

**Estructuración y Aplicación de un Algoritmo Cultural
en la Optimización de un Portafolio de Inversión
Enrique José García Arrazola y Julio César Ramírez Pisco
Universidad Tecnológica de Pereira**

**Facultad de Ingeniería Industrial
Maestría en Investigación Operativa y Estadística
Pereira, 2017**

**Estructuración y Aplicación de un Algoritmo Cultural
en la Optimización de un Portafolio de Inversión
Enrique José García Arrazola y Julio César Ramírez Pisco
Universidad Tecnológica de Pereira**

**Tesis presentada como requisito para optar el título de
Magister en Investigación Operativa y Estadística**

Director

Robinson Pacheco García

Ingeniero Industrial

MSc. en Gerencia Ambiental

Candidato a Doctor en Ingeniería Énfasis en el Área Industrial

Facultad de Ingeniería Industrial

Maestría en Investigación Operativa y Estadística

Pereira, 2017

Agradecimientos

A mi madre María Elvia,

A mi padre Julio César,

A mi esposa Martha Inés y mis hijas María Camila y Tatiana Alejandra,

A mis hermanos,

**A todo el sin número de estudiantes que han pasado por
mis aulas y cuyas inquietudes me han motivado a desarrollar estos estudios.**

Julio César Ramírez Pisco

Agradezco a Dios,

a mi hermano Julio Ramírez (juls),

a mis padres,

y a Napo.

Enrique José García Arrazola

Tabla de contenido

Resumen.....	14
Abstract.....	15
Objetivos.....	17
Objetivo General.....	17
Objetivos Específicos.....	17
Estado del Arte.....	18
Optimización de Portafolio.....	18
Teoría de la Cartera de Markowitz.....	18
Modelo de Harry Markowitz.....	19
Hipótesis del Modelo de Markowitz.....	19
Riesgo.....	21
Riesgo de cartera.....	22
Covarianza de retornos.....	22
Covarianza y correlación.....	22
Coeficiente de correlación.....	23
Actitud frente al riesgo.....	23
Retorno esperado de activos individuales.....	25
Retorno esperado de la cartera.....	26
Diversificación.....	26
Diversificación y números de activos.....	27
Aplicación del modelo de Markowitz al cálculo del valor del riesgo (VAR) a un portafolio de divisas, (Herrera, abril-junio 1999).....	27
Algoritmo genético.....	28
El Algoritmos Genético Simple.....	31

Aplicaciones de los algoritmos genéticos en la optimización de portafolios	32
Algoritmo cultural	58
Aplicación de los algoritmos culturales.....	63
Funcionamiento del Algoritmo Cultural.....	69
Composición de la aplicación	74
Funcionamiento del Algoritmo Cultural	74
Pseudocódigo.....	74
Funcionamiento del algoritmo genético convencional.....	81
Pseudocódigo.....	81
Funcionamiento del Algoritmo de Hill Climbing	85
Pseudocódigo.....	85
Aplicación en MATLAB.....	86
Análisis y Resultados	89
Condiciones generales.....	89
Ajustes de parámetros	90
Resultados	91
Caso 1: 2 Acciones – Máximo Desempeño	91
Algoritmo Cultural	91
Algoritmo Genético	93
Algoritmo Ascenso A la Colina.....	94
Caso 1: 2 Acciones – Mínimo Riesgo.....	97
Algoritmo Cultural	97
Algoritmo Genético	99
Algoritmo Ascenso A la Colina.....	100
Caso 2: 5 Acciones – Máximo Desempeño	103

Algoritmo Cultural	103
Algoritmo Genético	105
Algoritmo Ascenso A la Colina.....	106
Caso 2: 5 Acciones – Mínimo Riesgo.....	109
Algoritmo Cultural	109
Algoritmo Genético	111
Algoritmo Ascenso A la Colina.....	112
Caso 3: 8 Acciones – Máximo Desempeño	116
Algoritmo Cultural	116
Algoritmo Genético	118
Algoritmo Ascenso A la Colina.....	119
Caso 3: 8 Acciones – Mínimo Riesgo.....	122
Algoritmo Cultural	122
Algoritmo Genético	124
Algoritmo Ascenso A la Colina.....	125
Análisis	129
Conclusiones	131
Trabajo Futuro	133
Anexos	134
Bibliografía	158

Índice de Figuras

Figura 1 Marco de referencia de un Algoritmos Culturales	60
Figura 2 Diagrama algoritmo cultural propuesto.....	80
Figura 3 Interface Gráfica en Matlab.....	86
Figura 4 Respuesta Algoritmo Cultural - 2 Acciones - Máximo Desempeño	91
Figura 5 Mutación y Recombinación Algoritmo Cultural - 2 Acciones - Máximo Desempeño.	92
Figura 6 Composición Algoritmo Cultural - 2 Acciones - Máximo Desempeño.....	92
Figura 7 Respuesta Algoritmo Genético - 2 Acciones - Máximo Desempeño.....	93
Figura 8 Mutación y Recombinación Algoritmo Genético - 2 Acciones - Máximo Desempeño	93
Figura 9 Composición Algoritmo Genético - 2 Acciones - Máximo Desempeño.....	94
Figura 10 Respuesta Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño	94
Figura 11 Mutación y Recombinación Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño.....	95
Figura 12 Composición Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño ..	95
Figura 13 Respuesta Algoritmo Cultural - 2 Acciones - Mínimo Riesgo	97
Figura 14 Mutación y Recombinación Algoritmo Cultural - 2 Acciones - Mínimo Riesgo	98
Figura 15 Composición Algoritmo Cultural - 2 Acciones - Mínimo Riesgo	98
Figura 16 Respuesta Algoritmo Genético - 2 Acciones - Mínimo Riesgo	99
Figura 17 Mutación y Recombinación Algoritmo Genético - 2 Acciones - Mínimo Riesgo.....	99
Figura 18 Composición Algoritmo Genético - 2 Acciones - Mínimo Riesgo	100
Figura 19 Respuesta Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo.....	100

Figura 20 Mutación y Recombinación Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo	101
Figura 21 Composición Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo	101
Figura 22 Respuesta Algoritmo Cultural - 5 Acciones - Máximo Desempeño	103
Figura 23 Mutación Y Recombinación Algoritmo Cultural - 5 Acciones - Máximo Desempeño	104
Figura 24 Composición Algoritmo Cultural - 5 Acciones - Máximo Desempeño	104
Figura 25 Respuesta Algoritmo Genético - 5 Acciones - Máximo Desempeño.....	105
Figura 26 Mutación y Recombinación Algoritmo Genético - 5 Acciones - Máximo Desempeño	105
Figura 27 Composición Algoritmo Genético - 5 Acciones - Máximo Desempeño.....	106
Figura 28 Respuesta Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño	106
Figura 29 Mutación y Recombinación Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño.....	107
Figura 30 Composición Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño	107
Figura 31 Respuesta Algoritmo Cultural - 5 Acciones - Mínimo Riesgo	109
Figura 32 Mutación y Recombinación Algoritmo Cultural - 5 Acciones - Mínimo Riesgo	110
Figura 33 Composición Algoritmo Cultural - 5 Acciones - Mínimo Riesgo	110
Figura 34 Respuesta Algoritmo Genético - 5 Acciones - Mínimo Riesgo	111
Figura 35 Mutación y Recombinación Algoritmo Genético - 5 Acciones - Mínimo Riesgo....	111
Figura 36 Composición Algoritmo Genético - 5 Acciones - Mínimo Riesgo	112
Figura 37 Respuesta Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo.....	112

