

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**RESOLUCIÓN DEL PROBLEMA DE
SELECCIÓN DE PORTAFOLIO BI-OBJETIVO
UTILIZANDO ALGORITMOS CULTURALES**

CLAUDIA ANDREA VASCONCELLOS GAETE

TESIS DE GRADO
MAGISTER EN INGENIERIA INFORMÁTICA

SEPTIEMBRE 2011

Pontificia Universidad Católica De Valparaíso
Facultad De Ingeniería
Escuela De Ingeniería Informática

**RESOLUCIÓN DEL PROBLEMA DE
SELECCIÓN DE PORTAFOLIO BI-OBJETIVO
UTILIZANDO ALGORITMOS CULTURALES**

CLAUDIA ANDREA VASCONCELLOS GAETE

Profesor Guía: **Guillermo Cabrera Guerrero.**

Programa: **Magíster en Ingeniería Informática**

Septiembre 2011

Agradecimientos

Primero que todo, quiero agradecer a mi familia por ser mi apoyo incondicional en todo momento, y en especial durante estos largos años de estudio. A mi mamá, porque siempre confió en mí (aunque sé que en más de alguna ocasión tuvo motivos de sobra para preguntarse si esto terminaría algún día), a mi papá, porque con su ejemplo me enseñó el valor de la paciencia, y a mi hermano, quien tal vez sin darse cuenta, me regaló alegrías infinitas en momentos difíciles. De modo especial, quiero agradecer a mi abuela, que me muestra día a día que la resistencia ante la adversidad no tiene límites.

Al profesor Guillermo Cabrera, guía de esta tesis, por la ayuda y respaldo que me otorgó cada vez que necesité de un buen consejo para llevar a buen puerto este trabajo.

Finalmente, las palabras que encontrarán al final son un extracto de los cinco preceptos que me han acompañado estos últimos 9 años, y que resumen de algún modo el fin último de mis pensamientos y acciones.

“Hitotsu! Doryoku no seichin o yashinau koto” (“Tratar de superarse” –Dojo Kun)

Índice General

1. Introducción	7
2. Objetivos	8
2.1. Objetivo General.....	8
2.2. Objetivos Específicos	8
3. Estado del Arte	9
3.1. Problema de Selección de Portafolio	9
3.1.1. Modelo Media-Varianza de Markowitz.....	9
3.1.2. Variaciones al problema original.....	10
3.1.3. Otros mecanismos de resolución para PSP.....	12
3.2. Algoritmos Culturales.....	14
3.2.1. Antecedentes históricos	14
3.2.2. Espacio de Creencias	15
3.2.3. Población	16
3.2.4. Protocolo de comunicación entre los distintos espacios.	17
3.2.5. Aplicaciones de CA	17
3.3. Optimización Multiobjetivo.....	18
3.3.1. Conceptos Básicos.....	18
3.3.2. Manejo de Restricciones.....	20
3.4. Evaluación de Técnicas de Optimización Multiobjetivo	25
3.4.1. Relaciones de Excelencia	25
3.4.2. Métricas de Evaluación	26
4. Algoritmo Cultural Bi-objetivo para PSP	33
4.1. Formulación del Problema.....	33
4.2. Representación de PSP Bi-Objetivo a través de Algoritmos Culturales.....	34
4.2.1. Individuos	34
4.2.2. Tratamiento de individuos infactibles.....	34
4.2.3. Población	35
4.2.4. Espacio de Creencias.....	35
4.2.5. Influencia del Espacio de Creencias en la Población.....	37
4.2.6. Almacenamiento de las soluciones no-dominadas	37
4.2.7. Parámetros de la solución.	38
4.3. Descripción del Algoritmo.....	39
5. Ajuste de Parámetros y Resultados	40
5.1. Condiciones generales	40
5.2. Ajuste de Parámetros	41
5.2.1. Número de individuos	41
5.2.2. Número de generaciones	42
5.2.3. Probabilidad de mutación	43
5.2.4. Número de acciones que conforman el portafolio	45

5.3. Resultados.....	47
6. Conclusiones y Trabajo Futuro.....	51
6.1. Conclusión	51
6.2. Trabajo Futuro	52
7. Bibliografía.....	53
Anexos.....	57
A. Detalle de las pruebas de ajuste de parámetros	57

Lista de Figuras

Figura 3.1: Diversas funciones objetivo propuestas por Ehrgott.	12
Figura 3.2: Esquema de funcionamiento de un Algoritmo Cultural	15
Figura 3.3: Relación entre el Frente de Pareto y las variables del problema	19
Figura 3.4: Ejemplo de dominancia fuerte.	25
Figura 3.5: Ejemplos de cobertura de conjuntos.	26
Figura 3.6: Superficie alcanzable.	28
Figura 4.1: Representación de un individuo.	34
Figura 4.2: Representación de la población	35
Figura 4.3: Parte normativa multiobjetivo.	35
Figura 4.4: Rejilla para dos funciones objetivo con ocho intervalos.	36
Figura 4.5: Actualización de la rejilla	37
Figura 5.1: Comparación de resultados para P1 (Hang Seng), con $K=4$	48
Figura 5.2: Comparación de resultados para P5 (Nikkei), con $K=8$	48
Figura 5.3 – Frentes de Pareto obtenidos para cada una de las instancias de la Librería-OR	50

Lista de Tablas

Tabla 3-1: Resumen de métricas de evaluación para problemas multiobjetivo	31
Tabla 4-1: Parámetros de la solución propuesta.....	38
Tabla 5-1: Instancias tomadas desde OR-Library	40
Tabla 5-2: Diferencias entre los frentes calculados y las fronteras ideales	49
Tabla 7-1: Resultados al ajuste de la cantidad de individuos.	58
Tabla 7-2: Resultados al ajuste de la cantidad de generaciones	59
Tabla 7-3: Resultados al ajuste de la probabilidad de mutación	60
Tabla 7-4: Resultados al ajuste del número de acciones a incluir en el portafolio.....	60

Lista de Abreviaturas.

1. **ACO** : Ant Colony Optimization
2. **CA** : Cultural Algorithm
3. **GA** : Genetic Algorithm
4. **PSP** : Portfolio Selection Problem
5. **SA** : Simulated Annealing
6. **S&P** : Standard & Poors
7. **VS** : Version Spaces

Lista de Símbolos

\prec Relación de dominancia de un conjunto sobre otro.

P^* Conjunto de óptimos de Pareto

Resumen

La gestión de portafolios de inversión es una de las tareas más complejas que se dan en el mundo financiero, ya que cada decisión requiere de un riguroso análisis antes de ser ejecutada. Dentro de los problemas típicos que debe enfrentar un administrador de carteras se encuentra el Problema de Selección de Portafolio (PSP), el cual consiste en armar una cartera de inversiones tal que el retorno sea máximo, pero intentando asumir el menor nivel de riesgo posible según los intereses del inversor. Este problema que a simple vista parece ser simple en su definición, tiene muchas posibilidades de solución debido a las combinaciones de acciones y ponderaciones (dentro del portafolio) que se pueden formar.

En este trabajo de título, se intenta solucionar PSP a través de la implementación de un algoritmo cultural multiobjetivo que maximice la rentabilidad y minimice el riesgo del portafolio al mismo tiempo. Las ventajas de utilizar un algoritmo evolutivo radican en la posibilidad de explorar de forma simultánea y diversa un gran número de soluciones, lo cual ataca directamente al problema de “explosión combinatorial” existente en PSP. A su vez, el enfoque multiobjetivo, le otorga la posibilidad al algoritmo de analizar ambos objetivos simultáneamente y en la misma proporción.

Los resultados obtenidos indican que la solución propuesta es factible, considerando que el conjunto de soluciones obtenido posee una buena proximidad a la frontera eficiente de la librería OR y es calculada en un tiempo razonable; sin embargo, es necesario acotar que el algoritmo no logra superar a otras propuestas de naturaleza evolutiva, como en es el caso de Jürgen Branke en [Branke, 2009], el cual utiliza un algoritmo evolutivo basado en envolturas.

Palabras Clave: *Problema de Selección de Portafolio, Algoritmos Culturales, Optimización Multiobjetivo, Metaheurísticas*

Abstract

Portfolio management is one of the most complex tasks inside financial world, considering that every decision involves a rigorous analysis before execute it. One of the typical problems that a portfolio manager should face is the Portfolio Selection Problem (PSP), which consist into build a portfolio with the maximum return possible but taking the less risk possible according to the risk aversion of the investor. Even considering the easy definition for the problem, the truth is that there are infinite possibilities of solving, just taking into account the multiple combinations between stocks to include and weights to apply to the selected stocks.

The aim of this study is bring a solution for PSP using a cultural algorithm with a multiobjective approach to maximize return and minimize risk in the portfolio, -both objectives at the same time-. The advantages of using an evolutionary algorithm focuses into the possibility of explore a large space with several solutions and in a simultaneous way, attacking directly to the “combinatorial explosion” present in all combinatorial optimization problems. Also, the multiobjective approach allows to the algorithm to analyze both objectives at the same time and with the same importance.

The results obtained demonstrate that proposed solution is feasible, considering the proximity to efficient frontiers (taken from OR-Library) and the small time spent to get those results; however, the algorithm could not beat another evolutionary techniques proposed for authors like Jüergen Branke at [Branke, 2009], who implemented an envelope-based solution.

Keywords: *Portfolio Selection Problem, Cultural Algorithm, Multiobjective Optimization, Metaheuristics.*

1. Introducción

El Problema de Selección de Portafolio (PSP), consiste en armar una cartera de inversiones tal que el retorno sea máximo al menor nivel de riesgo posible. Este problema, -abordado por Markowitz en 1952-, supone de antemano la racionalidad del inversor, por lo cual busca de modo simultáneo la maximización de las ganancias, pero corriendo el menor riesgo posible (o bien, minimizar las pérdidas y asumir un mayor riesgo, considerando que riesgo es sinónimo de rentabilidad). La solución propuesta originalmente, consistía en elegir de antemano una de las dos versiones posibles (maximizar rentabilidad o minimizar pérdidas), la cual se enfoca en sólo una de las variables y deja de lado la optimización de un segundo objetivo.

Si bien la solución original ha sido ampliamente estudiada y reconocida en el mundo financiero, lo cierto es que desde el punto de vista de la optimización, se está desechando la posibilidad de analizar el efecto combinado de tener múltiples objetivos, ya que el resultado estará en función de sólo un objetivo, mientras que el otro objetivo será considerado como parte de las restricciones del problema.

Sin embargo, es posible aplicar técnicas de resolución que apunten a la optimización simultánea de diversos objetivos para un mismo problema. Para ello, primero es necesario recurrir a los conceptos de Optimización Multiobjetivo, -cuyos orígenes se remontan a 1881, con la idea de “óptimo” de Francis Edgeworth y Wilfredo Pareto- y aplicarlos al Problema de Selección de Portafolio. Conceptos como “Óptimo de Pareto”, “Dominancia de Pareto” y “Frente de Pareto” aparecerían representados en términos de portafolios, acciones, rentabilidad y riesgo asociado. Luego de esto, se estaría en condiciones de seleccionar una técnica de resolución apropiada para el problema en particular

Ahora bien, en cuanto a la optimización a través de algoritmos evolutivos, los Algoritmos Culturales [Reynolds, 1994] son una metaheurística de resolución de problemas basada en la idea de que el conocimiento se logra de modo natural en las sociedades, al combinar el elemento generacional con la influencia de la cultura y creencias que subyacen en ella. A pesar de ser relativamente reciente, ha cobrado mucha relevancia en el ámbito de optimización multiobjetivo, debido fundamentalmente a su capacidad para explorar espacio de soluciones tanto a nivel local como global, eficiencia -en términos de su capacidad de generar en una sola iteración una gran cantidad de soluciones debido a que pueden trabajar con poblaciones de datos, y finalmente, porque no son susceptibles a las discontinuidades que se puedan dar en el Frente de Pareto [Coello, 2002].

Este trabajo se conducirá de la siguiente manera: el capítulo 2 estará dedicado a explicar el marco metodológico del proyecto, describiendo objetivos y método de trabajo propuesto. El capítulo 3 contendrá el Estado del Arte para el problema aquí planteado, -vale decir-, se presentará el Problema de Selección de Portafolio, los conceptos de Optimización Multiobjetivo y la descripción de la técnica evolutiva de Algoritmos Culturales. En el capítulo 4, se describirá el Problema de Selección de Portafolio desde el punto de vista de la optimización multiobjetivo resuelta con Algoritmos Culturales, presentando el diseño e implementación de una solución de tipo multiobjetivo.

2. Objetivos

2.1. Objetivo General

Dar una solución al Problema de Selección de Portafolio utilizando Algoritmos Culturales y Optimización Multiobjetivo, de modo de maximizar la rentabilidad y minimizar el riesgo

2.2. Objetivos Específicos

- a.) Entregar un enfoque de Inteligencia Computacional al Problema de Selección de Portafolio, a través del uso de Algoritmos Culturales.
- b.) Implementar una solución para el Problema de Selección de Portafolio basada en Optimización Multiobjetivo Evolutiva y Algoritmos Culturales.
- c.) Evaluar el uso de Optimización Multiobjetivo mediante Algoritmos Culturales aplicado al Problema de Selección de Portafolio y comparar su rendimiento con implementaciones propuestas por otros autores.

3. Estado del Arte

3.1. Problema de Selección de Portafolio

3.1.1. Modelo Media-Varianza de Markowitz

En el año 1952, Henry Markowitz [Markowitz, 1952] propone un método de selección de activos para un portafolio, considerando de forma explícita los rasgos fundamentales de lo que en un principio se puede calificar como “conducta racional” del inversor, consistente en buscar aquella composición del portafolio que haga máxima la rentabilidad para un determinado nivel de riesgo, o bien, un mínimo el riesgo para una rentabilidad dada.

El modelo de Markowitz considera las siguientes hipótesis:

1. La rentabilidad de cualquier activo o portafolio, es una variable aleatoria de carácter subjetiva, cuya distribución de probabilidad para el periodo de referencia es conocido por el inversor. El valor medio (esperanza) de dicha variable aleatoria se acepta como medida de la rentabilidad de la inversión.
2. Se acepta como medida del riesgo la dispersión, medida por la varianza o la desviación estándar, de la variable aleatoria que describe la rentabilidad, ya sea de un valor individual o de una cartera.
3. La conducta del inversor lo llevará a preferir aquellas carteras con una mayor rentabilidad y menor riesgo.

Con lo anterior, el método para confeccionar un portafolio de acciones se reduce a dos etapas. En la primera se determina el conjunto de portafolios eficientes que proporcionan la máxima ganancia para un riesgo dado, o bien, proporciona el mínimo riesgo para un valor dado de ganancia, y después se determina el portafolio óptimo como aquél que produce una mayor rentabilidad para un riesgo dado.

La conclusión final es que si la correlación entre la rentabilidad de los activos es perfecta y negativa, la diversificación puede hacer desaparecer completamente el riesgo de la cartera y la rentabilidad de la cartera viene dado por el punto de equilibrio.

Luego, la formulación original propuesta por Markowitz consiste en minimizar el riesgo para un nivel de rentabilidad dado; matemáticamente esto corresponde a:

$$\text{Minimizar} \quad \sigma^2(Rp) = \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \quad (3.1.1.1)$$

$$\begin{aligned}
\text{Sujeto a:} \quad E(R_p) &= \sum_{i=1}^n w_i E(R_i) & (3.1.1.2) \\
1 &= \sum_{i=1}^n w_i \\
w_i &\geq 0, i \in [1..n]
\end{aligned}$$

Donde:

w_i : Peso del activo i dentro del portafolio p

R_p : Retorno del portafolio p

$E(R_p)$: Retorno esperado para el portafolio p .

$\sigma^2(R_p)$: Riesgo asociado al portafolio p

3.1.2. Variaciones al problema original

Sin duda que el trabajo de Markowitz representó un gran aporte al problema de selección de activos para una cartera de inversión; sin embargo, trabajos posteriores han postulado que la falta de elementos de la vida real en la formulación del problema genera una barrera a la precisión de los resultados obtenidos.

Numerosos autores han investigado variaciones al problema original. Entre ellos, Crama y Schyns [Crama y Schyns, 2003] en su investigación sobre Simulated Annealing aplicado al Problema de Selección de Portafolio, entregan una extensa lista de autores que consideraron restricciones adicionales, así como diversos enfoques para su resolución.

