

Algoritmos genéticos aplicados a la optimización de confiabilidad de redes

Federico Gil
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
federico.gil@fing.edu.uy

Paula Ripa
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
paula.ripa@fing.edu.uy

Abstract—Este informe tiene como objetivo el uso de algoritmos genéticos para mejorar la confiabilidad de redes. En este caso, se tomaron tres redes eléctricas de distintos tamaños y se propusieron planes de actualización para dichas redes, a modo de encontrar un balance entre el costo de actualización y la confiabilidad de la red. Se compararon las soluciones obtenidas con las generadas por algoritmos Greedy.

Index Terms—Evolución, Mono-objetivo, Confiabilidad, Monte Carlo, Optimización

I. INTRODUCCIÓN

Este documento es un informe del proyecto de fin de curso presentado en la edición 2022 de Algoritmos Evolutivos de la Facultad de Ingeniería de la Universidad de la República.

El proyecto aborda el problema de diseño de redes confiables. Consiste en construir nuevas conexiones en una red de forma que la probabilidad de que un grupo de nodos de la red quede aislado sea mínima, sujeto a un presupuesto.

Para los fines de este proyecto, se utilizara la definición de *all-terminal-reliability* [1], que consiste en garantizar que todos los nodos de la red pueden comunicarse con todos los otros.

II. PROBLEMA

A. Definición de conceptos

Sea $G = (V, E, p)$ un grafo conexo, no dirigido e incierto. Donde V es el conjunto de vértices, $E \subseteq V \times V$ es el conjunto de aristas, y $p : E \rightarrow (0, 1]$ es una función que determina la existencia de $e \in E$ en el grafo. Se denota $e = (v_1, v_2)$. La existencia de e queda determinada por la probabilidad $p(e)$ y su no existencia con probabilidad $1 - p(e)$. Se asumirá que la probabilidad de existencia de las distintas aristas es independiente.

1) *No dirigido*: un grafo no dirigido es un tipo de grafo en el cual las aristas representan relaciones simétricas. Esto quiere decir que si $xRy \iff yRx$

2) *Conexo*: decimos que un grafo no dirigido es conexo si, por cada par de nodos (u, v) , existe un camino desde u hasta v

B. Caso de estudio

El problema a resolver es maximizar la confiabilidad de la red ya existente, teniendo a disposición un presupuesto

para construir conexiones entre nodos. Mas específicamente, para este estudio se tendrán en cuenta una variedad de redes eléctricas de distintos tamaños que serán especificados en VI-B.

Los nodos V de la red representan las subestaciones de la red eléctrica, se considerara que los nodos son perfectos, es decir que su probabilidad de fallar es nula.

Las aristas E , representan las conexiones cableadas entre subestaciones. De estas, se conocen las capacidades en kV que a fines de este estudio, solo serán utilizadas para el calculo de costo [3].

Se conocen también los largos de los cables que conectan las subestaciones, así como la probabilidad p de que estos fallen, ambos parámetros están fuertemente correlacionados. Se considerara que las conexiones solo tienen dos estados posibles, operacional o no operacional. Finalmente, la probabilidad de que estén operacionales es independiente.

C. Justificación de uso AE

Hallar las conexiones óptimas puede ser formulado como un problema combinatorio, en el cual la selección de determinadas aristas maximiza la confiabilidad de la red dado una restricción de costo. Este problema es NP-Hard pues el espacio de búsqueda crece exponencialmente con el número de nodos. Esto despierta el interés en la aplicación de algoritmos genéticos.

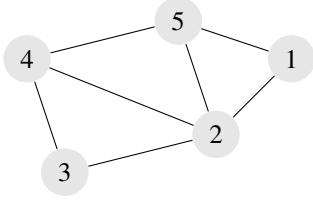
III. SOLUCIÓN

En esta sección se presenta la codificación de individuos y la función de fitness, aspectos básicos necesarios para las primeras pruebas.

A. Representación

El cromosoma de los individuos es una tira binaria que representa el estado de las aristas, cero si la arista no esta presente y uno si lo esta. Dentro de esta codificación se tiene en cuenta que no puede haber lazos ni aristas múltiples y tampoco se tienen en cuenta en la codificación las aristas originales de la red.

El largo del cromosoma del individuo sera $\binom{N}{2} - |E|$, esto es, todas las aristas posibles en un grafo no dirigido completo menos las aristas que ya están presentes en el grafo.



$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ x_{12} & x_{13} & x_{14} & x_{15} & x_{23} & x_{24} & x_{25} & x_{34} & x_{35} & x_{45} \end{bmatrix}$$

B. Función de fitness

Para la evaluación de los individuos es necesario lograr un compromiso entre el uso del presupuesto y la confiabilidad de la red. En este caso, se busca minimizar el objetivo costo y maximizar la confiabilidad.

Notación

- P - Presupuesto
- C_0 - Costo de la red original
- C_N - Costo de la nueva red
- R_λ - Confiabilidad de la nueva red
- C_{max} - Costo máximo

Se define la función de fitness:

$$\alpha \frac{C_N - C_0}{C_{max}} + \beta \frac{1}{R_\lambda} \quad (1)$$

Sujeto a $\alpha \in (0, 1)$ y $\beta = 1 - \alpha$

La función de fitness presentada busca encontrar un balance entre el costo de la actualización y la confiabilidad, los parámetros α y β serán utilizados para ajustar dicha función. Como se plantea en [6]

Esta función de fitness sera dinámica, en el sentido de que la precisión con la que se mide la confiabilidad de la red¹ se ajustara de forma de que en las primeras generaciones la precisión sera intencionalmente baja y a mitad que transcurran las generaciones esta ira aumentando en una taza λ , con el fin de hilar mas fino que individuos se seleccionan para el cruzamiento. De no hacer una inicialización greedy para la población, este paso puede ahorrar mucho tiempo de computo pues el calculo de la confiabilidad de una red muy densa y por lo tanto una red donde la falla o desconexión se considera un evento raro requiere un número de muestras muy elevado.

