

OPTIMIZACION DE PORTAFOLIOS CON
RESTRICCIONES EMPLEANDO ALGORITMOS
GENETICOS



TESIS

MAESTRIA EN CIENCIAS EN SISTEMAS INTELIGENTES

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

POR:

ING. CARLOS JOAN MARTINEZ ROMERO

DICIEMBRE 2008

OPTIMIZACION DE PORTAFOLIOS CON
RESTRICCIONES EMPLEANDO ALGORITMOS
GENETICOS



TESIS

MAESTRIA EN CIENCIAS EN SISTEMAS INTELIGENTES

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

POR:

ING. CARLOS JOAN MARTINEZ ROMERO

DICIEMBRE 2008

Optimización de portafolios con restricciones empleando algoritmos genéticos

TESIS

Maestría en Ciencias en
Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Por

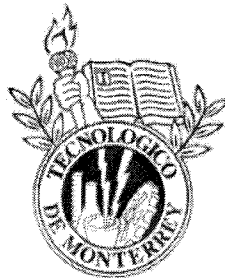
Ing. Carlos Joan Martínez Romero

Diciembre 2008

Optimización de portafolios con restricciones empleando algoritmos genéticos

Por

Ing. Carlos Joan Martínez Romero



TESIS

Presentada a la División de Mecatrónica y Tecnologías de Información
Este trabajo es requisito parcial para obtener el grado académico de Maestro en
Ciencias en Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Monterrey, N.L. Diciembre de 2008

Esta tesis está dedicada a mis Padres y mi hermano, a quienes agradezco de todo corazón por su amor, apoyo y comprensión. También está dedicada a mi novia Dulce Jacqueline quien me ha apoyado en todo momento.

Reconocimientos

Deseo externar un sincero agradecimiento a las personas que de alguna forma colaboraron en el desarrollo de esta tesis.

Al Dr. Manuel Valenzuela Rendón por su apoyo y paciencia a lo largo del desarrollo de la tesis, a mis sinodales el Dr. Horacio Martínez y el Dr. Eduardo Uresti por sus cometarios acertados.

A mi padre por su apoyo y consejos para la elaboración de la tesis.

A mi novia Dulce Jacqueline por sus esfuerzos y colaboración con este trabajo.

A mis profesores de la maestría que contribuyeron con sus conocimientos.

A mis amigos y compañeros de la maestría que aportaron buenos consejos.

CARLOS JOAN MARTÍNEZ ROMERO

*Instituto Tecnológico y de Estudios Superiores de Monterrey
Diciembre 2008*

Optimización de portafolios con restricciones empleando algoritmos genéticos

Carlos Joan Martínez Romero, M.C.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2008

Asesor de la tesis: Dr. Manuel Valenzuela Rendón

En esta tesis se desarrolla el problema de selección de portafolios de inversión empleando algoritmos genéticos en donde se trata el problema de optimización. En este trabajo se plantean dos variantes del problema, en la primera variante se busca resolver el problema de optimización para un solo portafolio en forma de un objetivo con un algoritmo genético, en el segundo enfoque se aplican algoritmos multiobjetivo para la selección de múltiples resultados distribuidos en el frente de Pareto, conocido como frontera de eficiencia.

En este trabajo se toma como referencia es el modelo de Markowitz (1959), donde se busca solucionar el problema de maximizar la rentabilidad esperada y minimizar el riesgo del portafolio. Este modelo tiene una solución mediante programación cuadrática. A pesar de que el método de programación cuadrática es muy eficiente, una de sus limitantes es que solo se consideran restricciones lineales.

Al emplear un algoritmo genético de un objetivo para resolver el problema de optimización de portafolios es posible incluir restricciones no lineales. Por otra parte al emplear un algoritmo genético con optimización multiobjetivo se puede seleccionar un conjunto de resultados que se encuentren próximos al frente de Pareto en una sola ejecución.

Aunque existen muchos avances en el tema de optimización de portafolios de inversión empleando algoritmos genéticos, en esta tesis se utilizan datos actuales y reales de la Bolsa Mexicana de Valores, además de explorar sobre el enfoque del problema de un objetivo. Por otra parte, se hace un análisis de este problema de optimización en cuanto a las ventajas de un método de un algoritmo evolutivo multiobjetivo.

Contenido

Reconocimientos	VII
Resumen	VIII
Lista de tablas	XII
Lista de figuras	XIV
Capítulo 1. Introducción	1
1.1. Definición del problema	3
1.2. Objetivos	4
1.2.1. Objetivos específicos	5
1.3. Consideraciones importantes	5
1.4. Hipótesis	6
1.4.1. Hipótesis específicas	6
1.5. Contribuciones	6
1.6. Organización de tesis	7
Capítulo 2. Antecedentes	8
2.1. Teoría moderna de portafolios	8
2.1.1. Rentabilidad de los activos	10
2.1.2. Riesgo de un activo	12
2.1.3. Rentabilidad del portafolio	12
2.1.4. Riesgo del portafolio	13
2.1.5. Frontera eficiente	13
2.1.6. Restricciones	14
2.2. Programación cuadrática	15
2.3. Optimización multiobjetivo	16
2.3.1. Dominancia de Pareto	16
2.3.2. Conjunto de óptimos de Pareto	17
2.3.3. Frente de Pareto	17
2.4. Algoritmos genéticos	17

2.4.1.	Función de aptitud	18
2.4.2.	Operadores de selección	18
2.4.3.	Manejo de restricciones	20
2.4.4.	Inyecciones de diversidad	20
2.5.	Algoritmos evolutivos multiobjetivo	21
2.5.1.	A Non-generational genetic algorithm for multiobjective optimization	23
2.5.2.	Nondominated Sorting Genetic Algorithm	24
2.5.3.	The Strength Pareto Evolutionary Algorithm	26
2.6.	Algoritmos genéticos multiobjetivo en TMP	28
2.7.	Conclusiones del capítulo	33
Capítulo 3. Optimización con restricciones lineales con AGs y PC		35
3.1.	Codificación del AG	35
3.1.1.	Representación del problema	35
3.1.2.	Función objetivo	36
3.1.3.	Variación de la función objetivo	36
3.1.4.	Cálculo de la rentabilidad y riesgo del portafolio	37
3.2.	Datos de prueba	37
3.3.	Selección de los operadores genéticos	39
3.4.	Ajuste de parámetros del AG	39
3.4.1.	Selección de probabilidad de cruce y mutación	40
3.4.2.	Ajuste de tamaño de población y número de generaciones	42
3.5.	Tamaño del problema	43
3.6.	Inyecciones de diversidad	45
3.6.1.	Ajuste de parámetros con inyecciones de diversidad	46
3.6.2.	Inyecciones de diversidad aleatorias y dirigidas	48
3.6.3.	Inyección de diversidad al inicio de la ejecución	49
3.7.	Restricción de distribución de acciones	49
3.7.1.	Programación cuadrática	50
3.7.2.	Algoritmo genético	51
3.8.	Restricción de pisos y techos	52
3.8.1.	Programación cuadrática	52
3.8.2.	Algoritmo genético	53
3.9.	Conclusiones del capítulo	55
Capítulo 4. Optimización con restricciones no lineales con AGs y PC		57
4.1.	Restricción de cardinalidad	57
4.1.1.	Programación cuadrática	58
4.1.2.	Algoritmos genéticos	60

4.1.3. Comparación entre PC y AGs	62
4.2. Costos de transacción	62
4.2.1. Algoritmos genéticos	63
4.3. Conclusiones del capítulo	65
Capítulo 5. Optimización con restricciones no lineales y diferentes niveles de riesgo	67
5.1. Portafolios eficientes con AGs variando aversión al riesgo	68
5.2. Optimización con restricción de cardinalidad variando aversión al riesgo	69
5.3. Optimización con costos de transacción variando aversión al riesgo . . .	73
5.4. Conclusiones del capítulo	74
Capítulo 6. Portafolios con restricciones no lineales empleando EMOs	76
6.1. Optimización sin restricciones con EMOs	77
6.1.1. NSGA 2	77
6.1.2. SPEA 2	77
6.1.3. vgGAMO	78
6.1.4. Comparación de algoritmos evolutivos multiobjetivo	79
6.2. Optimización con cardinalidad como restricción	83
6.3. Optimización con cardinalidad como objetivo	84
6.4. Optimización con costos de transacción	85
6.5. Optimización con costos de transacción como objetivo	88
6.6. Conclusiones del capítulo	89
Capítulo 7. Conclusiones	91
7.1. Conclusiones del AG de un objetivo	91
7.2. Conclusiones AG multiobjetivo	93
7.3. Trabajo Futuro	95
Bibliografía	97
Vita	100

Lista de tablas

3.1. Lista de 10 acciones para pruebas de concepto	38
3.2. Matriz de covarianza de 10 acciones para pruebas de concepto	38
3.3. Optimización del problema empleando programación cuadrática para problema de 10 acciones.	39
3.4. Pruebas del AG con diferentes operadores de selección	40
3.5. Resultados del AG con diferentes probabilidades de cruce y mutación .	40
3.6. Pruebas de AG con diferentes probabilidades de cruce y mutación para poblaciones de 800 individuos.	41
3.7. Resultados del AG con diferentes tamaños de población para 10 acciones.	42
3.8. Resultados del AG con diferentes tamaños de población para 25 acciones.	43
3.9. Resultados del AG con diferentes tamaños de problema.	44
3.10. Resultados del AG con diferentes tamaños de población y número de generaciones para 50 acciones.	44
3.11. Numero de evaluaciones promedio requeridas por el AG para obtener el valor óptimo.	45
3.12. Resultados del AG con diferente probabilidad de cruce y mutación para 50 acciones.	47
3.13. Resultados del AG con diferentes tamaños de población y numero de generaciones para 50 acciones.	48
3.14. Resultados del AG con diferentes tipos de inyecciones de diversidad para 50 acciones.	49
3.15. Resultados del AG con diferentes tipos de inyecciones de diversdad para 50 acciones.	49
3.16. Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.	51
3.17. Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.	52
3.18. Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.	53
3.19. Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con pisos y techos.	55

6.1. Evaluaciones del NSGA 2, SPEA 2 y vgGAMO para diferente número de evaluaciones de la función objetivo empleando métrica 6.4	80
6.2. Evaluaciones del NSGA 2, SPEA 2 y vgGAMO para diferentes número de evaluaciones de la función objetivo empleando métrica 6.5	82
6.3. SPEA 2 con diferente población inicial para problema con cardinalidad	84
6.4. SPEA 2 con diferente población inicial para el problema con costos de transaccion.	87

Lista de figuras

2.1.	Ejemplo de variación del precio de un activo en el tiempo	11
2.2.	Frontera eficiente.	14
2.3.	Proceso de optimización mediante algoritmos genéticos	18
2.4.	Comportamiento esperado cuando se emplea inyecciones de diversidad .	22
2.5.	Diagrama de flujo del SPEA [29]	27
4.1.	Pruebas con PC para diferentes valores de cardinalidad.	60
4.2.	Pruebas con AG para diferentes valores de cardinalidad.	61
4.3.	Pruebas con AG para diferentes valores de cardinalidad.	62
4.4.	Pruebas con AG para diferentes valores de cardinalidad.	64
4.5.	Evaluación de la función objetivo para diferentes valores de costos de transacción.	65
4.6.	Pruebas con AG para diferentes valores de cardinalidad.	66
5.1.	Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión lineal.	69
5.2.	Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial.	70
5.3.	Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial.	70
5.4.	Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	71
5.5.	Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	71
5.6.	Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	72
5.7.	Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	73
5.8.	Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	74
5.9.	Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.	75

6.1. NSGA 2 para un problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.	78
6.2. SPEA 2 con problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.	79
6.3. vgGAMO con problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.	80
6.4. Algoritmos EMO para un problema de 50 acciones, con 100,000 evaluaciones de la función objetivo.	81
6.5. Algoritmos SPEA 2 para problema de 50 acciones, con 250,000 evaluaciones de la función objetivo.	82
6.6. SPEA 2 con población inicial aleatoria y con PC para un problema de 50 acciones con cardinalidad 5	83
6.7. SPEA 2 con población inicial con PC para un problema de 50 acciones con cardinalidad 5,3 y 1	85
6.8. SPEA-II para problema de tres objetivos	86
6.9. SPEA 2 con población inicial aleatoria y con PC para un problema de 50 acciones con costos de transacción	87
6.10. SPEA 2 con población inicial con PC para un problema de 50 acciones con costos de transacción de 100 %, 10 % , 1 % y 0.1 %	88
6.11. SPEA 2 para problema de tres objetivos	89
7.1. Evaluaciones de individuos generados aleatoriamente	92
7.2. Evaluaciones de individuos generados en forma dirigida	93

Capítulo 1

Introducción

Un portafolio de inversión es un conjunto de activos de renta variable. En esta tesis se compara el método tradicional para optimizar un portafolio de inversión y un método alternativo de computación evolutiva, específicamente el uso de algoritmos genéticos, además de las restricciones que pueden ser integradas a cada uno de estos métodos. Los activos de renta variable se definen como los valores cuya principal característica es que la rentabilidad de la inversión se conoce hasta el momento de su vencimiento, dependiendo del desempeño de la entidad emisora y otras variables de mercado. La acción es el principal elemento dentro del grupo de activos de renta variable, y es la participación en el capital de una sociedad [1]. Comúnmente un portafolio no sólo está conformado por acciones de sociedades o compañías anónimas, sino también por bonos públicos y privados, certificados, títulos de participación, CETES, bienes inmuebles, metales preciosos y una variedad de instrumentos de inversión [7]; sin embargo en esta tesis nos referiremos a un portafolio como un conjunto de acciones principalmente de la Bolsa Mexicana de Valores (BMV). Actualmente la rentabilidad de las acciones puede variar en el orden de horas y por consiguiente el costo de las mismas. La rentabilidad varía con base en múltiples factores tales como [14]:

- Factores financieros, como los tipos de interés de la renta fija y del sistema crediticio, así como la expectativa sobre su evolución. También puede influir la situación de la economía del país, de los mercados financieros o de un sector productivo específico.
- Factores internos, que es principalmente la expectativa de crecimiento de la empresa. Los factores internos más tomados en cuenta son el crecimiento de las ventas o ingresos de la compañía, algunas veces denominado crecimiento bruto, el crecimiento de las ganancias de la compañía y la manera en que se administra la empresa y la calidad o ventaja de los productos o servicios que ofrece. Cuanta más alta sea la cotización más crecerá la expectativa de la empresa y por consiguiente crecerá el valor de sus acciones y viceversa, por lo que es una obligación de la empresa publicar sus estados financieros.

- Factores especulativos, son desequilibrios entre la oferta y la demanda, esto ocasiona una reacción en cadena, que se agrava conforme pasa el tiempo.

Los factores antes enlistados propician que los accionistas intenten adquirir o deshacerse de los activos propiciando variación en el precio de la acción. Los inversionistas dedicados a trasladar el dinero invertido, es decir, comprar y vender continuamente, son llamados especuladores. Por tanto, los activos comúnmente poseen dos características fundamentales: el valor esperado de retorno y el riesgo, que son datos fundamentales para la valoración de un activo. Comúnmente la rentabilidad de una acción se calcula por la suma de los dividendos y el valor de la compra-venta, aunque actualmente existen cálculos que permiten incluir datos como la especulación para el cálculo de la rentabilidad [1]. El riesgo se calcula a partir de la correlación entre el activo en sí con las demás opciones dentro del portafolio. Finalmente los inversionistas poseen por lo general más de una acción con el fin de maximizar el rendimiento y minimizar el riesgo de pérdida, a esto se le conoce como diversificación de portafolios [7]. Antes de realizar una inversión existen consideraciones importantes que se requieren para tomar una decisión, estas consideraciones normalmente influyen directamente en el rendimiento y riesgo, por ejemplo [5]:

- Definir el tamaño de la inversión para poder determinar un portafolio que satisfaga las capacidades del inversionista, ya que en muchas ocasiones un determinado activo sólo se puede adquirir con determinado dinero invertido en él, por tanto, es necesario definir exactamente el monto a invertir.
- Se deben determinar plazos en el que se puede mantener invertido el dinero, es decir, a plazos cortos (menor a un año), plazos medianos (entre uno y cinco años) o largo plazo (mayor de cinco años). Por ejemplo poder invertir una cantidad de capital en un portafolio que permita retirar la inversión a un determinado tiempo, para tener liquidez inmediata en caso de imprevistos.
- En diversas ocasiones el considerar riesgos altos no es una mala opción. El inversionista debe definir cuánto está dispuesto a asumir un riesgo que permita buscar un mayor rendimiento, a esto se le conoce como aversión al riesgo.
- La estrategia para diversificar el portafolio es una técnica de inversión que consiste en seleccionar distintos tipos de acciones, a fin de reducir significativamente el riesgo; sin embargo, esta tarea se puede extender en diferentes bolsas de valores, por lo que se debe elegir una estrategia de inversión que permita elegir activos en las diferentes bolsas de valores del mundo.
- Se debe considerar que existen costos en la transacción de un activo. Comprar y vender un número grande de acciones implica un costo elevado de transacción

en impuestos y los costos de compra venta de un activo determinado; además de considerar que en el mercado de la bolsa de valores, en muchas ocasiones, no se tiene la disponibilidad de adquirir las acciones al momento. Por lo que se debe cuidar el número de transacciones que se van a realizar para obtener un portafolio eficiente.

- En algunas ocasiones los diferentes instrumentos financieros (acciones) tienen un mínimo requerido para poder adquirirlos, por lo que se debe asegurar que el portafolio propuesto satisfaga esta propiedad, que es el precio mínimo fijado en cierta cantidad. A esto se le conoce como lotes mínimos de compra venta.

Las características antes enlistadas son catalogadas como restricciones [18]. Para este problema existen una gran cantidad de restricciones, que en algunas ocasiones se deriva una gran complejidad para su tratamiento, tanto que incluso no son tomadas en cuenta para tomar una decisión. En los métodos comunes de optimización (como es programación cuadrática), algunas restricciones no es posible incorporarlas para optimización de portafolios. Estas restricciones son propias del problema de optimización de portafolios cuyo enfoque matemático se explica en la capítulo 2.

1.1. Definición del problema

Cuando se seleccionan los activos de toda una gama de acciones que se cotizan alrededor del mundo, surge el problema de intentar seleccionar una combinación óptima de aquellos activos que tengan un mayor rendimiento y un menor riesgo de pérdida, éste es el problema simplificado de optimización de portafolios, para el cual se han desarrollado una serie de técnicas y herramientas capaces de calcular el rendimiento y riesgo de un conjunto de inversiones, como por ejemplo pronosticar el retorno de los activos o valor monetario esperado a un determinado plazo (EMV por sus siglas en inglés *Expected Monetary Value*), estimación del riesgo en base a datos históricos de las acciones (VaR por sus siglas en inglés *Value At Risk*), técnicas de diversificación de portafolio, etc. [5]. Este problema se introduce a través del modelo de Markowitz (MW).

El modelo de MW tiene solución mediante programación cuadrática (PC), sin embargo esta técnica tiene la desventaja de que restringe la solución a un único objetivo, ponderando la importancia del valor esperado de retorno o riesgo; esto conduce a experimentar con diferentes asignaciones de pesos para encontrar la frontera de eficiencia. Otro enfoque para encontrar la frontera de eficiencia es utilizar restricciones donde se minimiza el riesgo con un determinado nivel de retorno de un portafolio incluido como una restricción al problema.

La principal limitante para solucionar el modelo de MW mediante PC es que se incorporan sólo restricciones lineales. Existen algunas limitantes en este modelo, en-

tre ellas podríamos mencionar algunas restricciones que no es posible incorporar en el modelo como los costos de transacción e impuestos además se considera la perfecta divisibilidad de los acciones seleccionadas. Por otra parte no se proporciona alguna herramienta para que el inversor valore su actitud ante el riesgo y deduzca su función de utilidad, necesaria para la elección de su portafolio óptimo. La mayoría de estos inconvenientes pueden solucionarse con la introducción en el modelo de nuevas restricciones. Por ejemplo: introducir los costos de transacción en el modelo, asignándoles un límite de compra; considerar los rendimientos después de impuestos para un determinado inversor; añadir una nueva restricción que garantice un mínimo de liquidez para los activos.

En esta tesis se formula el problema desde un punto en el que se puedan encontrar varios resultados próximos al óptimo conjuntamente con aquellas restricciones que un inversionista pudiera necesitar. Esta formulación del problema debe permitir optimizar el desempeño de un portafolio de inversión y además integrar ciertas restricciones que comúnmente en un método convencional serían difíciles de incorporar. Estas restricciones incrementan la dificultad del problema, ya que limitan el espacio de búsqueda y el conjunto de resultados que satisfacen las restricciones, por tanto se vuelve más complejo, conforme se incrementa la cantidad de restricciones.

Es importante mencionar que el mercado accionario cambia rápidamente y esto se ve reflejado rápidamente, por lo que es indispensable utilizar técnicas avanzadas que permitan administrar las inversiones de forma eficiente. Actualmente en México existe una gran cantidad de capital económico, humano y tecnológico invertido en los mercados accionarios. El uso de tecnologías más eficientes permite a los inversionistas la utilización óptima de sus recursos. Existen actualmente muchos avances en el campo de optimización de portafolios de inversión, sin embargo, en algunas de las ocasiones los portafolios óptimos están contruidos sobre un modelo que ya tiene solución, esto significa que el portafolio se construye a partir de los mecanismos que pueden dar una solución al problema dadas las restricciones. Hay que hacer énfasis en la importancia de esta tesis, por la trascendencia que tiene tomar decisiones bajo restricciones que normalmente no se pudieran aplicar por la complejidad para obtener una solución.

1.2. Objetivos

El primer objetivo de esta tesis es optimizar portafolios de inversión utilizando algoritmos genéticos (AG) de un objetivo, esto es, utilizar el modelo MW y mediante un algoritmo genético, obtener el portafolio óptimo determinado por el nivel de riesgo seleccionado; posteriormente considerar las restricciones que permitan tomar una decisión para optimizar el portafolio de inversión. Por último, utilizar técnicas multiobjetivo y considerar las restricciones que permitan diseñar portafolios de una forma más

concisa y aplicada a más condiciones cercanas a las reales. En la etapa final del este desarrollo, los datos de entrada al proceso de optimización son las restricciones y los parámetros con los que se desea obtener los resultados. El proceso de optimización a través de un algoritmo genético consiste en encontrar un grupo opciones de portafolio que satisfagan las restricciones antes mencionadas (optimización multiobjetivo). Los resultados son el conjunto de opciones de selección de activos (frontera eficiente) y el porcentaje de la inversión en cada uno de ellos a partir del AG, además de proporcionar información que pudiera ser relevante para la toma de una decisión.

1.2.1. Objetivos específicos

Los objetivos específicos de la presente propuesta de tesis son los siguientes:

- Determinar y describir el grado de confianza logrado por un algoritmo genético comparado con el método de programación cuadrática.
- Evaluar y presentar los resultados utilizando datos reales de la Bolsa Mexicana de Valores (BMV). Si los resultados son buenos y el grado de confianza alcanzado en el objetivo específico anterior, se puede determinar si es conveniente optimizar un portafolio de inversión utilizando la técnica de optimización mediante algoritmos genéticos.
- Incorporar en la optimización de portafolio las siguientes restricciones y estrategias:

Costos de transacción. Considerar o penalizar realizar transacciones, ya que esto implica un costo al realizar una compra o una venta.

Aversión al riesgo. Delimitar el riesgo para maximizar el rendimiento o delimitar el rendimiento para minimizar el riesgo.

Estrategia de diversificación. Estrategia para diversificación del portafolio, porcentaje de acciones nacionales o internacionales.

Pisos y techos. Establecer lotes mínimos de compra-venta.

Cardinalidad. Establecer cardinalidad del portafolio, es decir establecer una optimización sujeta a un determinado número de acciones.

1.3. Consideraciones importantes

En las siguientes características está apoyado el desarrollo de la tesis, es decir, para que el proceso de optimización mediante AG tenga validez son necesarias una serie de requisitos que se explican a continuación:

Fuentes de información de acciones y base de datos. Se requiere construir un banco de datos basado en el comportamiento histórico de las acciones. Esta información es gratuita en diversos sitios web, sin embargo, es necesario construir una base de datos que ayude en las tareas de pronóstico y evaluación del riesgo. Para este proceso se utilizó una herramienta de código libre llamada “GimmeFreeData” [17] que ayuda a guardar el histórico del precio del cierre de una acción en un periodo de tiempo determinado.

Pronosticador del valor de rentabilidad y riesgo. Se requiere poseer un instrumento que a partir del comportamiento histórico de activos, pueda predecir un valor para la rentabilidad, por ejemplo el valor monetario esperado (EMV por sus siglas en inglés *Expected Monetary Value*) y una evaluación del riesgo, por ejemplo valor al riesgo (VaR por sus siglas en inglés *Value at Risk*).

1.4. Hipótesis

La hipótesis principal de la presente tesis es que utilizando algoritmos genéticos y técnicas de computación evolutiva multiobjetivo es posible encontrar la frontera eficiente donde estos resultados permitan seleccionar portafolios de inversión, siguiendo las estrategias y restricciones que comúnmente se plantean para formular portafolios.

1.4.1. Hipótesis específicas

- En optimización de portafolios con restricciones lineales y no lineales, emplear un AG de un objetivo dados los datos de prueba utilizados en este trabajo, se pueden obtener portafolios eficientes.
- Para optimización de portafolios con restricciones empleando un algoritmo evolutivo multiobjetivo dados los datos de prueba utilizados en la tesis, se pueden obtener diversos portafolios eficientes.
- Dada la representación del problema en el AG, es posible encontrar portafolios eficientes.
- Dadas las restricciones que definen al problema, para el AG es posible encontrar portafolios eficientes.

1.5. Contribuciones

Las contribuciones de esta tesis consisten en lo siguiente:

- Implementar un AG de un objetivo para el problema de optimización de portafolios.
- Proponer y probar un nuevo mecanismo de inyección de diversidad del AG para optimizar portafolios de inversión.
- Proponer algoritmos para implementar restricciones propias del problema de optimización de portafolios, para obtener soluciones válidas (que no violen la restricción) y aplicarlas en los AGs.
- Implementar una población inicial a los algoritmos evolutivos multiobjetivo que permita mejorar el desempeño para optimizar portafolios de inversión.
- Comparar el desempeño de tres algoritmos evolutivos multiobjetivo en base a métricas de evaluación específicas para este problema.

1.6. Organización de tesis

La organización de esta tesis se presenta de la siguiente forma.

En el capítulo 2 se muestran los conceptos financieros, como conceptos propios de AG y optimización estocástica, además se explican los antecedentes, en donde se tratan brevemente las investigaciones sobre el uso de los AGs y Algoritmos Multiobjetivo en el problema de optimización de portafolios de inversión.

El capítulo 3 se desarrolla las pruebas de concepto, en donde se analizan los parámetros del AG para optimizar de forma más eficiente un portafolio de inversión y se prueba con una variante del AG para mejorar el desempeño en el problema de optimización. Además se agregan al problema dos restricciones lineales (pisos, techos y estrategia de diversificación) y se realizan pruebas con el AG.

El capítulo 4 se agregan dos restricciones no lineales (cardinalidad y costos de transacción) que presentan aquellas restricciones que programación cuadrática no puede incorporar. Aquí se realizan una serie de pruebas con el AG de un objetivo para determinar si este puede optimizar un portafolio con dichas restricciones.

El capítulo 5 se presentan la pruebas relacionadas con aversión al riesgo, para determinar si el AG de un objetivo puede determinar el frente de Pareto; además se agregan restricciones no lineales haciendo énfasis en las que programación cuadrática no puede incorporar y que un AG si puede hacerlo.

El capítulo 6 presenta las pruebas de algunos de los principales algoritmos multiobjetivo aplicados a este problema. Al igual que en el capítulo anterior se integran restricciones no lineales para estudiar el comportamiento de estos algoritmos aplicados al problema de optimización de portafolios con restricciones no lineales.

El capítulo 7 presenta las conclusiones y algunas sugerencias para trabajos futuros.

Capítulo 2

Antecedentes

En este capítulo se presenta una descripción general del problema en donde se aplica esta investigación, se describe brevemente el funcionamiento de los mecanismos empleados para optimización como AG y algoritmo evolutivos multiobjetivo. En este capítulo se define el problema de optimización de portafolios de inversión y su solución mediante programación cuadrática, la definición de algoritmos genéticos y sus principales características, así como algunos conceptos para optimización multiobjetivo.

2.1. Teoría moderna de portafolios

La base de la teoría moderna de portafolios (TMP) es el modelo de Harry Markowitz que en 1959 desarrolló un modelo matemático para la selección óptima de un portafolio de acciones. Desde su aparición, el modelo de Markowitz ha sido el fundamento para la optimización de portafolios de inversión, dando lugar a múltiples desarrollos y derivaciones. A pesar de esto, cuando inicialmente surgió el modelo, su utilización en la práctica entre administradores de portafolios y analistas de inversiones no fue tan extensa como podría suponerse debido a la alta complejidad matemática del método. Al ser un problema de programación cuadrática, el algoritmo de solución era complejo y la tecnología no era tan avanzada como lo es en este momento [30].

Al poco tiempo William F. Sharpe propone una simplificación que consiste en suponer la existencia de una relación lineal entre los rendimientos de los activos; esto significa que podemos definir el riesgo del portafolio sin utilizar una matriz de covarianzas, simplificando en gran medida el cálculo para la selección del portafolio. Actualmente, se dispone de una gran cantidad de recursos computacionales y avances en los métodos de cálculo que facilitan la solución de problemas de programación cuadrática [30].

Markowitz desarrolló un modelo sobre la base del comportamiento del mercado y de los inversionistas, ya que estos últimos buscan la mayor rentabilidad del portafolio y el menor riesgo. Por tanto, el inversionista busca un portafolio eficiente, donde se maximiza la rentabilidad para cierto grado de aversión al riesgo, o de otra forma equivalente,

busca el menor riesgo posible para un nivel determinado de rentabilidad. El conjunto de porcentajes de compra de las acciones que resultan de un portafolio eficiente puede calcularse utilizando programación cuadrática. En la ecuación 2.2 se expone una forma de este modelo en una expresión matemática.

$$\min \sigma^2(R_p) = \min_x \left(\sum_{i=1}^n \sum_{j=1}^n x_i x_j \sigma_{ij} \right) \quad (2.1)$$

sujeto a:

$$E(R_p) = \sum_{i=1}^n E(R_i) x_i = V^*$$

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

$$0 \leq x_i \leq 1 \quad \forall i \in (1, 2, \dots, n)$$

Donde x_i es la porción del total del presupuesto del portafolio destinado al activo i , n es el número de activos evaluados, $\sigma^2(R_p)$ la varianza del portafolio p , y σ_{ij} es la covarianza entre los rendimientos de los valores i y j . $E(R_p)$ es la rentabilidad del portafolio p , $E(R_i)$ es la rentabilidad del activo i . De tal forma que variar el parámetro V^* obtendremos en cada caso un portafolio diferente. Al resolver el programa, el conjunto de proporciones x_i que minimizan el riesgo del portafolio, así como su valor correspondiente son el conjunto de pares $[E(R_p), \sigma^2(R_p)]$ o combinaciones rentabilidad-riesgo de todas los portafolios es denominado *frontera eficiente*.

Existen diversos enfoques en que se puede modelar este problema. A continuación se presenta una lista de los modelos más importantes [13].

- Problema 1. Minimización de Riesgo

$$\min_x (\mathbf{xHx}') \quad (2.2)$$

sujeto a:

$$r = a'x$$

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

Donde se minimiza la varianza con un valor de retorno específico r sujeto a restricciones lineales o no lineales.