Giacomo di Tollo, en su trabajo acerca de meta-heurísticas como método de resolución del Problema de Selección de Portafolio [DiTollo, 2006], menciona al menos tres restricciones que deberían ser incorporadas al problema pero que, sin embargo, elevan la complejidad (en términos computacionales) hasta hacerlo intratable aún para pequeñas instancias. Las restricciones a incorporar serían:

- a. Cardinalidad:** El número de activos que conformarán el portafolio debe ser limitado. Esto se logra introduciendo una variable de decisión z (1 en caso de pertenecer al portafolio y 0 en otro caso) para cada activo “candidato”. Luego, es necesario establecer que la suma de las variables de decisión debe ser igual número de activos requeridos en el portafolio.

$$\sum_{i=0}^n z_i = k \quad (3.2.1.1)$$

- b. Mínimos y máximos de cada constituyente del portafolio:** Para evitar el manejo de pequeños lotes de activos, o bien, para evitar la sobre-exposición a la variación de un único activo, es necesario incluir información sobre el rango permitido para cada constituyente del portafolio.

$$\varepsilon_i z_i \leq w_i \leq \delta_i z_i \quad (3.2.1.2)$$

- c. Lotes mínimos de transacción:** La conformación de un portafolio de acciones significa comprar una cierta cantidad de una determinada acción. En la formulación clásica del problema (y con el fin de evitar mayor complejidad), la cantidad de acciones a transar se considera como una variable continua, mientras que en la realidad los activos se transan por medio de múltiples “lotes mínimos” de acciones. Esto transforma el problema original, -de naturaleza continua-, en un problema mixto de tipo entero-continuo, en el cual la restricción a la cantidad mínima a transar se puede escribir en términos de Si p_j (precio para el activo j), y ml_j (cantidad mínima a transar).

$$c_j = ml_j p_j \quad (3.2.1.3)$$

Otra restricción frecuentemente citada se refiere a la rotación de los constituyentes dentro del portafolio, -vale decir, a las compras y ventas de un activo, necesarias para la conformación del portafolio-. Estas restricciones se expresan como límites superiores a la variación de los activos dentro del portafolio entre un periodo y otro. Aquí $x_i(0)$ corresponde al peso del activo i en el portafolio inicial, B_i es la compra máxima y S_i es la venta máxima del activo i en el periodo actual ($1 \leq i \leq N$). Nótese que estas restricciones también pueden ser modeladas indirectamente mediante la incorporación de costos de transacción a la función objetivo o dentro de otras restricciones

$$\max\{x_i - x_i^{(0)}, 0\} \leq B_i \quad \text{Rotación en las compras} \quad (3.2.1.4)$$

$$\max\{x_i^{(0)} - x_i, 0\} \leq S_i \quad \text{Rotación en las ventas} \quad (3.2.1.5)$$

$$1 \leq i \leq N$$

Destaca también el trabajo de Ehrgott [Ehrgott et al., 2004], que propone un modelo multiobjetivo basado en la idea de maximizar la utilidad global (inherente a cualquier problema de optimización) del portafolio. Para ello, considera distintas funciones relativas al rendimiento del portafolio en distintos periodos de tiempo (12 meses, 3 años, etc.), así como criterios estándares de riesgo establecidos por alguna agencia de análisis económico internacional (en este caso, el ranking entregado por la agencia S&P). La siguiente figura grafica los criterios antes expuestos:

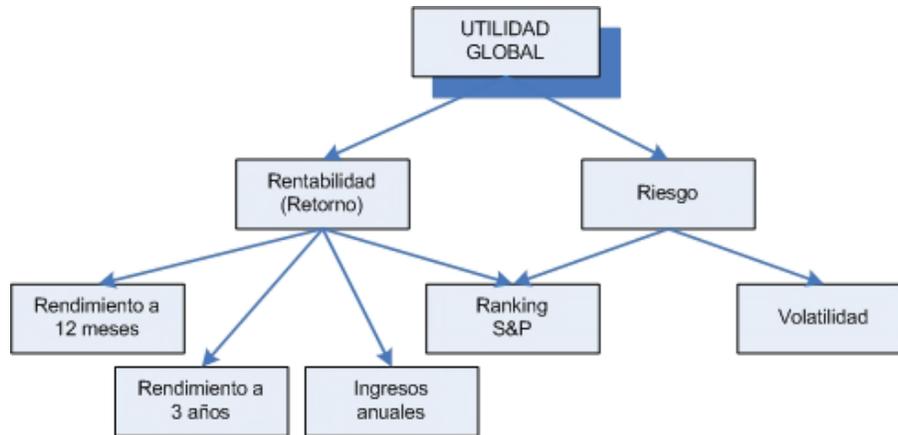


Figura 3.1: Diversas funciones objetivo propuestas por Ehrgott.

3.1.3. Otros mecanismos de resolución para PSP

Desde su formulación en 1952, el problema de Selección de Portafolio se ha resuelto a través de diversos métodos, pasando por técnicas matemáticas (Lagrangeano), técnicas estadísticas, hasta llegar a la aplicación de metaheurísticas (Simulated Annealing).

- a. **Lagrangeano:** Consiste básicamente en tomar un problema de optimización y transformarlo en un problema sin restricciones, por medio de la aplicación de multiplicadores de LaGrange. En esta técnica, destaca el trabajo de Gregory C. Chow [Chow, 1996] quien propone el uso del Lagrangeano para la optimización de portafolios, así como para otras decisiones de inversión.
- b. **Programación Cuadrática:** La programación cuadrática trata el problema de minimizar una función objetivo cuadrática sujeta a restricciones de desigualdad lineales sobre las variables o bien a restricciones de igualdad. Asimismo, se ha probado que estos problemas pueden ser resueltos (o puede mostrarse que no son factibles) en un número finito de iteraciones, pero el esfuerzo requerido para encontrar una solución depende fuertemente en las características de la función objetivo y del número de restricciones de desigualdad [Czyzyk et al., 1999].

A través de Programación Cuadrática, el modelado de PSP es:

Versión 1:	Minimizar riesgo dada la rentabilidad	
<i>Minimizar</i>	$\frac{k}{2} w^T Q w$	(3.1.3.1)

$$\begin{aligned}
 \text{Sujeto a:} \quad & \sum_{i=1}^n r_i w_i \geq r \\
 & \sum_{i=1}^n w_i = 1, w \geq 0
 \end{aligned}
 \tag{3.1.3.2}$$

Versión 2: Maximizar retorno dada una varianza

$$\text{Maximizar} \quad \sum_{i=1}^n r_i w_i
 \tag{3.1.3.3}$$

$$\begin{aligned}
 \text{Sujeto a:} \quad & \frac{k}{2} w^T Q w \leq \sigma^2 \\
 & \sum_{i=1}^n w_i = 1, w \geq 0
 \end{aligned}
 \tag{3.1.3.4}$$

- c. Metaheurísticas:** La utilización de metaheurísticas para el Problema de Selección de Portafolio es aún de reciente aplicación, pero cada vez gana más fuerza debido a los buenos resultados obtenidos. En [DiTollo y Rolli, 2006] se habla acerca de las “oportunidades para las metaheurísticas”, refiriéndose a la característica de intratabilidad del problema luego de añadirle restricciones propias de la selección de portafolio real, y de cómo esto puede ser utilizado por las metaheurísticas, considerando su capacidad de adaptación y simpleza en cuanto al método de resolución. Bajo esta misma idea, destaca el trabajo de Crama y Schyns [Crama y Schyns, 2003] en el cual resuelven el problema con adición de restricciones mediante la aplicación de la metaheurística de Simulated Annealing.

3.2. Algoritmos Culturales

3.2.1. Antecedentes históricos

Los Algoritmos Culturales, -propuestos por en [Reynolds, 1994]- surgen a partir de la idea de que la cultura (como conjunto de todas las formas y expresiones de una sociedad determinada) puede ser simbólicamente codificada y transmitida dentro y entre poblaciones, como una medio de herencia. Usando esta idea, Reynolds desarrolló un modelo computacional en el cual la evolución cultural es vista como un proceso hereditario que opera a nivel macro y micro evolutivo.

La cultura puede entenderse como un conjunto de fenómenos ideológicos compartidos por una población, y que modelan de uno u otro modo la forma en que un individuo interpreta sus experiencias y decide su comportamiento. La parte del sistema que es compartida por la población, se denomina *conocimiento*, se genera a consecuencia de los datos recolectados por miembros de la sociedad y se presenta de tal forma que sea potencialmente accesible a todos. Asimismo, existe una parte de la cultura que es individual: la *interpretación del conocimiento* (codificado como un conjunto de símbolos) y los comportamientos que trae como consecuencia su asimilación; también la parte individual incluye las experiencias vividas, y la forma en que éstas pueden aportar algo al conocimiento compartido. Reynolds intenta captar ese fenómeno de herencia doble en los algoritmos culturales diseñando 2 “niveles” claramente definidos: el macro-evolutivo y el micro-evolutivo; el objetivo es incrementar las tasas de aprendizaje o convergencia, y de esta manera, que el sistema responda mejor a un gran número de problemas.

En el nivel micro evolutivo, los individuos son descritos en términos de comportamiento (el cual puede ser “aceptable” o “inaceptable”). Estas características son pasadas de generación en generación usando distintos operadores. En el nivel macro evolutivo, los individuos son capaces de generar la “mappa” como descripciones generalizadas de sus experiencias; esta “mappa” individual puede ser mezclada y modificada para conformar la “mappa grupal” usando un conjunto de operadores (genéricos o específicos al problema). Entre ambos niveles existe un enlace. Para el nivel micro-evolutivo, Reynolds propone utilizar Algoritmos Genéticos (GA), y Version Spaces para el proceso macro-evolutivo.



Figura 3.2: Esquema de funcionamiento de un Algoritmo Cultural

El pseudo-código para la implementación de CA:

Algoritmo - Algoritmo Cultural

1. Generar la población inicial
 2. Iniciar el espacio de creencias
 3. Evaluar población inicial
 4. Repetir
 - 4.1. Actualizar el espacio de creencias con individuos aceptados
 - 4.2. Variar la población, influenciado por el espacio de creencias
 - 4.3. Evaluar cada hijo
 - 4.4. Realizar la selección
-

3.2.2. Espacio de Creencias

Es el lugar donde se residirán los conocimientos que han adquirido los individuos en generaciones anteriores. La información contenida en este espacio debe ser accesible a cualquier individuo, quien puede utilizarla para modificar su comportamiento.

Dentro de un algoritmo cultural, la actualización de este espacio corresponde al ciclo principal; es en ese momento donde el espacio de creencias incorpora las experiencias individuales de un grupo selecto de individuos. Tal grupo se obtiene entre toda la población con la *Función de Aceptación*.

Kobti [Kobti, 2004] postula que existen al menos cinco categorías de conocimiento que son relevantes para cualquier modelo evolutivo-cultural, ya que describen distintos aspectos relativos al espacio de creencias y a la forma en como este se estructura. Estos conocimientos son:

- a. **Situacional:** Contiene ejemplos específicos de eventos importantes para la población (Ej.: Encontrar una solución óptima)
- b. **Normativo:** Mantiene un conjunto de valores deseados dentro de la población (Ej.: comportamiento de un individuo)
- c. **Topográfico (Espacial):** Guarda información acerca de la topología del espacio de búsqueda
- d. **Histórico (Temporal):** Conocimiento acerca del historial del espacio de búsqueda (Ej.: Patrón del proceso de búsqueda)
- e. **De dominio:** Información acerca del dominio específico de conocimiento sobre el cual se está aplicando el algoritmo (Ej.: Finanzas)

En la formulación inicial para CA hecha por Reynolds, se propone que el Espacio de Creencias sea representado mediante el uso de Espacios de Versiones, el cual es un concepto del aprendizaje automático que representa el subconjunto de todas las hipótesis que son consistentes con los ejemplos de entrenamiento. Este conjunto contiene todas las hipótesis que no han sido eliminadas por provocar conflictos con los datos observados [Mitchell, 1978].

3.2.3. Población

Como en todos los métodos de computación evolutiva, la población corresponde a un conjunto de individuos. Cada individuo tiene un conjunto de características independientes de los otros, con las que es posible determinar su aptitud. A través del tiempo, y mediante un proceso de selección conocido como *Función de Aptitud*, tales individuos podrán ser reemplazados por algunos de sus descendientes, -en caso de éstos sean mejores que sus antecesores-.

Por otro lado, los operadores de variación de los individuos (como la recombinación o la mutación) son modificados por la *Función de Influencia*. La función de influencia ejerce cierta presión, para que los hijos resultantes de la variación se acerquen a los comportamientos deseables, y se alejen de los indeseables, según la información almacenada en el espacio de creencias.

En la formulación inicial para CA hecha por Reynolds, se propone el uso de Algoritmos Genéticos (GA) como método de selección y evolución del Espacio de Población.

Algoritmo - Algoritmo Genético

1. Generar la población inicial
 2. Evaluar la población inicial
 3. Repetir
 - 3.1. Seleccionar padres desde la población actual
 - 3.2. Recombinar y generar una nueva población
 - 3.3. Mutar la nueva población
 - 3.4. Evaluar cada individuo de la nueva población
-

3.2.4. Protocolo de comunicación entre los distintos espacios.

Para poder captar a fondo la metáfora de la doble herencia en la formación de conocimiento, se propone una interfaz de comunicación entre el espacio de creencias y la población. Los mejores individuos de la población podrán actualizar el espacio de creencias mediante la *Función de Aceptación*, mientras, la población podrá verse afectada por las distintas categorías de conocimiento a través de la *Función de Influencia*, la cual es capaz de alterar las acciones de un individuo en particular

3.2.5. Aplicaciones de CA

Algunas aplicaciones de Algoritmos Culturales han sido:

- a. **Mejoramiento del rendimiento de redes semánticas:** El trabajo propuesto por Rychtyckyj [Rychtyckyj y Reynolds, 1999] describe el uso de CA para mejorar la eficiencia del algoritmo de inclusión en la red, lo cual permitirá determinar si un nodo en la red es hijo de otro nodo; y a su vez utilizar esta información como parte del algoritmo de clasificación de nodos dentro de las aplicaciones basadas en la red semántica.
- b. **Aplicación a bases de datos espacio-temporales:** Reynolds propone en [Reynolds y Nazzal, 1997] el uso de CA para generar redes de sitios dentro de una base de datos histórica acerca de sitios en el Valle de Oaxaca. Estas redes son descritas según árboles de expansión de múltiples criterios calculados mediante Algoritmos Culturales con programación evolutiva. Los resultados son utilizados para explicar los cambios en la decisión de localización de sitios del valle.
- c. **Mantenimiento de bases de conocimiento:** Reynolds [Reynolds y Rychtyckyj, 2002] presenta un esquema basado en CA para mejorar el mantenimiento de la base de conocimientos en el entorno dinámico que involucra la planificación del proceso de ensamblado de vehículos para la Ford Motor Company. Aquí, el uso de CA se combina con un enfoque bottom-up y se aplica al conocimiento existente, permitiendo así descubrir nuevos conocimientos

3.3. Optimización Multiobjetivo

3.3.1. Conceptos Básicos

En un problema de optimización, se combinan los siguientes elementos:

- a. **Variables de decisión:** Son todas las expresiones del problema cuyos valores pueden ser modificados y que finalmente, expresan un valor tal que permite dar una solución al problema planteado.
- b. **Función(es) objetivo:** Función que se expresa en términos de las variables de decisión y que se desea optimizar (minimizar o maximizar). Es necesario, que en cada problema de optimización exista al menos una de estas funciones (caso monoobjetivo); en caso de haber mas de una, entonces el problema será de tipo multiobjetivo.
- c. **Restricciones:** Desigualdades o ecuaciones que se tienen que cumplir para que la solución se considere factible. En caso de que el problema no presente restricciones, el problema se traduce a optimizar sólo las funciones objetivo.

