



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**ANÁLISIS COMPARATIVO ENTRE EL ALGORITMO
CUÁNTICO DE GROVER Y UN ALGORITMO GRASP,
APLICADOS A LA BÚSQUEDA DE INDIVIDUOS ÓPTIMOS EN
LA POBLACIÓN INICIAL DE UN ALGORITMO GENÉTICO.**

Tesis para optar por el Título de Ingeniero Informático, que presenta el bachiller:

José Enrique Rivera Alejo

ASESOR: Mg. Martin Richard Kong Moreno

Lima, agosto del 2010

Resumen

Este trabajo trata sobre la aplicación de dos algoritmos de búsqueda a la selección de individuos óptimos en la población inicial de un algoritmo genético, y la consiguiente comparación entre ambos.

El primero de ellos es el algoritmo meta-heurístico GRASP, y el segundo es el algoritmo cuántico de Grover.

El algoritmo cuántico de Grover forma parte de una nueva generación en las ciencias de la computación: la computación cuántica. Y por tanto hace uso de conceptos matemáticos y físicos completamente distintos a los usados en la programación clásica.

En esta tesis se presenta un análisis general de ambos algoritmos, siendo de especial mención el análisis del algoritmo cuántico de Grover, ya que incluye un modelo matemático del funcionamiento del mismo. Este modelo será de suma importancia para simular la ejecución del algoritmo de Grover en una computadora clásica, dada la carencia de una computadora cuántica sobre la cual realizar esto.

Luego, se preparan dos procesos experimentales, los cuales se usarán para realizar la comparación de eficacia y eficiencia entre las ejecuciones de los dos algoritmos.

Posteriormente, se presenta el diseño e implementación de los algoritmos, ambos aplicados a la selección de individuos de un algoritmo genético genérico. Una vez ambos algoritmos se encuentren correctamente implementados y funcionales, se ejecutarán las pruebas experimentales que permitan realizar la comparación entre ellos.

Finalmente se realizan las conclusiones y observaciones del caso en base a los resultados numéricos obtenidos en la fase experimental.

Dedicado a:

*Mis padres.
Los lectores de este trabajo.*

Agradecimientos a:

*Dios, porque gracias a Él todo es posible.
Mi padre, mi madre y mi hermano, quienes me han apoyado en cada momento de mi vida.
Mi asesor por su apoyo durante el desarrollo de esta tesis.
Mis profesores por sus importantes enseñanzas a lo largo de mi vida académica.
Mis amigos por su ayuda, apoyo, ánimo y compañía, y especialmente a quien siempre me deseó suerte en todo y a quien me hizo abrir los ojos nuevamente a la investigación.*

*A todos ustedes, **muchas gracias.***

Tabla de Contenido

Introducción.....	1
1. Capítulo I: Generalidades.....	4
1.1 Definición del Problema.....	4
1.2 Marco Conceptual del Problema.....	6
1.3 Estado del Arte.....	12
1.3.1 Investigaciones realizadas.....	13
1.4 Descripción y Sustentación de la Solución.....	18
2 Capítulo II: Aporte de la Investigación.....	21
2.1 Aportes específicos.....	21
2.2 Desarrollo de la Metodología.....	23
2.2.1 Observación.....	23
2.2.2 Planteamiento de Hipótesis:.....	24
2.2.3 Experimentación.....	24
2.2.4 Conclusión.....	25
2.3 Lista de Resultados Esperados.....	26
2.4 Planteamiento de hipótesis.....	27
3 Capítulo III: Implementación del Aporte.....	32
3.1 Definición de los Experimentos.....	32
3.1.1 Selección de Individuos óptimos (Experimento-1).....	33
3.1.2 Experimentación Numérica.....	35
3.2 Definición de los Algoritmos.....	36
3.2.1 Definición del Algoritmo Genético.....	36
3.2.2 Definición del Algoritmo GRASP.....	41
3.2.3 Definición del Algoritmo Cuántico de Grover.....	42
3.3 Diseño de los Algoritmos.....	48
3.3.1 Diseño del Algoritmo GRASP.....	48
3.3.2 Diseño del Algoritmo Cuántico de Grover.....	50
3.4 Implementación de los Algoritmos.....	57
4 Experimentación Numérica.....	59
4.1 Ejecución de la Experimentación Numérica.....	60
4.1.1 Antes de la ejecución de la Experimentación Numérica.....	61
4.1.2 Prueba T-Student para dos muestras relacionadas.....	64
4.2 Resultados.....	64

5	Observaciones, conclusiones y recomendaciones	78
5.1	Contraste de Hipótesis	78
5.2	Conclusiones	80
5.3	Observaciones.....	82
5.4	Recomendaciones y trabajos futuros	84
6	Referencias	86



Índice de Figuras

Figura 4.1: Gráfica: Factor de Relajación (α) vs Valor de Bondad	62
Figura 4.2: Gráfica: Cantidad de Iteraciones vs Valor de Bondad	62
Figura 4.3: Análisis de Eficacia	73
Figura 4.4: Análisis de Eficiencia	74
Figura 4.5: Análisis de Bondad	75
Figura 4.6: Análisis de Bondad tras ajustes en Ponderación	77
Figura 5.1: Gráfica Valores de Bondad Promedio vs Razón Eficacia- Eficiencia/Complejidad	83



Índice de Tablas

Tabla 1.1: Cuadro Comparativo de las Investigaciones Encontradas	17
--	----



Introducción

La computación cuántica aparece como una nueva rama de las Ciencias de la Computación, la cual promete la posibilidad de realizar el procesamiento de inmensas cantidades de información en tiempos millones de veces menores de lo que son capaces las computadoras clásicas hoy en día.

La computación cuántica está atravesando una etapa de intensa investigación alrededor del todo el mundo, por ejemplo, en los laboratorios de entidades como IBM, Microsoft, Google y el Instituto Tecnológico de Massachusetts. Se perfila como la tecnología que reemplazará eventualmente a la computación clásica tal cual la conocemos hoy en día, gracias a los múltiples beneficios que supone.

La investigación en el campo de la computación cuántica es solo parte de un campo mucho más amplio conocido como información cuántica. En lo que respecta a su rama más cercana a las Ciencias de la Computación, se puede afirmar que en la actualidad ya han sido propuestos varios algoritmos de naturaleza cuántica, los cuales garantizan podrán realizarse operaciones que hoy en día tomarían miles o millones de años, en tan solo segundos. Uno de estos algoritmos, es el algoritmo cuántico de Grover, un algoritmo cuántico de búsqueda, especialmente útil en grandes universos de elementos sin ningún tipo de ordenamiento a priori.

El algoritmo cuántico de Grover es capaz de resolver búsquedas con una velocidad mucho mayor que los algoritmos clásicos con los que contamos hoy. La razón entre ambas velocidades demuestra que la velocidad del algoritmo de Grover es aproximadamente el cuadrado de la velocidad del mejor algoritmo clásico de búsqueda en un universo grande de elementos sin ordenar.

Lo que se plantea en esta tesis es recoger el conocimiento y fundamento físico-matemático del algoritmo de Grover, a fin de poder simular su implementación en un entorno clásico, analizar su eficiencia aplicada a un

caso en particular y compararla con un algoritmo de búsqueda clásico aplicado al mismo caso.

El caso que se ha decidido tomar como base para la aplicación (y posterior comparación) de ambos algoritmos, es la selección de individuos óptimos en la población inicial de un algoritmo genético.

Los algoritmos genéticos son una clase de algoritmo de la rama de la computación evolutiva, los cuales a través de la simulación de ciertos fenómenos de la naturaleza, buscan dar como respuesta final la solución óptima (o la más cercana a la óptima) de un problema. Los problemas que se tratan de resolver con los algoritmos genéticos, son en la mayoría de casos aquellos que son conocidos como problemas NP. Estos problemas se caracterizan por no tener una solución exacta, o, en todo caso, porque el cálculo de la solución exacta requiere un cómputo exhaustivo, el cual tomaría una cantidad excesiva de tiempo en finalizar (desde días hasta incluso miles de años).

Una de las fases características de un algoritmo genético es la selección de los mejores individuos de su población. Para resumir, un algoritmo genético cuenta con un conjunto de posibles soluciones al problema que busca resolver, este conjunto se conoce como su “población”, y cada posible solución se conoce como “individuo”. El algoritmo genético generalmente se apoya en otros algoritmos de búsqueda y optimización para seleccionar los mejores individuos de un universo y formar una población inicial, la cual irá mejorando progresivamente y, por tanto, irá acercándose a la mejor solución. Uno de los mejores algoritmos clásicos que se usan hoy en día para esta labor es el que se conoce como algoritmo GRASP (Greedy Randomized Adaptive Search Procedure o Procedimiento de Búsqueda Adaptativa Aleatoria y Golosa).

Tanto el algoritmo cuántico de Grover como el algoritmo GRASP serán aplicados a la selección de la población inicial de un algoritmo genético. No se definirá una aplicación particular al algoritmo genético, pues no es parte del objetivo de la presente tesis.

Se usarán los conceptos fundamentales de los algoritmos genéticos de manera genérica a fin de que los resultados obtenidos de la comparación

entre el algoritmo de Grover y el algoritmo GRASP sean válidos en cualquier aplicación concreta del algoritmo genético.

Por último, esta tesis busca fomentar el deseo de investigación en la rama de la computación cuántica, dentro del país, demostrando que existe un campo muy amplio para la aplicación de los algoritmos cuánticos en la resolución más eficiente de problemas ya presentes hoy en día.



1. Capítulo I: Generalidades

1.1 Definición del Problema

En este apartado, primero se presentará el campo de la computación cuántica, posteriormente se explicará brevemente qué es un algoritmo genético y cuál es la fase que se plantea optimizar.

La computación cuántica es un campo emergente y nuevo en las Ciencias de la Computación, el cual hace uso directo de los principios físicos y matemáticos que rigen la mecánica cuántica para poder lograr procesamiento de la información con mucha mayor eficiencia que la computación clásica.

La programación cuántica difiere de la clásica pues usa conceptos inherentes a la matemática y física de la mecánica cuántica, tales como amplitudes de probabilidad y superposición de elementos [1]. Cada uno de estos conceptos está fuertemente relacionado con la computación cuántica, desde sus elementos tangibles como las computadoras cuánticas hasta sus elementos intangibles como los algoritmos cuánticos.

El tratamiento de los elementos probabilísticos propios de la mecánica cuántica está siempre presente en el manejo de algoritmos y programación cuántica. Puede tomarse por ejemplo, que en todo momento al trabajar con qubits (equivalente cuántico de los bits), hay una probabilidad que determina el estado en el cual se encuentren (si en cero, en uno, o en superposición). Otro elemento propio de la computación cuántica es precisamente la superposición de estados (un qubit puede estar en 0 y 1 a la vez) lo cual crea un paralelismo completamente distinto del manejado por la programación clásica [2].

Un algoritmo genético es un tipo de algoritmo de la familia de la computación evolutiva. Su lógica y funcionamiento están inspirados en la teoría de la selección natural, herencia y la evolución. Los algoritmos genéticos son ampliamente usados en problemas de optimización y para buscar soluciones aproximadas a los problemas del tipo NP, los cuales no cuentan con una solución exacta, o, si en caso la tengan, resulta computacionalmente exhaustivo encontrarla, pues el tiempo necesario está en el orden de hasta cientos, miles o millones de años, de acuerdo a la magnitud del problema.

Un elemento propio de los algoritmos genéticos es su población de individuos. En el contexto de un algoritmo genético, los individuos que conforman su población son posibles soluciones al problema planteado y en las consiguientes iteraciones se simulará una progresiva evolución de esta población: los individuos “malos” (soluciones no-óptimas) son eliminados y los individuos “buenos” (soluciones cercanas a las óptimas) generan nuevos individuos los cuales heredan lo mejor de sus padres, es decir, se vuelven mejores candidatos de la solución general.

Una de las fases cruciales del algoritmo genético es la selección de la población inicial, es decir, la selección de los individuos “buenos” discriminando los individuos “malos” de todo el universo de candidatos. Este proceso es crítico pues los algoritmos genéticos no trabajan directamente con todo el universo de posibles soluciones, sino que necesitan un subconjunto de dicho universo, el cual permita que la labor sea menos exhaustiva. Los algoritmos genéticos utilizan en la mayoría de casos un algoritmo adicional para realizar esta selección. Uno de los algoritmos que se usan por excelencia en este tipo de casos, es el algoritmo GRASP, un algoritmo meta-heurístico.

Sin bien el algoritmo GRASP resulta efectivo al seleccionar una población inicial de individuos “buenos”, para que funcione correctamente se requiere una gran cantidad de iteraciones del mismo. Además, para que el algoritmo genético funcione mejor, lo recomendado es trabajar con grandes poblaciones. Siendo este el caso nos encontramos con una gran cantidad de búsquedas en un gran universo de elementos. Esto se traduce en un gran tiempo computacional, que si bien no demora años, ni siglos, puede tomar largas horas y días. Es precisamente esta fase de selección la cual se plantea optimizar a fin de poder conseguir mejores resultados con una mayor eficiencia. Usando los nuevos conceptos emergentes de computación cuántica y algoritmos cuánticos ya descritos líneas arriba, se planteará el uso del algoritmo cuántico de Grover para la selección de los mejores individuos de un universo, los cuales conformarán la población inicial de un algoritmo genético genérico. Posteriormente, se compararán estos resultados con los que se hubieran obtenido si se usara el algoritmo GRASP.

1.2 Marco Conceptual del Problema

Se explicarán las nociones básicas de la computación cuántica, enfatizando en aquellas características que permitan comprender el funcionamiento del algoritmo de Grover. Además se explicará el funcionamiento del algoritmo genético y el algoritmo GRASP.

La Computación Cuántica

La computación cuántica es una nueva rama tecnológica y científica que aparece como una poderosa alternativa para realizar procesamiento masivo de datos en menor tiempo de lo que se consigue con las computadoras que se usan en la actualidad. El comportamiento de una computadora cuántica está gobernado por las leyes de la mecánica cuántica [1]

En general, se usa la frase “computación cuántica” para describir procesos computacionales cuya eficiencia depende de las propiedades de la mecánica cuántica en el manejo de la información. [8]

¿Cómo aparece la computación cuántica?

La ley de Moore establece que el número de transistores presentes en un microprocesador va duplicándose cada 18 meses. Esto presupone que entre los años 2020 y 2030, los circuitos de procesamiento habrán llegado a una escala atómica. Es aquí donde se plantea la aparición de computadoras cuánticas, las cuales usarán el potencial de átomos y moléculas para realizar tareas de memoria y procesamiento [7].

¿Por qué la computación cuántica es un tema relevante?

Los mecanismos de procesamiento de información que se usan en la computación cuántica son notablemente más eficientes que los usados en la computación clásica. La capacidad de procesamiento de una computadora cuántica resulta ser exponencialmente mayor que la de una computadora clásica. Como se verá más adelante, el fenómeno de superposición en la mecánica cuántica permite que las unidades de información de la computación cuántica (qubits) aumenten exponencialmente la capacidad presente en las unidades de información actuales (bits).

Se puede considerar el siguiente ejemplo: la tarea de factorizar un número de 5000 dígitos es una tarea computacionalmente exhaustiva, irrealizable para una computadora clásica. Usando el mejor algoritmo clásico con el que se cuenta hasta la fecha, el tiempo de ejecución estimado es de 5 trillones de años. Este tiempo de ejecución sobrepasa incluso la edad del universo. Sin

embargo, si la misma tarea fuera realizada por el algoritmo cuántico de Shor, una computadora cuántica se demoraría tan solo dos minutos.

El ahorro exponencial de recursos provisto por la computación cuántica lo hace un tema sumamente atractivo en la comunidad científica mundial y está siendo investigado activamente por entidades tales como el *Massachusetts Institute of Technology* (MIT) o *IBM Research* entre muchas otras. [1]

Conceptos básicos de Computación Cuántica

Como ya se mencionó antes, la computación cuántica tiene principios basados en las leyes de la mecánica cuántica. Algunos conceptos importantes son:

1. Las probabilidades:

En la computación cuántica las probabilidades de un hecho deben ser siempre consideradas, aun si el hecho en cuestión nunca sucede [1].

Tanto una computadora cuántica como un algoritmo cuántico son sistemas probabilísticos, esto quiere decir que por su propia naturaleza nunca se puede saber con seguridad cuál es el estado del sistema. Sin embargo sí se puede saber la distribución de probabilidad de dichos estados. Los cálculos que se realizan en la computación cuántica son probabilísticos. Es por ello que, al igual que en la mecánica cuántica, nunca puede darse una respuesta cien por ciento segura, sino una respuesta probabilísticamente aceptable.

El término asociado a la *probabilidad* en la mecánica cuántica (y por tanto en la computación cuántica) se denomina **amplitud** y puede incluir términos negativos e incluso imaginarios. [1]

2. La superposición cuántica:

La mínima unidad de información en la computación clásica es el bit. Un bit tiene dos estados, puede estar en 0 (apagado) o 1 (prendido), siempre estará únicamente en uno de estos dos estados.

Por su parte, la computación cuántica tiene como mínima unidad de información al qubit. Un qubit también puede existir en alguno de los dos estados de la computación clásica: 0 y 1, sin embargo, también puede existir en una superposición de ambos. [3]

Esto puede explicarse de la siguiente forma:

- Un bit tiene dos estados posibles asociados. Si el bit está prendido, la probabilidad de que esté prendido es evidentemente uno, y la probabilidad de que esté apagado es cero. Notamos que las probabilidades son absolutas.
- Un qubit también tiene dos estados posibles asociados. Sin embargo, un qubit tiene una probabilidad de estar prendido y otra probabilidad de estar apagado. Estas probabilidades como se mencionó anteriormente, en la computación cuántica están asociadas a elementos denominados **amplitudes** y son términos imaginarios.