- Problema 2. Maximización del valor esperado de retorno.

$$\begin{aligned} & \max_{\mathbf{x}} \mathbf{f}'\mathbf{x} && (2.3) \\ & \text{sujeto a:} \\ & r = \mathbf{xHx}' \\ & \sum_{i=1}^n x_i = 1 \end{aligned}$$

En este caso se maximiza el valor esperado de retorno con un valor de aversión al riesgo fijo.

- Problema 3. Maximización del valor esperado de retorno con aversión al riesgo.

$$\begin{aligned} & \max_{\mathbf{x}} \mathbf{f}'\mathbf{x} + k(\mathbf{xHx}') && (2.4) \\ & \text{sujeto a:} \\ & \sum_{i=1}^n x_i = 1 \end{aligned}$$

Se maximiza el valor esperado de retorno con cierto valor de aversión al riesgo k sujeto a restricciones lineales o no lineales.

2.1.1. Rentabilidad de los activos

Rentabilidad simple de un activo

La rentabilidad simple de un activo i en el tiempo j está definida por la ecuación 2.5 [1]:

$$R_{it} = \frac{D_{it} + P_{it} - P_{it-1}}{P_{it-1}} = \frac{D_{it} + P_{it}}{P_{it-1}} - 1 \quad (2.5)$$

donde R_{it} es la rentabilidad del activo i en el tiempo j , D_{it} son los dividendos del activo i en el tiempo j , P_{it} y P_{it-1} son los precios del activo i al inicio y término del periodo del tiempo j .

Rentabilidad continua de un activo

La rentabilidad simple es comúnmente con la que están más familiarizados los inversionistas [1] y este cálculo permite al inversionista tener una idea de la rentabilidad de algún activo. Sin embargo si utilizamos un conjunto de datos ya sea mensuales, diarios, etc. siempre será más conveniente utilizar rentabilidades continuas porque para

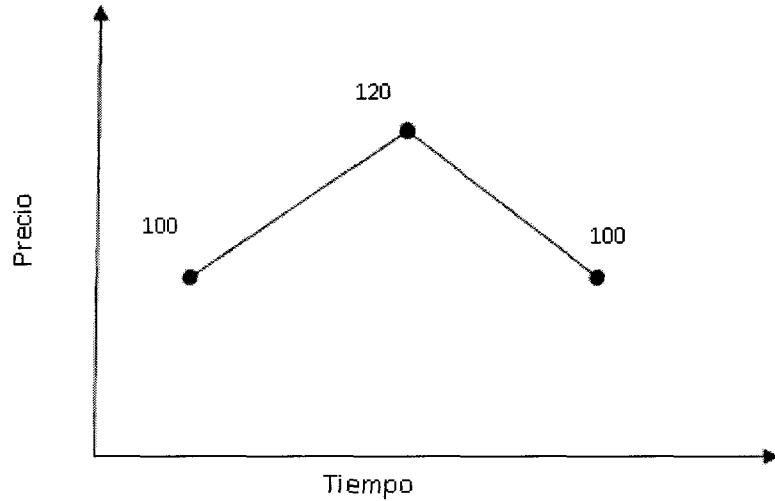


Figura 2.1: Ejemplo de variación del precio de un activo en el tiempo

calcular la rentabilidad anual de un activo se utiliza el producto de las rentabilidades como se muestra en la ecuación 2.6:

$$R_{total} = \left(\frac{P_1}{P_0} \times \frac{P_2}{P_1} \times \dots \times \frac{P_t}{P_{t-1}} \right) - 1 \quad (2.6)$$

Donde R_{total} es la rentabilidad anual o total de un activo, P_0, P_1, \dots, P_t es el precio de un activo del tiempo 0 al tiempo t . Si aplicamos esta fórmula en la variación del precio del activo, como se muestra en la figura 2.1, se interpreta que el precio del activo comienza en 100, sube el precio a 120 y nuevamente termina en 100, como mencionamos si utilizamos una rentabilidad simple y aplicamos el producto de las rentabilidades (ecuación 2.6) para obtener la rentabilidad anualizada se tendría una pérdida de 3.33 % lo cual parecería que hubiéramos perdido cuando en realidad la rentabilidad del activo es 0 porque el precio subió pero regreso al mismo precio inicial.

La rentabilidad continua de un activo consiste en obtener la tasa de crecimiento por la que está compuesta por un periodo de tiempo continuo. Por tanto la tasa de rentabilidad continua se calcula de la forma expresada en la ecuación 2.7:

$$R_{continua} = \left(\ln \frac{P_1}{P_0} + \ln \frac{P_2}{P_1} + \dots + \ln \frac{P_t}{P_{t-1}} \right) \quad (2.7)$$

donde $R_{continua}$ es la rentabilidad continua de un activo, P_0, P_1, \dots, P_t es el precio de un activo del tiempo 0 al tiempo t . Si aplicamos esta fórmula en la variación del precio del activo como se muestra en la figura 2.1, la rentabilidad continua es una tasa del 0% lo cual parecería acertado porque en realidad la rentabilidad del activo no subió ni bajo.

Media de la rentabilidad

Existen dos tipos de medidas utilizadas para calcular la media de la rentabilidad [4]: media aritmética y media geométrica (o media ponderada). La media aritmética de la rentabilidad es la suma de las rentabilidades en cada periodo dividido entre el número de períodos. La media aritmética es útil porque es una buena previsión del comportamiento en las rentabilidades futuras. La media geométrica de la rentabilidad es llamada también rentabilidad media ponderada por tiempo. La media geométrica es igual a la rentabilidad por período único que daría el mismo comportamiento acumulado como una secuencia de la rentabilidad actual, esto se expresa en la ecuación 2.9:

$$(1 + r_G)^n = \prod_{i=1}^n (1 + R_i) \quad (2.8)$$

$$r_G = \left(\prod_{i=1}^n (1 + R_i) \right)^{\frac{1}{n}} - 1 \quad (2.9)$$

2.1.2. Riesgo de un activo

Comúnmente el riesgo se mide a través de la volatilidad o dispersión de los rendimientos. Si asumimos que la rentabilidades de un activo se distribuyen de forma normal (curva de Gauss), con la desviación estándar podemos cuantificar al menos cuál será el intervalo en el que caerá nuestra rentabilidad o de otra forma equivalente es la probabilidad de obtener una rentabilidad determinada. La varianza es utilizada para medir la volatilidad de un activo como se muestra en la ecuación 2.10.

$$\sigma^2 = \sum_{i=1}^t [R_i - E(R_i)]^2 \quad (2.10)$$

donde R_i es la rentabilidad del activo en el tiempo i , y $E(R_i)$ es la rentabilidad esperada.

2.1.3. Rentabilidad del portafolio

La rentabilidad de un portafolio está en función de las rentabilidades individuales de cada activo del portafolio. El portafolio como se mencionó está conformado por un vector \mathbf{w} donde $\sum w_i = 1$; entonces para el activo i que no se encuentran en el portafolio contiene 0 en el valor w_i . La rentabilidad del portafolio será igual a la media ponderada de las rentabilidades de los activos.

$$E(R_p) = w_1 E(R_1) + w_2 E(R_2) + \dots + w_n E(R_n) = \sum_{i=1}^n E(R_i) w_i \quad (2.11)$$

2.1.4. Riesgo del portafolio

El riesgo de un portafolio está formado por la correlación que hay entre los activos del portafolio. El grado de riesgo de un portafolio se mide en función de la relación que existe entre los activos cuando en estos se produce una variación, como se presenta en la ecuación 2.12:

$$\sigma_{ij} = \sum_{k=1}^n (R_{ik} - E(R_i))(R_{jk} - E(R_j)) \quad (2.12)$$

El grado de relación entre los activos (σ_{ij} ecuación 2.12) nos dice en qué medida las acciones i y j cambian en el mismo sentido. Esto nos indica que cuando la covarianza es positiva y una acción sube la otra también sube, si la covarianza es negativa entonces cuando una acción sube la otra baja y si es cero los activos son independientes. Para interpretar la correlación entre los activos se utiliza coeficiente de correlación 2.13 que nos indica la magnitud de correlación entre los activos, es decir si una acción sube un 20% y la otra baja un 20% el cociente de correlación es -1 , caso contrario si las dos acciones subieran lo mismo el cociente de relación sería 1.

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \quad (2.13)$$

Por tanto el valor de riesgo R del portafolio p , como se muestra en la ecuación 2.14, utiliza la matriz H donde $H = [\rho_{ij}]$ y \mathbf{w} representa el vector de pesos asociados a cada acción.

$$R_p = \mathbf{wHw}' \quad (2.14)$$

2.1.5. Frontera eficiente

El frente de Pareto de las funciones rentabilidad y riesgo es conocida como la *frontera de eficiencia*. Este es un conjunto de soluciones óptimas que satisfacen las restricciones de modo que son superiores en algún sentido a las demás soluciones como se puede observar en la figura 2.2. Cada portafolio que se encuentra dentro del grupo de soluciones de la frontera de eficiencia es conocido como *portafolio eficiente*. Como se puede observar en la figura 2.2 existen dos regiones que son los portafolios factibles y no factibles estos últimos son aquellos portafolios que no existe forma de crearlos o de otra forma estos portafolios no son viables[27].

Para poder determinar una frontera eficiente se requiere de dos valores determinantes. Primero, el vector de retornos esperado, que proviene del análisis del retorno total para todos los activos que pueden estar en el portafolio, y segundo, de una matriz de riesgo, conocida como la matriz de varianzas y covarianzas de los retornos [18].

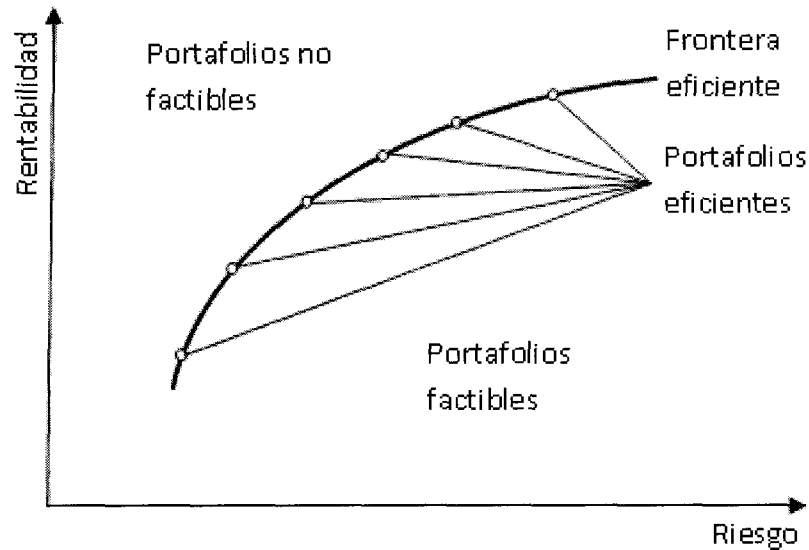


Figura 2.2: Frontera eficiente.

2.1.6. Restricciones

Para este problema existen una gran cantidad de restricciones que en algunas ocasiones deriva en una gran complejidad para su tratamiento, tanto que incluso no son tomadas en cuenta para tomar una decisión. Las restricciones son reglas que delimitan el problema. Las restricciones consideradas en este tesis son cuatro, pero existen una gran cantidad de restricciones que pueden ser incorporadas al problema, para efectos de separar estas restricciones se clasificaron en dos grupos diferentes: restricciones lineales que son las restricciones que programación cuadrática puede resolver y restricciones no lineales que son aquellas que no puede resolver.

Restricciones lineales

Restricción de pisos y techos. Esta restricción consiste en permitir ventas y compras en un máximo y mínimo establecido, por ejemplo si puede establecer vender hasta el 10% (piso) del valor de la lista en cualquier activo y además se puede limitar la inversión de un activo a 110% del valor de la lista (techo).

Estrategia de diversificación. Esta restricción consiste en una estrategia seguida por el inversionista donde intenta asegurarse de invertir en diferentes grupos de acciones, es decir se asegura que el capital invertido se encuentre distribuido dado ciertos porcentajes para ciertos grupos de acciones. Por ejemplo considere dos diferentes sectores: tecnología y energía con diferentes grupos de acciones. El inversionista considera que es necesario que su capital sea invertido en ambos grupos de acciones y asigna su capital en 70% para el sector de energía y 30% para el sector de tecnología. El pro-

ceso de optimización debe limitarse a que el portafolio óptimo alcance esta estrategia donde la suma del porcentaje invertido para el grupo de acciones del sector de energía sea exactamente el 70 % de la inversión y 30 % para el sector de tecnología.

Restricciones no lineales

Costos de transacción. Se refieren a todos los costos que implica comprar y vender acciones, en estos costos está implicado impuestos, pago por compra y venta, pago a terceros, etc. Se deben considerar que existen costos en la transacción de un activo porque comprar y vender un número grande de acciones implica un costo elevado de transacción, si se considera que en el mercado de la bolsa de valores no se tiene la disponibilidad de adquirir las acciones al momento se debe cuidar el número de transacciones que se van a realizar para obtener un portafolio eficiente.

Cardinalidad. Restricción de cardinalidad, esta restricción se refiere a cuando se decide que estratégicamente el portafolio se debe limitar a un número definido de activos.

2.2. Programación cuadrática

El problema de programación cuadrática es un modelo de optimización determinístico, con una función objetivo cuadrática convexa y restricciones lineales [25] que se formulará empleando la ecuación 2.15:

$$\min_x g(x) = \frac{1}{2} \mathbf{x} \mathbf{H} \mathbf{x}' + \mathbf{f}' \mathbf{x} \quad (2.15)$$

sujeto a:

$$\mathbf{A} \mathbf{x} \leq \mathbf{b}$$

$$\mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}}$$

$$\mathbf{L} \mathbf{b} \leq \mathbf{x} \leq \mathbf{U} \mathbf{b}$$

donde :

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]'$$

$$\mathbf{f} = [f_1 \ f_2 \ \dots \ f_n]$$

$$\mathbf{H} = \begin{bmatrix} h_{11} & \dots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{n1} & \dots & h_{nn} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

$$\mathbf{b} = [b_1 \ b_2 \ \dots \ b_m]'$$

$$\begin{aligned}
\mathbf{A}_{\text{eq}} &= \begin{bmatrix} A_{\text{eq}11} & \dots & A_{\text{eq}1n} \\ \vdots & \ddots & \vdots \\ A_{\text{eq}r1} & \dots & A_{\text{eq}rn} \end{bmatrix} \\
\mathbf{b}_{\text{eq}} &= [b_{\text{eq}1} \ b_{\text{eq}2} \ \dots \ b_{\text{eq}r}]' \\
\mathbf{Lb} &= [lb_1 \ lb_2 \ \dots \ lb_n] \\
\mathbf{Ub} &= [ub_1 \ ub_2 \ \dots \ ub_n]
\end{aligned}$$

La función \mathbf{xHx}' define una forma cuadrática convexa donde \mathbf{H} es simétrica. La matriz \mathbf{H} se define como negativa si el problema es de maximización y como positiva si el problema es de minimización. Esto significa que $g(x)$ es estrictamente convexa en \mathbf{x} para minimización y estrictamente cóncava para la maximización. En este caso las restricciones son lineales, lo que garantiza en espacio de solución convexo. Como $g(x)$ es estrictamente convexa y el espacio de solución es un conjunto convexo, se garantiza obtener un óptimo global.

2.3. Optimización multiobjetivo

Una gran parte de los problemas de optimización son multiobjetivo, esto significa que poseen dos o más funciones que se deben optimizar simultáneamente y que en gran parte de los problemas estas funciones están en conflicto entre sí. A pesar de esto muchos de estos problemas tienden simplificarse modelándose con una única función usando sólo una de las funciones originales y manejando las adicionales como restricciones. La noción más aceptada de “óptimo” en el entorno de problemas multiobjetivo es la propuesta originalmente por Francis Ysidro Edgeworth en 1881 y generalizada posteriormente por Vilfredo Pareto en 1896 [6].

2.3.1. Dominancia de Pareto

Se dice que hay dominancia cuando un vector $\mathbf{u} = (u_1, \dots, u_k)$ domina a otro $\mathbf{v} = (v_1, \dots, v_k)$ (denotado mediante $\mathbf{u} \preceq \mathbf{v}$) si y sólo si \mathbf{u} es parcialmente menor a \mathbf{v} (en el caso de minimización) ó \mathbf{u} es parcialmente mayor a \mathbf{v} (en el caso de maximización).

$$\forall i \in \{1, \dots, k\} u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$$

En otras palabras la expresión anterior nos dice que un vector \mathbf{u} domina a otro vector \mathbf{v} si en el vector \mathbf{u} existe un valor i por el cual sea mejor que en el vector \mathbf{v} .

2.3.2. Conjunto de óptimos de Pareto

Para todo problema multiobjetivo dado $\mathbf{f}(x)$ existen un conjunto de puntos que nos están dominados Ω , este es el conjunto de óptimos de Pareto P^* y se define como:

$$P^* = \{\mathbf{x} \in \Omega \mid \neg \exists \mathbf{x}' \in \Omega \mathbf{f}(x') \preceq \mathbf{f}(x)\}$$

La expresión anterior nos dice que el conjunto de óptimos de Pareto está compuesto por aquellas entradas \mathbf{x} que están dentro del grupo de las soluciones que no están en dominancia de Pareto (no dominados), tales que no existe un \mathbf{x}' que los supere en alguna de las evaluaciones de las entradas en el problema multiobjetivo.

2.3.3. Frente de Pareto

La expresión matemática para un problema multiobjetivo dado un $\mathbf{f}(x)$ y un conjunto de óptimos de Pareto P^* , el frente de Pareto (PF^*) se define como:

$$PF^* = \{\mathbf{u} = \mathbf{f} = f_1(x), \dots, f_k(x) \mid x \in P^*\}$$

En otras palabras el frente de Pareto son el conjunto de soluciones que no están dominadas o las soluciones que conforman el óptimo de Pareto.

2.4. Algoritmos genéticos

Los AGs son algoritmos de búsqueda que están inspirados en un proceso natural de selección y evolución [15]. El proceso de búsqueda en un algoritmo genético se basa en que los individuos más aptos son los que tienen mayores oportunidades de sobrevivir (tienen una mejor aptitud). Un algoritmo genético generalmente es utilizado como mecanismo de optimización. En el proceso de optimización mediante un AG se realizan cuatro etapas como se muestra en el diagrama de proceso en la figura 2.3.

El mecanismo descrito en la figura 2.3 menciona que existen cuatro etapas para optimizar con un algoritmo genético [23]. La primera consiste en codificar una función representativa del problema de optimización, en esta función el algoritmo genético busca un valor que maximice o minimice la función o busque algún elemento con ciertas características, etc. La segunda etapa del proceso de optimización es seleccionar los parámetros que tiene el algoritmo genético para que permitan optimizar la función. El tercer paso consiste en introducir la función objetivo y los parámetros del algoritmo genético para que éste se encargue de encontrar el resultado óptimo, el cuarto paso es seleccionar la mejor solución generado a través del algoritmo genético.

El algoritmo genético utiliza una población de individuos, donde cada individuo representa una solución al problema, cada individuo es evaluado por la función objetivo.

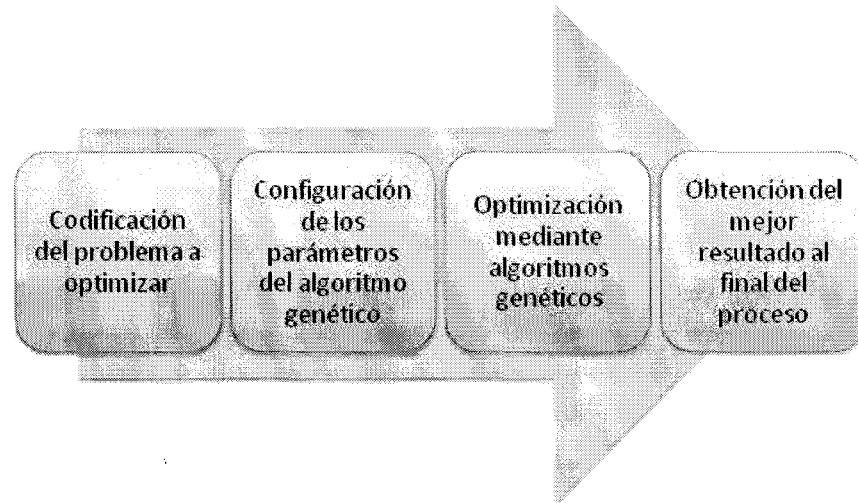


Figura 2.3: Proceso de optimización mediante algoritmos genéticos

Posteriormente los individuos se seleccionan en base a su evaluación y son cruzados y mutados para generar una población de nuevos individuos que en promedio mejora las evaluaciones anteriores; este proceso se repite hasta completar un número de iteraciones llamadas *generaciones*.

El AG para representar un individuo utiliza una cadena de valores comúnmente este conjunto de valores es binario (*cromosoma*) y al paso de las generaciones a través del proceso de evolución los valores se van modificando. El objetivo de un AG es que conforme el número de generaciones transcurre los individuos mejoren en cuanto a su evaluación. Esta mejora se debe a un proceso probabilístico donde se busca encontrar aquel valor o conjunto de valores que provean de mayor información en la optimización.

2.4.1. Función de aptitud

Cada individuo de una población es evaluado a través de una función que represente el problema (*función de aptitud*), esto con el fin de decidir cuáles individuos son más aptos. La función de aptitud arroja un número que representa el desempeño del individuo, este número se conoce como la aptitud del individuo.

2.4.2. Operadores de selección

El operador de selección es el mecanismo que elige a los individuos según su aptitud. De acuerdo a la evaluación de los individuos el operador selecciona a los mejores con el fin de utilizar el esquema de los individuos para formar nuevos individuos con una mejor evaluación. A partir de los individuos seleccionados por este operador se generan los individuos de la siguiente generación. Existe un gran variedad de métodos

de selección y hay dos forma de clasificar los métodos de selección [21]:

- Por fracción de la población que se produce para la siguiente generación, existen dos tipos:
 - Generacional. Se genera una población nueva en cada generación, la generación nueva está formada por el proceso de evolución. Se dice que hay *traslape generacional* cuando en la generación actual existen individuos que se mantuvieron de la generación anterior. Cuando un algoritmo genético generacional guarda a un individuo o a un número determinado de individuos para la siguiente generación se considera que es un algoritmo *elitista*.
 - No Generacional. Para este algoritmo no existe el concepto de generación, puesto que hay individuos que pueden permanecer durante mucho tiempo en la población. En lugar del término generación se ocupa el término ciclo. El AG de estado estable únicamente sustituye un individuo de la población o un porcentaje, por lo tanto existe *traslape generacional*. A la selección no generacional se le conoce usualmente como selección de estado estable.
- Por la forma en la que se utiliza la aptitud para la selección de individuos.
 - Proporcional. La probabilidades de ser seleccionado es proporcional a la aptitud del individuo.
 - Ranqueo. La probabilidad de que un individuo sea seleccionado es proporcional al rango o a lugar del individuo dentro de la población, el rango esta en función del la aptitud.
 - Determinística. Se tiene una regla predefinida de selección que determina que individuos son seleccionados.

A continuación se muestran algunos de los operadores de selección más importantes [21].

Selección de rueda ruleta

Este operador de selección es uno de tipo de proporcional donde los individuos son escogidos de manera proporcional con su aptitud. Este método es sencillo y fácil de implementar y consiste en simular el juego de una ruleta. Este mecanismo asigna a cada individuo una sección de la ruleta que es una fracción proporcional según su aptitud, posteriormente se genera un número aleatorio para conocer en qué valor de ruleta se selecciona. De esta forma los mejores individuos tendrán una fracción mayor de la ruleta, entonces se espera que estos tengan mayor probabilidad de ser escogidos esto en comparación con individuos menos aptos.

Selección universal estocástica

Este operador de selección es una variante de selección por rueda de ruleta y realiza selección proporcional. Al igual que rueda de ruleta se asigna una fracción de la ruleta a cada individuo según su aptitud. A diferencia de rueda de ruleta ahora se asignan n apuntadores igualmente espaciados, donde n es el tamaño de población. Al crear un número aleatorio (girar la rueda de ruleta) ahora se seleccionan aquellos individuos que se seleccionaron los n apuntadores creados.

Selección por torneo

La selección por torneo simula una competencia donde el individuo con mayor aptitud tiene mayor posibilidad de ganar el mayor número de competencias. Este método ha demostrado ser eficiente y también fácil de implementar. En este operador de selección es necesario definir el tamaño de torneo (m), esto quiere decir que las competencias realizadas en el proceso de selección tendrán m individuos:

- Los individuos de la población se mezclan y se ponen en una lista circular.
- Cada individuo participa en m torneos con los individuos que están en posiciones adyacentes a la suya. Los individuos que participan en un torneo se obtienen corriendo una ventana de tamaño m sobre la población.
- El ganador de cada torneo es el mejor de los individuos. A este individuo se le asigna una oportunidad reproductiva.

En el valor esperado, la selección por torneo de tamaño m asigna m copias al mejor individuo, $m/2$ a la mediana y cero al peor.

2.4.3. Manejo de restricciones

La mayor parte de los problemas requieren de restricciones, es decir no todo el conjunto de valores de una solución es válido y por lo tanto es necesario restringir el espacio de búsqueda. Existen cuatro métodos principales para poder manejar una restricción en algoritmos genéticos [22]: codificación, reinterpretación, funciones de castigo y letales. En esta tesis solo se manejan las restricciones como codificación que consiste en que en el proceso de codificación del problema se limite a generar individuos validos, es decir que gracias a la codificación sólo es posible generar regiones validas del espacio de búsqueda.

2.4.4. Inyecciones de diversidad

En el proceso de evolución del AG y dada las características del problema, la población del AG puede llegar a ser muy similar entre si y representar a una sola

solución parcial o a un máximo local, en otras palabras la población del AG puede llegar a perder su diversidad y por tanto el proceso de natural de evolución queda estancado en un valor no óptimo. Cuando la población es muy parecida entre si se dice que la población presenta convergencia este puede ser un criterio de terminación del algoritmo o de re inicialización de la población. Existen tres formas de medir convergencia [22]:

Convergencia genotípica. Esta convergencia mide que tan parecidos son los individuos sobre el cromosoma, es decir mide a nivel binario cuanto se asemejan los individuos.

Convergencia fenotípica. Esta convergencia mide que tan parecidos son los individuos sobre la representación de la solución, es decir mide a nivel de la cadena de valores que representan una solución y se verifica el nivel de igualdad entre los individuos.

Convergencia de aptitud. Esta convergencia mide que tan parecido son las evaluaciones de los individuos de la población. En esta medida de convergencia se desprecia el nivel de convergencia entre genotipo o el fenotipo.

Para evitar que la búsqueda sea limitada cuando la población presente convergencia, existen dos mecanismos de re inicialización que nos permiten mantener la diversidad:

Mutación alta. Este mecanismo consiste en que por algunas generaciones se aumente la probabilidad de mutación para obligar a la población tener diversidad.

Inyecciones de diversidad. Este mecanismo consiste en detectar cuando es que la población del algoritmo genético converge y en este momento remplazar una sección de la población por una nueva para mantener la diversidad en la población [22]. Como se muestra en la figura 2.4, el comportamiento esperado de este mecanismo consiste en que cuando la población presente convergencia la curva de la mejor evaluación encontrada tiene a quedarse en la misma evaluación hasta no tener una re inicialización de población que es cuando a través del proceso evolutivo la mejor evaluación tiende a tener mejores resultados, este proceso se repite sucesivamente.

2.5. Algoritmos evolutivos multiobjetivo

Los algoritmos evolutivos multiobjetivo (EMO) usan una combinación de técnicas para encontrar múltiples soluciones optimas de Pareto en una sola ejecución. Es posible incorporar de forma directa del concepto de optimalidad de Pareto en los algoritmo EMO [6]. Comúnmente se usa una clasificación y selección de soluciones no dominadas para mover una población hacia el frente de Pareto en un problema de optimización multiobjetivo. Este mecanismo fue llamado clasificación de Pareto (*ranqueo de Pareto*). La idea básica es encontrar el conjunto de individuos en la población que no son dominados por el resto de la población donde a las cadenas menos dominadas se asigna

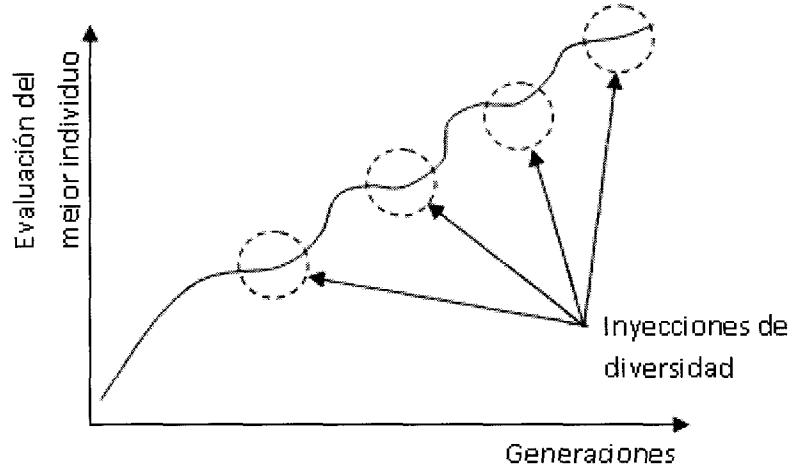


Figura 2.4: Comportamiento esperado cuando se emplea inyecciones de diversidad

el rango más alto.

El siguiente conjunto de cadenas que no son dominadas se eligen en base al resto de la población y se les asigna el siguiente más alto rango. Este proceso continúa hasta que la población esté debidamente clasificada. Goldberg también sugirió el uso de algún tipo de *conteo de nicho* técnica para mantener la diversidad de la población del AG a partir de la convergencia a un único punto de en los máximos de la función [8]. Un mecanismo de nicho tal que se comparte la aptitud permitiría al algoritmo evolutivo mantener todos los individuos a lo largo de la frontera no dominada. La expresión básica adoptada en la compartición de aptitud es la siguiente:

$$\text{Sh}(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_s}\right)^\alpha, & \text{si } d_{ij} < \sigma_s \\ 1, & \text{si no} \end{cases} \quad (2.16)$$

donde α indica la distancia entre las soluciones i y j , y σ_s es el conteo de nicho. Usando este mecanismo de nicho la evaluación de la función i es modificada como sigue:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^M \text{Sh}(d)} \quad (2.17)$$

en donde M es el número de individuos localizados en el vecindario de individuo i .

2.5.1. A Non-generational genetic algorithm for multiobjective optimization

Manuel Valenzuela Rendón y Eduardo Uresti Charre [26] propusieron un algoritmo genético no generacional multiobjetivo que incorpora selección basada en Pareto y evaluación de compartición. Específicamente la evaluación de un individuo está compuesto por dos partes: el conteo de dominación y el conteo de nicho. El conteo de dominación refleja el promedio ponderado del número de individuos por el cual un individuo es dominado. El conteo de nicho representa un valor ponderado del número de individuos que son cercanos a cierto individuo. En cada ciclo ambos valores son ajustados de acuerdo a los cambios en la población. Proponen un esquema de selección no generacional, y la evaluación del individuo es calculada incrementalmente, el objetivo del algoritmo es encontrar un conjunto de individuos coa aptados. La evaluación de un individuo es totalmente dependiente de los otros individuos en la población. El algoritmo tiene una evaluación en el cual la aptitud de los individuos en la población es ajustada en cada ciclo y el AG crea nuevos individuos a través del proceso de selección, cruce y mutación.