Figura 38 Mutación y Recombinación Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo	113
Figura 39 Composición Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo	113
Figura 40 Respuesta Algoritmo Cultural - 8 Acciones - Máximo Desempeño	116
Figura 41 Mutación y Recombinación Algoritmo Cultural - 8 Acciones - Máximo Desempeño	117
Figura 42 Composición Algoritmo Cultural - 8 Acciones - Máximo Desempeño	117
Figura 43 Respuesta Algoritmo Genético - 8 Acciones - Máximo Desempeño	118
Figura 44 Mutación y Recombinación Algoritmo Genético - 8 Acciones - Máximo Desempeño	118
Figura 45 Respuesta Algoritmo Genético - 8 Acciones - Máximo Desempeño	119
Figura 46 Respuesta Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño	119
Figura 47 Mutación y Recombinación Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño	120
Figura 48 Composición Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño	120
Figura 49 Respuesta Algoritmo Cultural - 8 Acciones - Mínimo Riesgo	122
Figura 50 Mutación y Recombinación Algoritmo Cultural - 8 Acciones - Mínimo Riesgo	123
Figura 51 Composición Algoritmo Cultural - 8 Acciones - Mínimo Riesgo	123
Figura 52 Respuesta Algoritmo Genético - 8 Acciones - Mínimo Riesgo	124
Figura 53 Mutación y Recombinación Algoritmo Genético - 8 Acciones - Mínimo Riesgo	124
Figura 54 Composición Algoritmo Genético - 8 Acciones - Mínimo Riesgo	125
Figura 55 Respuesta Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo Riesgo	125

Figura 56 Mutación y Recombinación Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo

Riesgo 126

Figura 57 Composición Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo Riesgo 126

Índice de Tablas

Tabla 1 Condiciones normales - algoritmo cultural.....	75
Tabla 2 Vector ordenado – algoritmo cultural.....	75
Tabla 3 Aceptación y rechazo – algoritmo cultural	76
Tabla 4 Población General – Algoritmo Cultural	76
Tabla 5 Individuos rechazados – algoritmo cultural.....	77
Tabla 6 Aplicación de función - algoritmo cultural.....	77
Tabla 7 Nueva población - algoritmo cultural	77
Tabla 8 Individuos Rechazados - Algoritmo Cultural	78
Tabla 9 Aplicación modificador - algoritmo cultural	78
Tabla 10 Población rechazada - algoritmo cultural	78
Tabla 11 Aplicación operador - algoritmo cultural.....	78
Tabla 12 Población rechazada - algoritmo cultural	79
Tabla 13 Operador a aplicar - algoritmo cultural.....	80
Tabla 14 Recombinación - Algoritmo Genético	83
Tabla 15 Mutación - algoritmo genético.....	83
Tabla 16 Individuo mutado - algoritmo genético.....	84
Tabla 17 Resultados Máximo Desempeño - 2 Acciones	96
Tabla 18 Composición Portafolio de Inversión 2 Acciones – Máximo Desempeño.....	96
Tabla 19 Resultados Mínimo Riesgo - 2 Acciones.....	102
Tabla 20 Composición Portafolio de Inversión 2 Acciones – Mínimo Riesgo	102
Tabla 21 Consolidado Portafolio Inversión 2 Acciones	102
Tabla 22 Resultados Máximo Desempeño - 5 Acciones	108

Tabla 23 Composición Portafolio de Inversión 5 Acciones – Máximo Desempeño	108
Tabla 24 Resultados Mínimo Riesgo - 5 Acciones.....	114
Tabla 25 Composición Portafolio de Inversión 5 Acciones – Mínimo Riesgo	114
Tabla 26 Consolidado Portafolio Inversión 5 Acciones	115
Tabla 27 Resultados Máximo Desempeño - 8 Acciones	121
Tabla 28 Composición Portafolio de Inversión 8 Acciones – Máximo Desempeño	121
Tabla 29 Resultados Mínimo Riesgo - 8 Acciones.....	127
Tabla 30 Composición Portafolio de Inversión 8 Acciones – Mínimo Riesgo	127
Tabla 31 Consolidado Portafolio Inversión 8 Acciones	127

Índice de Anexos

Anexo 1 FUNCION Aplicación.m	134
Anexo 2 FUNCION AG.m	139
Anexo 3 FUNCION Generar_Indivuo.m	142
Anexo 4 FUNCION Aceptar_Individuos.m	143
Anexo 5 FUNCION Cruzar.m	144
Anexo 6 FUNCION Evaluar_f.m	145
Anexo 7 FUNCION Mutar.m	146
Anexo 8 FUNCION Normalizacion.m	147
Anexo 9 FUNCION Markowitz.m	148
Anexo 10 FUNCION Retornos_Activos.m	149
Anexo 11 FUNCION Ruleta.m	151
Anexo 12 FUNCION Exportar.m	152
Anexo 13 FUNCION Cultural.m	153
Anexo 14 FUNCION Creencias.m	155
Anexo 15 FUNCION Hill.m	157

Resumen

Tomando como base los aspectos fundamentales del método de computo evolutivo estocástico, el cual usa procesos aleatorios para determinar la dirección de búsqueda de una solución, para este caso en particular y mediante la estructuración en Matlab de un código que permitió ejecutar el Algoritmo Cultural propuesto por Robert G. Reynolds en 1978; abordando el problema de la optimización de portafolios de inversión bajo el modelo de Harry Markowitz presentado en 1952; como alternativa metodológica, permitiendo generar un portafolio de inversión eficiente, estable y diversificado; bajo el criterio de máximo beneficio o mínimo riesgo; se analizaron los dos criterios para 2, 5 y 8 diferentes grupos de acciones que conformarían el portafolio; bajo un ambiente de Algoritmo Cultural y se confrontó frente a un Algoritmo Genético convencional y un Algoritmo de Ascenso a la Colina; permitiendo de esta forma presentar una propuesta para la diversificación del portafolio.

Palabras Clave: Portafolio de Inversión, Cartera de Markowitz, Algoritmos Genéticos, Algoritmos culturales, Riesgo.

Abstract

Taking as a base the fundamental aspects of the method of evolutionary calculation stochastic, which uses chance processes to determine the direction of search of a solution, for this case in particular and by means of the structure in Matlab of a code who allowed to execute the Cultural Algorithm proposed by Robert G. Reynolds in 1978; tackling the problem of the optimization of briefcase of investment under the model of Harry Markowitz presented in 1952; as methodological alternative, allowing to generate a briefcase of efficient investment, stable and diversified; under the criterion of maximum benefit or minimal risk; two criteria were analyzed for 2, 5 and 8 different groups of actions that would shape the briefcase; under an ambience of Cultural Algorithm and it bordered opposite to a Genetic conventional Algorithm and an Algorithm of Ascent to the Hill; allowing thus to present a proposal for the diversification of the briefcase.

Key words: Briefcase of Investment, Portfolio of Markowitz, Genetic Algorithms, cultural Algorithms, Risk.