C. Soluciones no factibles

Se tiene una restricción de presupuesto (P) lo que provoca que existan soluciones no factibles, es decir que no cumplen con la condición de que el costo de actualización sea menor que el presupuesto disponible. Existen diversos métodos para lidiar con dichas soluciones, entre ellos penalizarlas con valores de fitness muy malos, eliminarlas de la población o corregirlas, para la solución de este problema se utilizara la ultima.

Para la corrección de las soluciones se utilizara la siguiente heurística, Sea S_i una solución no factible², Si S_i supera el presupuesto (P), con una probabilidad γ se quitaran aristas de S_i hasta que el costo de la actualización ($C_N - C_0$) sea menor a (P). Quitando siempre la arista mas costosa primero. Mientras que, con probabilidad $1 - \gamma$ se quitaran aristas de manera aleatoria hasta que el costo sea menor al presupuesto disponible, esto con la finalidad de mantener en la población soluciones que no sean tan buenas pero que al ser combinadas con otras mediante los mecanismos de cruzamiento generen soluciones mejores. A continuación se presenta un pseudocódigo de la primera estrategia:

Input: Grafo G , Presupuesto P , Costos C

Output: S_i

- 1: **for all** arista \in Grafo **do**
- 2: Acumular costo
- 3: **end for**
- 4: **while** costo de la red $> P$ **do**
- 5: seleccionar (a,b) la arista más costosa y removerla
- 6: **end while**
- 7: **return** S_i

La segunda estrategia:

Input: Grafo G , Presupuesto P , Costos C

Output: S_i

- 1: **for all** arista \in Grafo **do**
- 2: Acumular costo
- 3: **end for**
- 4: **while** costo de la red $> P$ **do**
- 5: seleccionar (a,b) aleatoriamente y removerla
- 6: **end while**
- 7: **return** S_i

D. Estimación de la confiabilidad para la función de fitness

Esta estrategia presenta una clara dificultad, calcular la confiabilidad de al red. Esto, en si es un problema NP-Hard, por lo que se utilizaran métodos de estimación para aproximar el valor exacto.

Para esto se utilizara una implementación del método Montecarlo. A continuación se presenta el pseudocódigo del método [5].

Input: Grafo G , CantidadMuestras N , Delta δ

Output: Estimador, IDC de Chebyshev

Inicialización :

- 1: conexos = 0
- LOOP Principal*
- 2: **for** $i = 1 \dots N$ **do**
- 3: gt = g.Sample()
- 4: **if** (gt.esConexo()) **then**
- 5: conexos += 1
- 6: **end if**
- 7: **end for**
- 8: Estimador = conexos/ N

²Notar que solo pueden haber soluciones no factibles del tipo mencionado y no sobre las aristas posibles.

¹Basada en un muestreo Montecarlo que se explicara en III-D

- 9: Calcular IDC de Chevyshev
- 10: **return** Estimador, IDC

Para saber si un grafo es conexo se utilizara una búsqueda DFS, resultado de la cual, si todos los nodos son visitados, dicho grafo es conexo.

Mientras que el estimador brinda un resumen puntual de los datos, el IDC aporta un rango en el cual con una probabilidad dada $(1 - \delta)$, el estimador pertenece al intervalo. Dado que la distribución de las variables de la simulación es binomial [4], pues se cuenta el numero de éxitos, se utilizara el intervalo de confianza de Chevyshev. Este intervalo si bien garantiza el nivel de cobertura, es muy generoso, existen otros intervalos para este tipo de distribuciones que garantizando el nivel de cobertura son más angostos, un ejemplo de estos es Agresti-Coull. Por cuestiones de implementación se utilizara el de Chevyshev.

1) *Intervalo de confianza de Chevyshev*: Siguiendo el criterio de Chebyshev, se pueden calcular w_1 y w_2 utilizando las siguientes formulas.

$$w_1(z, n, \beta) = \frac{z + \frac{\beta^2}{2} - \beta \sqrt{\frac{\beta^2}{4} + \frac{z(n-z)}{n}}}{n + \beta^2} \quad (2)$$

$$w_2(z, n, \beta) = \frac{z + \frac{\beta^2}{2} + \beta \sqrt{\frac{\beta^2}{4} + \frac{z(n-z)}{n}}}{n + \beta^2} \quad (3)$$

El intervalo de confianza (IDC) queda dado por $(w_1(S, N, \delta^{-1/2}), w_2(S, N, \delta^{-1/2}))$. Donde S es la cantidad de grafos conexos resultado de la simulación, N el numero de muestras y δ la probabilidad de no pertenecer al IDC.

Otro aspecto importante de la simulación es la cantidad de muestras necesarias para garantizar el criterio de error, para esto se utilizara el Teorema de Hoeffding.

2) *Teorema de Hoeffding*: Para calcular la cantidad de muestras necesarias para las simulaciones utilizaremos el siguiente criterio basado en el teorema de Hoeffding, el cual recibe como entrada un criterio de error (ϵ, δ)

$$n_H(\epsilon, \delta) = \left\lceil \frac{\ln 2/\delta}{2\epsilon^2} \right\rceil \quad (4)$$

E. Inicialización Greedy

Inicializar la población de manera inteligente resulta de gran utilidad pues ahorra tiempo de evolución, sin embargo, también acarrea desventajas, una pobre diversidad genética, por lo que se propone para el caso de estudio, una una inicialización greedy no determinista.