La evaluación de un individuo está basado en dos meritos: δ un promedio ponderado del número de individuos que tiene dominados a estos individuos; ω un promedio ponderado del número de individuos que se encuentran cercanos de acuerdo a una función de compartición. La evaluación de un individuo i está dada por:

$$f_i = c_\delta \delta_i + c_\omega \omega_i \quad (2.18)$$

las constantes c_δ y c_ω expresan la ponderación entre los dos objetivos del algoritmo genético. La evaluación de los individuos en la población es ajustada por comparaciones por parejas. Cada vez que un individuo es encontrado dominado por otro individuo se realiza una de estas comparaciones, el conteo de dominancia se ajusta de acuerdo a la siguiente ecuación:

$$\delta(t+1) = \delta(t) - k_\delta(t) + D(t) \quad (2.19)$$

donde $D(t)$ es 1 si el individuo es dominado en la comparación t , y 0 si no es. El conteo de nicho de un individuo es ajustado de acuerdo a una función de compartición de la distancia al individuo que es comparado con la siguiente ecuación:

$$\omega(t+1) = \omega(t) - k_\omega(t) + \text{Sh}(d) \quad (2.20)$$

donde d representa la distancia entre los dos individuos comparados y $\text{Sh}(d)$ es una función de compartición.

2.5.2. Nondominated Sorting Genetic Algorithm

El algoritmo *Nondominated Sorting Genetic Algorithm* (NSGA) es propuesto por Kalyanmoy Deb y N. Srinivas [24], difiere de un algoritmo genético en sólo la forma en la que funciona el operador de selección. Las operaciones de cruce y mutación operan como usualmente se hace en un AG. Antes de que el mecanismo de selección se ejecute, la población es clasificada según el nivel de no dominación. Los individuos no dominados presentes en la población primero son identificados, después todos estos individuos se asumen que constituyen el primer frente no dominado en la población, se asigna un valor de evaluación ficticio grande [24]. El mismo valor de evaluación es asignado para dar un potencial reproductivo equivalente a todos los individuos no dominados. Para mantener la diversidad de la población, esta clasificación de individuos es compartida con los valores ficticios de evaluación. Los métodos de compartición son logrados ejecutando la operación de selección usando valores de evaluación degradados que son obtenidos de dividir el valor original de evaluación de un individuo por una cantidad proporcional al número de individuos cercanos a éste (conteo de nicho). Esto origina múltiples puntos óptimos que coexisten en la población. Antes de compartición estos individuos no dominados son ignorados temporalmente para procesar el resto de la población en el mismo camino de identificar individuos para la segunda frontera no dominada [24]. Estos puntos no dominados se le asigna un nuevo valor de evaluación ficticio que es mantenido más pequeño que la mínima evaluación ficticia compartida de la frontera anterior. Este proceso se continúa hasta que la población entera es clasificada en muchos frentes como se muestra en la algoritmo 1.

El algoritmo NSGA-II es una variación del algoritmo de NSGA propuesto por Kalyanmoy Deb. El algoritmo NSGA ordena una población de tamaño de N de acuerdo al nivel de no-dominancia, cada solución debe ser comparada con cada otra solución en la población para encontrar si esta es dominada. Esto requiere $O(MN)$ comparaciones para cada solución donde M es el número de objetivos. Cuando este proceso se continúa para encontrar los miembros de la primera clase no dominada para todos los miembros de la población, la complejidad total es $O((MN)^2)$. En este escenario, todos los individuos en el primer frente no dominado son encontrados. Para encontrar los individuos del siguiente frente, las soluciones del primer frente son descartadas temporalmente y el procedimiento es realizado otra vez. El procedimiento se repite para encontrar las subsecuentes frentes [9].

El enfoque de NSGA-II es similar en principio al enfoque anterior, excepto que tiene una mejor estrategia de contabilización que es ejecutada para hacer un algoritmo más rápido y eficiente. En este enfoque cada solución de la población es verificada con un parte de la población para definir su dominación. Para empezar la primera solución de la población se almacena en un conjunto P' . Después de esto cada solución p (la segunda solución) es comparada con todos los miembros del conjunto P' uno por uno,

Algorithm 1 Algoritmo principal del NSGA

```
1:  $p \leftarrow \text{InicializaPoblacion}()$ 
2:  $gen \leftarrow 0$ 
3: while  $gen < \text{maxgen}$  do
4:    $front \leftarrow 1$ 
5:   if  $\text{clasificada?}(p) = \text{FALSE}$  then {Si la población está clasificada según su nivel de dominación}
6:      $p \leftarrow \text{indetificaNodominados}(p)$  {Identifica individuos no dominados}
7:      $p \leftarrow \text{asignaAptitud}(p)$  {Asigna a la población una evaluación ficticia}
8:      $p \leftarrow \text{comparticion}(p)$  {Asigna una función de compartición a la evaluación ficticia}
9:      $front \leftarrow front + 1$ 
10:  else
11:     $p \leftarrow \text{reproduccion}(p)$  {Se hace reproducción por torneo de acuerdo a evaluación ficticia}
12:     $p \leftarrow \text{cruce}(p)$ 
13:     $p \leftarrow \text{mutacion}(p)$ 
14:  end if
15:   $gen \leftarrow gen + 1$ 
16: end while
```

si la solución p domina cualquier miembro q de P' , entonces la solución q es removida de P' . De esta forma los miembros no dominados del frente de los no-dominados es eliminado de P' . De otra manera la solución p es dominada por cualquier miembro de P' y la solución p es ignorada. Si la solución p es no dominada por cualquier miembro de P' , es almacenada en P' . Como el conjunto P' crece con las soluciones no dominadas cuando todas las soluciones de la población es verificada, lo miembros restantes de P' constituye el conjunto no dominado P' [9].

Algorithm 2 Procedimiento para encontrar los miembros no dominados del frente de Pareto en NSGA-II

```

1:  $P' \leftarrow \text{find} - \text{nondominated} - \text{front}(P)$ 
2:  $P' \leftarrow \{1\}$  {se incluye el primer miembro de  $P'$ }
3: for all  $p \in P \wedge p \notin P'$  do {Se selecciona una solución}
4:    $P' \cup \{p\}$  {se incluye  $p$  en  $P'$  temporalmente}
5:   for all  $q \in P \wedge q \notin P'$  do {compara  $p$  con los otros miembros de  $P'$ }
6:     if  $p \preceq q$  then {si  $p$  domina un miembro de  $P'$ }
7:        $P' \leftarrow P' - \{q\}$  {se elimina  $q$  en  $P'$ }
8:     else if  $q \preceq p$  then {si  $p$  es dominado por un miembro de  $P'$ }
9:        $P' \leftarrow P' - \{p\}$  {se elimina  $p$  en  $P'$ }
10:    end if
11:  end for
12: end for

```

2.5.3. The Strength Pareto Evolutionary Algorithm

El algoritmo *The Strength Pareto Evolutionary Algorithm* (SPEA) es propuesto por Eckart Zitzler and Lothar Thiele [29] y se propone un nuevo enfoque de optimización multiobjetivo que es el algoritmo evolutivo de fortaleza de Pareto. Este algoritmo evolutivo multiobjetivo utiliza:

- Almacenamiento de soluciones óptimas de Pareto encontradas hasta el momento
- Utiliza el concepto de dominancia de Pareto para ponderar y asignar un valor de evaluación escalar a los individuos, y
- Emplea *clustering* para reducir el número de soluciones no dominadas guardadas sin destruir las características del frente óptimo de Pareto.

Por otro lado, el algoritmo SPEA es único en cuatro aspectos:

1. Combina tres técnicas en un único algoritmo.

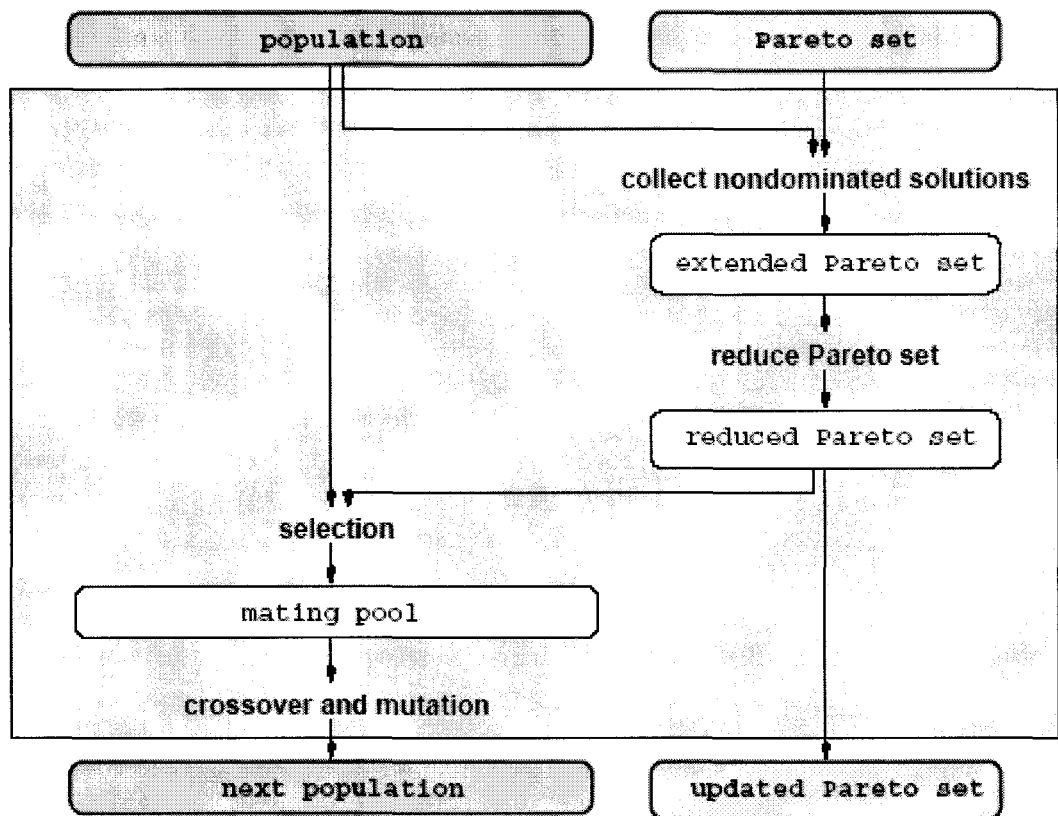


Figura 2.5: Diagrama de flujo del SPEA [29]

2. Los individuos no dominados son guardados en un frente de Pareto externo, diferente a la población del AG.
3. La evaluación de un individuo es determinada de las soluciones guardadas en el conjunto sólo del frente de Pareto externo.
4. Un nuevo método de conteo de nicho es propuesto para preservar la diversidad de la población; este método es basado en Pareto y no requiere un parámetro de distancia.

En la figura 2.5, se muestra el proceso que sigue el algoritmo SPEA. Primero el conjunto Pareto externo es actualizado: todos los individuos no dominados en la población son copiados al conjunto de Pareto externo y las soluciones dominadas son removidas de él. Si el número de soluciones almacenadas externamente excede un máximo dado, se realiza una selección por *clustering*.

Contrario a otros algoritmos EMO donde se soluciona el problema de la diversidad a través de una función de compartición (*sharing*) que requiere un valor de distancia.

En este algoritmo se plantea una función que depende del valor de dominancia de Pareto para mantener múltiples nichos estables. Esta función cumple con el objetivo de distribuir la población uniformemente cerca del frente óptimo de Pareto.

El principio de este método es el siguiente: En un primer paso cada solución en el conjunto de Pareto se le asigna un valor real $s \in [0, 1)$ llamado fortaleza, s es proporcional al número de miembros cubiertos en la población. La fortaleza de una solución de Pareto es al mismo tiempo su evaluación. En el segundo paso los individuos en la población son clasificados con respecto a las fortalezas. Esto es que para cada evaluación de las soluciones del frente de Pareto externo se añade 1 para garantizar que las soluciones de Pareto sean similares entre sí (para tener las mismas oportunidades reproductivas) y obtener el valor de la función f donde $f \in [1, N)$ y N denota el tamaño de la población.

El algoritmo SPEA2 [28] se creó como una alternativa que elimina las debilidades potenciales de su predecesor (SPEA). En particular las principales diferencias entre SPEA II y SPEA son:

- Una mejora de esquema para la asignación de aptitud que toma para cada individuo cuántos individuos domina y cuántos son dominados por él.
- Una técnica de estimación de densidad del vecino más cercano.
- Un nuevo método de truncamiento de archivo que garantiza que se preserve la soluciones de la frontera.

En contraste con el SPEA, el algoritmo SPEA2 usa una estrategia de asignación de aptitud fija que incorpora información de densidad. Además el tamaño del frente de Pareto externo llamado *archivo* es fijo. Siempre que el número de individuos no dominados sea menor que el tamaño predefinido del archivo es llenado con los individuos no dominados; con SPEA, el tamaño del archivo puede variar en el tiempo. Además la técnica de *clustering*, que es invocada cuando el número individuos no dominados excede el límite el archivo, es remplazada por un método alternativo de truncamiento que es similar en las características pero no pierde puntos de la frente de Pareto. Finalmente otra diferencia del SPEA es que sólo miembros de la población participan en el proceso de selección de apareamiento; en el SPEA 2 se utilizan los individuos del archivo y los individuos de la población. El proceso que realiza el algoritmo SPEA2 se describe en el algoritmo 3.

2.6. Algoritmos genéticos multiobjetivo en TMP

Heli Orelma en su trabajo de maestría [20], estudia la administración sistemática de un gran conjunto de patentes para la optimización de portafolios de inversión. El

Algorithm 3 Procedimiento general de SPEA-II

Require: N tamaño de población \bar{N} tamaño del archivo T máximo número de generaciones

Ensure: A conjunto de soluciones no dominadas

- 1: $P_0 \leftarrow \text{InitiazationPopulation}(N)$ {Genera un población inicial P_0 de tamaño N }
 - 2: $\bar{P}_0 \leftarrow \text{EmptyArchive}(\bar{N})$ {Crea un archivo vacío \bar{P}_0 (conjunto externo)}
 - 3: $t \leftarrow 0$ {se inicializa el número de generaciones en 0}
 - 4: $P_t, \bar{P}_t \leftarrow \text{AssignFitness}$ {Calcula los valores de aptitud para los individuos en P_t and \bar{P}_t }
 - 5: $P_{t+1} \leftarrow \text{nondominated}(P_t, \bar{P}_t)$ {Copia los individuos no dominados de P_t y \bar{P}_t a P_{t+1} }
 - 6: **if** $|P_{t+1}| > \bar{N}$ **then** {Si el tamaño de P_{t+1} excede \bar{N} }
 - 7: $P_{t+1} \leftarrow \text{TruncationOperation}(P_{t+1})$ {Se reduce el tamaño de P_{t+1} por la operación de truncamiento}
 - 8: **else** {si \bar{N} es menor que el tamaño de $|P_{t+1}|$ }
 - 9: $P_{t+1} \leftarrow \text{FillOperation}(P_t, \bar{P}_t)$ {Se llena P_{t+1} con los individuos no dominados en P_t y \bar{P}_t }
 - 10: **end if**
 - 11: **if** $t \geq T$ **then** {Si ya se realizaron las T generaciones}
 - 12: $A \leftarrow \bar{P}_t$
 - 13: **return** A
 - 14: **end if**
 - 15: $\bar{P}_{t+1} \leftarrow \text{MatingSelection}()$ {Ejecuta selección por torneo binario con remplazo}
 - 16: $\bar{P}_{t+1} \leftarrow \text{Variation}()$ {Aplica recombinación y mutación al pool de selección}
 - 17: $t++$
 - 18: GOTO STEP 4
-

objetivo de su trabajo es explorar las principales características de un gran conjunto de patentes para la selección y administración sistemática de activos. En primer lugar, el estudio analiza la gestión de las patentes en su conjunto y posteriormente se estudian diferentes modelos matemáticos y métodos computacionales para comprender qué tipo de modelo y que algoritmo se necesita. Los procedimientos también ayudan a evaluar qué partes del problema de optimización de portafolio de inversiones necesitan una mejora adicional. En esta memoria de patentes para la optimización se hace mención de la importancia de los buscadores estocásticos, entre ellos los más importante son los algoritmos genéticos para resolver este tipo de problema.

Rubén Armañanzas y Jose A. Lozano [2] emplean diversas técnicas de búsqueda para optimización aplicado a portafolios de inversión. En su trabajo, el problema es tratado como un problema multiobjetivo pero con valor fijo de nivel de riesgo, además el problema de optimización no incorpora restricciones de riesgo o de rentabilidad, pero incorpora restricciones de cardinalidad. Utiliza técnicas bien conocidas de optimización que son búsqueda avara, recocido simulado y colonia de hormigas. Para el frente de Pareto en la experimentación utiliza datos de cinco índices, mostrando los diferentes comportamientos de los algoritmos antes mencionados. El problema de optimización utilizado requiere un agente externo que decida fijar los niveles aceptados de riesgo para cada objetivo y satisfacer el criterio. En los resultados muestra como para valores pequeños de cardinalidad el número de soluciones encontradas en el frente de Pareto es muy baja, sin embargo cuando la cardinalidad es alta son muchos los portafolios encontrados en el frente de Pareto. El enfoque de colonia de hormigas demuestra ser muy competitivo. A pesar de que este enfoque originalmente no fue diseñado para problemas multiobjetivo, ha encontrado la mayor parte de las soluciones, y puede explorar en gran medida el espacio de búsqueda donde los otros enfoques no han sido capaces de llegar.

Karl Doerner, Walter J. Gutjahr, Richard F. Hartl, Christine Strauss y Christian Stummer [11] proponen optimizar un portafolio empleando una meta heurística que pueda elegir entre diversos métodos de optimización. El objetivo de su trabajo es seleccionar la mejor cartera de proyectos de un determinado conjunto de inversiones. La toma de decisiones debe examinar periódicamente varios objetivos el procedimiento emplea dos fases, primero determina el espacio solución (es decir, Pareto-óptimo) y, a continuación, los portafolios que les permiten explorar en forma interactiva en el espacio de búsqueda. Una Meta-heurística se utiliza para proporcionar una selección entre la cantidad cálculo de tiempo necesario y la calidad de la aproximación de la solución. En este trabajo se presenta un enfoque empleando “colonia de hormigas de Pareto” (P-ACO por su siglas en inglés *Pareto Ant Colony Optimization*) como una optimización y especialmente meta-heurísticas para resolver el problema de selección de cartera, además se compara su desempeño con otros métodos heurísticos (es decir, de Pareto Recocido Simulado *PSA* y el algoritmo genético No dominado ordenado, *NSGA*) por medio de experimentos computacionales con problemas generados aleatoriamente

y reales. La calidad de la solución de P-ACO es mostrada basándose únicamente en un análisis benchmarks entre PSA Y NSGA. Para comprobar la calidad y el desempeño aplicaron los tres enfoques a 18 instancias de problemas aleatorios heterogéneos y una instancia de un problema con datos reales y en los experimentos PACO demostró la mejor eficiencia. La experiencia que generó este trabajo muestra que el problema para algunos casos determinados de portafolios eficientes (que se encuentra por enumeración completa) son a veces muy difícil de encontrar. Como trabajo futuro podría ser útil para analizar con más detalle estos casos difíciles de encontrar para los portafolios para obtener ideas para seguir la mejora del enfoque P-ACO.

Laura Diosan [10] emplea un enfoque multiobjetivo y se refiere al problema de optimización de portafolios como un conocido problema difícil financiero que ocurren en el mundo real. El problema consiste en la selección de un conjunto óptimo de los activos con el fin de minimizar el riesgo y maximizar los beneficios de la inversión. Para este problema sugiere un enfoque multiobjetivo y utiliza tres Algoritmos Evolutivos ya conocidos (NSGA 2, SPEA 2 y PESA) para la solución de los dos objetivos para el problema de optimización de cartera. Para representar el problema utiliza una forma de normalizar el vector de entrada para poder satisfacer la restricción de que la suma de los pesos del portafolio sea igual a 1 y no considera restricciones adicionales. Realiza varios experimentos numéricos utilizando datos mundo real. Los resultados demostraron que el algoritmo “PESA” supera significativamente a los otros dos algoritmos NSGA 2 y SPEA 2.

Ghada Hassan y Christopher Clack [16] mencionan en su publicación acerca de como los métodos multiobjetivo son ideales para la evolucionar un conjunto de soluciones de optimización de portafolio que abarcan una amplia gama de portafolios desde alta rentabilidad pero de alto riesgo hasta una cartera de baja rentabilidad y bajo riesgo, donde un inversionista puede elegir un punto de su elección en la frontera de riesgo-retorno. Sin embargo, no hay garantías de que una solución de bajo riesgo seguirá siendo de bajo riesgo si tomamos en consideración los cambios del medio ambiente, la posiciones relativas de las soluciones previamente identificadas se pueden alterar. Una solución de bajo riesgo puede cambiar a una solución de alto riesgo y vice versa. Se plantea medir la solidez de programación genética Multiobjetivo (MOGP) para probar el algoritmo de SPEA2 ya que es de vital importancia en el contexto del mundo real de problema de optimización de portafolio. Este trabajo explora la solidez en este contexto, ofrece nuevas definiciones y una medida estadística para cuantificar la robustez de la solución. Una nueva medida de solidez es incorporada en la función de aptitud del algoritmo MGOP para evolucionar hacia soluciones más robustas. Este nuevo sistema (“R-SPEA2”) es comparado contra el algoritmo original y se presentan resultados. La solidez de una programación genética Multiobjetivo (MOGP) como algoritmo de SPEA2 es de vital importancia en el contexto del problema del mundo real de optimización de portafolios. Se demuestra que el SPEA2 (reconocido a ser uno de

los mejores algoritmos MOGP, y dando buenos resultados en problemas de optimización de portafolios) es, a pesar de ser excelente en otros aspectos, no puede garantizar soluciones robustas.

En el trabajo de tesis de Maestría de Salvador Fernando Castro [12], se hace un comparativo de tres enfoques, basados en algoritmos evolutivos multiobjetivo, para generar múltiples soluciones de portafolios de inversión. Utilizando una evaluación que utiliza datos reales tomados del mercado de valores mexicano. Para validar de manera cuantitativa el desempeño de los algoritmos analizados, se adoptaron métricas de desempeño reportadas en la literatura de optimización evolutiva multiobjetivo. Los algoritmos genéticos utilizados en dicho trabajo son NSGA-II, PAES, Micro AG además se propone para cada uno de ellos una variación en la forma de mutación que aumenta la probabilidad de obtener 0 en los cromosomas esto con el fin de aumentar el espacio de búsqueda. En el procedimiento de mutación alternativo los instrumentos financieros eliminados causan que las variables de decisión involucradas sean puestas a cero. Por lo tanto, el vector resultante de las variables de decisión estará “disperso”; y promoverá la diversidad de la población al aplicar los operadores de mutación y cruce, según corresponda. Con esta variación, los algoritmos AEMO pueden explorar con mayor facilidad el espacio de combinación de instrumentos financieros. Donde el algoritmo modificado del NSAGA-II mostro ser el algoritmo más rápido con restricciones de cardinalidad.

En el libro *Portfolio management with heuristic optimization* de Dietmar Maringer [19], se hace referencia al problema de optimización de portafolios con optimización de búsqueda ciega y consiste en dos partes. La primera parte son los fundamentos y se ocupa del modelo de optimización del portafolio (teoría de Markowitz), sus supuestos, algunos de los métodos junto con sus limitaciones de optimización. Además se presentan los conceptos básicos de varias técnicas de optimización con heurísticas aplicado para la optimización de problemas financieros con datos reales. La segunda parte son aplicaciones y contribuciones que abarcan distintos problemas de optimización financiera, entre ellos: los efectos de los costos de transacción (lineales, proporcionales y combinados); la diversificación en los pequeños portafolios; el efecto de la cardinalidad limitaciones a la eficiente línea de Markowitz; los efectos (y riesgos ocultos), de valor en riesgo cuando se utiliza la limitación de riesgo, el problema de factor de selección para el Arbitrage Teoría de Precios. Para probar los mecanismos utiliza datos reales de la bolsa principalmente del índice DAX y FTSE. El problema extendido de TMP se le agrega los costos de transacción y cardinalidad, en su trabajo propone una serie de mecanismos de optimización basados en búsqueda ciega, principalmente utilizando el método de recocido simulado. El modelo de solución para el problema utiliza una meta heurística que combina los principios de recocido simulado con estrategias evolutivas. Aplicando este algoritmo se tiene un método para incluir un número determinado de portafolios donde menciona que es realmente eficiente. También se menciona que este algoritmo es lo suficientemente flexible para permitir extensiones en el modelo de optimización

introduciendo restricciones adicionales como costos de transacción, impuestos límites inferiores y superiores para los pesos, medidas alternativas de riesgo y distribución de retornos, etc. Una de los principales aportaciones son las conclusiones que aporta al incluir costos de transacción en el modelo de optimización donde al introducirlos en el portafolio óptimo puede sufrir ciertas variaciones en su estructura. Incluso pequeños costos de transacción fijos pueden dirigir a una reducción substancial en el número de diferentes acciones incluidas en el portafolio, lo mismo ocurre cuando se utilizan costos de transacción proporcionales.

2.7. Conclusiones del capítulo

La teoría moderna de portafolios en una parte tiene solución mediante programación cuadrática, sin embargo esta técnica tiene la desventaja de que restringe la solución a un sólo objetivo, mediante una función de optimización lineal. El enfoque para encontrar la frontera de eficiencia empleando programación cuadrática es agregar restricciones donde se minimiza el riesgo con un determinado nivel de rentabilidad de un portafolio esto incluido como una restricción al problema.

Las restricciones consideradas en el planteamiento del problema son dos restricciones lineales y dos restricciones no lineales.

- Restricciones lineales

1. El portafolio puede estar sujeto a restricciones de pisos y techos y esto quiere decir que existe un mínimo o un máximo establecido para comprar proporcionalmente en determinada acción.
2. El portafolio puede estar restringido por una diversificación fija o por una estrategia de diversificación, esta restricción se presenta cuando el inversionista decide diversificar y destinar cierto porcentaje a un grupo de acciones.

- Restricciones no lineales

1. Restricción de cardinalidad, esta restricción se refiere a cuando se decide que estratégicamente el portafolio se debe limitar a un número definido de activos, por ejemplo con el fin de encontrar portafolios con poco riesgo pero pocos activos en el portafolio.
2. Costos de transacción, se refieren a todos los costes que implica comprar y vender acciones, en estos costos está implicado impuestos, pago por compra y venta, pago a terceros, etc.

Es importante mencionar que los algoritmos genéticos juegan un papel muy importante en el ámbito de optimización en gran medida porque ofrecen un método estocástico pero al mismo tiempo robusto. Los algoritmos genéticos han demostrado que

son un excelente mecanismo de optimización y gracias a los grandes avances que existen en los procesos de evolución, este tipo se ha convertido en el principal mecanismo de optimización en búsqueda ciega.

En el trabajo previo se tiene que se ha realizado investigación de métodos de búsqueda ciega principalmente algoritmo genéticos, recocido simulado y algoritmos evolutivos multiobjetivo aplicados de diversas formas al problema de optimización de portafolios de inversión. Existen muchos avances en el tema de optimización de portafolios de inversión empleando algoritmos genéticos y en alguno de ellos ya se emplean restricciones que en programación cuadrática no puede resolver, en esta tesis se utilizan datos actuales y reales de la bolsa mexicana de valores, además de explorar sobre los límites que tiene el algoritmo genético de un objetivo para encontrar portafolios óptimos. Es importante mencionar que a diferencia del trabajo previo en esta tesis se desarrolla una forma de implementar el AG de un objetivo para optimización del problema de portafolios, los resultados se comparan con programación cuadrática. Además en esta tesis se propone una variación al mecanismo de inyección de diversidad para optimizar portafolios de inversión. Por último se combinan el resultado de PC con un AG multiobjetivo para acelerar el proceso de optimización.

Capítulo 3

Optimización con restricciones lineales con AGs y PC

En este capítulo se presentan pruebas de concepto para demostrar que un AG optimiza portafolios de inversión con resultados similares a los que se obtiene con los métodos tradicionales como programación cuadrática (PC). El algoritmo genético utilizado en esta sección es el vgGA (por sus siglas en inglés *virtual gen Genetic Algorithm*)¹. Para el análisis de esta tesis se creó una codificación mencionada en el artículo de Laura Diosan [10]. Cabe resaltar que en este capítulo se hace referencia a optimización de un solo objetivo.

La metodología empleada para determinar los parámetros correctos en el AG consiste en primero determinar los operadores genéticos que ofrezcan mejores resultados, posteriormente ajustar las probabilidades de cruce y mutación en base a combinar sistemáticamente estas variables. Una vez fijadas las probabilidades, se estima el tamaño de población y número de generaciones en función del número de acciones a optimizar. Después se experimenta con diferentes tamaños de problema. Además se prueban variantes que pudieran mejorar el desempeño del algoritmo genético para optimizar portafolios. Por último se incorporan al AG restricciones de diversificación y pisos y techos que PC también puede incorporar.

3.1. Codificación del AG

3.1.1. Representación del problema

El problema de optimización de portafolios de inversión tiene como objetivo seleccionar el porcentaje óptimo de inversión sobre cada activo. Para este problema se utilizó un segmento del cromosoma para representar el porcentaje asignado a cada acción; el número asignado en cada segmento del cromosoma es un número entero,

¹Implementación del vgGA en MATLAB(R) por el Dr. Manuel Valenzuela (Valenzuela-Rendón, 2003). Centro de Sistemas Inteligentes, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey, Monterrey, Nuevo León, México.

el tamaño de cada segmento es de cinco bits, es decir por cada acción se asignan 5 bits al cromosoma. Posteriormente cuando se evalúa cada segmento del cromosoma se transforma en el porcentaje asignado a cada acción.

3.1.2. Función objetivo

En la función objetivo utilizada se crea un porcentaje a partir de la suma de los datos entrantes, a partir de esta función se creó una variante que se explica en la siguiente sección. La función objetivo aplicada para resolver este problema consiste en que cada individuo que es evaluado por la función objetivo se mide en valores que satisfagan las restricciones empleadas y además que representen un porcentaje del portafolio. En este capítulo el problema de optimización de portafolios se considera sin restricciones, por lo tanto cada peso se divide entre la suma de todos los datos entrantes para obtener un vector de porcentajes donde la suma de todos los componentes del cromosoma es igual a 1.

$$\mathbf{w} = \frac{\mathbf{x}}{\sum_{i=1}^n x_i} \quad (3.1)$$

donde \mathbf{w} es el vector de porcentajes del portafolio, \mathbf{x} es el vector de entrada a la función objetivo y n es el tamaño del vector \mathbf{x} . Como resultado de aplicar esta fórmula se tiene un vector de pesos \mathbf{w} donde la suma es igual a 1.

3.1.3. Variación de la función objetivo

En base a las múltiples pruebas realizadas se observa que los valores óptimos de los portafolios en gran parte están formados por pocas acciones, por lo tanto, el peso asignado a las acciones donde no se compra debe ser 0. Este factor aumenta la complejidad del algoritmo porque para que los porcentajes se hagan 0, se debe dividir 0, sobre el total o de otra forma el AGG buscaría número muy pequeños sobre el total, lo cual provoca un efecto de lento acercamiento al óptimo. Como se puede observar en la siguiente fórmula la variación incorpora a la función objetivo una normalización de 0 a 1 del vector de entrada antes de ajustar los pesos, de esta forma, en cada evaluación de la función objetivo mínimo se crea una evaluación con cero.

$$\mathbf{w}' = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \quad (3.2)$$

$$\mathbf{w} = \frac{\mathbf{w}'}{\sum_{i=1}^n w'_i}$$

donde \mathbf{w}' es el vector normalizado de porcentajes de los activos.

