Universidad Nacional Autónoma de México

Facultad de Ciencias

Construcción de Testigos de Enredamiento usando Algoritmos Genéticos

T E S I S

que para obtener el grado de:
físico

presenta:
Diego Alberto Olvera Millán

director de tesis:
Dr. Pablo Barberis Blostein

CDMX

Marzo, 2019

Construcción de Testigos de Enredamiento usando Algoritmos Genéticos

por

Diego Alberto Olvera Millán

Tesis presentada para obtener el grado de

FÍSICO

en la

Facultad de Ciencias

Universidad Nacional Autónoma de México

CDMX. Marzo, 2019

Agradecimientos

Aquí van a ir los agradecimientos

Índice general

Construcción de Testigos de Enredamiento usando Algoritmos Genéticos

por

Diego Alberto Olvera Millán

Resumen

En esta tesis se construyen testigos de enredamiento para sistemas de 2 qubits mediante el uso de algoritmos genéticos (GA). Se usan tres algoritmos en los que los operadores de generación de población inicial, selección, cruzamiento y mutación son iguales pero la función de fitness es distinta. Los elementos sobre los cuáles actúa el GA son matrices Hermitianas, W. Las tres funciones de fitness se evalúan calculando los valores esperados de los operadores en un conjunto fijo de estados separables generados aleatoriamente. Se restan puntos de fitness a un operador W cuando ocurre que este valor es negativo. En el primer GA se dan puntos positivos de fitness a un operador cuando ocurre que el valor esperado para los estados de Bell es negativo. En el segundo se generan matrices unitarias aleatorias y se aplican a los estados de Bell para generar una familia de 14 estados enredados aleatorios y se dan puntos positivos a W cuando el valor esperado para alguno de estos estados es negativo. En el tercero se aplican las matrices unitarias aleatorias a estados de Werner de bajo enredamiento (de acuerdo a la medida de enredamiento de formación) para generar una familia de 14 estados débilmente enredados y con estos se dan los puntos positivos de fitness. Se logró generar testigos de enredamiento válidos con los 3 algoritmos, en particular el segundo y tercer GA generan testigos que son mejores que un testigo óptimo conocido, W_o, para detectar estados débilmente enredados. Se encontró que para intervalos pequeños de enredamiento de formación los testigos generados detectan bastantes más estados enredados que W_o. Se generaron 500,000 estados aleatorios y se separaron en enredados y separables de acuerdo al criterio PPT. Combinando los testigos que generaron los 3 GA's se lograron detectar 122,552 estados de los 412,169 posibles ($\sim$29 %) mientras que W_o detecta 36,985 ($\sim$9 %).

Introducción

El enredamiento cuántico, descrito por primera vez por Einstein, Podolsky y Rosen (EPR) en 1935 (Einstein *et al.* [1935]), es un fenómeno en el que podemos observar correlaciones entre sistemas cuánticos, las cuales no tienen ningún análogo clásico. Aunque en su momento el enredamiento se quiso entender como una prueba de la incompletitud de la mecánica cuántica, en 1964 Bell formalizó la idea de EPR en un modelo de variables ocultas (Bell [1964]). Bell probó que asumir realismo, localidad y libre albedrío en mecánica cuántica resulta en correlaciones estadísticas en sistemas bipartitos para las cuales se deben cumplir ciertas desigualdades. En 1981, Aspect *et al.* [1981, 1982] probaron que las desigualdades de Bell se violan (aunque el experimento tiene una laguna, en la actualidad existen experimentos libres de lagunas que demuestran que las desigualdades se violan (Hensen *et al.* [2015]), lo cual implica que no se puede completar a la mecánica cuántica por medio de un modelo de variables ocultas.

Desde entonces y con el advenimiento de la teoría de información cuántica, se ha reconocido al enredamiento como un recurso en el cómputo cuántico (Mermin [2007]). Por este motivo se volvió de vital importancia su estudio. En este sentido son importantes varias preguntas: ¿Cómo puede uno producir un estado enredado? ¿Cómo se puede determinar si un estado está enredado? ¿Existen medidas y tipos de enredamiento?

En esta tesis se aborda la segunda de estas preguntas y para contestarla se usan los así llamados testigos de enredamiento. Un testigo de enredamiento es un operador Hermitiano con la propiedad de que el valor esperado para estados separables es positivo y será negativo para al menos un estado enredado. Los testigos son muy importantes, ya que, al ser observables, son realizables en un laboratorio y no requieren conocer completamente el sistema cuántico de interés, es decir, no es necesario conocer el operador de densidad de los estados.

El problema es construir un testigo de enredamiento y se ha demostrado que este problema es equivalente al problema de separabilidad y es NP-difícil (Gharibian [2008]).

En esta tesis se propone una forma de construir testigos de enredamiento para sistemas de dos qubits por medio de algoritmos genéticos. Estos son una clase de técnicas estocásticas basadas en el concepto de evolución Darwiniana (Holland [1975]).

Los algoritmos genéticos han sido usados en el contexto del enredamiento cuántico para encontrar el mínimo en las construcciones de techo convexo (Ramos y Souza [2002]) y para optimizar la concurrencia en sistemas cuánticos reticulares (Navarro-Muñoz *et al.* [2006]). Además de esto se han usado para resolver algunos problemas de computación cuántica y se ha intentado hacer algoritmos genéticos cuánticos (Giraldi *et al.* [2004]). Hasta donde sé, el uso de GA's para la construcción de testigos de enredamiento es nueva y en esta tesis se logró demostrar que definiendo funciones de fitness sencillas y operadores de selección y mutación básicos se logra construir buenos testigos.

El objetivo principal de la tesis es demostrar que es posible construir testigos de enredamiento usando algoritmos genéticos.

Plan de la tesis

En el capítulo 1 se hace un resumen de los conceptos básicos de enredamiento cuántico. Se presenta el problema de separabilidad, así como criterios y medidas de enredamiento. En el capítulo 2 se introducen los testigos de enredamiento y se exploran algunas propiedades de los mismos. El capítulo 3 está dedicado a los algoritmos genéticos. En el capítulo 4 se plantea el problema principal de la tesis, el de la construcción de testigos por medio de algoritmos genéticos, y en el capítulo 5 se analizan los resultados.

Capítulo 1

Conceptos básicos de enredamiento

En este capítulo estudiamos los conceptos escenciales sobre enredamiento que se necesitan para desarrollar la tesis. En toda la tesis se estudian *qubits*, sistemas cuánticos de dos niveles. Primero se introduce notación importante que se va a utilizar a lo largo de la tesis. Siguiendo a Chruściński y Sarbicki [2014], denotamos por $\mathcal{H}$ a un espacio de Hilbert (en esta tesis se asume que los espacios de Hilbert son de dimensión finita) en el cual viven los vectores de estado que representan al sistema cuántico de interés. Con $\mathfrak{L}(\mathcal{H})$ denotamos al espacio vectorial de operadores lineales que actúan sobre $\mathcal{H}$. Una vez que se fija una base uno puede identificar $\mathfrak{L}(\mathcal{H})$ con el espacio $M_n(\mathbb{C})$ de matrices complejas de $n \times n$ con $n = \dim(\mathcal{H})$.

Denotamos por $\mathfrak{L}_+(\mathcal{H})$ al subconjunto de operadores positivos en $\mathfrak{L}(\mathcal{H})$. También definimos el siguiente conjunto

$$\mathfrak{O}(\mathcal{H}) = \{\rho \in \mathfrak{L}_+(\mathcal{H}) | \mathrm{Tr}(\rho) = 1\}.$$

Recordamos que un *estado puro* es aquel que se puede representar como un vector en un espacio de Hilbert y un *estado mezcla* es un estado que sólo se puede representar mediante un operador en $\mathfrak{O}(\mathcal{H})$.

Las definiciones que se usan en este capítulo son tomadas del artículo de revisión de Gühne y Toth [2008] y se mezclan con alguna de la notación y comentarios de Chruściński y Sarbicki [2014] y de Horodecki *et al.* [2007].

1.1. Enredamiento

Asumamos que se tienen dos sistemas físicos. El primero lo maneja una física, Alicia y el segundo otro físico, Beto. Los estados físicos del sistema de Alicia pueden ser descritos por vectores en un espacio de Hilbert $\mathcal{H}_A$ de dimensión d_A y los estados del sistema de Beto por vectores en $\mathcal{H}_B$ de dimensión d_B. Entonces, el sistema compuesto de ambas partes se describe por medio de vectores en el espacio del producto tensorial de los dos espacios, $\mathcal{H} = \mathcal{H}_A \bigotimes \mathcal{H}_B$, que será de dimensión $d = d_A \times d_B$. Por tanto, dadas un par de bases (arbitrarias) ortonormales $\{|a_i\rangle\}_{i=1}^{d_A}$ y $\{|b_i\rangle\}_{i=1}^{d_B}$ de $\mathcal{H}_A$ y $\mathcal{H}_B$, respectivamente, cualquier vector $|\psi\rangle$ en $\mathcal{H}$ puede ser escrito como

$$|\psi\rangle = \sum_{i,j=1}^{d_A,d_B} c_{ij} \, |a_i\rangle \otimes |b_j\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B, \tag{1-1}$$

con $C = (c_{ij})$ una matriz compleja. Para simplificar la notación se suele escribir $|a\rangle \otimes |b\rangle \equiv |a\rangle|b\rangle \equiv |ab\rangle$. Con esto podemos definir la separabilidad y el enredamiento para estados puros.

Definición 1 *(Enredamiento para estados puros).*

A un estado puro, $|\psi\rangle \in \mathcal{H}$, se le llama *estado producto* o *separable* si existen estados $|\phi^A\rangle \in \mathcal{H}_A$ y $|\phi^B\rangle \in \mathcal{H}_B$ tales que

$$|\psi\rangle = |\phi^A\rangle \otimes |\phi^B\rangle. \tag{1-2}$$

Si no existen dichos vectores se dice que el estado $|\psi\rangle$ está *enredado*.

En general los sistemas cuánticos se encuentran en *estados mezcla*, es decir, sabemos que están en uno de varios posibles estados y sabemos la probabilidad de que se encuentre en uno de ellos, entonces existen números positivos $p_i \leq 1$ tales que $\sum_i p_i = 1$ y estados $|\phi_i\rangle$ que definen un *operador de densidad* dado por

$$\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|. \tag{1-3}$$

Dada una base, el operador de densidad está representado por una matriz compleja. Esta matriz es positiva semidefinida y Hermitiana. Dadas las condiciones sobre las p_i's, se cumple que $\mathrm{Tr}(\rho) = 1$, es decir, $\rho \in \mathfrak{D}(\mathcal{H}_{AB})$. Al mismo tiempo podemos decir que cualquier matriz positiva semidefinida, Hermitiana y con traza uno representa un operador de densidad y, por tanto, un

estado cuántico. Esto nos lleva a una imagen geométrica del conjunto de todos los estados como un conjunto convexo. Esto quiere decir que, dados dos estados, ρ_1 y ρ_2, su combinación convexa $\rho = \alpha\rho_1 + (1-\alpha)\rho_2$, con $\alpha \in [0,1]$, también es un estado. Esto se cumple para combinaciones de más de un estado siempre y cuando la suma de los coeficientes que acompañen a cada estado sea uno. Con esto podemos definir separabilidad y enredamiento para estados mezcla.

Definición 2 *(Enredamiento para estados mezcla).*

Sea ρ un operador de densidad para un estado compuesto. Se dice que ρ es un *estado producto* si existen estados $\rho_i^A \in \mathcal{L}_A$ de Alicia y $\rho_i^B \in \mathcal{L}_B$ de Beto y números $\{p_i | \sum_i p_i = 1\}$ tales que

$$\rho = \sum_i p_i \rho_i^A \otimes \rho_i^B. \tag{1-4}$$

De otro modo al estado ρ se le llama *enredado*.

Es importante notar que el conjunto de estados separables es convexo. Dado un estado cuántico, al problema de determinar si está enredado o no se le llama *problema de separabilidad*. Se conocen varios criterios que implican separabilidad o enredamiento en un estado dado pero, hasta ahora, no se ha encontrado una solución general al problema de separabilidad (Horodecki *et al.* [2007]).

1.2. Criterios de enredamiento

Introducimos brevemente algunos criterios de separabilidad, los cuales se implementan computacionalmente, esto para poder usarlos y poder poner a prueba los testigos que se generen por medio del GA.

1.2.1. Criterio PPT

Empezamos con el criterio de la transposición parcial. Sea $T : \mathfrak{L}(\mathcal{H}) \to \mathfrak{L}(\mathcal{H})$ la transposición con respecto a una base fija definida por

$$M^T := T(M) := \sum_i M_{ji} \left|i\right\rangle\!\left\langle j\right|, \tag{1-5}$$

donde $M = M_{ij}\,|i\rangle\langle j|$ es una matriz. Sea también $\mathrm{id}_A : \mathcal{H}_A \to \mathcal{H}_A$ la identidad del subsistema A. Se define entonces la *transpuesta parcial* como

$$\begin{aligned} &\mathrm{id}_A \otimes T : \mathfrak{L}(\mathcal{H}_{AB}) \to \mathfrak{L}(\mathcal{H}_{AB}). \\ &T \otimes \mathrm{id}_B : \mathfrak{L}(\mathcal{H}_{AB}) \to \mathfrak{L}(\mathcal{H}_{AB}). \end{aligned} \tag{1-6}$$

Ahora, notemos que, dadas las bases ortonormales $\{|i^A\rangle\}_{i=1}^N$ y $\{|j^B\rangle\}_{j=1}^M$ de los sistemas $\mathcal{H}_A$ y $\mathcal{H}_B$, respectivamente, podemos expandir el operador de densidad de un sistema cuántico como

$$\rho = \sum_{i,j}^{N}\sum_{k,l}^{M} \rho_{ij,kl}|i^A, k^B\rangle \otimes \langle j^A, l^B| \equiv \sum_{i,j}^{N}\sum_{k,l}^{M} \rho_{ij,kl}|i\rangle\langle j| \otimes |k\rangle\langle l|. \tag{1-7}$$

Entonces la *transpuesta parcial* de ρ con respecto al subsistema A está dada por

$$\rho^{T_A} := (T \otimes id_B)(\rho) = \sum_{i,j}^{N}\sum_{k,l}^{M} \rho_{ji,kl}|i^A, k^B\rangle \otimes \langle j^A, l^B| \equiv \sum_{i,j}^{N}\sum_{k,l}^{M} \rho_{ji,kl}|i\rangle\langle j| \otimes |k\rangle\langle l|. \tag{1-8}$$

Similarmente se define la transpuesta para el sistema B (T_B) intercambiando los índices k y l en lugar de i y j. Notamos que $\rho^T = (\rho^{T_A})^{T_B}$ y por tanto $\rho^{T_B} = (\rho^{T_A})^T$.

Es claro que el resultado que se obtiene depende de la base en la que se realice la descomposición de ρ, pero es posible demostrar que el espectro, los eigenvalores de ρ, de las transposiciones parciales no depende de la base. Se dice que un operador de densidad ρ tiene una *transposición parcial positiva* (PPT por sus siglas en inglés) si su transposición parcial no tiene eigenvalores negativos, i.e., es positiva semidefinida

$$\rho^{T_A} \geq 0 \iff \rho^{T_B} \geq 0, \tag{1-9}$$

i.e., si $\rho^{T_A} \in \mathfrak{D}(\mathcal{H})$.