La población sera inicializada con un 30%³ de individuos aleatorios la estrategia a utilizar será la siguiente:

Input: Grafo G , Presupuesto P , Costos C

Output: P_i

- 1: sample(G)

³Este valor se eligió de corridas preliminares, busca conservar la diversidad genética mientras que ofrece un buen punto de inicio

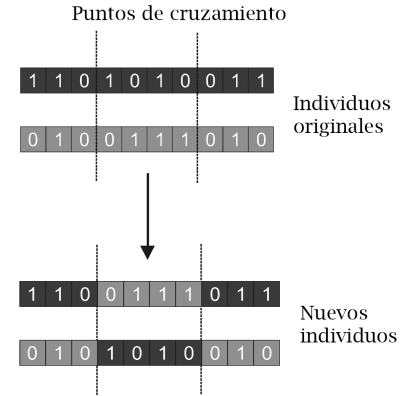
- 2: **for all** nodo \in Grafo **do**
- 3: Calcular grado de todos los nodos
- 4: **end for**
- 5: **while** costo de la red $< P$ **do**
- 6: Agregar (a,b) la arista entre los dos nodos con menos grado
- 7: **end while**
- 8: Codificar las aristas agregadas al cromosoma del individuo
- 9: **return** P_i

F. Operadores Evolutivos

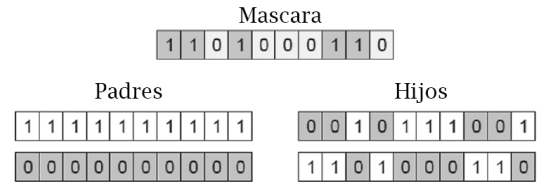
1) *Mutación*: Se utilizara el operador BitFlipMutation, el cual con una probabilidad dada invierte un bit en el cromosoma del individuo, posibilitando la exploración de diferentes secciones del espacio de búsqueda

2) *Cruzamiento*: El operador de cruzamiento busca explorar las buenas regiones del espacio de búsqueda, para este caso se utilizaran el cruzamiento de dos puntos (2PX) y el cruzamiento uniforme (UX).

El cruzamiento en dos puntos (2PX) consiste en elegir uniformemente dos puntos, estos puntos definiran que seccion del cromosoma se intercambiara.



Mientras que el cruzamiento uniforme (UX) genera una mascara donde la proporción de alelos a intercambiar queda definida por la probabilidad establecida. Dicha mascara es posteriormente aplicada a los padres. Se intercambia la información genética en la i -ésima posición si en la misma posición de la mascara hay un uno.



3) *Selección*: La selección es el proceso mediante el cual algunos individuos en una población son seleccionados para reproducirse basados en su aptitud, para la solución de este problema se utilizara BinaryTournamentSelection, el cual se basa en la mecánica de torneo para la selección de los individuos que se reproducirán.

IV. PROPUESTA DE EVALUACIÓN EXPERIMENTAL

A. Generación de instancias

Se dispone de información de tres redes eléctricas de tamaños 24, 54 y 118 nodos, con sus respectivas aristas y probabilidades de fallo [2]. Como se menciono anteriormente, los nodos representan las subestaciones, las aristas, las conexiones cableadas entre nodos y la probabilidad de que falle una arista es la probabilidad de que falle dicha conexión.

B. Comparación con otras técnicas o soluciones

Al ser incapaces de hallar la solución óptima al problema pues este es NP-Hard, se comparara las soluciones presentadas por los algoritmos genéticos con técnicas greedy. A continuación se presentan dos técnicas greedy intuitivas.

La primera técnica consiste en conectar con una nueva arista nodos con el grado mas bajo, hasta que agregar una nueva arista signifique pasarse de presupuesto.

Se presenta un pseudocódigo para dicha propuesta:

Input: Grafo G , Presupuesto P , Costos $C\delta$

Output: nuevasAristas

```
1: nuevasAristas = {}
2: for all nodo  $\in$  Grafo do
3:   calcular grado del nodo
4: end for
5: while costo de la red  $< P$  do
6:   seleccionar (a,b) los dos nodos con menor grado
7:   if costo de la red +  $C(a,b) < P$  then
8:     Agregar arista a nuevasAristas
9:   else
10:    return nuevasAristas
11:   end if
12: end while
13: return nuevasAristas
```

La segunda técnica, si bien similar a la primera, considera la probabilidad de desconexión de un nodo de la red, calculado como el producto de los pesos de las aristas adyacentes a el, de esta forma, se conectarán nodos que tengan las probabilidades de quedar desconectados mas altas. También se presenta su pseudocódigo.

Input: Grafo G , Presupuesto P , Costos $C\delta$

Output: nuevasAristas

```
1: nuevasAristas = {}
2: for all nodo  $\in$  Grafo do
3:   calcular probabilidad de desconexión del nodo
4: end for
5: while costo de la red  $< P$  do
6:   seleccionar (a,b) los dos nodos con mayor probabilidad de desconexión
7:   if costo de la red +  $C(a,b) < P$  then
8:     Agregar arista a nuevasAristas
9:   else
10:    return nuevasAristas
11:   end if
12: end while
13: return nuevasAristas
```

C. Métricas

Se presentaran las siguientes métricas

- Valores promedio del fitness de las soluciones obtenidas en cada ejecución.
- Promedio del numero de generaciones necesarias para alcanzar el mejor resultado obtenido, así como su desvío e intervalo de confianza normal.
- Comparativa con los resultados obtenidos de los algoritmos greedy.
- Métricas relacionadas con la eficiencia computacional, media y desvío de esta métrica, así como los detalles de la plataforma utilizada para la ejecución (CPU, memoria, sistema operativo, características de red, lenguaje de desarrollo y bibliotecas utilizadas).