3.1.4. Cálculo de la rentabilidad y riesgo del portafolio

Para este procedimiento se utilizó la misma función que utiliza el método de programación cuadrática (ecuación 3.3). Esta función resta valor de la rentabilidad el valor del riesgo, que es el valor de multiplicar el vector de pesos por la matriz de covarianza.

$$f(x) = \min_{\mathbf{w}} \left(\frac{1}{2} \mathbf{w} \mathbf{H} \mathbf{w}' + \mathbf{w} \mathbf{f} \right) \quad (3.3)$$

donde \mathbf{w} es el vector de pesos, \mathbf{H} es la matriz de covarianza (riesgo) y \mathbf{f} es el vector de rentabilidad para cada acción. Sin embargo a diferencia de PC en el AG, para la función objetivo se agregó el valor del riesgo para, posteriormente, obtener diversos portafolios. Como se puede observar en la ecuación 3.4, k representa el valor de la aversión al riesgo.

$$f(x) = \min_{\mathbf{w}} \left(\mathbf{w} \mathbf{f} - k \left(\frac{1}{2} \mathbf{w} \mathbf{H} \mathbf{w}' \right) \right) \quad (3.4)$$

3.2. Datos de prueba

Los datos de prueba iniciales son datos históricos del precio de cierre de 50 acciones de la Bolsa Mexicana de Valores de enero del 2007 a abril del 2008. Para realizar las pruebas en este capítulo los datos históricos se dividieron en 5 subproblemas de 10, 15, 20, 25 y 50 acciones, donde cada subproblema contiene las acciones de los subproblemas anteriores. En otras palabras, las acciones utilizadas para el problema de 15 acciones son las mismas acciones utilizadas para el problema de 10 acciones más 5 acciones diferentes, además desde el primer subproblema de 10 acciones se incluye la mejor acción del mercado. Los activos seleccionados para el problema de 10 acciones se presentan en la tabla 3.1. En la tabla 3.2 se presenta la matriz de correlación calculada como se menciona en el capítulo anterior.

Una propiedad importante de este modelo de optimización es que cuando el valor utilizado para la aversión al riesgo es cero, es decir, no se toma en cuenta el riesgo, el portafolio óptimo resulta de adquirir el 100% del capital sobre la acción con mayor rendimiento. En caso contrario, cuando el valor de la aversión al riesgo es muy alto, esto quiere decir que se le da mucha importancia al riesgo, el portafolio tiende a seleccionar la mayor cantidad de acciones pero mínimamente correlacionadas. Por tanto para las pruebas realizadas en los experimentos se utilizan tres tipos de problemas con diferentes niveles de aversión al riesgo. En la tabla 3.3 se muestran los valores de rentabilidad

Indice	Simbolo	Riesgo	Retorno
1	AMXL.MX	0.1498	0.9795
2	ARA.MX	0.0868	1.0035
3	AXTELCPO.MX	0.4967	0.4322
4	CEMEXCPO.MX	0.0323	1.0018
5	CICSAB-1.MX	0.4966	0.4360
6	CMOCTEZ.MX	0.0254	1.0016
7	COMERCIUBC.MX	0.0184	1.0014
8	FEMSAUBD.MX	0.0565	1.0026
9	GEOB.MX	0.0300	1.0013
10	GFNORTEO.MX	0.0278	1.0011

Tabla 3.1: Lista de 10 acciones para pruebas de concepto

Indice	1	2	3	4	5	6	7	8	9	10
1	0.094	0.014	0.009	0.007	0.018	0.010	0.011	0.006	0.002	0.000
2	0.014	0.185	0.017	0.012	0.028	0.017	0.019	0.012	0.005	0.001
3	0.009	0.017	0.108	0.007	0.020	0.010	0.012	0.008	0.002	0.000
4	0.007	0.012	0.007	0.083	0.015	0.009	0.010	0.005	0.002	0.000
5	0.018	0.028	0.020	0.015	0.211	0.020	0.023	0.012	0.005	-0.002
6	0.010	0.017	0.010	0.009	0.020	0.115	0.014	0.007	0.002	0.002
7	0.011	0.019	0.012	0.010	0.023	0.014	0.123	0.008	0.003	0.001
8	0.006	0.012	0.008	0.005	0.012	0.007	0.008	0.078	0.001	0.000
9	0.002	0.005	0.002	0.002	0.005	0.002	0.003	0.001	0.061	-0.001
10	0.000	0.001	0.000	0.000	-0.002	0.002	0.001	0.000	-0.001	0.056

Tabla 3.2: Matriz de covarianza de 10 acciones para pruebas de concepto

Aversión al riesgo	Programación Cuadrática	
	Rentabilidad	Riesgo
Nulo(0)	17.5588	0.4298
Intermedio(100)	14.4762	0.2071
Grande(1000)	5.8647	0.1149

Tabla 3.3: Optimización del problema empleando programación cuadrática para problema de 10 acciones.

y riesgo para diferentes niveles de aversión al riesgo calculados a través de PC para el problema de 10 acciones.

La tabla 3.3 muestra los resultados de optimizar un portafolio de un tamaño de 10 acciones, los valores encontrados son tres diferentes portafolios eficientes para diferentes niveles de aversión al riesgo. Posteriormente los datos de la tabla 3.3 nos darán una comparación para saber si el algoritmo genético puede alcanzar estos resultados.

3.3. Selección de los operadores genéticos

Los operadores genéticos, explicados en el capítulo 2, son tres operadores de selección: torneo de tamaño 2, por ranqueo proporcional con escalamiento y rueda de ruleta. Como se puede observar en la tabla 3.4, el mejor operador de selección encontrado es el operador de torneo. En la tabla se muestran nueve corridas del algoritmo genético, tres para cada operador de selección; en cada corrida se busca un portafolio con una aversión al riesgo intermedia esto quiere decir que busca un punto intermedio en el frente de Pareto. En la tabla se muestran los resultados alcanzados en rentabilidad (Rent.) y en riesgo para cada corrida del algoritmo y en la última línea se muestra el promedio de las soluciones. Los parámetros del AG utilizados para estas pruebas son probabilidad de cruce de 0.9 y probabilidad de mutación de 0.1, con una población de 150 individuos y 250 generaciones, lo que nos da un total de 37,500 evaluaciones de la función objetivo. El resultado óptimo calculado a través de PC se muestra en la tabla 3.3, como se puede observar de los tres métodos sólo selección por torneo de tamaño dos, alcanza el valor óptimo calculado por PC para todos los casos.

3.4. Ajuste de parámetros del AG

Los parámetros del AG que se ajustaron fueron: probabilidad de cruce, probabilidad de mutación, tamaño de población y número de generaciones. Las pruebas realizadas consisten en variar primero la probabilidad de cruce y mutación para verificar con cuáles valores del algoritmo genético se alcanzan los mejores resultados. Una vez

	Torneo (2)		Estocastica Universal		Ruleta	
Aversión al riesgo	Rent.	Riesgo	Rent.	Riesgo	Rent.	Riesgo
Intermedio(100)	14.463	0.207	14.332	0.206	13.898	0.196
	14.463	0.207	14.278	0.205	13.554	0.192
	14.484	0.207	13.916	0.196	13.795	0.197
Promedio	14.47	0.207	14.175	0.202	13.749	0.195

Tabla 3.4: Pruebas del AG con diferentes operadores de selección

	Prob. Cruce = 0.99 Prob. Mutación = 0.01			Prob. Cruce = 1 Prob. Mutación = 0		
Aversión al riesgo	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Intermedio(100)	14.463	0.207	2760	13.385	0.189	2180
	14.463	0.207	4729	14.271	0.202	2794
	14.463	0.207	3862	14.557	0.209	2580
Promedio	14.463	0.207	3784	14.071	0.2	2518
	Prob. Cruce = 0.9 Prob. Mutación = 0.1			Prob. Cruce = 0.5 Prob. Mutación = 0.5		
Aversión al riesgo	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Intermedio(100)	13.769	0.192	18712	12.216	0.179	9995
	14.310	0.206	21591	12.653	0.181	23572
	14.473	0.207	18378	13.933	0.217	20401
Promedio	14.184	0.202	19560	12.934	0.192	17989

Tabla 3.5: Resultados del AG con diferentes probabilidades de cruce y mutación

ajustados los parámetros de cruce y mutación, se ajusta el número de generaciones y tamaño de la población.

3.4.1. Selección de probabilidad de cruce y mutación

Para el proceso de ajuste de probabilidad de cruce y mutación de este análisis sólo se mostraron cuatro pruebas que se consideran que son las más representativas de todo el esquema de pruebas. El esquema de pruebas realizado consiste en combinar sistemáticamente las probabilidades de cruce y mutación. Las pruebas más representativas donde se ajusta la probabilidad de cruce y mutación son mostradas en la tabla 3.5, se utiliza selección de torneo de tamaño 2, tamaño de población de 10 individuos por acción y 25 generaciones por acciones, para un problema de 10 acciones, lo que nos da un total de 100 individuos y 250 generaciones realizando un total de 25,000 evaluaciones de la función objetivo.

En la tabla 3.5 se muestran doce corridas del algoritmo genético, tres para cada

Aversión al riesgo	Prob. Cruce = 0.99 Prob. Mutación = 0.01			Prob. Cruce = 1 Prob. Mutación = 0		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Nulo(0)	17.5588	0.42982	25235	17.5588	0.42982	20914
	17.5588	0.42982	26864	17.5588	0.42982	25034
	17.5588	0.42982	25076	17.5588	0.42982	21571
Promedio	17.559	0.43	25725	17.559	0.43	22506
Bajo(10)	14.5305	0.20848	21683	14.5305	0.20848	19281
	14.5305	0.20848	22143	14.5305	0.20848	22818
	14.5305	0.20848	26906	14.5305	0.20848	23086
Promedio	14.531	0.208	23577	14.531	0.208	21728
Medio	7.1651	0.11741	27220	7.17	0.117	21581
	7.1651	0.11741	15801	7.17	0.117	14490
	7.1651	0.11741	21740	7.1651	0.11741	18962
Promedio	7.165	0.117	21587	7.165	0.117	18344
Alto	5.8779	0.11511	28203	5.8319	0.11512	23134
	5.9638	0.11513	30989	6.0151	0.11513	23163
	5.8779	0.11511	30143	5.8779	0.11511	21231
Promedio	5.907	0.115	29778	5.908	0.115	22509

Tabla 3.6: Pruebas de AG con diferentes probabilidades de cruce y mutación para poblaciones de 800 individuos.

probabilidad de cruce y mutación. Como se puede observar en la tabla 3.5 los mejores resultados obtenidos en la tabla se obtienen de una probabilidad de cruce del 0.99 y una probabilidad de mutación del 0.01, ya que en promedio se obtienen mejores resultados que en los demás configuraciones del algoritmo. En la tabla, la columna Eval. representa el número de evaluaciones realizadas para obtener el resultado. Debido a que en estas pruebas el AG no alcanza el óptimo (encontrado a través de programación cuadrática en la tabla 3.3), se efectuaron nuevas pruebas con tamaño de población mayor, presentándose los resultados en la tabla 3.6.

Como puede observarse en la tabla 3.6 los demás parámetros del AG utilizados son los mismos que en la tabla 3.5, sin embargo ahora con una población de 800 individuos (80 individuos por acción). Esta prueba nos muestra que con una población suficientemente grande las probabilidades de cruce y mutación no logran obtener aún así resultados óptimos, es decir no son una variable significativa. Sin embargo, cuando la población no es grande, con un mayor número de generaciones permite tener resultados óptimos y por tanto un menor número de evaluaciones de la función objetivo. En base a las pruebas realizadas, se tiene que en cuanto mayor sea la probabilidad de cruce se obtienen mejores resultados en un menor número de evaluaciones, por otra parte, en la

Aversión al riesgo	Tamaño de población 100 250 generaciones			Tamaño de población 250 100 generaciones		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Intermedio(100)	14.463	0.207	2760	14.4793	0.20718	14294
	14.463	0.207	4729	14.4689	0.20693	19299
	14.463	0.207	3862	14.4793	0.20718	14862
Promedio	14.463	0.207	3784	14.476	0.207	16152
Generaciones			37.837			64.607

Tabla 3.7: Resultados del AG con diferentes tamaños de población para 10 acciones.

probabilidad de mutación, en cuanto a menor sea puede contribuir a que la población no tenga convergencia en máximos locales.

3.4.2. Ajuste de tamaño de población y número de generaciones

En la tabla 3.7 se muestran las pruebas para verificar el tamaño óptimo de población y número de generaciones. Se busca determinar, en función del número de acciones, cuál es el tamaño de población y número de generaciones necesarios para alcanzar resultados óptimos. Las pruebas realizadas son con un AG con selección de torneo de tamaño dos, probabilidad de cruce en 0.99 y mutación en 0.01. En la tabla 3.7 se muestran seis ejecuciones el algoritmo con dos tamaños de población diferentes. Los resultados obtenidos son con una población de 250 individuos y un tamaño de población de 100 (15,000 evaluaciones de la función objetivo), es decir 25 individuos por acción y 10 generaciones por acción, el AG puede alcanzar mejores resultados que un algoritmo con 100 individuos de población y 250 generaciones (15,000 evaluaciones de la función objetivo). Para determinar el número promedio de generaciones en cada algoritmo se verificó el promedio del número de evaluaciones realizadas entre el número de individuos en la población, resultante en un promedio de 38 generaciones para un tamaño de población de 100 y 65 generaciones para un tamaño de población de 250.

En la tabla 3.8 se muestra la misma prueba que en la tabla 3.7, pero se elevó la dificultad del problema a 20 acciones. Al igual que en la tabla 3.7 los resultados en la tabla 3.8 no son óptimos. A partir de una población de 250 individuos, es decir 25 individuos por acción, el AG con los parámetros seleccionados puede ofrecer resultados muy similares al óptimo para un problema de un portafolio de 25 acciones.

En resumen las dos pruebas realizadas en las tablas 3.7 y 3.8 los resultados no son óptimos. Sin embargo, lo más importante que se demuestra en estas pruebas es que a medida que la población es más grande el número de generaciones necesarias para el obtener valores óptimos se reduce, a pesar de esto utilizar una población pequeña y un

Aversión al riesgo	Tamaño de población 200 500 generaciones			Tamaño de población 500 200 generaciones		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Intermedio(100)	13.6692	0.18304	36184	13.6871	0.18326	80377
	13.8034	0.18626	91834	13.6986	0.18391	82147
	13.8644	0.18796	58514	13.677	0.18298	69662
Promedio	13.779	0.186	62177	13.688	0.183	77395
Generaciones			310.89			154.79

Tabla 3.8: Resultados del AG con diferentes tamaños de población para 25 acciones.

número mayor de generaciones, mejora la explotación de la función objetivo porque en un menor número de evaluaciones se obtiene los valores cercanos al óptimo.

3.5. Tamaño del problema

En esta sección se describen las pruebas realizadas para determinar el número de evaluaciones que se requieren para que el AG pueda optimizar un problema de portafolios de inversión en función del número de acciones a optimizar. En la tabla 3.9 se muestran cuatro ejecuciones del AG para diferentes tamaños de problema, los parámetros del AG son los mencionados en la sección anterior (3.4) con probabilidad de cruce en 0.99, probabilidad de mutación en 0.01. El tamaño de población y número de generaciones en cada prueba es asignado en función del tamaño del problema, donde cada acción representa 10 individuos de la población y 25 generaciones, es decir para un problema de 10 acciones se utiliza un AGG con una población 100 de individuos y 250 generaciones y en un problema de 50 acciones se utiliza una población de 500 individuos y 1250 generaciones.

Los resultados obtenidos en la tabla 3.9 muestran cómo a medida que el tamaño de problema crece, el AG encuentra cada vez resultados más pobres, aún y cuando el número de evaluaciones se incrementa en forma lineal. Por lo que claramente el número de evaluaciones de la función objetivo necesarias para obtener el óptimo, no crece en forma lineal si no exponencial. Los resultados presentados en la tablas 3.7 y 3.8 son para optimizar un problema de 10 y 20 acciones respectivamente, sin embargo para optimizar un problema mayor (portafolio con 50 o más acciones) se encuentra el problema de que el AG con los parámetros mencionados no encuentra valores óptimos. Para optimizar un problema con 50 acciones se emplearon diversas pruebas aumentando tanto el tamaño de población así como aumentando el número de generaciones, como se puede ver en la tabla 3.10. Incluso para optimizar un problema de 50 acciones se utilizaron poblaciones mayores a 80 individuos por acción, es decir poblaciones mayores a 4,000 individuos, en

	Problema de 10 acciones			Problema de 15 acciones		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
PC	14.476	0.207	—	13.749	0.185	—
AG	14.463	0.207	2760	13.595	0.183	51529
	14.463	0.207	4729	13.785	0.186	28569
	14.463	0.207	3862	13.700	0.183	24767
Promedio	14.463	0.207	3784	13.694	0.184	34955
	Problema de 20 acciones			Problema de 25 acciones		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
PC	13.749	0.185	—	13.5316	0.1723	—
AG	13.917	0.189	68417	13.5491	0.17342	137356
	13.917	0.189	67315	13.2585	0.1654	122454
	13.678	0.183	91967	13.5112	0.17394	70446
Promedio	13.837	0.187	75900	13.44	0.171	110085

Tabla 3.9: Resultados del AG con diferentes tamaños de problema.

	Tamaño de poblacion 1600 con 150 generaciones			Tamaño de poblacion100 con 2000 generaciones		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
AG Prueba 1	14.4583	0.21015	228026	15.7301	0.26628	188912
AG Prueba 2	14.5019	0.21739	228662	15.464	0.28042	156351
AG Prueba 3	14.6094	0.22123	230115	15.6208	0.2619	157535
AG Promedio	14.52	0.216	228934	15.60	0.270	167599

Tabla 3.10: Resultados del AG con diferentes tamaños de población y número de generaciones para 50 acciones.

más de 250 generaciones, incrementándose así de 4,000 evaluaciones promedio para un problema de 10 acciones, a más de 600,000 evaluaciones de la función objetivo. Esto sin considerar que el tamaño del cromosoma también crece considerablemente por lo que optimizar este tamaño de problema o mayores implica un alto costo computacional.

La tabla 3.10 muestra los resultados del AG para un problema de 50 acciones, donde se prueba con una población de 1,600 individuos por 150 generaciones y una población de 100 individuos por 2,000 generaciones. En los dos casos el AG no alcanza el valor óptimo aún que en el proceso haya realizado más de 20,000 evaluaciones de la función objetivo (el valor óptimo es calculado a través de PC en la tabla 3.3). El principal razón por la cual el AG no puede lograr un valor óptimo es debido a que la población converge en un máximo local.

Con el objetivo de verificar el orden de crecimiento del número de evaluaciones

	10 acciones	25 acciones	50 acciones
Promedio de evaluaciones	3897	12042	**600,000
Generaciones promedio	40	120	200
Tamaño de poblacion	100	100	100
Total de evaluaciones	4000	12000	**600,000

Tabla 3.11: Numero de evaluaciones promedio requeridas por el AG para obtener el valor óptimo.

dependiendo del número de acciones, se estimó el número de evaluaciones necesarias para un problema 50 acciones. En la tabla 3.11 se muestran en el número promedio de evaluaciones que se requieren para llegar al óptimo en función del tamaño del problema, donde el signo ** significa que es un valor aproximado, ya que debido al costo computacional no es posible ofrecer un promedio. Podemos concluir en esta sección que el número de evaluaciones de la función objetivo crece de forma exponencial y además que para un problema grande (aproximadamente de 50 acciones o más) se tiene un alto costo computacional, donde probablemente un AG simple con una población pequeña no pueda determinar el valor óptimo, ya que tiene una alta posibilidad de que la población converja a un máximo local.

3.6. Inyecciones de diversidad

Como se puede observar en la sección anterior la complejidad del problema aumenta en forma exponencial y para lograr que un AG pueda optimizar el portafolio se requiere de más evaluaciones de la función objetivo, por lo tanto una solución es orientar el AG con métodos que ayuden a la optimización. En esta sección se propone una inyección de diversidad orientada, es decir, se busca que los portafolios que se añadan a la población tengan altas probabilidades de alcanzar buenos resultados o en su defecto, que contribuyan a ampliar del espacio de búsqueda. La inyección de diversidad consiste en detectar cuándo la evaluación de los individuos de la población del algoritmo genético converge (convergencia de fenotipo) y en este momento remplazar una sección de la población por una nueva población. Este mecanismo permite al AG diversificar la población cuándo se encuentre en un máximo local; de esta forma se mejora el desempeño y además es posible minimizar el número de evaluaciones de la función objetivo.

Cuando en un portafolio la aversión al riesgo es nula, es decir no se tiene importancia para el riesgo, el portafolio óptimo resulta de adquirir el 100% del capital sobre la acción que tenga mayor rentabilidad. Para que el AG mediante la función de evaluación propuesta pueda alcanzar el portafolio óptimo con aversión al riesgo nula, es necesario que en todos los elementos del vector de pesos tenga ceros a excepción del

valor correspondiente a la acción con mayor rendimiento, donde debe contener un uno. Esta característica del problema de optimización de portafolios aumenta la dificultad del problema, y es aquí donde se propone que los individuos inyectados a la población contengan en su mayoría ceros. El algoritmo 4 nos indica el tratamiento de cada nuevo individuo inyectado a la población.

Algorithm 4 Procedimiento para generar nuevos individuos con inyección de diversidad

Require: m size of individual i

Ensure: Ind vector of individual i

- 1: $Ind \leftarrow \{ \}$ {Se crea un individuo vacío}
 - 2: **for** $i = 1$ **to** m **do** {Para cada segmento del individuo}
 - 3: **if** $\gamma < \beta$ **then** {Con cierta distribución de probabilidad}
 - 4: $Ind(i) \leftarrow \gamma(\chi)$ {El individuo Ind contiene números aleatorios en el cromosoma}
 - 5: **else**
 - 6: $Ind(i) \leftarrow 0$ {El individuo Ind contiene ceros en el cromosoma}
 - 7: **end if**
 - 8: **end for**
 - 9: **return** Ind
-

El algoritmo 4 nos indica cómo se crean los individuos en la inyección de diversidad, donde γ representa una función que genera un número aleatorio y para cada segmento del nuevo individuo (Ind) se evalúa si éste toma un valor aleatorio o un valor en cero, esta decisión se ajusta a cierta probabilidad β donde si un número aleatorio es menor que β entonces el valor del segmento i del individuo Ind toma un valor aleatorio ajustado χ , de otra forma el valor del segmento i del individuo toma el valor de cero.

3.6.1. Ajuste de parámetros con inyecciones de diversidad

Debido a que las características del AG cambiaron notablemente, los parámetros ajustados en secciones anteriores varían. En las diversas pruebas realizadas se identificó que con inyección de diversidad el número de evaluaciones con respecto a la función objetivo se reducen notablemente, por lo que se realizó nuevamente un conjunto de pruebas para determinar los parámetros óptimos cuando se emplean inyecciones de diversidad. También se realizaron experimentos para determinar el porcentaje de población que se requiere reemplazar en la inyección de diversidad, al experimentar con un porcentaje pequeño de la población se encuentra que la insertar los nuevos individuos no influyen sobre el proceso evolutivo.

Parametros	Prob. Cruce = 0.99 Prob. Mutación = 0.01			Prob. Cruce = 1 Prob. Mutación = 0		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Prueba 1	11.902	0.143	67235	13.546	0.173	60999
Prueba 2	12.384	0.152	73265	13.426	0.170	63279
Prueba 3	12.062	0.144	69347	13.596	0.174	56722
Promedio	12.116	0.147	69949	13.523	0.173	60333

Tabla 3.12: Resultados del AG con diferente probabilidad de cruce y mutación para 50 acciones.

Ajuste de probabilidad de cruce y mutación

Si se tienen inyecciones de diversidad, los mejores parámetros de ajuste de probabilidad de cruce y mutación son diferentes a los parámetros sin inyecciones de diversidad. Para esta prueba se tomó en cuenta los valores de cruce y mutación seleccionados en la sección 3.4.1 que son 0.99 para probabilidad de cruce y 0.01 para mutación. Los mejores resultados alcanzados con un AG con inyecciones son de 1 para cruce y 0 para mutación. En esta prueba se optimiza un problema de 50 acciones que es el problema en el cual un AG sin inyecciones no puede determinar el óptimo en un número razonable de evaluaciones de la función objetivo, con un tamaño de población de 500 individuos que representan 10 individuos por acción y 150 generaciones, que son los parámetros seleccionados para un AG simple. Como se puede observar en la tabla 3.12 las probabilidades que presentan mejores resultados son 1 para cruce y 0 para mutación, ya que en promedio las corridas los resultados ofrecen valores cercanos al óptimo. Esto se debe a que en el AG con inyecciones de diversidad orientadas, si la población converge prematuramente se insertan nuevos individuos que potencialmente pueden ser mejores individuos, esto aumenta los límites de exploración del proceso del AG, mejorando notablemente el resultado. En contra parte cuando las probabilidades son de 0.99 y 0.01, el proceso de convergencia de la población es mucho más lento por tanto las inyecciones de diversidad son menores y la exploración del espacio es menor.

Ajuste de de tamaño de población y numero de generaciones

Como se mencionó en la sección 3.6.1, ahora los parámetros del AG cambian cuando se emplean inyecciones de diversidad. En el caso del tamaño de la población y el número de generaciones los valores son reducidos, debido a que con estas mejoras se requieren menos evaluaciones de la función objetivo. En la tabla 3.13 se exponen los resultados encontrados para un problema de 50 acciones, con probabilidad de cruce 1 y mutación en 0 y con 250 generaciones. De esta prueba podemos concluir que empleando inyecciones de diversidad dirigidas, el número de evaluaciones de la función objetivo

Poblacion	Tamaño de poblacion 500 generaciones 250			Tamaño de poblacion 250 generaciones 500		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Prueba 1	11.7483	0.14381	67608	13.5428	0.17265	33617
Prueba 2	11.0244	0.13189	58536	13.515	0.17187	66426
Prueba 3	11.794	0.14504	64932	13.526	0.17216	40995
Promedio	11.522	0.14	63692	13.528	0.172	47013
Generaciones Prom.			127.38			188.05

Tabla 3.13: Resultados del AG con diferentes tamaños de población y numero de generaciones para 50 acciones.

se reduce en gran medida, por ejemplo para determinar el óptimo sin inyecciones de diversidad se requiere alrededor de 600,000 evaluaciones, mientras que con inyecciones se requiere menos de 4,000. También se puede argumentar que al reducir el tamaño de población aumenta el número de generaciones requeridas, aunque la explotación de la función objetivo se mejora al emplear más generaciones con menor tamaño de población. En este punto se encuentra la disyuntiva de que al reducir el tamaño de población se aumenta la explotación de la función, pero se reduce la exploración de tal manera que si se reduce demasiado el tamaño de población en algún punto, el algoritmo no podrá determinar los valores óptimos.

3.6.2. Inyecciones de diversidad aleatorias y dirigidas

En esta prueba se comparan dos tipos de inyecciones, el primero método es el ya mencionado en esta sección que es una inyección de diversidad dirigida, donde los nuevos individuos tienen una mayor probabilidad de aportar algún valor que permita la exploración del espacio de búsqueda. La inyección de diversidad aleatoria consiste en generar individuos totalmente aleatorios sin ninguna heurística que permita al algoritmo ampliar su espacio de búsqueda. En la tabla 3.14 se muestra la comparación de estos los métodos, las pruebas se realizaron para un problema de 50 acciones, con una probabilidad de cruce de 1 y 0 de mutación, y 500 individuos de tamaño de población, es decir 10 individuos por acción y 500 generaciones. Las inyecciones dirigidas en promedio encuentran valores similares al óptimo. Los dos procedimientos en la misma cantidad de evaluaciones de la función objetivo, mientras que las inyecciones aleatorias no pueden determinan un valor óptimo.

Tipo de inyección	Inyección aleatoria			Inyección dirigida		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Prueba 1	11.7483	0.14381	67608	13.526	0.17216	97281
Prueba 2	11.0244	0.13189	58536	13.5473	0.1728	83867
Prueba 3	11.794	0.14504	64932	13.526	0.17216	92985
Promedio	11.522	0.14	63692	13.533	0.172	91378

Tabla 3.14: Resultados del AG con diferentes tipos de inyecciones de diversidad para 50 acciones.

Población Inicial	Poblacion aleatoria			Poblacion inicial asistida		
	Rent.	Riesgo	Eval.	Rent.	Riesgo	Eval.
Prueba 1	13.526	0.17216	97281	13.526	0.17216	39915
Prueba 2	13.5473	0.1728	83867	13.526	0.17216	55482
Prueba 3	13.526	0.17216	92985	13.5224	0.17206	87969
Promedio	13.533	0.172	91378	13.525	0.172	61122

Tabla 3.15: Resultados del AG con diferentes tipos de inyecciones de diversidad para 50 acciones.

3.6.3. Inyección de diversidad al inicio de la ejecución

Por último se analiza la alternativa de insertar una población inicial dirigida, es decir generar una población inicial conformada por la distribución propuesta por el algoritmo de diversidad dirigida. La tabla con esta prueba se muestra en 3.15, donde se utilizan los mismos parámetros utilizados en las tablas anteriores, la prueba compara el algoritmo que utiliza una población inicial aleatoria y una población inicial asistida. Los resultados son que los dos métodos alcanzan el óptimo sin problemas, sin embargo cuando existe una población inicial asistida el número de generaciones se reduce considerablemente de 6713 evaluaciones a 2269 evaluaciones para un problema de 50 acciones.

3.7. Restricción de distribución de acciones

Para definir una estrategia de optimización de portafolios de inversión en ciertas ocasiones es necesario definir una distribución “forzosa” de acciones dependiendo de variables de mercado. La restricción de distribución de acciones se refiere a restringir el capital para cierto grupo de acciones a un determinado porcentaje. Las acciones pueden ser catalogadas de diferentes formas, en estas pruebas se catalogaron en 2 grupos, 25 acciones nacionales y 25 acciones extranjeras, asignando un valor de 60% para las acciones nacionales y 40% para acciones extranjeras. Una característica importante de

esta restricción es que cuando el valor del riesgo es nulo, es decir no se le da importancia al riesgo, el portafolio óptimo resulta de adquirir un portafolio con las acciones de mayor rentabilidad en cada grupo, asignando a cada acción el porcentaje seleccionado a cada grupo.

3.7.1. Programación cuadrática

Programación cuadrática puede incluir la restricción de distribución de acciones agregando al problema de programación cuadrática, una restricción donde la suma de las acciones que pertenecen al grupo debe sumar el porcentaje seleccionado para ese grupo. Como se muestra en la ecuación 3.5, se modifica el problema original de programación cuadrática y se agregan nuevas restricciones al problema.

$$\min_x \frac{1}{2} \mathbf{xHx}' + \mathbf{x}'\mathbf{f} \quad (3.5)$$

sujeto a:

$$E(R_p) = \sum_{i=1}^n E(R_i)x_i = V^*$$

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

$$\sum_{i \in g_1}^n x_i = \alpha$$

$$\sum_{i \in g_2}^n x_i = (1 - \alpha)$$

$$0 \leq x_i \leq 1 \quad \forall i \in (1, 2, \dots, n)$$

donde la suma del vector \mathbf{x} debe ser igual a 1, además la suma de los activos del vector \mathbf{x} de las acciones que pertenecen al grupo uno (g_1) debe ser igual a un valor determinado α y la suma de los activos de las acciones que pertenecen al grupo dos (g_2) deben ser igual a $(1 - \alpha)$. Al resolver el programa cuadrático, el conjunto de proporciones x_i minimizan el riesgo del portafolios, su valor correspondiente al valor del riesgo y rentabilidad de portafolio son el conjunto de pares $[E(R_p), \sigma^2(R_p)]$ o combinaciones rentabilidad-riesgo.