Ahora planteamos el criterio PPT (Horodecki *et al.* [1996]):

Teorema 3 *(Criterio PPT). Sea ρ un operador de densidad de un estado separable. Entonces ρ es PPT.*

Dado este resultado es natural preguntarse si el inverso es cierto, es decir, si encontramos que $\rho^{T_A} \geq 0$, ¿podemos concluir que ρ es separable? Resulta que para sistemas de dimensión 2×2

y 2×3 se da el si y sólo si.

Teorema 4 *(Teorema de Horodecki). Si ρ representa a un estado de un sistema de dimensiones 2×2 o 2×3 entonces $\rho^{T_A} \geq 0$ implica que ρ es separable.*

En dimensiones más altas esto no ocurre. Entonces el criterio PPT caracteriza por completo el enredamiento en sistemas de 2 qubits.

1.3. Medidas de enredamiento

Empezamos definiendo a las *transofrmaciones unitarias locales*. Se dice que una transformación $U \in \mathfrak{L}(\mathcal{H})$ es unitaria si $UU^\dagger = \mathrm{id}$, donde $U^\dagger$ es el operador transpuesto conjugado a U e id es la identidad. Se le llama *transformación unitaria local* a una transformación $U \in \mathfrak{L}(\mathcal{H})$, con $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, tal que

$$U = U_A \otimes U_B \ , U_A \in \mathfrak{L}(\mathcal{H}_A) \text{ y } U_B \in \mathfrak{L}(\mathcal{H}_B). \tag{1-10}$$

Un concepto importante en información cuántica es el de *operaciones locales y comunicaciones clásicas* (LOCC, por sus siglas en inglés). Éste es un método mediante el cual se realizan operaciones locales sobre subsistemas de un sistema cuántico de varias partes y los resultados de estas operaciones se comunican mediante comunicaciones clásicas.

Las medidas de enredamiento se deben definir de tal forma que cuantifiquen la cantidad de enredamiento en un estado dado. Así, una medida de enredamiento general $E(\rho)$ debería tener necesariamente algunas propiedades que listamos a continuación.

(I) $E(\rho)$ debe hacerse cero cuando el estado ρ es separable.

(II) Una medida de enredamiento debería ser invariante bajo un cambio de base local. Esto significa que debe ser invariante bajo transformaciones unitarias locales,

$$E(\rho) = E(U_A \otimes U_B \rho U_A^\dagger \otimes U_B^\dagger).$$

(III) Debido a que el enredamiento no puede ser creado mediante LOCC es razonable pedir que $E(\rho)$ no incremente bajo dichas transformaciones. Esto es, si Λ^{LOCC} es un mapa positivo

que puede ser implementado mediante LOCC, entonces

$$E[\Lambda^{\mathrm{LOCC}}(\rho)] \leq E(\rho).$$

A menudo se sustituye esta condición por el requerimiento de que $E(\rho)$ no aumente en promedio bajo LOCC. Esto es, si una transformación LOCC mapea a ρ a estados ρ_k con probabilidades p_k, entonces

$$\sum_k E(\rho_k) \leq E(\rho).$$

(IV) Se requiere que la medida sea convexa, esto es, el enredamiento debe disminuir si se mezclan dos o más estados

$$E(\sum_k p_k \rho_k) \leq \sum_k p_k E(\rho_k).$$

Esta desigualdad expresa el hecho de que si uno empieza con un ensemble de estados ρ_k y uno pierde información sobre la instancia única ρ_k entonces el enredamiento disminuye.

1.3.1. Concurrencia y enredamiento de formación

Hay varias formas de obtener medidas de enredamiento. Una de ellas empieza imponiendo una medida, E, en estados puros y se extiende a estados mezcla por medio del techo convexo

$$E(\rho) = \inf \sum_i p_i E(\psi_i), \quad \sum_i p_i = 1, \quad p_i \geq 0,$$

donde el ínfimo se toma sobre todos los posibles ensambles $\{p_i, \psi_i\}$ para los cuales ρ se puede escribir como (1-3). El ínfimo se alcanza para algún ensamble en particular y E es igual al promedio sobre ese ensamble.

El *enredamiento de formación* (EoF, por sus siglas en inglés) fue la primera medida de enredamiento así construida. Cuantifica cuántos estados de Bell se necesitan por copia para generar muchas copias de un estado ρ usando LOCC. Se define como sigue (Bennett *et al.* [1996]): consideremos un sistema conjunto de Alicia y Beto. Para estados puros el enredamiento

de formación, E, se define como la entropía de cualquiera de los dos subsistemas:

$$E(|\psi\rangle) = -Tr[\rho_A \log(\rho_A)] = -Tr[\rho_B \log(\rho_B)], \tag{1-11}$$

donde ρ_α es la *traza parcial* de $|\psi\rangle\langle\psi|$ sobre el subsistema α ($\alpha \in \{A, B\}$), dada por:

$$\mathrm{Tr}_B(\rho) := \rho_A = \sum_{i,j}^{N} \sum_{k,l}^{M} \rho_{ij,kl} \, |i\rangle\langle j| \, \langle k|l\rangle \,,$$

donde ρ está dada como en la ecuacion (1-7). El enredamiento de formación se define como el techo convexo:

$$E(\rho) = \min \sum_i p_i E(\psi_i). \tag{1-12}$$

En el caso de 2 qubits se puede encontrar una fórmula cerrada para calcular esta cantidad, pero antes debemos introducir otra, la *concurrencia*, definida para estados puros como (Hill y Wootters [1997])

$$C(|\psi\rangle) = \sqrt{2[1 - \mathrm{Tr}(\rho_A^2)]}.$$

Ahora introducimos la operación de cambio de spin

$$\left|\tilde{\psi}\right\rangle = \sigma_y \, |\psi^*\rangle \,, \tag{1-13}$$

donde $|\psi^*\rangle$ es el conjugado complejo de $|\psi\rangle$ cuando se expresa en una base fija y σ_y es la matriz de Pauli expresada en esa misma base. Para un estado arbitrario de dos qubits el estado con el spin cambiado es

$$\tilde{\rho} = (\sigma_y \otimes \sigma_y)\rho^*(\sigma_y \otimes \sigma_y). \tag{1-14}$$

Ahora definimos la siguiente matriz:

$$X(\rho) = \sqrt{\sqrt{\rho}\tilde{\rho}\sqrt{\rho}}. \tag{1-15}$$

Se demostró en Hill y Wootters [1997] que la concurrencia para un estado arbitrario está dada entonces por

$$C(\rho) = \max\{0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4\}, \tag{1-16}$$

donde λ_i son los eigenvalores ordenados en orden decreciente de la matriz $X(\rho)$. Con esto podemos calcular la entropía de formación para un estado arbitrario de dos qubits como (Wootters [1998])

$$E(\rho) = h\Big[\frac{1+\sqrt{1-C^2(\rho)}}{2}\Big], \tag{1-17}$$

donde $h(p) = -p\log(p) - (1-p)\log(1-p)$ es la función de entropía binaria y el logaritmo se calcula en base 2. Con esto podemos implementar una medida de enredamiento computacionalmente. Se menciona en Wootters [1998] que las λ_i's de (1-14) pueden ser las raíces cuadradas de los eigenvalores de la matriz ρX. Esto es más fácil de implementar, así que es lo que se usa en esta tesis.

Capítulo 2

Testigos de enredamiento

Los criterios de separabilidad mencionados tienen algo en común: para aplicarlos se asume que se conoce la matriz de densidad. Afortunadamente existe otro tipo de criterio basado en observables medibles. A estas observables se les conoce como *testigos de enredamiento*. Estos operadores tienen la propiedad de que el valor esperado para estados separables siempre es mayor que cero, pero es negativo para algunos estados enredados. Se dice que el testigo *detecta* el enredamiento de estos estados.

Este capítulo se basa en en Chruściński y Sarbicki [2014], Gühne y Toth [2008], Horodecki *et al.* [2007] y Audretsch [2007].

2.1. Definición

Primero definimos a un operador *positivo por bloques*.

Definición 5 *(Operador positivo por bloques)*

A un operador Hermitiano $W \in \mathfrak{L}(\mathcal{H}_{AB})$ se le llama *positivo por bloques* si

$$\langle\psi| \otimes \langle\phi| W |\psi\rangle \otimes |\phi\rangle \geq 0, \tag{2-1}$$

para todos los vectores producto $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H}_{AB}$.

Con esto podemos definir a un testigo de enredamiento.

Definición 6 *(Testigo de enredamiento).*

A un operador Hermitiano W se le llama *testigo de enredamiento* (o simplemente *testigo*) si es positivo por bloques pero no positivo.

Entonces, los testigos de enredamiento cumplen que

$$\begin{aligned} &\mathrm{Tr}(W\rho_s) \geq 0 \text{ para estados separables, } \rho_s \\ &\mathrm{Tr}(W\rho_e) < 0 \text{ para al menos un estado enredado, } \rho_e. \end{aligned} \tag{2-2}$$

Parte de la importancia de los testigos de enredamiento viene del siguiente teorema, demostrado por Terhal [1999].

Teorema 7 *Un estado $\rho \in \mathfrak{D}(\mathcal{H}_{AB})$ está enredado si y sólo si existe un testigo de enredamiento $W \in \mathfrak{L}(\mathcal{H}_{AB})$ tal que $Tr(W\rho) < 0$.*

Es decir, no hay estado enredado que no sea *detectado* por al menos un testigo de enredamiento.

2.2. Interpretación geométrica

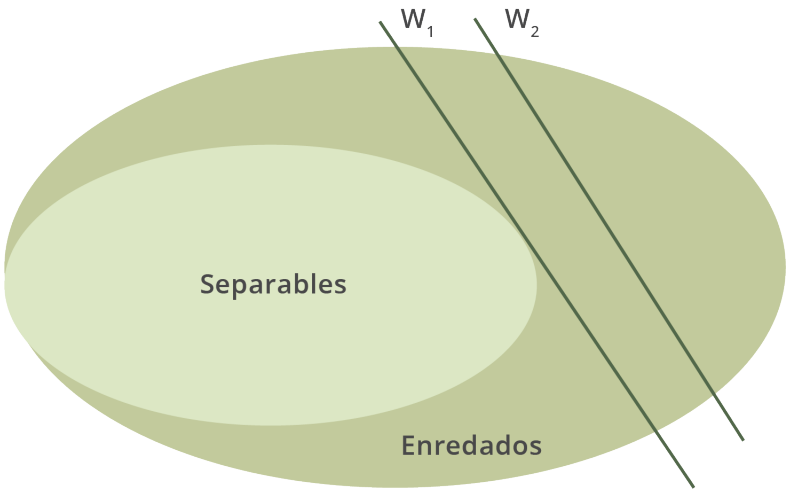


Figura 2-1: Representación esquemática del conjunto $\mathfrak{D}(\mathcal{H}_{AB})$ y la separación inducida por testigos de enredamiento.

Es importante notar que hay una interpretación geométrica de los testigos de enredamiento. El valor esperado de una observable depende linealmente con el estado. Por tanto, para un testigo de enredamiento W, el conjunto $\{\rho \in \mathfrak{D}(\mathcal{H}_{AB}) | \mathrm{Tr}(W\rho) = 0\}$ define un hiperplano en

$\mathfrak{D}(\mathcal{H}_{AB})$, cortándolo en dos partes (ver figura 2-1). En la parte para la cual $\mathrm{Tr}(W\rho) > 0$ se encuentran los estados separables y en la otra parte están los estados detectados por W.

Sea W un testigo de enredamiento. Definimos el conjunto de todos los estados enredados detectados por W como

$$D_W := \{\rho \in \mathfrak{D}(\mathcal{H}_{AB}) | \mathrm{Tr}(\rho W) < 0\}. \tag{2-3}$$

Definición 8 *(Testigo óptimo)*

Sean W_1 y W_2 dos testigos de enredamiento. Se dice que W_1 es más fino que W_2 si $D_{W_2} \subseteq D_{W_1}$. A un testigo W se le llama *óptimo* si no hay ningún otro testigo más fino que él.

En la figura 2-1 W_1 es más fino que W_2 y es óptimo.

En esta tesis se usa el siguiente testigo óptimo, tomado de Wang *et al.* [2014]:

$$W_o := \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}. \tag{2-4}$$

Vamos a usarlo para comparar los testigos que genere el algoritmo genético.

Definición 9 *(Testigos equivalentes)*

Sean W_1 y W_2 dos testigos de enredamiento. Se dice que W_1 es *equivalente* a W_2 si $D_{W_2} = D_{W_1}$.

Es claro que dos testigos que difieran por un factor contaste, $W_1 = cW_2,\ c \in \mathbb{R}$, son equivalentes. Esto se ve en la ecuación (2-2) ya que $\mathrm{Tr}(cA) = c\mathrm{Tr}(A)$.

Capítulo 3

Algoritmos genéticos

Los algoritmos genéticos (GA, por sus siglas en inglés) fueron inventados en la década de los sesentas por John Holland con la intención de estudiar formalmente los procesos de adaptación tal como ocurren en la naturaleza. Desde entonces se ha desarrollado la idea de Holland para usar a los GA's en problemas de optimización (Mitchell [1989]). Los GA's se basan en la evolución biológica. En particular se inspiran en el concepto de selección natural, al menos en su forma más simple, es decir, comenzamos con una población inicial de individuos de los cuales escogemos de manera aleatoria a algunos de los mejor adaptados para recombinarlos, dando lugar a una segunda generación. De ésta escogemos aleatoriamente a los individuos mejor adaptados y seguimos el proceso hasta que se cumpla un criterio de término. Le llamamos una corrida del algoritmo a una ejecución de este proceso. Un algoritmo así es inherentemente estocástico, de modo que los resultados que se obtienen de cada corrida del mismo pueden ser distintos a la corrida anterior.

Nos vamos a referir a cada individuo de una generación como *cromosoma*. En general, éste será un conjunto de números y a cada uno de los números de un cromosoma le llamaremos *gen*. El tamaño de los cromosomas y la cantidad de genes que pueda haber varía de problema a problema.

Ahora, la función a optimizar se conoce como *función de fitness*, cada individuo de una generación dada se evalúa con la función de fitness. Se escoge aleatoriamente a dos individuos para cruzarse y aquellos con la fitness más alta tendrán mejores probabilidades de ser escogidos. A estos individuos les llamaremos *padres*. El proceso de cruzamiento consiste en combinar los

conjuntos de números, los genes, de los dos padres (Goldberg [1999]).

Una de las mejores maneras de explicar esto es con un ejemplo, así que en lo que sigue del capítulo haremos alusión al siguiente problema: Supongamos que se quiere maximizar la siguiente función por medio de un GA

$$f(x) = 100 - (x - 10)^4 + 50(x - 10)^2 - 8x.$$

Esto es fácil de lograr con cálculo y resulta que el máximo se encuentra en $x = 5$ (ver figura 3-1), pero se puede llegar al resultado usando un algoritmo genético. Podemos usar como cromosoma la representación binaria de los números; cada gen va a ser un 1 ó un 0. Vamos a buscar el máximo entre los números 1 a 31, de modo que sólo se usan 5 bits.

