

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**IMPLEMENTACIÓN DE UN ALGORITMO BÚSQUEDA TABÚ  
PARA EL PROBLEMA DE SELECCIÓN DE PORTAFOLIO  
APLICADO A INVERSIONES EN BOLSAS DE VALORES**

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

**Kevin Baba Yamakawa**

**ASESOR: Ing. Rony Cueva Moscoso**

Lima, julio de 2015

## RESUMEN DE TESIS

Las acciones son títulos emitidos por una sociedad que representan parte de su capital social. Las acciones son atractivas para inversionistas que desean obtener rentabilidad de su capital porque confieren a sus tenedores el derecho a recibir dividendos. Además, tienden a revaluarse en el tiempo y un inversor puede beneficiarse de una operación de compra y venta.

La selección de acciones para formar un portafolio de inversión ha sido un problema a resolver para los inversionistas desde el auge de los mercados de valores. En el pasado, se trataba de predecir el comportamiento de las acciones de manera rudimentaria leyendo noticias o graficando las cotizaciones. Con el pasar del tiempo, las alternativas de solución para resolver este problema han ido evolucionando y en la actualidad es común encontrar trabajos que se apoyan del poder de la informática (por ejemplo, algoritmos genéticos o redes neuronales). Seleccionar las acciones que formen parte de un portafolio de inversión es un problema de complejidad NP-difícil, lo que justifica el uso de métodos heurísticos para obtener soluciones aproximadas. El presente trabajo de fin de carrera presenta una meta heurística Búsqueda Tabú como alternativa de solución a este problema. Esta es una propuesta novedosa pues hasta el momento no se ha intentado resolver el problema de esta forma.

Para medir el desempeño del algoritmo Búsqueda Tabú, se implementó un algoritmo genético que atacaba el mismo problema y se comparó las soluciones producidas por ambos algoritmos mediante una experimentación numérica. Para el juego de datos usados en este proyecto de fin de carrera, se observó que el algoritmo búsqueda tabú tuvo mejor desempeño que el algoritmo genético produciendo soluciones con mayor rentabilidad esperada y menor riesgo.

Se concluye que el proyecto ha sido exitoso debido a que el algoritmo produce un portafolio de acciones rentable en un tiempo relativamente corto. El algoritmo puede ser usado para apoyar a los inversionistas en la toma de decisiones al construir un portafolio de inversión para una bolsa de valores.

## TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

**TÍTULO:** IMPLEMENTACIÓN DE UN ALGORITMO BÚSQUEDA TABÚ PARA EL PROBLEMA DE SELECCIÓN DE PORTAFOLIO APLICADO A INVERSIONES EN BOLSAS DE VALORES

**ÁREA:** CIENCIAS DE LA COMPUTACIÓN

**PROPOSITOR:** ING. RONY CUEVA MOSCOSO

**ASESOR:** ING. RONY CUEVA MOSCOSO

**ALUMNO:** KEVIN BABA YAMAKAWA

**CÓDIGO:** 20101812

**TEMA N°:** 612

**FECHA:** 06/04/2015

### DESCRIPCIÓN

La selección de acciones para formar un portafolio de inversión ha sido un problema al que los inversionistas se han tenido que enfrentar, desde el auge de los mercados de valores. En el pasado, se trataba de predecir el comportamiento de las acciones de manera rudimentaria leyendo noticias o graficando las cotizaciones. Con el pasar del tiempo, las alternativas de solución para resolver este problema han ido evolucionando y en la actualidad es común encontrar trabajos que se apoyan del poder de la informática (por ejemplo, algoritmos genéticos o redes neuronales). Seleccionar las acciones que formen parte de un portafolio de inversión es un problema de complejidad NP-difícil, lo que justifica el uso de métodos heurísticos para obtener soluciones aproximadas. El presente trabajo de fin de carrera presenta una meta heurística Búsqueda Tabú como alternativa de solución a este problema. Esta es una propuesta novedosa pues hasta el momento no se ha intentado atacar el problema usando esta forma de meta heurística.

Para medir el desempeño del algoritmo búsqueda tabú, se hará uso de una adaptación del algoritmo genético de Lin propuesto en el trabajo “Un algoritmo genético efectivo para el problema de optimización del portafolio multiobjetivo” [LIN, 2007]. El algoritmo de Lin ha tenido éxito atacando el problema de selección de portafolio aplicado a la Bolsa de Taiwán y será usado en este trabajo para comparar el desempeño del algoritmo búsqueda tabú propuesto mediante una experimentación numérica.

### OBJETIVO GENERAL

El objetivo general de este proyecto de fin de carrera es implementar un algoritmo búsqueda tabú como alternativa de solución para resolver el problema de selección de portafolio aplicado a inversiones en bolsas de valores y comparar el desempeño de dicho algoritmo contra algoritmos genéticos usados en la actualidad.

## OBJETIVOS ESPECÍFICOS

Los objetivos específicos de este proyecto de fin de carrera se presentan a continuación:

1. Definir la función objetivo a ser usada en los algoritmos genético y búsqueda tabú.
2. Diseñar un algoritmo genético como alternativa de solución al problema de selección de portafolio en la bolsa de valores.
3. Diseñar un algoritmo de búsqueda tabú como alternativa de solución al problema de selección de portafolio en la bolsa de valores.
4. Implementar los algoritmos genético y búsqueda tabú en un lenguaje de programación.
5. Seleccionar y desarrollar la experimentación numérica para comparar el desempeño de los algoritmos genético y búsqueda tabú.

## ALCANCE

Este proyecto de fin de carrera pretende presentar una alternativa de solución al problema de selección de portafolio aplicado a inversiones en bolsas de valores. Es decir, debe de ayudar a seleccionar acciones para formar un portafolio de inversión el fin de obtener la mayor ganancia posible. Para lograr este objetivo, se tomarán en cuenta ciertas variables que pueden ser consideradas como las más importantes para la adecuada selección del proyecto. Estas variables son: el capital de inversión disponible, el histórico de cotizaciones de las acciones, el beneficio estimado de cada acción y el riesgo estimado de cada acción.

Para implementar el algoritmo búsqueda tabú que ayudará a resolver este problema primero se definirá la función objetivo que el algoritmo intentará maximizar. Luego, se diseñará el algoritmo teniendo en cuenta las estructuras de datos en las que tiene que apoyarse. Posteriormente, se implementará el algoritmo búsqueda tabú propuesto y el algoritmo genético de Lin en un lenguaje de programación. Finalmente, se llevará a cabo la experimentación numérica que permitirá comparar el desempeño de los dos algoritmos planteados. El algoritmo genético es en la actualidad una alternativa muy popular para el problema de selección de portafolio aplicado a bolsas de valores y servirá como punto de referencia para medir el desempeño del algoritmo búsqueda tabú implementado en este proyecto de fin de carrera.

## REFERENCIAS

**[LIN, 2007] Un algoritmo genético efectivo para el problema de optimización del portafolio multiobjetivo.**

LIN, Chi-Ming.

2007. Applied Mathematical Sciences, Vol. 1, p.201-210.

## Tabla de contenido

<b>1 PROBLEMÁTICA</b>	<b>3</b>
<b>1.1 OBJETIVO GENERAL</b>	<b>5</b>
<b>1.2 OBJETIVOS ESPECÍFICOS</b>	<b>5</b>
<b>1.3 RESULTADOS ESPERADOS</b>	<b>5</b>
<b>1.4 HERRAMIENTAS, MÉTODOS, METODOLOGÍAS Y PROCEDIMIENTOS</b>	<b>7</b>
<b>1.4.1 INTRODUCCIÓN</b>	<b>7</b>
<b>1.4.2 HERRAMIENTAS</b>	<b>7</b>
<b>1.4.3 MÉTODOS Y PROCEDIMIENTOS</b>	<b>8</b>
<b>1.4.4 METODOLOGÍAS</b>	<b>9</b>
<b>1.5 ALCANCE</b>	<b>10</b>
<b>1.5.1 LIMITACIONES</b>	<b>10</b>
<b>1.5.2 RIESGOS</b>	<b>10</b>
<b>1.6 JUSTIFICACIÓN Y VIABILIDAD</b>	<b>11</b>
<b>1.6.1 JUSTIFICACION</b>	<b>11</b>
<b>1.6.2 VIABILIDAD</b>	<b>11</b>
<b>2 MARCO TEÓRICO</b>	<b>14</b>
<b>2.1 MARCO CONCEPTUAL</b>	<b>14</b>
<b>2.1.1 INTRODUCCIÓN</b>	<b>14</b>
<b>2.1.2 COMPLEJIDAD DE UN PROBLEMA COMPUTACIONAL</b>	<b>14</b>
<b>2.1.3 PROBLEMA DE LA MOCHILA</b>	<b>14</b>
<b>2.1.4 CONCEPTOS GENERALES DE HEURÍSTICAS</b>	<b>15</b>
<b>2.1.5 CONCEPTOS GENERALES DE META-HEURÍSTICAS</b>	<b>15</b>
<b>2.1.6 CONCEPTOS GENERALES DE MODELADO Y SIMULACIÓN</b>	<b>18</b>
<b>2.1.7 CONCEPTOS GENERALES DEL MERCADO DE VALORES</b>	<b>19</b>
<b>2.2 CONCLUSIÓN</b>	<b>21</b>
<b>3 ESTADO DEL ARTE</b>	<b>22</b>
<b>3.1 INTRODUCCIÓN</b>	<b>22</b>
<b>3.2 OBJETIVOS DE LA REVISIÓN DEL ESTADO DEL ARTE</b>	<b>22</b>
<b>3.3 MÉTODO USADO EN LA REVISIÓN DEL ESTADO DEL ARTE</b>	<b>22</b>
<b>3.4 MÉTODOS CONVENCIONALES</b>	<b>22</b>
<b>3.5 PROYECTOS DE INVESTIGACIÓN PARA RESOLVER EL PROBLEMA DEL PORTAFOLIO</b>	<b>23</b>
<b>3.6 OTROS PROYECTOS DE INVESTIGACIÓN RELACIONADOS A LA BOLSA DE VALORES</b>	<b>24</b>
<b>3.6 PRODUCTOS COMERCIALES PARA RESOLVER EL PROBLEMA</b>	<b>25</b>
<b>3.7 CONCLUSIONES SOBRE EL ESTADO DEL ARTE</b>	<b>27</b>
<b>4 DEFINICIÓN DE LA FUNCIÓN OBJETIVO</b>	<b>28</b>
<b>4.1 FUNCIÓN OBJETIVO</b>	<b>28</b>
<b>4.2 ESTRUCTURAS DE DATOS</b>	<b>29</b>

<b>5 DISEÑO DEL ALGORITMO GENÉTICO</b>	<b>32</b>
<b>5.1 VARIABLES Y PARÁMETROS</b>	<b>32</b>
<b>5.2 PSEUDOCÓDIGO DEL ALGORITMO GENÉTICO</b>	<b>32</b>
<b>6 DISEÑO DEL ALGORITMO BÚSQUEDA TABÚ</b>	<b>37</b>
<b>6.1 ESTRUCTURAS DE DATOS</b>	<b>37</b>
<b>6.2 VARIABLES Y PARÁMETROS</b>	<b>38</b>
<b>6.3 PSEUDOCÓDIGO DEL ALGORITMO BÚSQUEDA TABÚ</b>	<b>38</b>
<b>7 INTERFAZ GRÁFICA</b>	<b>46</b>
<b>8 CALIBRACIÓN DE VARIABLES Y PARÁMETROS</b>	<b>49</b>
<b>9 EXPERIMENTACIÓN NUMÉRICA</b>	<b>51</b>
<b>9.1 RESULTADOS DE LAS EJECUCIONES</b>	<b>51</b>
<b>9.2 PRUEBA KOLMOGOROV-SMIRNOV</b>	<b>52</b>
<b>9.3 PRUEBA F DE FISHER</b>	<b>53</b>
<b>9.4 PRUEBA Z</b>	<b>53</b>
<b>10 CONCLUSIONES</b>	<b>55</b>
<b>10.1 OBSERVACIONES</b>	<b>55</b>
<b>10.2 CONCLUSIONES</b>	<b>55</b>
<b>10.3 RECOMENDACIONES</b>	<b>55</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>57</b>

## CAPÍTULO 1

### 1 Problemática

Una acción es un título emitido por una sociedad que representa una fracción de su capital social. Las acciones confieren a sus emisores derechos políticos (votar en la junta de accionistas) o económicos (recibir parte de las utilidades de la empresa en la forma de dividendos) [MORA ENGUDANOS, 2008]. A lo largo de la historia, las acciones han sido un tema de interés para inversionistas que desean obtener rentabilidad de su capital. Además de los dividendos que se reciben por ser un tenedor de acciones, las acciones son atractivas debido a que tienden a revaluarse en el tiempo y un inversor puede beneficiarse de una operación de compra y venta [CHIAVENATO, 1989].

Un primer problema que han tenido los inversionistas es identificar patrones en el comportamiento y fluctuación de los precios de las acciones. Es importante que el accionista pueda predecir los cambios en las cotizaciones de las acciones para así comprar al comienzo de una tendencia alcista y vender antes de que empiece una tendencia bajista, de esta manera obteniendo la mayor rentabilidad posible. En el pasado, existían dos técnicas tradicionales para intentar predecir el comportamiento del mercado bursátil: el análisis técnico y el análisis fundamental [KROHA, 2004]. En el análisis técnico, se recolectaba y graficaba la cotización histórica de las acciones. Luego, el accionista debía de analizar los gráficos generados y trataba de identificar patrones. En el análisis fundamental, el enfoque del accionista era mantenerse al tanto de los acontecimientos que podían afectar el desempeño de una empresa leyendo documentos como estados financieros, memorias anuales y noticias. Una vez identificado algún evento trascendente, el inversor debía de actuar rápido antes de que el evento tenga efecto en la cotización de la acción. Ambos métodos tenían sus limitaciones. El análisis técnico solo identificaba comportamientos, más no las causas de estos comportamientos; mientras que el análisis fundamental requería de una respuesta rápida del accionista y se veía influenciado por el juicio experto del inversor al momento de elegir las acciones a estudiar [KROHA, 2004]. Con el desarrollo de las computadoras, aparecieron nuevas técnicas para identificar patrones en las cotizaciones de las acciones [ZABIR, 2011], siendo una herramienta muy poderosa que se emplea en la actualidad las redes neuronales. Las redes neuronales aprovechan varios procesadores en paralelo para analizar un gran volumen de data histórica. A partir de esa data se pueden armar modelos matemáticos (por ejemplo, usando el método de mínimos cuadrados) que pueden permitir predecir tendencias [ZABIR, 2011].

Otro problema que han tenido que enfrentar los inversionistas es el identificar si una acción está valorada cara o barata [CASANOVA, 2006]. Percatarse de que una acción está subvalorada o sobrevalorada es importante porque permite identificar una buena oportunidad para comprar o vender dicha acción. Un método muy popular que se usa en la actualidad es el de la media móvil, donde se compara la media móvil a largo plazo y la media móvil a corto plazo de las cotizaciones de una acción [CHOU, 1975]. Existen diversos trabajos que emplean esta técnica para analizar el mercado bursátil. Por ejemplo, el software Stock Signal Pro utiliza medias móviles para alertar al usuario cuando es buen momento de comprar o vender las acciones a las que se les está haciendo seguimiento [STOCK SIGNAL PRO, 2014].

Adicionalmente, una consideración que los accionistas han tenido que tomar en cuenta al momento de realizar inversiones es el nivel de riesgo que quieren asumir. Siguiendo el principio de Markowitz, los inversionistas buscarán obtener la mayor rentabilidad con el menor riesgo posible [RUBINSTEIN, 2002]. La manera tradicional para afrontar el riesgo en el mercado bursátil es diversificar el portafolio de acciones adquiriendo acciones de diferentes sectores. Cabe decir que, debido a que gran parte de la información relacionada al nivel de riesgo de una acción es de carácter no numérico recientemente han aparecido trabajos en el que se emplea lógica fuzzy para cuantificar el riesgo de inversiones en el mercado bursátil [LIAN, LI, 2010].

Una vez que se identifican los patrones en las cotizaciones de las acciones y se determina si dichas acciones están subvaloradas o sobrevaloradas, el inversionista debe de seleccionar la combinación óptima de acciones a adquirir con el capital que se tiene disponible. Sin embargo, en problemas de combinatoria como estos, el número de soluciones posibles suele ser tan grande que examinarlas todas no es viable. Incluso con una computadora, analizar todas las combinaciones posibles cuando estas crecen de manera no polinómica con respecto al número de acciones que se cotizan es demasiado caro computacionalmente. Por esta razón, estos problemas se clasifican como de clase NP-difíciles (polinómico no determinístico) y justifican el uso de algoritmos heurísticos para resolverlos [RIOS MERCADO, BARD, 2000].