En la tabla 3.16 se muestran los resultados de aplicar programación cuadrática para optimizar un portafolio de 50 acciones con la restricción de distribución de acciones donde se aplica 60% para acciones nacionales y 40% para acciones extranjeras, los valores encontrados son tres diferentes portafolios eficientes para diferentes niveles de aversión al riesgo. Posteriormente los datos de la tabla 3.16 nos darán una comparación para saber si es que el algoritmo genético puede alcanzar estos resultados.

Aversión al riesgo	Rentabilidad	Riesgo	Valor de la función
Nulo(0)	12.0316	0.2887	12.0316
Intermedio(100)	9.9993	0.1403	8.0310
Grande(1000)	3.8622	0.0637	0.00203

Tabla 3.16: Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.

3.7.2. Algoritmo genético

Para introducir la restricción de distribución de acciones, en el AG es necesario replantear la función objetivo que se diseñó en las secciones anteriores ya que ésta debe cumplir ahora con la restricción de distribución. Para que cualquier individuo introducido por el AG cumpla con las restricciones cada grupo de acciones se divide sobre el porcentaje de inversión que se seleccionó para ese grupo de acciones. En este caso los datos entrantes de las acciones que pertenecen a acciones nacionales se dividen sobre el 60% y las extranjeras sobre el 40%. En la ecuación 3.7 se presenta la función objetivo que utiliza el AG con restricciones de distribución de acciones.

$$w_i = \alpha_{x_i} \left(\frac{x_i}{\sum_{j \in g_i} x_j} \right)$$

$$f(x) = \min_{\mathbf{w}} \left(\mathbf{w}'\mathbf{f} - k\frac{1}{2}\mathbf{xHx}' \right) \quad (3.6)$$

donde g_i representa el grupo de acciones que pertenece el activo i , α_i representa el porcentaje de capital destinado para el grupo de acciones que pertenece la acción i . La expresión $\sum_{j \in g_{x_i}} x_j$ significa que es la suma de todos los pesos de las acciones de grupo de acciones g_i . f El vector de rentabilidades de cada acción, H es la matriz de covarianza entre los rendimientos de las acciones.

Para las pruebas con un algoritmo genético para portafolios con restricción de distribución de acciones se utilizó un algoritmo genético con inyecciones de diversidad. En la tabla 3.17 se realizaron tres ejecuciones del algoritmo con probabilidad de cruce en 1 y mutación en 0, 25 individuos por acción para un problema de 50 acciones, por lo que nos da una población de 1250 individuos. Como se puede observar en la tabla 3.17 el AG encuentra el mismo valor de la función objetivo que el valor encontrado por PC en la tabla 3.16, mas no encuentra los mismos portafolios, ya que los valores encontrados tienen diferentes rentabilidades y diferentes niveles de riesgo. En conclusión el AG con inyecciones para resolver un problema de optimización con distribución de

AG con inyecciones				
Aversión al riesgo	Rent.	Riesgo	Evaluaciones	Valor de la función
Intermedio(100)	10.004	0.140	88124	8.031
	9.998	0.140	173858	8.031
	9.997	0.140	162657	8.031
Promedio	9.999	0.14	141546	8.031

Tabla 3.17: Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.

acciones de un portafolio de 50 acciones requiere un gran número de evaluaciones de la función objetivo en comparación con un problema simple en donde no se incorporan restricciones.

3.8. Restricción de pisos y techos

Una estrategia utilizada para optimizar un portafolio de inversión es definir los límites que pueden tomar las acciones dependiendo de variables de mercado. La restricción de pisos y techos se refiere a restringir el valor mínimo y máximo para cada acción en un determinado porcentaje. En estas pruebas los límites seleccionados para las acciones son del 0.01 como el mínimo y 0.5 como el máximo valor asignado a cada acción. Una característica importante de esta restricción es que cuando el valor del riesgo es nulo, es decir no se le da importancia al riesgo, el portafolio óptimo resulta de adquirir un portafolio con el valor máximo que se le puede asignar a la acción con mayor rendimiento, y el mínimo a las acciones restantes, en caso de que no esté cubierto el 100% del capital, entonces se le agrega el capital restante a la segunda acción con mayor rendimiento, y así sucesivamente hasta completar el 100% del capital.

3.8.1. Programación cuadrática

Al igual que la restricción anterior en la sección 3.7.1 programación cuadrática puede incluir restricciones lineales y la forma en que se puede incluir esta restricción es modificar las restricciones de la definición del problema de programación cuadrática en donde únicamente se modifican los límites establecidos originalmente entre 0 y 1, en nuevos límites para cada acción dependiendo de la estrategia seleccionada. Como se muestra en la ecuación 3.8 se modifica el problema original de programación cuadrática.

Aversión al riesgo	Rent.	Riesgo	Valor de la función
Nulo(0)	10.5593	0.2335	10.5593
Intermedio(100)	9.1464	0.1363	7.2892
Grande(1000)	3.5433	0.0740	2736.2000

Tabla 3.18: Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con distribuciones de acciones.

$$\min_x \frac{1}{2} \mathbf{xHx}' + \mathbf{x}'\mathbf{f} \quad (3.7)$$

sujeto a:

$$E(R_p) = \sum_{i=1}^n E(R_i)x_i = V^*$$

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

$$lb_i \leq x_i \leq ub_i \quad \forall i \in (1, 2, \dots, n)$$

donde la suma del vector \mathbf{x} debe ser igual a 1, sin embargo ahora los límites para cada x_i están establecidos entre un límite inferior (\mathbf{lb}) y un límite superior (\mathbf{ub}). Al resolver el programa cuadrático, el conjunto de proporciones x_i minimizan el riesgo de la cartera y maximizan la rentabilidad sujeta a la restricción de pisos y techos.

En la tabla 3.18 se muestran los resultados de aplicar programación cuadrática para optimizar un portafolio de 50 acciones con la restricción de pisos y techos donde se aplica 1% como el mínimo límite de determinado activo y 50% para el máximo porcentaje del activo, los valores encontrados son tres diferentes portafolios. Posteriormente los datos de la tabla 3.18 nos darán una comparación para saber si el algoritmo genético puede alcanzar estos resultados.

3.8.2. Algoritmo genético

Para introducir la restricción de distribución de pisos y techos en el AG con inyecciones de diversidad, es necesario cambiar la función objetivo para que ahora los individuos introducidos a la función puedan cumplir con la restricción. La función objetivo normaliza la entrada del vector para posteriormente modificar aquellas acciones en las cuales el valor sobrepasa los límites establecidos, este proceso nos da como resultado un vector que no cumple con la restricción de que la suma del vector sea igual a uno, el resto necesario para obtener una suma que sea igual a uno se distribuye uniformemente sobre todo el vector en las acciones donde es posible sumar sin sobrepasar los límites. El algoritmo que utiliza el AG con restricciones de pisos y techos se expone en el algoritmo 5.

Algorithm 5 Normalización del vector de entrada del AG para restricción de pisos y techos

Require: x vector of inputs and ub vector of upper bounds and lb vector of lower bounds

Ensure: w weights output

```
1: FUNCTION NormalizaAcciones( $x, ub, lb$ )
2:  $\sum_{Max} \leftarrow 0$  {suma de los valores que excedan el maximo}
3:  $\sum_{Min} \leftarrow 0$  {suma de los valores que excedan el minimo}
4:  $resultado =$  {se crea la variable donde se va guardando la solución}
5: for  $i = 0$  to  $n$  do {se recorre el vector de entrada  $x$ }
6:   if  $x(i) > ub(i)$  then {si el valor de  $x$  sobrepasa el máximo}
7:      $\sum_{Max} \leftarrow \sum_{Max} + (x(i) - ub(i))$  {se suman los valores que sobrepasan}
8:      $resultado(i) \leftarrow ub(i)$  {se intercambia el valor por el limite superior}
9:   else if  $x(i) < lb(i)$  then {si el valor de  $x$  sobrepasa el mínimo}
10:     $\sum_{Min} \leftarrow \sum_{Min} + (lb(i) - x(i))$  {se suman los valores que sobrepasan}
11:     $resultado(i) \leftarrow lb(i)$  {se intercambia el valor por el limite inferior}
12:   else
13:      $resultado(i) \leftarrow x(i)$  {si no sobrepasa ningún límite se asigna el mismo valor}
14:   end if
15: end for
16:  $resultado = distribuyeMaximos(resultado, \sum_{Max})$  {La suma de los valores que sobrepasaron el límite superior se distribuye uniformemente sobre todo el vector en las acciones donde es posible sumar sin sobrepasar los límites superiores}
17:  $resultado = distribuyeMinimos(resultado, \sum_{Min})$  {La suma de los valores que sobrepasaron el límite inferior se distribuye uniformemente sobre todo el vector en las acciones donde es posible restar sin sobrepasar los límites inferiores}
18: return  $resultado$ 
```

AG con inyecciones				
Aversión al riesgo	Rent.	Riesgo	Evaluaciones	Valor de la función
Intermedio(100)	9.1488	0.1364	201834	7.2892
	9.1453	0.1362	274349	7.2892
	9.1488	0.1364	78979	7.2892
Promedio	9.148	0.136	185054	7.289

Tabla 3.19: Resultados de PC con diferentes niveles de aversión al riesgo para problema de 50 acciones con pisos y techos.

En el algoritmo 5 se recorre el vector \mathbf{x} para buscar que los pesos de las acciones se mantengan entre los rangos establecidos, aquellos valores que sobrepasan los rangos se suman en dos variables \sum_{Max} y \sum_{Min} y se asigna para el valor el límite establecido para esos valores, posteriormente los valores de \sum_{Max} y \sum_{Min} se distribuyen de forma uniforme en vector de salida a través de las funciones *distribuyeMaximos* y *distribuyeMinimos* respectivamente. Este pseudocódigo produce un vector normalizado que no sobrepasa los límites establecidos para optimizar a través del AG un portafolio de inversión con restricción de máximos y mínimos.

Para las pruebas con un algoritmo genético con inyecciones de diversidad para portafolios con restricción de pisos y techos se realizaron tres ejecuciones del algoritmo. En este caso se asignó a todas las acciones un mínimo de 1% y un máximo de 50%. En la tabla 3.17 las ejecuciones tienen probabilidad de cruce en 1 y mutación en 0, 25 individuos por acción para un problema de 50 acciones, por lo que nos da una población de 1250 individuos. Como se puede observar en la tabla 3.19 el AG encuentra el mismo valor de la función objetivo que el valor encontrado por PC en la tabla 3.18, también los portafolios encontrados tienen rentabilidades y niveles de riesgo muy similares. En conclusión el AG con inyecciones para resolver un problema de optimización de un portafolio de 50 acciones con restricción de pisos y techos requiere un gran número de evaluaciones de la función objetivo, en comparación con un problema simple en donde no se incorporan restricciones.

3.9. Conclusiones del capítulo

El objetivo primario de este capítulo es mostrar que mediante un AG es posible optimizar portafolios de inversión sin restricciones y con algunas restricciones lineales con resultados similares a los que se obtienen con PC. A pesar de que el número de evaluaciones que requiere el algoritmo para obtener el valor óptimo crece exponencialmente si se incrementa el tamaño del problema, un AG con los parámetros correctamente seleccionados, puede optimizar un portafolio e incluso determinar la frontera de eficiencia.

Por otra parte, en las pruebas realizadas con el tamaño de población y el número de generaciones se concluye que un tamaño de población mayor implica un grado mayor de exploración del espacio de búsqueda a costa de aumentar el número de evaluaciones, por otra parte numerosas generaciones implican mayor grado de explotación de la función.

La diferencia entre las dos funciones propuestas al inicio del capítulo es mínima (ligeramente mejor la función 2), sin embargo, este cambio no es determinante para que un AG obtenga un valor óptimo para un problema. En la primera sección de este capítulo se muestra lo complicado que puede ser un problema sin las métricas adecuadas, es decir, en este caso un AG simple sin inyección de diversidad requiere de un gran costo computacional para resolver problemas grandes (mayores de 50 acciones). La dificultad del problema crece a un ritmo exponencial, probablemente un AG simple con una población pequeña no pueda determinar el valor óptimo, ya que tiene una alta posibilidad de que converja en un máximo local. Podemos concluir que el número de evaluaciones de la función objetivo crece de forma exponencial y además que para un problema grande se tiene un alto costo computacional.

Al observar el comportamiento del modelo de optimización de un portafolio de inversión, se concluyó que para aumentar el grado de exploración del AG es necesario incluir nuevos individuos que difícilmente pueden ser generados por un proceso totalmente aleatorio, por esta razón se optó por un método que generase individuos donde se tenga una mayor probabilidad de que contengan ceros en el cromosoma. Este mecanismo se aplica al introducir nuevos individuos a la población cuando se detecte convergencia. Otra variante introducida es utilizar el mecanismo antes descrito para generar la población inicial en el AG. El resultado de aplicar este método es que se reduce considerablemente el número de evaluaciones de la función objetivo mejorando el desempeño frente a problemas de portafolios considerablemente grandes.

Al utilizar un AG para optimizar portafolios de inversión con restricciones lineales se incrementa notablemente el número de evaluaciones de la función objetivo. Sin embargo un AG puede alcanzar valores óptimos o muy cercanos al óptimo. En este último caso para que el AG encuentre valores muy cercanos al óptimo es necesario aumentar el número de bits asignado a cada acción y por tanto aumentar el número de población y número de generaciones. Se puede concluir que aunque en este capítulo no se tenga como objetivo comparar un AG contra PC, es posible generar los mismos resultados óptimos a través de los dos métodos.

Capítulo 4

Optimización con restricciones no lineales con AGs y PC

En este capítulo se presentan pruebas para dos restricciones no lineales (cardinalidad y costos de transacción), que pueden ser incorporadas al problema de optimización de portafolios de inversión. Se hace especial énfasis en estas restricciones ya que en programación cuadrática no se puede incorporar de forma directa restricciones no lineales, y es necesario aplicar ciertas simplificaciones al problema de optimización para poderlo resolver mediante el método antes mencionado. Por otra parte en un AG es posible integrar restricciones no lineales. Las restricciones incluidas en este capítulo son dos restricciones, cardinalidad y costos de transacción. Cardinalidad significa limitar el número de acciones que conforman el portafolio, costos de transacción son todos aquellos costos que se adquieren al realizar el intercambio de un portafolio inicial para transformarlo a otro portafolio óptimo. Cabe resaltar que en este capítulo se hace referencia a optimización de un objetivo debido a que uno de las hipótesis de esta tesis es probar que el algoritmo genético puede optimizar un portafolio de inversión con restricciones no lineales.

Los experimentos realizados en este capítulo consisten primero en obtener portafolios a través PC y AG para diferentes valores de cardinalidad y comparar los resultados. Posteriormente se varía el porcentaje para la restricción de costos de transacción para obtener portafolios a través de un AG y se exponen los resultados. En este capítulo se busca demostrar que el proceso de optimización mediante un AG puede alcanzar resultados que PC no puede encontrar, por lo menos en su formulación básica.

4.1. Restricción de cardinalidad

En ciertos mercados es necesario restringir un portafolio a un número determinado de activos por la gran complejidad que representa monitorear un conjunto muy grande de acciones. La restricción de cardinalidad se refiere a forzar un portafolio óptimo a un determinado número de acciones α , donde el portafolio óptimo es aquel que maximiza la rentabilidad y minimiza el riesgo para el número de acciones seleccionado. Hay que

hacer énfasis en la importancia de estas restricción de cardinalidad porque cuando se minimiza el riesgo el portafolio óptimo incluye una gran cantidad de activos, si suponemos el caso de que existen 100 acciones en el mercado resultaría una portafolio cercano a 100 acciones lo que significaría una gran cantidad de recursos humanos y computacionales para monitorear las acciones adquiridas.

4.1.1. Programación cuadrática

En programación cuadrática no es posible incluir la restricción de cardinalidad en forma directa o como una restricción más, debido a que este método sólo permite incluir restricciones lineales. Como se puede ver la formulación de programación cuadrática en la ecuación 4.1 sólo se integran tres posibles restricciones lineales; de desigualdad $\mathbf{Ax} \leq \mathbf{b}$, de igualdad $\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}$ y de límites del vector donde $\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$.

$$f(x) = \min_x \left(\frac{1}{2} \mathbf{xHx}' + \mathbf{x}'\mathbf{f} \right) \quad (4.1)$$

sujeto a:

$$\mathbf{Ax} \leq \mathbf{b}$$

$$\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}$$

$$\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$$

donde \mathbf{x} es el vector de pesos que se minimiza, \mathbf{H} es una matriz y \mathbf{f} es un vector . Este problema es equivalente a resolver a un sistema de ecuaciones lineales, estableciendo las condiciones iniciales para generar una solución inicial y posteriormente buscar el valor mínimo [25]. El problema es que no es posible integrar una restricción que dado un vector \mathbf{x} y a través de una función $C(\mathbf{x})$ nos indique el número de valores del vector mayores a 0, para este resultado igualarlo a un número determinado de acciones α , como se muestra en la ecuación 4.2.

$$f(x) = \min_x \frac{1}{2} \mathbf{xHx}' + \mathbf{x}'\mathbf{f} \quad (4.2)$$

sujeto a:

$$\mathbf{Ax} \leq \mathbf{b}$$

$$\mathbf{A}_{eq}\mathbf{x} = \mathbf{beq}$$

$$\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$$

$$C(\mathbf{x}) = \alpha$$

Una manera de encontrar un resultado aproximado con PC, es seleccionar los α activos con mayor beneficio y desechar los activos restantes, es decir sólo seleccionar aquellos activos que dado el nivel de aversión al riesgo sean más óptimos. El vector

resultante no forma un vector donde la suma del vector sea igual a 1, por lo tanto, el valor restante se distribuye de forma uniforme sobre todos los elementos del nuevo vector. Como se muestra en el pseudocódigo 6. Este mecanismo propuesto no asegura optimalidad solo es un método de aproximación para comparar los resultados obtenidos con el AG.

Algorithm 6 PC con restriccion de cardinalidad

Require: *conditions* inputs of PC program and α cardinality

Ensure: \vec{w} weigths output

```

1: FUNCTION PC.Cardinalidad(conditions,  $\alpha$ )
2:  $x \leftarrow PC(\textit{conditions})$  {se soluciona el problema a través de PC}
3: resultado = {se crea la variable donde se va guardando la solución}
4: for  $i = 0$  to  $\alpha$  { se restan los  $\alpha$  mejores activos } do
5:   Maximo  $\leftarrow \max(x, i)$  {se selecciona el k-ésimo maximo}
6:   resultado[Maximo]  $\leftarrow x$ [Maximo] {se guarda el valor del k-ésimo maximo}
7: end for
8: Total  $\leftarrow \frac{(1-\textit{sum}(\textit{resultado}))}{\alpha}$  {se calcula el valor restante y se divide en partes iguales}
   }
9: resultado  $\leftarrow \textit{resultado} + \textit{Total}$  {el resto se distribuye sobre todos los elementos}
10: return resultado

```

Las pruebas realizadas con restricciones lineales con PC son diversas ejecuciones de programación cuadrática con diferentes valores de cardinalidad para un problema con 50 acciones. Los datos empleados son los mismos empleados en la sección 3.2 que son datos históricos del precio de cierre de 50 acciones de la Bolsa Mexicana de Valores de enero del 2007 a abril del 2008. Como se mencionó en la sección 4.1.1 se seleccionan los α mejores acciones y se normaliza el vector de salida para que la suma sea igual al 100%, el resultado de este procedimiento se grafica en la figura 4.1 un diagrama de área de 100%.

Como se puede observar, en la figura 4.1 se muestra cómo a medida que se incrementa la cardinalidad del portafolio, se incrementa también el número de acciones que se incluyen en el portafolio óptimo como se esperaría fuese el comportamiento normal con dicha restricción. Sin embargo, cuando se incrementa a un número de cardinalidad mayor los portafolios resultantes incluyen acciones con pesos muy pequeños los cuales no son significativos en una compra, ya que estos valores prácticamente se convertirían en muy poco dinero dependiendo de la cantidad de capital invertido, esto nos indica que el algoritmo propuesto a través del AG.

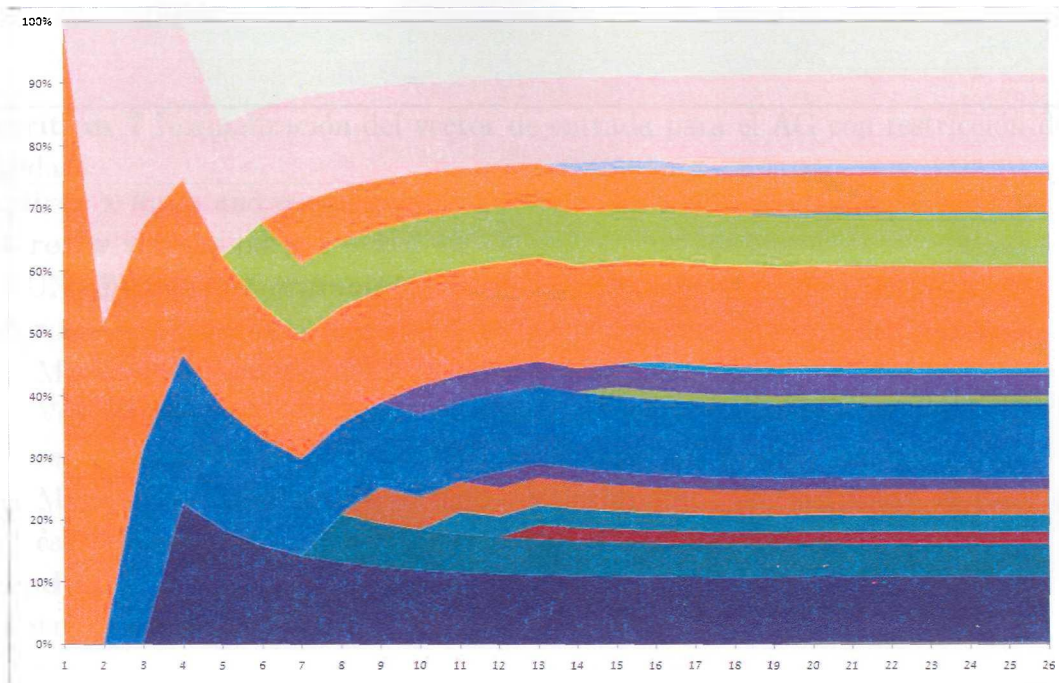


Figura 4.1: Pruebas con PC para diferentes valores de cardinalidad.

4.1.2. Algoritmos genéticos

Para implementar la restricción de cardinalidad para optimizar un portafolio de inversión en un algoritmo genético, es necesario cambiar la función objetivo. La función que se implementa con la restricción de cardinalidad cumple con el objetivo de modificar el vector de entrada y convertirlo en un vector de pesos donde $\sum \mathbf{w} = 1$ y $C(\mathbf{w}) = \alpha$. Esto quiere decir que se reinterpreta el vector de entrada \mathbf{x} para transformarlo en un vector \mathbf{w} donde su suma es igual a uno y el número de w_i 's mayores que cero, sea igual a un parámetro α (*cardinalidad*). En el pseudocódigo 7 muestra el algoritmo propuesto para normalizar el vector en la forma mencionada.

Las pruebas realizadas con el AG son diversas ejecuciones del algoritmo con diferentes valores de cardinalidad para un problema con 50 acciones. Los datos empleados son los mismos empleados en la sección 4.1.1. Como se mencionó en la sección 4.1.2 el AG selecciona aquellas α mejores acciones y el valor restante se normaliza para que la suma sea igual a uno, el resultado de este procedimiento se grafica en la figura 4.2.

En la figura 4.2 se muestran los portafolios que se encontraron a medida que se incrementa la restricción cardinalidad para un problema de 50 acciones, como se esperaría en el comportamiento normal, el portafolio encontrado incrementa también el número de acciones que se incluyen en el portafolio óptimo con dicha restricción. Para las pruebas mostradas en esta sección se utilizó un algoritmo con inyecciones de diversidad y con los parámetros seleccionados en la sección 3.6.

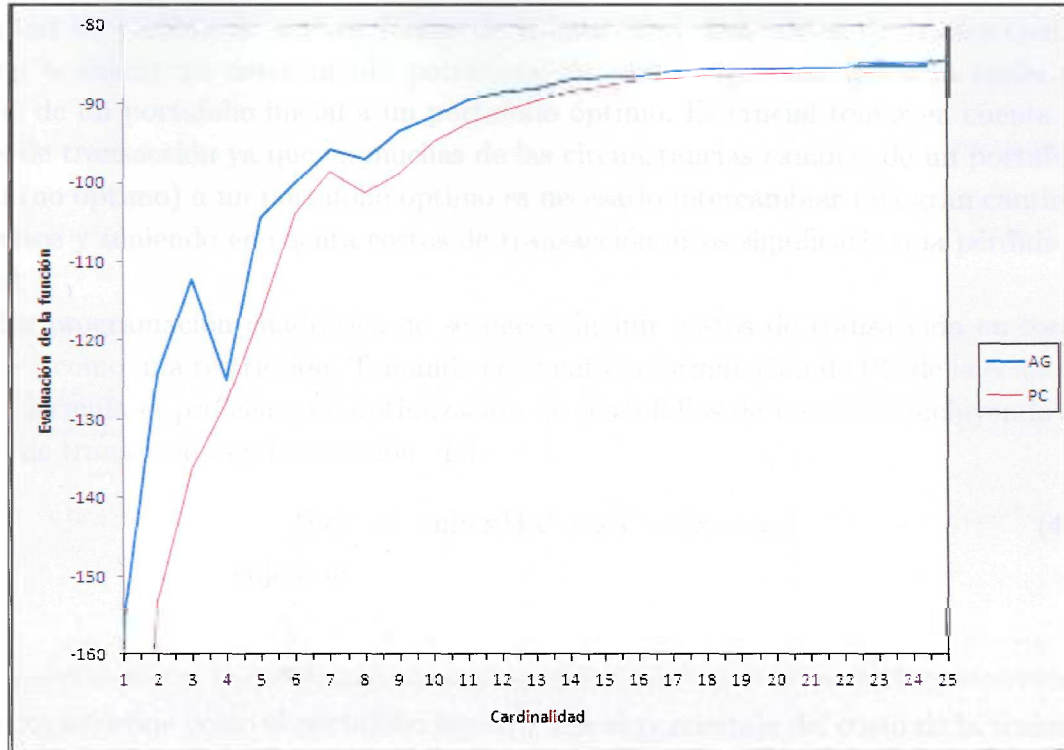


Figura 4.3: Pruebas con AG para diferentes valores de cardinalidad.

4.1.3. Comparación entre PC y AGs

Para comparar los resultados obtenidos a través de programación cuadrática y algoritmos genéticos, se grafica la misma función de evaluación para los dos métodos para cada nivel de cardinalidad. Como se puede observar en la figura 4.3 el AG obtiene mejores resultados ya que obtiene mejores evaluaciones en la función objetivo, sin embargo a medida que el nivel de cardinalidad aumenta el valor encontrado a través de PC se asemeja mucho al encontrado con PC. Esto se debe a que el portafolio óptimo sin restricciones incluye aproximadamente 25 acciones esto indica que aproximadamente el valor calculado a través de PC sin restricción de cardinalidad ya se asemeja al valor óptimo con restricción de cardinalidad.

4.2. Costos de transacción

Los costos de transacción son todos aquellos gastos generados por el proceso de compra y venta, en algunas ocasiones estos gastos provienen de impuestos, comisiones, etc. Los costos de transacción no son una restricción en sí misma, si no un complemento de la función de optimización. En ciertos mercados los costos de transacción son muy altos, por esta razón es necesario restringir un portafolio óptimo a una determinada

magnitud de cambio de activos (costo de transacción). Los costos de transacción se refieren a añadir un determinado porcentaje de costo ϵ que multiplica la razón del cambio de un portafolio inicial a un portafolio óptimo. Es crucial tomar en cuenta los costos de transacción ya que en muchas de las circunstancias cambiar de un portafolio inicial (no óptimo) a un portafolio óptimo es necesario intercambiar una gran cantidad de activos y teniendo en cuenta costos de transacción altos significaría una pérdida de capital.

En programación cuadrática no se puede incluir costos de transacción en forma directa o como una restricción. Tomando en cuenta la formulación de PC de la ecuación 4.1, se formula el problema de optimización de portafolios de inversión incluyendo los costos de transacción en la ecuación 4.3.

$$f(x) = \min_x \mathbf{xHx}' + \mathbf{x}'\mathbf{f} - \epsilon|\mathbf{x} - \mathbf{x}_0| \quad (4.3)$$

sujeto a:

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

donde \mathbf{x}_0 se define como el portafolio inicial y ϵ es el porcentaje del costo de la transacción. Este problema en su forma más simple sólo está sujeto a una restricción donde la suma del vector \mathbf{x} es igual a uno.

4.2.1. Algoritmos genéticos

Al igual que en la restricción de cardinalidad para optimizar un portafolio de inversión con costos de transacción en un algoritmo genético, es necesario cambiar la función objetivo. La función que se implementa para los costos de transacción busca modificar el vector de entrada y convertirlo en un vector de pesos donde $\sum \mathbf{w} = 1$ y en donde la evaluación del vector \mathbf{x} se le resta de la rentabilidad, la diferencia entre el portafolio inicial \mathbf{x}_0 por un costo de transacción ϵ . Esto quiere decir que se reinterpreta el vector de entrada \mathbf{x} para transformarlo en un vector \mathbf{w} , donde su suma es igual a uno y la evaluación de la rentabilidad es la diferencia entre el portafolio del vector normalizado menos la diferencia entre del vector inicial $\epsilon|\mathbf{x} - \mathbf{x}_0|$. En el algoritmo 8 se muestra la función propuesta para evaluar un individuo en la forma antes mencionada.