Estructuración y Aplicación de un Algoritmo Cultural en la Optimización de un Portafolio de Inversión

En gran medida el sector financiero y el mercado de capitales se ha visto envuelto en diferentes cambios que involucran un alto grado de incertidumbre y riesgo ocasionado por los diferentes sucesos económico-financieros que se presentan en la economía, como por ejemplo, la inflación y la deflación, las tasas de interés, los mercados internacionales, las diferentes fluctuaciones que ha tenido el dólar en los últimos tiempos, los desarrollos internos que se producen en las empresas, los sucesos mundiales, la inflación y tasas de Interés, las tasas de cambio y el despliegue publicitario de nuevos productos o servicios y sobre la base, (Universidad EAFIT, 2007), que el mercado de capitales es una “herramienta” básica para el desarrollo económico de una sociedad, ya que, mediante él, se hace la transición del ahorro a la inversión; moviliza recursos principalmente de mediano y largo plazo, desde aquellos sectores que tienen dinero en exceso (ahorradores o inversionistas) hacia las actividades productivas (empresas, sector financiero, gobierno) mediante la compraventa de títulos valores; se hace necesario plantear una propuesta de solución al problema de la optimización de portafolio de inversión bajo la teoría de cartera de Harry Markowitz y los Algoritmos Culturales propuestos por Robert G. Reynolds

El alcance planteado para el presente trabajo de investigación es poder hacer un acercamiento a la optimización de portafolios de inversión mediante la aplicación de un algoritmo de computo evolutivo como herramienta genérica de optimización; sustentado sobre una metaheurística utilizada dentro de la familia de los Algoritmos Genéticos específicamente un Algoritmo Cultural.

El marco teórico que sustentará el desarrollo del presente trabajo, se estructura bajo los aspectos fundamentales de los mercados de capitales, la gestión de cartera, la teoría de cartera de Harry Markowitz y los Algoritmos Culturales propuestos por Robert G. Reynolds

La primera sección de este documento describe los aspectos fundamentales de la optimización de portafolio y el modelo de Markowitz, de conformidad con lo expuesto por (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), mediante el cual se demostró que los inversores deberían actuar de un modo totalmente diferente; es decir estos deben optar por portafolios de varios activos en vez que invertir en un solo activo. Siguiendo este consejo de mantener un portafolio de activos, un inversor puede reducir el nivel de riesgo al cual está exponiéndose, mientras que mantiene el nivel esperado de rentabilidad.

La segunda parte aborda lo relacionado con los algoritmos genéticos, su aplicación, estructura y la participación de estos en la optimización de portafolios de inversión.

La tercera parte describe la construcción del algoritmo cultural, su aplicación, estructura y la participación de estos en la optimización de portafolios de inversión.

La cuarta parte de este documento se trabajan diferentes portafolios de inversión, conformados por 2, 5 y 8 diferentes tipos de acciones.

Objetivos

Objetivo General

A través de la construcción de un marco conceptual centrado en la teoría de la optimización de portafolio de inversión propuesta por Harry Markowitz en 1952 y utilizando la estructura de los Algoritmos Culturales desarrollados por Robert G. Reynolds en 1978; desarrollar una aplicación en Matlab que permita modelar una solución al problema de selección de un Portafolio de Inversión.

Objetivos Específicos

a) Desarrollar una propuesta de solución al problema de selección de portafolio de inversión basada en un Algoritmo Cultural bajo la propuesta desarrollada por Robert G. Reynolds en 1978

y soportada en la aplicación de un algoritmo de computo evolutivo como herramienta genérica de optimización.

b) Estructurar un código en Matlab que permita evidenciar la aplicación del algoritmo cultural en la optimización de portafolios de inversión.

c) Evaluar la aplicación del Algoritmos Culturales al problema de Selección de Portafolio de Inversión y comparar su rendimiento frente a un Algoritmo Genético convencional y al algoritmo de ascenso a la colina.

Estado del Arte

Optimización de Portafolio

Tanto en los mercados de capitales, como en la gestión de cartera, para la aplicación del Modelo de Markowitz, (López, Mercado de Capitales y Gestión de Cartera, 2008, pág. 62), de acuerdo con lo establecido por López, se ha dicho que se puede dividir la historia de las Inversiones en 2 partes, antes y después de 1952, año en que el economista Harry Markowitz publicó su tesis doctoral acerca de Selección de Portafolios.

Harry Markowitz fue el primero en poner atención en la práctica de diversificación de los portafolios. Esta es la base donde los inversores generalmente prefieren mantener portafolios de activos en vez que activos individuales, debido a que ellos no tienen en cuenta solamente los retornos de dichos activos sino también el riesgo de los mismos (López, Mercado de Capitales y Gestión de Cartera, 2008, pág. 62).

Teoría de la Cartera de Markowitz

De igual forma López (2008), nos presenta de manera clara la fundamentación del trabajo de Markowitz (1952), entendiéndose que los inversores solamente prestaban atención en maximizar el nivel esperado de retornos. Si esto era lo que hacían, entonces un inversor calcularía

simplemente el grado esperado de rendimientos de un conjunto de activos y luego invertiría todo su dinero en aquel activo que proporcione la mayor rentabilidad esperada.

En el trabajo de selección de Inversiones, Markowitz demostró que los inversores deberían actuar de un modo totalmente diferente. Los inversores deben optar por portafolios de varios activos en vez que invertir en un solo activo. Siguiendo este consejo de mantener un portafolio de activos (Diversificación) un inversor puede reducir el nivel de riesgo al cual está exponiéndose, mientras que mantiene el nivel esperado de rentabilidad.

Modelo de Harry Markowitz

La descripción del Modelo de Harry Markowitz, se describe de conformidad el documento presentado por Luis Franco, (Luis C. Franco Arbelaez, Junio de 2011); quien describe que en 1952 el economista norteamericano Harry Markowitz, especialista en análisis de inversiones, publicó un artículo llamado “Portfolio Selection” en donde expone su teoría sobre cómo hallar la composición óptima de un portafolio de valores, maximizando la rentabilidad para un determinado nivel máximo de riesgo aceptable; o en forma alternativa, minimizar el riesgo para una rentabilidad mínima esperada.

Hipótesis del Modelo de Markowitz

El modelo de Markowitz parte de las siguientes hipótesis:

a) El rendimiento de cualquier portafolio, es considerado una variable aleatoria, para la cual el inversionista estima una distribución de probabilidad para el periodo de estudio. El valor esperado de la variable aleatoria es utilizado para cuantificar la rentabilidad de la inversión;

b) la varianza o la desviación estándar son utilizadas para medir la dispersión, como medida del riesgo de la variable aleatoria rentabilidad; ésta medición debe realizarse en forma individual, a cada activo y a todo el portafolio.

c) la conducta racional del inversionista lo lleva a preferir la composición de un portafolio que le represente la mayor rentabilidad, para determinado nivel de riesgo.

La formulación matemática primal del modelo de Markowitz, que se presenta en (1), consiste en determinar las ponderaciones w_i que maximizan el rendimiento esperado del portafolio, sujeto a un riesgo máximo admitido. Es decir:

$$\text{Max } E(R_p) = \sum_{i=1}^n w_i \cdot E(R_i) \quad (1)$$

Sujeto a

$$\sigma^2(R_p) = \sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \cdot \sigma_{ij} \leq \sigma_0^2$$

$$\sum_{i=1}^n w_i = 1; \quad w_i \geq 0 \quad (i = 1, \dots, n)$$

donde n es el número de activos en el portafolio; R_i es la variable aleatoria rendimiento del activo i ; $E(R_i)$ es el rendimiento esperado del activo i ; R_p es la variable aleatoria rendimiento del portafolio; $E(R_p)$ es el rendimiento esperado del portafolio; w_i es la proporción del presupuesto del inversionista destinado al activo i ; $\sigma^2(R_p)$ es la varianza del rendimiento del portafolio; σ_{ij} es la covarianza entre los rendimientos de los activos i y j ; y σ_0^2 es la varianza máxima admitida.