Así como un qubit puede existir en superposición de estados. El mismo principio es aplicable a registros enteros de qubits. Si se tiene un registro de “n” qubits, el principio de superposición de la mecánica cuántica dice que se podrían tener hasta 2^n estados a la vez. Dicho de otra forma, un registro de “n” qubits es equivalente a 2^n registros de “n” bits. En conclusión, un qubit tiene una capacidad de procesamiento exponencialmente mayor a la de un bit convencional gracias al principio de superposición.

3. *Sistemas cuánticos:*

Al considerar un sistema cuántico de dos qubits donde existen 4 posibles estados superpuestos a la vez., la notación cuántica es la siguiente: [1]

$$\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$$

$$\sum |\alpha_i|^2 = 1$$

Esto quiere decir que el estado del sistema está determinado por la superposición de todos los estados posibles. Dos qubits determinan que los cuatro posibles estados del sistema son: (00), (01), (10) y (11). Cada uno de estos posibles estados tiene asociada una **amplitud** α_i , cuyo valor elevado al cuadrado determina la probabilidad de que el sistema colapse a dicho estado. Además, la suma de las probabilidades de todos los casos posibles en un sistema debe ser siempre uno. En el sistema cuántico mostrado, la

probabilidad efectiva de cada caso está dada por el cuadrado del módulo de la amplitud asociada. Entonces la suma de los cuadrados del módulo de cada amplitud, es uno.

¿Qué es el algoritmo de Grover?

El algoritmo de Grover es, como su creador, Lov Grover, lo definió: “Un algoritmo mecánico cuántico rápido para búsqueda en base de datos” (“A fast quantum mechanical algorithm for database search”) [11].

El algoritmo de Grover realiza búsquedas de elementos dentro de un universo sin ordenar, con una velocidad pasmosa que supera de manera cuadrática la velocidad de un algoritmo clásico.

Clásicamente, si se tiene un universo con elementos sin ordenar, la opción más segura que queda es revisar secuencialmente cada uno de los elementos (podría ser en orden aleatorio) a fin de poder encontrar el elemento deseado. Si el universo consta de ‘N’ elementos, clásicamente, en promedio, se necesitan $N/2$ intentos para encontrar el elemento deseado.

Sin embargo, el algoritmo de Grover puede realizar búsquedas en un universo de elementos desordenados en “la raíz cuadrada de N” iteraciones. Es decir, si el universo es de 10000 elementos, clásicamente se necesitarían en promedio 5000 iteraciones, sin embargo, usando el algoritmo de Grover solo son necesarias 100 iteraciones. Y esta diferencia se hace más evidente al hacer ‘N’ un número más grande.

¿Cómo funciona un algoritmo genético?

Un algoritmo genético, como se mencionó líneas arriba, está inspirado en los fenómenos naturales de selección natural, herencia y evolución. Dependiendo del problema que se esté tratando de resolver, primero se establecerá un conjunto inicial de soluciones candidatas (individuos de la población inicial). Todas las soluciones son correctas, sin embargo, algunas son mejores que otras, el objetivo es buscar la solución óptima, es decir, el individuo “más apto” de toda la población.

La función de fitness, es un indicador que permite distinguir si una solución candidata es mejor que otra. La función de fitness asigna a cada individuo un valor numérico el cual determina cuán óptimo es dicho individuo, es decir, cuan buena es la solución. [18]

El algoritmo genético simula la selección natural, discriminando los individuos con menor valor de función de fitness y promoviendo que los individuos con alto valor de función de fitness, tengan “hijos”, los cuales hereden lo mejor de sus padres, es decir, se volverán soluciones mejores.

Los algoritmos genéticos no trabajan con el universo entero de posibles soluciones al problema, pues no es parte de sus características ser exhaustivos, ya que tomaría una mayor cantidad de tiempo computacional. Es por ello que una fase muy importante en la ejecución de un algoritmo genético, es seleccionar la población inicial, la cual es un subconjunto del universo total de soluciones. Es ventajoso que los individuos de la población inicial tengan un alto grado de bondad según la función de fitness.

El objetivo de la fase de selección de la población inicial es eminentemente una **búsqueda**: El universo son todos los individuos de la población, (los cuales no se encuentran ordenados), y lo que se está buscando son los individuos con mayor valor de función de fitness.

Es este proceso de selección el cual será simulado utilizando primero un algoritmo GRASP y luego el algoritmo cuántico de Grover. No es parte del objetivo de esta tesis simular el proceso evolutivo de la población inicial seleccionada.

El algoritmo genético también simula otros procesos propios de la naturaleza, como la eventual aparición de mutaciones y la eliminación de aberraciones, sin embargo la simulación de estos procesos no forma parte del objetivo de esta tesis y, por lo tanto, no serán considerados.

¿Qué es un algoritmo GRASP?

Es un algoritmo meta-heurístico usado generalmente para hallar un conjunto de soluciones candidatas en problemas de optimización combinatoria. El algoritmo GRASP evita realizar una búsqueda exhaustiva de las combinaciones de elementos que pueden considerarse solución, y en lugar de eso, aplica un criterio voraz a la par de un grado de variabilidad en la selección. Esta variabilidad surge como resultado de seleccionar ciertos elementos del universo de una forma aleatoria, estos elementos seleccionados al azar son asignados como parte de una solución.

Es esta variabilidad la que “relaja” la voracidad natural del algoritmo a fin de evitar las desventajas naturales de un algoritmo voraz heurístico, como la selección de máximo locales en lugar del máximo global.

El algoritmo GRASP es ampliamente utilizado en problemas de tipo NP a fin de obtener un conjunto de posibles soluciones. La eficacia del algoritmo GRASP aumenta con la cantidad de iteraciones, sin embargo, a la par, el tiempo de ejecución también aumenta de manera considerable. Esta es una de las razones por las cuales el algoritmo cuántico de Grover surge como una alternativa muy atractiva para superar este inconveniente.

1.3 Estado del Arte

Actualmente existen numerosos estudios sobre computación y algoritmos cuánticos, la mayor parte de ellos son realizados en Estados Unidos y parte de Europa, si bien en Latinoamérica, México sobresale como uno de los países con la mayor cantidad de investigaciones activas realizadas sobre el tema.

Los estudios realizados sobre las aplicaciones al algoritmo de Grover son muy variados y han servido de gran apoyo al desarrollo de la presente tesis. En particular, hay un estudio de la Universidad Politécnica de Timisoara en Rumania en el cual se propone el modelo de un algoritmo genético eminentemente cuántico. El funcionamiento de este algoritmo sigue estando inspirado en los procesos de evolución y selección natural, sin embargo con ciertos cambios que lo permitan amoldarse más a la lógica y principios de la mecánica cuántica.

Este algoritmo genético cuántico tiene como motor de búsqueda al algoritmo de Grover, el cual sirve para realizar una veloz y eficiente selección de las mejores soluciones en cada iteración.

Respecto a los algoritmos genéticos y a los algoritmos GRASP, las investigaciones y estudios que se han realizado son bastante amplios y diversos. Hay una gran cantidad de aplicaciones prácticas a los algoritmos genéticos, desde optimización de rutas hasta planeamiento de carteras de proyectos de inversión.

Las investigaciones que han sido de gran ayuda para la elaboración de la presente tesis son aquellos que tratan de computación cuántica, algoritmos cuánticos, su formulación, fundamento e implementación. Fundamentalmente aquellas que logran una cierta unión entre los algoritmos clásicos y cuánticos en lo que son conocidos como algoritmo híbridos. Dado que en esta tesis se plantea la implementación del algoritmo cuántico de Grover en un entorno clásico, es de mucha ayuda conocer los resultados de investigaciones ya realizadas, que han intentado y en muchos casos armonizar dos paradigmas tan diferentes.

1.3.1 Investigaciones realizadas

Implementing Quantum Genetic Algorithms: a Solution based on Grover Algorithm

Esta investigación llevada a cabo en la Universidad Politécnica de Timisoara en Rumania, plantea que es posible una “conjunción amistosa” entre dos ramas de las ciencias de la computación altamente prometedoras: la computación cuántica y los algoritmos genéticos. [5]

Para ello se apoyan en lo que se conoce como Programación Cuántica Evolutiva (Quantum Evolutionary Programming – QEP), la cual viene a ser una nueva rama en las ciencias de la computación que engloba e incluye la lógica y fundamento de los algoritmos genéticos en conjunto con los principios de la mecánica y computación cuántica. Lo que la investigación plantea es que es posible realizar un algoritmo cuántico genético (Quantum Genetic Algorithm- QGA), si bien, también es descrita como una tarea ardua, pues en la actualidad no existe ninguna base, ni idea coherente sobre la cual plantear uno.

En la computación clásica, la población de individuos (soluciones candidatas) que se selecciona inicialmente y la cual irá siendo “mejorada” progresivamente a lo largo de las iteraciones, es solo una muestra del universo total de posibles soluciones al problema. Esto es un prerequisite indispensable para todo algoritmo genético ya que, una de los objetivos de este algoritmo es evitar realizar cálculos exhaustivos, por ejemplo, procesar todas las posibles soluciones al problema. Esto tomaría un tiempo

computacional excesivo y el algoritmo perdería su eficiencia que lo caracteriza resolviendo problemas.

Sin embargo, esta investigación plantea que un algoritmo genético cuántico debe incluir toda la población para poder encontrar de forma efectiva y exacta la mejor solución de todas. En este caso, no existe el problema del excesivo tiempo computacional requerido para seleccionar las mejores soluciones de toda la población, ya que la eficiencia del algoritmo de Grover logra que incluso para grandes cantidades de individuos, el ahorro de tiempo computacional sea notable. Esta diferencia en cuanto al tamaño de la población inicial es una de las diferencias que presenta el algoritmo genético cuántico frente a su homólogo clásico [5]

Los objetivos principales de la investigación fueron:

- “Plantear un modelo cuántico del algoritmo genético clásico”. [5]
- “Evaluar la reducción del algoritmo genético cuántico a una búsqueda cuántica con el algoritmo cuántico de Grover. [5]

En el paper se describen principalmente:

- Las fases de un algoritmo cuántico genético cuántico en contraste al algoritmo genético clásico.
- Pseudocódigo del algoritmo genético cuántico.
- Modelo matemático del algoritmo genético cuántico
- Formulación de la función oráculo que se usará en el algoritmo cuántico de Grover, el cual forma parte del algoritmo genético cuántico.

Finalmente las conclusiones de la investigación fueron las siguientes:

- Es posible reducir cualquier problema que se trate de resolver con el algoritmo genético clásico, a una búsqueda cuántica del individuo con mayor función de fitness. Sin destruir la correlación entre los valores de fitness individuales [5]

- El algoritmo de Grover resulta especialmente efectivo al resolver los problemas a los cuales está dirigido el algoritmo genético cuántico. Es un buen ejemplo de una aplicación efectiva de búsqueda usando principios de computación cuántica [5]

El método científico usado por el equipo de la Universidad Politécnica de Timisoara fue deductivo. En base al conocimiento general sobre el algoritmo genético clásico, se planteó un nuevo modelo inspirado en el mismo, el cual, sin embargo, tomó principios de la computación cuántica, a fin de lograr una mayor eficiencia y mejores resultados a los problemas que se buscaban solucionar con la aplicación de los algoritmos genéticos clásicos.

Hybrid Quantum-Classical Computing with Applications to Computer Graphics

Otra investigación que utilizó los conceptos de computación cuántica y computación clásica de forma conjunta fue la que realizaron Marco Lanzagorta y Jeffrey K. Uhmann en el Departamento de Ciencias de la Computación de la Universidad de Missouri-Columbia.

Esta investigación planteó que un algoritmo clásico podía ser “aumentado” para explotar los beneficios de la computación cuántica.

Particularmente se utilizó el algoritmo cuántico de Grover para incrementar la funcionalidad de un algoritmo de búsqueda lineal, y luego se describieron las consideraciones necesarias para aplicar el algoritmo híbrido resultante en geometría computacional y gráficos en computación. [6]

En esta investigación no se usó un lenguaje de programación en particular, el análisis de los algoritmos fue realizado a nivel abstracto sin llegar a su implementación explícita. El paper publicado con los resultados de la investigación, se llamó *Hybrid Quantum-Classical Computing with Applications to Computer Graphics* (Computación Híbrida Cuántica-Clásica con Aplicaciones para Gráficos de Computadora) [6]

La investigación de Lanzagorta y Uhmann es relevante en esta tesis pues se hace uso del algoritmo de Grover para potenciar un algoritmo de búsqueda clásico, es decir, se hace uso del algoritmo de Grover aplicado a un entorno clásico. Esto es precisamente, será un aspecto fundamental en la

implementación del algoritmo de Grover durante la fase de experimentación de esta tesis.

El método científico utilizado por Lanzagorta y Uhmman fue deductivo. A partir del conocimiento sobre computación cuántica y clásica, particularizaron algoritmos híbridos que contuvieran ciertas características particulares de ambas.

Designing and Implementing Small Quantum Circuits and Algorithms

Este estudio realizado por Ben Travaglione del Laboratorio de Computadoras del Centro para la Computación Cuántica en la Universidad de Cambridge en el Reino Unido provee conocimiento necesario para la formulación de sencillos algoritmos en pequeñas computadoras cuánticas que no necesiten de más que algunos pocos qubits. [4]

Travaglione publicó un paper donde exponía los resultados del estudio que realizó, el paper se llamó *Designing and Implementing Small Quantum Circuits and Algorithms* (Diseñando e Implementando Pequeños Circuitos y Algoritmos Cuánticos). [4]

En la investigación, se plantearon y definieron algunas estructuras de datos matemáticas que permitían guardar la información de un sistema cuántica y cada uno de sus elementos. Los algoritmos que se plantean son realizados bajo la transición de los estados de qubits a través de compuertas lógicas y operaciones simples basadas directamente sobre las leyes de la mecánica cuántica. [4]

El método científico usado por Travaglione fue deductivo. Concentró parte del conocimiento general de las leyes de la mecánica cuántica y tomó formulaciones matemáticas ya existentes para determinar que transiciones serían las más adecuadas para plantear un algoritmo sencillo.

En base a la información de las tres investigaciones presentadas, se elaboró el siguiente cuadro comparativo donde se resumen las características principales de cada uno de los estudios.

	<i>Implementing Quantum Genetic Algorithms: A Solution based on Grover Algorithm</i>	<i>Hybrid Quantum-Classical Computing with Applications to Computer Graphics</i>	<i>Designing and Implementing Small Quantum Circuits and Algorithms</i>
Método Científico	Deductivo	Deductivo	Deductivo
Herramientas de programación usadas	Pseudo código	Pseudo código	Pseudo código
Relación con la tesis	Hace uso del algoritmo cuántico de Grover para formular un algoritmo genético análogo a su contraparte clásica, pero con características eminentemente cuánticas. De manera similar, en esta tesis se usará el algoritmo de Grover, pero para potenciar un algoritmo genético clásico.	Plantea algoritmos que contienen características tanto de la computación clásica como de la cuántica. Precisamente utiliza el algoritmo de Grover para potenciar un algoritmo clásico. De manera similar, en esta tesis será necesario implementar el algoritmo de Grover en un entorno clásico.	Presenta principios básicos en la formulación de algoritmos cuánticos sencillos. Este conocimiento será útil, pues brindará la perspectiva necesaria en la formulación del algoritmo de Grover, así como su implementación.

¿El conocimiento generado es reutilizable?	Sí	Sí	Sí
¿Presenta conceptos de mecánica cuántica?	Sí	Sí	Sí
¿Hace uso del algoritmo de Grover?	Sí	Sí	No
¿Combina el entorno clásico y cuántico?	No	Sí	No

Tabla 1.1: Cuadro Comparativo de las Investigaciones Encontradas

1.4 Descripción y Sustentación de la Solución

La solución que se plantea para mejorar la eficiencia en la selección de individuos óptimos en la población inicial de un algoritmo genético, es utilizar el algoritmo cuántico de Grover y, posteriormente realizar un análisis comparativo respecto a un algoritmo clásico que se usa hoy en día para el mismo propósito: el algoritmo GRASP.

Para el desarrollo de esta investigación se llevarán a cabo los siguientes pasos:

- Se investigará y recopilará información sobre los conceptos físicos y matemáticos propios de la mecánica cuántica que influyen directamente en las características y el comportamiento de una computadora cuántica. En particular, se investigará sobre el

fundamento físico y matemático del algoritmo de Grover a fin de poder implementarlo en un entorno clásico.