Las pruebas realizadas en esta sección consisten en variar el valor del costo de transacción y encontrar portafolios eficientes a través de una AG, en las figuras 4.5 4.6 y 4.4 se muestran las misma corrida del algoritmo genético, pero en cada figura se muestra una perspectiva diferente del resultado. Los parámetros del algoritmo genético utilizado son probabilidades de cruce 1 mutación en 0, 500 individuos de población y 250 generaciones para un problema con 50 acciones. Para las pruebas realizadas en esta sección el valor de portafolio óptimo inicial es calculado a través del AG con restricción de cardinalidad de 5, en otras palabras el portafolio \mathbf{x}_0 es calculado con los datos del

Algorithm 8 Función objetivo del AG con costos de transacción

Require: \mathbf{x} input and ϵ transaction cost**Ensure:** w function evaluation

- 1: FUNCTION $AG.TransactionCost(\mathbf{x}, \epsilon)$
 - 2: $w \leftarrow \frac{\mathbf{x}}{\sum \mathbf{x}}$ {se normaliza el vector de entrada }
 - 3: evaluation $\leftarrow \mathbf{wHw}' + \mathbf{w}'\mathbf{f} - \epsilon|\mathbf{w} - \mathbf{w}_0|$ {se evalúa el vector w }
 - 4: **return** evaluation
-

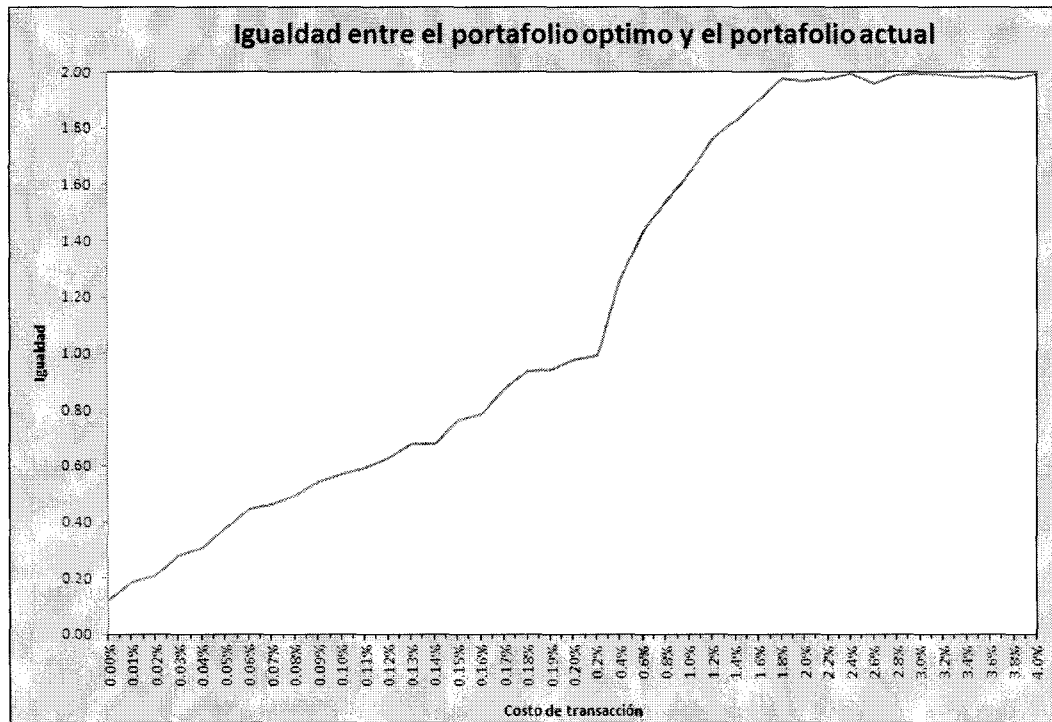


Figura 4.4: Pruebas con AG para diferentes valores de cardinalidad.

problema de 50 acciones, pero considerando una cardinalidad de 5 acciones, con un valor de aversión al riesgo intermedio.

En la figura 4.5 se muestran los portafolios eficientes encontrados por el AG con diferentes valores del costo de transacción. A través de un diagrama de área del 100%, cada color representa un acción y el tamaño del polígono de diferente color representa el porcentaje asignado del portafolio. Como se puede observar, la tendencia de variar el costo de transacción es mantener las mismas acciones pero con diferentes porcentajes. En la parte extrema derecha de la figura 4.5 se encuentra el portafolio inicial, es decir el portafolio \mathbf{x}_0 . La figura 4.5 nos muestra como a medida que el costo de transacción aumenta el portafolio óptimo se asemeja más al portafolio inicial \mathbf{x}_0 .

En la figura 4.4 se muestra el porcentaje de similitud entre los portafolios encontrados por el AG y el portafolio inicial (\mathbf{x}_0), se grafica el costo de transacción contra

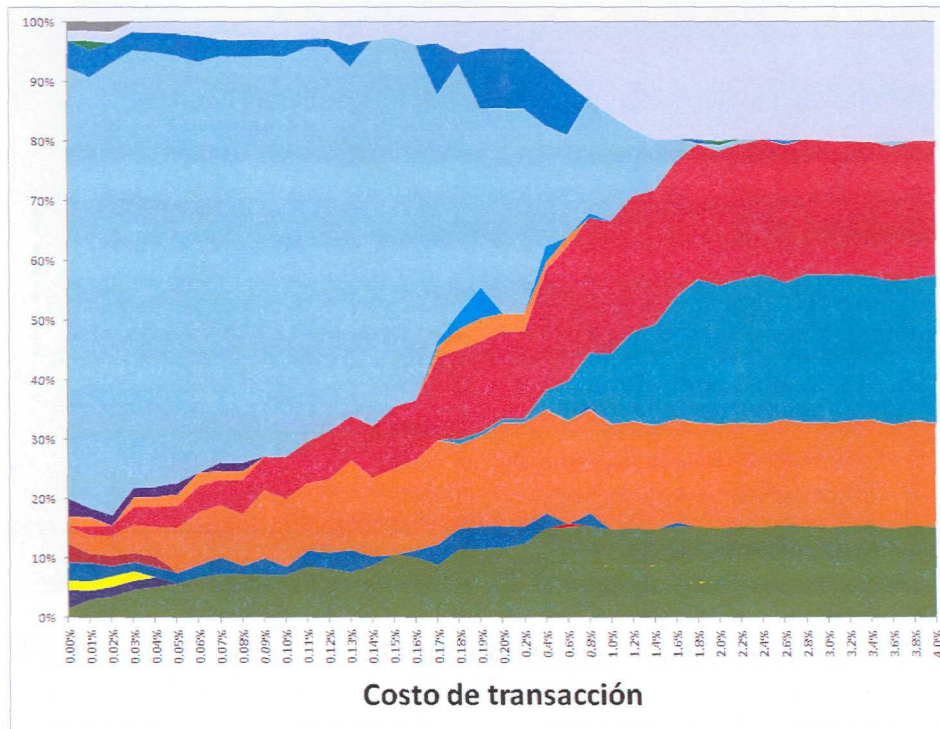


Figura 4.5: Evaluación de la función objetivo para diferentes valores de costos de transacción.

el porcentaje de igualdad del portafolio. Se consideró que la igualdad del portafolio es la suma de las diferencias absolutas entre los dos portafolios. Como se puede observar en la figura 4.4, la tendencia de variar el costo de transacción es que a medida que el costo de transacción aumenta, el portafolio óptimo se asemeja más al portafolio inicial. En la parte extrema izquierda de la figura 4.4 cuando el costo por transacción es cero, el portafolio óptimo no se asemeja al portafolio inicial (\mathbf{x}_0) y el portafolio tiene evaluaciones que se acercan al óptimo, cuando el costo de transacción es muy alto (extrema derecha de la figura) el portafolio óptimo se asemeja al portafolio inicial (\mathbf{x}_0) como se puede observar en la figura 4.6.

4.3. Conclusiones del capítulo

En este capítulo se busca demostrar que un AG puede optimizar un portafolio de inversión con restricciones no lineales a través de dos restricciones de cardinalidad y costos de transacción. Para optimizar un portafolio de inversión con restricciones de cardinalidad con un AG, se probaron diferentes niveles de esta restricción de cardinalidad y se compararon con una aproximación mediante programación cuadrática; en un AG cuando los valores de cardinalidad son pequeños se encuentra mejores resultados



Figura 4.6: Pruebas con AG para diferentes valores de cardinalidad.

que PC, pero a medida que se incrementa el valor de cardinalidad la diferencia entre ambos métodos se reduce, esto se debe a que el valor de cardinalidad se acerca al número de acciones del portafolio óptimo sin restricciones, por lo tanto, el resultado de PC sin aproximación ya es cercano al portafolio óptimo con restricción de cardinalidad.

Por otra parte, se realizaron una serie de pruebas con diferentes niveles de costos de transacción, donde el portafolio origen \mathbf{x}_0 es obtenido con PC con la un portafolio óptimo con cardinalidad de 5 acciones. Los resultados obtenidos para esta restricción es que cuando el costo de transacción es bajo el portafolio obtenido con el AG se asemeja al portafolio óptimo sin restricción de costo de transacción, y a medida que el costo de transacción aumenta el portafolio óptimo obtenido con el AG se asemeja al portafolio inicial \mathbf{x}_0 .

Capítulo 5

Optimización con restricciones no lineales y diferentes niveles de riesgo

En este capítulo se realizan una serie de pruebas con el fin de obtener portafolios eficientes con y sin restricciones no lineales (cardinalidad y costos de transacción). Los portafolios eficientes como se mencionó con anterioridad son todos aquellos portafolios que están en el frente de Pareto. En las pruebas se emplea un algoritmo genético con inyecciones de diversidad donde el objetivo es maximizar la rentabilidad y minimizar el riesgo, utilizando para ello un factor de ponderación entre la rentabilidad y el riesgo (aversión al riesgo); para cada portafolio eficiente la evaluación se estima con diferentes niveles de aversión al riesgo para obtener diferentes portafolios. En este capítulo se busca demostrar que un AG puede obtener diversos portafolios eficientes e incluso obtener la frontera de eficiencia, ponderando el valor entre el riesgo y la rentabilidad y, por otra parte, se realizan pruebas para producir el frente de Pareto con diferentes formas para variar el valor de aversión al riesgo.

Es importante mencionar que en los capítulos anteriores se busca determinar si un algoritmo genético puede obtener portafolios eficientes con restricciones sin embargo no se busca obtener un conjunto de portafolios eficientes. En este capítulo se realizaron una serie de pruebas que consisten en obtener portafolios eficientes con restricciones de cardinalidad y restricción de costos de transacción con diferentes niveles de aversión al riesgo, para esto se fija un nivel de cardinalidad y costo de transacción. En los resultados se muestra el frente de Pareto con los resultados encontrados por programación cuadrática sin restricciones y el algoritmo genético con las restricciones antes mencionadas. Esto es con el fin de presentar el resultado de incluir restricciones no lineales, que en programación cuadrática no es posible incluir en una gráfica donde se compare el comportamiento del frente de Pareto.

5.1. Portafolios eficientes con AGs variando aversión al riesgo

En esta sección se muestra como un algoritmo genético puede obtener diferentes portafolios eficientes sin restricciones variando la aversión al riesgo. Como se mencionó en el capítulo 3, la función utilizada para optimizar un portafolio de inversión de un objetivo es 3.4. La ecuación completa que normaliza y calcula el valor de la función como se muestra en la ecuación 5.2.

$$\begin{aligned} \mathbf{w} &= \frac{\mathbf{x}}{\sum_{i=1}^n x_i} \\ f(x) &= \min_w \left(\mathbf{f}\mathbf{w} - k \left(\frac{1}{2} \mathbf{w}\mathbf{H}\mathbf{w}' \right) \right) \end{aligned} \quad (5.1)$$

En esta sección se busca demostrar que al variar la aversión al riesgo dada la ecuación 5.2 es posible determinar diversos portafolios eficientes. Para las prueba realizadas se toma el problema mencionado en el capítulo 3 sección 3.2, que es un problema de 10 acciones. El resultado de la primera prueba se muestra en la figura 5.1, en esta se prueba se varía el riesgo en un incremento constante de la aversión al riesgo (k). En esta figura se grafica la rentabilidad contra el riesgo de las evaluaciones de los portafolios. En la gráfica se presenta una curva que es el resultado de las evaluaciones de portafolios eficientes obtenidos a través de programación cuadrática sin restricciones y los puntos sobresalientes sobre la línea son las evaluaciones de los portafolios obtenidos por el algoritmo genético. Se puede observar que el algoritmo genético puede igualar los resultados que PC pero a medida que se incrementa el riesgo en forma constante se disminuye el espacio entre las evaluaciones de portafolios.

Para obtener portafolios eficientes donde las evaluaciones de riesgo y rentabilidad se encuentran distribuidas a lo largo del frente de Pareto, se propone variar la aversión al riesgo. En las gráficas 5.2 y 5.3 se varía el riesgo en forma exponencial con exponentes de 1.35 y 2 respectivamente, como se puede observar en estas gráficas el algoritmo genético con este tipo de progresión exponencial obtiene resultados a lo largo de la frontera de eficiencia, pero cuando la aversión al riesgo es muy baja, los resultados se encuentran muy dispersos uno del otro, lo cual no produce como tal el frente de Pareto, es decir los primeros portafolios obtenidos tienen entre sí una distancia mayor, por lo cual no están distribuidos de forma equitativa. Por último se evalúa el algoritmo genético con una progresión que crece en forma exponencial logarítmica como se muestra en la ecuación 5.2 donde la función β es la aversión al riesgo, i es el número de iteración, que es el numero de veces que se ejecuta el AG y c es una constantes que es

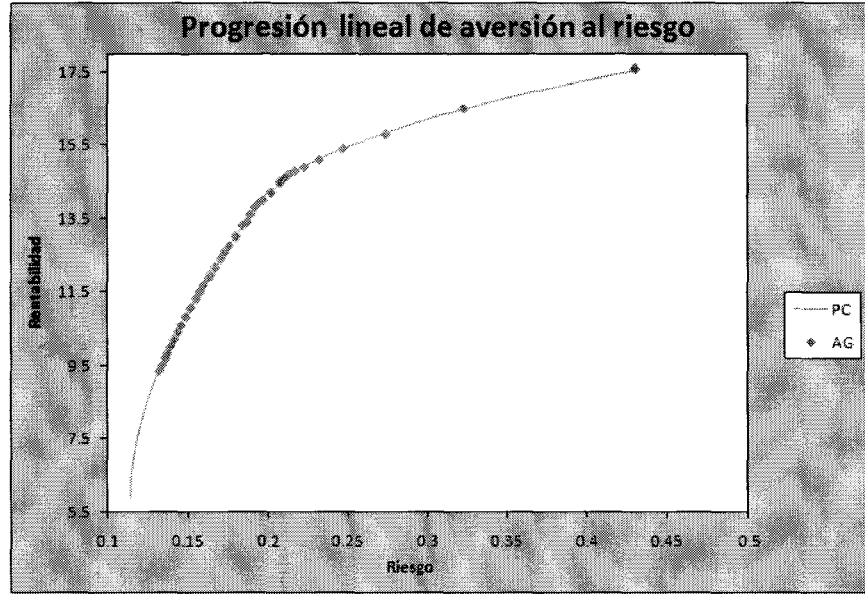


Figura 5.1: Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión lineal.

un parámetro ajustable para aumentar o reducir la amplitud del frente de Pareto, el resultado de aplicar este tipo de progresión se grafica en la figura 5.4. En esta figura (5.4) el algoritmo genético puede encontrar resultados a lo largo de toda la frontera de eficiencia y además puede encontrar resultados de forma en que la distancia entre los portafolios es más uniforme que una progresión exponencial.

$$\beta(i) = i^{\log_{10}(i)^c} \quad (5.2)$$

En la figura 5.5 se presenta el resultado de variar la aversión al riesgo de forma en la que se menciona en la ecuación 5.2, sin embargo, en esta ocasión se aplica para el problema de 50 acciones, como se puede observar en la gráfica las evaluaciones de los portafolios encontrados por el algoritmo genético son muy similares a los obtenidos por programación cuadrática. Por tanto, es posible concluir que un algoritmo genético de un objetivo como se propone en la ecuación 5.2 y variando la aversión al riesgo (k) puede encontrar portafolios eficientes que están distribuidos sobre el frente de Pareto.

5.2. Optimización con restricción de cardinalidad variando aversión al riesgo

En esta sección se realizan una serie de experimentos para encontrar una serie de portafolios eficientes con restricción de cardinalidad, que permitan mostrar la diferencia

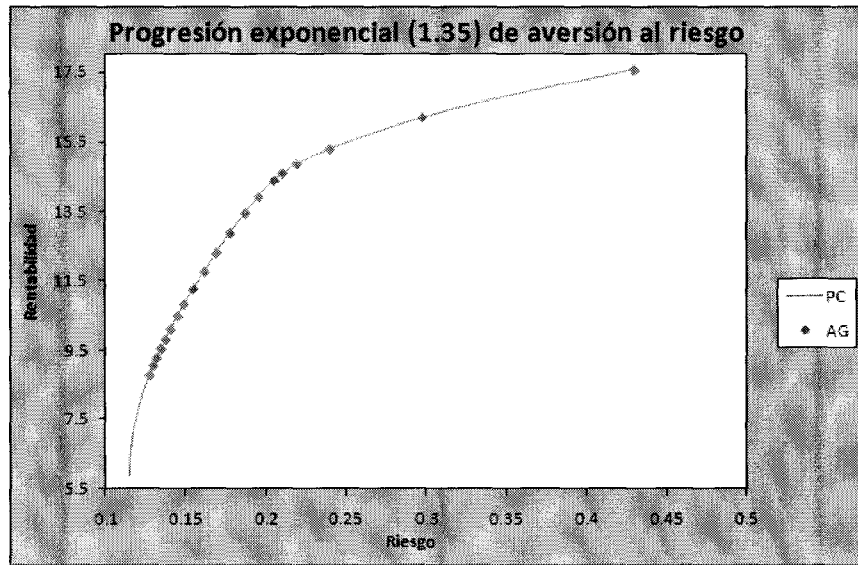


Figura 5.2: Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial.

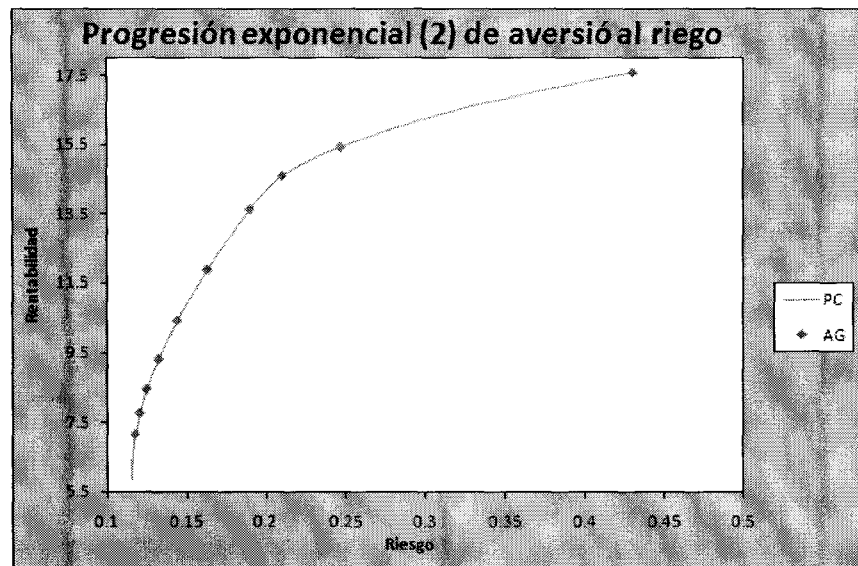


Figura 5.3: Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial.

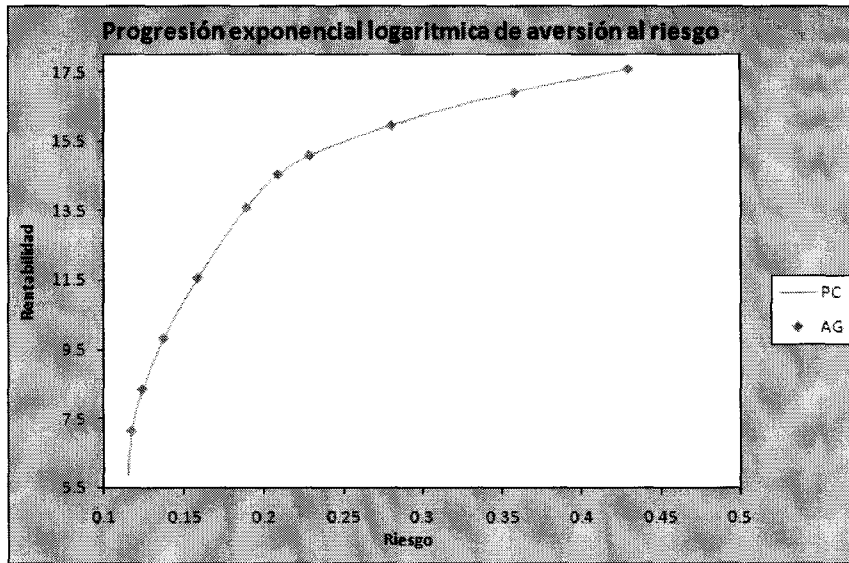


Figura 5.4: Pruebas con AG para un problema de 10 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

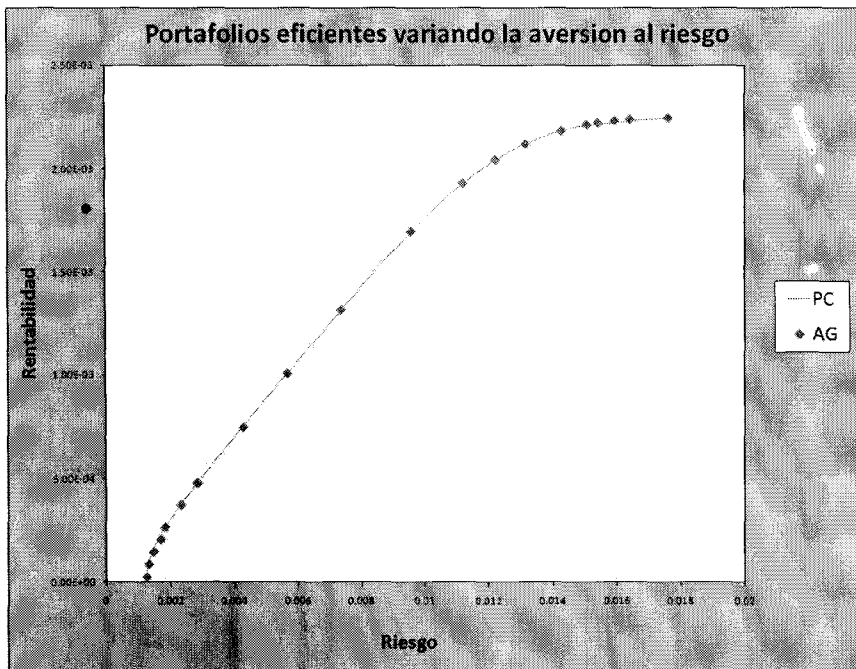


Figura 5.5: Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

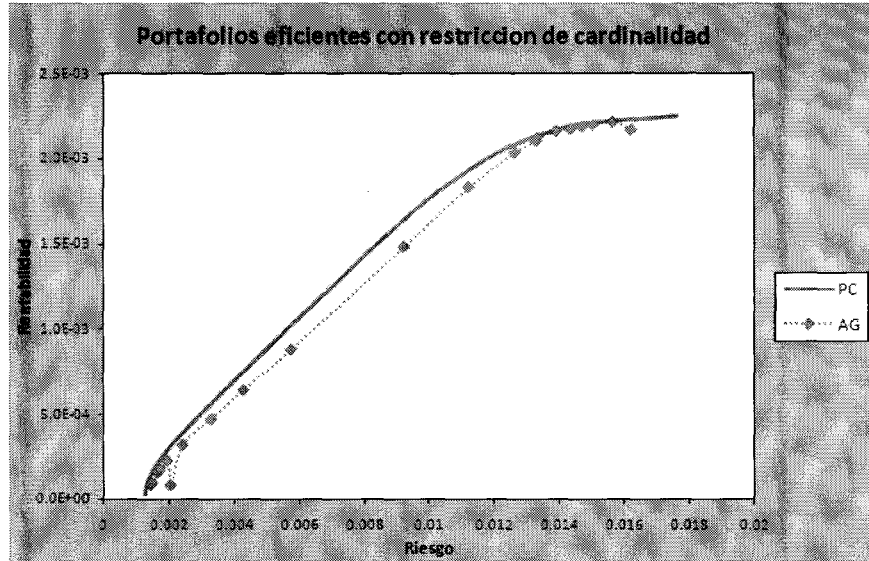


Figura 5.6: Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

entre el frente de Pareto sin restricciones y con la restricción de cardinalidad. Para la pruebas realizadas se utilizó la codificación de cardinalidad propuesta en la sección 4.1.1, en un problema de 50 acciones, el valor de cardinalidad utilizado es de 5 acciones. El resultado de esta prueba se muestra en la figura 5.6, en ella se varia la aversión al riesgo en un incremento exponencial logarítmico como se mencionó en la sección 5.1. En esta figura se grafica la rentabilidad contra el riesgo de las evaluaciones de los portafolios. En la gráfica se presenta una curva que es el resultado de diversas evaluaciones de portafolios eficientes obtenidos a través de programación cuadrática sin restricciones y la línea punteada son las evaluaciones de los portafolios obtenidos por el algoritmo genético con la restricción de cardinalidad.

Se puede observar en la figura 5.6 el algoritmo genético encuentra portafolios que en su evaluación se encuentran por debajo de los obtenidos por programación cuadrática debido a que PC no está considerando la restricción de cardinalidad. En la gráfica se puede observar que los portafolios eficientes con restricción de cardinalidad tienen a encontrar la frontera de eficiencia. En la figura 5.7 se muestra el comportamiento de las acciones incluidas en el portafolio a medida que se incrementa la aversión al riesgo. Un algoritmo genético con la suficiente cantidad de evaluaciones de la función objetivo puede encontrar portafolios eficientes con restricción de cardinalidad e incluso determinar el frente de Pareto.

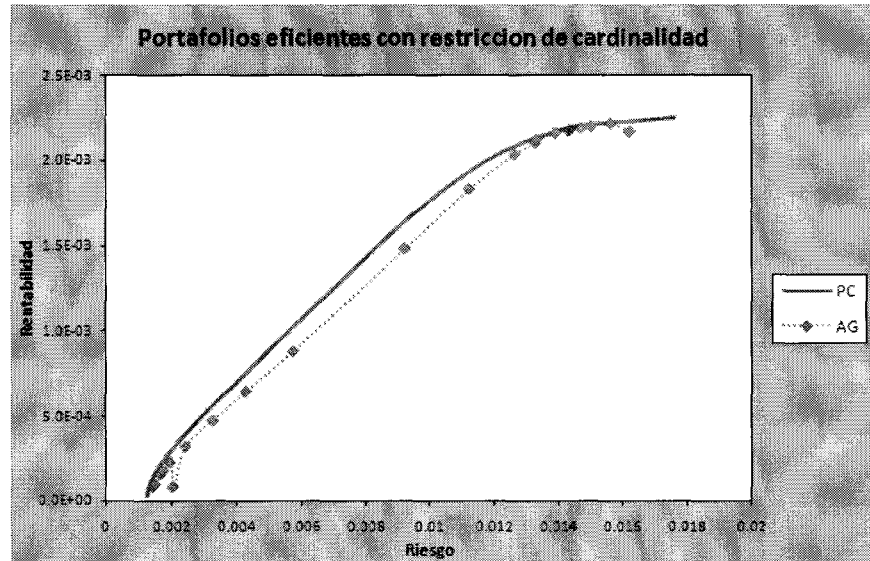


Figura 5.7: Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

5.3. Optimización con costos de transacción variando aversión al riesgo

En esta sección se realizan experimentos para encontrar portafolios eficientes con costo de transacción, que permitan mostrar la diferencia entre el frente de Pareto sin y con costos de transacción. Para la pruebas realizadas se utilizó la codificación de costos de transacción propuesta en la sección 4.2.1 donde a la función de evaluación 5.2 se le agrega el costo de transacción. Esta prueba se aplica para un problema de 50 acciones con un costo de transacción fijo del 0.1 % y como portafolio inicial se tiene un portafolio con cardinalidad de 5 acciones (este portafolio es el portafolio óptimo para el problema de 50 acciones con cardinalidad de 5). La prueba consiste en variar la aversión al riesgo en un incremento exponencial logarítmico como se mencionó en la sección 5.1. En la figura 5.8 se presenta una curva que es el resultado de las evaluaciones de portafolios eficientes obtenidos a través de programación cuadrática sin ninguna restricción y sin costos de transacción y en la línea punteada son las evaluaciones de los portafolios obtenidos por el algoritmo genético con costos de transacción.

Se puede observar en la figura 5.8 que el algoritmo genético encuentra portafolios que en cuanto a su evaluación se encuentran por debajo de los obtenidos por programación cuadrática debido a que PC no está considerando los costos de transacción; estos resultados con menor evaluación son el resultado de incluir un costo de transacción a la función de evaluación de un objetivo. En la figura 5.9 se muestran el comportamiento de las acciones incluidas en el portafolio a medida que se incrementa la aversión al

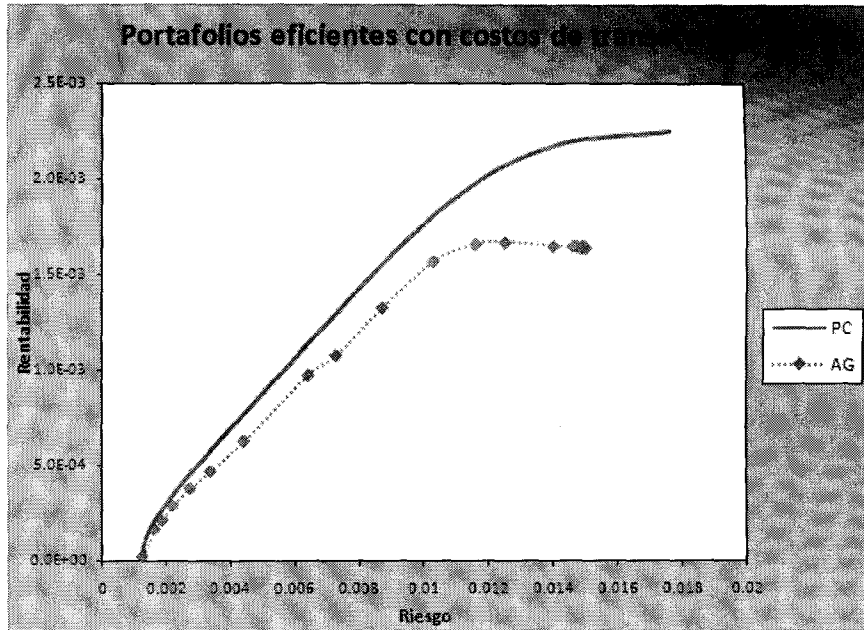


Figura 5.8: Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

riesgo. Se puede concluir que un algoritmo genético con la suficiente cantidad de evaluaciones de la función objetivo puede encontrar portafolios eficientes con restricción de cardinalidad e incluso determinar el frente de Pareto.

5.4. Conclusiones del capítulo

Un algoritmo genético con la evaluación objetivo 5.2 y con los valores apropiados para la aversión al riesgo, puede encontrar portafolios eficientes que se encuentran distribuidos en la frontera de eficiencia, lo cual genera el mismo resultado generado a través de programación cuadrática, sin embargo, a diferencia que programación cuadrática, el algoritmo genético para encontrar los resultados emplea una gran cantidad de recursos computacionales. La ventaja de un algoritmo genético es que es posible incorporar restricciones como costos de transacción y costos por el número acciones en el portafolio (cardinalidad).

En las pruebas realizadas podemos observar que aplicar un algoritmo genético con la restricción de cardinalidad y variando la aversión al riesgo se obtienen portafolios con evaluaciones por debajo del frente de Pareto, ya que se limita el espacio de búsqueda. Cuando se aplican costos de transacción en la función objetivo que optimiza el algoritmo genético al variar la aversión al riesgo los portafolios obtenidos son subóptimos ya que se encuentran por debajo del frente obtenido por programación cuadrática sin

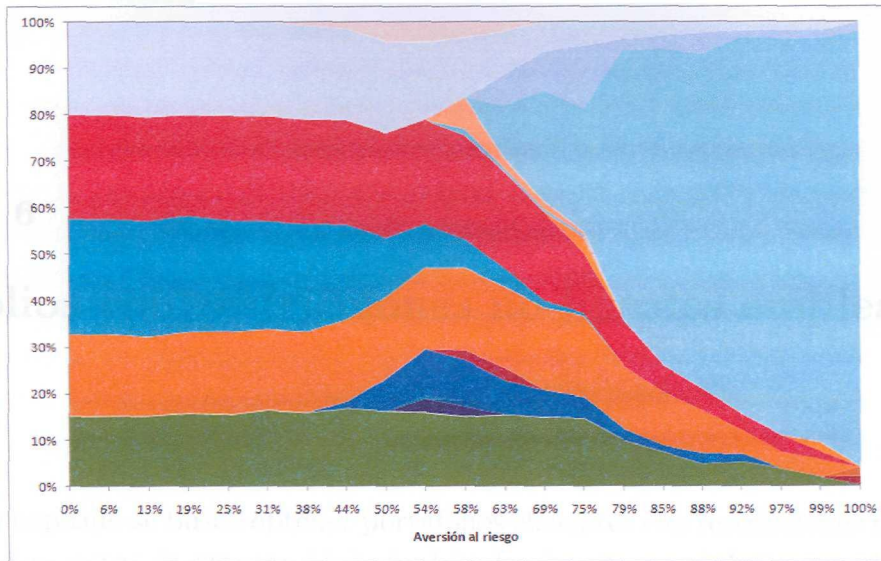


Figura 5.9: Pruebas con AG para un problema de 50 acciones con diferentes valores de aversión al riesgo con progresión exponencial logarítmica.

restricciones, esto se debe a que cuando se toma en cuenta los costos por comprar y vender se agrega un valor por cada movimiento que se realiza.