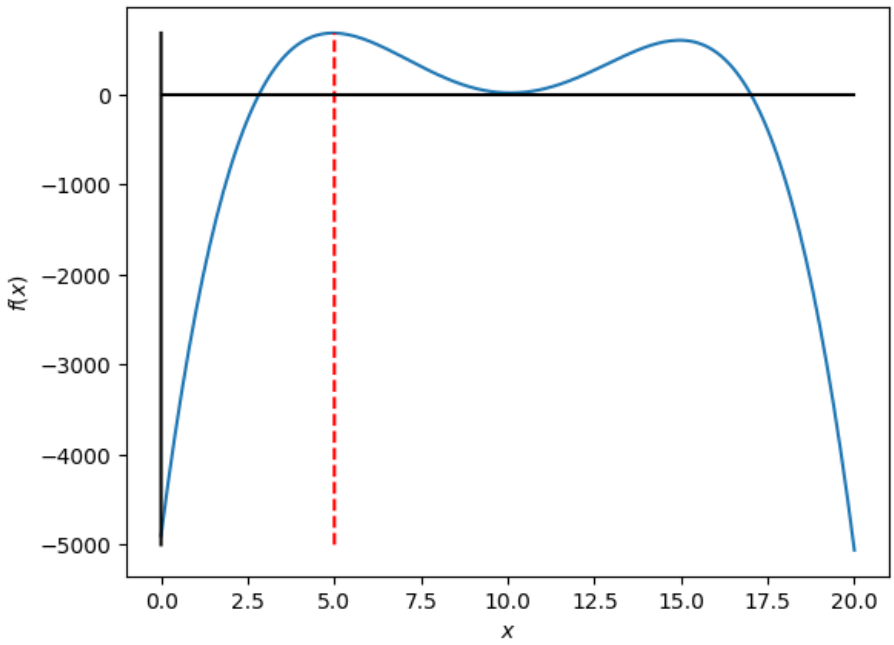


Figura 3-1: Función $f(x)$. El máximo se encuentra en x =5.

3.1. Espacio de búsqueda

Todo lo que sigue del capítulo se basa en el libro de Goldberg [1999]. Dado un problema que se intente resolver con un GA, se conoce como *espacio de búsqueda* al conjunto de posibles

soluciones al problema. Cada individuo de la población es una posible solución. Se le llama *cromosoma* al conjunto de números ordenados que representan a cada posible solución y se le llama *gen* a cada posición o a una colección de posiciones del cromosoma. Cada gen es una instancia de un alfabeto previamente escogido. Este alfabeto es distinto para cada problema y puede ser discreto o continuo. En esta tesis se usa un alfabeto de números de tipo flotante.

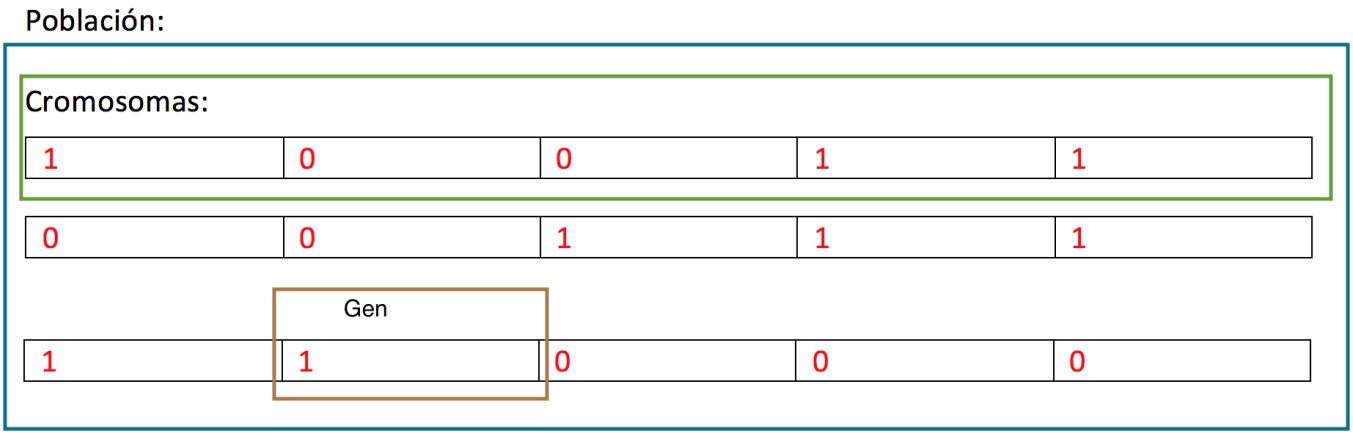


Figura 3-2: Representación esquemática de la población de un GA. En el ejemplo que se dio los genes podrían ser 1 ó 0 y el cromosoma tiene largo 5.

3.2. Selección

Cada individuo de la solución tiene asociado una fitness. Los individuos con la mejor fitness son los que se escogen con mayor probabilidad para tener descendencia. Se le llama *operador de selección* a la función que determina a los cromosomas que se han de reproducir en una generación dada. Hay varios esquemas para implementar un operador de selección: selección por torneo, selección truncada, selección proporcional, etc. En esta tesis se usa la selección proporcional, también conocida como selección por ruleta, donde la probabilidad de escoger a un individuo como padre es proporcional a su fitness.

En el ejemplo el fitness se evalúa con la misma función que se quiere maximizar. Usando los números de la figura 3-1 tendríamos que evalúar los números binarios 10011, 111 y 11000 cuya

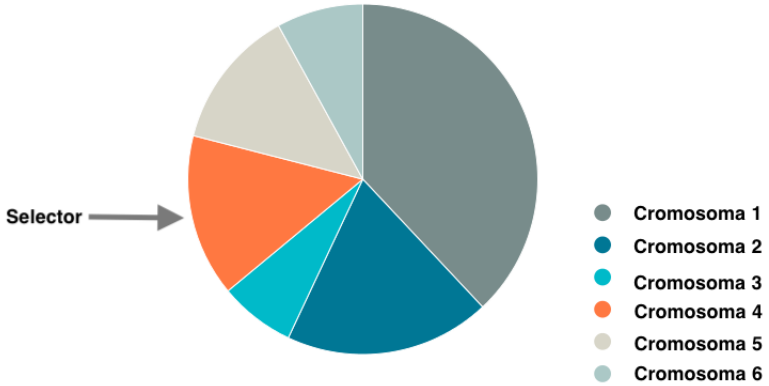


Figura 3-3: Selección por ruleta. En el ejemplo que se dio la fitness se calcula evalúando el número con la función f.

representación decimal es 19, 7 y 24. Al evalúar tenemos que:

$$f(19) = -2563,$$

$$f(7) = 413,$$

$$f(24) = -28708,$$

de modo que es el segundo cromosoma quien tiene el mejor fitness. Notemos que en este ejemplo, y en general, es posible que la función de fitness de un resultado negativo. Para lidiar con esto se puede sumar el número más negativo que se obtenga de las fitness de la generación a todas las fitness. Con esto tenemos podemos tratar sólo con números positivos.

3.3. Reproducción

El *operador de reproducción* toma a los dos padres y los mezcla para generar *descendencia*. Este operador debe determinar qué tan probable es que se crucen los cromosomas padres, cuánta descendencia genera una pareja de padres y cómo se recombinan los cromosomas padres. En esta tesis se define una probabilidad de cruce, p_c, y en cada reproducción se genera un punto de corte aleatoriamente. Este punto de corte determina en qué gen se van a cortar los cromosomas

padres para así recombinar (ver figura 3-3).

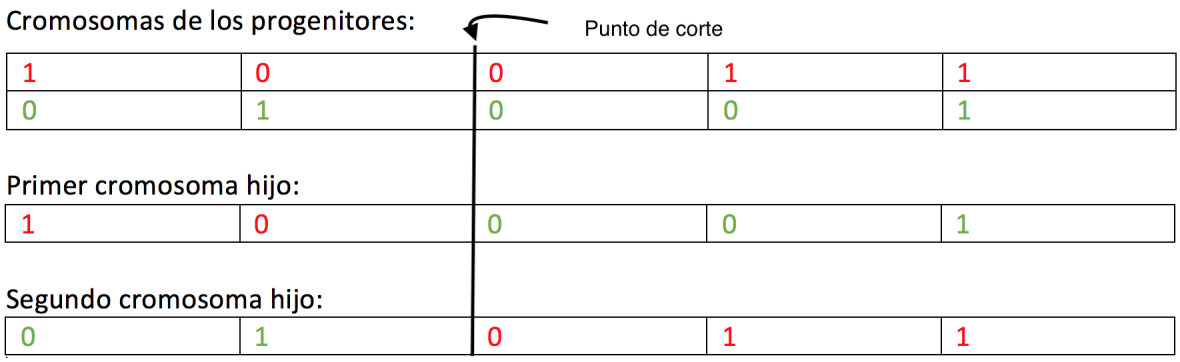


Figura 3-4: Recombinación de padres para generar descendencia. En este ejemplo se mezclan los primeros 2 genes del primer progenitor con los últimos 4 del segundo progenitor para generar al primer descendiente y los últimos 4 genes del primer progenitor con los primeros 2 del segundo progenitor para el segundo descendiente.

3.4. Mutación

El *operador de mutación* toma a la descendencia y cambia los genes. Se define una probabilidad de mutación, p_m, que determina si el operador de mutación actúa. De actuar, el operador escoge al azar entre los genes del cromosoma y los cambia al azar por otra instancia del alfabeto.

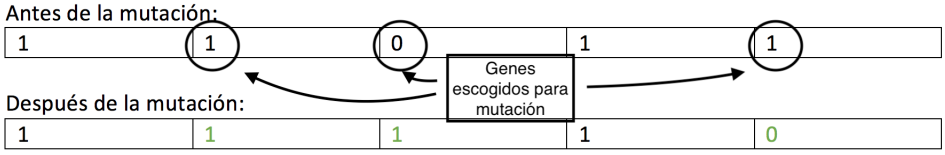


Figura 3-5: En este ejemplo el operador de mutación escogió al azar al segundo, tercer y quinto gen. Notemos que el segundo gen quedó igual después de la mutación mientras los otros cambiaron.

En esta tesis el operador de mutación tiene tres suboperadores, es decir, si un gen es escogido para mutar puede que mute de una de las tres siguientes maneras:

- Mutación aleatoria. El gen se convierte en otro número aleatorio.
- Mutación a cero. El gen se convierte en cero.

- Mutación de copia. El gen se convierte en otro de los genes del cromosoma escogido aleatoriamente.

Capítulo 4

Formulación del problema de contruir testigos

Para comenzar a programar un GA que construya testigos de enredamiento se necesita definir varias funciones para tareas específicas. En primer lugar se necesita una función que produzca una población inicial de N posibles testigos, los cuales, al ser operadores que actúan sobre el espacio de Hilbert de 2 qubits, se representan con matrices Hermitianas de 4×4. Esto se puede ver del hecho de que los qubit viven en un espacio de Hilbert de dimensión 2 ($\mathcal{H}^{(2)}$), donde un vector $|\phi\rangle$ se representa como

$$|\phi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \alpha |0\rangle + \beta |1\rangle ,$$

donde $|\alpha|^2 + |\beta|^2 = 1$ y $\{|0\rangle , |1\rangle\}$ es la así llamada *base computacional* de $\mathcal{H}^{(2)}$. Así, un vector $|\psi\rangle$ en el espacio de 2 qubits será

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle = \begin{bmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{bmatrix} ,$$

y por tanto los operadores que actúan sobre este espacio se representan con matrices Hermitianas de 4×4.

Tomemos en cuenta que una matriz Hermitiana de 4×4 sólo tiene 10 entradas independientes. Las demás entradas quedan determinadas por la condición $H = H^\dagger$. Podemos tomar como cromosoma estos 10 números.

También se debe definir la función de fitness, de lo cual depende el posible éxito del programa. La primera aproximación será tomar la inspiración directamente de la definición de un testigo y de la ecuación (2-2). Se empieza generando una familia de estados separables y se evalúa el resultado de $\mathrm{Tr}(W\rho_s)$ para todos ellos en cada uno de los integrantes de nuestra población inicial de observables. Como queremos que nuestras matrices Hermitianas sean testigos, debemos exigir que el resultado sea positivo para todos estos estados, entonces se penalizan resultados negativos de $\mathrm{Tr}(W\rho_s)$. Por otro lado, queremos obtener un resultado negativo para al menos un estado enredado, por lo que debemos proveer al algoritmo con algunos estados enredados, ρ_e, y recompensar el que detecte alguno de ellos, es decir recompensar el que $\mathrm{Tr}(W\rho_e) < 0$. Para escribir correctamente las funciones que generan la población inicial y a los estados separables y enredados aleatorios es necesario repasar un poco de matrices aleatorias. Los siguientes capítulos se basan en (Ozols [2009]).

4.1. Matrices aleatorias

Primero introducimos algunas definiciones y un poco de notación. Denotamos por id a la matriz identidad, por M^T a la matriz transpuesta a M, por M^* a la matriz cuyos elementos son los complejos conjugados de M y por $M^\dagger$ a la transpuesta conjugada a M. También definimos a

$O(n) := \{O \in \mathbb{R}^{n\times n} | O^T O = \mathrm{id}\}$ el grupo ortogonal.
$SO(n) := \{O \in O(n) | \det(O) = 1\}$ el grupo ortogonal especial.
$U(n) := \{U \in \mathbb{R} | U^\dagger U = \mathrm{id}\}$ el grupo de matrices unitarias.
$\mathbb{S}^{n-1} := \{\mathbf{x} \in \mathbb{R}^n | \sum_{i=1}^n x_i^2 = 1\}$ la esfera unitaria en $\mathbb{R}^n$.

4.1.1. Distribuciones uniformes

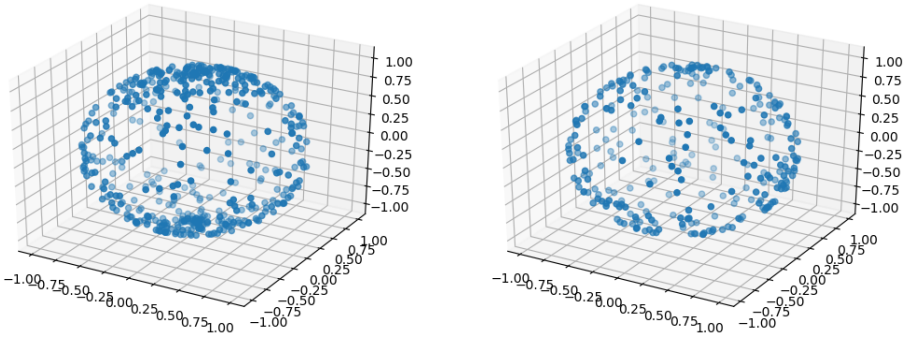
Un problema al generar conjuntos de matrices aleatorias es el de representar uniformemente a todas las posibles matrices. Para ejemplificar esto considérese el problema de tomar puntos de la esfera $\mathbb{S}^2$. Uno puede parametrizar los puntos $(x, y, z) \in \mathbb{S}^2$ como

$$\begin{aligned} x &= \text{sen}\theta\cos\phi, \\ y &= \text{sen}\theta\cos\phi, \\ z &= \cos\theta, \end{aligned}$$

donde $\theta \in [0, \pi)$ y $\phi \in [0, 2\pi)$. Uno no obtiene una distribución uniforme sobre $\mathbb{S}^2$ si se escoge a θ y a ϕ uniformemente, sino que los puntos así generados van a concentrarse cerca de los polos (ver figura 4-1(a)). Para ver esto notemos que el elemento de área de $\mathbb{S}^2$ está dado por

$$dS = \text{sen}\theta d\theta d\phi.$$

Vemos que el elemento de área depende de θ. Entonces la probabilidad de encontrar un punto



(a) Distribución no uniforme. (b) Distribución uniforme.