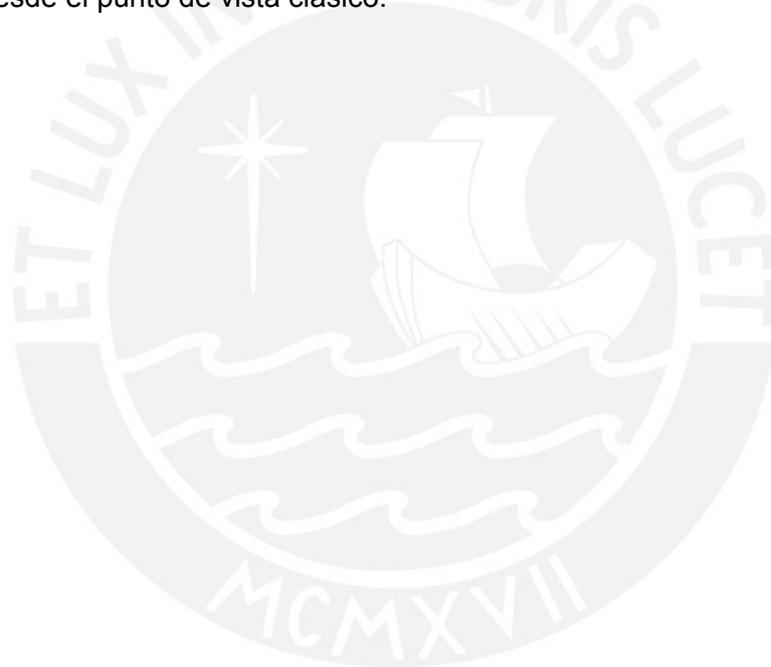
- Se planteará una hipótesis que relacione y compare las implementaciones del algoritmo de Grover y el algoritmo GRASP, ambas aplicadas a la selección de individuos óptimos en la población inicial del algoritmo genético.
- Se diseñará un experimento que permita la comparación objetiva entre ambas implementaciones de los algoritmos. El diseño de este experimento incluye la determinación de los factores que se usarán para realizar el análisis comparativo y la definición de la función de fitness del algoritmo genético genérico.
- Una vez se haya preparado la estructura del experimento a realizar, se procederá a la implementación de los algoritmos. Este proceso presupone la definición de las estructuras de datos necesarias y, en el caso del algoritmo de Grover, el modelo matemático del mismo. Antes de realizar la implementación se procederá al diseño en pseudocódigo de ambos algoritmos. Cuando el diseño de los dos algoritmos haya terminado, se procederá a la elección del lenguaje de programación, y, finalmente, se procederá a la implementación propiamente dicha.
- Tras culminar las implementaciones del algoritmo cuántico de Grover y del algoritmo GRASP, se procederá a la ejecución del experimento diseñado. A partir de los resultados obtenidos se realizará la experimentación numérica, el análisis comparativo entre ambas implementaciones y las conclusiones finales sobre las características de complejidad, eficacia y eficiencia de dichas implementaciones.

El desarrollo de esta investigación permitirá explicar de forma teórica y práctica, gracias al diseño e implementación de un algoritmo cuántico en un entorno clásico, cómo es el funcionamiento y cuáles son las características de un algoritmo cuántico (en este caso el algoritmo de Grover), resaltando las diferencias entre éste y un algoritmo clásico (en este caso, un algoritmo GRASP).

La comprensión de las leyes que rigen el funcionamiento de un algoritmo cuántico servirá de ayuda didáctica a toda persona interesada en una introducción al campo de la computación cuántica.

Finalmente, la aplicación del algoritmo cuántico de Grover a un algoritmo genético genérico, deja la puerta abierta a futuras optimizaciones del propio algoritmo genético, o, más allá, tratar de replantear su fundamento, como por ejemplo está haciendo la nueva rama de la Programación Cuántica Evolutiva.

Permitirá además ampliar la perspectiva del lector y sugerir el posible diseño de un modelo híbrido entre la programación clásica y la programación cuántica a fin de resolver problemas que, hoy en día, solo son enfocados desde el punto de vista clásico.





2 Capítulo II: Aporte de la Investigación

2.1 Aportes específicos

El presente proyecto de tesis presenta los siguientes aportes específicos:

- Recopila y explica de manera resumida y sucinta los conceptos físicos y matemáticos que rigen el funcionamiento de una computadora cuántica. Siendo la computación cuántica una rama de las Ciencias de la Computación que se encuentra en pleno desarrollo, esta tesis se presenta como un primer apoyo para toda persona que se encuentre interesada en dedicarse a esta área. Dado que los conceptos de la mecánica cuántica son altamente complejos y abstractos, resultan de difícil comprensión para un lector que está comenzando a adentrarse

en el tema. Es por ello que esta tesis trata de presentar dichos conceptos y términos de una forma clara, que permita al lector comprender a grandes rasgos las características más importantes de la computación cuántica.

- Define el modelo matemático de un algoritmo cuántico usando principios tanto matemáticos como físicos. El desarrollo de este modelo es vital para el futuro diseño e implementación del mencionado algoritmo, el cual en este caso es el algoritmo cuántico de Grover. Usando este modelo matemático y el diseño del algoritmo, el lector podría realizar la implementación del mismo usando un lenguaje diferente al que se utilizará en el desarrollo de la tesis o, incluso, usando un diferente paradigma de programación.
- Presenta el funcionamiento de un algoritmo cuántico dentro de un entorno clásico. La codificación del algoritmo dejará entrever claramente cuál es el funcionamiento del mismo, cuál es su utilidad y por qué resulta más natural su implementación en una computadora cuántica a diferencia de en una computadora clásica.
- Establece criterios de evaluación y comparación entre las implementaciones de dos algoritmos que usan paradigmas muy distintos: la programación cuántica y la programación clásica. Además, en este proyecto de tesis se han definido criterios que permitirán comparar de forma objetiva cuál es la implementación más eficaz y eficiente. Estos criterios pueden ser reutilizados posteriormente para cualquier otra comparación de implementaciones algorítmicas de índole similar a la presentada en esta tesis.
- Constituye una investigación pionera en el tema de computación cuántica dentro del ámbito universitario y del país en general. El nuevo campo de la computación cuántica, si bien cuenta con un gran número de investigaciones en proceso en las grandes potencias del mundo, carece de una línea seria de dedicación dentro de nuestro país: Perú. Las investigaciones que se realizan aquí sobre computación cuántica son escasas. Se espera que este proyecto de tesis abra las puertas a posteriores investigaciones que sigan la línea de indagación planteada

en información cuántica y generen nuevo conocimiento que será útil en la nueva era de información cuántica que se encuentra por llegar.

2.2 Desarrollo de la Metodología

Tal y como se mencionó líneas arriba, la metodología que se empleará para la investigación de este proyecto de tesis está basada en el método científico. A continuación se realizará una descripción de cada fase que forma parte de esta metodología, debidamente aplicada al presente proyecto en particular:

2.2.1 Observación

Esta es la primera fase del proyecto de investigación y consiste básicamente en revisar el conocimiento existente hasta la fecha sobre la computación cuántica que sea aplicable a la implementación del algoritmo de Grover en un entorno clásico, a fin de ser aplicado a la selección óptima de individuos. Además involucra investigar sobre los fundamentos del algoritmo genético, a fin de formular la estructura de los individuos de la población y definir la función de fitness. Finalmente, incluye la investigación sobre el fundamento del algoritmo GRASP a fin de realizar una implementación correcta del mismo.

La fase de Observación consta del siguiente proceso:

- *Recopilación de Información:*

En este proceso se realizará la investigación previa del tema para poder realizar un planteamiento de hipótesis adecuado y una correcta experimentación. Particularmente los temas que se investigarán son los principios de la mecánica cuántica, su aplicación en la computación cuántica, el algoritmo de Grover, los algoritmos genéticos y el algoritmo GRASP.

2.2.2 Planteamiento de Hipótesis:

Segunda fase del proyecto de tesis. Una vez que se ha recopilado conocimiento sobre computación cuántica puede plantearse una hipótesis la cual será contrastada después de realizar la experimentación y a partir de la verdad o falsedad de esta hipótesis generar nuevo conocimiento debidamente sustentando. Consta del siguiente proceso:

- *Formulación de la hipótesis:*

En este proceso, se propondrá una hipótesis o supuesto que responderá de forma parcial la interrogante propuesta. Será después en la fase de conclusión cuando la hipótesis, si es aceptada, responderá la interrogante de forma total.

2.2.3 Experimentación

Tercera fase del proyecto de tesis. Constituye a la vez la etapa de mayor duración y complejidad durante el desarrollo del proyecto. Es en esta etapa donde se aplicará todo el conocimiento recogido en las dos etapas anteriores para diseñar e implementar el algoritmo cuántico de Grover y el algoritmo GRASP. Consta de los siguientes procesos:

- *Diseño del proceso Experimental*

En este proceso se determinarán las características del experimento a realizar que permitirá obtener los datos necesarios para llevar a cabo la experimentación numérica. Este experimento está relacionado con el proceso de selección de la población inicial del algoritmo genético. Además, se definirá el método estadístico adecuado que permita contrastar los resultados obtenidos del experimento.

También se definirá la función de fitness del algoritmo genético genérico al cual se aplicarán el algoritmo de Grover y el algoritmo GRASP. Por último, en este proceso también se determina qué características de los

algoritmos (ya implementados) pueden ser usadas como factores de comparación.

- *Diseño e Implementación de los Algoritmos*

En este proceso se definen las estructuras de datos que se usarán en el diseño e implementación de los algoritmos. En el caso del algoritmo cuántico de Grover se cuenta con un paso adicional: es necesario analizar su modelo físico-matemático.

Luego se realizará el diseño de cada uno de los algoritmos usando pseudocódigo, en el caso del algoritmo cuántico de Grover, este diseño debe considerar su inclusión en un entorno clásico. Por último, ambos diseños serán implementados (codificados) utilizando un mismo lenguaje de programación.

- *Ejecución de los Experimentos:*

En este proceso, una vez que ya se tienen los algoritmos implementados, se siguen los pasos definidos en el experimento principal. Tras la ejecución de este, se recopilarán y almacenarán los resultados exactos, tal cual fueron obtenidos del experimento. Con estos datos se ejecuta la experimentación numérica.

2.2.4 Conclusión

Última fase del proyecto de tesis. Es la etapa de mayor análisis deductivo en base a los resultados obtenidos de la experimentación. Los resultados son empleados como datos de entrada del experimento numérico que permitirá contrastar los datos reales con los datos supuestos de la hipótesis inicial. Además se realizará el análisis comparativo de las implementaciones y se determinarán las conclusiones finales del proyecto de tesis.

Consta de los siguientes procesos:

- *Análisis de los resultados obtenidos de la experimentación numérica:*

Este proceso involucra un análisis completo y exhaustivo de los resultados que devuelva la experimentación numérica

- *Análisis comparativo de las Implementaciones:*

Este proceso consta de un análisis basado en las experiencias que hayan sucedido durante el desarrollo de la investigación, y los resultados numéricos obtenidos tanto del experimento principal como de la experimentación numérica. El análisis comparativo tiene como objetivo contrastar las implementaciones de los dos algoritmos.

- *Redacción de las conclusiones de la investigación:*

Esto incluye los resultados del análisis comparativo y la aceptación o rechazo de la hipótesis.

2.3 Lista de Resultados Esperados

Durante el progresivo avance del proyecto de tesis se obtendrán una serie de resultados que en conjunto permitirán cumplir los objetivos propuestos inicialmente. Cuando el proyecto de tesis haya finalizado, se espera haber obtenido los siguientes resultados:

- A. Definición de la población inicial del algoritmo genético.
- B. Definición de la función de fitness del algoritmo genético.
- C. Diseño en pseudocódigo de un algoritmo GRASP aplicado a la selección de individuos óptimos en la población del algoritmo genético.
- D. Implementación del algoritmo GRASP.
- E. Modelo Matemático del algoritmo cuántico de Grover aplicado a la selección de individuos óptimos en la población del algoritmo genético.
- F. Diseño en pseudocódigo del algoritmo cuántico de Grover en un entorno clásico.
- G. Implementación del algoritmo cuántico de Grover en un entorno clásico.
- H. Definición de factores de comparación entre las implementaciones de los dos algoritmos.
- I. Resultados de la Experimentación Numérica.
- J. Conclusiones de la idoneidad del uso del algoritmo cuántico de Grover en la selección de individuos óptimos de la población de un algoritmo genético genérico.

2.4 Planteamiento de hipótesis.

La computación cuántica presenta una serie de novedosas características las cuales la hacen una opción sumamente atractiva en lo que respecta a procesamiento de grandes cantidades de información y búsqueda de soluciones a problemas altamente complejos.

Dada la notable eficiencia lograda por el algoritmo de Grover en búsquedas dentro de un universo con una gran cantidad de elementos, surge la pregunta ¿Puede aplicarse el algoritmo cuántico de Grover para realizar búsquedas que a través de la computación clásica toman un tiempo considerablemente alto, el cual puede incluso considerarse absurdo? ¿Puede el algoritmo cuántico de Grover usarse en problemas clásicos?

Como respuesta para estas dos preguntas, se propone en esta tesis, aplicar el algoritmo cuántico de Grover para realizar la selección de individuos óptimos en la población inicial de un algoritmo genético. Este procedimiento, al formar parte de un algoritmo genético, tiene de forma inherente un contexto clásico al cual debe adaptarse el algoritmo de Grover. Surge entonces una nueva duda, ¿el algoritmo cuántico de Grover puede aplicarse de forma efectiva en un entorno clásico como el de un algoritmo genético?

Por ello plantearemos la siguiente hipótesis:

Hipótesis 1: Es posible la implementación del algoritmo cuántico de Grover en un entorno clásico, de tal forma que pueda aplicarse de forma efectiva en un algoritmo clásico, como es un Algoritmo Genético.

Asumimos esta hipótesis por verdadera, dado que existen investigaciones realizadas anteriormente donde, en efecto, se pudo lograr el diseño de modelos de programación híbridos, los cuales usaban lo mejor tanto del paradigma de programación clásica, como del paradigma de programación cuántica. (Sección 1.4 - Estado del Arte).

Dado que la hipótesis 1 es asumida como verdadera, surge la siguiente pregunta: Si en realidad el algoritmo cuántico de Grover puede aplicarse en un entorno clásico (en este caso, un algoritmo genético), **¿Es el algoritmo cuántico de Grover más eficiente que los algoritmos clásicos de búsqueda usados por un algoritmo genético?** A raíz de esta pregunta, es necesario realizar una comparación, y antes que eso, es necesario elegir el algoritmo clásico contra el cual se realizaría la comparación.

Se eligió como candidato para la comparación a un algoritmo GRASP. Esto debido a que el algoritmo GRASP ha sido durante mucho tiempo (hasta la fecha) uno de los que han dado resultados más exitosos en lo que respecta a su aplicación en la selección de población de un algoritmo genético.

Por tanto, para poder realizar esta comparación, es necesario definir las características con las cuales serán comparados los dos algoritmos. Los factores de comparación generales que se proponen son los siguientes:

La eficacia del algoritmo cuántico de Grover y del algoritmo de GRASP en la selección de individuos óptimos de la población del algoritmo genético. La función de fitness que se definirá para el algoritmo genético genérico nos permitirá cuantificar qué individuos seleccionados por los algoritmos son efectivamente los mejores. Por lo tanto, el algoritmo que seleccione los mejores individuos tendrá el mayor grado de eficacia.

La eficiencia de los algoritmos también jugará un papel importante en la comparación. Si la eficacia de ambos algoritmos puede llegar a considerarse equiparable, será la eficiencia el factor decisivo en la comparación de ambos. Para la comparación de la eficiencia de ambos algoritmos, se considerará la cantidad de operaciones (clásicas para el algoritmo GRASP y cuánticas para el algoritmo de Grover) necesarias para terminar la ejecución efectiva de cada uno de ellos. A fin de, evitar detalles innecesarios en cuanto el tiempo de ejecución de ambos algoritmos, se plantea la siguiente hipótesis:

Hipótesis 2: El tiempo de ejecución de una operación cuántica del algoritmo de Grover es igual al tiempo de ejecución de una operación clásica del algoritmo GRASP.

Esta hipótesis es necesaria aceptarla como verdadera, debido a la carencia de una computadora cuántica sobre la cual ejecutar el algoritmo de Grover.

La complejidad de la implementación de los algoritmos, es decir, cuán difícil resulta ser la realización del proceso. Cada uno de los algoritmos tiene tras de sí una serie de fundamentos inherentes los cuales pueden hacer más o menos difícil su diseño e implementación. De forma aparente, un algoritmo GRASP es menos complejo de integrar en un algoritmo genético, dada la cantidad de veces que esta integración ha resultado exitosa en numerosos estudios y casos de aplicación. De forma contraria, la integración del algoritmo de Grover y un algoritmo de Grover es compleja dado el escaso estudio al respecto y la nulidad de casos prácticos del tema. Sin embargo, se cuantificará la complejidad tanto del algoritmo GRASP como del algoritmo cuántico de Grover, a fin de poder considerar este factor dentro del proceso de comparación. Esta cuantificación estará basada fundamentalmente en la experiencia de la implementación que se seguirá en esta tesis

Entonces cada uno de los tres criterios mencionados tendrán asociados un factor por cada uno de los algoritmos, estos factores permitirán hacer la comparación entre los dos algoritmos a utilizar. Es decir, se definen:

R_{eficacia} : Factor de eficacia de la aplicación del algoritmo GRASP.

$R_{\text{eficiencia}}$: Factor de eficiencia de la aplicación del algoritmo GRASP.

$R_{\text{complejidad}}$: Factor de complejidad de la implementación del algoritmo GRASP.

G_{eficacia} : Factor de eficacia de la aplicación del algoritmo cuántico de Grover.

$G_{\text{eficiencia}}$: Factor de eficiencia de la aplicación del algoritmo cuántico de Grover.

$G_{\text{complejidad}}$: Factor de complejidad de la implementación del algoritmo cuántico de Grover.