V. ENTORNO DE DESARROLLO

El proyecto fue desarrollado en Eclipse y compilado en java 16. Se utilizaron las librerías de JMetal versión 5.10.

VI. EVALUACIÓN EXPERIMENTAL

Como se menciona anteriormente, para la prueba del algoritmo se evaluó el mismo con datos de la realidad utilizando los datos de la IEEE de redes de distribución de pruebas para estudios basados en confiabilidad, en esta sección serán utilizadas las redes eléctricas de tamaño 24 y 54.

Se fijaron los siguientes parámetros basados en corridas preliminares:

- $\alpha = 0.5$
- $\beta = 0.5$
- El porcentaje de población greedy fue fijado en 30%
- Máximo de evaluaciones en 50.000 para el caso de 24 nodos y 30.000 para el caso de 54 nodos.
- El método de cruzamiento seleccionado es UX⁴.

El presupuesto asignado a la red de 24 nodos es de 2 millones de dolares y para la red de 54 nodos se disponen de 4 millones, si bien estos valores se eligieron arbitrariamente, fuerzan a la evolución a buscar mejores aristas, pues si se dispusiera de un presupuesto mas alto, agregar aristas entre cualquier par de nodos mejoraría mucho la confiabilidad.

Los datos sobre el costo por kilometro de cable fueron extraídos de [3] y se utilizaron los costos internacionales presentados en dicho informe, estos son, de 87.310 dolares por kilometro de cable.⁵

Para los costos de las aristas que no están presentes en el grafo original se tomo la mínima distancia entre cada par de nodos utilizando el conocido algoritmo de Dijkstra y se multiplico el valor obtenido por 0.7 a modo de representar que agregar una nueva arista entre un par de nodos que no estén directamente conectados es mas económico que el recorrido previo. Esta decisión se tomo pues no se dispone de los costos de aristas no presentes.

⁴Se eligió UX sobre 2PX pues luego de corridas preliminares se obtuvieron resultados más prometedores.

⁵Esto no incluye costos de instalación.

A. Configuración paramétrica

En esta etapa se busca analizar y comparar el desempeño del algoritmo con distintas combinaciones de valores para los parámetros de mutación. Estos parámetros son el tamaño de población, la probabilidad de cruzamiento y la probabilidad de mutación. Los valores considerados para cada parámetros son:

- Tamaño de población: 20, 80, 150
- Cruzamiento: 0.6, 0.85, 0.95
- Mutación: 0.01, 0.001

Esto genera un total de 18 combinaciones posibles, cada combinación se ejecuta 20 veces. Las medidas muestrales que se toman en cuenta durante el proceso de configuración son la media y la desviación estándar. También se realizó el test de Kolmogorov-Smirnov (K-S de aquí en mas) para contrastar si el conjunto de datos se ajustan o no a una distribución normal.

Las tablas que se presentan en XII resumen los experimentos propuestos anteriormente, donde se puede apreciar para cada configuración perimétrica, el valor de fitness medio, el desvío estándar, el promedio del porcentaje de presupuesto usado así como la confiabilidad y el p-valor, se omiten los valores del estimador del test de K-S que se detallaran en la siguiente sección.

B. Test de normalidad

La prueba de Kolmogórov-Smirnov (también prueba K-S) es una prueba no paramétrica que determina la bondad de ajuste de dos distribuciones de probabilidad entre sí.

- Si $D \leq D_\alpha \Rightarrow \text{Aceptar } H_0$
- Si $D > D_\alpha \Rightarrow \text{Rechazar } H_0$

Donde el valor D_α se elige de tal manera que

$P(\text{Rechazar } H_0 / H_0 \text{ es cierta}) = P(D > D_\alpha / \text{Los datos siguen la distribución } M) = \alpha$

Para este caso, se toma $H_0 = \text{"La distribución de los valores de fitness es normal"}$

Se considera la siguiente tabla de valores de D_α para 20 muestras.

α	D_α
0.001	0.42085
0.0100	0.3524
0.02	0.32866
0.05	0.29407
0.1	0.26473
0.15	0.24587
0.2	0.23152

TABLE I: D_α

Con los experimentos realizados para el grafo de 24 nodos, no es conveniente rechazar H_0 para los valores de $D_{0.001}$, $D_{0.01}$, $D_{0.02}$ ni $D_{0.05}$. Esto pues, el valor mas alto del estimador de K-S es 0.2773.

Así mismo, para el caso de 54 nodos sucede que no es conveniente rechazar H_0 tampoco en los casos $D_{0.1}$, $D_{0.15}$ ni $D_{0.2}$. El valor mas grande del estimador es 0.1924

Por lo que se podría concluir que con gran probabilidad, los datos provienen de distribuciones normales.

C. Confiabilidades originales

Para poder estudiar que tanto mejora la confiabilidad de las redes, es necesario estimar la confiabilidad de dichas redes antes de la actualización. Para esto, se realizo un muestreo utilizando el método de Monte Carlo especificado en secciones anteriores, con la siguiente configuración perimétrica:

- $N = 2.649.158$ muestras
- $\delta = 0.01$
- $\epsilon = 0.001$

Para el grafo de 24 nodos se obtiene el estimador puntual de la confiabilidad de 0.7839 y el IDC de Chevyshev (0.7837, 0.7841).