Los algoritmos heurísticos son estrategias que limitan el espacio de búsqueda para encontrar soluciones aproximadas en tiempos razonables. Sin embargo, se sabe que los algoritmos heurísticos son altamente voraces y no miden las consecuencias que algunas decisiones podrían tener en el futuro [TUPIA, 2005]. Esto ocasiona que a veces los algoritmos heurísticos devuelvan soluciones que se alejan demasiado del óptimo global. Por esta razón, históricamente se ha preferido utilizar algoritmos meta-heurísticos para el problema de inversiones bursátiles, en particular variantes de algoritmos genéticos (algoritmos que se basan en la evolución biológica y en la supervivencia del más fuerte). A pesar de que los algoritmos genéticos tampoco resuelven el problema de forma exacta, utilizan aleatoriedad para superar la voracidad de los algoritmos heurísticos y así generan soluciones que se acercan de manera aceptable a la solución óptima. Por ejemplo, en 2007, Lin aplica un algoritmo genético con múltiples objetivos al problema de selección del portafolio obteniendo rentabilidad positiva en todas las corridas [LIN, 2007]. Sin embargo, es conocido que los algoritmos genéticos suelen tener tiempos de respuesta relativamente elevados en comparación a otros algoritmos meta-heurísticos [ZAHRA, 2004]. Esto representa un problema para el inversor dado a que es importante reaccionar rápido en el mercado bursátil debido a la volatilidad del mismo.

En conclusión, se ha visto que son varios los problemas relacionados a las inversiones en el mercado de valores: es complicado identificar patrones en las cotizaciones de las acciones, es difícil determinar si una acción está sobrevalorada o subvalorada, y es complejo minimizar el riesgo de un portafolio de acciones. Existen diversos trabajos en los que se han usado el poder de la informática para presentar alternativas de solución a estas dificultades. Sin embargo, aún queda un problema por resolver: la carencia de herramientas que permitan seleccionar acciones a adquirir en

el momento oportuno. Este es el problema central que será atacado en este proyecto de fin de carrera por medio de un algoritmo meta-heurístico que permita obtener una solución aceptable en un tiempo relativamente corto. En particular, se usará una variante del algoritmo búsquedas tabú debido a que Zahra demuestra que este tipo de algoritmo suele tener menor tiempo de respuesta que los algoritmos genéticos y produce soluciones de calidad similar [ZAHRA, 2004]. Se desea corroborar esta aseveración en este proyecto de fin de carrera. Las variables que tomará en cuenta el algoritmo son: el capital disponible, la tolerancia al riesgo del inversionista, la ventana de tiempo de la inversión y la cotización diaria de las acciones. El éxito del proyecto se determinará por medio de una experimentación numérica que permita comparar el desempeño del algoritmo de Lin con el algoritmo propuesto en este trabajo. Se espera que este proyecto de fin de carrera pueda beneficiar tanto a los inversionistas como a las empresas emisoras de acciones. Los inversionistas pueden maximizar su rentabilidad y los emisores de acciones prometedoras puedan obtener el capital necesario para sacar a la empresa adelante.

### 1.1 Objetivo general

Implementar un algoritmo de búsquedas tabú para resolver el problema de selección de portafolio aplicado a inversiones en bolsas de valores y comparar el desempeño de dicho algoritmo contra algoritmos genéticos usados en la actualidad.

### 1.2 Objetivos específicos

Objetivo 1:

Definir la función objetivo a ser usada en los algoritmos genético y búsquedas tabú.

Objetivo 2:

Diseñar un algoritmo genético como alternativa de solución al problema de selección de portafolio en la bolsa de valores.

Objetivo 3:

Diseñar un algoritmo de búsquedas tabú como alternativa de solución al problema de selección de portafolio en la bolsa de valores.

Objetivo 4:

Implementar los algoritmos genético y búsquedas tabú en un lenguaje de programación.

Objetivo 5:

Seleccionar y desarrollar la experimentación numérica para comparar el desempeño de los algoritmos genético y búsquedas tabú.

### 1.3 Resultados esperados

Resultados esperados del objetivo 1:

Resultado	Indicadores
1) Estructuras de datos a usar en los algoritmos genéticos y búsquedas	Existencia de la estructura de datos para representar la solución.

tabú.	
2) Función objetivo que mida el desempeño del portafolio de acciones seleccionadas.	Presencia de la función objetivo que mida el desempeño del portafolio de acciones seleccionadas.

Resultados esperados del objetivo 2:

Resultado	Indicadores
1) Estructuras de datos que soporten la implementación del algoritmo genético.	Existencia de la estructura de datos que soporte la implementación del algoritmo genético.
2) Algoritmo genético diseñado.	Presencia del algoritmo genético diseñado.

Resultados esperados del objetivo 3:

Resultado	Indicadores
1) Estructuras de datos que soporten la implementación del algoritmo búsqueda tabú.	Existencia de la estructura de datos que soporte la implementación del algoritmo búsqueda tabú.
2) Algoritmo búsqueda tabú diseñado.	Presencia del algoritmo búsqueda tabú diseñado.

Resultados esperados del objetivo 4:

Resultado	Indicadores
1) Módulo para la carga masiva de datos de las cotizaciones de las acciones.	Existencia del módulo para la carga masiva de datos de las cotizaciones de las acciones.
2) Módulo del algoritmo genético implementado.	Presencia del módulo del algoritmo genético implementado.
3) Módulo del algoritmo búsqueda tabú implementado.	Existencia del módulo del algoritmo búsqueda tabú implementado.

Resultados esperados del objetivo 5:

Resultado	Indicadores
1) Calibración de variables de la experimentación numérica	Presencia de las variables de la experimentación numérica calibradas.
2) Informe de experimentación numérica que contendrá la descripción de las pruebas a realizar, los resultados de las pruebas y la interpretación de los resultados.	Existencia del informe de experimentación numérica.
3) Interfaz para mostrar los resultados obtenidos por el algoritmo genético y el algoritmo Búsqueda Tabú.	Presencia de la interfaz para mostrar los resultados de la experimentación numérica.

## 1.4 Herramientas, métodos, metodologías y procedimientos

### 1.4.1 Introducción

A continuación se presentarán las herramientas, métodos y procedimientos a usar en el presente proyecto de fin de carrera.

Resultados esperado	Herramientas a usarse
RE1: Estructura de datos para representar una solución.	
RE2: Función objetivo que mida la rentabilidad esperada de una acción en función a su comportamiento, a su sobrevaloración.	
RE3: Estructuras de datos que soporten la implementación del algoritmo genético.	
RE4: Algoritmo genético desarrollado.	
RE5: Estructuras de datos que soporten la implementación del algoritmo Búsqueda Tabú.	
RE6: Algoritmo búsqueda tabú desarrollado.	
RE7: Calibración de variables de la experimentación numérica	<ul style="list-style-type: none"> <li>- Scrum</li> <li>- Netbeans</li> <li>- Java</li> </ul>
RE8: Informe de experimentación numérica que contendrá la descripción de las pruebas a realizar, los resultados de las pruebas y la interpretación de los resultados.	<ul style="list-style-type: none"> <li>- Comparación de dos tratamientos (Prueba t de Student, Prueba F de Fisher, Prueba Kolmogorov-Smirnov)</li> <li>- Microsoft Excel 2010</li> </ul>
RE9: Interfaz para mostrar los resultados obtenidos por el algoritmo genético y el algoritmo búsqueda tabú.	<ul style="list-style-type: none"> <li>- Scrum</li> <li>- Netbeans</li> <li>- Java</li> </ul>
RE10: Módulo para la carga masiva de datos de las cotizaciones de las acciones.	<ul style="list-style-type: none"> <li>- Scrum</li> <li>- Netbeans</li> <li>- Java</li> </ul>
RE11: Módulo del algoritmo genético implementado.	<ul style="list-style-type: none"> <li>- Scrum</li> <li>- Netbeans</li> <li>- Java</li> </ul>
RE12: Módulo del algoritmo búsqueda tabú implementado.	<ul style="list-style-type: none"> <li>- Scrum</li> <li>- Netbeans</li> <li>- Java</li> </ul>

Tabla 1.4.1.1: Herramientas, Métodos y Procedimientos

### 1.4.2 Herramientas

#### 1.4.2.1 NetBeans

NetBeans es un IDE (entorno de desarrollo integrado) desarrollado para escribir, compilar y ejecutar programas en diversos lenguajes de programación. Cuenta con una interfaz que permite editar código de una manera rápida y facilita la administración de proyectos de programación que se tienen [NETBEANS, 2014].

Se ha optado por el uso de NetBeans debido a que es una herramienta que facilita la administración de proyectos de programación. Conjuntamente, NetBeans ofrece funcionalidades que simplifican la edición del código (como autocompletar), lo cual agiliza la programación. Finalmente, otra razón por la elección de NetBeans es el hecho de que es compatible con el lenguaje de programación Java que se desea utilizar.

#### 1.4.2.2 Java

Java es un lenguaje de programación orientado a objetos. Este lenguaje de programación es simple, distribuido, interpretado, robusto, seguro, portable y multiplataforma (funciona en varios sistemas operativos) [JAVA, 2014].

Se ha elegido este lenguaje de programación debido a que es multiplataforma y es compatible con el IDE Netbeans. Otro motivo para elegir este lenguaje es que ya se cuenta con el conocimiento del mismo.

#### 1.4.2.3 Microsoft Excel

Microsoft Excel es una aplicación que pertenece a la suite Microsoft Office 2010. Es un programa que está compuesto por hojas de cálculo. Permite generar trabajos que requieren de cálculos como reportes estadísticos, informes de ventas, entre otros. Microsoft Excel también permite la programación de macros usando VBA (Visual Basic for Applications). Estas funcionalidades permiten ahorrar tiempo, simplificar el trabajo y aumentar la productividad [MICROSOFT OFFICE, 2010].

Se justifica el uso de esta herramienta debido a que Microsoft Excel cuenta con funciones estadísticas necesarias para procesar los resultados de la experimentación numérica.

### 1.4.3 *Métodos y Procedimientos*

#### 1.4.3.1 Comparación de dos tratamientos

Esta es una prueba estadística usada para comparar las medias de dos o más poblaciones. La comparación de dos tratamientos está compuesta de tres subpruebas que se detallan a continuación:

##### Prueba Kolmogorov-Smirnov

La prueba de Kolmogorov-Smirnov para una muestra es un procedimiento de bondad de ajuste que permite medir el grado de concordancia existente entre la distribución de un conjunto de datos y una distribución teórica específica. Su objetivo es señalar si los datos provienen de una población que tiene la distribución teórica especificada; es decir, decide si las observaciones podrían proceder de la distribución especificada [GARCIA BELLIDO, 2010]

La razón para usar esta prueba es para asegurar que los resultados obtenidos se comportan de manera normal y así poder aplicar la prueba estadística t para la comparación de dos muestras.

### Prueba F de Fisher

La prueba F de Fisher se emplea para probar si dos muestras provienen de poblaciones que poseen varianzas iguales. Esta prueba es útil para determinar si una población normal tiene mayor variación que la otra [YCART, 2014].

La prueba F de Fisher se utiliza debido a que un requisito para realizar la prueba estadística t es saber si las varianzas de las muestras a comparar son homogéneas o no.

### Prueba t de Student

La prueba t se utiliza para comparar las medias de dos muestras normales. Se basa en la diferencia entre las proporciones de dos muestras. Esta prueba puede ser usada para la diferencia entre las medias de dos muestras para determinar si existe una diferencia en la media de dos grupos (prueba de dos colas) o si un grupo tiene mayor media que el otro grupo (prueba de una cola).

Para comparar el desempeño de los algoritmos genético y búsqueda tabú se ejecutará cada algoritmo 30 veces y se almacenarán los resultados. La prueba t de Student es la más apropiada para comparar las medias de muestras pequeñas (menor a 35 miembros) y por lo tanto su uso está justificado.

## 1.4.4 Metodologías

### 1.4.4.1 Scrum

Scrum es una metodología de desarrollo ágil desarrollada por Takeuchi y Nonaka. La particularidad de esta metodología es que se enfoca más en la entrega de valor que en tener una planificación estricta.

Scrum tiene como pilares: la transparencia, la inspección y la adaptación [TAKEUCHI, NONAKA, 1986].

En este proyecto de fin de carrera, se usarán algunas buenas prácticas de la metodología Scrum. Se mencionan las siguientes:

- Tener un *product backlog* para gestionar requisitos.
- Trabajar en iteraciones.
- Realizar un *daily scrum*, donde se responde las siguientes preguntas: ¿qué hice?, ¿qué haré?, ¿cuáles fueron los impedimentos?.
- Realizar una retrospectiva al final de cada iteración para revisar qué se hizo bien y qué se pudo hacer mejor.

La razón por la cual se ha optado por usar esta metodología es que al estar en un proyecto de fin de carrera de desarrollo de algoritmos, es preferible enfocarse más en la programación que en una planificación rígida. Scrum es una metodología ágil madura –lleva más de 20 años siendo usada- y con amplia documentación disponible.

## 1.5 Alcance

Este proyecto de fin de carrera pretende presentar una alternativa de solución al problema de inversiones en bolsas de valores. Es decir, debe de ayudar a seleccionar acciones en las cuales invertir y decidir el momento más adecuado para comprar/vender con el fin de obtener la mayor ganancia posible. Para lograr este objetivo, se tomarán en cuenta ciertas variables que pueden ser consideradas como las más importantes para la adecuada selección del proyecto. Estas variables son: el capital disponible, la tolerancia al riesgo del inversionista, la ventana de tiempo de la inversión y la cotización diaria de las acciones.

A continuación, se procederá a implementar los algoritmos búsqueda tabú y algoritmo genético. Se identificarán las estructuras de datos que servirán de apoyo en la implementación del proyecto. Luego, se debe de desarrollar la aplicación para finalmente llevar a cabo la experimentación numérica que permitirá comparar los dos algoritmos planteados. El algoritmo genético es en la actualidad la alternativa de solución más popular para el problema de las inversiones en bolsas de valores y servirá como punto de referencia para medir el desempeño del algoritmo búsqueda tabú.

Cabe decir que escapa del alcance el armar un portafolio de inversión con acciones que coticen en divisas diferentes.

### 1.5.1 Limitaciones

La siguiente lista muestra las limitaciones de este proyecto de fin de carrera:

- El tiempo de ejecución de los algoritmos puede variar dependiendo de las características del hardware y software de la computadora en la que se ejecuten. Esto se debe a la complejidad de los algoritmos y a la cantidad de veces que deben de ser ejecutados para obtener los resultados. Se obtendrá el tiempo de ejecución de los algoritmos comparando la diferencia entre la hora del sistema al inicio de la ejecución y la hora del sistema al final de la ejecución. Ambos algoritmos se ejecutarán en el mismo equipo para que la prueba sea justa.
- Los resultados del proyecto pueden verse afectados por la cantidad de los datos de entrada debido a que se tendrá que procesar más información para obtener los resultados. Para este proyecto, la data de las cotizaciones de las acciones se obtendrá del portal web de la Bolsa de Valores de Lima.
- El alcance del proyecto no abarca el desarrollo de un sistema de información como soporte a los algoritmos implementados.

### 1.5.2 Riesgos

En la siguiente tabla se muestran los riesgos identificados que podrían impactar este proyecto de fin de carrera.

Riesgo identificado	Impacto en el proyecto	Medidas correctivas para mitigar
Mala planificación del proyecto.	- Entregables revisados fuera de tiempo - Rechazo de los	Revisar los plazos de entrega para organizar el tiempo de tal manera que se pueda cumplir

	entregables	con el proyecto
Pérdida parcial o total de la información y avances del proyecto.	<ul style="list-style-type: none"> <li>- No se presentan entregables</li> <li>- Pérdida de tiempo, ya que se debe de volver a hacer el trabajo para obtener la información perdida</li> <li>- Desaprobar el curso</li> </ul>	Tener un repositorio en la nube del proyecto y hacer respaldos de manera periódica
Enfermedades.	<ul style="list-style-type: none"> <li>-Entregables entregados fuera de tiempo</li> <li>-Rechazo de los entregables</li> </ul>	Organizar la ejecución del proyecto de tal manera que se tengan los entregables con anticipación
Mala comunicación con el asesor.	<ul style="list-style-type: none"> <li>- Entregables no pueden ser corregidos</li> <li>-Rechazo de los entregables</li> <li>- Ya que no hay asesoría de un experto, se pueden cometer errores graves en el proyecto</li> </ul>	Comunicar oportunamente al profesor de curso para la asignación de otro asesor

## 1.6 Justificación y viabilidad

### 1.6.1 Justificación del proyecto de fin de carrera

En la actualidad, las bolsas de valores es un tema de interés para inversionistas que desean obtener rentabilidad de su capital. Esto se debe a que las acciones tienden a revaluarse en el tiempo y se puede tener un beneficio significativo de una operación de compra y venta.