En el caso de los problemas de optimización multiobjetivo, existen los tres elementos antes descritos, y se añade el hecho de que se mezclan diferentes funciones que deben ser optimizadas de forma simultánea y que, -por lo general-, se encuentran en conflicto entre ellas. Debido a ello, es necesario definir conceptos que ayuden a comprender y a establecer un marco de referencia apropiado para el trabajo con este tipo de problemas. Algunos de los conceptos más relevantes en este punto son:

1. **Problema general de optimización:** Encontrar el vector $\vec{x}^* = [x_1, x_2, \dots, x_n]^T$ tal que satisfaga “ m ” restricciones de desigualdad, “ p ” restricciones de igualdad y que optimice el vector de función f .

$$\text{Restricciones de desigualdad: } g_i(\vec{x}) \geq 0 \quad i = 1, 2, \dots, m \quad (3.3.1.1)$$

$$\text{Restricciones de igualdad: } h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p \quad (3.3.1.2)$$

$$\text{Vector Función } \vec{f}(\vec{x}) = [\vec{f}_1(\vec{x}_1), \vec{f}_2(\vec{x}_2), \dots, \vec{f}_k(\vec{x}_k)]^T \quad (3.3.1.3)$$

2. **Óptimo de Pareto:** Un punto x es un “óptimo de Pareto” si no existe vector x factible que haga disminuir algún criterio, sin causar un aumento significativo en al menos algún otro, considerando todo el espacio de las variables de decisión. En términos formales, un Óptimo de Pareto se define como:

$$\begin{aligned} \text{Si para cada } i \in \Omega \text{ y } I = \{1, 2, \dots, k\} \text{ tanto, } & \forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)) \\ \text{O haya al menos un } i \in \Omega \text{ tal que: } & f_i(\vec{x}) > f_i(\vec{x}^*) \end{aligned} \quad (3.3.1.4)$$

3. **Dominancia de Pareto:** Relación de orden parcial entre las variables en la cual un vector w “dominará” a un vector v , sólo en caso de que w sea “parcialmente menor” a v :

$$\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\}: u_i < v_i \quad (3.3.1.5)$$

4. **Conjunto Óptimo de Pareto:** Conjunto en el cual todos los puntos (soluciones) que lo conforman, corresponden a un óptimo para la función objetivo a tratar. Matemáticamente, esto se define como:

Para un problema $\vec{f}(x)$, el conjunto óptimo de Pareto (P^*) será: $P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \vec{f}(x') < \vec{f}(x)\}$ (3.3.1.6)

5. **Frontera de Pareto:** Se define como la región conformada sólo por puntos óptimos. Gráficamente, en esta región se encuentra el conjunto de óptimos de Pareto.

$$PF := \{\vec{u} = \vec{f} = (f_1(x), \dots, f_k(x)) \mid x \in P\} \quad (3.3.1.6)$$

En el caso de un problema bi-objetivo el concepto de Frente Pareto y su relación con las variables del problema se puede ilustrar del siguiente modo:

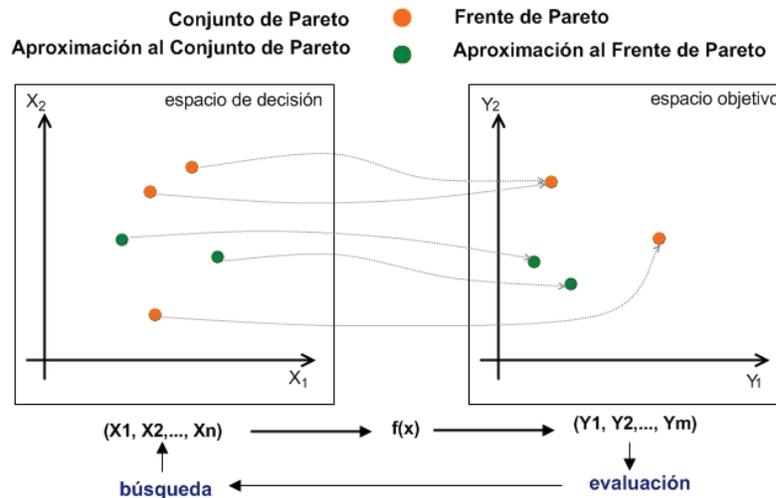


Figura 3.3: Relación entre el Frente de Pareto y las variables del problema

3.3.2. Manejo de Restricciones

Como se dijo anteriormente, las restricciones son desigualdades o ecuaciones que se tienen que cumplir para que la solución se considere factible; sin embargo, no todas las restricciones son iguales y debido a ello, se puede establecer una clasificación en base a diversas medidas. Según [Smith y Coit, 1997], la clasificación de las restricciones para un problema de optimización puede hacerse en base a:

- a. **Cantidad:** Algunos problemas multiobjetivo pueden ser reformulados utilizando algunos de los objetivos como restricciones; sin embargo, hay que tener en cuenta que la dificultad para satisfacer las restricciones será incremental a medida que estas aumentan.
- b. **Métrica:** Dependiendo del espacio en el cual se mueven las variables que la componen, se pueden clasificar en “continuas” o “discretas”.
- c. **Criticidad:** Este criterio responde a la pregunta: ¿Es necesario cumplir estrictamente esta restricción? Si la respuesta es si, entonces la restricción será de tipo “hard”, en otro caso, será de tipo “soft”.
- d. **Dificultad:** Depende del tamaño de la región factible comparada con el tamaño de espacio de muestras.

Asimismo, dependiendo del tipo de problema de optimización a tratar (monobjetivo o multiobjetivo) y de la técnica de resolución que se adoptará, los enfoques para manejo de restricciones irán variando. En [Coello, 1999] se presentan las siguientes técnicas de manejo de restricciones que pueden ser utilizadas junto a algoritmos evolutivos:

- a. Funciones de Penalización
- b. Mantenimiento de poblaciones factibles mediante representaciones especiales y operadores genéticos
- c. Separación de objetivos y restricciones
- d. Métodos híbridos
- e. Enfoque para optimización evolutiva multiobjetivo.

3.3.2.1. Funciones de Penalización

Hasta ahora, la forma más común de tratar las restricciones en problemas de optimización, ha sido el uso de **Funciones de Penalización**. Estas se basan en transformar un problema de optimización con restricciones a uno sin restricciones mediante la incorporación de las restricciones dentro de la misma función objetivo, pero multiplicadas por un factor de penalización. El modelo general para la aplicación de funciones de penalización es:

$$\Phi(\vec{x}) = F(\vec{x}) \pm \left[\sum_{i=1}^n R_i G_i(g_i(\vec{x})) + \sum_{j=1}^p C_j H_j(h_j(\vec{x})) \right] \quad (3.3.2.1.1)$$

En donde:

$\Phi(\vec{x})$	Función objetivo modificada.
$F(\vec{x})$	Función objetivo original
R_i	Factor de penalización para la restricción de desigualdad i
G_i	Función para la restricción de desigualdad i
$g_i(\vec{x})$	Restricción de desigualdad i
C_j	Factor de penalización para la restricción de igualdad j
H_j	Función para la restricción de igualdad j
$h_j(\vec{x})$	Restricción de igualdad j

En la tesis magistral de Landa [Landa, 2002] se describen algunos lineamientos generales (propuestos originalmente por Jon T. Richardson en 1989) para el diseño de funciones de penalización:

- Tendrán mejor desempeño aquellas funciones que dependen de la distancia a la zona de factible en vez de aquellas que únicamente dependen de las restricciones violadas.
- En problemas que tienen pocas restricciones, y una región factible pequeña, las funciones de penalización que sólo dependen del número de restricciones violadas probablemente no producirán soluciones factibles.
- Las buenas funciones de penalización deben considerar tanto el costo de cumplimiento máximo, como el costo de cumplimiento esperado (el costo de cumplimiento se refiere a la distancia a la región factible).
- Las funciones de penalización deben estar cerca del costo de cumplimiento esperado, pero sin ser inferiores a él frecuentemente. En caso contrario, es posible que no se encuentren soluciones factibles.

Para simplificar esto último, Coello presenta en [Coello, 2002] dos funciones de amplio uso para G_i y H_j , basándose en la idea de penalizar la restricción considerando la distancia de la violación hacia la región factible.

$$\begin{aligned} G_i(g_i(\vec{x})) &= \text{Max}(0, g_i(\vec{x}))^\beta \\ H_j(h_j(\vec{x})) &= |h_j(\vec{x})|^\gamma \\ \beta, \gamma &\in \{1, 2\} \end{aligned} \tag{3.3.2.1.2}$$

Sin embargo, en el trabajo de Kalyanmoy Deb [Deb, 2000] se exponen dos problemas asociados a la aplicación de funciones de penalización estáticas a problemas de optimización:

- a. La solución óptima para $\Phi(x)$ depende de los parámetros de penalización R_i y C_j . En general, se deben probar diferentes valores para estos parámetros antes de dar con aquél que conducirá la búsqueda en la región factible. Algunos investigadores han utilizado diferentes valores según el nivel de la restricción violada y otros han utilizado métodos sofisticados de penalización a través de generaciones, basados en parámetros que describen la tasa de evolución de la población.
- b. La inclusión de los términos de penalización distorsionan la función objetivo. Si se consideran valores pequeños, la distorsión puede ser pequeña, pero eso no significa estar cerca del “verdadero” óptimo. Por otro lado, si se consideran valores grandes, el óptimo de $\Phi(x)$ podría estar cerca, pero la distorsión podría ser tan severa que $\Phi(x)$ podría tener óptimos locales artificiales. Para evitar este último caso, el enfoque clásico de penalizaciones trabaja en secuencias, en donde los parámetros de penalización son incrementados gradualmente y la secuencia actual de optimización comienza desde el óptimo encontrado en la secuencia inmediatamente anterior. Aun así, los métodos más clásicos usan búsquedas basadas en gradiente descendiente, teniendo dificultades generalmente en espacios discretos y en el manejo de muchas variables.

3.3.2.2. Representaciones Especiales y Operadores Genéticos.

Esta técnica se basa en el empleo de Algoritmos Genéticos para la resolución del problema y en la aplicación de representaciones y operadores especiales, -diferentes a los típicamente utilizados-, y así lograr simplificar el espacio de búsqueda en problemas cuya principal característica es la de tener espacios de tipo convexos o bien, tener un número muy reducido de soluciones factibles. Aun así, su mayor desventaja es la de que este tipo de representaciones sólo es posible de aplicar a problemas específicos.

En [Coello, 1999] se describen variados ejemplos de aplicaciones que utilizan representaciones como estas. Algunas de ellas son: generación de trayectorias para robots, optimización de horarios y análisis de ADN, entre otras.

3.3.2.3. Separación de Restricciones y Objetivos

Algunos autores, como Paredis, proponen que restricciones y objetivos deben tratarse como dos sub-poblaciones diferentes, en donde cada individuo de la sub-población de restricciones se debe evaluar en base a la cantidad de individuos pertenecientes a los objetivos que violan las restricciones representadas.

Por otro lado, Kalyanmoy Deb [Deb, 2000] utiliza un enfoque en el que los puntos factibles se evalúan con un mejor fitness, ocupando para ellos torneos binarios regidos por las siguientes reglas:

1. Si un individuo es factible y el otro no, siempre gana el individuo factible.
2. Si ambos son factibles, gana el que tenga mejor valor de fitness.

3. Si ambos individuos no son factibles, gana el con la magnitud más baja de violación de restricciones.

Uno de los algoritmos más conocidos que utilizan este enfoque es COMOGA, sin embargo, este se revisará en la sección siguiente, ya que fue pensado para operar en problemas multiobjetivos.

3.3.2.4. Enfoque para Optimización Evolutiva Multiobjetivo (MOEA)

A diferencia de la optimización mono-objetivo, -en donde todos los esfuerzos se concentran en una sola función-, en el caso de optimización multiobjetivo se debe tener presente que los objetivos pueden tener conflictos entre si; entonces, encontrar el óptimo puede verse como un esfuerzo por tratar de generar un compromiso entre todas las funciones objetivo del problema.

Existen principalmente tres mecanismos tomados de la optimización evolutiva multiobjetivo que se utilizan para manejar restricciones:

- a. Uso de la dominancia de Pareto como un criterio de selección
- b. Uso de la jerarquía de Pareto para asignar una aptitud a los individuos
- c. Dividir la población en sub-poblaciones que serán evaluadas cada una con respecto a una restricción del problema o a la función objetivo original.

A continuación, se presentan una serie de algoritmos desarrollados para la resolución de problemas multiobjetivo a través de Algoritmos Evolutivos:

- a. **VEGA (*Vector Evaluated Genetic Algorithm*)**: Desarrollado por David Schaffer en 1984. Consiste en incorporar una estrategia de selección que considera los objetivos múltiples dentro de un algoritmo evolutivo; así, para un problema con ' m ' objetivos se generan ' m ' sub-poblaciones en las cuales sus individuos son seleccionados según cada uno de los objetivos, logrando así la selección de soluciones sólo localmente dominadas en cada generación. No utiliza dominancia de Pareto.
- b. **MOGA (*Multi-Objective Genetic Algorithm*)**: Propuesto por Fonseca y Fleming en 1993. Implementa una variante de la jerarquización de Pareto y utiliza rangos de dominancia para la asignación del fitness, en el cual los individuos no dominados tienen un rango de 1, en tanto que los individuos dominados son penalizados de acuerdo a la densidad de las regiones que lo dominan.
- c. **NPGA (*Niched-Pareto Genetic Algorithm*)**: Propuesto por Horn en 1993. La idea del método consiste en utilizar selección mediante torneo basada en la Dominancia de Pareto con respecto a un subconjunto de la población; en caso de empates se decide por "fitness sharing", tanto en el espacio de las funciones objetivo como en el espacio de las variables de decisión.

- d. **COMOGA (Constrained Optimisation by Multi-Objective Genetic Algorithms):** En [Surry y Radcliffe, 1995] se propone el uso de la jerarquía de Pareto para manejar restricciones. Acá la población se ordena basándose en las violaciones de restricción, contando el número de individuos dominados por cada solución. Fue utilizado para resolver un problema de diseño de una red de gas y mostró un mejor desempeño que una función de penalización. Su principal desventaja es que requiere de parámetros extra definidos por el usuario.
- e. **CAEP (Cultural Algorithms with Evolutionary Programming):** Basados en los buenos resultados presentados por los Algoritmos Culturales, Chung y Reynolds desarrollan en 1998 el algoritmo CAEP (“CAEP: An evolution-based tool for real-valued function optimization using cultural algorithms”). En él, los intervalos donde se encuentran buenas soluciones se dejaron dentro del espacio de creencias y se les llamó “Conocimiento Normativo”; también se incorporó el conocimiento situacional al espacio de creencias, que almacena los puntos exactos de los mejores individuos encontrados hasta el momento, logrando que los mejores individuos se conviertan en líderes a seguir.

Jin y Reynolds (en “Using knowledge-based evolutionary computation to solve non-linear constraint optimization problems: a cultural algorithm approach”) utilizan la parte normativa del espacio de creencias de Chung, pero eliminan la parte situacional y la sustituyen por un conocimiento de restricciones, que es un esquema basado en regiones. El conocimiento de restricciones en el espacio de creencias, toma la región determinada por los intervalos almacenados en el conocimiento normativo y divide esa región en “celdas de creencias”, las cuales se basan en las características de los individuos en la población e indican si la región es factible o no factible.

La parte de restricciones del espacio de creencias se encarga de crear una especie de mapa de navegación, para guiar a los individuos a la zona factible. Como las celdas son independientes entre sí, la zona factible puede ser no convexa, e incluso puede ser disjunta, sin que esto sea una dificultad para el manejo del conocimiento de restricciones.

Algoritmo - CAEP

1. Generar la población inicial de tamaño p .
 2. Evaluar la población inicial.
 3. Iniciar el espacio de creencias.
 4. Repetir
 - 4.1. Mutar la población, bajo la influencia del espacio de creencias.
 - 4.2. Evaluar cada hijo.
 - 4.3. Realizar torneos binarios con 'c' contrincantes aleatorios para cada individuo.
 - 4.4. Seleccionar los 'p' individuos con más victorias en los torneos.
 - 4.5. Actualizar el espacio de creencias con los individuos aceptados.
-

3.4. Evaluación de Técnicas de Optimización Multiobjetivo

Para evaluar los resultados obtenidos por distintas técnicas de optimización es necesario disponer de elementos que permitan establecer parámetros comunes a las técnicas en comparación.

3.4.1. Relaciones de Excelencia

En el trabajo de Farhang-Mehr [Farhang-Mehr y Azarm, 2003] se describen una serie de “*Relaciones de Excelencia*” (propuestas previamente por Hansen y Jaszkievicz) entre conjuntos, las cuales permiten establecer un orden estrictamente parcial entre dos conjuntos no-dominados y que por cierto, ayudan a evaluar cuan bueno es un conjunto respecto de otro.