En el caso del grafo de 54 nodos se tiene un estimador puntual de 0.1569 y el IDC (0.1567, 0.1571).

Por ultimo, la red de 118 nodos tiene un estimador puntual de 0.5961 y su IDC de Chevyshev es (0.5960, 0.5962).

D. Resultados greedy

Tal como se menciono anteriormente, los resultados serán comparados con las técnicas greedy propuestas en la sección IV-B. Para esto, se corrieron veinte experimentos para cada red, con el primer algoritmo, donde se resume media y desvío de cada experimento en la tabla a continuación:

Nodos	Media	Desvío
24	0.9312	0.0301
54	0.2991	0.0831
118	0.7364	0.1080

TABLE II: Resultados Greedy

Para la red de 24 nodos, la técnica greedy planteada mejora la confiabilidad de la red en 1.187 veces y para la red de 54 nodos se obtiene una mejora de 90.630%. Del mismo modo para la red de 118, se tiene una mejora del 23.532 %⁶

La siguiente tabla condensa los resultados de la aplicación del segundo algoritmo Greedy planteado en IV-B, el cual, a diferencia del primero, es determinista.

Nodos	Presupuesto (%)	Confiabilidad
24	75.907	0.9383
54	97.564	0.3545
118	89.671	0.8708

TABLE III: Resultados Segundo Greedy

Para estimar ambas confiabilidades se consideraron los siguientes parámetros del método de Monte Carlo:

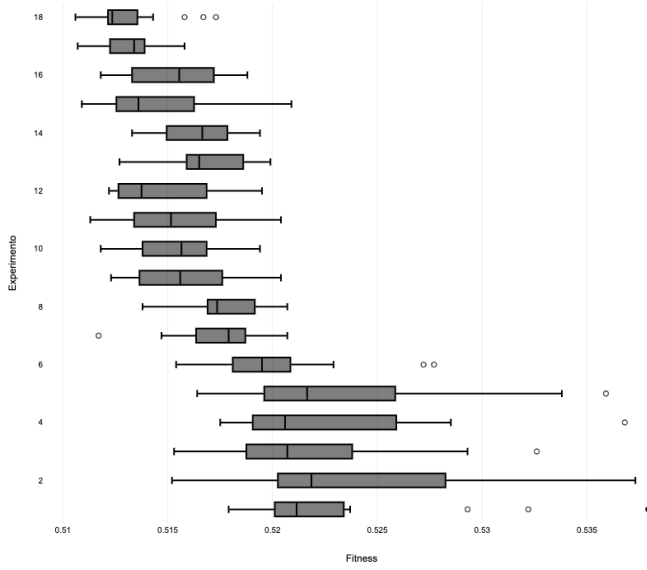
- $\epsilon = 0.001$
- $\delta = 0.01$
- $N = 2.649.158$ ⁷

⁶Nos interesa solamente comparar el valor de la confiabilidad para el caso greedy y el resultado del algoritmo genético.

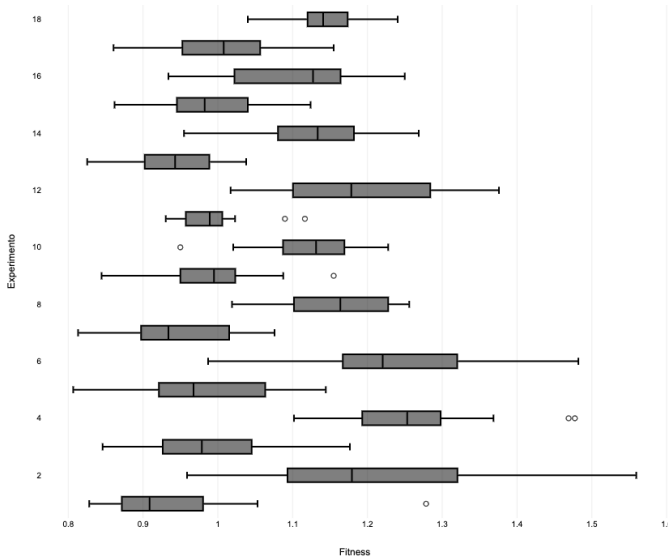
⁷Este valor se obtiene del teorema de Hoeffding planteado en III-D2

E. Análisis de resultados

El próximo gráfico resume los valores de fitness de cada experimento en un Boxplot para cada configuración paramétrica, primero el caso de 24 nodos:



Del mismo modo, el de 54 nodos:



Para el caso de 24 nodos, es claro que los mejores valores de fitness son los de los experimentos 17 y 18, dado que son muy similares sus distribuciones, resta estudiar si existen diferencias significativas entre ellas. Para esto se utilizara el test de hipótesis de ANOVA [7]. De este test se obtiene que el que no hay diferencias significativas para valores $p < 0.10$. Por lo que no podemos asegurar que las distribuciones sean distintas.

Es necesario entonces poder decidir que configuración paramétrica es mejor, para esto, nos basaremos en el tiempo

promedio de computo, así como el promedio de generaciones hasta el mejor fitness.

Para el 17 se tiene que el promedio de generaciones al mejor fitness es 155 mientras que para el 18 es 190, y los tiempos del 17 son en promedio 20.89 % mas cortos. Por lo que se elegirá el 17 como mejor configuración.

El tiempo medio para el experimento 17 es de 13 minutos aproximadamente.