Elegir acciones atractivas para invertir y decidir el momento oportuno para comprar y vender es clave para obtener la máxima rentabilidad posible. Además, es importante considerar el nivel de riesgo al cual se quiere exponer el inversionista y evitar acciones demasiado volátiles si la tolerancia al riesgo es baja. Actualmente, existen algunas herramientas que resuelven de manera parcial estos problemas. Sin embargo, la mayoría de estas soluciones emplean variantes de algoritmos genéticos cuando es conocido que el algoritmo de búsqueda tabú tiene mejor desempeño.

El presente proyecto de fin de carrera, pues, permitirá apoyar a los inversionistas en la selección de acciones y orientará a decidir el mejor momento de comprar y vender. Esto se realizará estudiando los patrones en las cotizaciones de las acciones usando un algoritmo meta-heurístico que dé soluciones aceptables en tiempos razonables. El algoritmo tomará en cuenta el nivel de riesgo al cual se quiere exponer el inversionista.

### 1.6.2 Análisis de la viabilidad del proyecto de fin de carrera

En el siguiente apartado se presentan los factores que influyen en la viabilidad del proyecto de fin de carrera.

### **Viabilidad técnica**

Se cuenta con los conocimientos necesarios para alcanzar los objetivos del presente proyecto de fin de carrera. Se ha llevado cursos a lo largo de la carrera de Ingeniería Informática en los que se ha empleado el entorno de desarrollo NetBeans y el lenguaje de programación Java. Estas herramientas también son ventajosas porque tienen buen soporte y son estables.

Conjuntamente, se cuenta con la asesoría del asesor que domina los temas de algoritmos del proyecto a presentar. Hay amplia documentación sobre los algoritmos a presentar que permiten aclarar los conceptos.

Adicionalmente, se usarán metodologías que servirán de apoyo para cumplir con los objetivos del proyecto proporcionando un conjunto de buenas prácticas.

Finalmente, cabe decir que las herramientas a utilizar son software libre o aplicaciones que ya se encuentran instaladas en los equipos de la universidad y por lo tanto son de fácil acceso. En consecuencia, no serán un obstáculo para el desarrollo normal del proyecto.

### **Viabilidad temporal**

Con respecto a la viabilidad temporal, a pesar de que el alcance del proyecto es moderado, se cuenta con el suficiente tiempo para culminar el proyecto. Se empezará la implementación del proyecto inmediatamente después de culminar el curso de Proyecto de Tesis I, obteniendo así un tiempo aproximado de siete meses para llevar a cabo el cronograma.

A continuación se muestra el cronograma del presente proyecto:

Nombre de la tarea	Inicio	Fin
<b>Planificación</b>		
<b>Elaboración del plan de proyecto</b>	Lun 16/03/15	Lun 16/03/15
<b>Ejecución</b>		
<b>Exposición 1</b> Exposición de Proyecto de Tesis 1 y plan de trabajo	Lun 23/03/15	Lun 23/03/15
<b>Exposición 2</b> Desarrollo de la función objetivo Documentación de la función objetivo	Lun 23/03/15	Lun 30/03/15
<b>Exposición 3</b> Pseudocódigo del algoritmo genético Pseudocódigo del algoritmo de búsqueda tabú	Lun 30/03/15	Lun 06/04/15
<b>Exposición 4</b> Algoritmo genético Algoritmo de búsqueda tabú	Lun 13/04/15	Lun 20/04/15
<b>Exposición 5</b> Interfaz para mostrar los resultados	Lun 20/04/15	Lun 27/04/15
<b>Exposición 6</b> Experimentación numérica	Lun 27/04/15	Lun 04/05/15
<b>Exposición Parcial</b>	Lun 04/04/15	Lun 11/05/15

<b>Exposición 8</b> Correcciones	Lun 11/05/15	Lun 18/05/15
<b>Exposición 9</b> Entrega final del borrador	Lun 18/05/15	Lun 25/05/15

### Viabilidad económica

El presente proyecto no tendrá costos debido a que las herramientas necesarias para la ejecución del proyecto son gratuitas o se encuentran disponibles en los equipos de la universidad. Este es el caso del entorno de desarrollo NetBeans, el lenguaje de programación Java y la aplicación Microsoft Excel 2010.

### Conclusión

En conclusión, se cuenta con los conocimientos necesarios para iniciar el proyecto el cual puede ser complementado con la documentación disponible. Las herramientas a usar son estables y presentan buen soporte. Adicionalmente, como se puede apreciar del cronograma, el proyecto de fin de carrera se puede llevar a cabo en los siete meses que se tienen disponibles para realizar el trabajo. Más aún, se cuenta con los recursos suficientes para llevar a cabo el proyecto. En conclusión, el proyecto es perfectamente viable.

## CAPÍTULO 2

### 2 Marco teórico

#### 2.1 Marco conceptual

##### 2.1.1 Introducción

En este apartado se presentarán conceptos que permitirán entender mejor el problema que se desea resolver y la solución propuesta. Se cubrirán conceptos de algoritmos, de modelado y simulación y del mercado de valores. Con respecto a los algoritmos, se discutirán las diferentes categorías de complejidad computacional en las que puede caer un problema. Además, se presentarán conceptos básicos de heurística y metaheurística. De modelado y simulación se presentarán las técnicas de medias móviles y de regresión polinomial por medio de mínimos cuadrados. Finalmente, se explicarán los siguientes conceptos relacionados a la bolsa de valores: las acciones, el principio de Markowitz, la media móvil y el índice general de la Bolsa de Valores de Lima (IGBVL).

##### 2.1.2 Complejidad de un problema computacional

La complejidad de un problema computacional está relacionada con los recursos que intervienen en el cálculo para resolver un problema determinado. Se pueden clasificar en dos tipos:

###### 2.1.2.1 Complejidad P

Se dice que un problema tiene complejidad P cuando un algoritmo es capaz de resolverlo en tiempo polinomial. Estos problemas son fáciles y conocidos [GAREY, JOHNSON, 2012].

###### 2.1.2.2 Complejidad NP

Se dice que un problema tiene complejidad NP (polinómico no determinístico) cuando el problema no puede ser resuelto por un algoritmo que toma un tiempo polinomial. En 1971, Cook demuestra que un subconjunto de los problemas NP (denominados NP-difíciles) son especialmente complejos dado a que no existe manera de encontrar su solución exacta debido a que la cantidad de variables es inmanejable y en la práctica no es posible trabajarlas una a una. Los problemas NP-difíciles constituyen la mayoría de los problemas relevantes en el sector industrial y financiero [RIOS MERCADO, BARD, 2000].

##### 2.1.3 “Portfolio problem” y el problema de la mochila (“knapsack problema”)

El problema de la mochila es un problema clásico de la computación que ha sido ampliamente usado para modelar otros problemas. En su versión original, se cuenta con una mochila de capacidad limitada y un conjunto de objetos de diferentes pesos y valores. El objetivo es seleccionar un subconjunto de objetos para llenar la mochila de tal manera que el valor acumulado de los objetos de la mochila sea el máximo posible.

Se puede representar el problema de la mochila matemáticamente de la siguiente forma [MARTELLO, TOTH, 1990]:

Maximizar:

$$\sum_{i=1}^n v_i x_i$$

Sujeto a:

$$\sum_{i=1}^n w_i x_i \leq W$$

Donde:

- $x_i$ : es 1 si se selecciona el  $i$ -ésimo objeto, y 0 en el caso contrario.
- $v_i$ : es el valor del  $i$ -ésimo objeto.
- $W$ : es la capacidad de la mochila.
- $w_i$ : es el peso del  $i$ -ésimo objeto.
- $n$ : es el número de objetos.

Se puede usar el problema de la mochila para modelar el problema de la selección de acciones para conformar un portafolio de inversión (“Portfolio problem”). La capacidad de la mochila representa el capital del accionista. Los objetos con los que se puede llenar la mochila simbolizan las acciones. El peso de los objetos se representa por la cotización actual de las acciones. Finalmente, el valor de los objetos es análogo a la utilidad que se espera obtener de la acción en lo que dure la inversión.

#### 2.1.4 Conceptos generales de heurísticas

Los algoritmos heurísticos aportan soluciones a problemas combinatorios de alta calidad con un costo computacional razonable. Son procedimientos que exploran un conjunto de candidatos y según los hallazgos van dirigiendo la búsqueda hacia un resultado que no es del todo óptimo, ya que no analiza todas las posibles combinaciones [SILVER, 2004].

Los algoritmos heurísticos son de fáciles de implementar y tienen un tiempo de respuesta razonable. Sin embargo, es conocido que tienen ciertos defectos pues no siempre obtienen buenas soluciones. Esto se debe a dos características de dichos algoritmos: la voracidad y la miopía [TUPIA, 2005].

- Voracidad: siempre se elige el mejor candidato de un conjunto de soluciones posibles para que forme parte del conjunto solución. Esta solución es aquella que tiene el mejor resultado al ser evaluada con la función objetivo.
- Miopía: una vez que se ha seleccionado un candidato para formar parte de la conjunto solución, esta elección no se puede modificar y no se analiza el impacto que se podría tener en la solución final.

#### 2.1.5 Conceptos generales de meta-heurísticas

El término meta-heurística fue usado por primera vez por Fred Glover, en un artículo sobre búsqueda tabú en 1986 [GLOVER, 1986]. El sufijo meta significa “más allá” o a “un nivel superior”. Las meta-heurísticas son estrategias inteligentes para mejorar y optimizar procedimientos heurísticos, generando un alto rendimiento con respecto a los algoritmos heurísticos tradicionales.

### 2.1.5.1 Algoritmo búsqueda tabú

Búsqueda tabú es un procedimiento adaptativo de mejora que puede hacer uso de otras metodologías como la programación lineal y otras heurísticas. Tiene como objetivo superar las limitaciones del óptimo local y encontrar un óptimo global.

El algoritmo usa ciertos procedimientos de aprendizaje para resolver problemas. Esto es posible ya que el algoritmo búsqueda tabú cuenta con una característica especial: la memoria adaptativa. El uso de esta memoria permite explorar la historia del proceso de resolución del problema. Es necesario, pues, crear estructuras de memoria que soporten la aplicación de este algoritmo.

A continuación se presenta el pseudocódigo del algoritmo tabú [ZAHRA, 2004]:

**Procedimiento BusquedaTabu (Instancia del problema)**

```

1. Inicializar lista tabú(T)
2. Inicializar función objetivo(f)
3. Inicializar solución de arranque(S)
4. Mientras <no se cumpla condición de parada> hacer
    2.1 Generar un conjunto V de soluciones  $S_i$  en
        la vecindad de S tales que  $f(S) \geq f(S_i)$  y
         $S_i$  no pertenezca a la lista tabú T
    2.2 Seleccionar la solución  $S^*$  de V que
        minimice la función objetivo
    2.3 Actualizar la función objetivo y la lista
        tabú
    2.2 Si ( $f(S^*) < f(\text{Mejor } S)$ ) entonces
        2.2.1 Mejor S :=  $S^*$ 
        Fin Si
    2.5 S :=  $S^*$ 
    Fin mientras
5. Retornar (Mejor S)

```

**Fin BusquedaTabu**

A continuación se presentarán algunos conceptos que permitirán entender mejor este algoritmo:

#### 1) Uso de memoria

Se debe de tener en cuenta que la memoria adaptativa usada en búsqueda tabú hace referencia a cuatro dimensiones principales: el carácter reciente, la frecuencia, la calidad y el impacto.

- Carácter reciente: se hace un seguimiento a las características que han cambiado recientemente.
- Frecuencia: se hace un seguimiento a las características que han cambiado con mayor frecuencia.
- Calidad: se distingue buenas soluciones de entre las que fueron evaluadas durante la búsqueda.

- Impacto: se estudia el efecto que podría tener el proceso de búsqueda sobre la calidad de una solución.

Otro factor clave dentro de la memoria adaptativa del algoritmo tabú es conseguir un equilibrio entre la intensificación de la búsqueda y la diversificación. La intensificación se basa en modificar las reglas de elección de movimientos para que se favorezcan los movimientos que han tenido buen desempeño históricamente. La diversificación, por el contrario, incentiva a explorar regiones que no fueron visitadas cuando se generaron las soluciones anteriores [GLOVER, LAGUNA, 1997].

### 1) Búsqueda por entorno

Cada solución tiene un conjunto asociado de vecinos al que se conoce como entorno. En el algoritmo búsqueda tabú, la búsqueda por entorno busca encontrar un óptimo local en el entorno: aquella solución que sea mejor o igual de buena que todas las soluciones de su entorno. Sin embargo, en la mayoría de los casos el óptimo local no será un óptimo global. Para superar esta limitación, se usa un enfoque aleatorio. Es decir, se comienza el método con diferentes soluciones seleccionadas aleatoriamente y se ejecuta el método múltiples veces. De esta manera, se incrementa la posibilidad de encontrar el óptimo global.

### 2) Estrategias basadas en memorias a corto y largo plazo

Como se ha mencionado previamente, el algoritmo búsqueda tabú usa estructuras de memoria para mantener el historial de movimientos en el proceso de búsqueda. Estas estructuras de memoria pueden ser de corto plazo o de largo plazo.

Cuando se utilizan estrategias a corto plazo, se construyen estructuras de memoria volátiles conocidas como listas tabú. Las listas tabúes almacenan los movimientos realizados en cada iteración. Dichos movimientos no se podrán repetir hasta que se cumpla un cierto número de ciclos.

Cuando se usan estrategias a largo plazo se arma una estructura que graba la frecuencia con la que se realizan ciertos movimientos. A diferencia de las estrategias a corto plazo, la estructura de memoria construida en las estrategias a largo plazo se mantiene durante toda la ejecución a modo de historial.

### 3) Criterios de aplicación

Los criterios de aplicación se usan en el algoritmo búsqueda tabú para determinar cuándo pueden ser reemplazadas las restricciones tabú. Es muy importante usar estos criterios apropiadamente para garantizar los mejores niveles de ejecución del algoritmo.

La base principal para los criterios de aspiración es la influencia: el grado de cambio introducido en la estructura de la solución. Los movimientos de menor influencia normalmente pueden ser tolerados ya que el cambio en la estructura que generan es insignificante [GLOVER, MELIAN, 2003].

### 2.1.5.2 Algoritmo genético

Los algoritmos genéticos están basados en la evolución biológica. Se simula el proceso de selección natural en el que sobrevive el más fuerte: aquel que maximice la función fitness.

A continuación se presenta un esquema general para la implementación de un algoritmo genético [MATRI, 2007]:

**Procedimiento AlgoritmoGenético (Instancia del problema)**

1. Generar soluciones aleatorias (Instancia)
2. Mejorar soluciones
3. Mientras <no se cumpla condición de parada> hacer
  - 3.1 Evaluar soluciones
  - 3.2 Calcular probabilidad de supervivencia de las soluciones
  - 3.4 Combinar soluciones
  - 3.5 Mutar soluciones
 Fin mientras
4. Retornar (Mejor S)

**Fin AlgoritmoGenético**

La complejidad de un algoritmo genético está en proponer la representación cromosómática y la función fitness.

A lo largo de la historia ha habido diversos trabajos que usan algoritmos genéticos para resolver el problema del portafolio de inversiones. Por ejemplo, en 2012, Sefiane propone el siguiente algoritmo genético para resolver el problema del portafolio.

Al ser puesto en práctica, el algoritmo de Sefiane obtuvo una rentabilidad positiva en un tiempo aceptable en todas sus ejecuciones [SEFIANE, 2012].

### 2.1.6 Conceptos generales de modelado y simulación

#### 2.1.6.1 Media móvil

La media móvil es un cálculo estadístico que permite crear una serie de promedios en base a un conjunto de datos [CHOU, 1970]. Es decir, las medias móviles son una lista de números, donde cada número representa el promedio de un subconjunto de los datos que se están estudiando.

Existen dos tipos principales de medias móviles:

- Media móvil simple: es la media aritmética de los  $n$  últimos datos. Mientras mayor sea el valor de  $n$ , los datos antiguos tendrán más influencia.
- Media móvil ponderada: es similar a la media simple, pero los datos son multiplicados por factores que le dan determinados pesos a determinados datos. En el contexto del mercado bursátil, se suelen usar medias móviles

ponderadas para darle mayor peso a las cotizaciones más recientes y restarle importancia a las cotizaciones más antiguas.