La formulación dual alternativa consiste en determinar las ponderaciones que minimizan la varianza del portafolio, sujeto a un rendimiento mínimo requerido para el portafolio. En forma matemática (2):

$$\text{Min } \sigma^2(R_p) = \sum_{i=1}^n \sum_{j=1}^n w_i \cdot w_j \cdot \sigma_{ij} \quad (2)$$

Sujeto a

$$E(R_p) = \sum_{i=1}^n w_i \cdot E(R_i) \geq \mu_0$$

$$\sum_{i=1}^n w_i = 1; \quad w_i \geq 0 \quad (i = 1, \dots, n)$$

donde μ_0 es el rendimiento mínimo requerido.

Con cualquiera de las dos alternativas, optimizando la varianza o el valor esperado, se encuentran las ponderaciones de los activos, que optimizan el objetivo con las restricciones dadas, y se puede determinar un conjunto de portafolios eficientes, que proporcionen el máximo rendimiento para cada nivel de riesgo.

El principal aporte del modelo de Markowitz para la selección de un portafolio óptimo se encuentra en su utilidad para recoger los aspectos fundamentales que deben guiar a un inversionista racional en la elección de la composición de su portafolio, de tal forma, que le produzca la máxima rentabilidad, al controlar el riesgo; o en forma alternativa, minimizar el riesgo, controlando el rendimiento.

Riesgo

Basado en el desarrollo teórico de López, en su documento: Mercado de Capitales y Gestión de Cartera, (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017); se sustenta la Teoría de Riesgo, Riesgo de Cartera, Covarianza de Retornos, Covarianza y Correlación; el riesgo de un activo es medido comúnmente como la desviación estándar de sus retornos. Es decir, el grado en el que los retornos de ese activo se dispersan del retorno esperado promedio del mismo. En este caso la medida de riesgo utilizada es la *varianza* o la *desviación típica*.

Si la distribución de los rendimientos es simétrica, entonces la varianza es el doble de la semi-varianza y en consecuencia proporcionan la misma información, pero con la ventaja que su tratamiento matemático es mucho más simple.

Para un activo dado cuanto mayor es el grado de dispersión de los retornos mayor es el grado de riesgo o volatilidad, como comúnmente se lo llama en finanzas.

Se debe tener en cuenta que el riesgo de una cartera no es igual al promedio ponderado de los riesgos de los activos que la componen. El grado en el que los activos de una cartera son similares o diferentemente afectados por determinados hechos tiene un peso en cuanto al riesgo de la cartera.

Esto tiene importantes implicancias para los inversores, como el significado en la cual ellos pueden elegir ciertos activos para incluirlos en una cartera el cual es ayudará a reducir el riesgo de la misma.

Riesgo de cartera

El riesgo de una cartera es una función del nivel de riesgo individual de cada uno de los activos que la componen y también del grado de correlación existente entre los retornos esperados de cada uno de los activos que componen la cartera.

Covarianza de retornos

La desviación estándar de una cartera no es simplemente el promedio ponderado de los desvíos estándar de los activos que la componen. La diferencia entre las dos cifras es debido a la covarianza de retornos entre un activo A y otro activo B. El riesgo de la cartera no solo depende del riesgo (desviación estándar) de cada uno de los activos que lo componen sino también de la manera en que covarían cada uno de los activos de la cartera.

Covarianza y correlación

El riesgo de una cartera dependerá de la extensión por la cual los activos son similarmente afectados por eventos subyacentes. La covarianza de dos activos podría ser positiva o negativa. Si dos activos son afectados de manera similar ante un evento cualquiera donde los retornos de los mismos aumentan o disminuyen en forma conjunta, entonces la covarianza será positiva, del

mismo modo, si ante un evento subyacente los retornos del activo A se comporta de manera totalmente opuesta (Aumentan) a los del otro activo B (Disminuyen), entonces la covarianza será negativa. La covarianza será cercana a cero si el desvío de los retornos de cada uno de los activos se encuentra incorrelacionados.

Coefficiente de correlación

Con base en lo tratado por Szretter, (Eugenia, 2013), el análisis de correlación nos da una perspectiva de la dirección de la relación existente entre dos o más variables, Estandarizando la covarianza todos los valores de correlación estarán comprendidos entre -1 y +1 llegando a lo que se denomina coeficiente de correlación. Si los retornos sobre dos Activos se mueven exactamente juntos, entonces el coeficiente de correlación será igual a +1 (correlación lineal perfecta positiva). Si los retornos de dos activos se mueven en direcciones exactamente opuestas, el coeficiente de correlación será igual a -1 (correlación lineal perfecta negativa). Finalmente, si los retornos de dos activos se mueven independientemente el coeficiente de correlación será igual a cero (incorrelacionados).

Actitud frente al riesgo

La teoría de la cartera de Markowitz se basa en la idea que el comportamiento de un inversor se caracteriza por el grado de aversión al riesgo que tenga y el grado de maximización de utilidades que espera.

Lo descrito por Delfino, (Delfino, 2017) y López (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), los inversores pueden encontrarse dentro los siguientes grupos de aversión al riesgo:

- a) Aversos al riesgo: Es aquel inversor que elegiría una inversión con el menor grado de riesgo frente a dos alternativas con el mismo nivel de retorno esperado.

- b) Propensos al riesgo: Es aquel inversor que elegiría una inversión con el mayor grado de riesgo frente a dos alternativas con el mismo nivel de retorno esperado.
- c) Neutrales al riesgo: Es aquel inversor que se mantendría indiferente si tuviera que elegir entre dos alternativas con el mismo nivel de retorno esperado.

La actitud frente al riesgo de disímiles inversores depende de diferentes cosas como por ejemplo su edad, su situación financiera, un inversor con un nivel de ganancias altos sin obligaciones financieras estaría más dispuesto a soportar potenciales pérdidas de capital y tendría menor aversión al riesgo que un inversor con un nivel de ganancias menores. Es bueno aclarar que la mayoría de los inversores se encuentran dentro del grupo de aversos al riesgo.

La idea de aversión al riesgo está basada sobre la teoría de la utilidad, de conformidad con López (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), la misma se refiere a un conjunto de alternativas entre las que se define una relación de indiferencia y una relación de preferencia para un individuo que debe tomar una decisión. Mide el grado de satisfacción de un agente económico de acuerdo a distintos niveles de riesgo.

Se define a su vez una función denominada función de utilidad, tal que a cada alternativa le corresponde un número llamado utilidad de esa alternativa, y tal que, si una alternativa es preferida a otra, entonces la utilidad de la primera es mayor a la de la segunda.

$$U(A) > U(B)$$

Sobre la base de esta teoría el criterio óptimo de decisión es el de máxima utilidad esperada, se entiende que la utilidad esperada de una opción A que puede tener las n_i alternativas y cada una de ellas con probabilidad p_i , es:

$$E U(A) = \sum U(A_i) \cdot p_i$$

La función de *utilidad* siempre es *creciente* (Primera Derivada), debido a que en economía se dice que los agentes económicos son racionales si una mayor Riqueza le produce una mayor utilidad.