De las tablas y boxplots presentados anteriormente, se tiene que para el gafo de 24 nodos, la combinación de los parámetros de evolución que mejores valores de fitness produce es:

- Tamaño de población: 150 individuos
- Probabilidad de cruzamiento: 0.95 %
- Probabilidad de mutación: 0.001 %

El cual genero un valor de fitness de 0.5132 con un desvío de 0.0013. Dicho valor de fitness se alcanzo en promedio en la generación 155, Utiliza un 91.1711% del presupuesto y logra una confiabilidad de 0.9813. El valor de confiabilidad es 25.1818% mejor que la confiabilidad original (0.7839) y 5.3802% mejor que el primer algoritmo greedy (G1) y 4.5828% mejor que el segundo algoritmo greedy (G2).

Para el grafo de 54 nodos, los parámetros que logran un mejor valor de fitness son:

- Tamaño de población: 20 individuos
- Probabilidad de cruzamiento: 0.6 %
- Probabilidad de mutación: 0.001 %

El cual genero un valor de fitness de 0.9393 con un desvío de 0.1016. Dicho valor de fitness se alcanzo en la generación 947, Utiliza un 98.182% del presupuesto y logra una confiabilidad de 0.4888. La nueva de confiabilidad es 3.11 veces mejor que la confiabilidad original (0.1569) y 63.423% mejor que el promedio de los greedy.

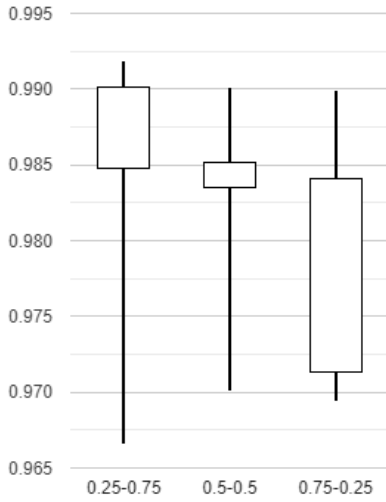
El tiempo promedio de las ejecuciones de esta configuración paramétrica es de 8 minutos aproximadamente⁸.

Tomando en cuenta los parámetros que logran una mejor evolución, se procede a realizar experimentos modificando los valores de α y β de la función fitness. Las siguientes tablas y boxplots resumen los resultados obtenidos.

α	β	Presupuesto	Confiabilidad
0.25	0.75	97.4465	0.9855
0.50	0.50	96.0408	0.9832
0.75	0.25	67.7087	0.9781

TABLE IV: Experimento α - β 24 Nodos

⁸La cantidad de evaluaciones para la red de 24 nodos y de 54 son distintas, así como la plataforma de ejecución, así se explica la diferencia de tiempos.

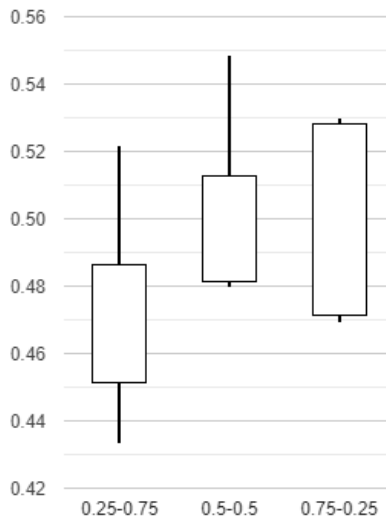


Para el caso de 24 Nodos, no se observan diferencias significativas entre las confiabilidades en ninguna de las variaciones, sin embargo, para el presupuesto se observan valores notablemente mas bajos a costa de una pequeña disminución de la confiabilidad. Quien vaya a realizar la actualización de la red, sera quien deba definir si esta dispuesto a aceptar una pequeña reducción en la confiabilidad en pos de un ahorro sustancial de presupuesto.

Se plantea lo mismo para la red de 54 nodos

α	β	Presupuesto	Confiabilidad
0.25	0.75	99.1977	0.4613
0.50	0.50	98.6342	0.4946
0.75	0.25	98.6849	0.4993

TABLE V: Experimento α - β 54 nodos



En este caso, no se observan mejoras significativas en ninguno de los parámetros estudiados.

El ultimo experimento a realizar es para la red de 118 nodos con la siguiente configuración perimétrica:

- Población: 150 individuos

- Probabilidad de cruzamiento: 0.6
- Probabilidad de mutación: 0.001
- $\alpha = \beta = 0.5$

De estos se obtiene un promedio de fitness 0.5311 con un desvío de 0.0040, el promedio de generaciones hasta alcanzar el mejor fitness es de 235, la media del porcentaje de presupuesto usado es 96.545 y la de confiabilidad 0.9344, lo que supone una mejora del 26.288 % sobre el primer greedy y del 7.303 % sobre el segundo.

VII. PLATAFORMAS DE EJECUCIÓN

Para la obtención de los datos experimentales se dispuso de dos plataformas de ejecución, la primera cuenta con un procesador i7-9700k @ 4.2 GHz con 32 GB de memoria RAM a 3200 MHz corriendo Windows 10, y la segunda cuenta con un procesador 2 GHz Dual-Core Intel Core i5 con 8 GB de memoria RAM en sistema operativo macOS Monterey.

VIII. CONCLUSIONES

En este informe se trabajó sobre una red eléctrica de 24 y 54 nodos, debido a la complejidad computacional que acarrea el grafo de 118 nodos extraído de la base de datos de la IEEE [2], ocasionalmente se hizo mención de la red mas grande.

La generación de individuos greedy en la población inicial pone un punto de partida bastante alto para las soluciones, ahorrando mucho tiempo de computo, a coste de una menor diversidad⁹.