La media móvil es una herramienta muy valiosa en el mercado bursátil que puede ser aplicada de distintas maneras. Por ejemplo, una media móvil simple a largo plazo es un buen estimado del valor real de una acción. Si se calcula la media móvil simple a corto plazo (que refleja la cotización actual) y se compara con la media móvil a largo plazo, se puede determinar si una acción está subvalorada o sobrevalorada [CHOU, 1970].

En el siguiente trabajo de fin de carrera, se sacará la media móvil simple de la rentabilidad de las acciones en los últimos tres meses para calcular sus beneficios estimados.

#### **2.1.6.2 Desviación estándar**

Para un conjunto de observaciones, la desviación estándar es una medición estadística que refleja la dispersión promedio de estas observaciones alrededor de su media [SAEED 2000].

Se calcula de la siguiente manera:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$$

Donde:

- $\sigma$ : desviación estándar de la muestra
- $x_i$ : cada elemento de la muestra
- $\mu$ : media de la muestra
- N: número de observaciones

En este trabajo de fin de carrera, se usará la desviación estándar de la rentabilidad de una acción en los últimos tres meses como medida del nivel de riesgo asociado a dicha acción. La desviación estándar indica qué tan volátil ha sido la rentabilidad alrededor de su media.

#### **2.1.7 Conceptos generales del mercado de valores**

##### **2.1.7.1 Acciones**

Una acción es un título emitido por una sociedad que representa el valor de una parte del capital social. Las acciones confieren a su titular derechos políticos (como votar en la junta de accionistas) y económicos (como participar de la repartición de dividendos de la empresa). Usualmente, las acciones se pueden negociar sin ninguna restricción [MORA ENGUIDANOS, 2008].

#### 2.1.7.2 Cotización

En el contexto del mercado de valores, una cotización es el precio de una acción de una compañía. Se obtiene de traer todos los flujos de dinero de la compañía al presente y dividir el resultado entre el número de acciones. La cotización de una acción, pues, está altamente relacionada a la expectativa de ingresos futuros y a la capacidad de pagar dividendos de la empresa [EHRHARDT, 2010].

#### 2.1.7.3 Riesgo

El riesgo combina la probabilidad de que ocurra un evento negativo y el daño que dicho evento causaría. En el contexto del mercado de valores, el riesgo abarca la posibilidad de que cambie la posición financiera de las acciones y entonces los resultados esperados no ocurran [GUILLEN, 2008].

#### 2.1.7.4 Principio de Markowitz

Harry Markowitz fue un reconocido economista americano que popularizó la teoría moderna del portafolio. El principio de Markowitz indica que el inversor debe de obtener la máxima rentabilidad con el menor riesgo posible. Esto se consigue teniendo un portafolio diversificado en el que intervengan un gran número de acciones. De esta manera, el impacto del riesgo en caso se materialice será atenuado [RUBINSTEIN, 2002].

#### 2.1.7.5 Índice General de la Bolsa de Valores de Lima (IGBVL)

El IGBVL es un indicador que mide el comportamiento del mercado bursátil y sirve para determinar el rendimiento alcanzado por las diversas acciones que forman parte de la Bolsa de Valores de Lima en un determinado periodo de tiempo. El IGBVL se calcula tomando en cuenta las acciones más significativas del mercado bursátil, asignándoles pesos según el volumen de transacciones. Al 18 de setiembre de 2014, el IGBVL está compuesto de las siguientes acciones [BVL, 2014]:

- Alicorp
- Alturas Minerales
- Agroindustrial Pomalca
- Atacocha
- Austral Group
- Banco Continental
- Candente
- Agroindustrial Casa Grande
- Cementos Lima
- Cementos Pacasmayo
- Cerro Verde
- Corporación Aceros Arequipa
- Corporación Lindley
- Credicorp
- Edegel
- El Brocal
- Ferreyros
- Gold Fields – La Cima
- Graña y Montero
- Intergroup Financial Services
- Luz del Sur
- Maple Energy
- Milpo
- Minera IRL
- Minsur
- Relapasa
- Rio Alto Mining
- Scotiabank Perú
- SiderPerú
- Simsa
- Southern Copper Corporation
- Telefónica
- Volcan

## 2.2 Conclusiones sobre el marco conceptual

Para concluir, el marco conceptual describe en detalle los conceptos necesarios para entendimiento de las ideas expresadas a lo largo de este documento.

La sección de complejidad computacional es importante debido a que permite explicar por qué el problema que se intenta resolver es de clase NP-difícil y aclara por qué se puede usar el “portfolio problem” para representar el problema. El apartado de conceptos generales de heurística y meta-heurística justifica el uso de un algoritmo meta-heurístico para resolver el problema y explica los beneficios de utilizar enfoques meta-heurísticos en lugar de heurísticos. La sección de modelado y simulación permite aclarar las técnicas que serán utilizadas para modelar el comportamiento de las acciones en el futuro y predecir las cotizaciones futuras. Finalmente, el apartado de conceptos generales del mercado de valores describe los diversos factores y variables que serán tomados en cuenta al momento de desarrollar el algoritmo.



## CAPÍTULO 3

### 3 Estado del arte

#### 3.1 Introducción

A lo largo de la historia, el problema de inversiones en la bolsa de valores ha sido atacado de diversas maneras. En el pasado, existían únicamente dos técnicas para decidir qué acciones comprar: el análisis técnico y el análisis fundamental. Con el desarrollo de la tecnología, aparecieron nuevas estrategias inteligentes para resolver el problema que aprovechan el poder de las computadoras: redes neuronales, algoritmos genéticos, lógica fuzzy y sistemas expertos. Hoy en día es posible adquirir diversos software comerciales que aplican estas estrategias.

#### 3.2 Objetivos de la revisión del estado del arte

El objetivo de la revisión del estado del arte es estudiar las diferentes estrategias que se han usado para atacar el problema de las inversiones bursátiles a lo largo de la historia.

Se explicarán las diferentes fortalezas y debilidades de cada alternativa de solución para luego detallar cómo el proyecto de fin de carrera aportará valor en comparación a las demás soluciones existentes.

#### 3.3 Método usado en la revisión del estado del arte

Se utilizó el método tradicional para la revisión del estado del arte.

#### 3.4 Métodos Convencionales

Para realizar predicciones del mercado bursátil, las dos técnicas convencionales más importantes son: el análisis técnico y el análisis fundamental.

##### 3.4.1 Análisis técnico

El análisis técnico consiste en recolectar datos cuantitativos de las acciones en diversos instantes de tiempo, graficarlos y analizar los gráficos producidos. El análisis técnico se basa en la hipótesis de que el efecto de las noticias se ve reflejado en los precios de las acciones. Por esta razón, el análisis técnico ignora las noticias. Se enfoca en identificar patrones y predecir tendencias en el mercado bursátil a través de gráficos. Este método es limitado debido a que los gráficos solo muestran eventos y no las causas de dichos eventos [KROHA, 2004].

##### 3.4.2 Análisis fundamental

El análisis fundamental investiga los factores que pueden afectar la oferta y la demanda. El objetivo es recolectar e interpretar información para actuar antes de que las noticias tengan un impacto en el valor de las acciones. Se basa en el supuesto de que existe un retraso entre un evento y la correspondiente respuesta del mercado bursátil. Los documentos más usados en el análisis fundamental son aquellos que las compañías están obligadas a publicar con frecuencia: estados financieros, memorias anuales, reportes anuales, reportes de auditorías, entre otros. A diferencia del análisis técnico, en el análisis fundamental sí estudia los factores que pueden causar cambios en el mercado bursátil. Sin embargo, tiene la debilidad de que las decisiones tienen un carácter subjetivo dado a que se toman según el juicio experto del analista [KROHA, 2004].

### **3.5 Proyectos de investigación para resolver el problema de selección de portafolio**

En este apartado se presentarán algunos proyectos de investigación que tienen como objetivo solucionar el problema de toma de decisiones en las inversiones en la bolsa de valores.

#### **3.5.1 Toma de decisiones fuzzy en el problema del portafolio**

La lógica fuzzy involucra cuantificar información cualitativa. Se usan funciones que transforman información cualitativa en números reales que van de 0 a 1.

La lógica fuzzy es altamente aplicable en el problema de las inversiones en el mercado bursátil dado a que hay una gran cantidad de información cualitativa que puede ser utilizada para explicar el comportamiento de las cotizaciones de las acciones. Los sistemas basados en lógica fuzzy, pues, permiten aprovechar esta información.

Hay diversos trabajos que proponen usar lógica fuzzy para evaluar el retorno y el riesgo de un portafolio de inversión. Por ejemplo, en el trabajo “Toma de decisiones fuzzy en el problema de portafolio” [LIAN, LI, 2010], los autores usan la opinión de expertos para crear funciones fuzzy que permiten estimar el riesgo asociado a una acción.

Cabe decir que los sistemas basados en lógica fuzzy tienen un grado de subjetividad importante debido a que la asignación de los valores numéricos a la información cualitativa depende del criterio del experto.

#### **3.5.2 Algoritmo genético para resolver el problema de selección de proyectos**

Este proyecto propone el uso de un algoritmo genético para el problema del portafolio aplicado a la selección de proyectos de Tecnologías de Información. El algoritmo busca optimizar la ganancia, la factibilidad de ejecución, los costos y los riesgos al seleccionar los proyectos que pasan a formar parte del portafolio.

Las variables que considera el algoritmo para la selección de proyectos son las siguientes: el número de proyectos, el costo de cada proyecto, las dependencias de los proyectos, los posibles riesgos de cada proyecto y el beneficio estimado de cada proyecto. Las variables descritas pueden ser operadas por medio de operadores genéticos de modo que en cada iteración se vayan generando mejores soluciones [CUEVA, GUANIRA, TUPIA, 2013].

#### **3.5.3 Un algoritmo para el problema de optimización de portafolios**

Este proyecto trata de resolver el problema del portafolio aplicado a la bolsa de valores de Corea. El proyecto usa un modelo de programación lineal denominado “desviación de la media absoluta”. En este modelo, se mide el riesgo asociado a una acción como una desviación absoluta. Para cada mes, se resuelve el modelo de programación lineal para decidir qué acciones comprar y vender en ese momento.

El modelo fue probado con 600 acciones que fueron cotizadas mensualmente entre julio 1998 y junio de 2000. Los resultados del algoritmo fueron exitosos, obteniendo rentabilidades entre 1.25% y 10% sobre la inversión [KIM, 2005].

### 3.5.4 Selección de portafolio usando algoritmos genéticos

Este proyecto ataca el problema de selección de portafolio de inversión por medio de un algoritmo genético. La función objetivo de este algoritmo busca maximizar el retorno y minimizar el riesgo del portafolio.

El proyecto trabaja con cinco acciones que son estudiadas por un periodo de cinco años (de 2007 a 2011). El algoritmo logra una rentabilidad de 0.2222 y logra mantener el riesgo controlado a 0.0325 [SEFIANE, 2012].

## 3.6 Otros proyectos de investigación aplicados a la Bolsa de Valores

### 3.6.1 Tradinnova

Tradinnova es un proyecto que emplea un algoritmo heurístico para simular el comportamiento de un inversor inteligente en el mercado de valores. Tradinnova, al igual que a mayoría de las alternativas de solución que hay en la actualidad, predice el comportamiento de las acciones en el futuro y recomienda la adquisición de una cartera de valores.

Sin embargo, Tradinnova tiene la peculiaridad de que además ayuda en la toma de decisiones en el momento de la venta de las acciones ya adquiridas. Para lograr esto, Tradinnova genera para cada acción un rango de precios de venta, y vende las “peores” acciones según funciones fitness que miden rentabilidad, volatilidad y riesgo. Cada cierto tiempo, los precios de ventas se van actualizando [CASANOVA, 2006].

### 3.6.2 Predicción de precios en el mercado de acciones usando redes neuronales artificiales

Las redes neuronales son un sistema de estructura de datos y programas que aproximan el funcionamiento del cerebro humano. Involucra un gran número de procesadores operando en paralelo, todos con una esfera de conocimiento y con acceso a la data en la memoria local. Tradicionalmente, una red neuronal primero debe de ser “entrenada” introduciéndole un gran volumen de data y reglas sobre las relaciones entre la data. A partir de esta información, la red neuronal sabe cómo debe de reaccionar a estímulo externo. En el contexto de las inversiones bursátiles, Zabir pasa a su red neuronal el histórico de las cotizaciones bursátiles y a partir de esa data la red neuronal es capaz de predecir las cotizaciones futuras [ZABIR, 2011].

### 3.6.3 Diseñando agentes seguros, rentables y automatizados usando algoritmos evolutivos

Este proyecto se basa en el principio de que existen diversas reglas que los inversionistas pueden usar para razonar sobre patrones en las cotizaciones de las acciones. Sin embargo, debido a la naturaleza volátil del mercado de valores, estas reglas no son aplicables en todo momento. Los autores de esta investigación proponen aprovechar el poder de las computadoras para combinar estas reglas de manera inteligente y así crear un agente autónomo.

El proyecto es un sistema experto en combinación con un algoritmo evolutivo. Primero, se alimenta al sistema experto con reglas recolectadas de entrevistas a inversionistas experimentados. Cada regla toma una decisión con respecto a una acción: comprar, hacer nada o vender. Para tomar la decisión final, el algoritmo analiza los veredictos de las reglas ponderándolos con un peso.

El peso de cada regla se asigna usando la data histórica: se asignan los pesos que hubieran maximizado la rentabilidad obtenida. Conforme pasa el tiempo y el entorno del mercado de valores cambia, el algoritmo actualiza los pesos que le asigna a cada regla [SUBRAMANIAN, RAMAMOORTHY, 2006].

### **3.6.4 Un algoritmo genético efectivo para el problema de optimización del portafolio multiobjetivo**

Los algoritmos genéticos están basados en la evolución biológica. Se simula el proceso de selección natural en el que sobrevive el más fuerte: aquel que maximice la función fitness.

Cada solución posible se denomina “cromosoma”. Se parte de una población inicial de soluciones (denominada “generación”) que se genera de manera aleatoria. Para generar más soluciones, se le aplican operadores genéticos como lo son el casamiento y la mutación a la generación actual y las mejores soluciones pasan a formar parte de la generación futura.

La complejidad de un algoritmo genético está en proponer la representación cromosómática y la función fitness. En su trabajo, Lin representa al cromosoma como un arreglo de números reales donde cada número representa la proporción del capital de inversión a invertir en una acción determinada. La función fitness de una solución se calcula en base a la utilidad de cada acción del portafolio en los últimos tres meses y al riesgo estimado [LIN, 2007].

## **3.7 Productos comerciales para resolver el problema**

Para ayudar a los inversionistas a predecir el comportamiento del mercado bursátil existen algunas soluciones software que se detallarán a continuación:

### **3.7.1 Hotstocked Precision**

Hotstocked Precision es un software empaquetado que ofrece diversas funcionalidades relacionadas al mercado bursátil de Wall Street. Además de mostrar el histórico de las cotizaciones de las acciones de Wall Street, Hotstocked Precision es capaz de proyectar el comportamiento de las acciones en el futuro.

Hotstocked Precision trabaja bajo el supuesto de que las tendencias se repiten. Al estudiar una acción, el software busca otras acciones que se han comportado de manera similar en el pasado y utiliza esa data para realizar las predicciones [HOTSTOCKED PRECISION, 2014].

### **3.7.2 Stock Signal Pro**

Stock Signal Pro es una herramienta cuyo objetivo es ayudar al usuario a elegir el mejor momento para comprar o vender una acción. También trabaja bajo el supuesto de que la cotización de las acciones no es completamente aleatoria: se sigue un patrón definido.

Stock Signal Pro trabaja con un “watchlist” al cual el usuario le puede agregar o quitar acciones. Las acciones que pertenecen a dicho “watchlist” son analizadas a demanda del usuario. Si el sistema identifica que hay una buena oportunidad para comprar o vender una acción del “watchlist” alerta al usuario [STOCK SIGNAL PRO, 2014].



Diagrama 3.7.2.1: Reporte de Stock Signal Pro

### 3.7.3 PPO

PPO (Project Portfolio Optimization) es una herramienta que busca resolver el problema del portafolio con el fin de seleccionar una cartera de proyecto que maximice la rentabilidad obtenida.

El usuario debe de ingresar en un formulario los costos y los beneficios de cada proyecto. Asimismo, es importante incluir las metas de la organización y los recursos con los que se cuenta. De este modo, se priorizan los proyectos que permitan cumplir con los objetivos de la organización. El programa permite generar reportes que apoyen a la toma de decisiones para maximizar la ganancia de la empresa [PWC, 2013].

### 3.7.4 Project Portfolio Office

Esta herramienta permite apoyar en temas de planificación, gestión y ejecución de portafolios de proyectos. Este software ha sido galardonado con diversos premios en software de gestión de portafolios.