Un individuo es **averso al riesgo**, cuando su función de *utilidad marginal* (segunda derivada) *es decreciente*.

Un individuo es **neutral al riesgo**, cuando su función de *utilidad marginal* (segunda derivada) *es constante*.

Un individuo es **propenso al riesgo**, cuando su función de *utilidad marginal* (segunda derivada) *es creciente*.

Retorno esperado de activos individuales

López (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), reconoce que la mayoría de los inversores prefieren invertir en portafolios en vez de activos individuales. Es importante entender como el riesgo y retorno esperado de un activo contribuye a determinar el riesgo y retorno esperado de un portafolio.

Para calcular retornos esperados, debemos evaluar las distintas probabilidades de ocurrencia que pueden llegar a tener en un futuro las condiciones del mercado.

Se puede obtener el rendimiento esperado de un activo a través de la sumatoria del producto de cada probabilidad por el rendimiento que le corresponde a cada condición de mercado.

El rendimiento esperado de un activo cualquiera se calcula utilizando la siguiente fórmula:

$$EA = P_1R_1 + P_2R_2 + P_3R_3 + \dots + P_nR_n$$

Donde:

EA = Retorno esperado del Activo A

P_n = Probabilidad de ocurrencia de la condición de mercado

R_n = Retorno del Activo A si se diera la condición de mercado.

Retorno esperado de la cartera

De acuerdo a lo descrito por Medina (Medina, 2033) y López (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), es igual al promedio ponderado del retorno de cada uno de los activos que la componen. La fórmula para calcular el rendimiento o retorno esperado de una cartera es la siguiente:

$$EC = X_1E_1 + X_2E_2 + X_3E_3 + \dots + X_nE_n$$

Donde:

EC = Retorno esperado de la Cartera

X_n = Proporción invertida en el activo n

E_n = Retorno esperado del Activo n

Diversificación

Siguiendo a López (López, http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf, 2017), el objetivo fundamental de las inversiones puede ser expresado de dos maneras, los inversores deben tender a maximizar sus retornos dados un nivel x de riesgo o a minimizar el riesgo dado un nivel x de retorno. La diversificación puede ayudar a los inversores a lograr este objetivo fundamental. Esto significa invertir la totalidad de los fondos en un número de activos de diferentes sectores donde los retornos no están directamente relacionados la mayoría del tiempo, con el objetivo de reducir el riesgo.

Un ejemplo de una cartera con solamente dos activos, la efectividad de la diversificación puede ser analizada de acuerdo a distintos niveles de correlación.

Retornos con correlación perfecta y positiva, Si los retornos de dos activos están correlacionados perfectamente y de forma positiva, los mismos tendrán movimientos en la misma

dirección y unificados. Entonces el rendimiento y riesgo de esa cartera será iguales al promedio ponderado del riesgo y retorno de los activos que la componen.

Retornos con correlación perfecta y negativa, Si los retornos de dos activos están correlacionados perfectamente de forma negativa, los mismos tendrán movimientos en direcciones opuestas. Entonces si logramos formar una cartera con activos que se encuentren perfectamente correlacionados en forma negativa podemos garantizar la eliminación del riesgo de la cartera.

Retornos incorrelacionados, si los retornos de ambos activos no tienen ninguna relación, quieren decir que los activos están incorrelacionados, si combinamos 2 activos incorrelacionados pueden ayudar a reducir el riesgo de una cartera.

Diversificación y números de activos

Cuanto mayor sea el número de activos incluido en un portafolio, mayor será el impacto en el riesgo debido a la diversificación.

Aplicación del modelo de Markowitz al cálculo del valor del riesgo (VAR) a un portafolio de divisas, (Herrera, abril-junio 1999).

Para dar una explicación breve acerca de la teoría de portafolios de Markowitz en el cálculo del valor del riesgo (VAR) a un portafolio de divisas, se toma lo propuesto por Herrera (Herrera, abril-junio 1999, págs. 53-60), quien describe, a grandes rasgos, el planteamiento de como seleccionar de una manera óptima el portafolio de inversión esbozado por Harry Markowitz y que, posteriormente, se convirtiera en la simiente de la teoría moderna en este tipo de inversiones. El objetivo es sentar las bases de su aplicación, y en última instancia contribuir a su divulgación. Es necesario aclarar que el desarrollo de este trabajo no está dirigido para especialistas de la materia, si no para quien desee o necesite tener un primer acercamiento a la misma; por estas razones se aborda en enfoque de Markowitz desde una perspectiva teórica son intento alguno de crítica y

presenta un ejemplo de su aplicación en la estimación del valor en riesgo de un portafolio de divisas.

Partiendo del concepto del valor del riesgo o VaR el cual resume la pérdida máxima esperada (o peor pérdida) a lo largo de un horizonte de tiempo objetivo dentro de un intervalo de confianza dado. Esta medida presenta la exposición al riesgo de los activos de manera simple y fácil de entender.

Para ejemplificar la aplicación del modelo de Markowitz al cálculo del VaR, se supone que se cuenta con una inversión cuyo valor presente es de 100.000 dólares en un portafolio conformado por libras esterlinas, marcos alemanes, y yenes japoneses.

En relación con la definición de VaR, el problema consiste en encontrar la máxima pérdida probable a que está sujeto el valor de este portafolio en un día de operación dado un nivel de confianza, Para ello lo que se realiza comúnmente es tomar a consideración los datos de cotización de cada una de las monedas en dólares en un periodo de 100 días de operación.

Para determinar el riesgo en el portafolio se utilizó la fórmula propuesta por Markowitz para el cálculo de la varianza (riesgo), y a partir de esta calcular la desviación estándar del portafolio para poder determinar el VaR dado un nivel de confianza seleccionado.

Con los datos completos se procede al cálculo de la varianza y desviación estándar del portafolio y teniendo en cuenta este resultado se puede estimar el VaR, al nivel de confianza seleccionado.

Algoritmo genético

Es necesario incluir algunos de los aspectos fundamentales sobre los cuales se ha construido un nuevo marco teórico, en relación con la aplicación de los algoritmos genéticos a la optimización

de portafolios accionarios, que permitan resolver el modelo de selección de inversiones planteado por Markowitz.

De conformidad con lo planteado por Gipuzkoa, (Algoritmos Genéticos, Campus Gipuzkoa, 2015); los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización.

Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los algoritmos genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

Los principios básicos de los algoritmos genéticos fueron establecidos por Holland (1975), usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos (descendientes de los anteriores), los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el algoritmo genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

El poder de los algoritmos genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el algoritmo genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria. El gran campo de aplicación de los algoritmos genéticos se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas hibridándolas con los algoritmos genéticos.

Tomando como referencia lo planteado por Granada (Granada, 2013), el mecanismo original de la evolución de los seres vivos se fundamenta en la competición, la cual selecciona los individuos mejor adaptados a su ambiente y asegura su descendencia. De esta forma las características más usadas por los padres, que les permitieron sobrevivir, son heredadas a los hijos. En el contexto de los algoritmos genéticos, los individuos sometidos a la evolución son las alternativas de solución, las cuales hasta ahora han sido representadas por arreglos vectoriales que comúnmente son denominados pseudo-cromosomas.