Como se puede apreciar en la sección VI-E y en la tabla de esta misma sección, los resultados de la evolución del algoritmo genético es notablemente mejor que la obtenida por los greedy. Para la red de 24 nodos se obtiene una mejora de al menos el 4.64 % y de 7.32 % para el caso de la red de 54. El motivo por el que esto puede suceder es estudiado en XI.

Los mejores resultados son los que se muestran en la tabla a continuación, donde se hace también una comparación entre las confiabilidades originales y de los distintos algoritmos Greedys.

Red	R Original	R Nuevo	Mejora	G1	G2
24	0.7839	0.9818	25.25%	5.44%	4.64%
54	0.1569	0.4888	211.54%	63.42%	37.88%
118	0.5961	0.9344	56.75%	26.89%	7.32%

TABLE VI: Comparación de métodos

En este problema el tiempo de ejecución del algoritmo no era una condicionante. De todas formas generamos un registro del mismo, como se aprecia en la sección VI-E el tiempo de ejecución del algoritmo para el grafo de 24 nodos requirió un tiempo de computo mayor al grafo de 54 nodos. Esto se debe a que los criterios de parada son distintos, para el grafo de 24 nodos se utilizo un máximo de 50.000 evaluaciones y para el grafo de 54 nodo se utilizo un máximo de 30.000. Las plataformas de ejecución también influyen en los valores del tiempo de ejecución del algoritmo.

⁹Para mantener una diversidad mas alta solo se inicializo con un máximo de 30% de población greedy

Como se menciono en la sección IV, los valores de alfa y beta no fueron analizado pues se entiende que quien lleve a cabo la actualización de la red eléctrica es quien debe decidir si esta dispuesto a sacrificar confiabilidad a favor de un ahorro sustancial de presupuesto.

En la siguiente sección se presentaran mejoras que se entienden podrían aportar valor a una futura implementación de esta técnica.

IX. MEJORAS A FUTURO

El trabajo presentado podría ser mejorado de diversas formas, una de ellas es mejorar el muestreo por Monte Carlo, aplicando métodos de reducción de varianza como muestreo antitético o reducciones serie-paralelo como se menciona en [5]. Esto a modo de mejorar el tiempo que toma la evaluación de cada individuo así como la precisión de dicha evaluación.

Otro aspecto en el que se podría mejorar es la paralelización de la evaluación de individuos, al ser una función tan costosa, en este caso, seria muy beneficioso explorar una estructura de procesamiento paralela.

Para contar con mejor entendimiento de las condiciones del problema también se podría hacer experimentos con diferentes valores de presupuesto a modo de "muestreo" del frente de Pareto.

Del mismo modo se podría aplicar en vez de un algoritmo mono-objetivo uno multi-objetivo, todos los objetivos tienen la misma importancia por lo que se podría utilizar uno como NSGA-II.

X. IMPLEMENTACIÓN

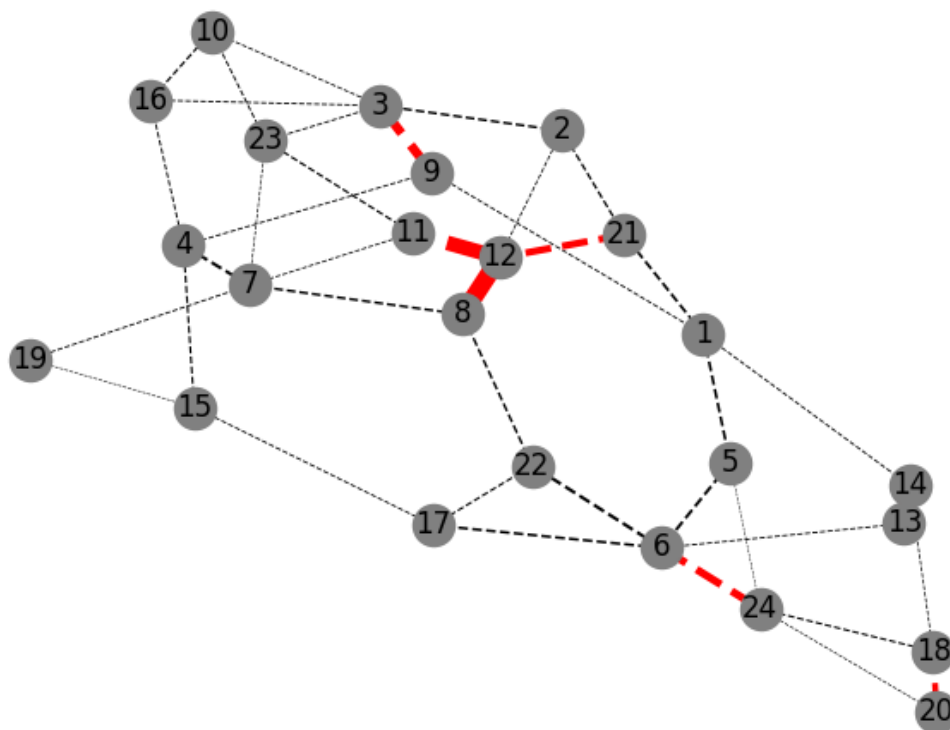
Se puede consultar el código resultado de la implementación de este método en [8] así como los resultados de las ejecuciones.