Project Portfolio Office permite a los usuarios priorizar los proyectos según reglas del negocio. Asimismo, el software permite realizar seguimiento al portafolio del proyecto y generar reportes del estado de los proyectos [PROJECT PORTFOLIO OFFICE, 2014].

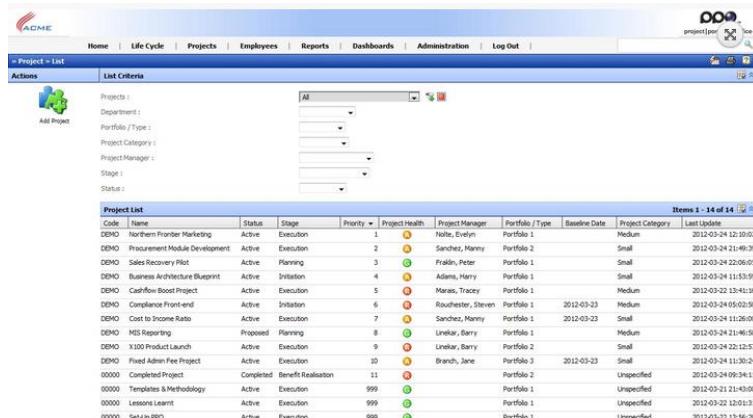


Diagrama 3.7.4.1: Pantalla de Project Portfolio Office

### 3.8 Conclusiones sobre el estado del arte

En el pasado, existían únicamente dos estrategias para tomar decisiones relacionadas al mercado bursátil: el análisis técnico y el análisis fundamental. Sin embargo, como ya se mencionó en el apartado anterior, ambas técnicas presentaban debilidades importantes. El análisis técnico se limitaba a estudiar tendencias en la data histórica y era incapaz de explicar las causas de dichas tendencias. Mientras que el análisis técnico dependía mucho en el juicio experto del analista y por lo tanto las decisiones que se tomaban tenían una naturaleza subjetiva.

Con el desarrollo de las computadoras, en la actualidad se ha popularizado el uso de diversos sistemas inteligentes para apoyar en las inversiones bursátiles: redes neuronales, algoritmos genéticos y sistemas expertos. Los resultados han sido muy prometedores: varios de los proyectos han tenido rentabilidades muy superiores a los índices bursátiles de los mercados en los que fueron implementados.

Cabe decir que luego de una investigación exhaustiva solo se han encontrado agentes inteligentes que emplean algoritmos heurísticos. Como se mencionó anteriormente, las soluciones heurísticas tienen limitaciones debido a la miopía y a la voracidad. El desarrollar un algoritmo meta-heurístico que supere estas deficiencias representa una oportunidad importante para mejorar el estado del arte del problema de la inversión bursátil.

## CAPÍTULO 4

### 4 Definición de la función objetivo

#### 4.1 Función objetivo

Como los algoritmos pretenden encontrar el mejor portafolio posible dentro de un conjunto de posibles candidatos, se requiere de una función objetivo. Esta función objetivo establece el criterio de selección, la cual está dada por:

$$\text{Max} \sum_{i=1}^n \frac{B_i * W_i}{R_i}$$

Donde:

- $B_i$ : Beneficio estimado de la acción i.
- $R_i$ : Riesgo estimado de la acción i.
- $W_i$ : Peso de la acción i (proporción del capital a invertir en la acción i)
- n: número de acciones que considera el algoritmo

Como se puede observar, al ser la función objetivo de maximización, se busca:

- Obtener el mayor beneficio  $B_i$ .
- Reducir el riesgo  $R_i$ .

Finalmente, se debe tener en cuenta las siguientes restricciones:

- Capital disponible: se cuenta con capital limitado para la compra de acciones el cual no puede ser excedido.
- Acciones enteras: solo se pueden comprar y vender acciones enteras. No está permitido realizar transacciones con fracciones de acciones.

#### 4.1.1 Beneficio estimado de una acción

En este trabajo de fin de carrera, se considerará como más atractivas a las acciones que han tenido un desempeño sobresaliente en el corto plazo. Se definirá el beneficio de una acción i como el incremento porcentual que ha tenido el precio de esa acción en los últimos tres meses. Es decir, por la utilidad de los últimos tres meses. Se ha elegido este periodo de tiempo debido a que es el que mejor representa la rentabilidad a corto plazo esperada de una acción [LIN, 2007]:

$$B_i = \frac{P_i^T - P_i^{T-3}}{P_i^{T-3}}$$

Donde:

- $B_i$ : Beneficio estimado de la acción i.
- $P_i^t$ : Precio de la acción i en el tiempo t.

#### 4.1.2 Riesgo estimado de una acción

El apartado 4.1.1 muestra el beneficio de una acción i, el cual representa la rentabilidad promedio de una acción en los últimos tres meses. Sin embargo, también es importante considerar qué tan volátil ha sido esta rentabilidad para saber cuán probable es desviarse de dicho promedio en el futuro. Este es el concepto de riesgo que será usado en este

trabajo. Se considerará como el riesgo de una acción i a la desviación estándar de su rentabilidad en los últimos tres meses.

$$R_i = \sigma_i$$

Donde:

- $R_i$ : Riesgo estimado de la acción i.
- $\sigma_i$ : Desviación estándar de la rentabilidad de la acción i en los últimos 3 meses.

#### 4.1.3 Peso de una acción

El peso de una acción en un portafolio está dado por la proporción del capital total a invertir en la acción i.

$$W_i = \frac{P_i^t * Q_i}{C}$$

Donde:

- $W_i$ : Peso de la acción i.
- $P_i^t$ : Precio de la acción i en el tiempo t.
- $Q_i$ : Cantidad de acciones i a adquirir.
- $C$ : Capital total a invertir.

### 4.2 Estructuras de Datos

En el siguiente apartado se presentarán las principales estructuras de datos a usar en los algoritmos que buscan resolver el problema del portafolio aplicado a la selección de acciones a invertir de una bolsa de valores. La complejidad del problema recae en maximizar la rentabilidad esperada del portafolio de acciones mientras se controla el riesgo.

Se presentarán las estructuras de datos principales y otras estructuras auxiliares que permitirán dar soporte a las operaciones y cálculos que se harán sobre las estructuras principales.

Cabe decir que las estructuras de datos presentadas en este apartado serán usados en ambos algoritmos: genético y búsqueda tabú. No obstante, se usarán estructuras de datos adicionales particulares a cada algoritmo que serán presentadas en los pseudocódigos de los mismos.

#### 4.2.1 Variables

A continuación se presentan las variables que son necesarias para llevar a cabo la selección de acciones para formar el portafolio de inversión:

Variable	Representación
Cantidad de acciones que se usarán en los algoritmos	N
Capital inicial a invertir	Capital
Precio unitario de la acción i en nuevos soles	Precio <sub>i</sub>
Beneficio unitario esperado de la inversión de la acción i	Beneficio <sub>i</sub>
Riesgo estimado de la acción i	Riesgo <sub>i</sub>
Proporción del capital a invertir en la acción i	Peso <sub>i</sub>

Tabla 4.1: Variables a usar en los algoritmos

#### 4.2.2 Clase Accion

La clase *Accion* es una estructura de datos que se usará para almacenar la información de las acciones que es relevante para los algoritmos. Dicha estructura es una clase que almacena los siguientes atributos de una acción: el nombre, el precio a la fecha, el beneficio de los últimos 3 meses, y el riesgo de los últimos 3 meses. Además, debido a que es un valor que se usa de manera frecuente, se almacena la proporción del capital a invertir que representa una acción ( $Precio_i/Capital$ ) (denominada incremento).

Se representa de la siguiente manera:

```
Clase Accion{
    Cadena nombre
    Real precio
    Real beneficio
    Real riesgo
    Real incremento
}
```

#### 4.2.3 Arreglo acciones

El arreglo *acciones* es una estructura de datos que se usará para almacenar la información de todas las acciones con las que trabajará el algoritmo. Dicha estructura es una lista de objetos de la clase Accion. Se representa de la siguiente manera:

Lista<Accion> acciones

A continuación se presenta un ejemplo del arreglo *acciones*:

Código acción	1	2	...	N
Nombre	Atacocha	BBVA	...	Volcan
Precio <sub>i</sub>	5.23	1.06	...	0.52
Beneficio <sub>i</sub>	0.25	0.12	...	0.10
Riesgo <sub>i</sub>	0.14	0.23	...	0.30
Incremento <sub>i</sub>	0.00523	0.00106	...	0.0052

Tabla 4.2: Arreglo acciones

#### 4.2.4 Cromosoma

El cromosoma es la estructura de datos que se usará para representar las posibles soluciones al problema de selección de un portafolio de inversión. Dicha estructura es una clase con las siguientes propiedades: una lista de  $n$  números reales denominada *pesos* y una lista de  $n$  enteros denominada *cantidades* (donde  $n$  es el número de acciones que se usarán en los algoritmos). La lista de *pesos* contiene la proporción del capital a invertir en cada acción ( $Peso_i$ ) según la solución propuesta. La lista de *cantidades* contiene la cantidad de acciones a comprar según la solución propuesta. El  $Peso_i$  de cada acción  $i$  podría calcularse multiplicando la  $Cantidad_i$  por el  $Incremento_i$  de la acción (que se obtiene del arreglo *acciones* presentado en el apartado 4.2.3).

El cromosoma se representa de la siguiente manera:

Clase Cromosoma{

    Lista<Real> peso;

    Lista<Enteros> cantidad;

}

A continuación se presenta un ejemplo de un cromosoma:

Código acción	1	2	...	N
Peso <sub>i</sub>	0.523	0.0212	...	0.26
Cantidad <sub>i</sub>	100	20	...	50

Tabla 4.3: Cromosoma

## CAPÍTULO 5

### 5 Diseño del algoritmo genético para el problema de selección de portafolio aplicado a inversiones en la bolsa de valores

#### 5.1 Variables y parámetros

La siguiente tabla muestra los parámetros que recibirá el algoritmo genético:

Variable/Parámetro	Representación
Ruta del archivo con las cotizaciones de las acciones de los últimos tres meses	Archivo
Capital disponible para invertir	Capital
Número de iteraciones a realizar	Iteraciones
Tamaño de la población en cada generación	tamanoPoblacion
Número de mutaciones en cada iteración	numeroMutaciones

Tabla 5.1: Variables a usar en los algoritmos

#### 5.2 Pseudocódigo del algoritmo genético

El siguiente algoritmo ha sido adaptado del algoritmo propuesto por Lin en el trabajo “An Effective Decision-Based Genetic Algorithm Approach to Multiobjective Portfolio Optimimzation Problem” [LIN, 2007]. Dicho algoritmo se aplica a la Bolsa de Taiwán. Las adaptaciones del algoritmo se realizaron con el fin de que el algoritmo genético pueda presentar soluciones con la representación propuesta en el apartado 4.2.4:

#### Inicio Algoritmo Genético (archivo, capital, iteraciones, tamanoPoblacion, numeroMutaciones)

1. acciones = inicializarAcciones(archivo, capital)
2. pob = inicializarPoblacion(tamanoPoblacion)
3. mejorSol = mejorElemento(pob)
4. Para i desde 1 hasta iteraciones hacer
  - a. nuevaPob = casamiento(pob)
  - b. pob = seleccion(pob, nuevaPob, tamanoPoblacion)
  - c. Si (fitness(mejorSol, acciones) < fitness(mejorElemento(pob, acciones)))
   
entonces
    - i. mejorSol = mejorElemento(pob)
  - fSi
  - d. pob = mutacion(pob, numeroMutaciones)
- fPara
5. retornar(mejorSol)

#### Fin Algoritmo Genético

Se explican las siguientes líneas:

- Línea 1: se inicializa la lista de acciones a partir de la información en el archivo ingresado. Se explicará este procedimiento más adelante.

- Línea 2: se inicializa la población de cromosomas con *tamanoPoblacion* cromosomas generadas de manera aleatoria.
- Línea 3: se inicializa la mejor solución con el mejor elemento de la población. El mejor elemento es el cromosoma de la población que maximiza la función objetivo.
- Línea 4: se itera hasta cumplir la condición de parada. La condición de parada se cumple cuando se realizan un número de iteraciones.
  - Línea 4.a: se genera la nueva población casando los elementos de la población antigua.
  - Línea 4.b: entre la población antigua y la población nueva, se seleccionan los *tamanoPoblacion* mejores elementos para formar la nueva generación
  - Línea 4.c: si el mejor elemento de la población tiene un fitness mayor que la mejor solución, se actualiza la mejor solución. El fitness de una solución está determinado por la función objetivo (revisar acápite 4.1), se selecciona el cromosoma de la población que maximice la función objetivo.
  - Línea 4.d: se muta la población.
- Línea 5: se retorna la mejor solución.

### 5.2.1 Pseudocódigo del procedimiento de inicializar el arreglo de acciones

A continuación se presenta el pseudocódigo para inicializar el arreglo de acciones:

#### Inicio inicializarAcciones(archivo, capital)

1. archivoAbierto = abrirArchivo(archivo)
2. acciones = leerNombres(archivoAbierto)
3. Para cada (accion en acciones) hacer
  - a. accion.precio = leerPrecio(archivoAbierto)
  - b. accion.beneficio = calcularBeneficio(accion, archivoAbierto)
  - c. accion.riesgo = calcularDesviacionEstandar(accion, archivoAbierto)
  - d. accion.incremento = accion.precio/capital
4. Para i desde acciones.tamano() hasta 1 hacer
  - a. Si (acciones(i).beneficio <= 0) entonces
    - i. acciones.remover(i)
- fSi
- fPara
5. retornar (acciones)

#### Fin inicializarAcciones

Se explican las siguientes líneas:

- Línea 1: se abre el archivo.

- Línea 2: se leen los nombres de las acciones del archivo y se colocan en el arreglo acciones.
- Línea 3: se itera por cada acción encontrada.
  - Línea 3.a: por cada acción, se lee el precio actual de la acción.
  - Línea 3.b: por cada acción, se calcula la utilidad promedio de la acción en los últimos tres meses.
  - Línea 3.c: por cada acción, se calcula la desviación estándar de las utilidades a la utilidad promedio en los últimos tres meses.
  - Línea 3.d: por cada acción, se calcula la proporción de capital que representa la compra de una unidad de la acción (precio/capital).
- Línea 4: se examina todo el arreglo de acciones, y se eliminan todas aquellas que tengan beneficio negativo o nulo. De esta manera sólo se trabajará con acciones que beneficiarán el portafolio.
- Línea 5: se retorna la lista de acciones.

### 5.2.2 Pseudocódigo del procedimiento de casamiento

A continuación se presenta el pseudocódigo del procedimiento de casamiento:

#### Inicio casamiento(pob)

1. nuevaPob = {}
2. Mientras (tamano(nuevaPob) < n) hacer
  - a. padre = obtenerCromosoma(Pob)
  - b. madre = obtenerCromosoma(Pob)
  - c. puntoCorte = obtenerPuntoCorte(padre)
  - d. hijo1 = casar(padre, madre, puntoCorte)
  - e. hijo2 = casar(madre, padre, puntoCorte)
  - f. Si (no esAbominacion(hijo1, pob, nuevaPob))
    - i. nuevaPob = agregar(hijo1, nuevaPob)
- fSi
- g. Si (no esAbominacion(hijo1, pob, nuevaPob))
  - i. nuevaPob = agregar(hijo2, nuevaPob)

fSi

fMientras

3. retornar (nuevaPob)

#### Fin casamiento

Se explican las siguientes líneas:

- Línea 1: se inicializa la nueva población como un arreglo vacío.
- Línea 2: se itera hasta que la nueva población tenga n elementos.
  - Línea 2.a-2.b: se obtienen el padre y la madre extrayendo dos cromosomas de manera aleatoria de la población.
  - Línea 2.c: se selecciona la posición del cromosoma que será usado como punto de corte.

- Línea 2.d-2.e: se aplica la operación de casamiento a los padres con el punto de corte, generando dos hijos.
- Línea 2.f-2.g: se examinan los hijos para ver si son abominaciones (el hijo es abominación si ya hay un cromosoma idéntico a él en la población). Si no lo son, se agregan los hijos a la nueva población.
- Línea 3: se retorna la nueva población.