Capítulo 6

Portafolios con restricciones no lineales empleando EMOs

En este capítulo se busca obtener portafolios eficientes con restricciones no lineales (cardinalidad y costos de transacción) empleando algoritmos evolutivos multiobjetivo como fue propuesto en el planteamiento del problema en la sección 5. En las pruebas se emplea tres algoritmos evolutivos multiobjetivo (EMO por sus siglas en inglés *Evolutionary Multiobjective Optimization*) NSGA 2, SPEA 2 y vgGAMO, este último es propuesto por Valenzuela y Uresti [26], los objetivos en dichos algoritmos es maximizar la rentabilidad y minimizar el riesgo. En este capítulo en primera instancia se quiere verificar que un algoritmo evolutivo multiobjetivo puede obtener diversos portafolios eficientes sin restricciones después de una cantidad considerable de evaluaciones. En la segunda parte, se añaden restricciones al problema para mostrar el comportamiento de los algoritmos EMO aumentando la complejidad del problema.

En los capítulos anteriores se determina como un algoritmo genético de un objetivo puede obtener portafolios eficientes con restricciones no lineales. En este capítulo se busca obtener portafolios eficientes con restricciones de cardinalidad y restricción de costos de transacción para un algoritmo evolutivo multiobjetivo. En las pruebas se muestra el frente de Pareto con los resultados encontrados por programación cuadrática sin restricciones y el algoritmo genético multiobjetivo con las restricciones antes mencionadas. Esto es con el fin de mostrar en una gráfica el comportamiento de la frontera de eficiencia con restricciones no lineales empleando el enfoque de algoritmos multiobjetivo. A diferencia del trabajo previo en [12], en esta sección se incluyen los costos de transacción con un planteamiento alternativo que le resta a la función de rentabilidad los costos de transacción (costos de transacción proporcionales). Por otra parte para la restricción de cardinalidad se reinterpretan los individuos para que estos se conviertan en un vector que satisfaga esta restricción.

6.1. Optimización sin restricciones con EMOs

En esta sección se muestra como un algoritmo genético multiobjetivo, busca diferentes portafolios eficientes sin restricciones. En el capítulo 3 se utiliza una función de un objetivo para optimizar un portafolio de inversión 3.4 pero en esta ocasión el enfoque multiobjetivo permite separar en dos ecuaciones los valores de rentabilidad y riesgo. Las ecuaciones 6.2 y 6.3 son las funciones que representan riesgo y rentabilidad respectivamente. A diferencia del enfoque de un objetivo en éste se elimina el factor de aversión al riesgo.

$$\mathbf{w} = \frac{\mathbf{x}}{\sum_{i=1}^n x_i} \quad (6.1)$$

$$f_1(x) = \min_w (\mathbf{wHw}')^{\frac{1}{2}} \quad (6.2)$$

$$f_2(x) = \min_w -\mathbf{w}'\mathbf{f} \quad (6.3)$$

Para las pruebas realizadas se tomó el problema mencionado en la sección 3.2 que es el problema de 50 acciones. Los resultados de aplicar un EMO se muestran en una gráfica una curva que es el resultado de diversas evaluaciones de portafolios eficientes, obtenidos a través de programación cuadrática sin restricciones y los puntos en la gráfica representan las evaluaciones de los portafolios obtenidos por el algoritmo EMO.

6.1.1. NSGA 2

Como se mencionó anteriormente, uno de los algoritmos multiobjetivo empleados en esta tesis es el algoritmo propuesto por Deb (NSGA 2) [9]. El algoritmo utiliza las funciones 6.2 y 6.3 como objetivos. En la figura 6.1 se muestran diferentes pruebas realizadas con este algoritmo. Los parámetros seleccionados del NSGA 2 son probabilidad de cruce 0.9 y probabilidad de mutación 1/50 y selección binaria de torneo, con una población de 20 individuos con 1,000, 10,000 y 100,000 ciclos, como se mencionó este algoritmo es no generacional por lo tanto se obtienen 1,000, 10,000 y 100,000 evaluaciones respectivamente. Se puede observar que el algoritmo NSGA 2 al cabo de 100,000 evaluaciones de la función objetivo (figura 6.1) no logra obtener portafolios eficientes, sin embargo se puede notar que conforme el número de evaluaciones aumenta la tendencia del algoritmo es que la población se mueve hacia el frente de Pareto.

6.1.2. SPEA 2

El algoritmo es SPEA 2 es propuesto por Eckart Zitzler and Lothar Thiele [28], y es utilizado para con las funciones 6.2 y 6.3 como objetivos. Los parámetros seleccionados

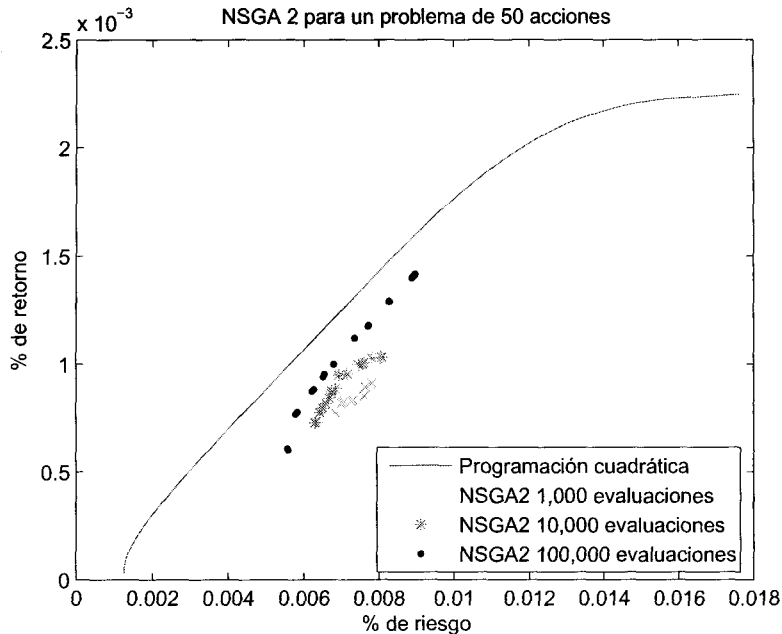


Figura 6.1: NSGA 2 para un problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.

del SPEA 2 son probabilidad de cruce 0.9 y probabilidad de mutación 1/50 y selección binaria de torneo con una población de 20 individuos con 50, 500 y 5,000 generaciones, dado que este algoritmo es un algoritmo genético generacional se logran 1,000, 10,000 y 100,000 evaluaciones de la función objetivo (mismas que en las pruebas del algoritmo NSGA 2). Se puede observar en la figura 6.2 que el algoritmo SPEA 2 al cabo de 100,000 evaluaciones de la función objetivo se acerca a la frontera de eficiencia. Al aumentar el número de evaluaciones se muestra la tendencia del algoritmo donde la población se mueve hacia el frente de Pareto.

6.1.3. vgGAMO

Este algoritmo es propuesto por Manuel Valenzuela y Eduardo Uresti [26] y es una adaptación del algoritmo vgGA para problemas multiobjetivo. Este algoritmo es de estado estable, no generacional y consiste en generar una población inicial aleatoria para posteriormente sólo modificar un individuo de la población con aquellos padres que estén menos dominados y que tengan un conteo de nicho pequeño, como se explica en el capítulo 2 en la sección 2.5. En las pruebas se aplicaron las funciones 6.2 y 6.3 como objetivos del algoritmo para un problema de 50 acciones. Los parámetros seleccionados del algoritmo son probabilidad de cruce 0.9, probabilidad de mutación 0.01, constante de nicho (C_m) en 1, constante de dominancia (C_d) en 2 y selección binaria de torneo

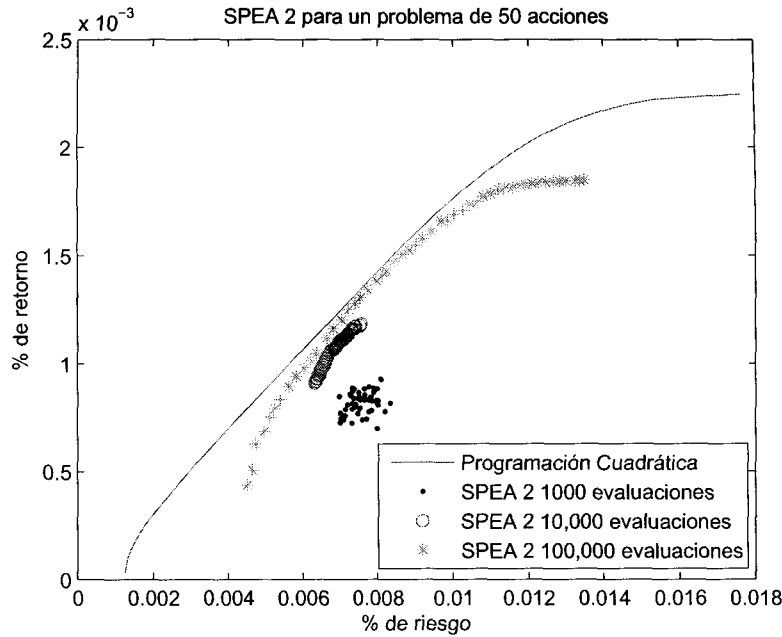


Figura 6.2: SPEA 2 con problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.

con una población de 20 individuos para 1,000, 10,000 y 100,000 ciclos, dado que este algoritmo genético es no generacional se logran 1,000, 10,000 y 100,000 evaluaciones de la función objetivo. Se puede observar en la figura 6.3 en el algoritmo al cabo de 100,000 evaluaciones de la función objetivo no se encuentran soluciones en el frente de Pareto pero sin embargo al aumentar el número de evaluaciones se muestra que la tendencia del algoritmo es que la población se mueva en dirección al frente de Pareto.

6.1.4. Comparación de algoritmos evolutivos multiobjetivo

Existen una gran cantidad de métricas de evaluación de algoritmos multiobjetivo sin embargo en este trabajo para comparar los resultados obtenidos con los algoritmos se propone una métrica que evalúa la población final de individuos no-dominados. En esta métrica se mide que tan cercanas son las soluciones al frente de Pareto, esta medición consiste en que para cada individuo del algoritmo evolutivo se calcula la distancia entre el individuo y el punto más cercano de un conjunto de puntos que se encuentran en el frente de Pareto calculados a través de programación cuadrática. En la ecuación 6.4 se resume el procedimiento explicado para calcular esta métrica (M_1), donde n es el número de puntos encontrados por el algoritmo evolutivo (individuos no dominados), m es el número de puntos encontrados en el frente de Pareto (programación cuadrática), r_i y σ_i es la rentabilidad y riesgo del punto i respectivamente del algoritmo EMO, r_{fe_j} y

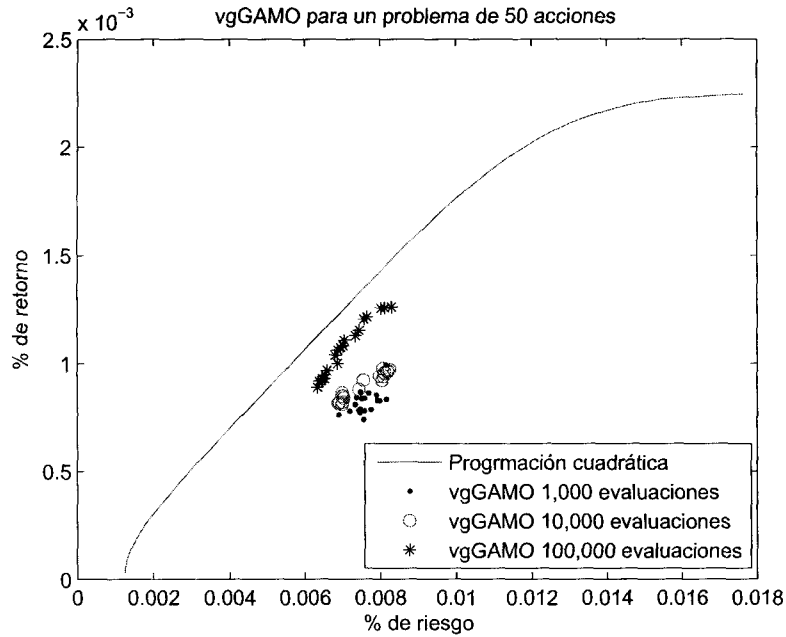


Figura 6.3: vgGAMO con problema de 50 acciones, con 1,000, 10,000 y 100,000 evaluaciones de la función objetivo.

Número de evaluaciones	SPEA 2	NSGA 2	vgGAMO
10,000	0.000079159	0.00031661	0.000268337
50,000	0.000143941	0.000349939	0.000296724
100,000	0.000170446	0.000264544	0.000246731

Tabla 6.1: Evaluaciones del NSGA 2, SPEA 2 y vgGAMO para diferente número de evaluaciones de la función objetivo empleando métrica 6.4

σ_{fe_j} es la rentabilidad y riesgo del punto j respectivamente de la frontera de eficiencia. Nótese que en esta métrica el mejor algoritmo es aquel que posee un menor valor porque está más cercano al frente de Pareto.

$$M_1 = \frac{1}{n} \sum_{i=1}^n \min \left\{ |r_i - r_{fe_j}| + |\sigma_i - \sigma_{fe_j}| \forall j \in (1, \dots, m) \right\} \quad (6.4)$$

Los resultados de aplicar la métrica 6.4 se muestran en la tabla 6.1. El algoritmo SPEA 2 encuentra resultados más cercanos al frente de Pareto, ya que en promedio la distancia que hay de sus individuos no dominados es menor que los obtenidos de los individuos del algoritmo NSGA 2 y vgGAMO. El resultado de los algoritmos multiobjetivo se muestra en la figura 6.4 donde los individuos del algoritmo SPEA 2 son más cercanos al frente de Pareto que los dos algoritmos para el mismo número de evaluaciones de la función objetivo.

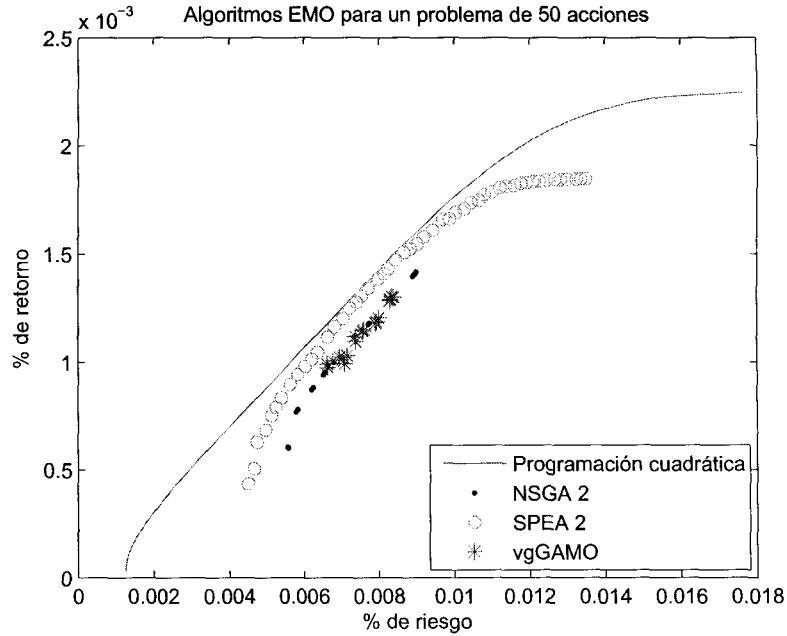


Figura 6.4: Algoritmos EMO para un problema de 50 acciones, con 100,000 evaluaciones de la función objetivo.

Otra forma de evaluar los resultados generados por estos algoritmos es contar el número de individuos dominados por las otras poblaciones. En la ecuación 6.5 se muestra el procedimiento para calcular dicha evaluación (M_2), donde A es el conjunto de individuos no dominados de la población del algoritmo evaluado y \bar{A} es el conjunto de individuos de las poblaciones de los algoritmos no evaluados, $\rho(i)$ es una función que devuelve 1 si el individuo i es dominado por algún individuo del conjunto \bar{A} , y 0 cuando no es dominado. En esta métrica el mejor algoritmo es aquel que posee un menor valor porque esto significa que tiene menos individuos dominados.

$$M_2 = \sum_{\forall i \in A} \rho(i) \quad (6.5)$$

donde :

$$\rho(i) = \begin{cases} 1 & \text{si } i \prec j (\forall j \in \bar{A}) \\ 0 & \text{si no} \end{cases}$$

Los resultados de aplicar la métrica 6.5 a los tres resultados de los algoritmos se muestran en la tabla, como se puede observar los individuos del SPEA 2 dominan a los individuos de NSGA 2 y vgGAMO. Por lo tanto el algoritmo SPEA 2 para este problema de optimización encuentra mejores individuos que los otros algoritmos como se muestra en la figura 6.4.

Número de evaluaciones	SPEA 2	NSGA 2	vgGAMO
10,000	0	20	20
50,000	0	20	20
100,000	0	20	20

Tabla 6.2: Evaluaciones del NSGA 2, SPEA 2 y vgGAMO para diferentes número de evaluaciones de la función objetivo empleando métrica 6.5

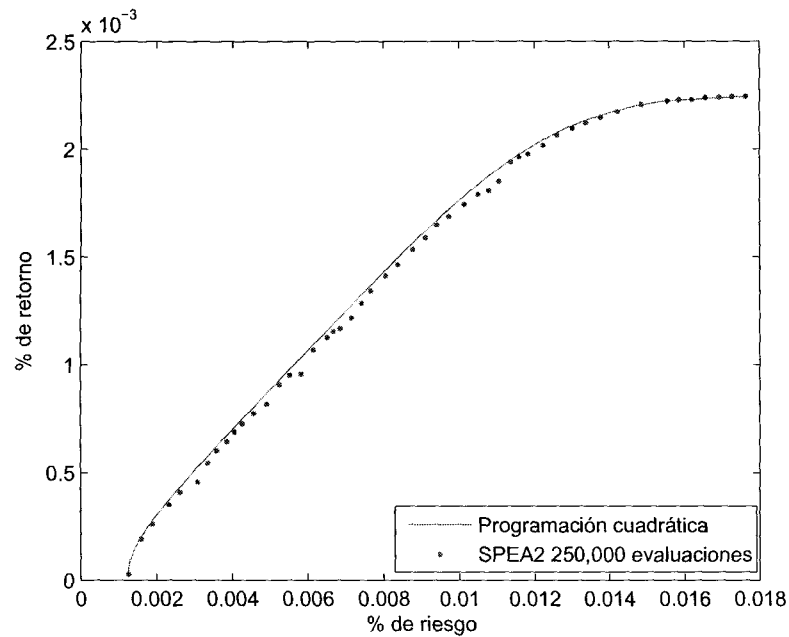


Figura 6.5: Algoritmos SPEA 2 para problema de 50 acciones, con 250,000 evaluaciones de la función objetivo.

Para verificar la calidad de la solución generada por el algoritmo SPEA 2, se realiza una prueba con una gran cantidad de evaluaciones. En la figura 6.5 se muestra una ejecución con 250,000 evaluaciones para el problema de 50 acciones. Los parámetros seleccionados del SPEA 2 son probabilidad de cruce 0.9 y probabilidad de mutación 1/50 y selección binaria de torneo con una población de 20 individuos y 12,500 generaciones. En esta prueba se aprecia como algunos de los individuos de la población se encuentran en el frente de Pareto, pero no la totalidad de la población y los individuos se encuentran distribuidos a lo largo de todo el frente de Pareto. Una desventaja de este algoritmo es que para obtener óptimos de Pareto, se realizan un gran número de evaluaciones que derivan en un alto consumo de recursos computacionales y tiempo (aproximadamente 12 horas).

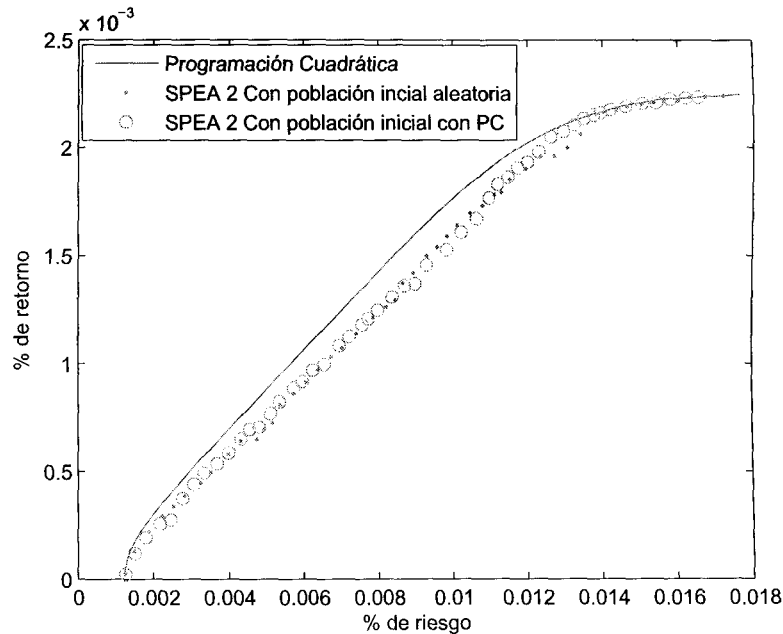


Figura 6.6: SPEA 2 con población inicial aleatoria y con PC para un problema de 50 acciones con cardinalidad 5

6.2. Optimización con cardinalidad como restricción

En investigaciones previas [16] e incluso en la sección anterior el algoritmo evolutivo multiobjetivo SPEA 2 al parecer tiene un mejor desempeño que el NSGA II y el vgGAMO en el problema de optimización de portafolios de inversión. En esta sección se realiza una serie de pruebas con el algoritmo SPEA 2 pero agregando al problema la restricción de cardinalidad. Las funciones objetivo consideradas son 6.2 y 6.3 pero antes de evaluarlas se aplica una normalización en el individuo (como se menciona en la sección 4.1) para que este cumpla con la restricción de cardinalidad. En esta sección se utilizan dos mecanismos para el algoritmo SPEA 2, el primer mecanismo consiste en generar una población inicial totalmente aleatoria y otro mecanismo con una población inicial con los valores óptimos que están en el frente de Pareto sin restricción de cardinalidad, calculados a través de programación cuadrática. Los resultados son mostrados en la figura 6.6 con un valor fijo de cardinalidad de 5. Las pruebas se aplican en un problema de 50 acciones, con probabilidad de cruce 0.9 y probabilidad de mutación 1/50 y selección binaria de torneo con una población de 20 individuos.

En la figura 6.6 se puede observar el comportamiento del algoritmo SPEA 2 con dos variantes aplicado al problema de optimización con restricción de cardinalidad de 5 acciones. En la gráfica la línea representa el frente de Pareto sin restricciones calculado

Algoritmo SPEA 2	Evaluación métrica 6.4	Evaluación métrica 6.5	Número de evaluaciones
Población inicial aleatoria	0.00293953	17	250,000
Población inicial de PC	0.00283526	10	20,000

Tabla 6.3: SPEA 2 con diferente población inicial para problema con cardinalidad

con PC, la primera variante del algoritmo SPEA 2 con una población inicial aleatoria es marcada como un punto y la segunda variante es una población inicial construida con el frente de Pareto sin restricciones marcada en la gráfica como un círculo. Como se pueda observar en la gráfica los resultados son muy similares sin embargo al AG con población inicial aleatoria requiere de 250,000 evaluaciones de la función objetivo y el AG con población inicial dirigida realiza en 20,000 evaluaciones objetivo, encontrando prácticamente el mismo resultado. En la tabla 6.3 se muestra la evaluación propuesta en la ecuaciones 6.4 y 6.5 para los dos algoritmos. Como se puede observar en la tabla el resultado de la métricas, es que el algoritmo SPEA 2 con población inicial con PC es ligeramente mejor ya que en promedio sus individuos son más cercanos al frente de Pareto y además tiene menos individuos dominados. Por lo tanto al insertar una población inicial con individuos que se encuentran en la frontera de eficiencia sin restricciones reduce notablemente el número de evaluaciones que se requieren para encontrar portafolios con cardinalidad.

Por último en la figura 6.7 se muestra diferentes evaluaciones del algoritmo SPEA 2 insertando una población con el frente de Pareto, con diferentes valores de cardinalidad, como se puede observar en esta gráfica al aumentar el valor de cardinalidad la frontera resultante se asemeja a la frontera de eficiencia obtenida por programación cuadrática sin restricciones pero medida que el valor de cardinalidad disminuye la frontera resultante se aleja además de incluir una menor cantidad de puntos no dominados.

6.3. Optimización con cardinalidad como objetivo

En esta sección se agrega un tercer objetivo, además de la mencionada en para mostrar el comportamiento del algoritmo SPEA 2 cuando se aumenta el espacio de búsqueda. Se ejecuta el algoritmo SPEA 2 con una población inicial generada con los óptimos de Pareto con programación cuadrática, para buscar portafolios que además de buscar minimizar el riesgo y maximizar la rentabilidad, se busca minimizar el número de acciones que se encuentran en el portafolio (cardinalidad). En la figura 6.8 se muestra una gráfica de tres dimensiones donde el eje x representa el riesgo, el eje y la cardinalidad, y el eje z la rentabilidad, los puntos son las evaluaciones de los individuos no dominados de la población encontrados por el algoritmo SPEA 2 y la línea es la curva del frente de Pareto sin restricción cardinalidad. Como se puede observar el algoritmo

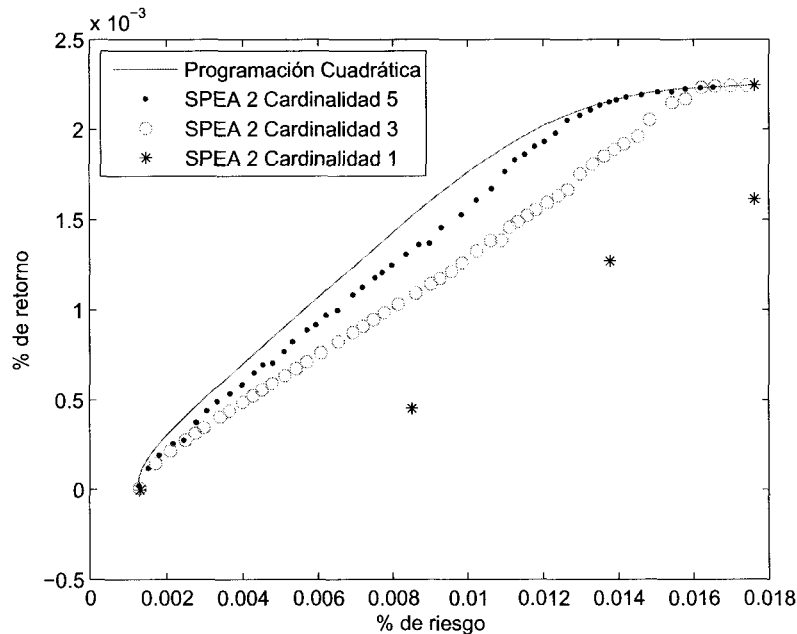


Figura 6.7: SPEA 2 con población inicial con PC para un problema de 50 acciones con cardinalidad 5,3 y 1

SPEA 2 encuentra diversos puntos no dominados sin embargo los puntos no encontrados en la mayor parte no cubren el espacio de búsqueda. Otro aspecto importante es que el algoritmo multiobjetivo permite visualizar puntos por ejemplo que para el método de PC serían “invisibles”, dado que en el modelo simplificado no es posible minimizar el número de activos en el portafolio.

6.4. Optimización con costos de transacción

Al igual que en la sección anterior en esta sección se utiliza el algoritmo evolutivo multiobjetivo SPEA 2. Se realiza una serie de pruebas con el algoritmo pero agregando al problema costos de transacción partiendo de un portafolio inicial con cardinalidad de 5 acciones. Las funciones objetivo consideradas son 6.2 y 6.3 pero agregando a la función 6.2 los costos de transacción (como se menciona en la sección 4.2 en el capítulo 4). Al igual que en la sección anterior también se utilizan dos mecanismos para el algoritmo SPEA 2, generando una población inicial totalmente aleatoria y otro mecanismo con una población inicial con los óptimos en el frente de Pareto sin restricciones, calculados a través de programación cuadrática. Los resultados son mostrados en la figura 6.9 con un portafolio inicial de cardinalidad de 5 y un costo de transacción fijo del 0.1%. La pruebas se aplican en un problema de 50 acciones, con probabilidad de cruce 0.9 y probabilidad de mutación 1/50 y selección binaria de torneo con una población de 20

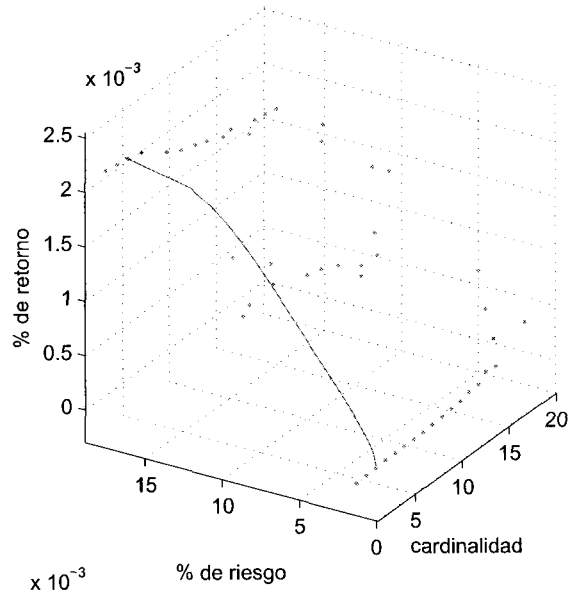


Figura 6.8: SPEA-II para problema de tres objetivos

individuos.

En la figura 6.9 se muestran las evaluaciones de los individuos resultantes del algoritmo SPEA 2 con las variantes antes mencionadas para problema de optimización con costos de transacción. En la gráfica la línea representa el frente de Pareto sin restricciones calculado con PC, la primera variante del algoritmo SPEA 2 con una población inicial aleatoria es marcada como un punto y la segunda variante es una población inicial construida con el frente de Pareto sin restricciones marcada en la gráfica como un círculo. En la gráfica en el algoritmo SPEA 2 con población inicial aleatoria se emplearon 250,000 evaluaciones y el SPEA 2 con población inicial con PC se emplea 20,000 evaluaciones, este último encontrando un mejor resultado. En la tabla 6.4 se muestra la evaluación propuesta en la ecuaciones 6.4 y 6.5 para los dos algoritmos. El algoritmo SPEA 2 con PC es mejor ya que en promedio sus individuos son más cercanos al frente de Pareto y además tiene menos individuos dominados, además el número de evaluaciones requerido es mucho menor. Por lo tanto al insertar una población inicial con individuos que se encuentran en la frontera de eficiencia sin restricciones se reduce notablemente el número de evaluaciones que se requieren para encontrar portafolios con costos de transacción.