Definición 1: Un conjunto no-dominado A gana *de modo fuerte* a otro conjunto no-dominado B (denotado por $A \text{ Ro } B$), si y sólo si A es distinto a B ($A \neq B$), y para cada y perteneciente a B existe un x (perteneciente a A) tal que $x \succ y$.

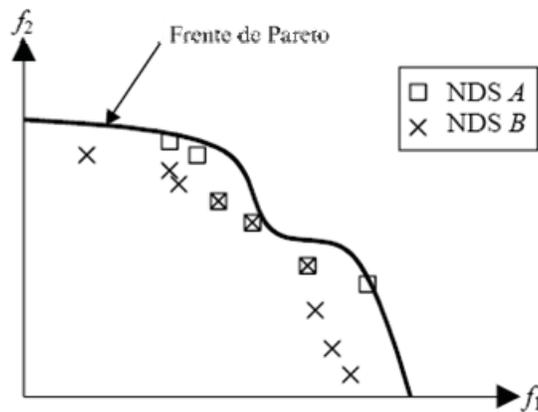


Figura 3.4: Ejemplo de dominancia fuerte.

Definición 2: Una relación de excelencia, denotada por R, está definida en un estricto orden parcial dentro de U, el cual se refiere a todas las soluciones no-dominadas que son objetivamente comparables con respecto a un aspecto de calidad en común (por ejemplo: las relaciones de cobertura).

Definición 3: Un conjunto no-dominado B, es *estrictamente superior* a otro conjunto no-dominado A en términos de cobertura (denotado por $B \text{ Rc } A$), si y sólo si todas las soluciones de A están contenidas en un cono convexo generado por el conjunto B, mientras exista al menos una solución del conjunto B que no esté contenido en el cono convexo generado por el conjunto A.

En este caso, se asumirá que el punto situado en el límite inferior del Frente de Pareto se encuentra en el origen Cartesiano del espacio objetivo.

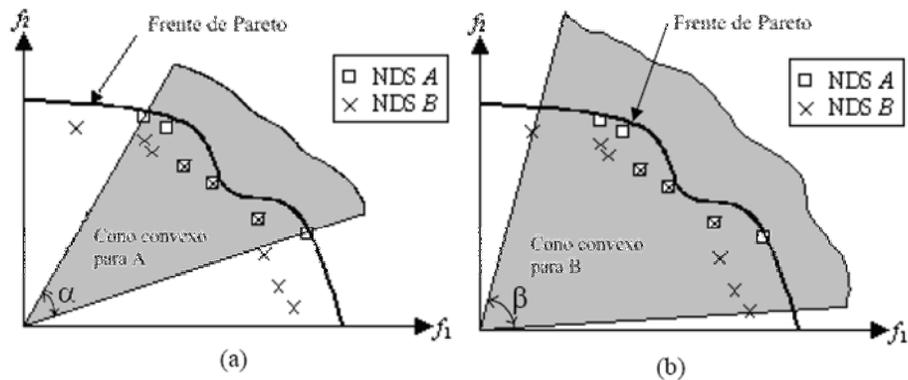


Figura 3.5: Ejemplos de cobertura de conjuntos.

3.4.2. Métricas de Evaluación

Aparte de las relaciones de excelencia, también es necesario medir de forma más cuantitativa el Frente de Pareto según criterios fundamentales:

- **Distancia del Frente de Pareto:** Suponiendo que el Frente de Pareto es conocido, se mide la distancia de las soluciones producido por el algoritmo en uso con respecto al verdadero frente. Mientras menor sea esta distancia, mejor será el algoritmo.
- **Distribución de soluciones obtenidas:** Se busca que sea máxima, de manera de contar con una distribución de vectores tan uniforme como sea posible.
- **Cantidad de elementos del conjunto de óptimos de Pareto:** Mientras más alto sea este valor, más soluciones óptimas estarán disponibles.

El problema fundamental de las métricas es que los 3 criterios deben combinarse de diferentes maneras para poder establecer el desempeño de un algoritmo de manera cuantitativa. Sin embargo, dichas combinaciones no son más que combinaciones lineales de pesos. Irónicamente, el problema de las métricas también es de naturaleza multiobjetivo.

A lo largo de la últimas dos décadas, diversos autores han descrito técnicas que permiten medir uno o varios de estos criterios, y también, se han propuesto nuevos criterios que otorguen nuevos puntos de vista para la evaluación del conjunto de soluciones encontrado. En orden cronológico, las distintas métricas que podemos encontrar serían:

3.4.2.1. Métrica de Esparcimiento (Spread metric)

En [Schott, 1995] se define una métrica para evaluar cuán eficiente es el espaciado del conjunto de soluciones, la cual se basa únicamente en calcular la distancia de cada punto del

conjunto obtenido hacia el punto más cercano que tenga en el Frente de Pareto. Para ello se debe calcular:

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - d_i)^2} \quad \begin{array}{l} N: \text{Tamaño de la población} \\ M: \text{Número de objetivos} \end{array}$$

Donde la distancia se considera como:

$$d_i = \min \left\{ |J_1^i - J_1^j| - |J_2^i - J_2^j| \right\} \quad (3.4.1)$$

La interpretación de esta métrica es que a menor valor de S, mejor la distribución en el conjunto. En algunos problemas, esta métrica puede estar relacionada con el número de soluciones obtenidas.

3.4.2.2. Promedio de Sumas Ponderadas

Introducida por Esbensen y Kuh en 1996 y mencionada en [Zitzler et al, 2000]. Esta métrica, el conjunto de soluciones obtenidas es evaluado respecto a una combinación lineal dada mediante la determinación del promedio de las sumas ponderadas mínimas de todos los vectores objetivo de X' correspondientes. Basado en este concepto, un ejemplo de combinación lineal es escogido de forma aleatoria con respecto a una distribución de probabilidad y las sumas ponderadas mínimas entre todas las combinaciones son sumadas y promediadas. El valor resultante es tomado como una medida de calidad. Una desventaja de esta métrica es que sólo la peor solución determinará el valor de calidad por combinación lineal.

3.4.2.3. “Superficie alcanzable”

Propuesto por Fonseca y Fleming en 1996 [Fonseca y Fleming, 1996] especialmente para comparar entre dos algoritmos multiobjetivos que generen un gran conjunto de soluciones.

Dados dos conjuntos de soluciones no-dominadas generados por dos algoritmos A₁ y A₂, se aplicará una función de “superficie alcanzable” que unirá los puntos de ambos conjuntos mediante líneas y dividirá el espacio objetivo en dos regiones, dependiendo de si los vectores objetivos asociados a sus correspondientes soluciones son o no son cubiertos por uno u otro conjunto. Como resultado, las muestras representadas por la superficie alcanzable media pueden ser evaluadas relativamente a través de pruebas estadísticas y así facilitar la comparación y ver que algoritmo sobrepasa al otro.

En el año 2002, Knowles y Corne [Knowles y Corne, 2002] amplían el estudio para algoritmos que correrán un cierto número de veces. Cada línea de intersección ahora tendrá un

mayor número de puntos de intersección. En este caso, se requerirá de pruebas estadísticas para determinar el porcentaje del algoritmo vencedor en cada sección.

Para ambos métodos, el resultado final serán dos números que determinaran el porcentaje en que un algoritmo vence al otro, lo cual puede ser visto como una desventaja, ya que no deja en claro cómo la diferencia de calidad puede ser expresada (o sea, cuanto mejor es un algoritmo que otro).

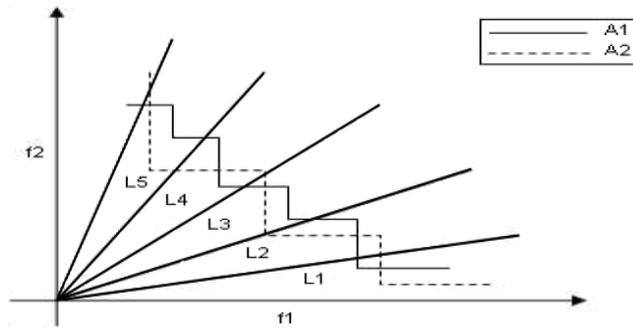


Figura 3.6: Superficie alcanzable.

3.4.2.4. Métricas de rendimiento de Hansen y Jazskiewicz (1998)

En 1998, Hansen y Jazskiewicz proponen usar métricas que valoren un conjunto de soluciones desde el punto de vista del tomador de decisiones en vez de sólo fijarse en los criterios “matemáticos” antes descritos.

Dados dos conjuntos no-dominados A y B, estas métricas utilizan diversas funciones de utilidad para determinar el número esperado de ocasiones en los que las soluciones de un conjunto serán mejores que el otro.

Métrica R1

$$R1(A, B, U, p) = \int_{u \in U} C(A, B, u) p(u) du \quad (3.4.2)$$

Donde U es el conjunto de todas las funciones de utilidad consideradas, $p(u)$ es la frecuencia de ocurrencia de la función de utilidad u y la función C se define como:

$$C(A, B, u) = \begin{cases} 1, & u^*(A) > u^*(B) \\ 0.5, & u^*(A) = u^*(B) \\ 0, & u^*(A) < u^*(B) \end{cases} \quad (3.4.3)$$

Métrica R2

Mide el grado de superioridad esperado. En este caso, $u^*(A)$ es el máximo valor de la función de utilidad u en el conjunto A .

$$R2(A, B, U, p) = \int_{u \in U} (u^*(A) - u^*(B)) p(u) du \quad (3.4.4)$$

Métrica R3

Mide la proporción esperada de superioridad

$$R3(A, B, U, p) = \int_{u \in U} \frac{(u^*(A) - u^*(B))}{u^*(A)} p(u) du \quad (3.4.5)$$

3.4.2.5. Proporción de cobertura de conjuntos.

En 1999 Zitzler y Thiele, [Zitzler y Thiele, 1999], proponen considerar el tamaño del área dominada en el espacio objetivo, la cual combina los 3 criterios de evaluación (distancia, distribución y extensión) en uno sólo, por lo tanto, conjuntos diferentes en más de 1 criterio pueden no ser distinguidos.

Esta métrica define el área bajo la curva definida por un conjunto de soluciones no-dominadas, permitiendo así, hacer una comparación en 2 o más conjuntos. Por ejemplo, para un problema bi-objetivo, se definirá un rectángulo limitado por el origen (0,0) y $(f_1(x), f_2(x))$. La unión de todas las áreas rectangulares determinadas por cada vector en $PF_{conocido}$ será entonces, la métrica de comparación; la cual es definida formalmente como:

$$H = \left\{ \bigcup_i a_i \mid v_i \in PF_{conocida} \right\} \quad (3.4.6)$$

En esta definición, v_i es un vector no-dominado dentro de PF y a_i es el área determinada por los componentes de v_i y el origen. Asimismo, se propone el uso de un ratio que compare el área que cada frente son capaces de cubrir. En otras palabras, se comparará el área cubierta por el $PF_{conocido}$ PF y el área obtenida mediante alguna técnica de optimización multiobjetivo.

$$HR = \frac{H_1}{H_2} = \frac{PF_{conocido}}{PF_{real}} \quad (3.4.7)$$

Luego, para el caso de minimización este ratio será 1 en caso de que $PF_{conocido} = PF_{real}$ y mayor a 1 si el área bajo la curva de $PF_{conocido}$ es mayor.

3.4.2.6. Distancia al Conjunto de Óptimos de Pareto

Autores como Rudolph, Van Veldhuizen y Lamont han considerado la distancia promedio de un conjunto dado al conjunto de óptimos de Pareto según la función M1 descrita más abajo. En esta métrica, la distribución no es tomada en cuenta, debido a que no era parte del enfoque que se quería dar; sin embargo, en estudios comparativos, se ha mostrado que la distancia por si sola no es suficiente para evaluar el rendimiento, ya que frentes distribuidos de forma extremadamente diferente pueden tener la misma distancia al Frente de Pareto.

$$M_1(X') = \frac{1}{|X'|} \sum_{a' \in X'} \min\{\|a' - \bar{a}\|; \bar{a} \in \bar{X}\} \quad (3.4.8)$$

3.4.2.7. Razón de error

En 1999, Veldhuizen [Van Veldhuizen, 1999] propone medir el porcentaje de soluciones que no están en el Frente de Pareto

$$ER = \frac{\sum_{i=1}^N e_i}{N} \quad (3.4.9)$$

Donde N es el tamaño del conjunto obtenido y e_i será 1 si la solución esta en el frente o 0 en otro caso.

Mientras mas pequeño sea ER, mejor será la convergencia al Frente. Aun así, esta métrica no funciona en el caso cuando todas las soluciones o dos conjuntos comparable no están en el Frente de Pareto. En este caso, se utilizará un umbral de distancia desde la solución i hasta el Frente; si es mayor al umbral, entonces $e_i=1$ y 0 en otro caso.

3.4.2.8. Distancia Generacional (GD)

Distancia promedio desde el conjunto obtenido hasta el Frente de Pareto. [Van Veldhuizen y Lamont, 1999]

$$GD = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{N} \quad (3.4.10)$$

La distancia euclidiana “d” en el espacio objetivo, desde la solución i hasta la solución más cercana dentro del frente de Pareto. En caso de haber una gran variación en las distancias, será necesario calcular la varianza de la métrica también. Además, los valores deben ser normalizados antes de aplicar la métrica.

Tabla 3-1: Resumen de métricas de evaluación para problemas multiobjetivo

Año	Autor(es)	Métrica	Descripción	Ecuación
1995	Schott	Esparcimiento	Distribución del conjunto obtenido.	$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - d_i)^2}$
1996	Esbensen y Kuh.	Promedio de Sumas Ponderadas	Combinación lineal dada mediante la determinación del promedio de las sumas ponderadas mínimas de todas las soluciones.	Depende de la combinación lineal escogida.
1996	Fonseca y Fleming	Superficie alcanzable	Dados dos algoritmos (A, B), determina quien vence al otro a través de líneas de intersección entre ambos conjuntos, las cuales conformarán una “superficie alcanzable”.	
1998	Hansen y Jaskiewicz	Métricas R	Dados dos conjuntos no-dominados, estas métricas determinan el número de ocasiones en que un conjunto es mejor que otro.	$R1 = \int_{u \in U} C(A, B, u) p(u) du$ $R2 = \int_{u \in U} (u^*(A) - u^*(B)) p(u) du$ $R3 = \int_{u \in U} \frac{(u^*(A) - u^*(B))}{u^*(A)} p(u) du$

Año	Autor(es)	Métrica	Descripción	Ecuación(es)
1999	Zitzler y Thiele	Cobertura de conjuntos	Proporción de cobertura entre dos conjuntos no-dominados	$H = \left\{ \bigcup_i a_i \mid v_i \in PF_{conocida} \right\}$ $HR = \frac{H_1}{H_2} = \frac{PF_{conocida}}{PF_{real}}$
1999	Rudolph	Distancia al Conjunto de Óptimos de Pareto		$M_1(X') = \frac{1}{ X' } \sum_{a' \in X'} \min \{ \ a' - \bar{a}\ ; \bar{a} \in \bar{X} \}$
1999	Veldhuizen	Razón de Error	Porcentaje de soluciones que no están en el Frente de Pareto	$ER = \frac{\sum_{i=1}^N e_i}{N}$
1999	Veldhuizen and Lamont	Distancia Generacional	Distancia promedio desde el conjunto obtenido hasta el Frente de Pareto.	$GD = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{N}$

4. Algoritmo Cultural Bi-objetivo para PSP

4.1. Formulación del Problema

Para el Problema de Selección de Portafolio se han descrito diversos enfoques de modelado, tales como los propuestos en [Loraschi, 1994], [Li et al., 2001] y [Lin y Liu, 2006]; sin embargo, para efectos de este estudio, se considerará sólo el planteamiento básico propuesto por Markowitz, en el cual los dos objetivos que entran en juego corresponden a rentabilidad y riesgo. Sean:

- N Número de acciones disponibles
- w_i Peso de la acción i -ésima en el portafolio
- μ_i Esperanza para la acción i
- σ_{ij} Covarianza entre la acción i y j

El problema bi-objetivo a resolver estará entonces sujeto sólo a la condición de que la suma de pesos del portafolio debe ser estrictamente igual a 1, lo cual asegura el uso total de la inversión.

$$\text{Función objetivo: } \quad \textit{Minimizar} \quad \sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \quad (4.1.1)$$

$$\quad \quad \quad \textit{Maximizar} \quad \sum_{i=1}^N w_i \mu_i \quad (4.1.2)$$

$$\text{Sujeto a:} \quad \sum_{i=1}^N w_i = 1 \quad (4.1.3)$$

$i = 1, 2, \dots, N$

En las secciones siguientes se describirá el modo en que estos problemas se incorporaron en las soluciones propuestas y como se diseñaron las estructuras de datos necesarias para el uso de Algoritmos Culturales.