### 5.2.3 Pseudocódigo del procedimiento de mutación

A continuación se presenta el pseudocódigo del procedimiento de mutación:

#### Inicio mutación(**pob, numeroMutaciones**)

1. Para (i desde 1 hasta numeroMutaciones) hacer
  - a. cromosoma = obtenerCromosoma(pob)
  - b. cromosomaMutada = mutar(cromosoma)
  - c. Si(no esAbominacion(cromosomaMutada)) entonces
    - i. pob = reemplazar(cromosoma, cromosomaMutada, pob)
- fSi
- fPara
2. retornar (pob)

#### Fin mutación

Se explican las siguientes líneas:

- Línea 1: se itera *numeroMutaciones* veces.
  - Línea 1.a: se obtiene un cromosoma de manera aleatoria de la población. Este será el cromosoma a mutar
  - Línea 1.b: se obtiene el cromosoma mutado mutando el cromosoma original. En el proceso de mutación, se seleccionan dos genes del cromosoma y se intercambian de posición.
  - Línea 1.c se verifica que el cromosoma mutado no sea una abominación. Si no lo es, se reemplaza el cromosoma original por el cromosoma mutado en la población.
- Línea 2: se retorna la población mutada.

### 5.2.4 Pseudocódigo de la función fitness

A continuación se presenta el pseudocódigo de la función fitness:

**Inicio fitness(cromosoma, acciones)**

1. sumatoria = 0
2. Para i desde 1 hasta acciones.tamano() hacer
  - a. sumatoria = sumatoria + (cromosoma.pesos(i) \*acciones(i).beneficio/acciones(i).riesgo)
- fPara
3. retornar (sumatoria)

**Fin fitness**

Se explican las siguientes líneas:

- Línea 1: se inicializa la variable sumatoria en 0.
- Línea 2: se itera por cada acción en el arreglo de acciones
  - Línea 2.a: El fitness de la cromosoma se determina sumando -por cada acción- la proporción del capital que se invierte en cada acción (peso) multiplicado por el beneficio de la acción, dividido por el riesgo asociado a la acción. El beneficio y el riesgo se leen del arreglo acciones.
- Línea 3: se retorna el fitness calculado de la acción.

## CAPÍTULO 6

### 6 Diseño del algoritmo búsqueda tabú para el problema de selección de portafolio aplicado a inversiones en la bolsa de valores

#### 6.1 Estructuras de datos

##### 6.1.1 Matriz R [nxn]

La matriz R es una matriz de  $nxn$  (donde  $n$  es el número de acciones con las que trabaja el algoritmo) que contiene la lista tabú basada en el corto plazo. Cada entrada contiene el número de veces que un movimiento (inserción o intercambio) está prohibido (es tabú). El número disminuye en cada iteración hasta que llega a cero, lo que implica que el movimiento se puede realizar nuevamente.

Se considerará como inserción el agregar una acción  $i$  al portafolio seleccionado. La cantidad de turnos en los que se prohíbe el movimiento de agregar la acción  $i$  al portafolio de inversión se representan por las celdas  $e_{ii}$  de la matriz R. Se considerará como intercambio el remover acciones  $i$  del portafolio para agregar acciones  $j$ . La cantidad de turnos que un intercambio está prohibido se representan en la matriz R por las celdas  $e_{ij}$ .

A continuación se presenta un ejemplo de la matriz R:

	1	2	3	...	n
1	5	0	2	...	0
2	10	4	0	...	5
3	50	100	10	...	4
...	...	...	...	...	...
N	20	15	0	...	0

Tabla 6.1.1.1: Matriz R

##### 6.1.2 Matriz F [nxn]

La matriz F es una matriz de  $nxn$  que contiene la lista tabú basada en el largo plazo. Cada entrada contiene el número de veces que un movimiento (inserción o intercambio) se ha llevado a cabo. Si un movimiento se ha llevado a cabo muchas veces, es posible que se esté estancado en un óptimo local y no se exploren otras soluciones que se acerquen más al óptimo global. Por esta razón, si un movimiento se repite muchas veces se le pondrá una penalización de X iteraciones en la matriz R con la finalidad de explorar otras soluciones.

Se considerará como inserción el agregar una acción  $i$  al portafolio seleccionado. La cantidad de turnos en los que se prohíbe el movimiento de agregar la acción  $i$  al portafolio de inversión se representan por las celdas  $e_{ii}$  de la matriz R.

Se considerará como intercambio el remover acciones  $i$  del portafolio para agregar acciones  $j$ . La cantidad de turnos que un intercambio está prohibido se representan en la matriz R por las celdas  $e_{ij}$ .

A continuación se muestra un ejemplo de la matriz F:

	1	2	3	...	n
1	100	1	10	...	10
2	10	78	0	...	5
3	0	0	0	...	0
...	...	...	...	...	2
N	20	10	15	...	0

Tabla 6.1.2.1: Matriz F

## 6.2 Variables y parámetros

La siguiente tabla muestra los parámetros que recibirá el algoritmo búsquedatabú:

Variable/Parámetro	Representación
Ruta del archivo con las cotizaciones de las acciones de los últimos tres meses	archivo
Capital disponible a invertir en las acciones	capital
Número de iteraciones a realizar en la búsquedalocal	itLocal
Número de iteraciones a realizar en la fase de intensificación	itIntensificacion
Número de iteraciones a realizar en la fase de diversificación	itDiversificacion
Turnos mínimos a penalizar un movimiento (tenencia tabú)	tenenciaMin
Turnos máximos a penalizar un movimiento (tenencia tabú)	tenenciaMax
Cantidad de movimientos para considerar un movimiento como “frecuente”	movimientosFrecuentes

Tabla 6.2.1: Variables a usar en los algoritmos

## 6.3 Pseudocódigo del algoritmo búsquedatabú

A continuación se presenta el pseudocódigo del algoritmo búsquedatabú que será usado en este proyecto de fin de carrera. Un ejemplo de una corrida de este algoritmo se puede apreciar en el anexo 1:

**Inicio Algoritmo Búsqueda Tabú (archivo, capital, itLocal, itIntensificación, itDiversificación, tenenciaMin, tenenciaMax, movimientosFrecuentes)**

1. acciones = inicializarAcciones(archivo, capital)
2. R = inicializarMatrizTabu()
3. F = inicializarMatrizTabu()
4. solucion = solucionAleatoria()
5. mejorSolucion = solucion

/\*Fase de búsqueda local\*/

6. Para i desde 1 hasta itLocal hacer
  - a. movimiento(R,F,mejorSolucion,solución,tenenciaMin,tenenciaMax,acciones)

fPara

/\*Fase de intensificación\*/

7. solución = mejorSolucion
8. R = inicializarMatrizTabu()
9. Para i desde 1 hasta itIntensificación hacer
  - a. movimiento(R,F,mejorSolucion,solución,tenenciaMin,tenenciaMax,acciones)

fPara

/\*Fase de diversificación\*/

10. solución = mejorSolucion
11. R = inicializarMatrizTabu()
12. R = penalizarMovimientos(R,F,movimientosFrecuentes)
13. Para i desde 1 hasta itDiversificación hacer
  - a. movimiento(R,F,mejorSolucion,solución,tenenciaMin,tenenciaMax,acciones)

fPara

14. retornar mejorSolucion

**Fin Algoritmo Búsqueda Tabú**

Se explican las siguientes líneas:

- Línea 1: se inicializa la lista de acciones a partir de la información en el archivo ingresado. Se explicará este procedimiento más adelante.
- Línea 2-3: se inicializan las matrices tabú (se colocan todos los valores en 0). De esta manera, ningún movimiento está prohibido.
- Línea 4-5: se genera una solución inicial de manera aleatoria, y se inicializa la mejor solución con esta solución inicial.
- Línea 6: en la fase de búsqueda local, se itera hasta cumplir la condición de parada. La condición de parada se cumple cuando se realiza un número limitado de iteraciones. En cada iteración, se realiza un movimiento no prohibido (se explica en mayor detalle en el apartado 6.3.2).
- Línea 7-9: en la fase de intensificación, se realiza un proceso similar al de la búsqueda local pero partiendo de la solución encontrada en la búsqueda local. Se

comienza inicializando la matriz tabú de lo reciente para que ningún movimiento esté prohibido. Luego, se itera nuevamente hasta cumplir la condición de parada. La condición de parada se cumple cuando se realizar un número fijo de iteraciones. En cada iteración, se realiza un movimiento no prohibido (se explica en mayor detalle en el apartado 6.3.2).

- Línea 10-13: en la fase de diversificación, se realiza un proceso similar al de la búsqueda local pero penalizando fuertemente a los movimientos más frecuentes. Esto se hace con el fin de poder escapar algún óptimo local. Se parte de la solución encontrada en la fase de intensificación. Se comienza inicializando la matriz tabú de lo reciente para que ningún movimiento esté prohibido. Luego, examinando la matriz de lo frecuente, se penaliza fuertemente los movimientos más frecuentes en la matriz R. Posteriormente, se itera nuevamente hasta cumplir la condición de parada. La condición de parada se cumple cuando se realiza un número fijo de iteraciones. En cada iteración, se realiza un movimiento no prohibido (se explica en mayor detalle en el apartado 6.3.2).
- Línea 14: se retorna la mejor solución.

### **6.3.1 Pseudocódigo del procedimiento de inicializar el arreglo de acciones**

A continuación se presenta el pseudocódigo para inicializar el arreglo de acciones:

#### **Inicio inicializarAcciones(archivo, capital)**

1. archivoAbierto = abrirArchivo(archivo)
2. acciones = leerNombres(archivoAbierto)
3. Para cada (accion en acciones) hacer
  - a. accion.precio = leerPrecio(archivoAbierto)
  - b. accion.beneficio = calcularBeneficio(accion, archivoAbierto)
  - c. accion.riesgo = calcularDesviacionEstandar(accion, archivoAbierto)
  - d. accion.incremento = accion.precio/capital
4. Para i desde acciones.tamano() hasta 1 hacer
  - a. Si (acciones(i).beneficio <= 0) entonces
    - i. acciones.remover(i)
- fSi
- fPara
5. retornar (acciones)

#### **Fin inicializarAcciones**

Se explican las siguientes líneas:

- Línea 1: se abre el archivo.
- Línea 2: se leen los nombres de las acciones del archivo y se colocan en el arreglo acciones.
- Línea 3: se itera por cada acción encontrada.

- Línea 3.a: por cada acción, se lee el precio actual de la acción.
- Línea 3.b: por cada acción, se calcula la utilidad promedio de la acción en los últimos tres meses.
- Línea 3.c: por cada acción, se calcula la desviación estándar de las utilidades a la utilidad promedio en los últimos tres meses.
- Línea 3.d: por cada acción, se calcula la proporción de capital que representa la compra de una unidad de la acción (precio/capital).
- Línea 4: se examina todo el arreglo de acciones, y se eliminan todas aquellas que tengan beneficio negativo o nulo. De esta manera sólo se trabajará con acciones que beneficiarán el portafolio.
- Línea 5: se retorna la lista de acciones.

### 6.3.2 Pseudocódigo del procedimiento de realizar movimiento

A continuación se presenta el pseudocódigo del procedimiento para realizar un movimiento:

#### Inicio movimiento(R, F, mejorSolucion, solución, tenenciaMin, tenenciaMax, acciones)

1. mejorMovimiento = vacío
2. mejorI = vacío
3. mejorJ = vacío
4. Para i desde 1 hasta acciones.tamaño() hacer
  - a. Para j desde 1 hasta acciones.tamaño() hacer
    - i. soluciónAux = solución
    - ii. Si (R[i][j] == 0) entonces
      1. banderaMovimiento = falso
      2. Si (i == j) entonces
        - a. Si (soluciónAux.totalPesos + acciones(i).incremento <= 1) entonces
          - i. soluciónAux.pesos(i) += acciones(i).incremento
          - ii. soluciónAux.cantidades(i) += 1
          - iii. banderaMovimiento = verdadero
  - fSi
  3. Sino
    - a. Si (acciones(i).precio < acciones(j).precio) entonces
      - i. factor = acciones(j).precio / acciones(i).precio
      - ii. Si (soluciónAux.cantidades(i) >= factor) entonces
        1. soluciónAux.cantidades(i) -= factor
        2. soluciónAux.pesos(i) -= factor \* acciones(i).incremento
        3. soluciónAux.cantidades(j) += 1
        4. soluciónAux.pesos(j) += acciones(j).incremento
        5. banderaMovimiento = verdadero

fSi

b. Sino

- i. factor = acciones(i).precio / acciones(j).precio
- ii. Si (solucionAux.cantidades(i) >= 1) entonces
  - 1. solucionAux.cantidades(i) -= 1
  - 2. solucionAux.pesos(i) -= acciones(i).incremento
  - 3. solucionAux.cantidades(j) += factor
  - 4. solucionAux.pesos(j) += factor\*acciones(j).incremento
  - 5. banderaMovimiento = verdadero

fSi

fSi

fSi

4. Si (banderaMovimiento) entonces

- a. Si (mejorMovimiento == vacio) entonces
  - i. mejorMovimiento = solucionAux
  - ii. mejorI = i
  - iii. mejor = j
- b. Sino
  - i. Si (fitness(solucionAux,acciones) > fitness(mejorMovimiento,acciones)) entonces
    - 1. mejorMovimiento = solucionAux
    - 2. mejorI = i
    - 3. mejorJ = j

fSi

fSi

fSi

- iii. Si ( $R[i][j] > 0$ ) entonces
  - 1.  $R[i][j] =$

fSi

fPara

fPara

- 5. Si (mejorMovimiento <> vacio) entonces
  - a. solucion = mejorMovimiento

- b.  $R[\text{mejorI}][\text{mejorJ}] += \text{aleatorio}(\text{tenenciaMin}, \text{tenenciaMax})$
- c.  $F[\text{mejorI}][\text{mejorJ}] += 1$
- d. Si ( $\text{fitness}(\text{solución}, \text{acciones}) > \text{fitness}(\text{mejorSolucion}, \text{acciones})$ )
  - entonces hacer
    - i.  $\text{mejorSolucion} = \text{solucion}$

fSi

fSi

**Fin movimiento**

Se explican las siguientes líneas:

- Línea 1-3: se inicializan las variables que almacenan la información del mejor movimiento a realizar. La variable `mejorMovimiento` almacena la solución que se genera al realizar el mejor movimiento, y las variables `mejorI` y `mejorJ` almacenan coordenadas del mejor movimiento en las matrices Tabú.
- Línea 4: se itera para examinar todos los movimientos posibles que se pueden realizar a partir de la solución actual
  - Línea 4.a.i: se crea una variable auxiliar `solucionAux` que almacenará la solución generada al realizar el movimiento de la iteración que se está estudiando.
  - Línea 4.a.ii: se verifica que el movimiento no esté prohibido ( $R[i][j]$  es igual a 0). Si el movimiento no está prohibido:
    - Línea 4.a.ii.1: se crea una variable binaria `banderaMovimiento` que indica si se pudo realizar el movimiento o si no se pudo realizar el movimiento debido a una restricción.
    - Línea 4.a.ii.2: se verifica si el movimiento a realizar es una inserción ( $i==j$ ) o un intercambio ( $i<>j$ ). Si el movimiento es una inserción:
      - Línea 4.a.ii.2.a: se verifica que agregar una acción más al portafolio no haga sobrepasar el presupuesto disponible. Si no hay problema, se agrega la acción al portafolio incrementando los elementos correspondientes en los arreglos pesos y cantidades. Conjuntamente se prende la bandera `banderaMovimiento`.
    - Línea 4.a.ii.3: en cambio, si se trata de un intercambio:
      - Línea 4.a.ii.3.a: se revisa qué acción tiene mayor precio unitario: la que entra o la que sale de la solución. Si la acción que entra tiene mayor precio unitario, esto implica que por cada unidad que entra van a tener que salir varias unidades de la acción que sale. Entonces, se verifica que el portafolio cuente con suficientes unidades de la acción que sale para poder realizar el intercambio. Se realiza el intercambio y se prende la bandera `banderaMovimiento`. Caso contrario, el intercambio no se podrá realizar.