Figura 4-1: El problema de la distribución uniforme. (a) Los puntos se concentran en los polos. (b) Los puntos de distribuyen uniformemente en la superficie de la esfera. En ambas figuras se usa el mismo número de puntos, 5000.

en alguna sección de la esfera no va ser uniforme, sino que dependerá de θ.

La densidad de probabilidad que se espera tener para una distribución uniforme de puntos debe estar dada por $f(\theta, \phi) = \frac{1}{4\pi}$, de modo que la probabilidad de encontrar un punto en una vecindad infinitesimal de cualquier lugar de la superficie de la esfera sea: $P(\theta, \phi) = dSf(\theta, \phi) =$

constante. Por tanto, para obtener una distribución uniforme sobre $\mathbb{S}^2$, uno tiene que escoger $\phi \in [0, 2\pi)$ y $t \in [-1, 1)$ uniforme y aleatoriamente y calcular θ como

$$\theta = \arccos(t).$$

Esto sirve ya que el elemento de área para este cambio de variable será

$$dS = -dtd\phi,$$

el cual no depende de ninguna de las variables.

Para terminar la discusión se debe introducir, al menos de manera intuitiva, el concepto de medida.

Definición 10 *(Medida de probabilidad)*[1]

Una *medida* μ sobre un grupo G es una función que le asigna un valor a subconjuntos del grupo, algo que se puede interpretar como el *tamaño* del mismo. Entonces, si $M \subset G$, el tamaño del subconjunto será: $\mu(M) = \int_M d\mu$. Una *medida de probabilidad* μ es una medida tal que $\mu(G) = 1$.

Se puede pensar en el problema de la distribución de puntos en la esfera de otro modo. La distribución de la figura 4-1(a) no es invariante bajo rotaciones. Si hacemos una rotación de $\pi/2$ a lo largo del eje x la concentración de puntos se dará en el ecuador. Por otro lado, la distribución de la figura 4-1(b) es invariante bajo estas rotaciones. Esta imagen geométrica da una pista sobre qué hay que exigir a una distribución de probabilidad para que represente uniformemente el espacio donde se está trabajando. El grupo de simetrías sobre $\mathbb{S}^2$ es $SO(3)$ ya que este grupo representa las rotaciones en $\mathbb{R}^3$ y la esfera es invariante bajo rotaciones. Vemos entonces que una medida de probabilidad que sea invariante bajo este grupo, es decir, una medida de probabilidad tal que $\mu(S) = \mu(OS)$ donde $S \subset \mathbb{S}^2$ y $O \in SO(3)$, será uniforme sobre $\mathbb{S}^2$.

[1]Esta no es la definición matemática formal pero para los propósitos de la tesis la definición intuitiva es suficiente.

4.1.2. Medida de Haar

Sea f una función definida en $\mathbb{R}$ e integrable. Entonces para cualquier $a \in \mathbb{R}$ se tiene que

$$\int_{\mathbb{R}} f(x)dx = \int_{\mathbb{R}} f(a+x)dx. \tag{4-1}$$

Notemos que $(\mathbb{R}, +)$ define a un grupo. Se puede tener una invariancia similar a (4-1) para otros grupos. Si se hace la elección correcta de medida, μ, sobre un grupo G entonces, para todo $g \in G$ tendremos que:

$$\int_G f(x)d\mu(x) = \int_G f(gx)d\mu(x). \tag{4-2}$$

Definición 11 *(Medida de Haar)*

A una medida no cero $\mu : G \to [0, \infty]$ tal que para todo $S \subseteq G$ y $g \in G$ ocurra que

$$\mu(gS) = \mu(Sg) = \mu(S), \tag{4-3}$$

donde

$$\mu(S) := \int_{g \in S} d\mu(g), \tag{4-4}$$

se le llama *medida de Haar*.

La medida de Haar existe para cualquier grupo[2] y es única hasta una constante multiplicativa. Si además se tiene que $\mu(G) = 1$ entonces a μ se le conoce como *medida de probabilidad* sobre G. En efecto, si f es una densidad de probabilidad sobre G y $d\mu(g) := f(g)dg$, entonces

$$\mu(S) := \int_{g \in S} d\mu(g) = \int_{g \in S} f(g)dg \tag{4-5}$$

es la probabilidad total de S.

Volviendo al ejemplo de la esfera, la medida de interés es el área de un pedazo de la superficie

[2]Estrictamente la medida de Haar se define sobre grupos topológicosde Hausdorff localmente compactos. La definición matemática precisa está más allá del alcance de esta tesis y basta con que exista para el grupo $U(2)$.

de la esfera, es decir, si $C \subset \mathbb{S}^2 \subset \mathbb{R}^3$ es un pedazo de la superficie de la esfera tendremos que

$$\mu(C) := \int_{x \in C} d\mu(x) = \int_{x \in C} \frac{1}{4\pi} dS = \frac{\text{área}(C)}{\text{área(Esfera)}}.$$

Justamente esa es la probabilidad de encontrar en el pedazo C un punto que fue generado aleatoriamente con una distribución uniforme sobre la superficie de la esfera. Es claro que esta elección de medida es invariante bajo rotaciones ya que la esfera lo es; el área de un pedazo de esfera no va a cambiar por ser rotado. Por tanto dS define la medida de Haar sobre la esfera.

4.1.3. Distribución uniforme sobre $U(2)$

El grupo unitario $U(2)$ se puede parametrizar de la siguiente manera

$$U(\alpha, \phi, \psi, \chi) := e^{i\alpha} \begin{pmatrix} e^{i\psi}\cos\phi & e^{i\chi}\text{sen}\phi \\ -e^{-i\chi}\text{sen}\phi & e^{-i\psi}\cos\phi \end{pmatrix}, \tag{4-6}$$

donde $\phi \in [0, \frac{\pi}{2})$ y $\alpha, \psi, \chi \in [0, 2\pi)$. El elemento de volumen está dado por (Fyodorov [2004])

$$d\mu(U) = 2\text{sen}\phi\cos\phi d\phi d\alpha d\psi d\chi = \frac{1}{2} d(\text{sen}^2\phi) d\alpha d\psi d\chi. \tag{4-7}$$

Con esto queda claro que para obtener una distribución uniforme sobre $U(2)$, uno debe escoger $\alpha, \psi, \chi \in [0, 2\pi)$ y $\xi \in [0, 1)$ uniforme y aleatoriamente y calcular

$$\phi = \text{arcsen}\sqrt{\xi}. \tag{4-8}$$

Veamos: de (4-7) se llega a $\xi = \text{sen}^2\phi$, de modo que $2\text{sen}\phi\cos\phi d\phi = d\xi$ y por tanto (4-7) se puede escribir como

$$d\mu(U) = d\xi d\alpha d\psi d\chi,$$

lo cual claramente no depende de ninguna de las variables.

La expresión (4-7) es invariante con respecto a la multiplicación $U \to VU$ para cualquier matriz unitaria V del mismo grupo (Fyodorov [2004]). Por lo tanto (4-7) define la medida de Haar en el grupo.

4.2. Estados y operadores aleatorios

Ahora volvemos al tema central de la tesis, el algoritmo genético. Como ya se mencionó, el espacio de búsqueda consiste de matrices Hermitianas de 4×4. La población inicial se va a generar aleatoriamente. Como ya se explicó los cromosomas consisten de 10 números aleatorios, 4 reales y 6 complejos. Para generar la población inicial se inicia un arreglo de 10 elementos vacíos. El arreglo se llena con 4 números reales aleatorios escogidos de acuerdo a una distribución normal y con 6 complejos cuyas partes imaginaria y real se escogen con la misma distribución. Estos diez números se acomodan en una matriz hermitiana, los primeros cuatro son los elementos de la diagonal y los siguientes 6 son la parte triangular superior. El código que se usa se encuentra en el anexo. Las matrices así generadas son parte del *ensamble Gaussiano unitario* inducido por la medida de Haar sobre $U(2)$ (Fyodorov [2004]).

Para evaluar el fitness de los posibles testigos es necesario probarlos con un gran número de estados separables. Estos han de ser generados aleatoriamente. Para hacerlo (Zyczkowski *et al.* [2010]) consideremos un estado producto arbitrario en un sistema bipartito, $|0,0\rangle$. Una operación local unitaria $U = U_A \otimes U_B$ no puede producir enredamiento y por tanto el estado definido por dos matrices unitarias aleatorias

$$|\psi_{AB}\rangle = U\,|0,0\rangle = U_A\,|0\rangle_A \otimes U_B\,|0\rangle_B = |\psi\rangle_A \otimes |\psi\rangle_B\,, \tag{4-9}$$

también será separable. Notemos que esto implica que un estado puro separable aleatorio se puede construir haciendo el producto de dos estados aleatorios (lado derecho de (4-9)). Para generar un estado mixto separable aleatorio podemos construir varios estados puros separables aleatorios y mezclarlos para que tengan la forma (1-4).

En esta tesis se estudian sistemas de 2 qubits, por lo que los estados aleatorios que debemos generar se representan mediante vectores complejos normalizados de dos entradas que corresponden a $|\psi\rangle_A$ y $|\psi\rangle_B$ de (4-9). El código que se usó se encuentra en el anexo.

4.3. Función de fitness

La función que evalúe el fitness debe calcular el resultado de $\mathrm{Tr}(W\rho)$ para un conjunto grande de estados separables y para algunos estados enredados. Es claro que para los estados separables el resultado debe ser positivo, así que se penalizan resultados negativos. Para estados enredados se dan puntos cuando el resultado sea negativo. Unos estados enredados importantes son los *estados de Bell* definidos como

$$|\psi_{\pm}\rangle := \frac{1}{\sqrt{2}}\big(\,|0^A\rangle \otimes |1^B\rangle \pm |1^A\rangle \otimes |0^B\rangle\,\big),$$

$$|\phi_{\pm}\rangle := \frac{1}{\sqrt{2}}\big(\,|0^A\rangle \otimes |0^B\rangle \pm |1^A\rangle \otimes |1^B\rangle\,\big).$$

Ahora pongamos un ejemplo: Llamémosle $\mathcal{S}$ al conjunto de n estados separables que se generaron aleatoriamente, $\mathcal{E}$ al conjunto de l estados enredados prueba y $\mathcal{W}$ a la primera generación de m observables del algoritmo.

$$\mathcal{S} := \{\rho_s^{(i)} \in \mathfrak{D}(\mathcal{H}) | \rho_s^{(i)} \text{ es separable e } i = 1, \ldots, n\}.$$

$$\mathcal{E} := \{\rho_e^{(i)} \in \mathfrak{D}(\mathcal{H}) | \rho_e^{(i)} \text{ está enredado e } i = 1, \ldots, l\}.$$

$$\mathcal{W} := \{W_i | W_i \text{ es Hermitiano e } i = 1, \ldots, m\}.$$

Entonces el fitness del operador j se evalúa de la siguiente forma:

1. Se inicializa el contador `fitness=0.`
2. `for` $\{i \in [1:n]\}$ `do`
   - `if` $\{Tr(W_j\rho_s^{(i)}) < 0\}$ `do`
     - `fitness = fitness - penalización`
3. `for` $\{i \in [1:l]\}$
   - `if` $\{Tr(W_j\rho_e^{(i)}) < 0\}$ `do`
     - `do fitness = fitness + abs(`$\mathrm{Tr}(W_j\rho_e^{(i)})$`).`

El código que se usó se encuentra en el anexo. Para escribir matemáticamente la función de fitness definimos las siguientes dos funciones auxiliares

$$P_i(W) := \begin{cases} 0 & \text{si } \mathrm{Tr}(W\rho_s^{(i)}) \geq 0 \\ -P & \text{si } \mathrm{Tr}(W\rho_s^{(i)}) < 0 \end{cases},$$

$$R_i(W) := \begin{cases} 0 & \text{si } \mathrm{Tr}(W\rho_e^{(i)}) \geq 0 \\ R \cdot \mathrm{Tr}(W\rho_e^{(i)}) & \text{de otro modo} \end{cases},$$

donde P y R son constantes mayores que cero que representan la penalización y la recompenza. De modo que la función de fitness se escribe como

$$F(W) = \sum_{i=1}^{n} P_i(W) + \sum_{j=1}^{l} R_j(W). \tag{4-10}$$

Se sabe que las operaciones locales unitarias $U = U_A \otimes U_B$ preservan el enredamiento, así que un estado transformado localmente

$$|\phi_{\mathrm{enr}}\rangle = (U_A \otimes U_B) \left|\Psi^+\right\rangle, \tag{4-11}$$

donde $|\Psi^+\rangle$ es uno de los estados de Bell, se mantiene enredado. Siguiendo a Zyczkowski *et al.* [2010] podemos generar matrices unitarias aleatorias, $U_A, U_B \in U(2)$ de acuerdo a la medida de Haar (4-7) y con ello obtener un ensamble de estados enredados aleatorios.

Esto nos permite definir una segunda función de fitness usando un conjunto aleatorio de estos estados enredados con los cuales se va a evalúar $\mathrm{Tr}(W\rho_{\mathrm{enr}})$, con $\rho_{\mathrm{enr}} = |\phi_{\mathrm{enr}}\rangle\langle\phi_{\mathrm{enr}}|$, para dar puntos al fitness. En esta tesis se generan 14 dichos estados, es decir, el conjunto $\mathcal{E}$ tiene 14 elementos y son los estados aleatorios generados con (4-11).

