csv).

Por ejemplo, para el experimento 5, las imágenes se guardan en los archivos:

* 1. distGradosHist\_test\_5.png
  2. img\_graph\_test\_5.png
* **4creaGEXF.py**

Para cada ejecución de experimento (p.e. ResultadosCN/Formación/D2/ExperimentosCooperadores/R1/malla8x8/CR/5) crea un archivo GEXF de la red final, por ejemplo, para el experimento 5, el archivo se guarda con el nombre graph\_test\_5.gexf y se puede visualizar en gephi.

El archivo contiene las coordenadas de cada nodo y se pueden usar en gephi con el plugin geoLayout.

* **5creaSemicooperadores.py**

Crea promedio para los experimentos para los experimentos semicooperadores. Falta documentar.

* **7creaGraficasFormacion.py**

Para las redes finales obtenidas en todas las configuraciones de experimentos agrupa los datos promedio de las siguientes medidas estructurales:

* + Promedio del coeficiente de agrupamiento
  + Promedio de la trayectoria más corta
  + Diámetro
  + Distribución de grados

## III. Experimentos de degradación de redes

### Configuración de experimentos de degradación de las redes obtenidas

Para configurar un experimento de degradación, se deben haber ejecutado los experimentos de formación y considerar los siguientes parámetros de dichos experimentos y el tipo de degradación:

* **Tipo de degradación.** Ataques, Fallas
* **Longitud del enlace dinámico.** D, D/2, D/4, D/8, D/16, etc.
* **Regla de reconexión.** R1, R2, R3
* **Topología inicial.** Malla, Anillo
* **Orden de la topología inicial.**
* **Algoritmo de ruteo.** Compass routing (CR), Shortest path (SP). Random walk (RW).

A continuación, se indican los pasos para configurar un experimento con los siguientes parámetros de entrada:

* **Tipo de degradación:** Ataques.
* **Longitud del enlace dinámico:** D, D/2, D/4.
* **Regla de reconexión:** R1
* **Topología inicial:** Malla
* **Orden de la topología inicial:** 8 columnas x 8 renglones = 64
* **Algoritmo de ruteo:** Compass routing (CR).

### Pasos:

1. Dentro de la carpeta de resultados, crear una carpeta Degradación, dentro de ella dos carpetas: Ataques y Fallas; dentro de cada una de estas carpetas crear una carpeta GrafosFinales. Un ejemplo de los directorios creados es:

/Documents/Repositorios/ResultadosCN/Degradación/Ataques/GrafosFinales

/Documents/Repositorios/ResultadosCN/Degradación/Fallas/GrafosFinales

1. Dentro de la carpeta Degradación, crear las siguientes carpetas:

/Documents/Repositorios/ResultadosCN/Degradación/Gráficas\_Degradación\_Ataques

/Documents/Repositorios/ResultadosCN/Degradación/Gráficas\_Degradación\_Fallas

1. Dentro de las carpetas GrafosFinales, copiar las carpetas con las redes obtenidas en los experimentos de formación, las carpetas que se deben copiar son las que qué comienzan con D y están dentro de la carpeta Formación. De este modo se obtiene el siguiente árbol de directorios:

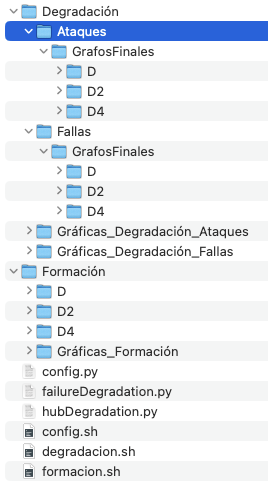


Figura 4. Estructura del directorio de resultados de Degradación.

1. Modificar el archivo config.sh para indicar los siguientes datos del experimento de formación y degradación.

#--------FORMACION-------------

# config.sh

# Estos directorios debe estar definidos en el home

CONDA\_DIR="/anaconda3/envs/networks/bin"

RESULTADOS\_DIR="/Documents/Repositorios/ResultadosCN"

# Las contantes de EXPERIMENTO y RED definen

# los nombres de las carpetas de los resultados

#--------FORMACION-------------

ROUTING="CR" # Algoritmo de encaminamiento: "CR", "RW", "SP"

REGLA="1" # 1,2,3

CICLOS="5"

declare -a ARR=("D" "D2" "D4")

#--------RED-------------------

RED="malla8x8" #anillo

#--------DEGRADACION-----------

DEGRADACION="Ataques" # "Ataques" / "Fallas"

FILE\_DEGRADATION="hubDegradation.py" # "failureDegradation.py" / "hubDegradation.py"

Notar que DEGRACION y FILE\_DEGRADATION deben ser congruentes, para Ataques se usa hubDegradation.py, mientras que para Fallas usa failureDegradation.py.