4.2. Representación de PSP Bi-Objetivo a través de Algoritmos Culturales.

El Problema de Selección de Portafolio puede ser representado mediante un conjunto de acciones para las cuales se encuentran disponibles sus rentabilidades y riesgos asociados, y para las cuales se debe determinar su proporción de participación dentro de un portafolio dado. Para cumplir esto, se requiere que las soluciones sean representadas de modo tal que el resultado se exprese en un intervalo $[0,1]$.

4.2.1. Individuos

Cada individuo dentro del problema representa una solución candidata, -o un posible portafolio-, definida en términos de la participación de cada una de las acciones que lo conforman. Su tamaño dependerá del número de acciones (N) y será de tipo real, pues sus valores deben estar siempre en el rango $[0,1]$.

Para facilitar la activación/desactivación de acciones dentro del portafolio, se incluirán variables binarias $(0,1)$ que indicaran si el stock pertenece o no al portafolio. Asimismo, se definirán parámetros específicos para la metaheurística en uso, tales como:

Núm. de acción	1	2	3	...	N-1	N
Z	1	0	0	...	0	1
Portafolio	0.5033	0.2517	0.1674	...	0.4262	0.4967

Figura 4.1: Representación de un individuo

4.2.2. Tratamiento de individuos infactibles

En el planteamiento multiobjetivo, la única restricción presente es respecto que el peso de las acciones dentro del portafolio debe sumar exactamente 1, lo cual se puede interpretar como que se debe hacer uso de todo el capital disponible para ese portafolio.

Para cumplir con lo anterior, se aplicó una regla de normalización tal que permite mantener la proporción entre los pesos (y así no modificar tan drásticamente los pesos obtenidos luego de la actualización de los individuos) y asimismo, que la suma de todos ellos sean exactamente 1.

Luego, el nuevo peso que tendrá el individuo estará dado por:

$$w_j = \frac{w_j}{\sum_{i=1}^N w_i} \quad j \in \{1,2,3\dots N\} \quad (4.1.4)$$

4.2.3. Población

- a. **Estructura:** Se generará con un tamaño de “POBL” individuos, parámetro que será definido previo a la ejecución del algoritmo. En un comienzo la población se generará de modo aleatorio.

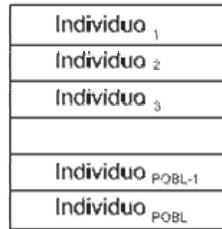


Figura 4.2: Representación de la población

- b. **Actualización:** En un algoritmo cultural tradicional, la actualización de la población se hubiera hecho en base a los operadores genéticos de selección, recombinación y mutación. Sin embargo, dadas las características multiobjetivo del problema, -y por lo tanto, no hay medición del individuo a través de su fitness-, se optara por actualizar la población a través de la adición de valores distribuidos uniformemente

4.2.4. Espacio de Creencias

- a. **Estructura:** El espacio de creencias estará formado por una parte normativa, que almacenará las cotas por cada una de las ‘k’ funciones objetivo (en vez de hacerlo por la cantidad de variables de decisión, que es lo usual en un caso mono-objetivo)



Figura 4.3: Parte normativa multiobjetivo.

Dada la naturaleza multiobjetivo del problema, se incorporará una estructura de datos tipo matriz que permita evaluar la distribución de las soluciones no-dominadas a lo largo del Frente de Pareto. Ésta se construirá con las cotas del espacio normativo, y se subdividirá en ‘s’ celdas de igual dimensión, las cuales almacenarán la cantidad de soluciones no-dominadas presentes en cada sub-división del Frente de Pareto, ayudando así a detectar situaciones de “aglomeración” de soluciones en torno a un

sector. A modo de ejemplo, en la siguiente figura se muestra una rejilla con $k = 2$ y $s = 8$.

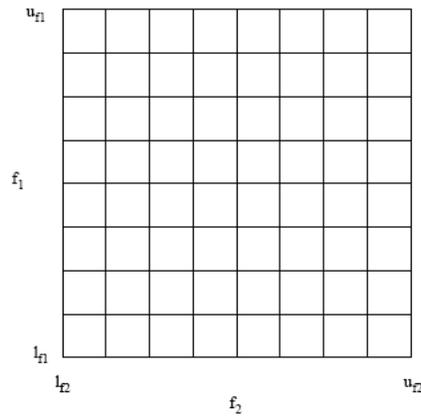


Figura 4.4: Rejilla para dos funciones objetivo con ocho intervalos.

- b. Inicialización:** Ocurrirá sólo una vez que la población inicial ha sido generada, ya que se utilizarán las soluciones no-dominadas allí encontradas (en [Van Veldhuizen, 1999] se demuestra que en toda población es posible encontrar al menos 1 solución no-dominada.) para determinar los límites inferior y superior de cada función objetivo. Asimismo, la rejilla iniciará todos sus contadores a cero, ya que ninguna solución ha sido añadida aún al archivo.
- c. Actualización:** La rejilla para contabilizar soluciones no-dominadas se actualizará en cada generación; sin embargo, la parte normativa que contiene los límites de la rejilla (y de cada función objetivo presente en el problema), se actualizará cada 'Gnorm' generaciones, debido a lo costoso de la operación.

Luego, bastará con actualizar los contadores durante cada generación, considerando que las cotas l y u no varían seguidamente. Sin embargo, la actualización de la parte normativa será mucho más compleja, debido a que se actualizarán las cotas inferior y superior de cada función, el tamaño de cada celda, y por lo tanto, la rejilla de modo completo.

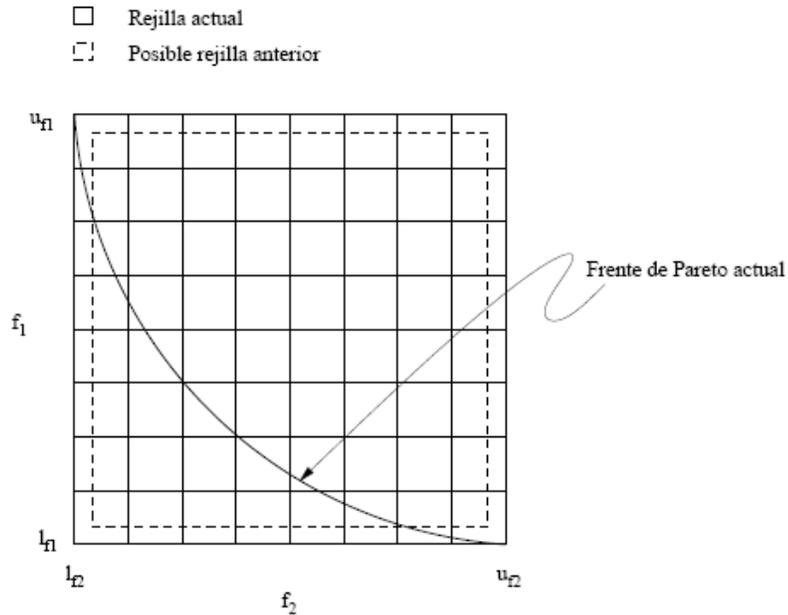


Figura 4.5: Actualización de la rejilla

4.2.5. Influencia del Espacio de Creencias en la Población

Al momento de generarse una nueva población, el Espacio de Creencias tratará de guiar a las nuevas soluciones hacia el óptimo más rápidamente. Dado que la población utilizada es la misma que la de un Algoritmo Genético, será en uno de sus operadores donde se manifestará esta influencia. En este caso, se aplicará específicamente al operador de Selección.

Landa [Landa, 2002] propone utilizar un torneo de selección aplicando las siguientes reglas:

1. Si un individuo domina al otro, gana el individuo no dominado.
2. Si no son comparables, o sus valores de las funciones objetivo son iguales, entonces:
 - a. Si ambos están dentro de la rejilla del espacio de creencias, gana el que se encuentre en una celda menos poblada (según el contador de las celdas).
 - b. Si alguno cae fuera de la rejilla, gana el que esté fuera.

4.2.6. Almacenamiento de las soluciones no-dominadas

El archivo se usará para almacenar sólo soluciones no-dominadas, no repetidas y encontradas durante toda la ejecución del algoritmo. En su tesis magistral, Landa [Landa, 2002] propone las siguientes reglas para añadir nuevas soluciones al archivo:

1. Si el individuo que se pretende agregar es dominado por algún individuo del archivo, entonces el individuo no se debe agregar.

2. Si el individuo que se pretende agregar domina a algún individuo del archivo, entonces se introduce en su lugar, pero continúa comparándose contra todos los demás. Si el mismo individuo, ya agregado, dominará a algún otro, éste (el individuo dominado) es eliminado del archivo.
3. Si el individuo que se pretende agregar no es dominado ni domina a ningún otro en el archivo, y el tamaño actual de este archivo es menor que su tamaño máximo (q), entonces se puede agregar el nuevo individuo.
4. Si el individuo que se pretende agregar no es dominado ni domina a ningún otro en el archivo, y el tamaño actual de este archivo es su tamaño máximo q , entonces se busca algún individuo del archivo cuya celda contenga más individuos que la celda a la que pertenece el individuo que se pretende agregar, y se reemplaza el individuo anterior con el nuevo. Con esto se obtiene una mejor distribución de los individuos no dominados entre las celdas.

Aunque el archivo esté lleno, su contenido seguirá cambiando durante la ejecución para obtener una mejor distribución. Al final, este archivo tendrá el frente de Pareto que se presentará al tomador de decisiones, quien elegirá una de las soluciones para utilizarla en el problema que se quiere resolver.

4.2.7. Parámetros de la solución.

Finalmente, los parámetros que deben considerarse para implementar una solución para el Problema de Selección de Portafolio con Algoritmos Culturales multiobjetivo son:

Tabla 4-1: Parámetros de la solución propuesta

Ámbito	Parámetro	Descripción	Valor
Problema de Selección de Portafolio	N	Número de variables de decisión	Depende del problema
	K	Número de acciones a incluir en el portafolio.	Depende del inversionista, pero para efectos prácticos, se definirá en el ajuste de parámetros
Algoritmo Cultural	POBL	Tamaño de la población	A determinar en el ajuste de parámetros
	g	Cantidad de generaciones	A determinar en el ajuste de parámetros.
	pmut	Probabilidad de mutación	A determinar en el ajuste de parámetros.
	Gnorm	Número de generaciones que deben pasar antes de actualizar el conocimiento normativo.	5

	aceptados	Porcentaje de individuos que actualizarán el Espacio de Creencias	50% de la población
--	-----------	---	---------------------

El ajuste de estos parámetros se verá en secciones posteriores. Sin embargo, para los casos de ‘Gnorm’ y ‘aceptados’ se consideraron los valores propuestos por Landa en su tesis magistral [Landa, 2002].

4.3. Descripción del Algoritmo

El algoritmo comienza generando de modo aleatorio los pesos de la población de individuos, de modo de obtener un portafolio preliminar; luego, se evalúa la población, se genera el espacio de creencias y se da comienzo a la evolución (representada por medio de un bucle condicional). En este punto, los hijos serán generados sólo a través de la mutación de sus padres (dado que no es posible utilizar el fitness como criterio de evaluación). Posteriormente se evaluará a cada hijo y se llevará a cabo un torneo entre toda la población, el cual dará como resultado el decidir a qué individuos seleccionar para la próxima generación y a cuales descartar. Una vez obtenida la nueva generación, se actualizará el espacio de creencias en base al conocimiento adquirido y los individuos no-dominados que se encuentren ahí serán añadidos al archivo externo. Finalmente, el ciclo evolutivo se detendrá una vez alcanzado el número de generaciones indicado por el usuario.

Es necesario mencionar que en el diseño de la solución multiobjetivo las restricciones se separarán de la función objetivo y pasarán a formar parte del Espacio de Creencias del algoritmo, tal como postula CAEP.

5. Ajuste de Parámetros y Resultados

5.1. Condiciones generales

- a. **Instancias de Prueba:** El ajuste de parámetros se ha hecho en base a las instancias de la librería digital OR-Library (<http://people.brunel.ac.uk/~mastjib/jeb/info.html>). En ella, se describen cinco casos de prueba tomados desde mercados accionarios reales. A modo general, las instancias son [Beasley, 1990]:

Tabla 5-1: Instancias tomadas desde OR-Library

Instancia	Origen	Nº de acciones
port1	Hong Kong	31
port2	Alemania	85
port3	Reino Unido	89
port4	Estados Unidos	98
port5	Japón	225

- b. **Instancias de Validación:** El trabajo a utilizar para la validación de la propuesta multiobjetivo es “Portfolio Optimization with an Envelope-based Multi-objective Evolutionary Algorithm” [Branke, 2009]
- c. **Hardware:** Para todas las pruebas, se ha utilizado una máquina con las siguientes características:
- Procesador: Intel ® Core™ Duo T2350 1.86 GHz
 - RAM: 2 GB
 - Sistema Operativo: Windows XP SP 2
- d. **Lenguaje de Programación:** El código ha sido escrito completamente en ANSI C.

5.2. Ajuste de Parámetros

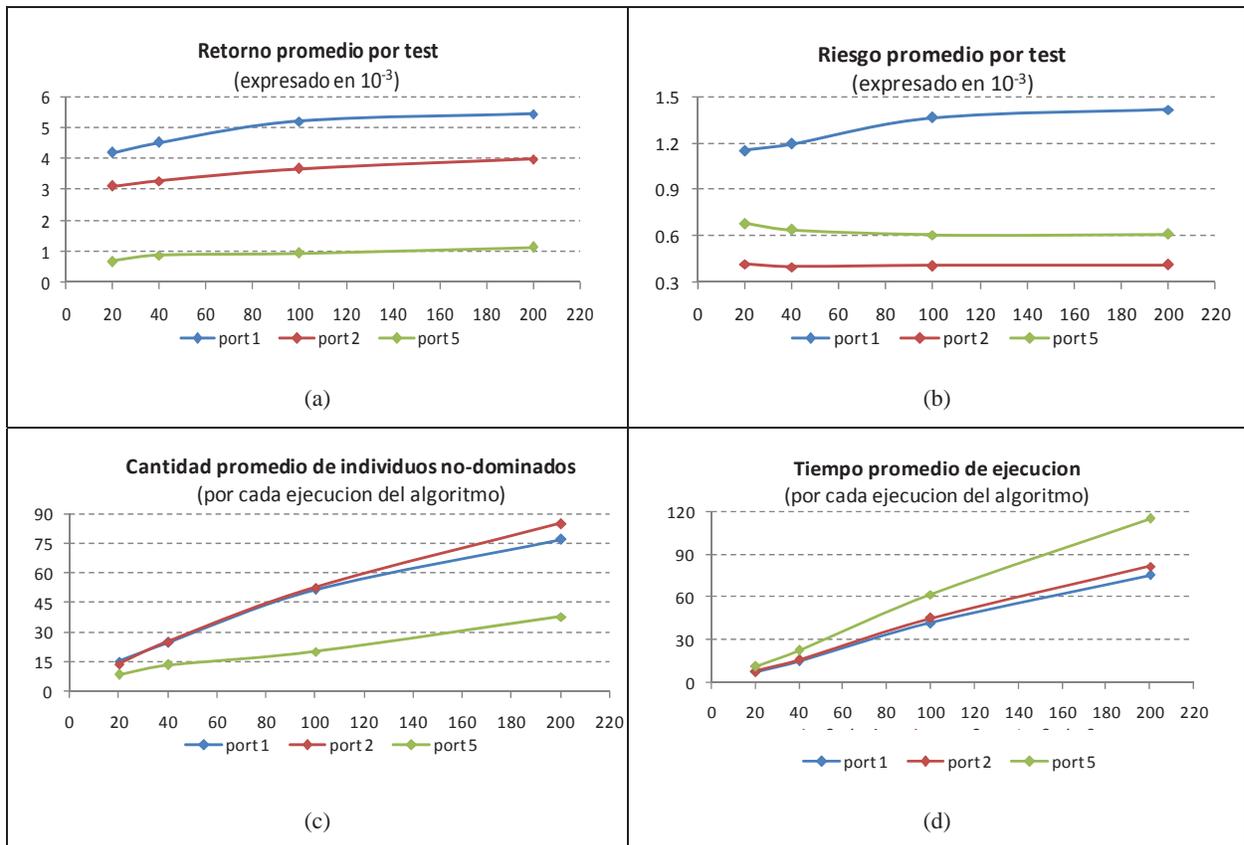
5.2.1. Número de individuos

Para ajustar este parámetro se evaluarán 3 instancias de prueba (Hong Kong, Alemania y Japón) las cuales se seleccionaron considerando la cantidad de acciones en cada índice; así, se podrá evaluar la respuesta del algoritmo ante distintas cantidades de acciones.