El código para generar las matrices unitarias aleatorias se encuentra en el apéndice y lo que hace es:

1. Se inicializa un arreglo donde se van a guardar las matrices. unitarias=Arreglo(Matrices($2n$))
2. `for` $\{l \in [1:2n]\}$ `do`
   - ξ=`rand()` ; α=2π*`rand()`; ψ =2π*`rand()` ; χ==2π*`rand()`;

- $\phi = \texttt{arcsen}(\sqrt{\xi})$
- $\texttt{unitarias[}l\texttt{]} = e^{i\alpha} \begin{pmatrix} e^{i\psi}\cos\phi & e^{i\chi}\text{sen}\phi \\ e^{-i\chi}\text{sen}\phi & e^{-i\psi}\cos\phi \end{pmatrix}$,

Estamos evalúando el fitness con base en estados máximamente enredados. Esto podría hacer que los testigos que resulten no sean buenos detectando estados débilmente enredados. Para intentar corregir esto consideramos los estados de Werner,

$$\rho_{\text{werner}} := p|\Phi_+\rangle\langle\Phi_+| + (1-p)\frac{1}{4}\text{id}. \tag{4-12}$$

Los eigenvalores de la matriz parcialmente transpuesta a ρ_{werner} son

$$\lambda_{1,2,3} = \frac{1}{4}(1+p),\ \lambda_4 = \frac{1}{4}(1-3p).$$

Por tanto, de acuerdo al criterio PPT, ρ_{werner} está enredado si y sólo si $p > \frac{1}{3}$. Adicionalmente el enredamiento de formación de los estados de Werner como función de p es monotónicamente creciente (ver figura 4-2). Esto nos indica que podemos generar una familia de estados débilmente

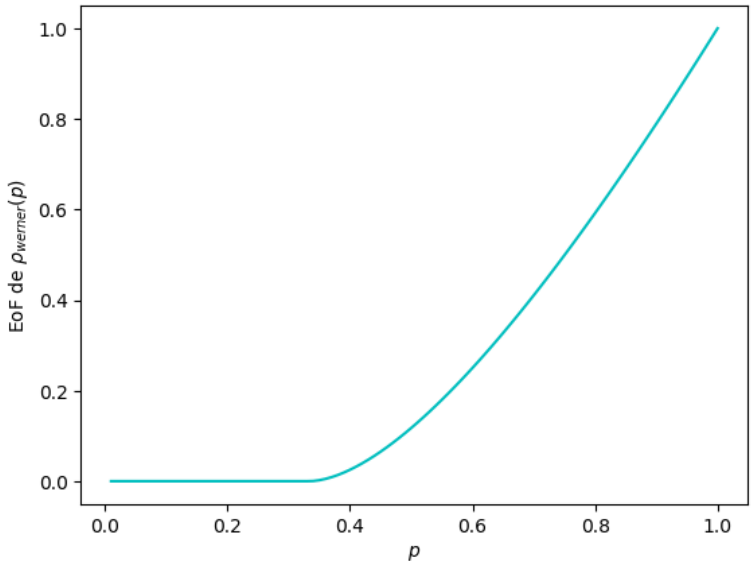


Figura 4-2: EoF como función de p para estados de Werner.

enredados usando el mismo procedimiento que en (4-11). Para ello se hace la transformación

$$\rho_{w_{enr}} = (U_A \otimes U_B)\rho_{werner}(U_A \otimes U_B)^\dagger. \tag{4-13}$$

Por la propiedad (II) de las medidas de enredamiento se genera así una familia de estados aleatorios con la misma cantidad de enredamiento. Con esto se define una tercera función de fitness, buscando así construir testigos de estados débilmente enredados. Para esta tesis se generaron 14 estados usando el estado de Werner con p=0.45.

A la función de fitness que usa a los estados de Bell para dar puntos de fitness se le llamó `GA1`, a la que usa estados enredados aleatorios generados mediante (4-11) se le llamó `GA_varios_enredados` y a la función que usa estados aleatorios enredados generados mediante (4-13) se le llamó `GA_werner`. Notemos que la definición (4-10) es general y sólo cambia el conjunto $\mathcal{E}$ de estados enredados sobre los que se suma. En los 3 casos se genera al conjunto $\mathcal{S}$ aleatoriamente.

4.4. El algoritmo genético

Quedan por definir los operadores de cruzamiento y mutación. Para el cruzamiento vamos a usar la selección por ruleta. La probabilidad de que un individuo sea escogido es

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}, \tag{4-14}$$

donde f_i es la fitness del individuo i. Puede ocurrir que $f_i < 0$ para algún i. Supongamos sin pérdida de generalidad que el individuo l tiene el fitness más negativo. Entonces podemos hacer la transformación $f_i \to f_i + |f_l|$ para todos los individuos. Con esto aseguramos que todos los fitness sean positivos y las proporciones se quedan igual. El código para implementar la ecuación (4-14) se encuentra en el anexo. La función se llama asignar probabilidad y funciona de la siguiente manera:

`asignar_probabilidad(p,X)`. X es el arreglo cuyo elemento i es el fitness del individuo i de una generación dada.

1. El primer argumento un número aleatorio entre 1 y la suma de fitnesses: `p = rand(1:sum(X))`

; l=length(X) ; contador=1 ; lb=0 (lower bound)

2. for $\{i \in [1:l]\}$ do

   - if $\{$lb $\leq$ p $<$ lb + X[contador]$\}$ do

     - break.

   - lb = lb + X[contador] ; contador = contador +1

3. return contador.

La forma en que se escoge a los cromosomas que han de cruzarse se implementa de la siguiente manera:
Operador de cruzamiento:

1. reparto = sum(valores de fitness de la generación actual)

2. Se inicia un contador de pobladores de la nueva generación: poblacion_nueva=1

3. while $\{$población_nueva $<$ tamaño_población+1$\}$ do

   - p=rand(1:reparto) ; q=rand(1:reparto)

   - padre1 = asignar_probabilidad(p,valores_de_fitness) ;
     padre2 = asignar_probabilidad(q,valores_de_fitness)

   - if $\{$rand()$<$ probabilidad de cruce$\}$ do

     - Se escoge el punto de corte, como los cromosomas son de largo 10 el punto de corte debe estar entre 1 y 9: corte=rand(1:9)

     - hijo1 = padre1[1:corte],padre2[corte+1:10] ;
       hijo2=padre2[1:corte],padre1[corte+1:10]

El operador de mutación se implementa como sigue:

1. mutación=rand()

2. if $\{$mutación $<$ probabilidad de mutación$\}$ do

   - for $\{i \in [1:10]\}$ do

- `ra = rand()`
- Dependiendo del valor de ra se hace una de las tres mutaciones siguientes:
  - `hijo[rand(1:10)] = randn()` con probabilidad 0.2

    ó
  - `hijo[rand(1:10)] = hijo[rand(1:10)]` con probabilida 0.35

    ó
  - `hijo[rand(1:10)] = 0` con probabilidad 0.1.

    ó
  - No se hace nada al gen con probabilidad 0.35.

Así se implementan los tres tipos de mutación: una que cambia el gen a cualquier otro número, uno que cambia el gen a cero y otra que hace que se duplique un gen. Esto se hace pensando en la forma del testigo óptimo: el testigo óptimo es una matriz con muchos elementos cero y sus entradas distintas de cero son todas iguales. El código de los operadores de cruzamiento y mutación se encuentran en el anexo.

Capítulo 5

Análisis de resultados

Para los cálculos se usa un testigo equivalente al óptimo:

$$W_o := 10 \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}. \tag{5-1}$$

En este capítulo se analizan los testigos que generan los GA's. Recordemos que el objetivo de la tesis es demostrar que se pueden contruir testigos usando GA's, por esta razón se van a comparar con el testigo óptimo definido en (5-1). Para el análisis vamos a estudiar las matrices que resultan así como algunas gráficas. Un tipo de gráfica que se usa es la de EoF contra $\mathrm{Tr}(W\rho)$. Para estas gráficas se crean 500,000 estados aleatoriamente (e_a estados aleatorios en el código), se clasifican de acuerdo a su EoF y se calcula $\mathrm{Tr}(W\rho)$ para cada uno de ellos con distintos testigos, W, y se dibuja el punto donde se dan estos valores. Lo que se obtiene para W_o es una gráfica como la de la figura 5-1. Se traza la línea que separa los estados detectados de los no detectados y los separables. Otra forma de análisis se hace mediante tablas de desempeño de los testigos. En estas tablas se clasifica cuántos estados enredados detecta cada testigo y además cuántos estados detecta en ciertos intervalos de EoF.

Para determinar si los operadores Hermitianos que resultan del GA son testigos validos se clasificaron los 500,000 estados en separables y enredados por medio del criterio PPT. Al

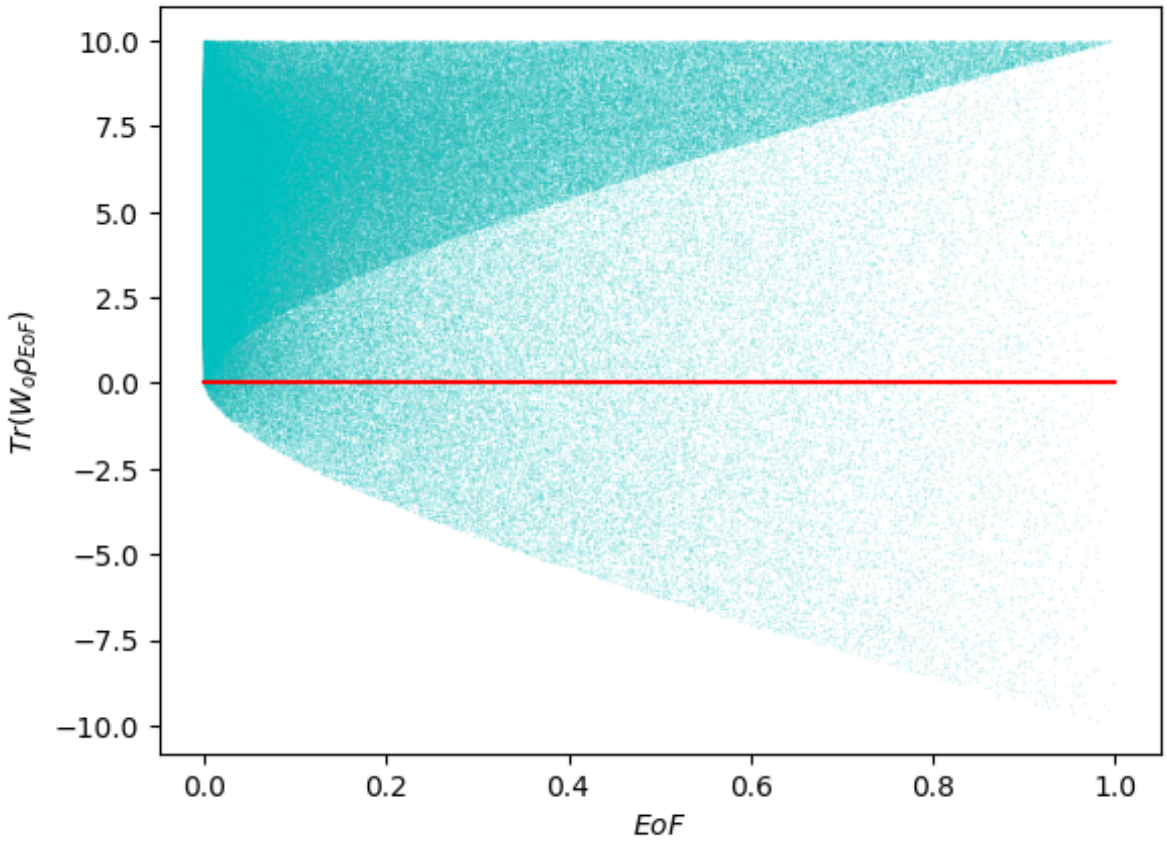


Figura 5-1: Estados clasificados por su EoF y el valor de $Tr(W_o\rho)$. La línea roja separa a los estados que detecta el testigo de los separables y los no detectados.

número de elementos en el conjunto de estados separables se le llamó e_{as} y al de enredados e_{ae}. Una vez separados se evalúa $\mathrm{Tr}(W\rho_s)$ para cada posible testigo y se clasifica a los estados en dos conjuntos, mal clasificados y bien clasificados. Cuando el conjunto mal clasificados no tiene ningún elemento concluimos que W es un testigo válido.

Como prueba para este método se realizó el siguiente experimento: Se generaron 5,000,000 estados aleatorios, se clasificaron y se evalúo $\mathrm{Tr}(W_o\rho_s)$ para el testigo óptimo y los estados separables y se observó que W_o nunca clasificó mal a los estados que se determinó eran separables de acuerdo al criterio PPT. De aquí se concluye que esta prueba es suficientemente fuerte para determinar si los operadores que resultan del GA son testigos válidos.

5.1. Función de fitness `GA1`

Para este algoritmo genético se usó la primera función de fitness en la que los puntos de fitness positivos se dan por detectar a los estados de Bell. El número de estados separables, n_s en el código, que se usan para evalúar el fitness fue de 4500. El algoritmo se dejó correr por 350 generaciones, el tamaño de población fue de 52 individuos, la probabilidad de cruce fue de 0.84 y la de mutación fue de 0.05.

n_s	4500
Generaciones	350
Tamaño de población	52
Probabilidad de cruce	0.84
Probabilidad de mutación	0.05

Tabla 5-1: Tabla de parámetros utilizados en `GA1`.

En la figura 5-2 vemos las gráficas hechas con algunos de los operadores que resultaron. Se encuentra que hay tres tipos de patrones recurrentes en las gráficas: uno de la forma del óptimo, una con una fuerte separación y otra sin estructura.
Las matrices de estos testigos son:

$$W_{1_{GA1}} = \begin{pmatrix} 11.6138 & 0 & 0 & 0 \\ 0 & 0 & 7.08247 - 0.651334i & 0 \\ 0 & 7.08247 + 0.651334i & 0 & 0 \\ 0 & 0 & 0 & 6.23824 \end{pmatrix} \tag{5-2}$$

$$W_{2_{GA1}} = \begin{pmatrix} 7.46957 & 0.806753 - 0.419403i & -0.513706 + 1.68753i & -8.92646 - 0.303835i \\ 0.806753 + 0.419403i & 4.74564 & 8.46547 - 0.872359i & 4.4718 + 0.355348i \\ -0.513706 - 1.68753i & 8.46547 + 0.872359i & 3.08535 & -0.937909 + 1.68053i \\ -8.92646 + 0.303835i & 4.4718 - 0.355348i & -0.937909 - 1.68053i & 5.06999 \end{pmatrix} \tag{5-3}$$

$$W_{3_{GA1}} = \begin{pmatrix} 5.58917 & -2.50504 - 4.08635i & 0 & 5.4297 + 0.800141i \\ -2.50504 + 4.08635i & 6.66923 & 0 & 0 \\ 0 & 0 & 2.51123 & 0 \\ 5.4297 - 0.800141i & 0 & 0 & 2.51123 \end{pmatrix} \quad (5\text{-}4)$$

También se hace una tabla donde se pone el fitness, el número de estados bien clasificados, número de estados mal clasificados, número de estados detectados y número de estados no detectados; la tabla 5-2.

	Fitness	Detectados	No detectados	Bien clasificados	Mal clasificados
W_o	30	37056	375199	87745	0
$W_{1_{GA1}}$	22	29235	383020	87745	0
$W_{2_{GA1}}$	22	37550	374705	87667	78
$W_{3_{GA1}}$	5	8516	403739	87745	0

Tabla 5-2: Tabla de desempeño. Fitness calculada con `GA1`.

Finalmente se estudia el desempeño de cada testigo por intervalos de EoF, tabla 5-3.