En concordancia con la corriente darwinista, la selección de los individuos para la reproducción debe tener preferencia por los mejores. De igual forma la selección de los que deben desaparecer debe tener preferencia por los peores. Para poder aplicar estos dos operadores de selección se hace necesario obtener una medida de calidad de cada individuo. La calidad de un individuo está asociada al valor de la función objetivo y al cumplimiento de las restricciones. La forma como se calcula esta métrica de calidad recibe el nombre de función de adaptación.

El Algoritmos Genético Simple

A manera de aclaración, se presenta el modelo general propuesto del pseudo-código del Algoritmo Genético Simple

BEGIN /* Algoritmo Genético Simple */

Generar una población inicial.

Computar la función de evaluación de cada individuo.

WHILE NOT Terminado DO

BEGIN /* Producir nueva generación */

FOR Tamaño población/2 **DO**

BEGIN /*Ciclo Reproductivo */

Seleccionar dos individuos de la anterior generación, para el cruce (probabilidad de selección proporcional a la función de evaluación del individuo).

Cruzar con cierta probabilidad los dos individuos obteniendo dos descendientes.

Mutar los dos descendientes con cierta probabilidad.

Computar la función de evaluación de los dos descendientes mutados.

Insertar los dos descendientes mutados en la nueva generación.

END

IF la población ha convergido **THEN**

Terminado: = TRUE

END

END

Aplicaciones de los algoritmos genéticos en la optimización de portafolios

La literatura que actualmente se encuentra en torno a la aplicación de los Algoritmos genéticos en la optimización de portafolios de inversión nos brinda lo planteado por (Kostas Metaxiotis, oct 2012), en su artículo *Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review*, el artículo reúne el análisis de 91 documentos en los cuáles el objetivo principal se centra en las medidas de riesgo aplicados en la formulación de modelos de gestión de cartera con el apoyo de los algoritmos evolutivos multiobjetivo (MOEA).

Los propósito del trabajo fueron los de presentar el estado actual de la investigación en gestión de la cartera con el apoyo de MOEA, desarrollar un marco metodológico para la realización de un estudio exhaustivo de la bibliografía sobre la base de los artículos publicados en MOEA para la gestión de la cartera en un período largo de tiempo a través de varias disciplinas; utilizar el marco teórico para obtener una comprensión de la situación actual de los MOEA para el campo de la investigación en gestión de la cartera e identificar las áreas potenciales de preocupación en lo que respecta a MOEA para la gestión de la cartera.

De conformidad con lo presentado la aplicación de algoritmos evolutivos para la solución de problemas de optimización multiobjetivo, presenta el siguiente desarrollo: la primera implementación de un algoritmo evolutivo multiobjetivo (MOEA) se remonta a mediados de la

década de 1980, en 1985 se introducen el algoritmo genético evaluación vectorial (VEGA) que fue la primera implementación de un algoritmo evolutivo multiobjetivo (MOEA), en 1989 se propone un procedimiento de selección no dominados para superar la debilidad de VEGA mediante algoritmo genéticos con sorteo no dominado (NSGA), en 1993 se presenta el algoritmo genético multi-Objetivo (MOGA), en el que el rango de un determinado individuo se corresponde con el número de individuos en la población actual por el cual es dominado, en 1994 se presenta el algoritmo genético nichos de Pareto (NPGA), utiliza la medida de dominancia de Pareto y el intercambio de la aptitud, en el año 2000 se implementa un operador de selección que crea una piscina de acoplamiento mediante la combinación de la matriz y las poblaciones de descendencia y la selección de la mejor, con respecto a la aptitud y la propagación, para N soluciones mediante la modificación del algoritmo genéticos con sorteo no dominado (NSGA-II), en 1999 se introduce el SPEA, algoritmo evolutivo con intensidad de Pareto, el cual utiliza un archivo que contiene las soluciones no dominadas y en cada generación, los individuos no dominados se copian en el conjunto no dominados externo; para este año 1999 se presenta la estrategia de evolución con archivo de Pareto, trabajando el algoritmo con una única matriz que genera una sola cría; para el año 2000 se introduce un nuevo algoritmo de selección basado en la envoltura de Pareto (PESA), evolutivo multiobjetivo en el que la selección y el mantenimiento de la diversidad son controlados a través de un esquema basado en la hiper cuadrícula simple; para el 2001, se emplea un nuevo método de selección sobre el algoritmo de selección basado en la envoltura de Pareto (PESA-II), la aptitud selectiva se asigna a los hiper cajas en el espacio objetivo que actualmente están ocupadas por al menos un individuo en la aproximación actual a la frontera de Pareto y para este mismo año se presenta la modificación al algoritmo genético nichos de Pareto II (NPGA-II), mejorando la selección por torneo, en este mismo año se presenta el *Strength Pareto Evolutionary*

Algorithm II (SPEA-II) que elimina los puntos débiles de su predecesor (SPEA) y utiliza un esquema de asignación de aptitud mejorada, que tiene para cada individuo en cuenta el número de individuos que domina y que contempla una técnica de estimación del vecino y un nuevo método de truncamiento archivo garantiza la preservación de soluciones de contorno.

La revisión de la literatura en relación con la aplicación de los algoritmos evolutivos multiobjetivo (MOEA) en la selección de la cartera moderna, se realiza sobre 31 artículos; partiendo en 1993 con de la adopción del modelo de Markowitz, con miras a maximizar la rentabilidad y minimizando el riesgo; seguido en 1996 con la combinación lineal de los pesos para la selección de la cartera basado en el modelo de Markowitz, para 1997 se utilizan un algoritmo genético con una función lineal de agregación que combina los dos objetivos en un solo valor escalar, y en el que los pesos son variados con el fin de generar diferentes soluciones no dominadas; para el año 2000, se consideran que el problema de encontrar la frontera eficiente asociado con el modelo estándar de optimización de la cartera media-varianza; en el 2001 se consideran una variación del modelo de media-varianza con los costos fijos de transacción y un monto de transacción mínimo; en el año 2002 se examina el modelo de media-varianza, pero aplica restricciones adicionales sobre la cardinalidad de la cartera y de la cantidad de acciones individuales; para el 2003 se consideran que el modelo de media-varianza clásica enriquecida con limitaciones realistas adicionales, para este mismo año se sugiere un algoritmo de búsqueda local híbrida que combina principios de recocido simulado y estrategias evolutivas para la selección de la cartera cuando hay restricciones en el número de los diferentes activos en la cartera y no negatividad de los pesos de los activos; en el 2004 se proponen un modelo para la optimización de la cartera que se extiende el modelo de media-varianza de Markowitz sobre la base de la cooperación utilizando cinco objetivos específicos relacionados con el riesgo y el retorno y