Y se definen además dos factores que representan en conjunto el grado de bondad de cada una de las implementaciones de los algoritmos:

R_B : Factor de bondad de la implementación del algoritmo GRASP.

$$R_B = R_{\text{eficacia}} + R_{\text{eficiencia}} - R_{\text{complejidad}}$$

G_B : Factor de bondad de la implementación del algoritmo cuántico de Grover.

$$G_B = G_{\text{eficacia}} + G_{\text{eficiencia}} - G_{\text{complejidad}}$$

Por tanto, estos serán los factores que indicarán si un algoritmo (y por extensión, su implementación) es mejor que otra. Lo cual lleva al planteamiento de la siguiente hipótesis:

Hipótesis 3: El grado de bondad de la aplicación del algoritmo cuántico de Grover es igual al grado de bondad de la aplicación del algoritmo GRASP, ambos aplicados a la selección de individuos óptimos de la población inicial de un algoritmo genético.

$$R_B = G_B$$

Del planteamiento de esta hipótesis podría surgir la duda: **¿Por qué asumir que ambos algoritmos tendrán un igual grado de bondad?** Esto tiene su razón en que, a este punto, aún no se han llevado a cabo las implementaciones de los algoritmos, y tampoco se han aplicado ambos a la selección de individuos de la población de un algoritmo genético. No se puede asegurar, por tanto, precipitadamente que un algoritmo será necesariamente mejor que el otro. Por ello resulta más sensato determinar que inicialmente ambas implementaciones tienen igual grado de bondad, y, tras realizar la experimentación numérica, si esta hipótesis llega a ser descartada, se sabrá

en efecto que una es mejor que otra, lo cual llevará a la formulación de una nueva hipótesis. Sin embargo es necesario recordar que podrá plantearse esa nueva hipótesis si y solo si en el caso la *Hipótesis 3* es rechazada.

Las tres hipótesis planteadas en esta sección serán asumidas por ciertas hasta la fase de Conclusión. En esta fase, los resultados de la experimentación numérica permitirán corroborar la verdad de estas hipótesis o la falsedad de las mismas.



3 Capítulo III: Implementación del Aporte

3.1 Definición de los Experimentos

En este apartado, se definirá cada uno de los experimentos que se realizarán en la fase de experimentación de la tesis. Esto incluye la experimentación numérica. Cada uno de los experimentos presupone una serie de requisitos como las implementaciones del algoritmo de Grover y del algoritmo GRASP. Estas serán detalladas en el siguiente apartado.

A fin de poder realizar el análisis comparativo entre el algoritmo GRASP y el algoritmo cuántico de Grover, primero se elaborará un experimento cuyo objetivo será aplicar ambos algoritmos a la selección de individuos óptimos en la población de un algoritmo genético. El resultado final del experimento será el grado de eficacia y eficiencia con la cual cada algoritmo realizó su

selección. La eficacia dada por el valor de la función de fitness de los individuos seleccionados, y, la eficiencia hallada a partir de la cantidad de operaciones que se necesitaron.

Posteriormente, se realizará la experimentación numérica. Se usará una prueba estadística T-Student para contrastar los resultados numéricos de las ejecuciones del experimento, primero usando el algoritmo de Grover y luego el algoritmo GRASP. El objetivo de la experimentación numérica es determinar qué algoritmo es el mejor de ambos. La experimentación numérica cuenta con dos fases:

- En la primera fase, se plantea como hipótesis que el valor de bondad de los individuos seleccionados por ambos algoritmos es el mismo. Es decir que ningún algoritmo es mejor. Si se logra rechazar esta hipótesis, se procede con la fase 2. Si no se logra rechazar la hipótesis, se debe concluir que ningún algoritmo es mejor que el otro.
- Se llega a la segunda fase de la experimentación numérica, si es que en la fase 1 se garantizó que efectivamente un algoritmo es mejor que el otro. En este caso, se compararán las medias de los grados de bondad obtenidos por cada algoritmo a fin de determinar cuál es el mejor.

A continuación se detallará la estructura de cada uno de estos experimentos:

3.1.1 Selección de Individuos óptimos (Experimento-1)

Este será el experimento principal de la investigación. Este experimento será aplicado a la fase de selección de la población inicial del algoritmo genético.

ENTRADAS

Las entradas que se necesitan para la ejecución de este experimento son:

N° de Entrada	Descripción
1	Algoritmos GRASP y Grover implementados: Evidentemente se necesita que los dos algoritmos estén implementados antes de ejecutarlos.
2	Tamaño del Universo (N) : Esta cantidad permitirá determinar cuán grande es el universo de posibles soluciones
3	Cantidad de Iteraciones (X): Este valor determinará cuantas veces se ejecutarán los algoritmos GRASP y Grover para seleccionar los individuos óptimos de la población inicial. Este valor debe ser grande para poder obtener resultados objetivos y consistentes.

PASOS DEL EXPERIMENTO

Los pasos que se siguen en el experimento son los siguientes:

Repetir X veces

- a. Formar el universo de posibles soluciones a partir del parámetro N.
- b. Aplicar el algoritmo GRASP y el algoritmo Grover al universo de individuos generado en la parte a) para seleccionar los elementos óptimos.
- c. Guardar el valor de fitness de los individuos seleccionados por cada algoritmo. (Este es el resultado en eficacia de una ejecución de uno de los dos algoritmos)

- d. Guardar la cantidad de operaciones clásicas o cuánticas que usaron el algoritmo GRASP y Grover respectivamente en la parte a). (Con este resultado se hallará el grado de eficiencia en una ejecución de uno de los dos algoritmos).

Fin de Repetir

Resultado Final: Valores de eficacia y eficiencia de ambos algoritmos en cada iteración del experimento

3.1.2 Experimentación Numérica

Este experimento jugará un papel muy importante al determinar qué algoritmo es más eficaz que el otro.

El nombre del experimento que se utilizará es: Prueba T-Student para Dos Muestras relacionadas. Este experimento es muy usado para contrastar resultados de diferentes procesos (por ejemplo un algoritmo) y determinar qué proceso es mejor, **siempre y cuando se trate de sólo dos procesos**. Si se tratara de más de dos, la alternativa es usar un experimento como el Análisis de Varianza (ANOVA). Como se van a comparar **dos** algoritmos, la prueba T-Student resulta apropiada [19]

ENTRADAS

Los datos necesarios para la ejecución de la experimentación numérica serán obtenidos de la ejecución del primer experimento.

PASOS DEL EXPERIMENTO

Los pasos del experimento son los siguientes:

1. Se plantea la hipótesis inicial de la experimentación numérica: “El valor de bondad promedio obtenido de la selección del algoritmo cuántico de Grover y el valor de bondad promedio obtenido de la selección del algoritmo GRASP, son iguales”

2. Se realiza la prueba de Contraste de Igualdad de Medias para determinar si la hipótesis del paso 1 es cierta.
3. Si la prueba de Contraste de Igualdad de Medias da valor positivo:
 - a. Se concluye que los grados de bondad del algoritmo cuántico de Grover y el algoritmo GRASP son equivalentes.
4. Si la prueba de Contraste de Igualdad de Medias da valor negativo:
 - a. Comparar los valores promedios de grado de bondad obtenido de cada algoritmo a fin de determinar cuál es el mejor.

3.2 Definición de los Algoritmos

En este apartado se definirán los algoritmos que serán utilizados en el proceso experimental. Se comienza con la definición del algoritmo genético genérico, enfocándose en la descripción del proceso de selección de su población inicial y la formulación de la función de fitness a utilizar.

Luego se procederá a la definición de las estructuras de datos y consideraciones adicionales necesarias en el diseño del algoritmo GRASP.

Por último se formulará el modelo matemático del algoritmo cuántico de Grover y se detallarán las consideraciones necesarias para implementar este algoritmo en un entorno clásico.

3.2.1 Definición del Algoritmo Genético

En este apartado se definirá las principales características del Algoritmo Genético, las cuales serán de suma importancia para la ejecución del experimento.

En lo sucesivo el algoritmo genético genérico será identificado con las siglas **AGG**.

Selección de la población inicial

Todo algoritmo genético cuenta con una población, es decir un conjunto de individuos, siendo cada uno de estos una solución potencial al problema que se busca resolver.

Todo problema que busca resolver un algoritmo genético tiene un universo general de soluciones, las cuales son generalmente combinaciones de una serie de elementos dados. La evaluación de cada una de las posibles combinaciones es una tarea excesiva que ocuparía una cantidad de tiempo irracional en la mayoría de casos. Es por eso que los algoritmos genéticos cuentan con una población inicial, es decir, un subconjunto del universo de soluciones, de preferencia, un subconjunto que incluya las mejores soluciones candidatas. De ahí la importancia en la fase de selección de la población inicial de un algoritmo genético.

Esta fase de selección resulta ser un excelente caso en el cual aplicar el algoritmo cuántico de Grover y el algoritmo GRASP, es por ello que el experimento principal fue diseñado con el objetivo de seleccionar la población inicial a partir del universo total de soluciones candidatas.

Ahora es necesario definir cada individuo para poder definir, eventualmente, el universo total de soluciones candidatas.

Cada individuo del universo contiene lo que se conoce como un cromosoma, y un cromosoma viene a ser un conjunto de genes. A su vez, cada gen contiene parte de la información del individuo, información vital para determinar si el individuo es en realidad una posible solución al problema y si es así, para determinar cuán buena es su solución. [13]

Para facilitar la formulación del algoritmo genético, se tratará de hacer de esta tarea lo más genérica posible.

Se define que un individuo cuenta con un cromosoma que incluye “N” genes y tiene la siguiente estructura:

Gen1	Gen2	Gen3	Gen4	Gen5	Gen6,	Gen N
------	------	------	------	------	------	-----	-----	-----	-------

Ahora toca decidir la estructura que tendrá cada gen. La estructura inicial propuesta por John Holland (conocido como el padre de los algoritmos genéticos) estaba dada por cadenas de bits [14]. Se usará una representación análoga, cada gen será un bit, el cual podrá tener el valor cero o uno. El valor cero indica que el gen se encuentra “apagado”, es decir, no proporciona información al cromosoma, por el contrario, si el valor es uno, el gen se encuentra “prendido” y sí contribuye a la información total del cromosoma.

Cada gen también tendrá cierta información adicional, inherente a cada uno de ellos. Se define lo siguiente:

Definición 1:

“Cada uno de los genes de un cromosoma tiene asociado una triada de tres valores: un factor positivo, un factor negativo y un factor de ponderación”.

Estos valores serán hallados de forma aleatoria pues no hay forma de asignarles valores exactos sin el contexto de una aplicación real. Además la naturaleza aleatoria de estos valores permitirá obtener resultados más objetivos en el experimento. Cabe resaltar que tanto el factor positivo y el factor negativo serán reales dentro de un rango específico, mientras que el factor de ponderación será siempre un número real entre cero y uno.

También se define lo siguiente:

Definición 2:

“Existe una función matemática F , la cual dado el valor de los genes de un cromosoma, así como los factores positivos, negativos y de ponderación de cada gen, puede calcular el grado de bondad del cromosoma. Esta función será llamada: función de fitness”

Además, el grado de bondad de un cromosoma es un valor numérico exacto que determina cuán bueno es el individuo, es decir cuán buena es su solución.

Ahora que ya se tienen definida la estructura de los cromosomas y genes, es necesario resolver una pregunta de suma importancia que ha quedado pendiente desde el principio: **¿Cuál es el problema que se está tratando de resolver?**

Como se ha venido mencionando, no existe un problema real al cual se esté aplicando el algoritmo genético, de ahí que se le agregue el adjetivo genérico. Por tanto plantearemos el problema como: **¿Cuál es el cromosoma que tiene el mayor grado de bondad?**

Sin embargo se colocará una restricción a todos los candidatos para ser una solución real. Si no existiera esta restricción, sería fácil hallar la mejor solución, sería simplemente la suma de todos los genes con mayor factor positivo. La restricción que se colocara es: la suma de los factores de ponderación de los genes que tengan valor uno, no puede exceder el valor promedio de factores de ponderación de los N genes. El promedio de factores de ponderación se halla fácilmente, dado que el menor valor posible es que todos los genes tengan factor de ponderación igual a cero, y el mayor valor, que todos los factores de ponderación sean igual a uno, el valor promedio de factores de ponderación es el siguiente:

$$\text{Valor promedio de factores de ponderación} = (1 \times N + 0 \times N) / 2 = N/2$$

En resumen, la restricción a toda verdadera solución del “problema” en cuestión, es que la suma de todos los factores de ponderación de los genes “prendidos” (con valor en uno), no exceda en valor a la mitad de la cantidad total de genes.

Como puede verse, el valor de N determina la cantidad de posibles soluciones y por tanto el tamaño del universo a considerar. Precisamente este valor N es el parámetro que se definió en el experimento principal para delimitar el tamaño del universo sobre el cual trabajarán tanto el algoritmo GRASP como el algoritmo de Grover.

Definición de la función de Fitness

En el apartado anterior se definió la estructura de un individuo del algoritmo genético genérico. También se mencionó a la función de fitness en la Definición 2, se dijo que era una función matemática que permitirá calcular el **grado de bondad** de un individuo, basándose en la información grabada en su cromosoma y por tanto, sus genes.

Se había dicho que cada gen tenía asociada una triada de valores dada por: {Factor positivo, factor negativo, factor de ponderación}, además, cada gen debía tomar un valor binario: cero o uno.

Entonces se definirá la función de fitness de la siguiente manera:

Sea **F** una función real que recibe como entrada los valores de cada uno de los “N” genes en el cromosoma de un individuo, y, devuelve un único valor real, el cual se llamará **grado de bondad** de dicho individuo.

$$F(\text{Cromosoma}) = \sum_{i=1}^N ((\text{FactorPositivo}_i + \text{FactorNegativo}_i) * \text{FactorPonderación}_i) g_i$$

Vale acotar que el factor positivo, factor negativo y factor de ponderación de cada gen, son constantes en cada iteración del experimento. Son generados de forma aleatoria al momento de generar el universo de posibles soluciones (ver paso “a” del Experimento-1).

Puede deducirse de la fórmula que:

- Los genes que tengan el valor cero no contribuyen al valor final del grado de bondad
- Si el valor absoluto del factor negativo es mayor al valor absoluto del factor positivo, el gen contribuye negativamente al grado de bondad total.

Queda entonces definida la función de fitness del algoritmo genético genérico que se utilizará en el experimento principal.

3.2.2 Definición del Algoritmo GRASP

En este apartado se definirán las principales características del Algoritmo GRASP que se utilizará para calcular la población inicial del algoritmo genético. Tomando en cuenta las definiciones del algoritmo genético, toca establecer las estructuras de datos a utilizar en el diseño y posterior implementación del algoritmo GRASP.

En lo sucesivo el Algoritmo GRASP será identificado con las siglas **AGR**.

El algoritmo GRASP consta de una serie continua de iteraciones, en las cuales trata de seleccionar una buena solución al problema. Para no dejarse llevar por lo que se conoce “criterio goloso” y elegir equívocamente un máximo local, el algoritmo GRASP crea una lista restringida de candidatos usando un factor aleatorio. Es precisamente esta aleatoriedad la que hace del algoritmo GRASP una alternativa con superior eficacia respecto a una gran cantidad de algoritmos heurísticos.

La cantidad de iteraciones “n” que se realizarán con el algoritmo GRASP será determinada antes de la ejecución del experimento principal. Se realizará un proceso de calibración del algoritmo GRASP, esto supondrá la ejecución del algoritmo GRASP varias veces, cada una con un diferente valor de “n” hasta determinar en promedio, qué valor halla buenos resultados y en un tiempo aceptable.

El algoritmo GRASP también hace uso de un parámetro importante en la creación de la lista restringida de candidatos. Este parámetro está simbolizado por la letra griega α y se le conoce como “**factor de relajación**”. El valor de este parámetro también será hallado con un proceso de calibración previo al experimento principal de la investigación, pero se tendrá en cuenta valores previos utilizados en aplicaciones similares a la presente.

Finalmente, es necesario determinar las estructuras de datos que se necesitarán para la implementación del algoritmo GRASP, estas estructuras de datos podrán también ser utilizadas por el Algoritmo Cuántico de Grover.

Definición de las Estructuras de Datos

Para el correcto diseño e implementación del algoritmo GRASP, es necesario contar con las siguientes estructuras de datos:

- a) Una estructura de Datos para representar los Genes. Esta es la estructura de datos primaria y de más bajo nivel. La entidad u objeto Gen deberá contar con los siguientes atributos: Número de Gen (número entero), Valor (binario) y Grado de Bondad del Gen (un número real).
- b) Una estructura de Datos para representar los Cromosomas: Esta entidad representará cada individuo y simboliza una posible solución. La entidad u objeto Cromosoma deberá contar con los siguientes atributos: Una lista de Genes y el Grado de Bondad del Individuo (un número real).
- c) Una estructura de Datos para los datos de la triada de cada Gen: Esta entidad tendrá los valores de los factores de la triada para cada Gen. Esta estructura será constante por cada ejecución completa del algoritmo GRASP y será generada cada vez que se forme un nuevo universo de soluciones. Esta estructura de datos, será una lista de "N" (número de genes) objetos Triada. Cada objeto Triada tendrá los atributos: Factor positivo, Factor Negativo y Factor de Ponderación, siendo todos, valores reales.

Estos son los factores necesarios para la futura formulación del diseño en pseudocódigo del algoritmo GRASP.

3.2.3 Definición del Algoritmo Cuántico de Grover

En este apartado se definirán las principales características del Algoritmo Cuántico de Grover, el segundo algoritmo candidato a utilizar para la selección de la población inicial del algoritmo genético. A partir de las definiciones del algoritmo genético establecidas en el apartado anterior, se definirá el modelo matemático del algoritmo cuántico de Grover y las

estructuras de datos necesarias para su diseño e implementación en un entorno clásico.