1. Modificar el archivo config.py (líneas 5, 6, 8, 9, 10, 11 y 13) para indicar los siguientes datos del experimento:

#Configuración del experimento

#--------ANILLO------------------

NODOS\_ANILLO=1500 # Número de nodos del anillo

#--------MALLA-------------------

ROWS=8 # Filas de la malla

COLUMNS=8 # Columnas de la malla

#--------FORMACION-------------

ROUTING="COMPASS-ROUTING" # Algoritmo de encaminamiento: "COMPASS-ROUTING", "RANDOM-WALK", "SHORTEST-PATH"

REGLA=1 # 1,2,3

LONG\_ENLACE=1 # Divisor de la longitud de enlace dinámico: 1, 2, 4, 8, 16, 32

#--------EJECUCION---------------

CICLOS=5 # Número de ciclos de simulación

#--------EXTRAS------------------

ENLACES\_DINAMICOS=2 # Número de enlaces dinámicos por nodo

EXPLORADORES=20 # Número de paquetes exploradores por ciclo

DIV\_CONEXIONES=1 # Divisor (con respecto al número de nodos, num\_nodos/DIV\_CONEXIONES) del máximo número de conexiones permitidas

#--------VISUALIZACION-----------#Parameteros para 3creaImagenesPNG.py

NODE\_SCALE = 10 #Usar 1 para orden de anillo=1500 y orde de malla=2500

#-------DEGRADACION------------

SAVE\_STEP=4 #Cada cuántos ataques registra medidas

1. Dentro del directorio ResultadosCN, copiar los siguientes archivos automatizan la ejecución de los experimentos de degradación:
   * **degradacion.sh**
   * **config.sh**
   * **failureDegradation.py.** Permite ejecutar degradación por fallos.
   * **hubDegradation.py**. Permite ejecutar degradación por ataques.
2. Ejecutar el script **degradacion.sh**.

chmod +x **degradacion.sh**

./**degradacion.sh**

Se ejecutará la degradación para las redes obtenidas en cada experimento. De modo que al terminar la ejecución del script **degradacion.sh**, dentro de cada carpeta de experimento, se encontrarán los siguientes archivos:

* + **AttackSequence.txt** – Lista de nodos en el orden que fueron eliminados.
  + **DegradationData.txt** – Reporte de medidas estructurales a lo largo del proceso de degradación (número de nodos, número de enlaces, longitud promedio de la trayectoria más corta, diámetro, coeficiente promedio de agrupamiento, asortatividad, conectividad de la red, número de componentes conexos, numero de nodos en el componente más grande).
  + **degreeHistograms.txt –** Lista con las distribuciones de grado de las redes a lo largo del proceso de degradación.

### Generación de imágenes de redes y gráficas de medidas

A continuación, se indican la lista de archivos para obtener distintos reportes de medidas estructurales e imágenes de las redes degradadas.

Antes de ejecutar los archivos se deben ejecutar los experimentos de degradación para fallas y ataques.

* **6creaPromediosDegradacion.py**

Para cada configuración de experimento de degradación, fallos y ataques (p.e. ResultadosCN/Degradación/Atques/GrafosFinales/D2/ExperimentosCooperadores/R1/malla8x8/CR) crea el archivo datos-promedio.csv que contiene los promedios de medidas estructurales de todas las ejecuciones a lo largo del proceso de degradación.

Los datos se muestran en una tabla con las siguientes columnas:

* + Promedio del coeficiente de agrupamiento
  + Promedio de la trayectoria más corta
  + Diámetro
  + Orden relativo del componente más grande
  + Promedio de asortatividad

También crea el archivo attr-promedio.csv que contiene los promedios de ATTR y m-ATTR de todas las ejecuciones a lo largo del proceso de degradación.

* **8creaGraficasDegradacion.py**

Para las redes finales degradadas en todas las configuraciones de experimentos agrupa los datos promedio de las siguientes medidas estructurales a lo largo del proceso de degradación:

* + Promedio del coeficiente de agrupamiento
  + Promedio de la trayectoria más corta
  + Diámetro
  + Orden relativo del componente más grande
  + Asortatividad