Asimismo, para ver con claridad la influencia del número de individuos en la solución final, el resto de los parámetros del algoritmo serán constantes durante toda la prueba. Los valores que se usarán son:

- Número de generaciones: 100
- Probabilidad de mutación: 0.2
- K (acciones a considerar): 50% (depende de la instancia de prueba)

Los resultados, -sobre pruebas de más de 100 ejecuciones por cada vez-, son:



En el caso de (a) se observa que, efectivamente, el número de individuos mejora la rentabilidad de los portafolios obtenidos o, mejor dicho, a mayor número de individuos el algoritmo logra explorar de mejor forma el espacio. Asimismo, si la rentabilidad aumenta directamente según el número de individuos es esperable que el riesgo del portafolio también

aumente, lo cual se puede confirmar en (b). En el caso de (c) se observa que mientras más individuos exploran el espacio, es posible encontrar mas soluciones no-dominadas, lo cual es bueno desde el punto de vista del inversor, puesto que existirán más alternativas disponibles. Ahora bien, no se puede dejar de lado el hecho de que a mayor número de individuos, también son mayores los recursos gastados en su procesamiento (gráfico (d)).

Tomando en cuenta los resultados obtenidos en esta prueba, se puede concluir que la cantidad de individuos:

- Es efectivamente relevante para el número de soluciones obtenidas en cada ejecución, ya que los conjuntos más grandes de no-dominados (“frente de Pareto”) se obtuvieron con las poblaciones más grandes (200 individuos)
- Como era de esperar, a mayor número de individuos, mayor tiempo de ejecución.

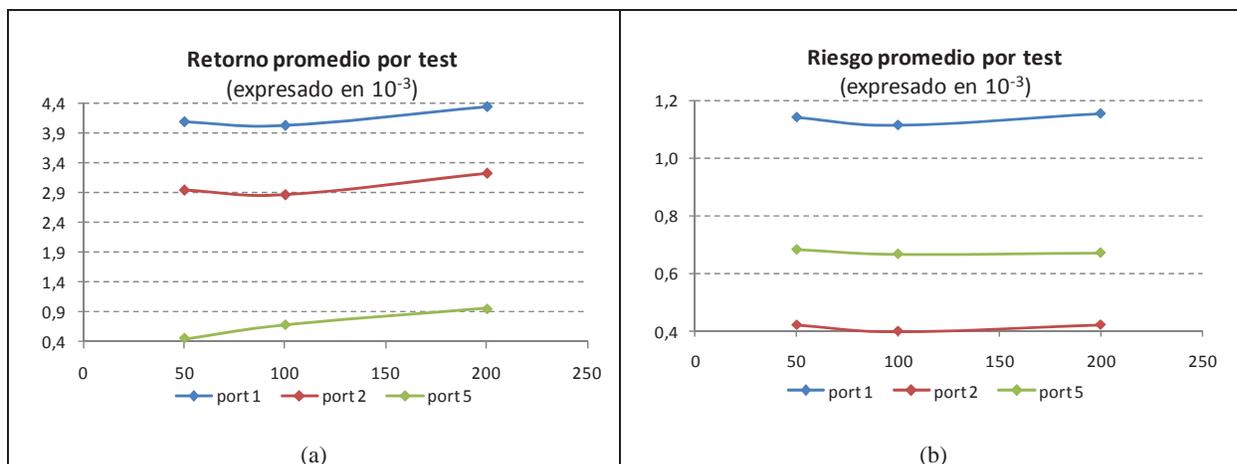
5.2.2. Número de generaciones

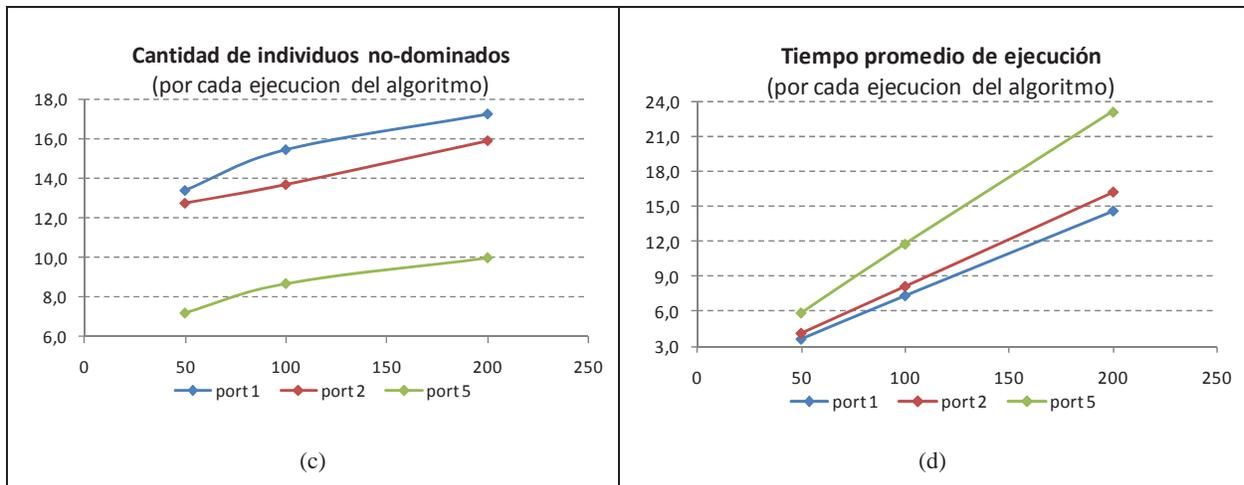
Para el ajuste de este parámetro se evaluarán 3 instancias de prueba (Hong Kong, Alemania y Japón) con un número de individuos, probabilidad de mutación y número de acciones a incluir constantes. Asimismo, el número de generaciones se moverá en los niveles de 50, 100 y 200.

Además, acá se ocupará un numero bajo de individuos (sólo 20) para medir de modo más preciso si el aumento en el número de generaciones provoca un cambio significativo. Luego, los valores de los parámetros que quedarán fijos serán:

- Tamaño de la población: 20 individuos
- Probabilidad de mutación: 0.2
- K (acciones a considerar): 50% (depende de la instancia de prueba)

Los resultados, -sobre pruebas de más de 100 ejecuciones por cada vez-, son:





Similar a lo que ocurre cuando se aumenta el número de individuos, el incrementar las generaciones mejora las capacidades de búsqueda del algoritmo, como se aprecia en (a) y (b), y también de exploración, ya que mientras mas generaciones, más soluciones no-dominadas es posible de encontrar en una ejecución del algoritmo, como se muestra en (c). Asimismo, -y como era de esperar, el este parámetro impacta directamente en el tiempo de ejecución, el cual aumenta en la misma proporción que el numero de generaciones (d).

5.2.3. Probabilidad de mutación

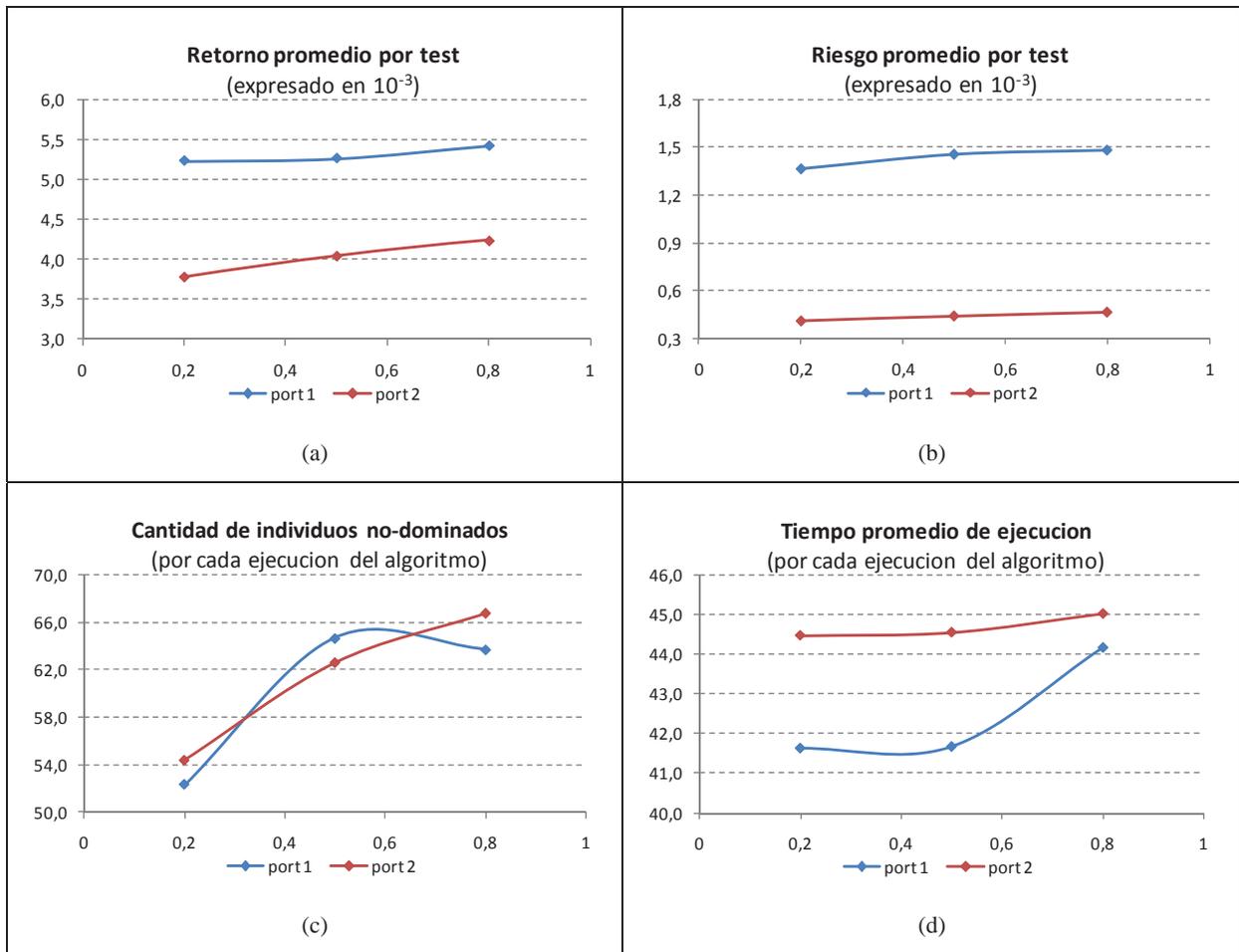
Para el ajuste de este parámetro se evaluarán 2 instancias de prueba (portafolios 1 y 2) con probabilidades de mutación de: 0.2, 0.5 y 0.8. Los valores de los parámetros a fijar son:

- Tamaño de la población: 100 individuos
- Número de generaciones: 100
- K (acciones a considerar): 50% (depende de la instancia de prueba)

El objetivo de esto, es determinar cómo este parámetro influye en el conjunto de soluciones obtenido, en términos de:

- a. **Cantidad:** Al haber una mayor probabilidad de mutación (y por tanto, un mayor nivel de exploración) ¿es posible obtener una mayor cantidad de soluciones?
- b. **Cobertura:** ¿Se mejora la distribución de las soluciones?

Los resultados, -sobre pruebas de más de 100 ejecuciones por cada vez-, son:



La variación en la probabilidad de mutación, pareciera no tener mayores efectos en los valores alcanzados por cada uno de los objetivos por separado, tal como se aprecia en los gráficos (a) y (b). Sin embargo, en cuanto a la cantidad promedio de individuos no-dominados encontrados por ejecución, el gráfico (c) muestra que incrementar la probabilidad de mutación (dicho en otras palabras, la posibilidad de variación), dota al algoritmo de mejores capacidades exploradoras. Ahora bien, el tiempo promedio de ejecución (expresado en segundos), no sufre mayores alteraciones, siendo 3 segundos la diferencia máxima alcanzada entre un 'pmut' de 0,2 y uno de 0,8.

5.2.4. Número de acciones que conforman el portafolio

Finalmente, es importante revisar cómo el incluir (o no) acciones dentro del portafolio puede afectar la calidad del conjunto de soluciones obtenidas. Esto, ya que cada vez que incluimos una nueva acción, se podría estar mejorando el retorno (dependiendo de la rentabilidad de la acción incluida), pero a su vez, se estaría incorporando un nuevo elemento de riesgo. A priori, esto podría indicar que un número bajo de acciones sería mejor que uno alto.

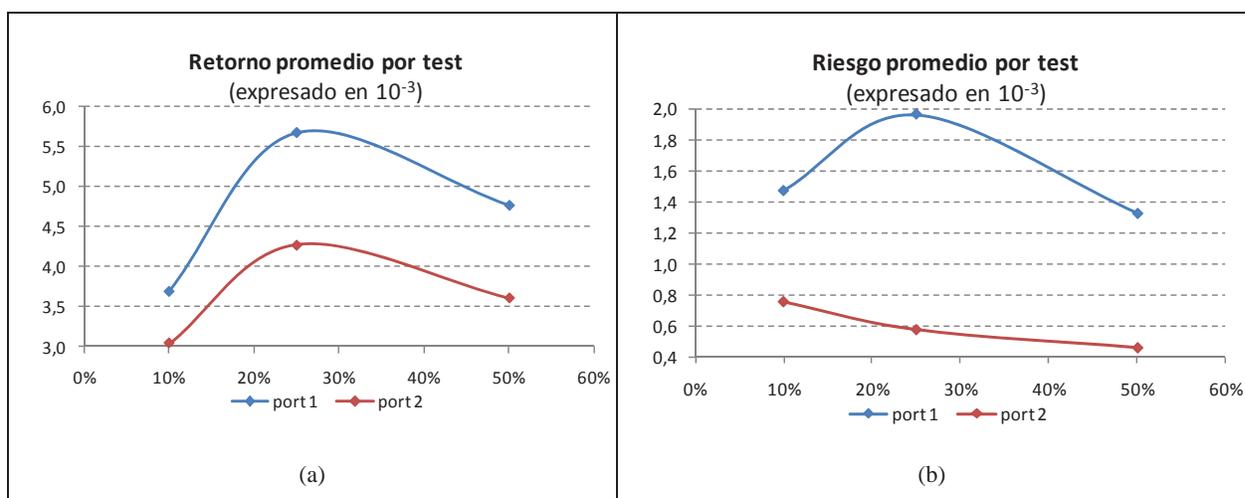
Para verificar lo anterior, se evaluarán 3 instancias de prueba (Hong Kong, Alemania y Japón) con un número de individuos, probabilidad de mutación y cantidad de generaciones constantes. Asimismo, el parámetro K variará en 10%, 25%, 50% (el número de acciones variara según el tamaño de los instancias usadas).