En lo sucesivo el Algoritmo Cuántico de Grover será identificado con las siglas **ACG**.

El ACG tiene un fundamento completamente diferente al del Algoritmo GRASP. En esta sección se comprenderá los aspectos más importantes que se usarán en su diseño e implementación.

Tanto en el AGR como en el ACG se parte de un universo total de posibles soluciones. El objetivo es encontrar la mejor solución, es decir realizar una búsqueda en todo ese universo hasta encontrar un único resultado. Para llevar a cabo esto, el algoritmo de Grover usa lo que se conoce como **el oráculo**.

Por el momento, lo más adecuado es considerar al oráculo como una caja negra, que de una forma u otra, es capaz de *identificar* la solución óptima al problema [15].

Se definió antes que se tendrá una cantidad de N genes en el cromosoma de cada individuo, y que cada uno de esos genes podía tomar el valor de cero o uno, y, por ello, el universo total de posibles soluciones era de 2^N elementos. Utilizando algunos conceptos claves de computación cuántica se puede definir lo siguiente:

Definición 4:

Si se tiene un universo de 2^N elementos, es posible formular un sistema cuántico que lo represente. Este sistema cuenta con una superposición de N estados, cada uno de los cuales es un qubit.

Además es necesario definir formalmente como se considerará un qubit. Se tomará la definición más general, encontrada en el libro de Nielsen y Chuang [15]

Definición 5:

Un qubit es un estado cuántico que puede tomar el valor de cero o uno. Usando la notación de Dirac, un qubit puede estar en el estado $|0\rangle$ ó en el estado $|1\rangle$.

Entonces, si por ejemplo, N tuviera el valor de 2, se tendrían 4 elementos, representados de la siguiente forma:

$$|n_0\rangle = |0\rangle|0\rangle$$

$$|n_1\rangle = |0\rangle|1\rangle$$

$$|n_2\rangle = |1\rangle|0\rangle$$

$$|n_3\rangle = |1\rangle|1\rangle$$

Una representación bastante similar a la que se haría con conceptos de computación clásica, la única diferencia está en la concepción de qubits como estados, en vez de bits convencionales.

Usando las definiciones anteriores, ahora es posible describir mejor la naturaleza del oráculo:

Definición 6:

Cada estado $|n_i\rangle$ tiene asociado un oráculo O_i tal que:

$$O_i |n_i\rangle = - |n_i\rangle$$

$$O_i |n_j\rangle = +|n_j\rangle, \text{ donde } i \neq j$$

Es decir, si se aplica el oráculo a su estado correspondiente, el oráculo provoca un “**cambio de fase**” en dicho estado. Podemos considerar que este cambio de fase se manifiesta como un cambio de signo.

Por otra parte, si se aplica el oráculo a algún otro estado, el oráculo no causa ningún cambio.

Con todos los conceptos mencionados anteriormente podemos establecer la siguiente afirmación:

Si se quiere buscar el estado $|n_i\rangle$ y se conoce su función oráculo O_i , es posible encontrarlo aplicando dicho oráculo en cada estado del universo total, dado que solamente el estado $|n_i\rangle$ presentará un cambio de fase.

En consecuencia, en el problema de búsqueda de individuos del algoritmo genético, si se conoce la función oráculo del mejor individuo, es posible determinar cuál es el mejor individuo aplicando esa función oráculo a cada elemento del Universo.

Entonces, hasta ahora, surgen dos inconvenientes al seguir el camino cuántico en la búsqueda de individuos, el primero, que es necesario calcular o definir la función oráculo del mejor individuo antes de proceder a buscarlo. Y el segundo inconveniente radica en que no se está logrando ninguna ventaja en comparación con el camino clásico: aún si se tiene definida la función oráculo del mejor individuo, es necesario aplicar dicha función oráculo a **todos** los elementos del universo hasta ver cuál de ellos presenta un “cambio de fase”.

Es aquí donde aparece el Algoritmo Cuántico de Grover para evitar aplicar la función oráculo a cada uno de los elementos del Universo, y, en cambio, realizar una búsqueda sumamente más eficiente. Grover plantea una serie de pasos, a través de los cuales, el número de iteraciones necesarias para hallar la respuesta, es del orden de \sqrt{N} .

Usando los conceptos descritos podemos enunciar el modelo matemático del algoritmo Grover en el cual se describen los elementos que usa y la relación entre estos:

Modelo Matemático del Algoritmo de Grover:

Sea “M” la cantidad de elementos del universo, tal que:

$$M = 2^x, x \in \mathbb{N}$$

Sea “N” la cantidad de qubits necesarios para representar el universo, donde:

$$N = \log_2(M)$$

Sea $|\varphi\rangle$ el estado inicial del universo:

$$|\varphi\rangle = \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |n_i\rangle$$

El sistema está en una superposición con máxima probabilidad: puede colapsar con igual probabilidad a cualquier estado.

Sea G, el operador de Grover:

$$G = (2|\varphi\rangle\langle\varphi| - I)O_k$$

Donde I es la matriz identidad y O_k la función oráculo de elemento buscado.

Cada aplicación **de la función oráculo** crea un desfase en el estado $|n_k\rangle$. Por tanto el estado del universo quedaría como:

$$O_k |\varphi\rangle = |\varphi_1\rangle = \frac{\sqrt{2^N - 1}}{\sqrt{2^N}} \sum_{\substack{i=0 \\ i \neq k}}^{2^N-1} |n_i\rangle - \frac{1}{\sqrt{2^N}} |n_k\rangle$$

El nuevo estado del universo está dado entonces por:

$$|\varphi'\rangle = G |\varphi\rangle = (2|\varphi\rangle\langle\varphi| - 1)O_k |\varphi\rangle$$

Reemplazando $|\varphi_1\rangle$:

$$|\varphi'\rangle = (2|\varphi\rangle\langle\varphi| - 1) |\varphi_1\rangle$$

$$|\varphi'\rangle = 2|\varphi\rangle - |\varphi_1\rangle$$

Dado que el producto interno de $\langle\varphi|$ y $|\varphi_1\rangle$ es un escalar, sea:

$$C = \langle\varphi|\varphi_1\rangle$$

Entonces:

$$|\varphi'\rangle = 2C|\varphi\rangle - |\varphi_1\rangle$$

$$|\varphi'\rangle = \frac{2C}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |n_i\rangle - \left(\frac{\sqrt{2^N-1}}{\sqrt{2^N}} \sum_{\substack{i=0 \\ i \neq k}}^{2^N-1} |n_i\rangle - \frac{1}{\sqrt{2^N}} |n_k\rangle \right)$$

$$|\varphi'\rangle = \frac{2C - \sqrt{2^N-1}}{\sqrt{2^N}} \sum_{\substack{i=0 \\ i \neq k}}^{2^N-1} |n_i\rangle + \left(\frac{2C+1}{\sqrt{2^N}} |n_k\rangle \right)$$

Como se puede apreciar, la amplitud de todos los estados distintos al buscado ha disminuido. Por el contrario la amplitud del elemento buscado ha aumentado.

Conclusión: Sucesivas aplicaciones del operador de Grover hacen que la amplitud del estado buscado $|n_k\rangle$ siga creciendo y que, por tanto, sea el valor más probable de obtener al hacer una medición al sistema.

En cuanto a las estructuras de datos necesarias para la implementación del Algoritmo Cuántico de Grover, una estructura indispensable es la que represente un estado cuántico (qubit). Una alternativa inicial será usar valores binarios (cero o uno), dado que es imposible representar superposición real en una computadora clásica, se tratará de simular este comportamiento durante la implementación del algoritmo.

Queda pendiente también, la formulación de la función oráculo del mejor individuo en el universo completo de soluciones candidatas.

3.3 Diseño de los Algoritmos

En este apartado, se procederá a realizar la formulación en pseudocódigo del Algoritmo GRASP y el Algoritmo Cuántico de Grover. En el caso del Algoritmo Genético, no es necesario realizar un diseño, dado que los conceptos que se utilizarán del mismo ya han sido definidos en el apartado anterior.

En el caso del algoritmo GRASP, su diseño será bastante directo al no existir mayor complicación. Sin embargo, en el caso del Algoritmo Cuántico de Grover, será necesario presentar primero una formulación cuántica del algoritmo, y, luego, analizar la forma correcta de realizar una simulación clásica.

3.3.1 Diseño del Algoritmo GRASP

En el caso del Algoritmo GRASP, no existe ninguna consideración en particular a la cual atenderse. Es posible realizar la formulación en pseudocódigo partiendo de la propia definición del Algoritmo GRASP y los conceptos establecidos en la definición del Algoritmo Genético y del Algoritmo GRASP de esta tesis.

La formulación en pseudocódigo sería como sigue:

N: número de genes

ListaFactoresBondadxGen: Lista con el valor resultante del siguiente cálculo: $(F_{\text{positivo}} + F_{\text{negativo}}) * F_{\text{ponderación}}$, por cada gen. El tamaño de esta lista es N.

Inicio Algoritmo GRASP

1. Inicializa_ListaFactoresBondadxGen()
2. PoblacionInicial= Lista_Vacia_Cromosomas()

3. Para $i=0$ hasta $i=\text{NumTotalDeteraciones}$
 - a. $\text{SumaFactorPonderación}=0$
 - b. $\text{Inicializa_Nuevo_Cromosoma}(\text{cromosoma})$
 - c. Mientras ($\text{SumaFactorPonderacion} < N/2$)
 - i. $\text{Ordenar_ListaFactoresBondadxGen}()$
 - ii. $\text{IndiceSuperior} = N-1$
 - iii. $\text{IndiceInferior} = 0$
 - iv. $\text{IndiceRCL} = (\text{Aleatorio}()) * (\text{IndiceInferior} + \text{Alfa} * (\text{IndiceSuperior} - \text{IndiceInferior})) + \text{IndiceInferior}$
 - v. $\text{ElementoSeleccionado} = \text{ListaFactoresBondadGen}[\text{IndiceRCL}]$
 - vi. $\text{Cromosoma.ActivaGen}(\text{ElementoSeleccionado})$
 - vii. $\text{SumaFactorPonderacion} += \text{ElementoSeleccionado.FactorPonderacion}$
 - viii. $\text{ListaFactoresBondadGen.Remove}(\text{IndiceRCL})$
 - d. Fin Mientras
 - e. Si ($\text{SumaFactorPonderacion} > N/2$)
 - i. $\text{Cromosoma.ApagaGen}(\text{IndiceRCL})$
 - f. Fin Si
 - g. $\text{PoblacionInicial.Agregar}(\text{Cromosoma})$

4. Fin Para
5. EliminarRepetidos (PoblacionInicial)

Fin Algoritmo GRASP

3.3.2 Diseño del Algoritmo Cuántico de Grover

Ahora toca realizar el diseño del Algoritmo Cuántico de Grover. Esta es quizá la parte más compleja de esta investigación pues comprende la simulación de un algoritmo eminentemente cuántico a un entorno clásico, en el cual podrá ser implementado.

Primero se describirá el Algoritmo Cuántico de Grover desde la perspectiva cuántica, y luego se planteará un enfoque clásico aceptable.

Formulación Cuántica del Algoritmo de Grover

En el apartado de la definición del Algoritmo Cuántico de Grover se plantearon algunos conceptos básicos del funcionamiento de una búsqueda cuántica estándar. Se propuso representar el universo de soluciones como una superposición de distintos qubits. Cada una de estas posibles soluciones tiene asociada una función oráculo, y esta función oráculo solo reacciona con el estado al cual está asociada, esta reacción, como se mencionó antes, se manifiesta en un “cambio de fase”, lo cual se visualiza en un cambio de signo en el estado de dicha solución candidata.

Sin embargo, hasta estos momentos, la búsqueda cuántica no representa ninguna mejora respecto a un algoritmo clásico, dado que de todas formas es necesario aplicar la función oráculo del mejor individuo a todos los candidatos del universo completo. El algoritmo de Grover proporciona una serie de pasos, para realizar esta búsqueda en una cantidad mucho menor de iteraciones.

Los pasos del algoritmo de Grover son los siguientes:

Sea $|n_k\rangle$ el estado de la solución con mayor valor de función de fitness

Inicio del Algoritmo de Grover

1. Inicializar un estado base como una superposición de todas las posibles soluciones:

$$|\varphi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{2^N-1} |n_i\rangle$$

2. Repetir \sqrt{N} veces
 - a. Se aplica el operador de Grover (G) al estado base:

$$G = (2|\varphi\rangle\langle\varphi| - I)O_k$$

Donde I es la matriz Identidad y O_k es el operador Oráculo del estado que se busca.

3. Realizar una medición del sistema y se obtendrá el estado $|n_k\rangle$ buscado.

Fin Algoritmo Grover.

Sin embargo, aquí surge el problema de que no es posible formular la función oráculo de la mejor solución candidata, porque simplemente no se conoce este valor. Entonces, es necesario buscar una solución alternativa para hallar el mejor candidato a solución óptima, utilizando el algoritmo de Grover.

Gracias a los estudios realizados en [16] y [17], podemos formular un algoritmo más completo, que presupone la ejecución del algoritmo de Grover una determinada cantidad de veces, lo cual reducirá el grado de eficiencia, pero, sin embargo, garantiza un alto nivel de eficacia.

Inicio Búsqueda Cuántica

1. Elegir de forma aleatoria un índice "m" entre 0 y 2^N-1 . Este valor será un índice tentativo del estado con mayor valor de función de fitness.

2. Repetir \sqrt{N} veces
 - a. Ejecutar el algoritmo de Grover usando la siguiente función oráculo

$$O_m(i) = \begin{cases} -1, & \text{si } f(i) > f(m) \\ 1, & \text{caso contrario} \end{cases}$$

- b. Reemplazar "m" con el índice hallado por el algoritmo cuántico de Grover en el paso a)
3. El último valor de "m" tras las iteraciones del algoritmo de Grover es la mejor solución.

Fin Búsqueda Cuántica

Resumiendo, el funcionamiento del algoritmo de búsqueda cuántica propuesto es como sigue: En cada iteración se buscará usando el algoritmo de Grover, un candidato a solución mejor que el candidato anterior (el primer candidato a solución se selecciona de forma aleatoria). Finalmente, tras \sqrt{N} ejecuciones del algoritmo, se podrá obtener la mejor solución con una altísima probabilidad (en mecánica cuántica no existen resultados cien por ciento ciertos, sino con altas probabilidades). Esto puede explicarse de la siguiente forma:

Inicialmente todos los elementos del universo tienen asociada una misma probabilidad, sin embargo al aplicar sucesivamente el operador de Grover, los elementos más cercanos en valor al elemento buscado adquieren un mayor valor de probabilidad, y, en consecuencia, los elementos que se encuentran lejos del valor buscado adquieren un valor de probabilidad menor (como se dijo, la suma de probabilidades siempre debe mantenerse constante en 1).

Tras todas las iteraciones, el elemento buscado tiene el valor de probabilidad más alto en todo el sistema, de forma que al realizar una medición cuántica, este elemento sea muy probablemente el encontrado.

Como no se sabe con exactitud cuál es el valor buscado, la función oráculo plantea que el valor que se busca es simplemente un valor mayor al de la conjetura inicial. Una vez que se termina una ejecución del algoritmo Grover, el valor que se obtenga al medir el sistema pasa a ser la nueva conjetura y así sucesivamente.

Con esto quedaría terminada la formulación del Algoritmo Cuántico de Grover, todo desde una perspectiva cuántica, el siguiente paso consiste en plantear la simulación del algoritmo de búsqueda cuántica para un entorno clásico.

Formulación Clásica del Algoritmo de Grover

Ya se ha definido de manera correcta el procedimiento necesario para calcular los mejores individuos en un universo total de soluciones candidatas, todo desde una perspectiva cuántica. El problema natural que surge es que no se cuenta con una computadora cuántica en la cual realizar una implementación del pseudocódigo ya descrito. Entonces, es necesario formular una simulación clásica, si bien, no se podrá emular la enorme eficiencia del algoritmo, pues está exclusivamente ligada a la naturaleza cuántica (superposición de estados), inexistente en una computadora clásica.

Es precisamente la carencia de elementos claves en la computación cuántica como la posibilidad de realizar superposición de estados, o el colapso no-determinístico a un estado tras realizar una medición [12], lo cual hace sumamente difícil, sino imposible, emular dicho ambiente en un entorno que por su propia naturaleza es distinto.

Por fortuna, cada elemento usado en mecánica cuántica, (y por consiguiente, en computación cuántica) puede ser representado como una matriz, o de forma genérica, por un tensor (una matriz de n dimensiones). Una superposición de estados puede representarse como un vector de N -estados. Y un operador cuántico puede representarse como una matriz de N filas por N

columnas. Aplicar un operador a una superposición de estados se reduce al producto de una matriz con un vector.

Naturalmente, el manejo de matrices y operaciones entre estas, es un proceso computacionalmente costoso en una computadora clásica. Razón por la cual la eficiencia no podrá ser medida con el tiempo que demora cada algoritmo dado que la simulación de la búsqueda cuántica en un entorno clásico, elimina de forma aparente la enorme rapidez del algoritmo de Grover. Por ello, el tiempo no será una unidad de medición de eficiencia válida. En cambio se usará la unidad: cantidad de operaciones necesarias por cada algoritmo. Es decir, una simple operación cuántica, podría demorar 5 segundos siendo simulada en una computadora clásica (debido a las operaciones matriciales a realizar), sin embargo esos 5 segundos contarán como una sola operación.