Por último en la figura 6.10 se muestra diferentes ejecuciones del algoritmo SPEA 2 insertando una población con PC, y con diferentes valores de costo de transacción, como se puede observar en esta gráfica al aumentar el costo de transacción se reduce el número de portafolios encontrados por el algoritmo, es decir a medida que aumenta el costo de

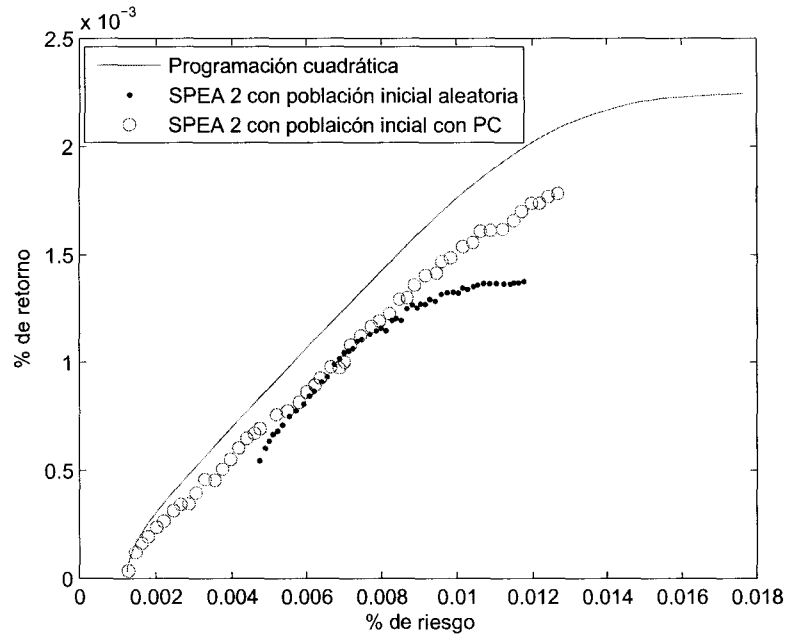


Figura 6.9: SPEA 2 con población inicial aleatoria y con PC para un problema de 50 acciones con costos de transacción

Algoritmo SPEA 2	Evaluación métrica 6.4	Evaluación métrica 6.5	Número de evaluaciones
Población inicial aleatoria	0.000373194	41	250,000
Población inicial de PC	0.000224351	2	20,000

Tabla 6.4: SPEA 2 con diferente población inicial para el problema con costos de transacción.

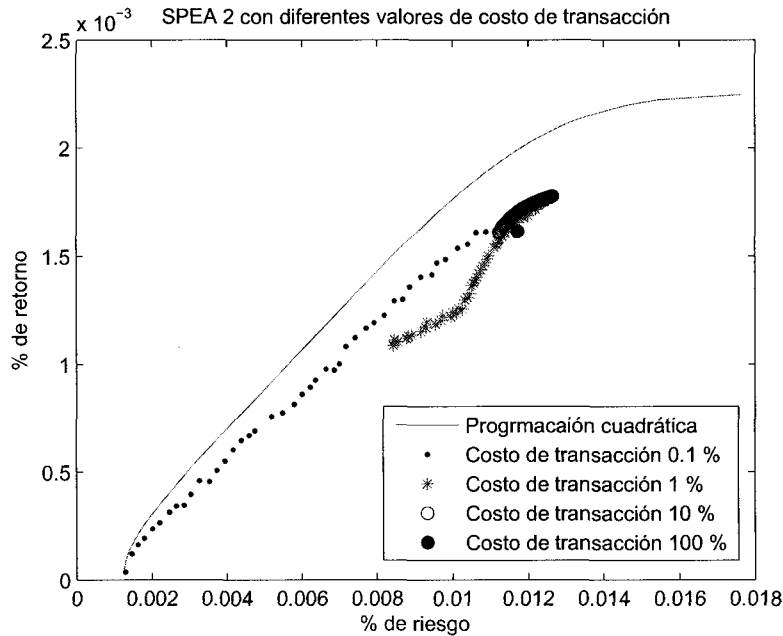


Figura 6.10: SPEA 2 con población inicial con PC para un problema de 50 acciones con costos de transacción de 100 %, 10 %, 1 % y 0.1 %

transacción las evaluaciones de los portafolios encontrados por el algoritmos son más similares entre sí, cuando el costo de transacción es máximo (100 %) se encuentra un portafolio que es el portafolio inicial, esto significa que cuando el costo de transacción es muy alto el mejor el portafolio es que se asemeja al portafolio inicial y cuando es costo de transacción es muy bajo los portafolios se asemejarán a la frontera de eficiencia sin restricciones.

6.5. Optimización con costos de transacción como objetivo

En esta sección se representa el costo de transacción como un tercer objetivo, además de los objetivos mencionados 6.2 y 6.3 empleando el algoritmo SPEA 2. Se ejecuta el algoritmo con una población inicial generada con los óptimos de Pareto con programación cuadrática, para buscar portafolios que a demás de buscar minimizar el riesgo y maximizar la rentabilidad, minimicé la diferencia entre el portafolio inicial (w_0) y el portafolio evaluado como se muestra en la ecuación 6.7. En la figura 6.11 se muestra una gráfica de tres dimensiones donde el eje x representa el riesgo, el eje y el costo de transacción, y el eje z la rentabilidad, los puntos es la población de evaluaciones no dominadas encontrados por el algoritmo SPEA 2 y la línea es la curva del frente

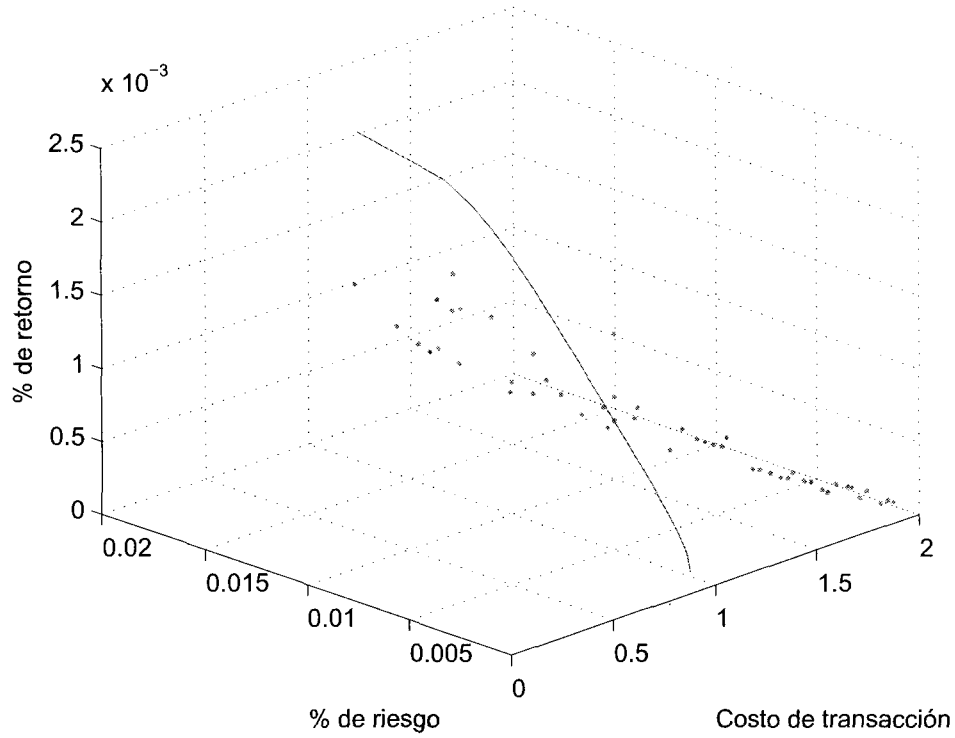


Figura 6.11: SPEA 2 para problema de tres objetivos

de Pareto sin restricción. Como se puede ver en la gráfica, los portafolios encontrados no cubren todo el espacio tridimensional dado que este espacio está limitado por las restricciones y además que el algoritmo está sujeto a los tres objetivos mencionados.

$$\begin{aligned}
 \mathbf{w} &= \frac{\mathbf{x}}{\sum_{i=1}^n x_i} \\
 f_1(x) &= (\mathbf{w}\mathbf{H}\mathbf{w}')^{\frac{1}{2}} \\
 f_2(x) &= \mathbf{w}'\mathbf{f} \\
 f_3(x) &= \sum_{i=1}^n |w_i - w_{0_i}|
 \end{aligned} \tag{6.6}$$

6.6. Conclusiones del capítulo

El algoritmo SPEA 2 tiene un mejor desempeño que el algoritmo NGSa 2 y vgGAMO, como se puede ver en la figura 6.4. En comparación con el SPEA 2, el

NSGA 2 y vgGAMO tienen un proceso de exploración más lento, ya que son algoritmos no generacionales, mientras que el SPEA 2 es un algoritmo generacional que explora en un menor número de evaluaciones el espacio de búsqueda, aunque requiere mayor capacidad de memoria porque además de almacenar la información propia de la población del algoritmo, se almacena un conjunto de soluciones no dominadas.

Dadas las características de este problema de dos objetivos se tiene un gran espacio de búsqueda, pero los puntos óptimos dentro de este espacio son puntos que relativamente predefinidos, por ejemplo cuando se maximiza la rentabilidad sin importar el riesgo, el punto óptimo resulta de un portafolio con una acción de mayor rendimiento, en contra parte cuando se busca minimizar el riesgo sin importar la rentabilidad entonces se busca distribuir el portafolio en diversas acciones conforme se minimice la correlación entre las acciones. Dada esta característica del problema el espacio de búsqueda se puede explorar con mayor facilidad y al incluir una población inicial con el frente de Pareto sin restricciones potencialmente puede encontrar el óptimo, esto acelera la exploración del proceso de búsqueda del algoritmo.

Al incluir cardinalidad en el problema de optimización el espacio de búsqueda es reducido y por tanto el algoritmo genético multiobjetivo encuentra resultados por debajo de la frontera de eficiencia sin restricciones. El mismo resultado es encontrado por el algoritmo genético de un objetivo para diferentes valores de aversión al riesgo en la sección 5.2, sin embargo en esta sección el algoritmo EMO en una sola ejecución encuentra la frontera de eficiencia. Además si en el algoritmo incluimos como población inicial con PC, el algoritmo puede encontrar la frontera eficiente en un menor número de evaluaciones que un algoritmo con una población inicial aleatoria. Cuando la restricción de cardinalidad disminuye el frente de Pareto encontrado por el algoritmo deja de asemejarse al frente de Pareto obtenido sin restricción.

Cuando al problema de optimización se le agrega un costo de transacción el espacio de búsqueda es reducido por esta razón el algoritmo multiobjetivo encuentra resultados por debajo de la frontera de eficiencia sin restricciones. Conforme se aumenta el costo de transacción el espacio cubierto por el frente de Pareto también es reducido, hasta llegar al costo máximo cuando los costos de transacción son el 100 %, en este caso el frente de Pareto es un punto que es la evaluación del portafolio inicial. Al igual que en la restricción de cardinalidad, si en el algoritmo EMO incluimos como población inicial de individuos creados con PC sin restricciones, este algoritmo puede encontrar la frontera eficiente en un menor número de evaluaciones que un algoritmo multiobjetivo con una población inicial aleatoria.

Capítulo 7

Conclusiones

En esta tesis se utilizaron algoritmos genéticos para resolver el problema de optimización de portafolios. Este problema se trató desde diferentes enfoques, primero se resolvió en forma de un objetivo con un algoritmo genético para un solo portafolio, y después se aplicó algoritmos multiobjetivo con el propósito de seleccionar múltiples resultados que se estén en el frente de Pareto. Además, este trabajo se enfocó en el modelo de Markowitz donde se busca solucionar el problema multiobjetivo de maximizar la rentabilidad y minimizar el riesgo del portafolio. Este modelo tiene una solución mediante programación cuadrática, a pesar de que es muy eficiente, una de sus limitantes es que sólo es posible incorporar restricciones lineales, por lo que existen restricciones que no se pueden incorporar a este método.

Al emplear un algoritmo genético para resolver el problema se pueden buscar soluciones que satisfagan restricciones no lineales, también reduce la dificultad para incorporar estrategias que podrían ser utilizadas en la selección de un portafolio (por ejemplo VaR o EMV). Es importante mencionar que los algoritmos genéticos juegan un papel muy importante en el ámbito de optimización porque ofrecen un método estocástico pero al mismo tiempo robusto. Los algoritmos genéticos han demostrado que son un excelente mecanismo de optimización y gracias a los grandes avances que existen en los procesos del algoritmo, este mecanismo se ha convertido en uno de los principales métodos de optimización en búsqueda ciega.

7.1. Conclusiones del AG de un objetivo

Se demostró que mediante un AG de un objetivo es posible optimizar portafolios de inversión al igual que el método de PC, al menos en la evaluación del portafolio, además que el AG puede optimizar el problema agregando restricciones que también puede incorporar PC. El número de evaluaciones que requiere el AG crece exponencialmente conforme se aumenta el tamaño de problema.

El AG requiere de un gran costo computacional para resolver problemas grandes (por ejemplo problemas mayores de 50 acciones) ya que la dificultad del problema

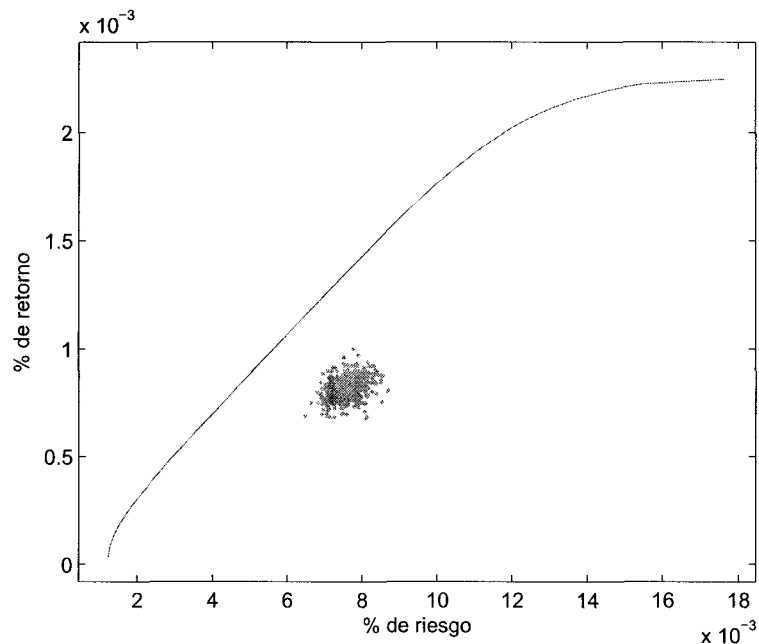


Figura 7.1: Evaluaciones de individuos generados aleatoriamente

crece a un ritmo exponencial. Por lo tanto, se concluye que para aumentar el grado de exploración del AG es necesario incluir nuevos individuos no aleatorios, por esta razón se optó por utilizar un método que generase individuos donde se tenga una mayor probabilidad de que estos contenga varios ceros en el cromosoma, como se muestran en las gráficas 7.1 y 7.2. En estas gráficas se muestra una población de individuos donde se grafica el riesgo en función de la rentabilidad, la línea representa el frente de Pareto encontrado mediante PC y cada punto representa la evaluación de un individuo de la población. En la gráfica 7.1 se muestran individuos generados en forma aleatoria y en la gráfica 7.2 se muestran individuos generados con una mayor probabilidad de contener cero en el cromosoma, como se puede ver la diferencia entre las dos poblaciones es que los individuos generados aleatoriamente están contenidos en un pequeño espacio de evaluación, es decir las evaluaciones de los individuos son similares entre sí, en cambio a medida cuando los individuos tienen en su mayoría ceros el espacio de evaluaciones cubierto es mayor y por tanto son menos similares entre sí. A medida que más ceros son incluidos en el cromosoma mas distribuidas estarán las evaluaciones de las soluciones creadas.

Para optimizar portafolios de inversión con restricciones lineales empleando un AG con inyecciones de diversidad se incrementa el número de evaluaciones de la función objetivo con respecto al problema sin restricciones, sin embargo el AG puede alcanzar valores óptimos o muy cercanos al óptimo. Para que el AG encuentre valores más cercanos al óptimo es necesario aumentar el número de bits asignado a cada acción y

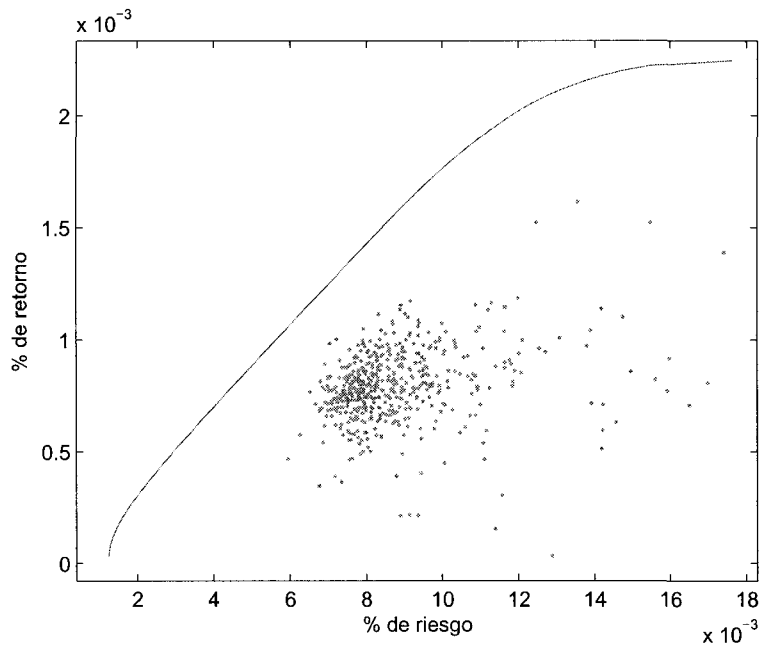


Figura 7.2: Evaluaciones de individuos generados en forma dirigida

por tanto es necesario aumentar el tamaño de la población y el número de generaciones.

Para optimizar un portafolio de inversión con restricciones de cardinalidad con un AG, se utilizó un método de normalización del individuo del tal forma que este cumpla con la restricción de cardinalidad. Se utilizaron diferentes niveles de cardinalidad y se comparó el AG con inyecciones de diversidad contra una aproximación mediante programación cuadrática. Un AG encuentra mejores resultados que PC pero cuando el valor de cardinalidad se acerca al número de acciones del portafolio óptimo sin restricciones el resultado de PC sin aproximación es cercano al portafolio óptimo por tanto se asemeja al encontrado por el AG.

Un algoritmo genético con la evaluación objetivo propuesta en este trabajo y con valores apropiados para la aversión al riesgo, puede encontrar portafolios eficientes que se encuentran distribuidos en la frontera de eficiencia lo cual genera el mismo resultado generado a través de programación cuadrática, a diferencia de programación cuadrática el algoritmo genético para encontrar cada punto del frente de Pareto emplea una gran cantidad de recursos computacionales y evaluaciones de la función objetivo.

7.2. Conclusiones AG multiobjetivo

Dado que este problema se plantea como multiobjetivo se buscó obtener portafolios eficientes con restricciones no lineales (cardinalidad y costos de transacción) empleando algoritmos EMO. En las pruebas se emplearon tres algoritmos evolutivos multiobjetivo

NSGA 2, SPEA 2 y vgGAMO, este último propuesto por Valenzuela y Uresti [26]. En investigaciones previas como [16], el algoritmo SPEA 2 tiene un mejor desempeño que el algoritmo NSGA 2. En comparación con el NSGA 2 y vgGAMO, el SPEA 2 tienen un proceso de exploración más rápido, ya que éste es un algoritmo generacional que explora en un menor número de evaluaciones el espacio de búsqueda, aunque requiere mayor capacidad de memoria porque además de almacenar la información propia de la población del algoritmo, se almacena un conjunto de soluciones no dominadas.

Dadas las características de este problema de dos objetivos se tiene un gran espacio de búsqueda, pero los puntos óptimos dentro de este espacio son puntos relativamente predefinidos, por ejemplo cuando se maximiza la rentabilidad sin importar el riesgo, el punto óptimo resulta de un portafolio con la acción de mayor rendimiento, en contra parte cuando se busca minimizar el riesgo sin importar la rentabilidad entonces se busca distribuir el portafolio en diversas acciones conforme se minimicé la correlación entre las acciones. Dada esta característica del problema el espacio de búsqueda se puede explorar con mayor facilidad y al incluir una población inicial con el frente de Pareto sin restricciones (este puede ser generado con PC) potencialmente puede encontrar el óptimo, esto acelera la exploración del proceso de búsqueda del algoritmo.

Al problema se añaden restricciones de cardinalidad y restricción de costos de transacción para mostrar el comportamiento de un algoritmo evolutivo multiobjetivo, en este caso se utilizó el algoritmo SPEA 2. En las pruebas se mostró una gráfica con el comportamiento de la frontera de eficiencia con restricciones no lineales empleando el enfoque de algoritmos multiobjetivo.

Cuando al problema de optimización se le agrega un costo de transacción el espacio de búsqueda es reducido, por esta razón el algoritmo multiobjetivo encuentra resultados por debajo de la frontera de eficiencia sin restricciones. Conforme se aumenta el costo de transacción el espacio cubierto por el frente de Pareto también es reducido, hasta llegar al costo máximo cuando los costos de transacción son el 100 %, en este caso el frente de Pareto es un punto que es la evaluación del portafolio inicial. Al igual que en la restricción de cardinalidad, si en el algoritmo EMO incluimos como población inicial a individuos creados con PC sin restricciones, este algoritmo puede encontrar la frontera eficiente en un menor número de evaluaciones que un algoritmo multiobjetivo con una población inicial aleatoria.

La hipótesis principal de esta tesis es que utilizando algoritmos genéticos y técnicas de computación evolutiva multiobjetivo es posible encontrar la frontera eficiente donde estos resultados permitan seleccionar portafolios de inversión, siguiendo restricciones que se plantean para optimización de portafolios. Esta hipótesis es válida en el ejercicio realizado en este trabajo, donde se utilizó datos históricos de 50 acciones de la Bolsa Mexicana de Valores y el índice "*Dow Jones Industrial*". En esta tesis se limitó el problema a un número determinado de acciones, sin embargo, con los parámetros adecuados los algoritmos (AG y EMO) pueden encontrar valores óptimos.

7.3. Trabajo Futuro

En esta tesis se incrementa el tamaño del problema para identificar el costo en evaluaciones que es requerido para obtener resultados óptimos a través de un algoritmo genético. Sin embargo, en los resultados no se determinó el grado de confianza alcanzado por un AG cuando el tamaño de problema crece, en otras palabras, es deseable conocer hasta cuántas acciones el AG puede determinar portafolios óptimos en un tiempo razonable.

La representación utilizada en este problema tiene ciertas limitantes dado que utiliza una matriz de n por n que es la correlación entre acciones y un vector de tamaño n que son las rentabilidades individuales de cada acción, donde n es el número de acciones. Cuando se aumenta el número de acciones en el algoritmo genético se incrementa el tamaño del cromosoma utilizado para representar una solución, ya que cada segmento del cromosoma representa una acción y por lo tanto el espacio de búsqueda se vuelve más grande; por lo que el AG requiere de un mayor tiempo y evaluaciones de la función objetivo para alcanzar valores óptimos. Un trabajo futuro es modificar la representación del problema de forma que el cromosoma no represente el peso de cada acción n .

Una de las ventajas de un AG sobre PC es la capacidad del AG para incluir funciones, restricciones y representaciones del problema de optimización de portafolios que PC no puede incluir. Un trabajo futuro es realizar una investigación empleando un AG donde se utilice una forma alterna para calcular la evaluación del portafolio por el ejemplo VaR y EMV.

En las pruebas realizadas con un AG de un objetivo para encontrar diferentes portafolios eficientes en el frente de Pareto se utiliza diferentes valores para el factor de aversión al riesgo. La forma propuesta en esta tesis es incrementar el valor de aversión al riesgo en forma exponencial logarítmico. Una de las limitantes de esta forma es que es necesario buscar ajustar el valor de la constante para cada problema.

Un trabajo futuro es para evitar el factor de aversión al riesgo, se puede normalizar el vector \mathbf{w} , incluyendo una restricción al problema de manera que el algoritmo genético genere sólo vectores de pesos que satisfagan un valor dado de rentabilidad.

Al introducir ceros en el cromosoma de cada individuo mediante inyecciones de diversidad, se tiene como desventaja que no es posible aplicarlo a optimización multiobjetivo y es deseable definir alguna manera de introducir ceros al cromosoma para cada individuo, sería bueno utilizar algún otro mecanismo, como por ejemplo introducir ceros a través del operador de mutación.

Dadas las características observadas de este problema, y en algunas pruebas no mencionadas en este trabajo, y en ciertas investigaciones como [19] se utilizan buscadores que tengan funciones de vecindad, por ejemplo un algoritmo como recocido simulado, obteniendo buenos resultados. Sería conveniente conocer el impacto en cuanto a número de evaluaciones y tiempo que tendría un algoritmo multiobjetivo de recocido

simulado por ejemplo AMOSA [3], para este problema de optimización de portafolios.

Dadas las condiciones del problema de optimización es posible emplear un algoritmo memético que pueda reducir el número de evaluaciones y por tanto reducir el tiempo requerido por el algoritmo para encontrar una solución óptima. Por otra parte, programación cuadrática como se ha mencionado es un mecanismo de optimización muy eficiente que podría incorporarse como buscador local, es decir, utilizar un algoritmo memético que en lugar de incorporar un buscador local, se incluya un método de programación cuadrática para obtener diversos individuos que por lo menos satisfagan las restricciones que puede incluir programación cuadrática.

Una forma alternativa para representar cardinalidad es incluir un costo por la cardinalidad del portafolio que se calcula por el número de acciones que se tenga por un costo fijo por acción (similar a los costos de transacción proporcionales). Otra forma de plantear esta restricción, es determinar un nivel de cambio que pudieran tener los portafolios con cardinalidad, es decir definir la magnitud de cambio que el inversionista está dispuesto a aceptar en función del número de acciones remplazadas de un portafolio inicial a un portafolio final.

En las pruebas realizadas para optimización multiobjetivo con restricciones no lineales se desea poder comparar los resultados contra algún otro mecanismo de optimización, una solución es incorporar un modelo de programación cuadrática mixta con restricciones no lineales.

Bibliografía

- [1] Eduardo Martíncz Abascal. *Invertir en Bolsa*. McGraw-Hill, 1999.
- [2] Ruben Armañanzas y Jose A Lozano. A multiobjective approach to the portfolio optimization problem. Reporte técnico, University of the Basque Country - P.O. Box 649 20018 Donostia-San Sebastián, Spain, 1999.
- [3] Sanghamitra Bandyopadhyay, Sriparna Saha, y Kalyanmoy Deb. A simulated annealing-bases multiobjective optimizatón algorithm: Amosa. *IEEE Transactions on evolutionary computacion*, 12:269–283, 2008.
- [4] Zvi Bodie, Alex Kane, y Alan J. Markus. *Principios de Inversiones.*, Capítulo Teoría de carteras., páginas 98–120. McGraw-Hill, 2004.
- [5] Arturo Morales Castro. Inversiones en acciones y portafolios de inversión. *Administraste Hoy*, 0:10, 2000.
- [6] Carlos A. Coello Coello. 20 years of evolutionary multi-objective optimization: What has been done and what remains to be done. Reporte técnico, CINVESTAV-IPN, 2006.
- [7] CONASEV. *Dos caminos para invertir en la bolsa*. INDECOPI Protección al inversionista.
- [8] Kalyanmoy Deb y David E. Goldberg. An investigation of niche and species formation in genetic function optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 1:42–50, 1989.
- [9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, y T. Meyarivan. A fast and elitst multi-objective genetic algorithm: NSGA-II. Reporte técnico, Indian Institute of technology Kanpur, 2001.
- [10] Laura Diosan. A multi-objective evolutionary approach to the portfolio optimization problem. Reporte técnico, Babes-Bolyai University, Kogalniceanu, Cluj-Napoca, Romania, 2005.

- [11] Karl Doerner, Walter J. Gutjahr, Richard F. Hartl, Christine Strauss, y Christian Stummer. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Reporte técnico, Department of Management Science, University of Vienna, 2002.
- [12] Salvador Fernando Castro Enciso. Creación de portafolios de inversión utilizando algoritmos evolutivos multiobjetivo. Master's thesis, Centro de investigación y de estudios avanzados del instituto politécnico nacional, Agosto 2005.
- [13] K. V. Fernando. Practical portfolio optimization. Reporte técnico, Oxford, United Kingdom, 2003.
- [14] Enrique Gallego. El precio de las acciones. Bibiloteca Megabolsa, 2002.
- [15] David E. Goldberg. *Genetic Algorithms, in search, optimization & machine learning*. Addison-Wesley, 1989.
- [16] Ghada Hassan y Christopher Clack. Multiobjective robustness for portfolio optimization in volatile environments. *Genetic and Evolutionary Computation Conference*, 1:8, 2008.
- [17] <http://www.thegrafster.com/>. Economical software for investors, September 2007.
- [18] Improsa de valores. *Proceso de Gestión de Portafolios de Inversión*, 2005.
- [19] Dietmar Maringer. *Portfolio managment with heuristic optimization*. Springer, 2005.
- [20] Heli Orelma. Large patent portfolio optimization. Master's thesis, Helsinki University of Technology, 2007.
- [21] Manuel Valenzuela Rendón. *Métodos de selección en algoritmos genéticos*. Instituto Tecnológico y de Estudios Superiores de Monterrey, 2000.
- [22] Manuel Valenzuela Rendón. *Aspectos practicos de la aplicación de algoritmos genéticos*. Instituto Tecnológico y de Estudios Superiores de Monterrey, 2004.
- [23] Manuel Valenzuela Rendón. *Implementación de un algoritmo genético en una aplicación*. Instituto Tecnológico y de Estudios Superiores de Monterrey, 2004.
- [24] N. Srinivas y Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic aigorithms. *Evolutionary Computation*, 2:221–248, 1995.
- [25] Hamdy A. Taha. *Investigación de operaciones*, Capítulo Algoritmos con restricciones, páginas 93–97,797–800. 1997.

- [26] Manuel Valenzuela-Rendón y Eduardo Uresti-Charre. A non-generational genetic algorithm for multiobjective optimization. *Proceedings of the Seventh International Conference on Genetic Algorithms*, 1997.
- [27] David A. Wood. Portfolio optimization benefits from integrating analysis of risk, strategy and valuation. *Oil & Gas Journal; ABI/INFORM Global*, 2002.
- [28] Eckart Zitzler, Marco Laumanns, y Lothar Thiele. Spea 2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary methods for design optimisation and control*, 1:6, 2002.
- [29] Eckart Zitzler y Lothar Thiele. An evolutionary algorithm for multiobjective optimization the strength pareto approach. Reporte técnico, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology ETH, 1998.
- [30] Alaitz Mendizabal Zulveldia, Luis M.^a Miera Zablaza, y Marian Zubia Zubiaurre. El modelo de Markowitz en la gestión de carteras. *Cuadernos de Gestión*, 2:14, 2002.

Tecnológico de Monterrey, Campus Monterrey



30002007186885

<http://biblioteca.mty.itesm.mx>