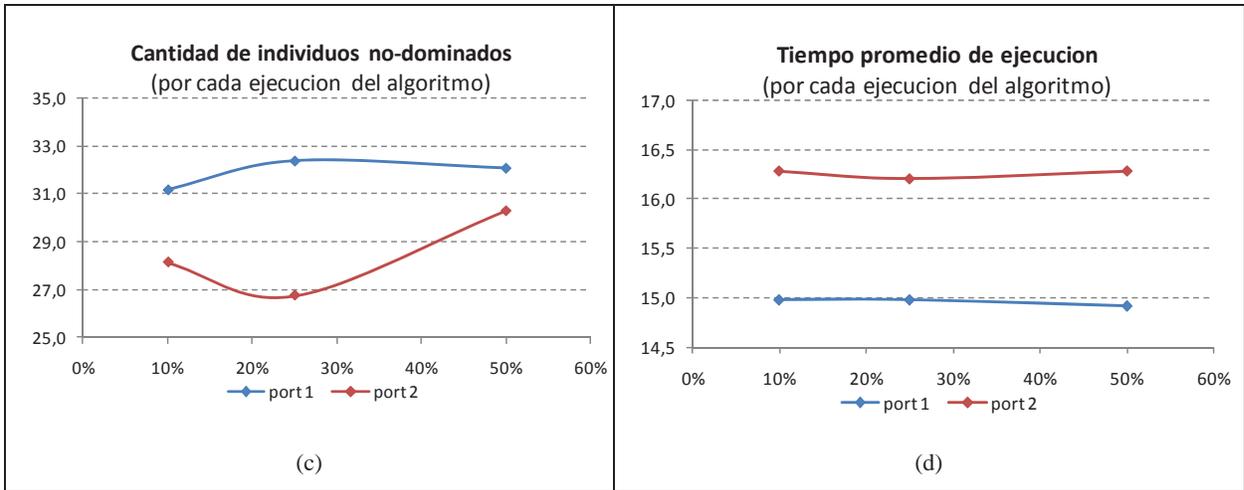
	Num. Acciones	10%	25%	50%
port 1	31	4	8	16
port 2	85	9	22	43
port 5	225	23	57	113

Los valores de los parámetros constantes serán:

- Tamaño de la población: 40 individuos
- Número de generaciones: 100
- Probabilidad de mutación: 0.5

Los resultados, -sobre pruebas de sobre 300 ejecuciones por cada vez-, son:





El caso del parámetro 'K' se observa claramente que tiene una gran influencia en cuanto a los valores de las soluciones encontradas (gráficos a y b), pero no así en los aspectos técnicos de su ejecución (gráficos c y d). Respecto a los valores de las soluciones no-dominadas, se puede apreciar que tener un número muy bajo de acciones a elegir conlleva que el portafolio en si no rinda demasiado, ya que hay poca diversificación; sin embargo, al tener un número alto de acciones también es perjudicial porque aumenta el riesgo combinado.

5.3. Resultados

Luego de haber estudiado el comportamiento de los parámetros del algoritmo, es necesario determinar la configuración final que se le darán, para así obtener el mejor resultado posible. En este caso, esa configuración se establecerá en:

- Tamaño de la población : 40 individuos
- Número de generaciones : 100
- Probabilidad de mutación : 60%
- K (acciones a considerar) : 4 ú 8, según portafolio a evaluar.

Para la validación de la solución propuesta (PSP-CA), los resultados obtenidos con la configuración final del algoritmo se compararán con el trabajo del profesor Jürgen Branke [Branke, 2009], el cual resuelve el mismo problema de selección de portafolio que acá se presenta, pero utilizando un algoritmo evolutivo basado en envolturas (“envelope-based”); sin embargo, antes de dar a conocer los resultados, es necesario tomar en cuenta las siguientes consideraciones:

1. El parámetro K sólo podrá tomar los valores 4 y 8.
2. Los portafolios se nombrarán como P1, P2, P3, P4 y P5 respectivamente para mantener la notación entregada en OR-Lib. En el caso del trabajo de Branke, el portafolio #5 fue nombrado como “P3”, mientras que los portafolios #3 y #4 no se utilizaron.
3. Luego, los portafolios que se utilizarán para la comparación serán:
 - a. P1: Índice Hang Seng, 31 acciones, K=4
 - b. P2: Índice S&P, 98 acciones, K=4
 - c. P5: Índice Nikkei, 225 acciones, K=8

Siguiendo la misma estructura que Branke utilizó en su capítulo de resultados, primero se presentarán 2 pares de gráficos que muestran el acercamiento de las soluciones obtenidas respecto del Frente de Pareto ideal para los portafolios P1 y P5, utilizando la restricción de cardinalidad. Sólo para efectos de visualización, tanto los gráficos de P-MOEA y E-MOEA como los de PSP-CA serán mostrados en la misma escala y respetando lo más posible el formato utilizado por Branke.

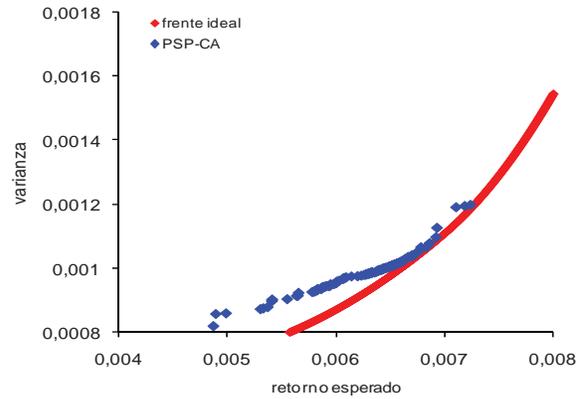
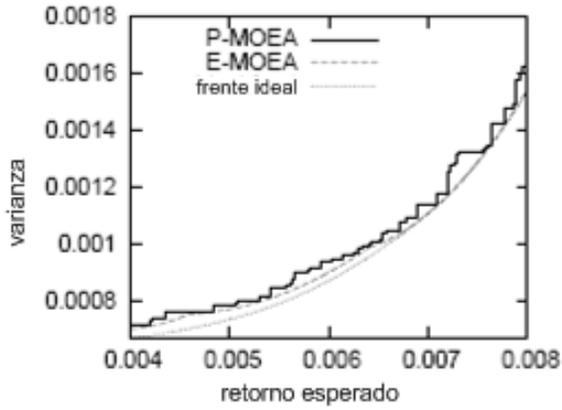


Figura 5.1: Comparación de resultados para P1 (Hang Seng), con K=4

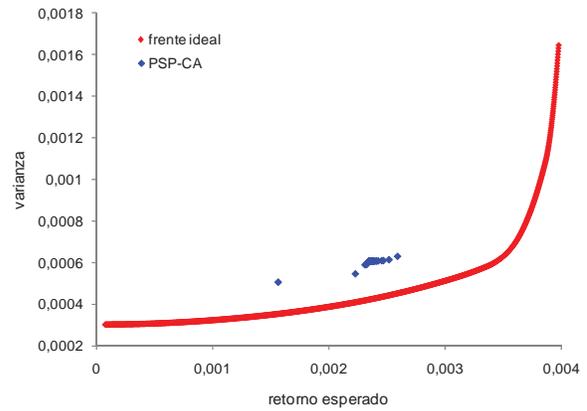
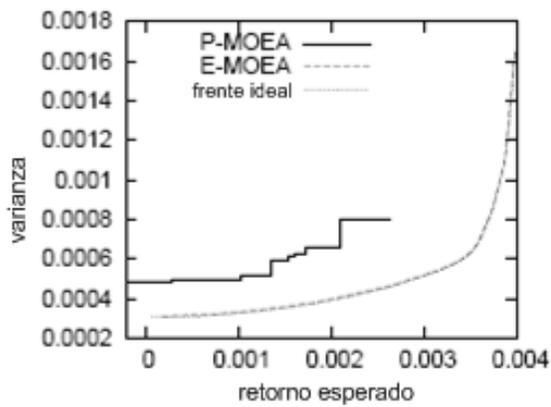


Figura 5.2: Comparación de resultados para P5 (Nikkei), con K=8

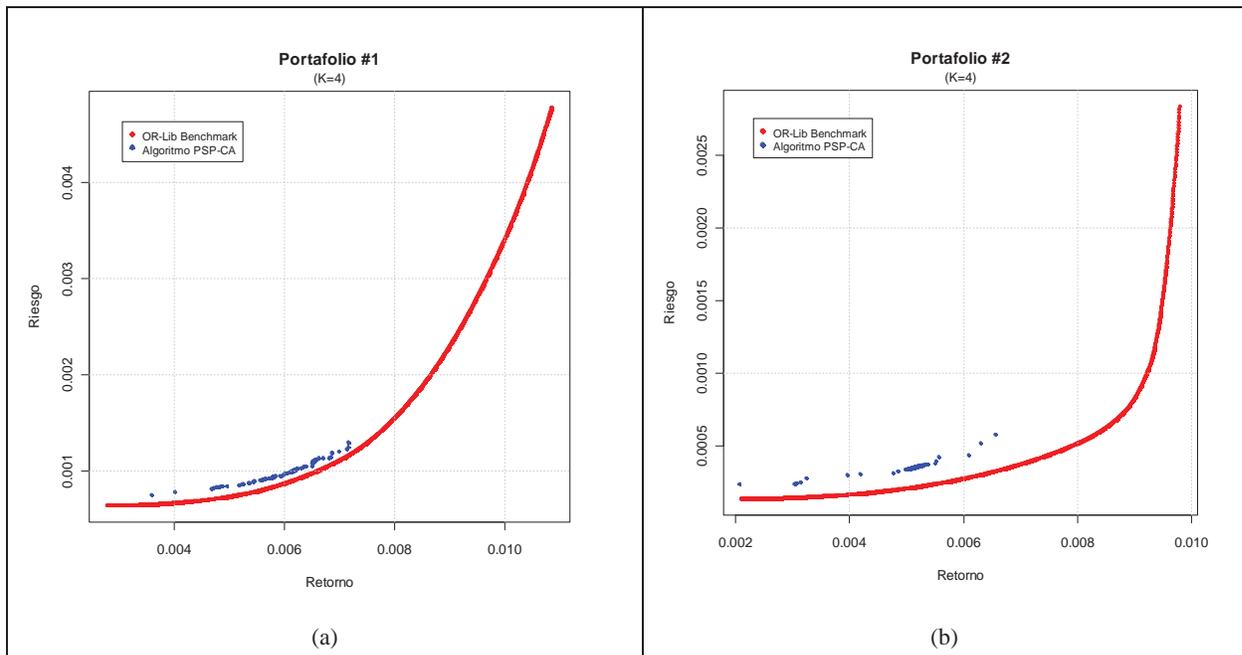
Ahora bien, para tener una mejor cuantificación de los resultados obtenidos, las fronteras obtenidas serán evaluados del siguiente modo: en cada caso, se presentará la diferencia entre el “área ideal” delimitada por el frente ideal (de OR-Lib) y el área cubierta por el frente obtenido a través de PSP-CA. Cabe mencionar que este mismo método fue el utilizado por Branke para sus propios resultados.

Cada resultado presentado considera la media y la desviación estándar (expresadas en términos de 10^{-6}) de las diferencias obtenidas entre los frentes calculados y los frentes ideales, luego de 30 ejecuciones.

Tabla 5-2: Diferencias entre los frentes calculados y las fronteras ideales

	P1 (K=4)	P2 (K=4)	P5 (K=8)
P-MOEA	1.1613 ± 0.0159	2.7787 ± 0.0521	9.3292 ± 0.2287
E-MOEA	0.2275 ± 0	0.8048 ± 0.00003	0.0561 ± 0.0052
PSP-CA	2.4514 ± 0.0126	10.998 ± 0.0034	4.5547 ± 0.0087

Finalmente, los resultados generales para cada uno de los portafolios de la Librería-OR muestran que el algoritmo tiene un buen acercamiento al frente ideal correspondiente, a pesar de la restricción de cardinalidad impuesta, la cual se manifiesta en la falta de extensión de los conjuntos no-dominados encontrados (recordar que el “frente ideal” corresponde al conjunto de soluciones encontrado mediante técnicas exactas para el problema de selección de portafolio sin restricciones y que por lo tanto es considerado el frente máximo posible de alcanzar).



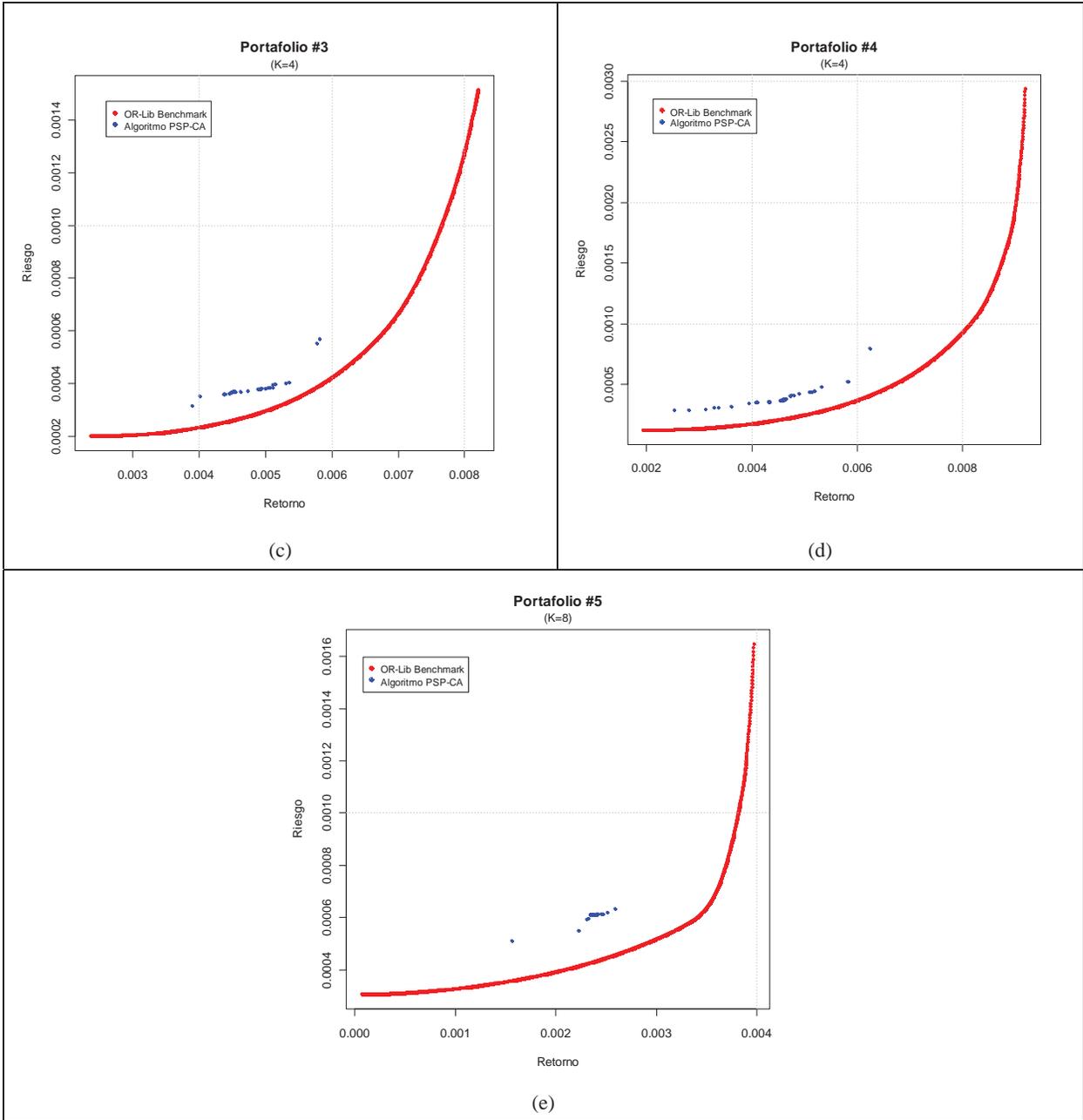


Figura 5.3: Frentes de Pareto obtenidos para cada una de las instancias de la Librería-OR

6. Conclusiones y Trabajo Futuro

El presente trabajo presenta una nueva forma de abordar el Problema de Selección de Portafolio, formulado a través del modelo Media-Varianza de Markowitz y resuelto con un Algoritmo Cultural con enfoque Multiobjetivo. La validación de dicho método se hizo a través de la comparación de los resultados obtenidos con el trabajo de Jürgen Branke [Branke, 2009], en el cual utiliza un algoritmo evolutivo basado en “envolturas”.

A modo de resumen, en este informe se presentaron desde el estado del arte para los temas de Problema de Selección de Portafolio, Optimización y Algoritmos Culturales; hasta las pruebas de sensibilización de parámetros para el algoritmo multiobjetivo a utilizar. Todo ello para dar paso a la comparación que valida el trabajo aquí expuesto.