Esto se verá a más detalle en la ejecución del experimento principal.

Para este caso necesitaremos una estructura de datos adicional para guardar la información de un estado cuántico. La estructura se llamaría:

- EstadoCuántico, con los siguientes atributos: númeroEstado (un número entero), amplitudProbabilidad (un número real), valorFitness (un número real), conDesfasePorOráculo (un valor booleano), existe (un valor booleano) y valorMatriz (un número real).

Considerando la representación vectorial y matricial de estados y operadores, ahora si es posible definir el algoritmo de búsqueda cuántica simulado en un entorno clásico. Este es:

Inicio Algoritmo Cuántico de Búsqueda

1. Elegir de forma aleatoria un índice “m” entre 0 y 2^N-1 . Este valor será la conjetura inicial del índice del estado con mayor valor de función de fitness.

2. Repetir \sqrt{N} veces

a. inicializarEstadoBase (vectorEstadosInicial,N):

Inicializar el estado base como una superposición de todos los posibles candidatos, vectorEstados es un vector unidimensional de N elementos.

b. Repetir \sqrt{N} veces

i. inicializarEstadoBase (vectorEstadosActual):

Es necesario inicializar nuevamente un estado base en cada iteración del algoritmo Grover.

ii. vectorEstadosOraculo = aplicarOraculo(vectorEstadosActual,m):

Este es el resultado de aplicar el operador cuántico representado por el oráculo a la superposición de todas las soluciones candidatas.

iii. aplicarGrover

(vectorEstadosInicial,vectorEstadosActual,vectorEstadosOráculo) :
Esta función simula la aplicación del operador cuántico de Grover a una superposición de estados.

c. índiceResultado= simularMedicion (vectorEstados):

Una medición cuántica supone el colapso no-determinístico del sistema a uno de los estados que lo componen. Cada uno de estos estados tiene asociada una probabilidad de ser medido. El algoritmo de Grover hace que el estado solución tenga el mayor valor de probabilidad (muy cercana a 1). Por tanto, la función simularMedición simplemente buscará el estado con el mayor valor de probabilidad. En computación cuántica, esta medición equivale a una sola operación.

d. m= índiceResultado:

El índice que se obtenga de la medición pasa a ser la nueva conjetura.

3. El último valor de `índiceResultado` es el mejor individuo del Universo total de soluciones candidatas.

Fin Algoritmo Cuántico de Búsqueda

Aún falta definir el pseudocódigo de las siguientes funciones usadas:

Inicio `inicializarEstadoBase (vectorEstados,N)`

1. `totalElementos=2^N`
2. `probabilidad= 1/raízCuadrada(totalElementos)`
3. Para `i=0` hasta `totalElementos`
 - a. `vectorEstados[i].amplitudProbabilidad=probabilidad`
 - b. `vectorEstados[i].valorMatriz=probabilidad`
4. Fin Para

Fin `InicializarEstadoBase`

Inicio `aplicarGrover(vectorInicial,vectorActual,vectorOraculo)`

1. `coeficiente=2*productoEscalar(vectorInicial,vectorOraculo)`
2. `vectorActual= productoVectorConEscalar(coeficiente,vectorInicial)`
3. `vectorActual=restaVectores(vectorActual,vectorOraculo)`

Fin `InicializarOperadorGrover`

Inicio simularMedicion(vectorEstados,N)

1. mayorProbabilidad=0
2. Para i=0 hasta N
 - a. Si (vectorEstados[i].amplitudProbabilidad > mayorProbabilidad)
 - i. mayorProbabilidad=vectorEstados[i]. amplitudProbabilidad
 - ii. indiceMayorProbabilidad=i
 - b. Fin Si
3. Fin Para
4. Devolver vectorEstados[indiceMayorProbabilidad]

Fin SimularMedicion

Y así quedan definidos tanto el algoritmo de búsqueda cuántica principal, así como las funciones de las que hace uso. Otras funciones como productoEscalar o restaVectores no han sido definidas pues se tratan de operaciones matemáticas conocidas.

3.4 Implementación de los Algoritmos

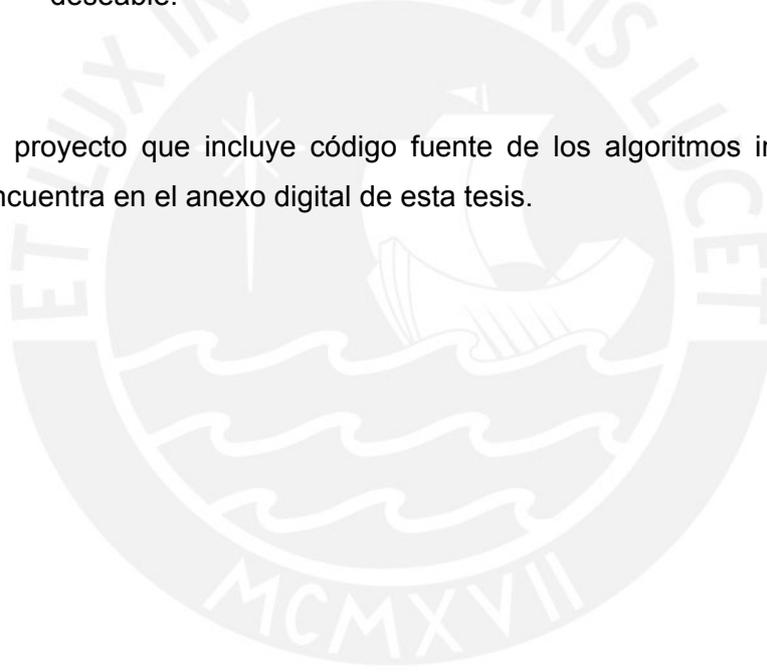
En este apartado, se procederá a realizar la implementación en código del Algoritmo GRASP y el Algoritmo Cuántico de Grover.

El lenguaje de programación que se usará será JAVA, dadas las siguientes razones:

1. Es un lenguaje de programación conocido por el tesista y el cual ha manejado durante más de un año tanto a nivel académico como laboral.

2. Cuenta con funciones gráficas sencillas las cuales serán de gran utilidad al momento de implementar el experimento principal y visualizar los resultados de la ejecución de los algoritmos.
3. Cuenta con una gran cantidad de librerías de apoyo para operaciones matemáticas, incluyendo operaciones entre matrices, las cuales serán de ayuda al momento de implementar el algoritmo de Grover
4. Tiene soporte natural para el paradigma de programación orientado a objetos. Dadas las definiciones de las estructuras necesarias para la implementación de los algoritmos, el uso de objetos es altamente deseable.

El proyecto que incluye código fuente de los algoritmos implementados se encuentra en el anexo digital de esta tesis.





4 Experimentación Numérica

En este apartado se describirá la ejecución de la experimentación numérica. Primero se explicarán los pasos que se deben seguir durante la ejecución del experimento y luego se analizarán los resultados finales obtenidos de esta experimentación numérica.

Como se mencionó en el capítulo dos, el nombre del experimento es Prueba T-Student para Dos Muestras relacionadas.

4.1 Ejecución de la Experimentación Numérica

Una vez que el algoritmo GRASP y el algoritmo cuántico de Grover ya han sido implementados, ya se puede comenzar a pensar en ejecutarlos y compararlos a través de los resultados que se obtengan. Es para ello que se formula el experimento principal descrito en el capítulo tres: para comparar la eficacia de los algoritmos (quién halló las mejores soluciones) y su eficiencia (quién lo hizo en la menor cantidad de operaciones).

Sin embargo, tras la ejecución del experimento principal, solo se obtienen resultados eminentemente numéricos. El experimento principal no tiene una forma de decir si el algoritmo GRASP o el algoritmo cuántico de Grover resultó ser más eficaz o más eficiente que el otro. Solamente puede decir a simple vista quién fue el mejor *en una iteración*, mas no qué algoritmo es en promedio el mejor. Es necesario concebir por ejemplo el caso que el algoritmo de Grover haya sido mejor que el GRASP en las diez primeras iteraciones, sin embargo en la iteración número once, el algoritmo GRASP fue mejor por un margen mucho mayor al que le sacó el algoritmo de Grover en las diez iteraciones pasadas. Este caso puede seguirse repitiendo a lo largo de cada una de las iteraciones del experimento principal, lo cual solo genera mayor incertidumbre respecto a cuál es el mejor algoritmo.

Es por esto que es necesario utilizar un método estadístico que tome como datos de entrada estos resultados numéricos, y que con sus datos de salida sea sencillo discernir qué algoritmo es el más eficiente, que algoritmo es más eficaz, y en general, que algoritmo es mejor.

Para realizar la experimentación numérica se utilizará el paquete estadístico SPSS, el cual cuenta con poderosas herramientas que permitirán realizar los análisis necesarios de una manera bastante rápida y sencilla. Solo habrá que ingresar los resultados del experimento principal, para que este software realice su labor.

Sin embargo, hay algunas consideraciones que deben tratarse antes de la ejecución propiamente dicha de la ejecución numérica con el uso del SPSS.

4.1.1 Antes de la ejecución de la Experimentación Numérica

Se deben tener las entradas listas, en este caso, serán los resultados obtenidos de la ejecución del experimento principal. Como el experimento principal requiere dos parámetros de entrada, se seleccionaron los valores tentativos de: $N=10$ (cantidad de genes por cromosoma) y cantidad de Iteraciones= 1000.

Se decidió que cada cromosoma tenga 10 genes debido a que las operaciones matriciales y vectoriales de las que hace uso la implementación del algoritmo Grover se vuelven excesivamente pesadas en una computadora clásica para valores de N mayores.

Cabe resaltar que la cantidad de iteraciones dentro del experimento principal es un valor tentativo, y podría incluso aumentarse si los resultados de la experimentación numérica no fueran del todo concluyentes.

Existían dos puntos pendientes en la implementación del algoritmo GRASP, los cuales eran: la calibración del valor del factor de relajación (alfa) y la calibración de la cantidad de iteraciones (no confundir con la cantidad de iteraciones del experimento principal).

En primer lugar, para calibrar el valor del factor de relajación (alfa), se generó una triada y se ejecutó el algoritmo GRASP sobre ésta cada vez con diferentes valores del factor. Los resultados pueden apreciarse en la siguiente gráfica:

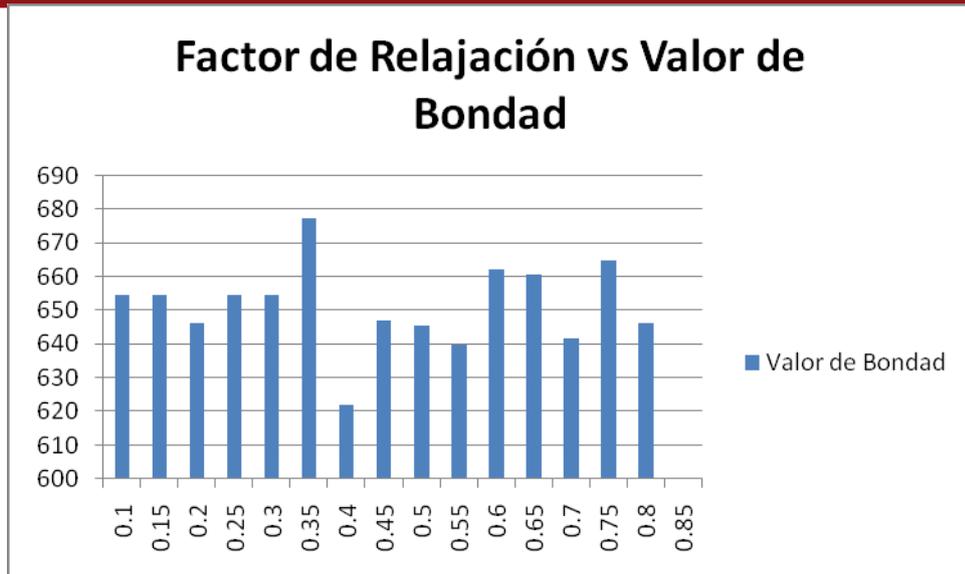


Figura 4.1: Gráfica: Factor de Relajación (alfa) vs Valor de Bondad

Como se puede apreciar, el valor de bondad más alto se obtuvo al usar el valor de 0.35 como factor de relajación. Por tanto este será el valor que se considerará en el algoritmo GRASP al ejecutar el experimento principal.

En segundo lugar, se realizó un proceso similar para calibrar la cantidad de iteraciones del algoritmo GRASP. Se usó una cantidad diferente de iteraciones en cada ejecución del algoritmo GRASP, los resultados son los siguientes:

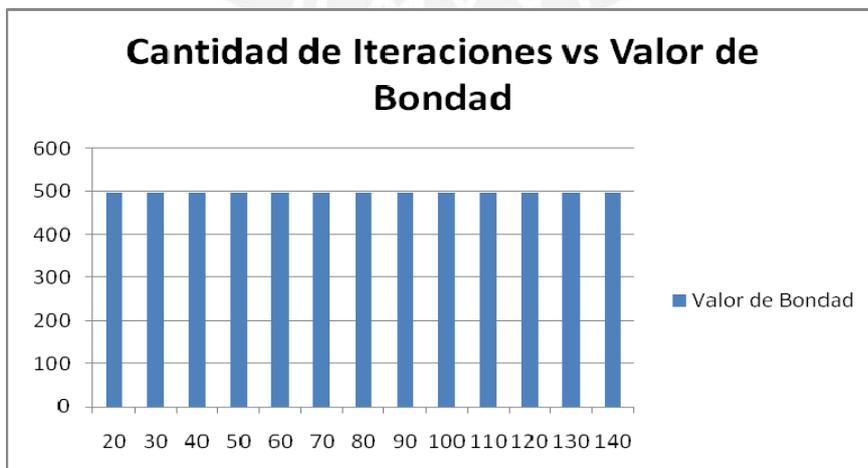


Figura 4.2: Gráfica: Cantidad de Iteraciones vs Valor de Bondad

Como puede verse, los valores de bondad hallados no muestran variación alguna al incrementar la cantidad de iteraciones. Esto se debe a que el universo no tiene un tamaño muy grande y no son necesarias muchas iteraciones para que el algoritmo GRASP llegue a una solución definitiva. Por tanto, se tomará el valor de 20 como la cantidad de iteraciones del algoritmo GRASP, dado que no exhibe ninguna variación frente a valores más grandes, y al ser el menor, permitirá una mayor rapidez en la ejecución del experimento principal.

La hoja de cálculo con los datos y las gráficas de las calibraciones se encuentran en el anexo digital de la tesis.

Ahora, como los algoritmos se encuentran totalmente definidos e implementados, se puede pasar a la Experimentación en sí.

Como se había descrito en el capítulo 3, la experimentación numérica cuenta con dos fases principales: siendo el objetivo de la primera fase demostrar empíricamente que, en efecto, un algoritmo es realmente mejor que el otro. Para ello, se plantea la siguiente hipótesis:

Sea R_{promedio} el grado de bondad promedio obtenido por el algoritmo GRASP en la ejecución del experimento principal, y sea G_{promedio} el grado de bondad promedio obtenido por el algoritmo Grover en la ejecución del experimento principal. Se plantea la siguiente hipótesis:

$$H_0 \quad R_{\text{promedio}} = G_{\text{promedio}} \quad \textit{hipótesis nula}$$

$$H_A: H_0 \text{ es falsa} \quad \textit{hipótesis alternativa}$$

Si se cumple la hipótesis nula, no habrá necesidad de realizar más pruebas, dado que eso quiere decir que ambos algoritmos tienen un grado de bondad equiparable, es decir, ninguno es significativamente mejor que el otro.

Para evaluar si la hipótesis nula planteada es correcta o no, se realizará la Prueba T-Student de dos muestras relacionadas.

4.1.2 Prueba T-Student para dos muestras relacionadas

Esta prueba puede realizarse de una forma sencilla utilizando la herramienta SPSS, las entradas de esta prueba son los valores de los grados de bondad obtenidos en cada iteración del experimento. Tanto los grados de bondad obtenidos del algoritmo GRASP, como los grados de bondad obtenidos por el algoritmo cuántico de Grover.

Lo que la prueba T-Student hace es evaluar en una distribución de probabilidad conocida, qué valores son los más representativos respecto a la media: si los valores obtenidos con la ejecución del algoritmo GRASP o los valores obtenidos bajo la ejecución del algoritmo de Grover.

Como ya se ha mencionado, si la conclusión de la prueba es que los resultados de cada algoritmo influyen de una forma igualmente significativa, debe aceptarse la hipótesis nula y concluir que ninguno de los dos algoritmos es superior al otro.

Cabe mencionar, que la prueba T-Student tiene un grado de confianza. En este caso asumiremos un 95%, es decir, de darse el caso de rechazar la hipótesis nula, se puede decir que existe un 95% de certeza de que un algoritmo sea efectivamente mejor que el otro.

Si la prueba T-Student realizada en el SPSS da como resultado que la hipótesis es falsa, quiere decir que en efecto, uno de los dos algoritmos influye de forma más significativa

Bajo la premisa anterior, es posible afirmar que el algoritmo cuyo valor de bondad promedio sea mayor, tiene una ventaja significativa sobre el otro algoritmo, y, puede ser considerado como el mejor de los dos.

4.2 Resultados

En esta sección se presentan los resultados obtenidos del experimento principal. Se determinó que se realizarían 1000 iteraciones en el experimento

y que el universo estaría determinado por una cantidad de diez genes por cromosoma.