- Línea 4.a.ii.3.b: si la acción que sale tiene mayor precio unitario, por cada unidad de la acción que sale entrarán varias unidades de la acción que entra. En este caso, se verifica que se cuente con por lo menos 1 unidad de la acción que sale en el portafolio y de ser así se procede a realizar el intercambio. Finalmente, si el intercambio se realizó se prende la bandera banderaMovimiento.
- Línea 4.a.ii.4: Se verifica si se pudo realizar el movimiento en la iteración.
  - Línea 4.a.ii.4.a: si es el primer movimiento que se estudia (la variable mejorMovimiento es vacía), el movimiento realizado esta iteración se considera como el mejor movimiento hasta el momento. Se coloca la solución generada con el movimiento de la iteración en la variable mejorMovimiento, y se actualizan las variables mejorI y mejorJ para que almacenen las coordenadas del movimiento realizado.
  - Línea 4.a.ii.4.b: si no es el primer movimiento que se estudia (la variable mejorMovimiento no es vacía), se compara el fitness (ver apartado 6.3.3) de la solución generada en la iteración con el fitness de la solución generada por el mejor movimiento hasta el momento. Si el movimiento realizado esta iteración genera una solución de mayor fitness, se actualiza el mejor movimiento. Se coloca la solución generada con el movimiento de la iteración en la variable mejorMovimiento, y se actualizan las variables mejorI y mejorJ para que almacenen las coordenadas del movimiento realizado.
- Línea 4.a.iii: Dado a que ha pasado un turno, se despenaliza en 1 los movimientos prohibidos en la matriz de lo reciente R.
- Línea 5: se valida si se eligió algún movimiento (si la variable mejorMovimiento no es vacía). Si se eligió un movimiento:
  - Línea 5.a: se actualiza la variable solucion, cambiándola por la solución generada al realizar el mejor movimiento.
  - Línea 5.b: se penaliza el movimiento realizado en la matriz de corto plazo. El movimiento realizado no se podrá hacer en un número aleatorio de turnos entre tenenciaMin y tenenciaMax.
  - Línea 5.c: se graba el movimiento que se hizo en la matriz de lo frecuente, incrementando el valor de la celda correspondiente en 1.
  - Línea 5.d: si la solución luego de realizar el movimiento tiene un fitness mayor que la mejor solución, se actualiza la mejor solución. El fitness de una solución está determinado por la función objetivo (revisar acápite 6.3.3).

### 6.3.3 Pseudocódigo de la función fitness

A continuación se presenta el pseudocódigo de la función fitness:

#### Inicio fitness(cromosoma, acciones)

1. sumatoria = 0
2. Para i desde 1 hasta acciones.tamano() hacer
  - a. sumatoria = sumatoria + (cromosoma.pesos(i) \*acciones(i).beneficio/acciones(i).riesgo)
- fPara
3. retornar (sumatoria)

#### Fin fitness

Se explican las siguientes líneas:

- Línea 1: se inicializa la variable sumatoria en 0.
- Línea 2: se itera por cada acción en el arreglo de acciones
  - Línea 2.a: El fitness de la cromosoma se determina sumando -por cada acción- la proporción del capital que se invierte en cada acción (peso) multiplicado por el beneficio de la acción, dividido por el riesgo asociado a la acción. El beneficio y el riesgo se leen del arreglo acciones.
- Línea 3: se retorna el fitness calculado de la acción.

## CAPÍTULO 7

### 7 Interfaz gráfica

A continuación se presenta la interfaz que se usará para mostrar los resultados de la ejecución de los algoritmos. Dicha interfaz se usará tanto para el algoritmo genético como para el algoritmo búsquedas tabú.

La interfaz está compuesta de dos ventanas. En la primera pantalla, se carga a memoria la información de las acciones a partir del archivo con las cotizaciones. Asimismo, se ingresan los parámetros para los algoritmos genético y búsquedas tabú.

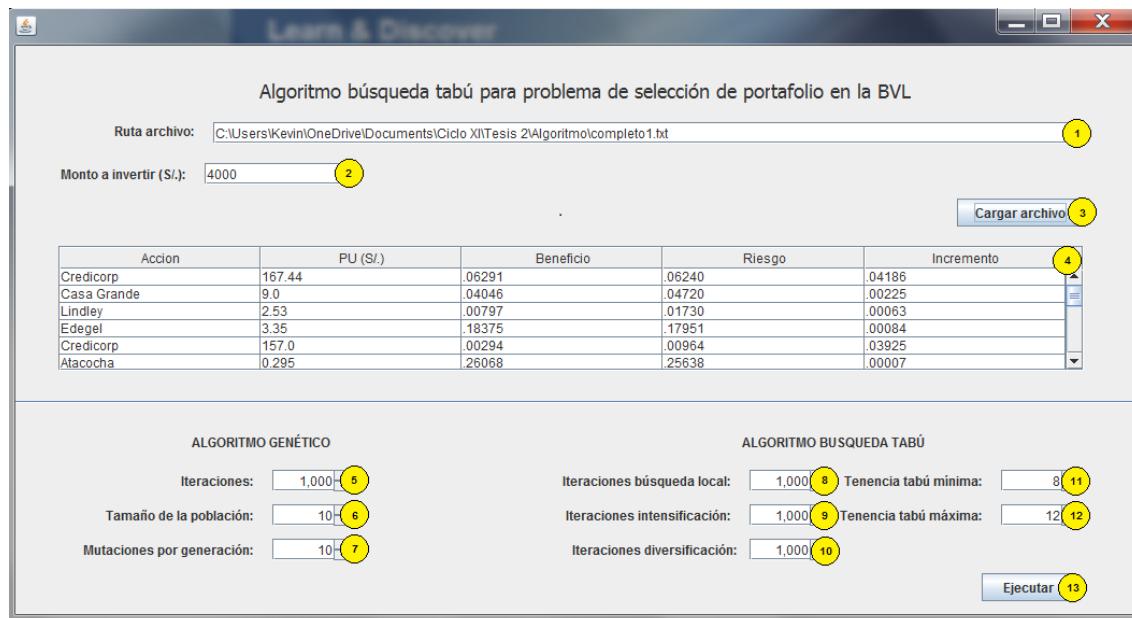


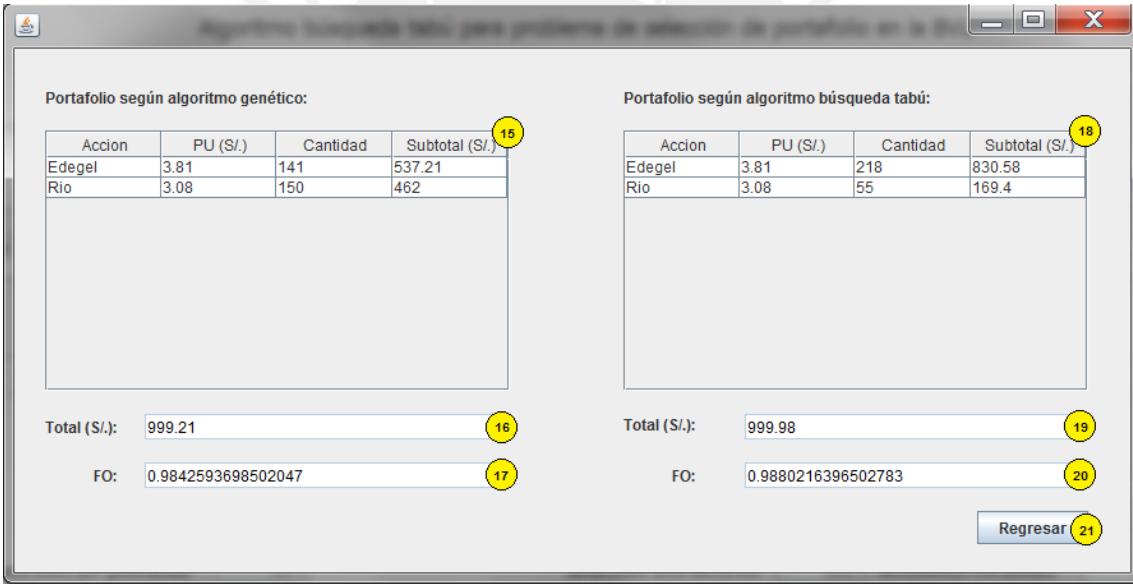
Imagen 7.1: Primera pantalla

Se explican los siguientes campos:

1. Se ingresa la ruta a un archivo con la cotización de las acciones.
2. Se debe de ingresar el monto a invertir en las acciones en soles (parámetro *capital*)
3. Se hace clic en el botón para cargar la información del archivo a memoria. La información de las acciones en el archivo se muestra en la tabla 4.
4. La tabla muestra la información de las acciones leídas en el archivo que usará el algoritmo (aquellas que tengan beneficio positivo). Por cada acción se muestra: el nombre, el precio unitario, el beneficio estimado a tres meses, el riesgo estimado a tres meses y el incremento.
5. Se puede configurar el número de iteraciones que se realizarán en el algoritmo genético
6. Se puede configurar el tamaño de la población con la que trabajará el algoritmo genético.

7. Se puede configurar el número de mutaciones por iteración del algoritmo genético.
8. Se puede configurar el número de iteraciones en la fase de búsqueda local del algoritmo búsqueda tabú.
9. Se puede configurar el número de iteraciones en la fase de intensificación del algoritmo búsqueda tabú.
10. Se puede configurar el número de iteraciones en la fase de diversificación del algoritmo búsqueda tabú.
11. Se puede configurar el número de turnos mínimo que se penalizará un movimiento.
12. Se puede configurar el número de turnos máximo que se penalizará un movimiento.
13. Se da clic en el botón para ejecutar los algoritmos con los parámetros indicados y mostrar la segunda pantalla con los resultados.

La segunda pantalla muestra los resultados de la ejecución de los algoritmos luego de dar clic en el botón 14.



Acción	PU (S/.)	Cantidad	Subtotal (S/.)
Edegel	3.81	141	537.21
Rio	3.08	150	462

Total (S/.): 999.21

FO: 0.9842593698502047

Acción	PU (S/.)	Cantidad	Subtotal (S/.)
Edegel	3.81	218	830.58
Rio	3.08	55	169.4

Total (S/.): 999.98

FO: 0.9880216396502783

**Imagen 7.2: Segunda pantalla**

Se explican los siguientes campos:

14. La tabla contiene el portafolio de acciones seleccionado por el algoritmo genético. Por cada acción, se muestran los siguientes datos: el nombre de la acción, el precio unitario de la acción, la cantidad de acciones que el algoritmo sugiere comprar y el subtotal de adquirir dichas acciones.
15. Se muestra el monto total a desembolsar por el portafolio de acciones generado por el algoritmo genético.
16. Se muestra el valor de la función objetivo de la solución generada por el algoritmo genético.
17. La tabla contiene el portafolio de acciones seleccionado por el algoritmo búsqueda tabú. Por cada acción, se muestran los siguientes datos: el nombre de

la acción, el precio unitario de la acción, la cantidad de acciones que el algoritmo sugiere comprar y el subtotal de adquirir dichas acciones.

18. Se muestra el monto total a desembolsar por el portafolio de acciones generado por el algoritmo búsquedatabú.
19. Se muestra el valor de la función objetivo de la solución generada por el algoritmo tabú.
20. El botón permite regresar a la primera pantalla. A partir de ahí, se puede realizar otra corrida de los algoritmos con archivos diferentes o parámetros distintos.



## CAPÍTULO 8

### 8 Calibración de variables y parámetros

El siguiente apartado muestra la justificación de los parámetros que recibirá el algoritmo búsquedas tabú de este trabajo de fin de carrera. Se abarcarán los siguientes parámetros:

- Monto de inversión
- Número de iteraciones a realizar en la fase de búsqueda local
- Número de iteraciones a realizar en la fase de intensificación
- Número de iteraciones a realizar en la fase de diversificación
- Tenencia tabú mínima
- Tenencia tabú máxima

#### 8.1 Monto de inversión

De acuerdo al boletín de la BVL, a partir del 5 de enero de 2015 el monto mínimo para realizar una operación en bolsa es de S/. 3850 [BVL, 2015]. Teniendo esto en cuenta, para este trabajo de fin de carrera se usará como monto de inversión S/. 4000. Este monto es apropiado debido a que es superior a la acción más cara (BAP a S/. 469.61 al 19 de abril de 2015) y permite que el algoritmo contemple todas las acciones disponibles. Además, al estar próximo al monto mínimo de inversión, los resultados obtenidos se pueden aplicar al inversionista más pequeño. Y para inversionistas más grandes, el portafolio de acciones obtenido con S/. 4000 se puede extrapolar de manera apropiada.

#### 8.2 Número de iteraciones a realizar en la fase de búsqueda local, número de iteraciones a realizar en la fase de búsqueda local y número de iteraciones a realizar en la fase de intensificación

El número de iteraciones en cada una de las fases varía según el problema a resolver. Para este trabajo, el número de iteraciones apropiado será calculado de manera empírica. En este trabajo, se partirá de 1000 iteraciones en cada fase debido a que es el número mínimo de iteraciones recomendada por la literatura [HAO, GALINIER, 1998]. Se irá incrementando el número de iteraciones en 1000 en cada prueba y se calculará la variación en la función objetivo. Se detendrá las pruebas cuando incrementar el número de iteraciones no cause un cambio significativo en la función objetivo calculada (menor al 0.1%).

A continuación se muestra el resultado de la calibración de variables:

Iteraciones	Función Objetivo de la Solución Encontrada
1000	1.008842
2000	1.009446
3000	1.009892

Se observa que al incrementar las iteraciones de 1000 a 2000, la función objetivo de la solución solo incrementó en 0.000604 (menos del 0.1%). Se concluye que 1000 es un

número aceptable de iteraciones. Incrementar el número de turnos no tiene un efecto significativo sobre la función objetivo para justificar el mayor esfuerzo computacional.

### 8.3 Tenencia tabú mínima y tenencia tabú máxima

La tenencia tabú es el número de turnos que se prohibirá un movimiento. De acuerdo al trabajo de Hao y Galinier [HAO, GALINIER, 1998], es conveniente que este valor no sea estático debido a que se obtienen mejores resultados si se trabaja de manera dinámica. Para fases de 1000 iteraciones, se recomienda usar cerca de 10 iteraciones. Por lo que se trabajará con una tenencia tabú que oscile entre 8 y 12 turnos.



## CAPÍTULO 9

### 9 Experimentación numérica

En este capítulo, se muestran los resultados de la experimentación numérica para comparar el desempeño del algoritmo genético contra el desempeño del algoritmo búsquedas tabú.

Los algoritmos se ejecutaron 40 veces cada uno. En cada ejecución, se utilizó un archivo diferente. Cada archivo contenía tres meses de cotizaciones de las veinte acciones más líquidas de la BVL en algún periodo entre enero de 2011 y enero de 2014.

#### 9.1 Resultados

A continuación se muestran los resultados obtenidos en la ejecución de ambos algoritmos. Se presenta el valor de la función objetivo de la mejor solución encontrada en cada ejecución. Una muestra de los datos usados se muestra en el anexo 2.

Muestra	Función Objetivo de la Solución del Algoritmo Genético	Función Objetivo de la Solución del Algoritmo Búsquedas Tabú
1	0.90085	1.02359
2	0.96886	1.02160
3	0.97925	1.02291
4	1.0055	1.00745
5	1.01008	1.01911
6	1.00593	1.02324
7	0.99735	1.02084
8	0.97111	1.01911
9	0.96793	1.00775
10	0.99846	1.02087
11	0.96905	1.01019
12	0.9219	1.02266
13	0.97966	1.02318
14	0.98454	1.02327
15	0.93299	1.02202
16	1.0127	1.01298
17	0.87618	1.01787
18	0.92073	1.02097
19	0.95743	1.01710
20	0.92702	1.02158
21	0.95849	1.01911
22	1.00663	1.02220
23	1.00000	1.01787
24	0.98415	1.01911
25	0.97184	1.02248
26	0.94152	1.00775

27	0.99641	1.02322
28	0.99282	1.02222
29	0.97026	1.01900
30	0.99019	1.02157
31	0.93783	1.01701
32	0.97903	1.01872
33	0.7578	1.02158
34	0.64656	1.02036
35	0.89408	1.02171
36	0.77152	1.02036
37	0.99524	1.02229
38	1.01477	1.02104
39	0.96957	1.02221
40	1.00162	1.02269

## 9.2 Prueba Kolmogorov-Smirnov

Es necesario demostrar que los resultados de la ejecución de los algoritmos presentan una distribución normal. Para ello, se realiza la prueba Kolmogorov-Smirnov.