REFERENCES

- [1] R.-H. Jan, F.-J. Hwang, and S.-T. Chen, "Topological optimization of a communication network subject to a reliability constraint," *IEEE Transactions on Reliability*, Vol. 42 (1993), 63-70
- [2] Mohammad Jooshaki, Matti Lehtonen, Mahmud Fotuhi-Firuzabad, Gregorio Muñoz-Delgado, Javier Contreras, José M. Arroyo, September 7, 2020, "Test Distribution Networks for Reliability-Based Studies", *IEEE Dataport*, doi: <https://dx.doi.org/10.21227/tj2p-6t88>.
- [3] Mercados Energeticos Consultores, Diciembre, 2012, "Remuneración Anual de Redes de Transición y Substancias eléctricas, y sus formas de actualización", doi: <https://acortar.link/FG0QEs>.
- [4] Wadsworth, G. P. (1960). *Introduction to Probability and Random Variables*. New York: McGraw-Hill. p. 52.
- [5] Héctor Cancela, Evaluación por métodos de Monte Carlo de medidas de confiabilidad de redes. p. 20.
- [6] AE para la optimización multiobjetivo. Del material teórico del curso. p. 15.
- [7] Test de ANOVA, Universidad de Valencia, Ultimo acceso 1/12/2022 doi: <https://www.uv.es/montes/biomecanica2004/anova>
- [8] Repositorio de GitHub de este proyecto <https://github.com/Federico-Gil/EA-Network-Optimization> (solicitar acceso).

XI. APÉNDICE I

El siguiente diagrama representa la red de 24 nodos, las aristas negras punteadas representan las conexiones existentes y su grosor representa la probabilidad de fallo de dichas aristas (cuanto más gruesa más probabilidad de fallar), las aristas rojas son las agregadas por un individuo aleatorio con la mejor configuración perimétrica definida en VI-E.



Se puede ver que el algoritmo no necesariamente toma las aristas con mas probabilidad de fallar (o con menos adyacentes) y los conecta entre ellos, sino que tiene en cuenta el costo de conectar nodos que distan. De esta manera es que el algoritmo genético logra una mejora respecto a los greedy planteados. Notese que las aristas nuevas se encuentran escaladas x10 a modo de ser mas visual la representación.

Las aristas que agrega el individuo son:

N1	N2	P-Fallo ¹⁰	Distancia
3	9	0.1295	2.59
11	12	0.206	4.12
8	12	0.252	5.04
12	21	0.098	1.96
6	24	0.1085	2.17
18	20	0.084	1.68

TABLE VII: Tabla de nuevas aristas

XII. APÉNDICE II

En esta sección se presentan las tablas que contienen los resultados de las ejecuciones de las diversas configuraciones paramétricas.

Población	Mutación	Cruzamiento	Media Fitness	Desvío Fitness	% Presupuesto	Confiabilidad	p-valor
20	0.6	0.001	0.5228	0.0049	95.3791	0.9535	0.0751
20	0.6	0.01	0.5235	0.0054	97.4693	0.9529	0.5383
20	0.85	0.001	0.5217	0.0043	94.2209	0.9562	0.7725
20	0.85	0.01	0.5226	0.0050	97.3134	0.9546	0.2535
20	0.95	0.001	0.5235	0.0057	94.1048	0.9513	0.4047
20	0.95	0.01	0.5200	0.0032	96.7878	0.9617	0.2559
80	0.6	0.001	0.5175	0.0021	93.8481	0.9672	0.6132
80	0.6	0.01	0.5176	0.0017	98.0835	0.9684	0.7855
80	0.85	0.001	0.5158	0.0024	92.8580	0.9736	0.9218
80	0.85	0.01	0.5156	0.0022	96.5005	0.9750	0.9418
80	0.95	0.001	0.5155	0.0026	94.2728	0.9774	0.9708
80	0.95	0.01	0.5148	0.0024	95.0459	0.9774	0.2208
150	0.6	0.001	0.5169	0.0020	94.6518	0.9705	0.6049
150	0.6	0.01	0.5164	0.0018	95.4035	0.9724	0.8946
150	0.85	0.001	0.5145	0.0027	92.1673	0.9769	0.5852

TABLE VIII: 24 Nodos

Población	Mutación	Cruzamiento	Media Fitness	Desvío Fitness	% Presupuesto	Confiabilidad	p-valor
20	0.6	0.001	0.9393	0.1016	98.1830	0.4888	0.6867
20	0.6	0.01	1.2111	0.1494	94.5690	0.4176	0.9172
20	0.85	0.001	0.9880	0.0855	98.7538	0.4630	0.7719
20	0.85	0.01	1.2621	0.1067	89.9126	0.3996	0.7034
20	0.95	0.001	0.9815	0.0986	98.6193	0.4665	0.3989
20	0.95	0.01	1.2403	0.1232	86.6898	0.4031	0.8411
80	0.6	0.001	0.9487	0.0752	98.4366	0.4884	0.8775
80	0.6	0.01	1.1559	0.0767	87.4855	0.4247	0.9138
80	0.85	0.001	0.9879	0.0758	98.1126	0.4715	0.9224
80	0.85	0.01	1.1225	0.0693	88.2867	0.4440	0.9204
80	0.95	0.001	0.9912	0.0470	98.0782	0.4723	0.5859
80	0.95	0.01	1.1855	0.1017	87.5099	0.4200	0.9318
150	0.6	0.001	0.9425	0.0621	98.4770	0.4971	0.8994
150	0.6	0.01	1.1329	0.0728	84.7787	0.4400	0.9099
150	0.85	0.001	0.9932	0.0693	98.1804	0.4774	0.8752
150	0.85	0.01	1.1040	0.0889	88.8551	0.4453	0.6666
150	0.95	0.001	1.0058	0.0767	97.5129	0.4765	0.8962
150	0.95	0.01	1.1414	0.0513	86.3654	0.4315	0.8673

TABLE IX: 54 Nodos