	Detectados				Enredados
Intervalo	W_o	$W_{1_{GA1}}$	$W_{2_{GA1}}$	$W_{3_{GA1}}$	
0-0.1	5545	3078	1055	676	205800
0.1-0.2	6239	4558	2997	582	62052
0.2-0.3	5775	4618	4735	866	41800
0.3-0.4	5005	4218	5453	1001	30968
0.4-0.5	4294	3760	5853	1113	23564
0.5-0.6	3512	3141	5431	1190	17932
0.6-0.7	2930	2452	4953	1111	13420
0.7-0.8	2055	1886	3774	1033	9443
0.8-0.9	1235	1119	2405	700	5446
0.9-1	466	405	894	244	1830

Tabla 5-3: Detección por intervalos.

5.1.1. Análisis

El algoritmo tiende a encontrar tres tipos de operadores. En la matriz (5-2) notamos que uno de ellos tiene una forma muy similar a la del testigo óptimo, (5-1). Esto nos lleva a concluir que este tipo de matrices constituyen un óptimo local del espacio de búsqueda. Para darse una idea de la frecuencia con la que aparecen este tipo de matrices se corrió el GA 20 veces, en cada ocasión se escogió a los 10 individuos con mejor fitness y se revisó cuántos de ellos eran de la forma del óptimo. Se encuentra que 29 de ellos fueron de esa forma, es decir, en promedio siempre hay al menos uno así para todas las corridas del GA.

La matriz (5-3) es muy distinta y no tiene ninguna simetría más allá de ser Hermitiana. Notemos de la tabla 5-2 que $W_{2_{GA1}}$ no es un testigo de enredamiento ya que clasifica mal algunos estados separables, aunque el error en la clasificación es del orden del 0.0008 %. Esto quiere decir que si toleramos un pequeño error tendremos observables muy buenas para identificar estados enredados. Como se puede ver en la tabla 5-3, $W_{2_{GA1}}$ detecta más estados que W_o, sobre todo si los estados tienen un enredamiento alto.

La última clase que genera es la de la matriz (5-4). Son testigos válidos, como podemos ver en la tabla, pero detectan muy pocos estados. Estas matrices puede que surjan debido a que la función de fitness usada pone mucho énfasis en que no clasifique mal estados separables y poco en la detección de estados enredados.

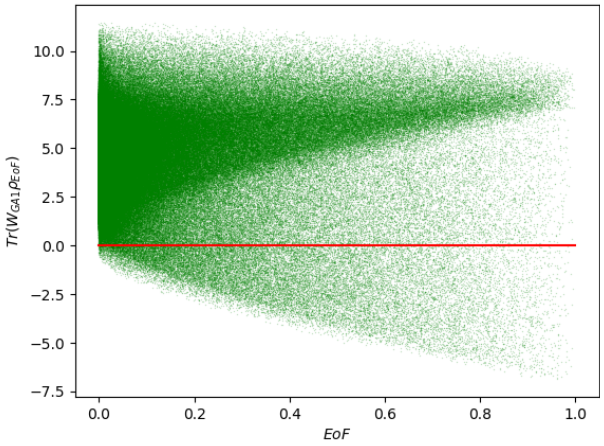
En la tabla 5-3 observamos que el testigo óptimo se desempeña mejor en todos los rangos de enredamiento que $W_{1_{GA1}}$ y $W_{3_{GA1}}$. Esto justifica la necesidad de otras funciones de fitness para generar otro tipo de testigos.

También se debe notar que W_o es mejor detectando estados fuertemente enredados. Para ver esto se presenta la tabla 5-4.

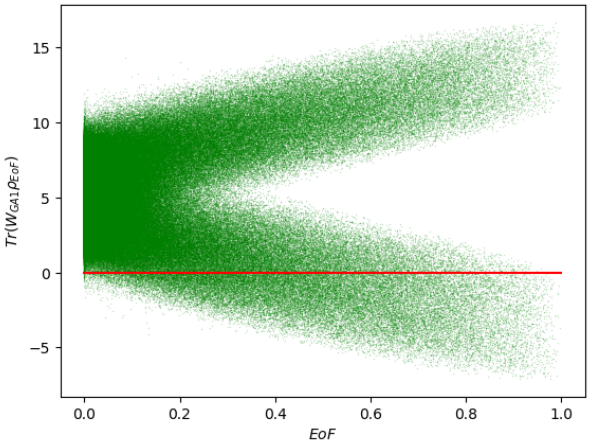
Podemos ver que, aunque los estados débilmente enredados son los más numerosos, el testigo óptimo detecta muy pocos de ellos.

	% de Detectados
Intervalo	W_o
0-0.1	2.7
0.1-0.2	10
0.2-0.3	13.8
0.3-0.4	16.1
0.4-0.5	18.2
0.5-0.6	19.6
0.6-0.7	21.8
0.7-0.8	21.7
0.8-0.9	22.7
0.9-1	25.4

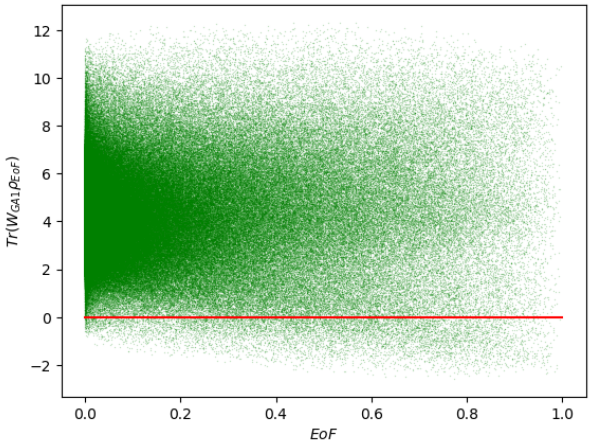
Tabla 5-4: Porcentaje de estados detectados por intervalos.



(a) Similar al óptimo.



(b) Separación fuerte.



(c) Sin estructura.

Figura 5-2: Algunos de los operadores obtenidos con `GA1`. (a) Éste sí es un testigo y tiende a una forma similar a la de W_o. (b) El operador no cumple la definición de testigo porque clasifica mal algunos estados separables. Nótese el espacio vacío entre los detectados y los no detectados. (c) Un testigo sin estructura clara.

5.2. Función de fitness `GA_varios_enredados`

Para este algoritmo se usa una función de fitness en la que los puntos positivos se dan al evalúar $\mathrm{Tr}(W\rho_{enr})$ para varios estados enredados generados con el método discutido en la sección 4.3. Los parámetros que se ocupan se dan en la tabla 5-3. n_e es el número de estados enredados que se usan.

n_s	5000
n_e	14
Generaciones	350
Tamaño de población	52
Probabilidad de cruce	0.84
Probabilidad de mutación	0.052

Tabla 5-5: Tabla de parámetros utilizados en `GA_varios_enredados`.

De nuevo dibujamos algunas gráficas que resultan representativas de los operadores que generó el algoritmo, éstas se pueden ver en la figura 5-3.
Las matrices correspondientes son:

$$W_{1_{GA_{ve}}} = \begin{pmatrix} 0 & 0 & 0 & 4.147 + 1.85761i \\ 0 & 6.19013 & 0 & 0 \\ 0 & 0 & 3.39011 & 0 \\ 4.147 - 1.85761i & 0 & 0 & 0 \end{pmatrix} \tag{5-5}$$

$$W_{2_{GA_{ve}}} = \begin{pmatrix} 0 & 0 & 0 & 2.80499 + 4.54369i \\ 0 & 10.7748 & 0 & 0 \\ 0 & 0 & 10.7748 & 0 \\ 2.80499 - 4.54369i & 0 & 0 & 0 \end{pmatrix} \tag{5-6}$$

$$W_{3_{GA_{ve}}} \begin{pmatrix} 3.11612 & 0 & 0 & -8.9391 + 2.63243i \\ 0 & 8.0061 & 0 & 0 \\ 0 & 0 & 6.43794 & 0 \\ -8.9391 - 2.63243i & 0 & 0 & 5.10996 \end{pmatrix} \tag{5-7}$$

Se presenta también la tabla de desempeño, tabla 5-6. El fitness de W_o se calcula con la misma función de fitness `GA_varios_enredados`.

	Fitness	Detectados	No detectados	Bien clasificados	Mal clasificados
W_o	35	37069	374919	88012	0
$W_{1_{GA_{ve}}}$	7	35543	376445	88012	0
$W_{2_{GA_{ve}}}$	16	14634	397354	88012	0
$W_{3_{GA_{ve}}}$	18	13503	398485	88012	0

Tabla 5-6: Tabla de desempeño. La fitness es medida usando a los varios estados enredados.

La tabla de desempeño de cada testigo por intervalos de EoF es (tabla 5-7):

	Detectados				Enredados
Intervalo	W_o	$W_{1_{GA_{ve}}}$	$W_{2_{GA_{ve}}}$	$W_{3_{GA_{ve}}}$	
0-0.1	5562	7766	4457	267	206520
0.1-0.2	6202	5302	2312	1024	61879
0.2-0.3	5637	4941	1852	1559	41701
0.3-0.4	5088	4409	1519	1963	30682
0.4-0.5	4432	3780	1271	2143	23623
0.5-0.6	3564	3311	1062	1998	18130
0.6-0.7	2807	2525	817	1805	13149
0.7-0.8	2065	1894	678	1468	9114
0.8-0.9	1286	1192	474	957	5397
0.9-1	426	423	192	319	1793

Tabla 5-7: Detección por intervalos.

	Detectados			
Intervalo	W_o	$W_{1_{GA_{ve}}}$	$W_{2_{GA_{ve}}}$	$W_{3_{GA_{ve}}}$
0-0.01	339	1842	1196	0

Tabla 5-8: Desempeño para enredamiento muy débil.

5.2.1. Análisis

Podemos ver de la gráfica 5-3(a) y de (5-5) que $W_{1_{GA_{ve}}}$ es un testigo de la misma forma del óptimo. Esto refuerza la idea de que ésta corresponde a un óptimo local importante en el

espacio de búsqueda.

Adicionalmente, los testigos $W_{1_{GA_{ve}}}$ y $W_{2_{GA_{ve}}}$ tienen un buen desempeño con estados muy débilmente enredados como se puede ver de la tabla 5-8. La forma de la matriz (5-6) es igual a la de W_o pero en este la parte imaginaria de las entradas en los extremos es mayor que la parte real. Para enfatizar el punto se presenta la gráfica de la figura 5-4 donde vemos que $W_{2_{GA_{ve}}}$ detecta varios estados débilmente enredados que W_o no.

$W_{3_{GA_{ve}}}$ pareciera ser un testigo que no tuvo tiempo de convergir a una forma óptima. La matriz (5-7) tiene entradas distintas de cero de más y, como se puede ver en la gráfica 5-3(c) y en la tabla 5-6, no es mejor detectando estados débilmente enredados que W_o.

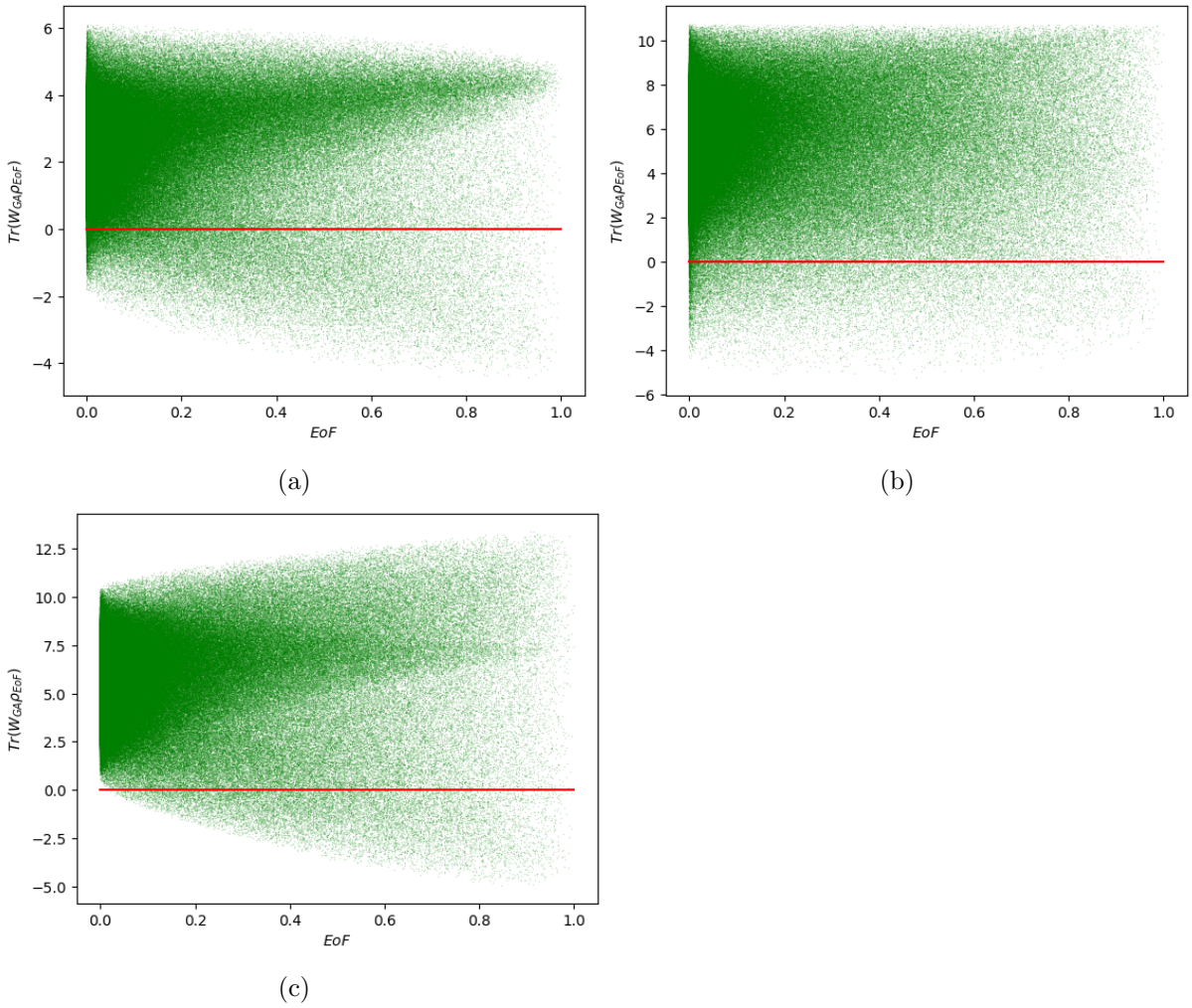


Figura 5-3: Algunos operadores obtenidos con `GA_varios_enredados`_enredados. (a) De nuevo vemos que la forma se asemeja a la del óptimo pero con mejor detección de estados poco enredados. (b) Un testigo que detecta varios estados débilmente enredados. (c) Un testigo con estructura menos definida pero parecido al óptimo.