Prueba K-S para algoritmo genético:

Media	0.95169625	Estadístico S-K	0.17599												
Desviación	0.075244747	Nivel de significancia	0.05												
Mínimo	0.64656	Grados de libertad	40												
Máximo	1.01477	Estadístico de la tabla	0.21012												
Rango	0.36821	Hipótesis	Se ACEPTA												
Número de datos (N)	40														
# de intervalos (sturges)	13.1733022														
# de intervalos (raíz de N)	6.32455532														
Tamaño del intervalo	0.058219113														
Intervalos															
1	0.64656	Límite inferior	0.70478	Límite superior	1.00000	FO	0.02500	FOR	0.02500	FOR acum	0.02500	FER acum	0.00052	Abs (For acum-Fer acum)	0.02448
2	0.70478		0.76300		1.00000		0.02500		0.05000		0.05000		0.00607		0.04393
3	0.76300		0.82122		1.00000		0.02500		0.07500		0.07500		0.04145		0.03355
4	0.82122		0.87944		1.00000		0.02500		0.10000		0.10000		0.16844		0.06844
5	0.87944		0.93766		6.00000		0.15000		0.25000		0.25000		0.42599		0.17599
6	0.93766		0.99587		19.00000		0.47500		0.72500		0.72500		0.72144		0.00356
7	0.99587		1.05409		11.00000		0.27500		1.00000		1.00000		0.91322		0.08678

Prueba K-S para algoritmo búsqueda tabú:

Media	1.01951975	Estadístico S-K	0.09326												
Desviación	0.004386172	Nivel de significancia	0.05												
Mínimo	1.00745	Grados de libertad	40												
Máximo	1.02359	Estadístico de la tabla	0.21012												
Rango	0.01614	Hipótesis	Se ACEPTA												
Número de datos (N)	40														
# de intervalos (sturges)	13.1733022														
# de intervalos (raíz de N)	6.32455532														
Tamaño del intervalo	0.002551958														
Intervalos															
1	1.00745	Límite inferior	1.01000	Límite superior	3.00000	FO	0.07500	FOR	0.07500	FOR acum	0.07500	FER acum	0.01501	Abs (For acum-Fer acum)	0.05999
2	1.01000		1.01255		1.00000		0.02500		0.10000		0.10000		0.05613		0.04387
3	1.01255		1.01511		1.00000		0.02500		0.12500		0.12500		0.15713		0.03213
4	1.01511		1.01766		2.00000		0.05000		0.17500		0.17500		0.33560		0.16060
5	1.01766		1.02021		8.00000		0.20000		0.37500		0.37500		0.56250		0.18750
6	1.02021		1.02276		19.00000		0.47500		0.85000		0.85000		0.77009		0.07991
7	1.02276		1.02531		6.00000		0.15000		1.00000		1.00000		0.90674		0.09326

Siendo las hipótesis para ambos algoritmos:

$H_0$ : El algoritmo genético presenta una distribución normal.

$H_1$ : El algoritmo genético no presenta una distribución normal.

$H_0$ : El algoritmo búsquedas tabú presenta una distribución normal.

$H_1$ : El algoritmo búsquedas tabú no presenta una distribución normal.

Como se puede observar, en ambos casos el valor del estadístico S-K es menor al valor crítico para 40 muestras. Esto quiere decir que la solución está fuera de la región crítica. Por lo tanto, en ambos casos se acepta la hipótesis  $H_0$ , aceptando que los datos presentan una distribución normal.

### 9.3 Prueba F de Fisher

Además de demostrar la normalidad de los datos mediante la prueba Kolmogorov-Smirnov, es necesario determinar si las soluciones presentan varianzas homogéneas o heterogéneas.

	Genético	Búsqueda Tabú
Media	0.95170	1.01952
Varianza	0.00566	0.00002
Observaciones	40	40
Grados de libertad	39	39
F	294.2937585	
P(F<=f) una cola	2.22272E-38	
Valor crítico para F (una cola)	1.704465067	
Hipótesis	Se ACEPTA	

Las hipótesis que se van a evaluar son:

$H_0$ : Las varianzas son homogéneas

$H_1$ : Las varianzas son heterogéneas

Como se puede observar de los resultados, el valor del estadístico F obtenido de los resultados de los algoritmos es menor al valor crítico para la prueba. Es decir, la solución está fuera de la región crítica. Por lo tanto, se acepta la hipótesis  $H_0$ , llegando a la conclusión de que las varianzas son homogéneas.

### 9.4 Prueba Z

La prueba Z permitirá comparar la media de los resultados del algoritmo genético con la media de los resultados del algoritmo búsquedas tabú. Esta prueba es apropiada cuando el número de muestras es elevado (mayor a 35 muestras), como en este proyecto de fin de carrera. Para que la prueba Z sea válida, hay que corroborar que los datos presenten una distribución normal y que posean varianzas homogéneas. Esto ha sido demostrado por medio de la prueba Kolmogorov-Smirnov y la prueba F de Fisher respectivamente.

Esta prueba está dividida en dos partes. En la primera parte, se determinará si la media de los resultados obtenidos con el algoritmo genético es igual o significativamente

diferente a la media de los resultados obtenidos con el algoritmo búsquedas tabú. Si se llega a la conclusión de que las medias son diferentes, en la segunda parte se determinará cuál de las dos medias es mayor a la otra.

	<i>Genético</i>	<i>Búsqueda Tabú</i>
Media	0.95169625	1.01951975
Varianza	0.005661772	1.92385E-05
Observaciones	40	40
Diferencia hipotética de las medias	0	
z	-5.691115487	
P(Z<=z) una cola	6.31061E-09	
Valor crítico de z (una cola)	1.644853627	
P(Z<=z) dos colas	1.26212E-08	
Valor crítico de z (dos colas)	1.959963985	
Hipótesis comparación 1	Se RECHAZA	
Hipótesis comparación 2	Se RECHAZA	

#### 9.4.1 Comparación 1

La primera comparación consiste en demostrar que los resultados de ambos algoritmos poseen medias diferentes, para lo cual se recurre a la prueba de dos colas usando como hipótesis:

$H_0$ : La media del algoritmo genético es igual que la media del algoritmo búsquedas tabú

$H_1$ : La media del algoritmo genético es diferente que la media del algoritmo búsquedas tabú

Para ello se usa el valor crítico de Z (dos colas). Para aceptar como cierta la hipótesis nula, el valor de Z debe estar entre -1.95996 y +1.95996. Sin embargo, se observa que el valor es -5.69111 por lo que caería en una región crítica haciendo que se rechace la hipótesis nula. Se llega a la conclusión que tanto las medias del algoritmo genético y el algoritmo búsquedas tabú son diferentes.

#### 9.4.2 Comparación 2

Al haber demostrado que las medias son significativamente diferentes, corresponde averiguar si la media de los resultados del algoritmo genético es mayor que la media de los resultados del algoritmo búsquedas tabú o viceversa. Se recurre a la prueba de una cola utilizando como hipótesis:

$H_0$ : La media del algoritmo genético es mayor que la media del algoritmo búsquedas tabú

$H_1$ : La media del algoritmo genético es menor que la media del algoritmo búsquedas tabú

Para ello se usa el valor el valor en tabla de valor crítico de Z (una cola).

Para aceptar como cierta la hipótesis nula, el valor de Z debe ser mayor a -1.64485. Sin embargo, se observa que dicho valor es -5.69111 por lo que caería en una región crítica haciendo que se rechace la hipótesis nula. Se llega a la conclusión que la media del algoritmo genético es menor a la media del algoritmo búsquedas tabú.

## 10 Conclusiones

### 10.1 Observaciones del proyecto

El algoritmo búsquedas tabú propuesto en este proyecto de fin de carrera cuenta con tres fases. La primera etapa es la búsqueda local donde se obtiene una solución inicial. En la segunda etapa, la fase de intensificación, se explora la vecindad de la solución obtenida en busca de mejores soluciones. Finalmente, en la etapa de diversificación, se penaliza fuertemente los movimientos más frecuentes con el fin de explorar soluciones nuevas y escapar de un óptimo local. En cada etapa se realizaron 1000 iteraciones. Se decidió este número debido a que es el número mínimo que exige la bibliografía, y para nuestro caso se demostró de manera empírica que incrementar este número no producía un cambio significativo en la calidad de la solución.

Otro factor clave a tomar en cuenta en el algoritmo búsquedas tabú es la tenencia tabú. Este parámetro indica por cuantos turnos un movimiento estará prohibido. En este proyecto, se optó por una tenencia tabú aleatoria que oscilara en un intervalo definido. Según la literatura, se obtienen mejores resultados de esta manera.

El algoritmo búsquedas tabú propuesto en este trabajo de fin de carrera se comparó con el algoritmo genético de Lin adaptado. Para lograr esto, se realizó una experimentación numérica que permitió comparar el desempeño de ambos algoritmos bajo condiciones iguales.

Cabe mencionar que ambos algoritmos son configurables. En el algoritmo genético, se puede configurar el número de iteraciones, la cantidad de mutaciones por iteración y el tamaño de la población. En el caso del algoritmo búsquedas tabú, se puede modificar el número de iteraciones en cada fase y los límites del intervalo de la tenencia tabú.

### 10.2 Conclusiones

Luego de haber realizado la experimentación numérica, se ha podido observar que la media de la función objetivo de las soluciones generadas por el algoritmo búsquedas tabú es significativamente mayor a la media de la función objetivo de las soluciones generadas por el del algoritmo genético. Se puede concluir que el algoritmo búsquedas tabú tiene mejor desempeño que el algoritmo genético para el conjunto de datos que se usó tomando en cuenta la rentabilidad de las acciones y el riesgo relacionado al portafolio de inversión.

Se puede afirmar que el presente proyecto de fin de carrera presenta una alternativa de solución al problema planteado. Se obtiene un portafolio de acciones rentable y en menor tiempo de lo que demora realizar un análisis técnico. Por lo tanto, el algoritmo podría ser usado para la toma de decisiones al armar un portafolio de inversión en una bolsa de valores.

### 10.3 Recomendaciones

Se proponen las siguientes investigaciones futuras a partir de este proyecto de fin de carrera:

- Adaptar el algoritmo para que pueda trabajar con acciones que cotizan en diferentes divisas.

- Implementar un sistema de información que permita realizar operaciones de compra y venta en tiempo real, el cual use este algoritmo para darle sugerencias al inversionista.



## Referencias bibliográficas

[AKDEMIR, YU, 2009] Predicción del comportamiento del mercado de valores usando el método euclídeo de normalización con redes neuronales artificiales.  
AKDEMIR, B. y YU, L.  
2009. IEEE Computer Society. p.562-567.

[BVL, 2014] BVL.  
2014. <http://www.bvl.com.pe>

[BRASSAR, BRATLEY, 1996] Fundamentos de algoritmia.  
BRASSAR, G. y BRATLEY, P.  
1996. New Jersey – USA: Editorial Prentice Hall.

[CASANOVA, 2006] TRADINNOVA: Un algoritmo heurístico de compra-venta inteligente de acciones.  
CASANOVA, I.J.  
2006. Universidad de Murcia.

[CHIAVENATO, 1989] Introducción a la Teoría General de la Administración.  
CHIAVENATO, Adalberto.  
1989. Bogotá: Editorial Mc Graw Hill.

[CHOU, 1975] Análisis estadístico.  
CHOU, Ya-Lun.  
1975. Holt International.

[EHRHARDT, 2010] Finanzas corporativas: un acercamiento enfocado.  
EHRHARDT, Michael.  
2010. Cengage Learning. p.278.

[FEO, RESENDE, 1995] Greedy Randomized Adaptive Search Procedure.  
FEO, T. y RESENDE, M.  
1995. Journal of Global Optimization. p.109-133.

[GARCÍA BELLIDO, 2010] SPSS: Pruebas No Paramétricas.  
GARCÍA BELLIDO, R.  
2010. Grupo de Innovación Educativa. Universitat de València.

[GAREY, JONSON, 1997] Búsqueda local en optimización combinatoria.  
GAREY, M. y JONSON, D.  
1997. Inglaterra: Editorial Wiley & Sons.

[GLOVER, 1986] Caminos futuros para programación entera y enlaces a la inteligencia artificial.  
GLOVER, Fred.  
1986. USA: Computers and Operations Research 13. p.533-549.

[GLOVER, LAGUNA, 1997] Principios de búsqueda tabú.  
GLOVER, Fred y LAGUNA, Manuel.

1997. USA: Kluwer Academic Publishers.

**[GLOVER, MELIÁN, 2003] Búsqueda tabú.**

GLOVER, Fred y MELIÁN, Belén.

2003. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Volumen 19. p29-48.

**[GUILLEN, 2008] Métodos Cuantitativos en Control de Riesgos.**

GUILLEN, Montserrat.

2008. Universidad de Barcelona.

**[HAO, GALINIER, 1998] Búsqueda tabú para asignación de frecuencias en redes móviles de radiofrecuencia.**

HAO, J.K. y GALINIER, P.

1998. Journal of Heuristics. p47-62.

**[HOTSTOCKED PRECISION, 2014] HOTSTOCKED PRECISION**

2014. <http://www.hotstocked.com/precisionapp>

**[JANUSEVICIUTS, 2003] Probando la eficiencia de mercados de valores usando redes neuronales en el caso de Lituania.**

JANUSKEVICIUTS, M.

2003. SSE Riga Working Papers 17.

**[JAVA, 2014] JAVA**

2014. <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>

**[KROHA, 2004] Clasificación de noticias de la bolsa, reporte técnico.**

KROHA, P.

2004. Department of Computer Science, Engineering School, Universidad de Chile.

**[LIAN, LI, 2010] Toma de decisiones fuzzy en problemas del portafolio.**

LIAN, K. y LI, C.

2010. IEEE International conference On Systems, Man And Cybernetics.

**[LIN, 2007] Un algoritmo genético efectivo para el problema de optimización del portafolio multiobjetivo.**

LIN, Chi-Ming.

2007. Applied Mathematical Sciences, Vol. 1, p.201-210.

**[MATRI, 2007] Procedimientos Metaheurísticos en Optimización Combinatoria.**

MATRI, Rafael.

2007. Matemàtiques. Universitat de València. p.1-60.

**[MICROSOFT OFFICE, 2014] ¿Qué es Excel?**

2014. <http://office.microsoft.com/es-mx/support/que-es-excel-HA010265948.aspx>

**[MORA ENGUÍDANOS, 2008] Diccionario de Contabilidad, Auditoría y Control de Gestión.**

MORA ENGUÍDANOS, Araceli.

2008. Madrid: Ecobook. p.12.

**[NETBEANS, 2014] NETBEANS**

2014. <http://www.netbeans.org/features/index.html>

**[ORTIZ, 2004] Métodos Numéricos Aplicados a la Ingeniería**

ORTIZ, Salomón.

2004. Universidad Nacional Jorge Basadre Grohmann.

**[PMI, 2014] Guía de los Fundamentos para la Dirección de Proyectos**

PROJECT MANAGEMENT INSTITUTE

2009. Cuarta edición. Pennsylvania: PMI Publications.

**[PPO, 2013] Project Portfolio Office**

PPO

2014. <http://www.pwc.com>

**[PWC, 2014] Project Portfolio Optimization**

PWC

2014. <http://www.projectportfoliooffice.com/index.php>

**[RIOS MERCADO, BARD, 2000] Heurística para Secuenciamiento de Tareas en Líneas de Flujo.**

RIOS MERCADO, R. y BARD, J.

2000. Ciencia UNAL – Facultad de Ingeniería Mecánica y Eléctrica – Universidad de Nuevo León.

**[RUBINSTEIN, 2002] La “selección del portafolio” de Markowitz: retrospectiva de cincuenta años.**

RUBINSTEIN, M.

2002. Journal of Finance 57, No. 3. p.1041-1045.

**[SAEED, 2000] Fundamentos de Probabilidad (2da edición).**

SAEED, Ghahramani.

2000. Prentice Hall: New Jersey. p.438.

**[SEFIANE, 2012] Selección de portafolio usando algoritmos genéticos.**

SEFIANE, Slimane.

2012. Journal of Applied Finance & Banking. Scienpress Ltd. p.143-154.

**[SILVER, 2004] Una revision a métodos heurísticos.**

SILVER, E.

2004. Journal of the Operational Resear Society. p.936-956.

**[STOCK SIGNAL PRO, 2014] STOCK SIGNAL PRO**

2014. [http://www.stocksignal.net/tour/stock\\_signal\\_pro\\_q\\_and\\_a.htm](http://www.stocksignal.net/tour/stock_signal_pro_q_and_a.htm)

**[SUBRAMANIAN, RAMAMOORTHY, 2006] Diseñando agentes seguros, rentables y automatizados usando algoritmos evolutivos.**

SUBRAMANIAN, Harish y RAMAMOORTHY, Subramanian.

2006. The University of Texas at Austin.

**[TAKEUCHI, NONAKA, 1986]** El nuevo nuevo juego de desarrollo de productos.  
TAKEUCHI, Hirotaka y NONAKA, Ikujiro.  
1986. Harvard Business review 64. p137-146.

**[TUPIA, 2005]** Un algoritmo GENÉTICO para resolver el problema de la programación de tareas dependientes en máquinas diferentes.  
TUPIA, Manuel.  
2005. Tesis para optar por el título de Magister en Ingeniería de Sistemas con la mención en Ingeniería de Software. Lima: Universidad Nacional Mayor de San Marcos.

**[YCART, 2014]** Tests de Fisher y de Student  
YCART, Bernard.  
2014. Laboratoire Jean Kuntzmann.

**[ZABIR, 2011]** Predicción de precios en el mercado de acciones usando redes neuronales artificiales.  
ZABIR, H.K.  
2011. International Journal of Computer Applications. p42-47.

**[ZAHRA, 2004]** Comparación de algoritmos metaheurísticos en el problema de horarios.  
ZAHRA, N.  
2004. Korean Society for Computational & Applied Mathematics and Korean SIGCAM.