permiten la consideración de las preferencias individuales a través de la construcción de la función de utilidad específica que toma decisiones; en el 2004 se proponen un procedimiento de dos fases que primero identifica el espacio de la solución de todas las carteras eficientes y luego permite una exploración interactiva de ese espacio, en este año se investiga el impacto de los diferentes operadores de cruce para un algoritmo evolutivo de valor real en el problema de selección de la cartera limitada basada en el modelo de media-varianza de Markowitz; en el 2005 se proponen un enfoque de optimización multiobjetivo híbrido que combina algoritmos evolutivos con la programación lineal para la optimización de la cartera de inversiones, en este año se aplican tres técnicas de optimización conocidos: búsqueda en los codiciosos, recocido simulado y optimización de colonia de hormigas para la solución del problema de optimización de la cartera; para el 2007 se propone el uso de algoritmos heurísticos para la decisión financiera de problemas de optimización debido a la complejidad y la no convexidad de problemas de ingeniería financiera, examinando el problema de riesgo-retorno optimización de la cartera y resolver con un enfoque computación evolutiva; para el 2007 se aplican diversas técnicas de algoritmos genéticos multiobjetivo para resolver la optimización de la cartera con algunas limitaciones realistas, como las restricciones de cardinalidad y lote limitaciones, en este mismo año se amplia para manejar diversas limitaciones realistas como piso y techo de limitación y restricción de cardinalidad; para el 2009 se presentan un modelo límite activos Markowitz (LAM), como lo llaman, con la introducción de restricciones de cantidad y cardinalidad, el modelo es un problema de programación entera mixta cuadrática, este mismo año se proponen para integrar un algoritmo conjunto activo optimizada para la selección de carteras en un algoritmo evolutivo multiobjetivo (MOEA), también en este mismo año se introduce el enfoque heurístico para problemas de optimización de cartera en diferentes medidas de riesgo mediante el empleo de algoritmos

genéticos (GA) y comparar su rendimiento del modelo de media-varianza en cardinalidad limitado frontera eficiente, se consideran igualmente un modelo de selección de la cartera que se basa en la cartera problema de selección de Markowitz incluyendo tres restricciones: lotes de transacción mínimo, restricciones de cardinalidad y capitalización bursátil; para el 2010 se considera la selección de la cartera como un problema de optimización de triple objetivo con el fin de encontrar soluciones de compromiso entre el riesgo, el retorno y el número de valores en la cartera, en este año se considera el problema de la estructuración de una cartera que supera a un índice de referencia, suponiendo que las restricciones sobre el número total de los activos negociables, también se considera que la optimización con enjambre de partículas (PSO) para resolver el problema de optimización de la cartera con restricción de cardinalidad; para el 2011 se aplican un algoritmo de relación genética (GRA) con un operador, llamado mutación guiada para resolver problemas de optimización portafolio a gran escala; terminando con propuesta del 2011 basada en la optimización de la cartera bi-objetivo de minimizar el riesgo y maximizar el rendimiento esperado.

Es necesario incluir algunos aspectos relevantes a tener en cuenta, como lo son: 199 autores han tratado el tema en las publicaciones presentadas hasta el 2011, las contribuciones de las diferentes instituciones de los países para el estudio, muestra a Estados Unidos y el Reino Unido con 32 y 30 instituciones respectivamente, Colombia participa con una; el número de objetivos utilizados por los estudiosos de la gestión de la cartera en mayor cuando se tiene dos, para cuatro y cinco objetivos el porcentaje es muy bajo; los objetivos más populares en la formulación de MOEA para los modelos de gestión de cartera es la media y la varianza, siendo esta última la medida de riesgo más popular, en relación con la cantidad de restricciones se prefieren aplicar dos restricciones para cuatro y cinco el porcentaje es mínimo; las limitaciones más populares en la

formulación para los modelos de gestión de cartera son las restricciones de cardinalidad, los límites inferior y superior y los costos de transacción.

Otra propuesta es la presentada por Gutiérrez (Mauricio Gutierrez Urzúa, 2014), es una investigación que muestra la optimización de portafolios accionarios mediante micro algoritmos genéticos, que resuelvan el modelo de selección de inversiones planteado por Markowitz, como una optimización multi-objetivo, en donde se maximiza la rentabilidad y se minimiza el riesgo, lo que implica, realizar una negociación entre ambos objetivos y buscar soluciones óptimas. La resolución de este problema requiere de un algoritmo genético para optimización multi-objetivo basado en óptimos de Pareto. Los resultados obtenidos señalan que esta aplicación es más eficiente que otros procesos de similares características tales como *Non-dominated Sorting Genetic Algorithm II (NSGA II)* y *Pareto Archive Evolution Strategy (PAES)*, pero debido al período de tiempo y las características del mercado local, su poder de predicción es bajo.

La metodología propuesta, se fundamenta en el problema que se encuentra en contemplado dentro de la teoría de selección de portafolios de Markowitz; conservando las características propias, tanto en su definición como en su solución; logrando el modelado matemático, a partir de los cuales se establecen los requerimientos para obtener buenas soluciones.

Las características del problema, se inicia con la selección de la frontera eficiente, es un problema de optimización multi-objetivo (POM) que se caracteriza por la variedad de activos financieros, y los porcentajes de riqueza invertida en cada uno de ellos; lo cual hace que el espacio de búsqueda formado por sus posibles combinaciones crezca de manera exponencial.

El problema establecido por Markowitz plantea los siguientes requerimientos:

- Dado un conjunto de activos financieros, mostrar una perspectiva de las mejores combinaciones posibles de rentabilidad y riesgo.

- Dado un nivel de riesgo, obtener la máxima rentabilidad.

- Dado un nivel de rentabilidad, obtener el mínimo riesgo.

Como propuesta de solución, se propone la construcción de una aplicación que por medio de un algoritmo evolutivo para optimización multi-objetivo permita la resolución del problema de media-varianza. La técnica a utilizar es el micro algoritmo genético para optimización multi-objetivo. La solución se fundamenta, en que el problema de Markowitz puede ser planteado de una forma multi-objetivo; que existen grandes similitudes en conceptos fundamentales entre la teoría de portafolio y la optimización multi-objetivo. En la teoría de portafolio se introdujo el concepto de dominación que establece que es preferible la cartera con menor riesgo; y de la misma manera para cada clase de riesgo es preferible la inversión con mayor rentabilidad, tipo Pareto. El Frente de Pareto, que representa el conjunto de los mejores valores para las funciones objetivo que se pretenden optimizar en distintas combinaciones de cada una de ellas, lo cual corresponde al concepto de frontera eficiente que es el conjunto de valores óptimos para el problema de media-varianza, ya que está compuesto por los portafolios con las rentabilidades más altas para cada nivel de riesgo. La computación evolutiva y en particular los algoritmos genéticos son métodos de búsqueda y optimización. Su efectividad y eficiencia está más que probada a través de diversas experiencias publicadas en la literatura referente a investigación de operaciones y ciencias de la computación.

La estructura de los datos es importante, en el desarrollo de la solución a plantear, esto es que es necesario contar con un volumen de datos que abarque un período importante de años y se requiere que estos datos se presenten de manera uniforme. En esta propuesta se trabajó con las

acciones que se transan en la bolsa de Valores de Santiago de Chile y se definió un período de tiempo de 10 años (1996-2005), 7 años para evaluar el comportamiento de los portafolios y 3 años para evaluar el valor predictivo del modelo. Se consideró una periodicidad de dos meses, quedando setenta y cuatro activos para realizar el ejercicio. Se realizaron dos procesos, uno con una cartera sin restricciones de cardinalidad y otra con una cartera compuesta por 12 activos Evans y Archer (1968).