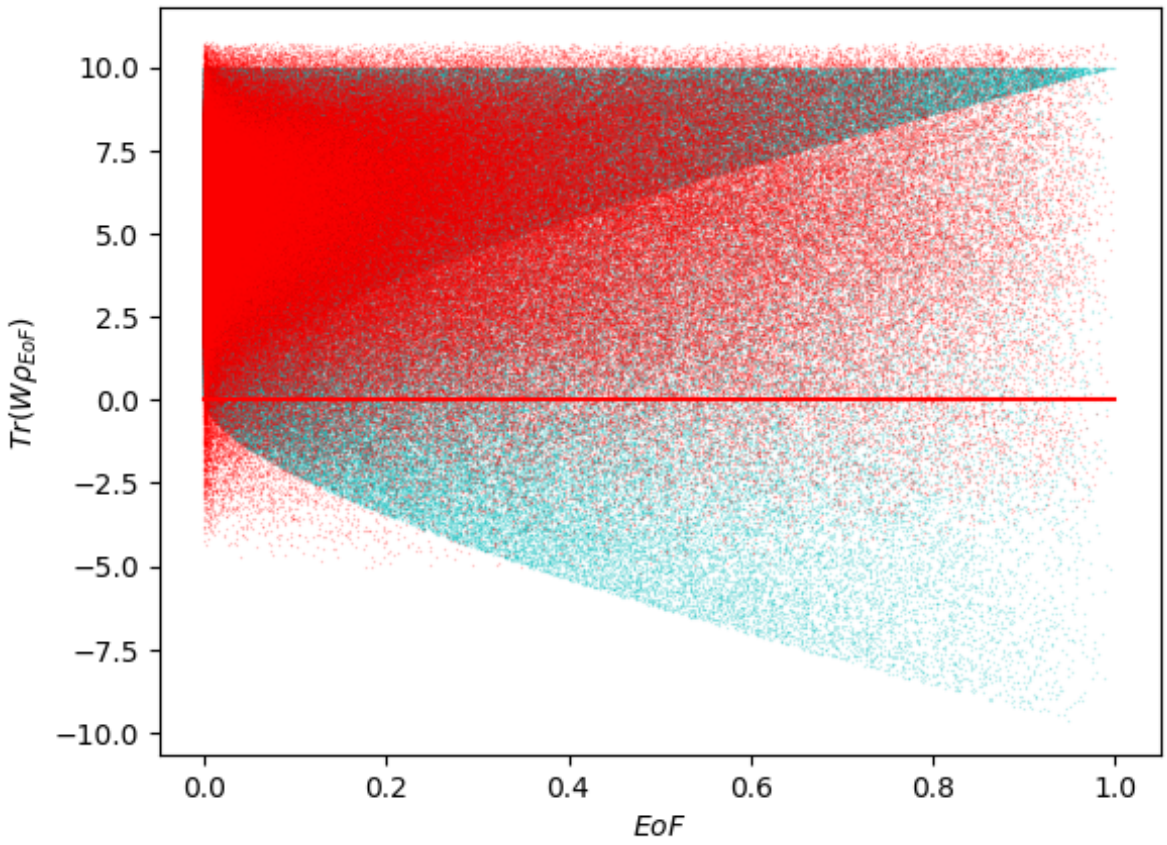


Figura 5-4: Comparación del testigo óptimo (azul) y $W_{2_{GA_{ve}}}$ (rojo).

5.3. Función de fitness `GA_Werner`

La función de fitness aquí usada es la que se discutió en la sección 4.3. Se genera una familia de estados débilmente enredados mediante transformaciones locales de un estado de Werner. n_{ew} es el número de estos estados.

n_s	5000
p	0.45
n_{ew}	14
Generaciones	350
Tamaño de población	52
Probabilidad de cruce	0.84
Probabilidad de mutación	0.047

Tabla 5-9: Tabla de parámetros utilizados en `GA_werner`.

Las gráficas de algunos de los operadores resultantes se dan en la figura 5-5.
Las matrices de estos operadores son:

$$W_{1_{GA_w}} = \begin{pmatrix} 3.5209 & 0 & 0 & 0 \\ 0 & 0 & 4.43096 - 0.574692i & 0 \\ 0 & 4.43096 + 0.574692i & 2.94662 & 0 \\ 0 & 0 & 0 & 7.44841 \end{pmatrix} \tag{5-8}$$

$$W_{2_{GA_w}} = \begin{pmatrix} 1.26725 & 0 & 0 & 3.03709 + 3.60107i \\ 0 & 6.53231 & 0 & 0 \\ 0 & 0 & 5.4016 & 0 \\ 3.03709 - 3.60107i & 0 & 0 & 0 \end{pmatrix} \tag{5-9}$$

$$W_{3_{GA_w}} = \begin{pmatrix} 0 & 0 & 0 & 2.9369 - 3.84165i \\ 0 & 5.36044 & 1.48164 - 1.40864i & 0 \\ 0 & 1.48164 + 1.40864i & 9.17426 & 0 \\ 2.9369 + 3.84165i & 0 & 0 & 0 \end{pmatrix} \tag{5-10}$$

$$W_{4_{GA_w}} = \begin{pmatrix} 3.1636 & 0 & -2.77763 + 1.58394i & 0 \\ 0 & 0 & 0 & 0 \\ -2.77763 - 1.58394i & 0 & 3.5568 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{5-11}$$

En la tabla de desempeño 5-10 se calcula el fitness de los operadores usando los n_{ew} estados enredados.

	Fitness	Detectados	No detectados	Bien clasificados	Mal clasificados
W_o	0	37405	374773	87822	0
$W_{1_{GA_w}}$	9	13316	398862	87822	0
$W_{2_{GA_w}}$	8	19920	392258	87822	0
$W_{3_{GA_w}}$	9	23610	388568	87822	0
$W_{4_{GA_w}}$	0	0	412178	87822	0

Tabla 5-10: Tabla de desempeño. La fitness es medida usando a los estados generados con estados de Werner.

Nuevamente se da la tabla de desempeño de cada testigo por intervalos de EoF, tabla 5-11.

	Detectados				Enredados
Intervalo	W_o	$W_{1_{GA_w}}$	$W_{2_{GA_w}}$	$W_{3_{GA_w}}$	
0-0.1	5537	478	5701	7735	205574
0.1-0.2	6339	1135	2993	3846	61919
0.2-0.3	5785	1652	2442	2990	41860
0.3-0.4	5051	1917	2107	2399	30908
0.4-0.5	4366	1976	1860	1928	23603
0.5-0.6	3720	1907	1556	1583	18159
0.6-0.7	2829	1735	1293	1308	13527
0.7-0.8	2060	1350	1043	960	9328
0.8-0.9	1263	854	686	643	5455
0.9-1	455	312	239	218	1845

Tabla 5-11: Detección por intervalos.

5.3.1. Análisis

Este algoritmo presenta un nuevo tipo de observable, la de la ecuación (5-11), $W_{4_{GA_w}}$. Este operador no es un testigo, ya que no detecta a ninguno de los estados enredados pero, por otro lado, tampoco clasifica mal a ninguno de los estados separables. La aparición de esta clase de operadores se podría explicar si asumimos que es más difícil detectar estados muy débilmente enredados. Así, operadores para los cuales no se clasifique mal llegarán a un fitness 0 y se estabilizarán ahí.

De la tabla 5-10 vemos que el valor de este fitness de W_o es 0. Esto quiere decir que no detecta ninguno de los estados que se generaron para evalúar el fitness. También podemos notar de las gráficas (b) y (c) en 5-5 que esta función de fitness logra generar testigos que detectan enredamiento débil y en 5-5(a) y (5-8) encontramos un testigo de forma similar a W_o. El algoritmo ejecutado con esta función de fitness logra generar testigos que detecten estados débilmente enredados y esto queda claro al ver el primer renglón de la tabla 5-11. Para estados aún menos enredados se tiene la tabla siguiente que ayuda para confirmar la utilidad de esta función de fitness.

	Detectados			
Intervalo	W_o	$W_{1_{GA_w}}$	$W_{2_{GA_w}}$	$W_{3_{GA_w}}$
0-0.01	333	19	1469	2054

Tabla 5-12: Desempeño para enredamiento muy débil.

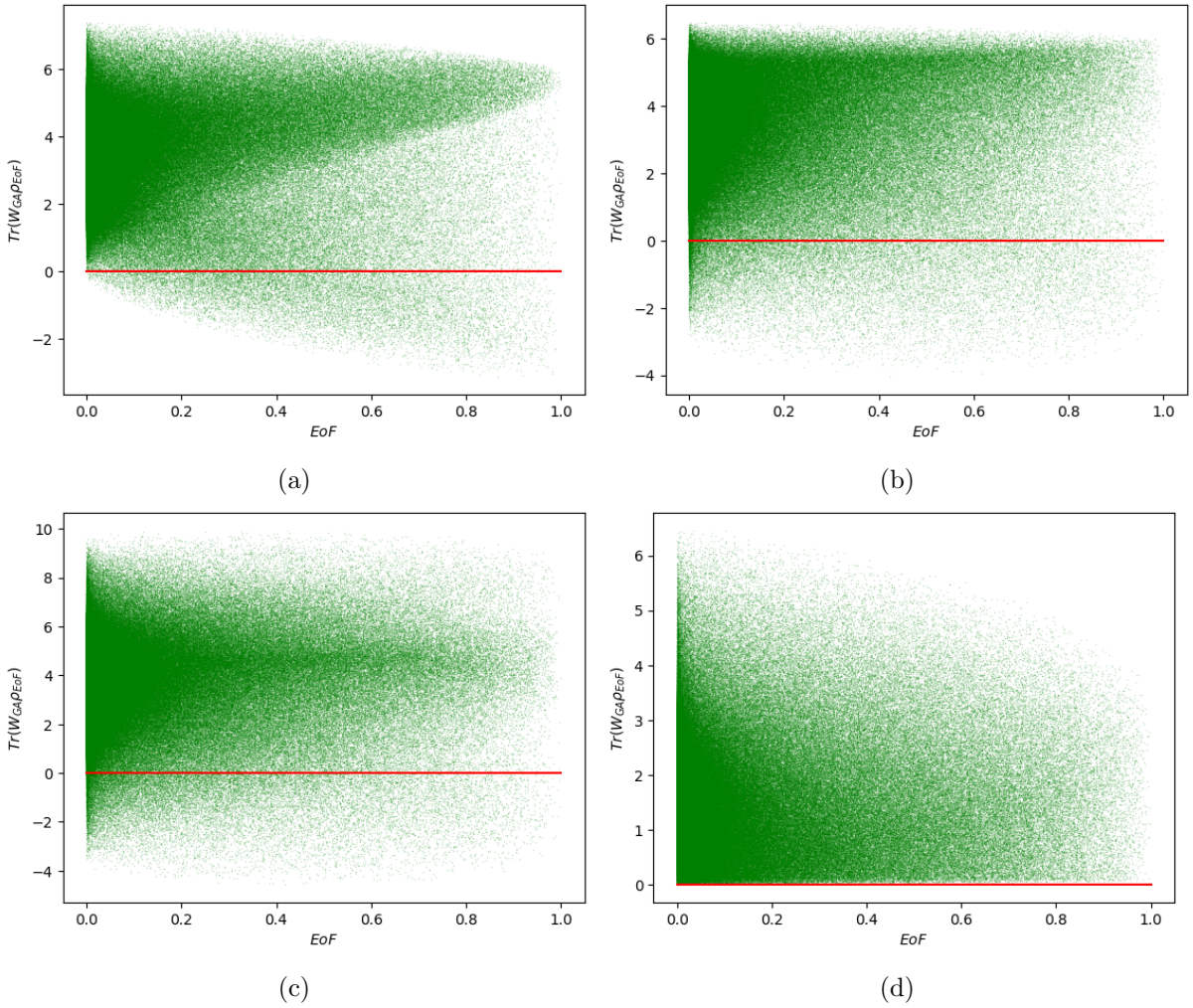


Figura 5-5: Algunos operadores obtenidos con `GA_werner`. (a) La forma se asemeja a la del óptimo. (b) Un testigo que detecta varios estados débilmente enredados. (c) Un testigo sin estructura clara. (d) Un operador que no detecta ningún estado enredado.

5.4. Combinación de varios testigos

En esta sección se estudia la cantidad de estados enredados que se pueden detectar juntando varios testigos que resulten de una corrida de los tres GA. Para esta sección se usan los mismos parámetros en todos los GA's:

n_s	5000
n_e	14
p	0.35, 0.40, 0.45
n_{ew}	14
Generaciones	350
Tamaño de población	52
Probabilidad de cruce	0.84
Probabilidad de mutación	0.052

Tabla 5-13: Tabla de parámetros.

Se corrieron 5 GA's, uno para la primera y segunda función de fitness y tres para la tercera función de fitness correspondientes a los valores de p de la tabla 5-11. De cada corrida tomamos a los mejores 5 individuos y nos quedamos con los que sí son testigos. De este proceso se obtuvieron 18 testigos válidos. Se generan 500,000 estados aleatorios y calculamos en orden $\mathrm{Tr}(W_i\rho)$ $(i = 1, 2, \ldots, 18)$ para todos ellos. Se guarda el valor para la primera i con la cual ocurra que $\mathrm{Tr}(W_i\rho) < 0$. Si para ninguno de los testigos ocurre esto se guarda el valor $\mathrm{Tr}(W_i\rho)$ para i escogida aleatoriamente entre 1 y 18. La gráfica que resulta de este proceso se puede ver en la figura 5-6.

Presentamos también la tabla de desempeño y denotamos por W a los 18 testigos juntos.

	Detectados	No detectados	Bien clasificados	Mal clasificados
W_o	36985	374770	88245	0
W	122552	289203	88245	0

Tabla 5-14: Tabla de desempeño.

El porcentaje de estados enredados detectados por W fue de 29 %, mientras que el testigo

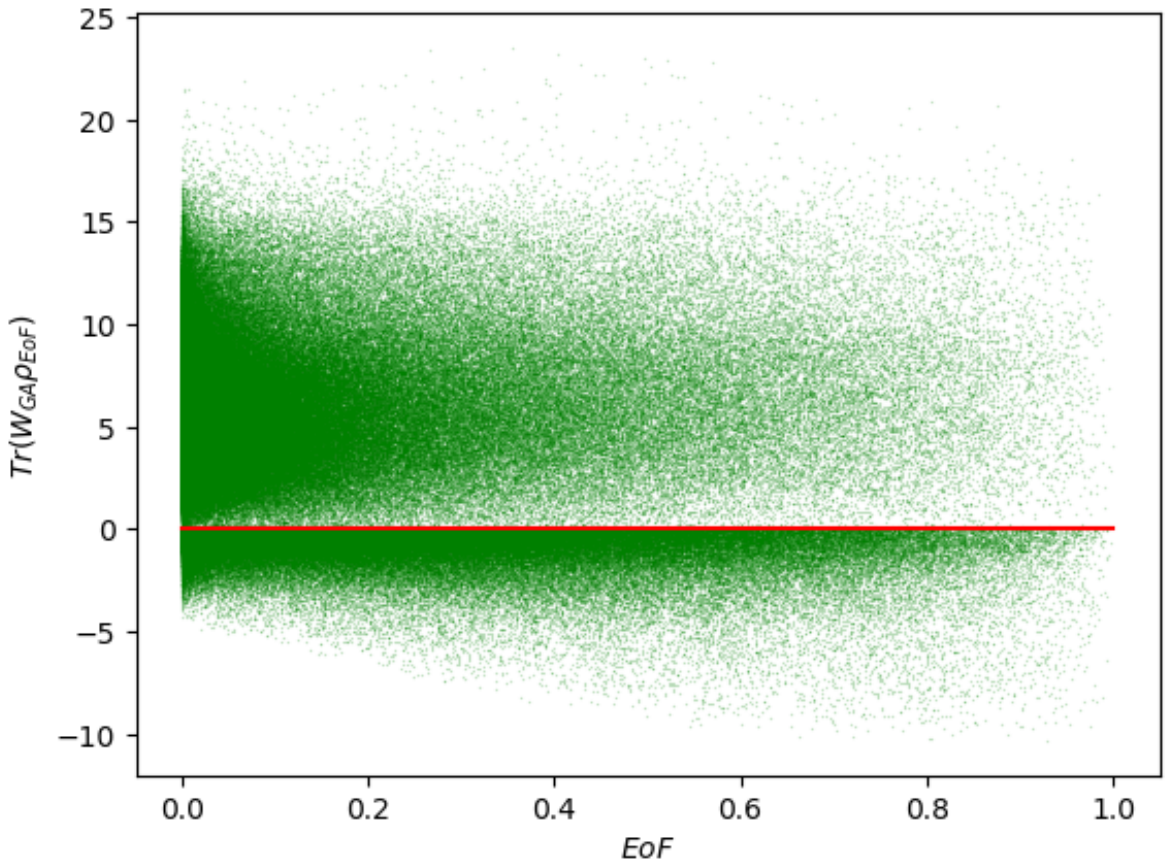


Figura 5-6: Estados clasificados con el conjunto de testigos.