6.1. Conclusión

Si bien el Problema de Selección de Portafolio ha sido ampliamente estudiado, formulado y resuelto, lo cierto es que la mayoría de las soluciones son de tipo monobjetivo, ya que el gran inconveniente radica en las múltiples combinaciones de pesos de que se pueden dar en un mismo portafolio. Sin embargo, es posible aplicar una metaheurística que evite el problema de la explosión combinatorial y que –junto a un enfoque multiobjetivo–, pueda ser capaz de satisfacer la mejor forma posible los dos objetivos simultáneamente: minimizar riesgo y maximizar rentabilidades.

En cuanto a la aplicación de Algoritmos Culturales, se puede decir que esta técnica cuenta con ventajas respecto de otras metaheurísticas. La sola inclusión del espacio de creencias, el cual permite tener un marco de referencia para las generaciones posteriores, implica una mayor convergencia del algoritmo y por ende, menor tiempo de ejecución, a pesar de que esto no pueda ser muy claro en una implementación multiobjetivo, donde la respuesta esperada no es una única solución, sino un conjunto de individuos no-dominados y donde ya no existe el concepto de fitness (clave para evaluar cuán buena es una solución respecto de otra), si no que acá se tiene la “dominancia” (un individuo/solución A sólo será mejor a uno B si todas las evaluaciones de los objetivos son mejores en A que en B).

Respecto a los resultados obtenidos, si bien estos son absolutamente factibles y prácticos desde el punto de vista de lo mínimo de las diferencias obtenidas, aun así no logran superar a una técnica mixta como la utilizada en [Branke, 2009]; sin embargo, si consideramos la aplicación práctica de la propuesta PSP-CA, los resultados obtenidos mejoran por mucho a los que se pueden obtener actualmente con elementos menos dinámicos como podría ser el uso de una planilla de cálculo, la cual probablemente se vería seriamente afectada al momento de determinar un conjunto de portafolios como los aquí presentados.

6.2. Trabajo Futuro

Finalmente el trabajo futuro de este proyecto puede verse tanto del ámbito computacional como financiero, teniendo muchas posibilidades en ambos casos. En este sentido, se proponen las siguientes extensiones:

- *Uso de nuevo objetivos en la formulación del problema:* Tal como se menciona en el trabajo de Erghott [Ehrgott et al., 2004], se pueden utilizar otros objetivos en el problema de optimización, los que (sin embargo) subyacen en la misma idea original de Markowitz.
- *Incorporación de nuevas restricciones:* Por ejemplo, se pueden evaluar las restricciones de costos propuestas por DiTollo [DiTollo y Rolli, 2006], o las restricciones de pesos 5-10-40 utilizadas en una parte del trabajo de Branke
- *Evaluación de la técnica multiobjetivo utilizada:* En este trabajo, se optó por usar una rejilla como forma última de evaluar las soluciones obtenidas en cada generación, lo cual intentaba fomentar el esparcimiento de las soluciones a lo largo del frente. Sin embargo, sería interesante observar si un cambio en esta estrategia (por ejemplo, utilizar una combinación lineal de objetivos) podría resolver más fácil o con mejores resultados el problema.
- *Comparación con otras metaheurísticas:* Si bien este trabajo ya se compara con otro algoritmo evolutivo, también sería posible evaluar otras técnicas tales como PSO, o ACO.
- *Considerar un procesamiento paralelo que aumente la velocidad de ejecución:* Como en todo algoritmo de optimización, el problema del tiempo de ejecución no puede ser dejado de lado y también se podría evaluar si este tipo de procesamiento es capaz (o no) de reducir significativamente los tiempos de ejecución.

7. Bibliografía

[Beasley, 1990] Beasley J.E., *OR-Library: Distributing test problems by electronic email*, Journal of the Operational Research Society, 41(11): 1069-1072, 1990.

[Branke, 2009] Branke J., Scheckenbach B., Stein M., Deb K., Schmeck H., *Portfolio optimization with an Envelope-based multi-objective evolutionary algorithm*, Journal European Research 199(2009), pp 684-693.

[Chang, 1998] Chang T., Meade N., Beasley J. B. and Sharaiha Y. (2000), *Heuristics for cardinality constrained portfolio optimisation*. Computers & Operations Research, 27:1271–1302.

[Chow, 1996] Chow G.C., *The Lagrange method of optimization with applications to portfolio and investment decisions*, Journal of Economic Dynamics and Control, 20 (1-3), pp. 1-18.

[Coello, 1999] Coello, C., *A Survey of constraint handling techniques used with evolutionary algorithms*, Technical Report, Lania-RI-99-04, Laboratorio de Informática Avanzada, Veracruz, México, 1999, <http://www.cs.cinvestav.mx/constraint/>

[Coello, 2002] Coello C., *Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art*, Computer Methods in Applied Mechanics and Engineering, 191(11-12):1245–1287, January 2002.

[Coello y Landa, 2003] Coello, C., Landa R. *Evolutionary Multiobjective Optimization using a Cultural Algorithm*, Swarm Intelligence Symposium, 2003. SIS '03, Proceedings of the 2003 IEEE. 2003

[Crama y Schyns, 2003] Crama, Y., Schyns M., *Simulated annealing for complex portfolio selection problems*. European Journal of Operational Research, 150(3), 2003, pp. 546–571.

[Czyzyk et al., 1999] Czyzyk, J., Wisniewski, T., Wright, S., *Optimization case studies in the NEOS Guide*, SIAM Review, 1999, Vol. 41, pp. 148–163.

[Deb, 2000] Deb K., *An efficient constraint-handling method for genetic algorithms*, Computer Methods Applied Mech. Eng., vol. 186, no. 2–4, pp. 311–338, 2000.

[DiTollo y Rolli, 2006] DiTollo G., Rolli A., *The Portfolio Selection Problem: Opportunities for constrained based metaheuristics*. Proceedings of CP2006 Doctoral Program (Nantes (FR), 24-29.09.2006)

[Ehrgott et al., 2004] Ehrgott M., Klamroth K., Schwehm C. *An MCDM approach to portfolio optimization*, European Journal of Operational Research, 155(3):752{770, 2004.

[Ehrgott, 2006] Ehrgott, M., *Multiobjective (Combinatorial) Optimization – Some Thoughts on Applications*, School of Engineering Reports, 2006, <http://researchspace.auckland.ac.nz/bitstream/2292/2665/1/esc-tr-640.pdf>

[Farhang-Mehr y Azarm, 2003] Farhang-Mehr A., Azarm S., *Minimal sets of quality metrics*, Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO 2003), Lecture notes in computer science, 2632, Springer, 405-417, (2003).

[Fernandez, 2005] Fernandez A., Gomez S., *Portfolio Selection using Neural Networks*, Computers and Operations Research, 2005.

[Fonseca y Fleming, 1996] Fonseca, C.M., Fleming, P. J. *On the performance assessment and comparison of stochastic multiobjective optimizers*. Fourth International Conference on Parallel Problem Solving from Nature (PPSN-IV), pages 584–593, Springer, Berlin, Germany.

[Knowles y Corne, 2002] Knowles J. D., Corne D. W., *On metrics for comparing non-dominated sets*, Proc. of 2002 Congress on Evolutionary Computation, pp. 711-716, Honolulu, May 12-17, 2002.

[Lagos, 2006] Lagos C, *Algoritmos culturales aplicados a la optimización con restricciones*, Proyecto final para optar al título de Ingeniero Civil Informático., 2006

[Landa, 2002] Landa R., *Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo*. Tesis Magistral, Instituto Politécnico Nacional de México, 2002.

[Li et al., 2001] Li X., Zhou X.Y., Lim A.E.B., *Dynamic mean–variance portfolio selection with no-shorting constraints*, SIAM Journal on Control and Optimization. 40 (2001) 1540-1555

[Lin et al, 2001] Lin D, Wang S., Yan H., *A Multiobjective Genetic Algorithm for Portfolio Selection*, Technical report, institute of Systems Science, Academy of Mathematics and Systems Science Chinese Academy of Sciences, Beijing, China, 2001.

[Lin y Liu, 2006] Lin, C.C. and Liu, Y.T., *Genetic algorithms for portfolio selection problems with minimum transaction lots*, European Journal of Operational Research. V185. 393-404, 2006

[Loraschi, 1994] Loraschi A, *An Evolutionary Algorithm For Portfolio Selection in a Downside Risk Framework*, European Journal of Finance, 1994

[Markowitz, 1997] Markowitz H., *Foundations of Portfolio Theory*, Journal of Finance, Volume 46, Issue 2, pages 469-477, Junio 1997

[Mitchell, 1978] Mitchell T.M., *Version Spaces: An Approach to Concept Learning*, Ph.D. dissertation, Electrical Engineering Department, Stanford University, December, 1978. Also Stanford Computer Science Department Technical Report STAN-CS-78-711.

[Reynolds, 1994] Reynolds, R., *Introduction to Cultural Algorithms*, Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel, Editors, World Scientific Press, Singapore, 1994, pp.131-139

[Reynolds y Nazzal, 1997] Reynolds, R., Nazzal, A., “*Using Cultural Algorithms with Evolutionary Computing to Extract Site location Decisions From Spatio-Temporal Databases*”, in Evolutionary Programming VI, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Editors, Springer-Verlag Press, New York, NY, 1997, pp. 323-334.

[Reynolds y Rychtyckyj, 2002] Reynolds, R., Rychtyckyj, N., *Knowledge Base Maintenance Using Cultural Algorithms: Application to the DLMS Manufacturing Process System at Ford Motor Company*, Proceedings of World Congress on Computational Intelligence, May 12-19, 2002, Honolulu, Hawaii, Volume 1, Issue , 12-17 May 2002, pp 855 - 860

[Reynolds et al., 2009] Reynolds R., Kobti Z., Kohler T., *Agent-Based Modeling of Cultural Change in Swarm Using Cultural Algorithms*, Proceedings of SWARMFEST 2004, University of Michigan, Ann Arbor, May 9-11, 2004.

[Rychtyckyj y Reynolds, 1999] Rychtyckyj N., Reynolds R. *Using Cultural Algorithms to Improve Performance in Semantic Networks*, Proceedings of the 1999 Congress of Evolutionary Computation, Washington D.C, July 6-9, vol. 3, pp. 1651-1656, IEEE Press, 1999.

[Schott, 1995] Schott, J. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Tesis Magistral, Departamento de Aeronáutica y Astronáutica, Instituto de Tecnología de Massachussets.

[Schaerf, 2002] Schaerf A., *Local Search techniques for Constrained Portfolio Selection Problems*, Computational Economics 20: 177–190, 2002.

[Smith y Coit, 1997] Smith A., Coit D.W., *Constraint Handling Techniques—Penalty Functions*, in Thomas Bäck, David B. Fogel and Zbigniew Michalewicz (editors), Handbook of Evolutionary Computation, chapter C 5.2, Oxford University Press and Institute of Physics Publishing, 1997.

[Surry y Radcliffe, 1995] Surry P., Radcliffe N., *A Multi-objective Approach to Constrained Optimization of Gas Supply Networks: The COMOGA Method*, Appears in "Evolutionary Computing: AISB Workshop", Ed: T.C. Fogarty, Springer-Verlag LNCS 993, pp166-180, 1995.

[Van Veldhuizen, 1999] Van Veldhuizen D., *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[Zitzler y Thiele, 1999] Zitzler E., Thiele L., *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*, IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257–271, 1999.

[Zitzler et al, 2000] Zitzler E., Deb K., Thiele L., *Comparison of multiobjective evolutionary algorithms: Empirical results*, Evol. Comput., vol. 8, no. 2, pp. 173–195, Summer 2000

Anexos

A. Detalle de las pruebas de ajuste de parámetros

En esta sección, se presentarán en detalle los resultados obtenidos en las diversas pruebas que se hicieron sobre el algoritmo para determinar la mejor combinación de parámetros posibles.

Por cada prueba, se presentan los siguientes valores:

- **Archivo:** Nombre del archivo de la librería OR utilizado como entrada del algoritmo.
- **Total acciones:** Número de acciones de las que se puede disponer para construir el portafolio de salida
- **K:** Número de acciones que deben estar en el portafolio de salida.
- **Tamaño población:** Número de individuos
- **Individuos aceptados:** Cantidad máxima de individuos que pueden ser incorporados al conjunto de soluciones no-dominadas.
- **Tiempo ejecución:** Tiempo promedio de ejecución de los algoritmos, dependiendo de los parámetros de entrada.
- **No dominados:** Cantidad promedio de individuos obtenidos en el conjunto de no-dominados al finalizar la ejecución del algoritmo.

También se analizó la estabilidad del algoritmo en términos de las soluciones obtenidas en cada conjunto de no-dominados; para ellos, se calculó la media y desviación estándar de los objetivos en juego (expresados en una escala de 10^{-3}).

Tabla 7-1: Resultados al ajuste de la cantidad de individuos.

#test	Archivo	Total acciones	K	Tamaño población	Individuos aceptados	Tiempo ejecución	No dominados	Retorno		Riesgo	
								Media (10 ⁻³)	Desv. Estándar (10 ⁻³)	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)
P1_020	port1.txt	31	16	20	10	7.64	15.34	4.207	0.781	1.153	0.255
P1_040				40	20	15.46	24.92	4.527	1.091	1.196	0.324
P1_100				100	50	42	51.72	5.215	1.294	1.366	0.424
P1_200				200	100	75.78	77.06	5.448	1.427	1.419	0.479
P2_020	port2.txt	85	43	20	10	8.08	13.86	3.115	1.001	0.42	0.11
P2_040				40	20	16.24	25.34	3.285	1.058	0.401	0.108
P2_100				100	50	45.2	52.94	3.69	1.253	0.408	0.133
P2_200				200	100	81.82	85.14	3.987	1.29	0.417	0.153
P5_020	port5.txt	225	110	20	10	11.56	8.52	0.683	0.51	0.682	0.124
P5_040				40	20	22.94	13.52	0.885	0.499	0.641	0.095
P5_100				100	50	61.84	20.3	0.948	0.595	0.608	0.093
P5_200				200	100	115.1	38.08	1.136	0.583	0.611	0.087

Tabla 7-2: Resultados al ajuste de la cantidad de generaciones

#test	Archivo	Total acciones	K	Generaciones	Tiempo ejecución	No dominados	Retorno		Riesgo	
							Media (10 ⁻³)	Desv. Estándar (10 ⁻³)	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)
P1_050	port1.txt	31	16	50	3,62	13,38	4,089	0,831	1,141	0,269
P1_100				100	7,34	15,44	4,028	0,781	1,114	0,226
P1_200				200	14,6	17,24	4,341	0,794	1,154	0,247
P2_050	port2.txt	85	43	50	4,1	12,74	2,948	0,925	0,423	0,119
P2_100				100	8,14	13,68	2,871	0,902	0,400	0,109
P2_200				200	16,26	15,9	3,224	0,946	0,423	0,111
P5_050	port5.txt	225	110	50	5,82	7,22	0,455	0,520	0,682	0,135
P5_100				100	11,76	8,68	0,688	0,511	0,667	0,115
P5_200				200	23,14	9,96	0,960	0,578	0,671	0,118

Tabla 7-3: Resultados al ajuste de la probabilidad de mutación

							Retorno		Riesgo	
#test	Archivo	Total acciones	K	Prob. de mutación	Tiempo ejecución	No dominados	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)
P1_02	port1.txt	31	16	0,2	41,64	52,3	5,232	1,276	1,365	0,420
P1_05				0,5	41,68	64,66	5,263	1,210	1,455	0,396
P1_08				0,8	44,18	63,7	5,421	1,176	1,481	0,388
P2_02	port2.txt	85	43	0,2	44,48	54,38	3,785	1,244	0,412	0,136
P2_05				0,5	44,56	62,56	4,047	1,188	0,442	0,136
P2_08				0,8	45,04	66,7	4,238	1,154	0,466	0,150

Tabla 7-4: Resultados al ajuste del número de acciones a incluir en el portafolio

						Retorno		Riesgo	
#test	Archivo	Total acciones	K (%)	Tiempo ejecución	No dominados	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)	Media (10 ⁻³)	Desv. Estándar (10 ⁻³)
P1_10	port1.txt	31	10%	14,98	31,16	3,693	1,472	0,453	0,177
P1_25			25%	14,98	32,36	5,666	1,962	1,493	0,519
P1_50			50%	14,92	32,06	4,762	1,325	1,015	0,327
P2_10	port2.txt	85	10%	16,28	28,14	3,042	0,757	1,203	0,287
P2_25			25%	16,2	26,74	4,269	0,579	1,440	0,150
P2_50			50%	16,28	30,3	3,606	0,463	1,065	0,137