Durante cada iteración del experimento principal se generarán los valores aleatorios de las triadas correspondientes a cada uno de los genes. Es decir los datos de entrada del algoritmo GRASP y el algoritmo de Grover van cambiando durante cada iteración del experimento principal. Si se pensara en un contexto real, se podría decir que cada iteración constituye un “problema” distinto.

La aplicación en la que se ha implementado el experimento principal cuenta con una interfaz gráfica en la cual se puede seleccionar la cantidad de iteraciones con las que se desea trabajar, así como la cantidad de genes por cromosoma. Esta aplicación muestra una simulación de la ejecución de cada algoritmo graficando qué algoritmo fue mejor en eficacia y eficiencia en cada iteración.

Sin embargo, además, la aplicación genera tres archivos de texto en los cuales se encuentran los resultados del experimento realizado.

En el primer archivo se llama “resultadosExperimento.txt” y cuenta con un resumen de los resultados obtenidos, indicando para cada iteración: los valores generados en las Triadas, el grado de bondad de los mejores individuos seleccionados por cada uno de los algoritmos, la cantidad de operaciones que se tomó cada algoritmo y una breve descripción de cuál fue el algoritmo más eficaz.

En el segundo archivo se llama “resultadosGrover.txt”, es mucho más simple y sólo imprime el grado de bondad del mejor individuo seleccionado por el algoritmo de Grover y la cantidad de operaciones cuánticas que se tomó en hallarlo.

El tercer archivo es similar al segundo, se llama “resultadosGrasp.txt” e imprime el grado de bondad del mejor individuo seleccionado por el algoritmo GRASP, así como la cantidad de operaciones clásicas necesarias para hallarlo. Los datos de este archivo junto con los del segundo conforman las

entradas que recibe el programa SPSS para realizar la experimentación numérica.

A continuación se muestra un ejemplo de los resultados obtenidos en cada uno de los archivos descritos:

- Archivo 1: "resultadosExperimento.txt"

Iteración: 0 de 1000

Datos de Entrada:

Triada 0:

Factor Positivo: 55,225

Factor Negativo: -46,249

Factor Ponderación: ,947

Triada 1:

Factor Positivo: 43,667

Factor Negativo: -32,427

Factor Ponderación: ,995

Triada 2:

Factor Positivo: 76,130

Factor Negativo: -41,958

Factor Ponderación: ,103

Triada 3:

Factor Positivo: 47,367

Factor Negativo: -44,115

Factor Ponderación: ,969

Triada 4:

Factor Positivo: 92,646

Factor Negativo: -4,463

Factor Ponderación: ,842

Triada 5:

Factor Positivo: 29,254

Factor Negativo: -10,351

Factor Ponderación: ,021

Triada 6:

Factor Positivo: 47,167

Factor Negativo: -27,754

Factor Ponderación: ,754

Triada 7:

Factor Positivo: 58,565

Factor Negativo: -3,602

Factor Ponderación: ,125

Triada 8:

Factor Positivo: 70,066

Factor Negativo: -7,403

Factor Ponderación: ,598

Triada 9:

Factor Positivo: 61,946

Factor Negativo: -4,689

Factor Ponderación: ,185

Datos de Salida:

Grado de Bondad Máximo - Algoritmo GRASP: 161,605

Grado de Bondad Máximo - Algoritmo Grover: 42,926

Algoritmo GRASP demoro: 1361 operaciones

Algoritmo Grover demoro: 1387 operaciones

Algoritmo GRASP ganó por: 118,679 unidades

- Archivo 2: “resultadosGrover.txt”

Algoritmo Grover:

42.926	1387
93.847	1387
5.975	1387
111.446	1387
103.416	1387
116.299	1387

Donde la primera columna representa los grados de bondad en cada iteración y la segunda columna, la cantidad de operaciones cuánticas necesarias.

- Archivo 3: “resultadosGrasp.txt”

Algoritmo Grasp:

161,605	1361
20,640	1501
192,633	1361
90,586	1501
45,878	1501
81,499	1501

Los archivos completos que contienen los datos que se usaron para la experimentación numérica forman parte del anexo A de la presente tesis.

Si bien es posible usar los valores de bondad máximos como grado de eficacia de cada algoritmo, existe una consideración a tomar en cuenta. El algoritmo de Grover no da respuestas con un cien por ciento de certeza, sino que da respuestas con cierto valor de probabilidad. Por tanto existe una probabilidad de que se equivoque, y es necesario reflejar este hecho en la eficacia de este algoritmo.

Una primera alternativa era multiplicar siempre el resultado obtenido por la probabilidad de conseguirla, y que sea este valor el que represente la eficacia del algoritmo de Grover. Sin embargo, existe un inconveniente con este enfoque:

Asumiendo la cantidad de genes por cromosoma igual a 10, se tiene un universo de 1024 elementos. Si aplicando el algoritmo de Grover se obtiene un resultado con un 90% de seguridad, esto quiere decir que la suma de las probabilidades de los 1023 elementos restantes debe sumar 0.10. Esto significa que, en promedio, cada uno de esos 1023 elementos tiene una probabilidad de:

$$(0.10 / 1023) * 100 = 0.009 \%$$

Es claro que una probabilidad de 90% contra 0.009% es estadísticamente aceptable para decir que el algoritmo de Grover prácticamente elegirá la respuesta correcta en una aplastante mayoría de veces.

Sin embargo, de seguir este enfoque, el valor de bondad del individuo seleccionado por Grover se reduciría en un 10%. Como se puede notar, esta reducción en el valor de bondad obtenido por Grover no queda justificada, pues aún con 90%, es prácticamente seguro que este será el resultado a obtener.

Por tanto se plantea realizar una modificación a este enfoque: multiplicar el grado de bondad por la probabilidad, sólo si la probabilidad asociada es menor que el valor promedio de todas las probabilidades obtenidas.

En el Anexo B se puede apreciar la lista de las probabilidades asociadas a cada iteración del algoritmo de Grover durante la ejecución del experimento principal.

El valor promedio de todas las probabilidades obtenidas es aproximadamente 0.7.

Por tanto, el grado de eficacia del algoritmo de Grover estará dado por el valor de bondad del individuo máximo que seleccione multiplicado por la probabilidad de obtenerlo, sólo si esta probabilidad es menor que 0.7.

En lo que respecta al grado de eficiencia de cada algoritmo en una iteración, éste será calculado como el valor inverso de la cantidad de operaciones utilizadas; dado que la cantidad de operaciones y la eficiencia están en relación inversamente proporcional. Sin embargo para que los valores de eficiencia no salgan muy pequeños se multiplicará ambos resultados por el tamaño del universo, es decir 2^N .

Ya se tienen los datos necesarios respecto a la eficacia y eficiencia de las ejecuciones de ambos algoritmos, sin embargo queda un último factor por considerar y es el de la complejidad de ambos.

Este factor se ha determinado a partir de los siguientes criterios adicionales:

Cantidad de líneas de código empleadas:

Algoritmo GRASP: 193 líneas de código del algoritmo

Algoritmo de Grover: 169 líneas de código del algoritmo

Razón de líneas de código: (GRASP/Grover) 193/169

Dificultades durante la implementación:

El algoritmo GRASP no tuvo mayor dificultad en ser implementado. Una vez se realizó su diseño en pseudocódigo, la implementación fue un trabajo directo.

Por su parte el algoritmo de Grover tuvo las siguientes dificultades durante su implementación:

- Buscar una alternativa clásica con la cual representar sus operaciones inherentemente cuánticas: Esto se resolvió a través del uso de matrices y vectores
- Una vez se determinó el uso de matrices y vectores, el siguiente problema fue garantizar un alto grado de eficiencia en las operaciones

que se realizarían dado que es conocido que las operaciones matriciales son computacionalmente costosas.

- Garantizar continuamente que el funcionamiento del algoritmo fuera aceptable. Era necesario hacer un seguimiento correcto al algoritmo a fin de asegurar que efectivamente cumplía su labor: aumentar la amplitud de probabilidad de los mejores individuos y reducía la amplitud de probabilidad de los demás.

Razón de dificultades: (GRASP/Grover) 1/2

Producto de las razones de ambos criterios: $193 / 169 * 1 / 2 = 193 / 338$

Por lo tanto el grado de complejidad del algoritmo GRASP y el algoritmo de Grover están en una razón de 193 a 388

Para que los factores de eficacia, eficiencia y complejidad contribuyan de forma equitativa al grado de bondad a calcular, se tomará una misma escala para cada uno de estos tres criterios y se asignarán los factores proporcionalmente de acuerdo a los valores hallados en el experimento principal.

Se tomará 1000 como valor máximo de la escala (este valor podría cambiar sin afectar el resultado, es una escala arbitraria), dado que es la base decimal superior más cercana a los valores de bondad calculados en el experimento. Entonces se repartirá proporcionalmente los valores hallados de la siguiente forma, por ejemplo:

Valor de bondad hallado por el algoritmo Grover: 128,78

Valor de bondad hallado por el algoritmo GRASP: 110.98

Entonces:

$$128.78 K + 110.98 K = 1000, K = 4.17$$

Por tanto:

Factor de Eficacia del algoritmo Grover: 537.120

Factor de Eficacia del algoritmo GRASP: 462.880

Como se puede ver, la relación inicial entre los valores no ha cambiado, 128.78 es a 110.98 como 537.120 es a 462.880.

Sin embargo, es necesario hacer una observación: los valores de bondad de cada algoritmo (eficacia) y la cantidad de operaciones de cada algoritmo (eficiencia) resultan en su mayoría de casos bastante cercanos, sin embargo, la diferencia de los factores de complejidad es mucho más pronunciada.

Por tanto para evitar que la diferencia entre complejidad influya de más en el factor de bondad, es necesario asignar un peso a cada uno de los factores.

Tanto la eficacia como la eficiencia son igualmente influyentes en el factor de bondad que se desea obtener, por consiguiente deben tener el mismo peso, a diferencia del factor de complejidad, el cual debe tener un peso definitivamente menor. Bajo esta premisa se plantea reformular la definición del factor de bondad de la siguiente forma:

R_B : Factor de bondad de la implementación del algoritmo cuántico de Grover.

$$R_B = 4 * (R_{\text{eficacia}} + R_{\text{eficiencia}}) - R_{\text{complejidad}}$$

G_B : Factor de bondad de la implementación del algoritmo cuántico de Grover.

$$G_B = 4 * (G_{\text{eficacia}} + G_{\text{eficiencia}}) - G_{\text{complejidad}}$$

Entonces, ahora sí ya quedan completamente definidos los factores que se ingresarán al SPSS.

El archivo del programa SPSS con los datos de las 1000 iteraciones del experimento principal se encuentran en el anexo digital de la presente tesis.

Tras ingresar los datos al programa SPSS, se ejecuta la prueba de T-Student tres veces: una para comparar la eficacia de los algoritmos, otra para comparar la eficiencia de los mismos y la última para comparar sus factores de bondad total. Se obtienen los siguientes resultados:

Comparación de factores de Eficacia:

	Mean	N	Std. Deviation	Std. Error Mean
Pair 1 EficaciaGRASP	461,618408	1000	194,9021426	6,1633469
EficaciaGrover	523,004692	1000	153,4256449	4,8517449

	N	Correlation	Sig.
Pair 1 EficaciaGRASP & EficaciaGrover	1000	-,755	,000

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
Pair 1 EficaciaGRASP - EficaciaGrover	-61,3862842	326,6376463	10,3291893	-81,6556806	-41,1168877	-5,943	999	,000

Figura 4.3: Análisis de Eficacia

En el primer cuadro se detallan las medias de los factores de eficacia, así como su desviación estándar y error estándar. En el segundo se muestra el valor de la correlación entre los datos. El tercer cuadro es el más importante, pues es el que indica si es posible rechazar o no la hipótesis nula. Es necesario que el valor de significancia hallado por la prueba T-Student sea

menor que el valor de significancia elegido antes del experimento, para poder rechazar la hipótesis nula. Como se dijo antes, el grado de confianza elegido es de 95%, al cual le corresponde un valor de significancia de 0.05, y como se puede ver en el gráfico, el valor de significancia de la prueba es menor que una diezmilésima. Luego, se rechaza la hipótesis nula y es posible concluir que un algoritmo es significativamente más eficaz que el otro. Comparando los valores promedio del primer cuadro, es posible afirmar que el algoritmo de Grover es más eficiente que el algoritmo GRASP, dado que el factor de eficacia promedio del primero (523,004) es mayor que el del segundo (461,618).

Comparación de factores de Eficiencia:

Paired Samples Statistics				
Pair1	Mean	N	Std. Deviation	Std. Error Mean
EficienciaGRASP	491,630615	1000	21,4675498	,6788635
EficienciaGrover	508,369385	1000	21,4675498	,6788635

Paired Samples Correlations			
Pair1	N	Correlation	Sig.
EficienciaGRASP & EficienciaGrover	1000	-.1000	,000

Paired Samples Test						
Pair1	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		Sig. (2-tailed)
				Lower	Upper	
EficienciaGRASP - EficienciaGrover	-16,7387699	42,9350996	1,3577271	-19,4030940	-14,0744458	,999
						df
						999
						t
						-12,329

Figura 4.4: Análisis de Eficiencia

El resultado de la prueba es bastante similar al anterior. En este caso, la prueba también da como resultado que la hipótesis nula debe rechazarse y efectivamente un algoritmo es más eficiente que el otro. Los datos del primer cuadro revelan que el algoritmo Grover es más eficiente que el algoritmo GRASP.

Comparación de factores de Bondad:

	Mean	N	Std. Deviation	Std. Error Mean
Pair1 BondadGRASP	3449,530832	1000	795,9325829	25,1895983
BondadGrover	3489,961468	1000	630,4047087	19,9351473

	N	Correlation	Sig.
Pair1 BondadGRASP & BondadGrover	1000	-.766	,000

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
Pair1 BondadGRASP - BondadGrover	-39,4305364	1,3416673E3	42,4272462	-122,6872807	43,8262079	-.929	999	,353

Figura 4.5: Análisis de Bondad

Esta es la prueba más importante pues compara los factores de bondad, y es la cual definitivamente permitirá afirmar si uno de los dos algoritmos es mejor que el otro.

Analizando los resultados del tercer cuadro se puede apreciar que el nivel de significancia es de 0.3, el cual es mucho mayor que 0.05. En consecuencia, NO es posible rechazar la hipótesis nula y se puede concluir que ningún algoritmo es mejor que el otro.

Sin embargo, este resultado guarda poca coherencia con los dos resultados anteriores. ¿Por qué ningún algoritmo es mejor que el otro, si el algoritmo Grover es mejor en eficacia y eficiencia? La respuesta es: porque el factor de complejidad del algoritmo GRASP es mucho mayor que el del algoritmo de Grover, y la diferencia entre ambos compensa la ventaja en eficacia y eficiencia que ya tenía el algoritmo de Grover.

Resulta evidente que si se asigna un peso mayor a los factores de eficacia y eficiencia, el algoritmo de Grover eventualmente podría ser calificado como mejor que el algoritmo GRASP.

Se realizó otra ejecución de la prueba T-Student pero considerando el peso del factor de complejidad como 0.5 en lugar de 1. Los resultados son los siguientes:

	Mean	N	Std. Deviation	Std. Error Mean
Pair 1 BondadGRASP2	3631,263512	1000	795,9325829	25,1695983
BondadGrover2	3807,228888	1000	630,4047087	19,9351473

	N	Correlation	Sig.
Pair 1 BondadGRASP2 & BondadGrover2	1000	-,766	,000

	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)
				Lower	Upper			
Pair 1 BondadGRASP2 - BondadGrover2	-175,9653763	1,3416673E3	42,4272462	-259,2221206	-92,7086320	-4,147	999	,000

Figura 4.6: Análisis de Bondad tras ajuste en ponderación

En este caso, la hipótesis nula si puede ser rechazada y es posible concluir que el algoritmo de Grover tiene un mayor factor de bondad que el algoritmo GRASP, y por lo tanto resulta ser el mejor de los dos de acuerdo al experimento planteado.



5 Observaciones, conclusiones y recomendaciones

Este es el último capítulo de la tesis y su contenido trata exclusivamente de analizar los resultados obtenidos en la experimentación numérica y a lo largo del desarrollo de toda la investigación. Es parte también del análisis formular nuevas interrogantes que han surgido con esta tesis a fin de plantear una serie de recomendaciones para posibles trabajos posteriores sobre este tema.

5.1 Contraste de Hipótesis

La primera parte de las conclusiones consiste en el contraste final entre los resultados de la experimentación numérica y toda la investigación, y las hipótesis planteadas inicialmente.

A continuación se listan las hipótesis planteadas y el análisis respectivo sobre su verdad o falsedad.

Hipótesis 1: Es posible la implementación del algoritmo cuántico de Grover en un entorno clásico, de tal forma que pueda aplicarse de forma efectiva en un algoritmo clásico, como es un Algoritmo Genético.

Análisis: El valor de esta hipótesis es **verdadero**. En efecto, uno de las principales complicaciones durante la fase de diseño e implementación fue la formulación del algoritmo cuántico de Grover en un entorno clásico. Al partir de una formulación eminentemente cuántica, parecía imposible realizar un algoritmo clásico equivalente que pudiera simular estructuras como estados cuánticos superpuestos, aplicación de operadores cuánticos y mediciones. No obstante, la solución que se encontró a este problema fue representar cada uno de los estados cuánticos y operadores cuánticos en su forma matricial, un enfoque matemático muy útil en la mecánica cuántica.