óptimo por sí mismo detecta el 9 %. Es claro entonces que los GA's producen testigos para los cuales no se puede decir que W_o es más fino que ellos, ni ellos son más finos que W_o y logra cubrir mejor el conjunto de estados enredados. Adicionalmente, agregar W_o a W hace que el número de estados detectados sea 128083, es decir, se detectan 5531 estados más o alrededor de 4.5 % más. Es razonable suponer que correr el programa por más tiempo resulte en más testigos que hagan que la cantidad de estados detectados que agrega W_o se vuelva menor.

Capítulo 6

Conclusiones

Los algoritmos genéticos mostraron ser una buena herramienta para la construcción de testigos de enredamiento. Además de poder construir tantos testigos como se quiera (y permita el tiempo y poder de cómputo disponibles) se puede sesgar cada corrida para que sea mejor detectando ciertos intervalos de enredamiento, esto gracias a que se pueden generar tantos estados con una cantidad de enredamiento dada como se quieran y evalúar el fitness con estos estados.

El algoritmo no depende del tamaño del sistema y es posible generalizarlo a sistemas de más de dos qubit o sistemas cuánticos de más de dos niveles. Esto se podría lograr con ligeros cambios en el código pero en esencia quedaría igual. Se sabe que para estados de más de dos qubits existen varias clases no equivalentes de enredamiento (Horodecki *et al.* [2007]) y el criterio PPT deja de funcionar; existen estados PPT enredados. También se encuentra enredamiento bipartito o enredamiento real de todos los qubits que conformen el sistema. El algoitmo de construcción propuesto en esta tesis podría ser una forma de atacar el problema de enredamiento en sistemas complejos. Considero entonces que el trabajo realizado fue una introducción importante y original en esta dirección.

Apéndice A

Código

Código que genera a la población inicial:

```
#Funcion que genera los cromosomas iniciales
function generar_pob_inicial(N)
    pob_ini_cromosomas = Array{Array{Complex{Float64},1}}(undef,N)
    for i in 1:N
        aux=im*zeros(10)
        aux[1:4] = 5*randn(4)
        aux[5:10] = 5*randn(6) + 5*im*randn(6)
        pob_ini_cromosomas[i]=aux
    end
    pob_ini_cromosomas
end

#Funcion para pasar de cromosomas a matrices
function cromosoma_a_matriz(V)
    #V sera, en este caso, un arreglo de arreglos.
    #Los arreglos que lo conforman son los cromosomas.
    m = length(V)
    testigos_ini = Array{Array{Complex{Float64},2}}(undef,m)
```

```
    for i in 1:m
        testigos_ini[i]=[(V[i])[1] (V[i])[5] (V[i])[6] (V[i])[7];
                         ((V[i])[5])' (V[i])[2] (V[i])[8] (V[i])[9];
                    ((V[i])[6])' ((V[i])[8])' (V[i])[3] (V[i])[10];
                ((V[i])[7])' ((V[i])[9])' ((V[i])[10])' (V[i])[4]]
    end
    testigos_ini
end
```

A continuación se muestra el código que se usó para generar estados separables aleatorios:

```
#Funcion para generar estados separables aleatoriamente.
#Pueden ser puros o mixtos.
function estados_separables(M)
    separables=Array{Array{Complex{Float64},2}}(undef,M);
    for i in 1:M
        m=rand(0:1)
        if m==0
            phi1 = randn(2) + im* randn(2) #Estado de Alicia
            phi1=phi1/norm(phi1)
            phi2 = randn(2)                #Estado de Beto
            phi2=phi2/norm(2)
            psi = kron(phi1,phi2)
            #Lo guardamos como una matriz de densidad
            separables[i]=kron(psi,psi')
        else
            j=rand(2:8)          #Cuantos estados hay en la mezcla
            rhox = Array{Any}(undef,j)
            p=rand(j); p=p/sum(p)
            for l in 1:j
                phi1 = randn(2) + im* randn(2)  #Estado de Alicia
```

```
                phi1=phi1/norm(phi1)
                phi2 = randn(2)                    #Estado de Beto
                phi2=phi2/norm(phi2)
                psi = kron(phi1,phi2)
                rhox[l] = kron(psi,psi')
            end
            #Suma convexa de los estados
            separables[i]=sum((p.*rhox))
        end
    end
    separables
end
```

Función de fitness:

```
function fitness(W)

    fits = 0 #inicializamos el contador de fitness

    penalizacion = 20 #=penalizacion que habra por
    dar resultados negativos para estados separables =#

    for i in 1:n_s
        traza=real(tr(W * conj_rho[i]))
        if traza < 0
            fits = fits - penalizacion
        end
    end
    bell=zeros(4)
    bell[1] = real(tr(W * rho_phi_mas))
    bell[2] = real(tr(W * rho_phi_menos))
```

```
    bell[3] = real(tr(W * rho_phi_mas))
    bell[4] = real(tr(W * rho_phi_menos))
    for n in 1:4
        if bell[n]<0
            #=La logica detras de esto es que
            mientras mas enredado este el estado
            mas negativo ha de ser el valor esperado. =#
            fits = fits + ceil(3*abs(bell[n]))
        end
    end
    fits
    #Puede ser positivo o negativo, de momento no importa.
end

 #Funcion que genera 2n matrices unitarias aleatorias.
 function Unitaria(n)
     unitarias = Array{Array{Complex{Float64},2}}(undef,2*n)
     for i in 1:2*n
         xi = rand() ; alpha = rand() + 2*pi ;
         psi = rand() + 2*pi ; chi = rand() + 2*pi
         phi = asin(sqrt(xi))
         unitarias[i]=(exp(im*alpha).*
         [exp(im*psi)*\text{cos}(phi) exp(im*chi)*sin(phi);
         -exp(-im*chi)*sin(phi) exp(-im*psi)*\text{cos}(phi)])
     end
     unitarias
 end
```

Función de ruleta.

```
 #=Funcion de ruleta
 Condiciones, X[i]>0 o =0 para toda i. sum(X)=TOTAL BIEN DEFINIDO.
```

```
p ha de ser un numero aleatorio entre 1 y el TOTAL BIEN DEFINIDO
de modo que ocurra i con probabilidad X[i].
Funcion para que ocurra i dadas las probabilidades X[i].
Si X es un arreglo de ceros se escogera
 aleatoriamente entre 1 y length(X)
=#
function asignar_probabilidad(p,X)
    l=length(X);cont=1;lb=0
    if X==zeros(l)
        rand(1:l)
    else
        for i in 1:l
            if lb<=p<=lb+X[cont]
                break
            end
            lb=lb+X[cont]
            cont=cont+1
        end
        cont
    end
end
```

Operador de cruzamiento:

```
#Operador de cruzamiento
reparto=sum(valores_de_fitness)
#Puede ocurrir que toda la poblacion se estabilice en 0
if reparto==0
    reparto=2
end
#Se inicializa un contador de los nuevos "pobladores" de la generacion G
```

```
poblacion_nueva=1

while poblacion_nueva < tamano_poblacion+1
    #variables aleatorias para la funcion asignar_probabilidad
    p=rand(1:reparto) ; q=rand(1:reparto)
    individuo1=asignar_probabilidad(p,valores_de_fitness)
    individuo2=asignar_probabilidad(q,valores_de_fitness)
    #Intentamos evitar que sea el mismo individuo
    cont=1
    while individuo1==individuo2 && cont <10
        q=rand(1:reparto)
        individuo2=asignar_probabilidad(q,valores_de_fitness)
        cont+=1
    end

    prog_1=matriz_a_cromosoma(matrices[individuo1])   #Progenitores
    prog_2=matriz_a_cromosoma(matrices[individuo2])

    if rand()<probabilidad_cruce #No siempre se van a "cruzar"
        #=Se escoge aleatoriamente hasta que punto de la
        cadena del cromosoma aportara el individuo
        1(y por consiguiente a partir de cual punto de
        la cadena aporta el individuo 2): =#
    corte=rand(1:9)   #9 porque nuestros cromosomas tienen 10 genes
        #Esto hace la mezcla
pob_aux[poblacion_nueva]=vcat(prog_1[1:corte],prog_2[corte+1:10]...)
pob_aux[poblacion_nueva+1]=vcat(prog_2[1:corte],prog_1[corte+1:10]...)
    else
        #Si no se cruzan se quedan igual
        pob_aux[poblacion_nueva]=pob_ini[individuo1]
```

```
        pob_aux[poblacion_nueva+1]=pob_ini[individuo2]
    end
```

La mutación se realiza de la siguiente manera.

```
#Operador de mutacion

mutacion=rand()
if mutacion<probabilidad_mutacion
    for n in 1:4
        ra=rand()
        if ra<.2
            pob_aux[poblacion_nueva][n]=5*randn()
            pob_aux[poblacion_nueva+1][n]=5*randn()
        elseif  .2<=ra<.55
            pob_aux[poblacion_nueva][n]=0
            pob_aux[poblacion_nueva+1][n]=0
        elseif .55<=ra<.65
    pob_aux[poblacion_nueva][n]=pob_aux[poblacion_nueva][rand(1:4)]
    pob_aux[poblacion_nueva+1][n]=pob_aux[poblacion_nueva][rand(1:4)]
        end
    end
    for n in 5:10
        ra=rand()
        if ra<.2
            pob_aux[poblacion_nueva][n]=5*randn()+5*im*randn()
            pob_aux[poblacion_nueva+1][n]=5*randn()+5*im*randn()
        elseif .2<=ra<.55
            pob_aux[poblacion_nueva][n]=0
            pob_aux[poblacion_nueva+1][n]=0
        elseif .55<=ra<.65
```

```
        pob_aux[poblacion_nueva][n]=pob_aux[poblacion_nueva][rand(5:10)]
        pob_aux[poblacion_nueva+1][n]=pob_aux[poblacion_nueva][rand(5:10)]
            end
        end
    end
```

Bibliografía

Aspect, A., Grangier, P., y Roger, G. Experimental Tests of Realistic Local Theories via Bell's Theorem. *Phys. Rev. Lett.* **47**:460–463 (1981)

Aspect, A., Grangier, P., y Roger, G. Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell's Inequalities. *Phys. Rev. Lett.* **49**:91–94 (1982)

Audretsch, J. *Entangled Systems: New Directions in Quantum Physics.* 1ª edición. Wiley (2007)

Bell, J.S. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika* **1**:195–200 (1964)

Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., y Wootters, W.K. Mixed-state entanglement and quantum error correction. *Phys. Rev. A* **54**:3824–3851 (1996)

Chruściński, D. y Sarbicki, G. Entanglement witnesses: construction, analysis and classification (2014). `arXiv:1402.2413`

Einstein, A., Podolsky, B., y Rosen, N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Phys. Rev.* **47**:777–780 (1935)

Fyodorov, Y.V. Introduction to the Random Matrix Theory: Gaussian Unitary Ensemble and Beyond (2004). `arXiv:math-ph/0412017`

Gharibian, S. Strong NP-Hardness of the Quantum Separability Problem (2008). `arXiv:0810.4507`

Giraldi, G.A., Portugal, R., y Thess, R.N. Genetic Algorithms and Quantum Computation (2004). arXiv:cs/0403003

Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning.* 5ª edición. Addison-Wesley (1999)

Gühne, O. y Toth, G. Entanglement detection (2008). arXiv:0811.2803

Hensen, B., Bernien, H., Dréau, A.E., Reiserer, A., Kalb, N., Blok, M.S., Ruitenberg, J., Vermeulen, R.F.L., Schouten, R.N., Abellán, C., Amaya, W., Pruneri, V., Mitchell, M.W., Markham, M., Twitchen, D.J., Elkouss, D., Wehner, S., Taminiau, T.H., y Hanson, R. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature* **526**:682 EP – (2015)

Hill, S. y Wootters, W.K. Entanglement of a Pair of Quantum Bits. *Phys. Rev. Lett.* **78**:5022–5025 (1997)

Holland, J.H. *Adaptation in Natural and Artificial Systems.* 1ª edición. MIT Press (1975)

Horodecki, M., Horodecki, P., y Horodecki, R. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A* **223**(1):1 – 8 (1996)

Horodecki, R., Horodecki, P., Horodecki, M., y Horodecki, K. Quantum entanglement (2007). arXiv:quant-ph/0702225

Mermin, N.D. *Quantum Computer Science.* 1ª edición. Cambridge University Press (2007)

Mitchell, M. *An Intoruction to Genetic Algorithms.* 1ª edición. MIT Press (1989)

Navarro-Muñoz, J.C., Rosu, H.C., y López-Sandoval, R. Genetic algorithm optimization of entanglement. *Phys. Rev. A* **74**:052308 (2006)

Ozols, M. How to generate a random unitary matrix. Informe técnico (2009)

Ramos, R.V. y Souza, R.F. Calculation of the Quantum Entanglement Measure of Bipartite States, Based on Relative Entropy, Using Genetic Algorithms. *Journal of Computational Physics* **175**(2):576 – 583 (2002)

Terhal, B.M. Bell Inequalities and the Separability Criterion (1999). arXiv:quant-ph/9911057

Wang, B.H., Xu, H.R., Campbell, S., y Severini, S. Characterization and properties of weakly optimal entanglement witnesses (2014). arXiv:1407.0870

Wootters, W.K. Entanglement of Formation of an Arbitrary State of Two Qubits. *Phys. Rev. Lett.* **80**:2245–2248 (1998)

Zyczkowski, K., Penson, K.A., Nechita, I., y Collins, B. Generating random density matrices (2010). arXiv:1010.3570