La repercusión de este enfoque fue perder la increíble ventaja en eficiencia que presenta el algoritmo de Grover, ya que está íntimamente relacionada con muchos aspectos inherentemente cuánticos e imposible de recrear en una naturaleza clásica. Sin embargo, el algoritmo cuántico de Grover formulada matricialmente aun mantiene su lógica y eficacia, y por tanto sigue siendo aplicable al problema de búsqueda de individuos en el algoritmo genético.

Hipótesis 2: El tiempo de ejecución de una operación cuántica del algoritmo de Grover es igual al tiempo de ejecución de una iteración del algoritmo GRASP.

Análisis: Como se mencionó en el capítulo 2, esta hipótesis **debe** ser considerada como **verdadera**. Y es vital al momento de comparar los factores de eficiencia de ambos algoritmos. Como se mencionó en el capítulo 3, no es posible simular la eficiencia natural del algoritmo de Grover en un entorno clásico, pues esta eficiencia es eminentemente cuántica y requiere de factores cuánticos, los cuales no pueden ser simulados. Por tanto, se plantea que una operación cuántica del algoritmo de Grover es equivalente a una operación clásica del algoritmo GRASP, es gracias a esta condición que la

cantidad de operaciones necesarias por cada uno de los algoritmos son valores comparables.

Hipótesis 3: El grado de bondad de la aplicación del algoritmo cuántico de Grover es igual al grado de bondad de la aplicación del algoritmo GRASP, ambos aplicados a la selección de individuos óptimos de la población de un algoritmo genético.

Análisis: Observando los resultados obtenidos durante la experimentación numérica, es posible concluir que esta hipótesis es **falsa** considerando el peso de los factores de eficacia y eficiencia como 4 y el peso del factor de complejidad como 0.5. Sin embargo, si se cambian estos valores de ponderación, la hipótesis 3 **podría resultar ser verdadera**. En definitiva, la determinación de que uno de los dos algoritmos sea mejor que el otro, depende directamente del peso que se le otorgue a cada uno de los criterios de eficacia, eficiencia y complejidad.

5.2 Conclusiones

Las principales conclusiones de esta investigación son listadas a continuación:

- El algoritmo de Grover, y en general todo algoritmo cuántico, puede ser simulado en un entorno clásico gracias a la representación matricial que cada uno posee por naturaleza. Sin embargo, esto conlleva a un decremento considerable en su eficiencia real. Siendo, entonces, fácilmente superables por algún algoritmo clásico. Esto a raíz que las operaciones entre matrices son altamente costosas computacionalmente hablando.
- El algoritmo cuántico de Grover es más eficaz y más eficiente que el algoritmo GRASP. Sin embargo, también es el más complejo de implementar de los dos.
- Existe un intervalo del valor de N (cantidad de genes por cromosoma) en el cual el algoritmo cuántico de Grover es mejor que

el algoritmo GRASP en la aplicación planteada. Para valores mayores fuera de este intervalo el algoritmo de Grover no se puede explicar pues las necesidades computacionales se vuelven demasiado altas. En esta tesis se utilizó el valor de $N=10$ (1024 elementos en el Universo) y el algoritmo de Grover funcionó, sin embargo, de usar valores cada vez más grandes como 15 o 20, se necesitaría una computadora con mayor capacidad de memoria. Para valores como $N=30$ (más de mil millones de elementos) la cantidad de memoria necesaria por el algoritmo de Grover (en su versión clásica planteada aquí) excede las capacidades actuales de una computadora promedio.

- Para que el algoritmo cuántico de Grover funcione en una búsqueda es necesario poder formular correctamente la función oráculo del elemento que se desea buscar. Esto resulta altamente deseable cuando se busca un elemento conocido en un universo grande. Sin embargo se vuelve un poderoso inconveniente cuando no se puede identificar el elemento deseado. Un claro caso de esto fue detectado durante el desarrollo de la tesis, pues no hay forma de saber si hemos encontrado el mejor individuo (o uno de los mejores) sin antes compararlo con cada uno de los elementos de la población. La solución a este problema es realizar varias ejecuciones del algoritmo de Grover como se puede comprobar en el diseño e implementación de la búsqueda cuántica.
- El algoritmo cuántico de Grover es una opción altamente recomendable para la realización de búsquedas siempre y cuando en el futuro pueda contarse con una computadora cuántica completamente operacional. Esto a raíz de que el algoritmo cuántico de Grover puede realizar búsquedas en todo el universo y no en un sub-conjunto de este universo, como hace el algoritmo GRASP.
- La implementación del algoritmo de Grover usando matrices y vectores como alternativa a la simulación de un entorno cuántico, restringe notablemente el campo de aplicaciones que podrían dársele al algoritmo en un entorno clásico. Si el tamaño del universo de

posibles soluciones crece, ésta implementación clásica del algoritmo de Grover requiere cada vez una mayor cantidad de memoria.

En la ejecución del experimento principal se trabajó con el valor de N igual a 10, lo cual significaba que cada cromosoma tuviera en total 10 genes. Ello también significaba que el algoritmo de Grover necesitaba usar matrices de 1024 elementos por 1024 elementos durante cada ejecución. Si se tratara de trabajar con un valor de $N=20$, el algoritmo de Grover necesitaría utilizar matrices de más de un millón de elementos por más de un millón de elementos, una labor computacionalmente muy costosa. Un problema real podría tener un universo de billones de posibles soluciones, lo cual descarta de inmediato la posibilidad de usar la implementación clásica del algoritmo de Grover.

5.3 Observaciones

Algunas observaciones que se notaron durante el desarrollo de esta tesis son las siguientes:

- La simulación de un algoritmo cuántico en un entorno clásico es una tarea no trivial. Optar por la representación matricial en estados y operadores cuánticos conllevó a un desgaste altamente significativo de eficiencia real, caso que no fue previsto al inicio de la formulación de la tesis.
- El tamaño del universo escogido para las ejecuciones de los algoritmos GRASP, algoritmo cuántico de Grover, y del propio experimento principal, podría no ser óptimo para establecer una conclusión más definitiva (el algoritmos GRASP podría ser más efectivo en un universo más grande), pero este tamaño estaba sujeto a las restricciones de memoria de una computadora clásica. Dado que se usaba representación matricial, solo teniendo 20 genes, se podían tener 2^{20} combinaciones, y por tanto matrices de dimensiones de más de un millón. El gasto de memoria para representar y operar estas matrices era muy alto.

- Las ponderaciones que se asignen a los factores de eficacia, eficiencia y complejidad alterarán notablemente el resultado de la experimentación numérica. Inicialmente se consideró que el peso de tanto los factores de eficacia y de eficiencia fuera 4, y el peso del factor de complejidad fuera 1. En este caso, el resultado de la experimentación numérica fue que ningún algoritmo podía ser considerado mejor que el otro. Sin embargo al reducir el peso del factor de complejidad a la mitad, la experimentación numérica dio como resultado que el algoritmo de Grover era efectivamente mejor que el algoritmo GRASP en las condiciones en las cuales ambos fueron implementados y ejecutados.

Es fácilmente deducible que si en lugar de reducir el peso del factor de complejidad, se hubiera duplicado el peso de los factores de eficacia y eficiencia, la experimentación numérica hubiera dado el mismo resultado: que el algoritmo de Grover tenía el mayor factor de bondad promedio.

Para analizar mejor este hecho, se realizó una gráfica donde se visualiza la variación del valor de bondad promedio de ambos algoritmos respecto a la razón entre el factor de eficacia y eficiencia (ambos tienen el mismo factor, por influir igual en el factor de bondad) y el factor de complejidad.

Factor de Bondad Promedio vs Razon Eficacia-Eficiencia/Complejidad

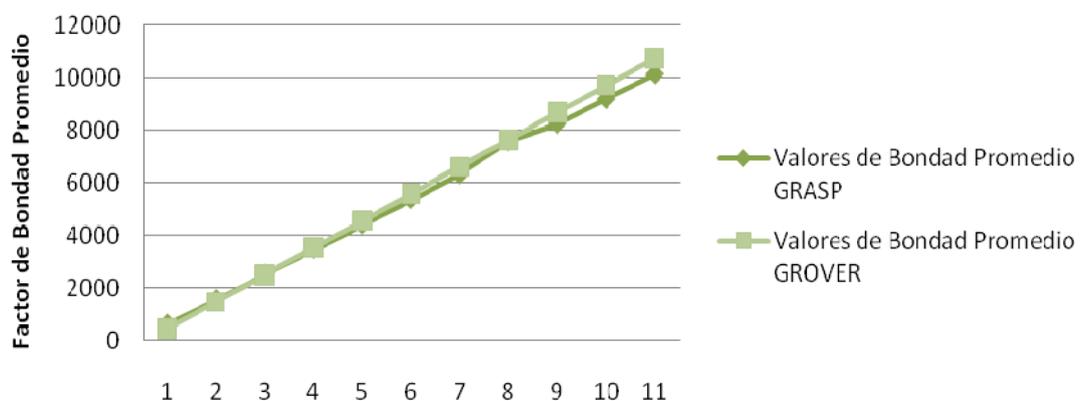


Figura 5.1: Gráfica Valores de Bondad Promedio vs Razón Eficacia-Eficiencia/Complejidad

El resultado es bastante claro, mientras más grande sea la razón entre el peso de los factores de eficacia y eficiencia, y el peso del factor de complejidad, la diferencia entre los valores de bondad promedio también aumenta, siendo el algoritmo de Grover el que alcanza los valores más altos.

5.4 Recomendaciones y trabajos futuros

Durante el desarrollo de esta tesis surgieron algunas ideas que podrían servir a investigaciones posteriores, las cuales están íntimamente relacionadas con los temas tratados en esta tesis. Estas son enumeradas a continuación:

- En la presente investigación una de las labores principales que se llevó a cabo fue la formulación clásica del algoritmo cuántico de Grover, sin embargo una interrogante que queda abierta, y se puede abordar en trabajos futuros, tiene que ver con realizar la labor inversa, representar un algoritmo clásico desde una perspectiva cuántica. En este caso la pregunta más importante a resolver sería: ¿pueden un algoritmo de ordenamiento y uno de búsqueda implementados en un contexto cuántico superar la eficiencia del algoritmo Grover? Una respuesta intuitiva podría ser no, dado que un algoritmo clásico no hace uso de todas las características ventajosas de una computadora cuántica, como sí lo hace el algoritmo de Grover. No obstante, el inconveniente más poderoso de una investigación de este tipo, sería encontrar un entorno cuántico real, en el cual simular el entorno clásico necesario para ejecutar el algoritmo de ordenamiento y el de búsqueda. Esta labor sería más complicada que su inversa (simular un algoritmo cuántico en un entorno clásico), y muy probablemente requeriría ya del uso de una computadora cuántica.
- Así como el algoritmo de Grover pudo ser representado en su forma matricial, puede quedar en manos del lector la formulación matricial de otro algoritmo cuántico como el algoritmo de Shor aplicado a la factorización de números. Es necesario recordar que no se podrá

aprovechar la eficiencia natural y real de un algoritmo cuántico debido al entorno clásico en el cual uno se encuentra y sobre el cual se realiza la implementación.

- Un trabajo futuro que se podría realizar es la ejecución del experimento planteado en esta tesis, pero definiendo factores de comparación distintos. Un inconveniente que se encontró durante la fase de la experimentación numérica fue que las diferencias entre eficacia, eficiencia y complejidad eran muy distintas entre sí, inconveniente que se solucionó asignando un peso a cada factor. Si se pudieran definir nuevos factores o cambiar la definición de los planteados, se podrían obtener quizá resultados más precisos.
- Es posible que existan otros enfoques respecto a la simulación de algoritmos cuánticos en un entorno clásico, tal es el caso de la investigación realizada en [2], en la cual se usaron estructuras primitivas y compuestas del lenguaje de programación funcional Haskell para simular superposición de estados. Sería interesante y provechoso plantear en una futura investigación, una simulación del algoritmo de Grover más eficiente y contrastar los resultados obtenidos, con los correspondientes a la simulación del algoritmo de Grover con representación matricial, tal como se realizó en la presente investigación.

6 Referencias

[1] Lee Spector 2007. *Quantum Computing*. Tutorial at the 2007 Genetic and Evolutionary Computation Conference. EE.UU. Association for Computing Machinery Digital Library. Consulta: 15 de abril del 2009
<http://portal.acm.org/citation.cfm?id=1274000.1274128&coll=ACM&dl=ACM&CFID=32151033&CFTOKEN=99217282>>

[2] Sabry Amr 2003. *Modeling Quantum Computing in Haskell*. Paper. EE.UU. Association for Computing Machinery Digital Library. Consulta: 15 de abril del 2009.
<http://portal.acm.org/citation.cfm?id=871895.871900&coll=ACM&dl=ACM&CFID=32151033&CFTOKEN=99217282>>

[3] Eleanor Rieffel, Wolfgang Polak. 2000. *An Introduction to Quantum Computing for Non-Physicists*. Paper. EE.UU. Association for Computing Machinery Digital Library. Consulta: 15 de abril del 2009.
<http://portal.acm.org/citation.cfm?id=367701.367709&coll=ACM&dl=ACM&CFID=32151033&CFTOKEN=99217282>>

[4] Ben Trivaglione. 2003. *Designing and Implementing Small Quantum Circuits and Algorithms*. Paper. EE.UU. Association for Computing Machinery Digital Library. Consulta: 15 de abril del 2009.
<http://portal.acm.org/citation.cfm?id=775832.776057&coll=ACM&dl=ACM&CFID=32151033&CFTOKEN=99217282>>

[5] Udrescu Mihai, Prodan Lucian, Vladutiu Mircea. "Implementing Quantum Genetic Algorithms: A Solution based on Grover's Algorithm". Paper. Rumania. 2006. Association for Computing Machinery Digital Library. Consulta: 10 de marzo del 2010.
http://www.fing.edu.uy/if/cursos/qcomp/clases/grover_2007.pdf

[6] Lanzagorta Marco, Uhlmann Jeffrey. 2005. *Hybrid Quantum-Classical Computing with Applications to Computer Graphics*. Paper. EE.UU. Association for Computing Machinery Digital Library. Consulta: 15 de abril del 2009.

<http://portal.acm.org/citation.cfm?id=1198555.1198723&coll=ACM&dl=ACM&CFID=32151033&CFTOKEN=99217282>>

[7] How Stuff Works. *How Quantum Computers Work*. Página Web. EE.UU. HowStuffWorks.Inc. Consulta: 1 de mayo del 2009. Disponible en:

<http://computer.howstuffworks.com/quantum-computer.htm>

[8] Lee Spector. 2004. *Automatic Quantum Computer Programming: A Genetic Programming Approach*. New York. EE.UU. Springer Science Business Media. Consulta 1 de mayo del 2009

<http://books.google.com/books?id=HzC58SW6qSQC&dq=Automatic+Quantum+Computer+Programming:+A+Genetic+Programming&printsec=frontcover&source=bl&ots=PO8DaVDn42&sig=0AzH-0uXRMQxXhlgueGtlkwikII&hl=en#PPP1.M1>

[9] Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK), PMI, 3era Edición, 2004

[10] Anthony Carpi, Ph.D. "The Scientific Method," Vision Learning Vol. SCI-1 (1), 2003.

Consulta 2 de mayo del 2009.

http://www.visionlearning.com/library/module_viewer.php?mid=45

[11] Universidad de la República | Uruguay, Facultad de Ingeniería. "Computación Cuántica- Algoritmo de Grover", Universidad de la República, Uruguay, 2003. Consulta: 10 de marzo del 2010. Disponible en

http://www.fing.edu.uy/if/cursos/qcomp/clases/grover_2007.pdf

[12] C.Lavor, L.R.U. Manssur, R. Portugal. *Grover's Algorithm: Quantum Database Search*. Paper. Instituto de Matemática e Estatística. Universidade do Estado do Rio de Janeiro. Brasil. 2003. Consulta 5 de abril del 2009. Disponible en:

<http://www.fing.edu.uy/if/cursos/qcomp/extras/notas/port-grover.pdf>

- [13] Heaton Research. *Implementation of a Generic Genetic Algorithm*. Página Web. EE.UU. Heaton Research, Inc. Consulta 29 de marzo del 2009. Disponible en <http://www.heatonresearch.com/online/introduction-neural-networks-java-edition-2/chapter-6>
- [14] Mathew Wall. *Overview of Genetic Algorithms* Página Web. EE.UU. Mechanical Engineering Department of the Massachusetts Institute of Technology. Consulta 1 de abril del 2009. Disponible en <http://lancet.mit.edu/~mbwall/presentations/IntroToGAs/P001.html>
- [15] Nielsen Michael, Chuang Isaac. 2001 (e.o. 2000). *Quantum Computation and Quantum Information*. Reino Unido. Cambridge University Press.
- [16] Whitehead Matthew. *Using Grover's Algorithm for Genetic Search*. Paper EE.UU. 2005. Consulta 4 de abril del 2009. Disponible en <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5598&rep=rep1&type=pdf>
- [17] Ahuja Ashish, Kapoor Sanjiv. *A Quantum Algorithm for Finding the Maximum*. Paper. Department of Computer Science and Engineering- Indian Institute of Technology. New Delhi. India. 1999. Consulta 4 de abril del 2009.
- [18] Goldberg, David. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Professional, 1 Edition.
- [19] Montgomery Douglas. (1991) *Diseño y Análisis de Experimentos*. Grupo Editorial Iberoamérica S.A.