Para la consecución de la solución son definidas las etapas del proceso, iniciando por la definición de funciones objetivos, diseño del cromosoma, propuesta del problema de mochila o similitudes a Knapsack, generación de la heurística de búsqueda local, diseño de los parámetros del algoritmo y obtención de los resultados.

Las conclusiones a que llegan los autores, establecen: en primera instancia que el desempeño de esta aplicación experimentó un notable progreso en la etapa de ensayo, en la cual el algoritmo fue probado en distintas situaciones, las cuales generaron algunas modificaciones del modelo presentado como diseño inicial, segundo de las mejoras realizadas, gran parte fueron producto de la observación empírica, motivada por la significancia de los cambios producidos, y también por la necesidad de reducción del tiempo de ejecución del algoritmo y tercero el resultado arrojó un rendimiento superior del micro-AG comparándolo con otros algoritmos, ya que en varias de las pruebas realizadas presentó un mejor rendimiento que otros más estudiados en el tiempo como el *Non-dominated Sorting Genetic Algorithm II (NSGA II)* y *Pareto Archive Evolution Strategy (PAES)*. Cuando se realizan pruebas con funciones objetivo que tienen restricciones, el micro algoritmo genético presente su rendimiento más bajo.

Una propuesta que presenta un enfoque heurístico para problemas de optimización de cartera en diferentes medidas de riesgo mediante el empleo de algoritmos genéticos (GA) y compara su

rentabilidad mediante el modelo de media-varianza en cardinalidad con frontera eficiente limitada, es la que se establece Chang (Tun-Jen Chang, Sep 2009), en su artículo *Portfolio optimization problems in different risk measures using genetic algorithm*; para lograr este objetivo, se recogieron tres medidas distintas de riesgo con base a media-varianza de Markowitz; semi-varianza, desviación media absoluta y contrarias a la asimetría. Se demuestra que estos problemas de optimización de carteras ahora pueden ser resueltos por el algoritmo genético si la media-varianza, la semi-varianza, la desviación media absoluta y la varianza de la asimetría se utilizan como las medidas de riesgo. La solidez del método heurístico es verificada por tres conjuntos de datos recopilados de los principales mercados financieros.

Los resultados empíricos muestran también que los inversores deberían incluir sólo un tercio del total de activos en la cartera que supera a los contenidos más activos.

La propuesta de optimización de la cartera en este caso, se base en la implementación del método de uso del algoritmo genético, partiendo de la inicialización de una población de tamaño 100. Los padres fueron elegidos por selección por torneo binario que funciona mediante la formación de dos grupos de individuos, cada uno formado por dos individuos extraídos de la población al azar. Los individuos con la mejor condición física, uno de cada una de las dos piscinas del torneo, se eligen para ser padres.

Establecido esto, se procede a la evaluación de la función objetivo de *fitness*, utilizando la función objetivo de media-varianza. Se continúa con la reproducción, cruce y mutación, modificando los operadores genéticos; los niños son generados por cruce uniforme. A sabiendas que en el cruce uniforme dos padres tienen un solo hijo. Si un activo i está presente tanto en los padres está presente en el niño (con un valor asociado s_i elegido al azar de uno u otro padre). Si un activo i está presente en sólo uno de los padres tiene probabilidad 0,5 de estar presente en el

niño. Seguidamente se hace el reemplazo utilizando la estrategia de reemplazo de la población en estado de equilibrio. Con esta estrategia cada nuevo niño se coloca en la población tan pronto como se genera. Se eligió para sustituir al miembro de la población con el peor valor de la función objetivo. La terminación o criterio de finalización se dan con respecto a todos los resultados computacionales reportados y con respecto al número de iteraciones utilizando 1,000 N para nuestra heurística del algoritmo genético. Estos valores significan que la heurística evalúa exactamente 1000 N soluciones.

Los resultados obtenidos de la investigación, corresponden al cálculo en diferentes medidas de riesgo en virtud de las restricciones de cardinalidad. Utilizando datos diarios históricos recogidos en el HANG SENG, FTSE y S & P 100 con los datos de precios de 33, 93 y 99 activos respectivamente entre enero 2004 y diciembre de 2006. La restricción de cardinalidad K se ajusta de 10 a 90 se incrementó en 10 cada vez. Resolvieron el problema utilizando el algoritmo genético heurístico codificado en C++ y se ejecutó en un ordenador personal.

Las conclusiones a que llegan los autores, establecen: en primera instancia que el algoritmo genético es robusto para resolver problemas de programación no lineal y entera mixta y eficaz para la solución de los problemas de optimización de cartera en diferentes medidas de riesgo; de igual forma establecen que los algoritmos genéticos presentan la ventaja importante sobre otros métodos de búsqueda exactos es su flexibilidad y su capacidad para obtener fácilmente una buena solución a un problema en el que los otros métodos deterministas no pueden lograr la optimalidad de una manera fácil; en el caso particular analizado y solucionado, la eficiencia lograda usando el algoritmo verificó que los inversionistas no deben considerar K valores por encima de un tercio del total de activos, ya que son, obviamente, dominados por los que tienen relativamente menos K valores; siendo un instrumento eficiente y conveniente para los inversores.

En términos de número de activos mantenidos en la cartera, la investigación ofrece un hecho claro que un pequeño tamaño de la cartera podría tener un mejor desempeño que los de uno más grande.

Bajo la propuesta de Aranha (Claus Aranha, 2088), en su artículo de investigación: *Modelling cost into a genetic algorithm-based portfolio optimization system by seeding and objective sharing*, utilizan la distancia de Euclides entre la selección de la cartera en dos períodos consecutivos como medida del costo, y el método de reparto objetivo de equilibrar los objetivos de maximizar la rentabilidad y minimizar la distancia en el tiempo. También mejora el método del algoritmo genético mediante la adición de material genético de carreras anteriores en la nueva población (siembra). Experimentaron el método a los datos mensuales históricos de los índices NASDAQ y el Nikkei, y se obtuvo un resultado mejor que el algoritmo genético puro, derrotando el índice en condiciones de mercado sin burbujas.

Aquí el recurso es el capital disponible para la inversión, los procesos son los diferentes activos que podemos invertir en, y el objetivo es el retorno de esta inversión.

El problema lo centran en la generación de una cartera de inversiones y se describe un algoritmo genético, como un punto de partida para los cambios necesarios para agregar el costo como una nueva función objetivo para el sistema; el sistema se basa en un modelo con los supuestos de que el volumen de operaciones no se tomará en cuenta y se supone que el inversor que se trate sólo con el rendimiento y el riesgo de la cartera.

La representación del portafolio se hace como una cartera por conformada por dos matrices, el índice (I) y los pesos (w), la optimización del portafolio a través del algoritmo genético, está dada por el genoma de cada individuo, la estrategia evolutiva fue la siguiente: Utilizaron la estrategia de la élite, donde los mejores individuos P-élite de una generación se copian en la siguiente

generación. Los individuos restantes en la nueva generación son entonces creados por cruce con una cierta probabilidad. La selección se realiza por una estrategia de torneo determinista, donde k individuos de la antigua generación son recogidos al azar, y los más aptos entre estos individuos k se elige para generar descendencia.

Se introduce a esta investigación una de las técnicas para generar una estrategia de cartera consistente en el tiempo se siembra. Al crear la nueva población para optimizar la selección de la cartera durante un período de tiempo i , introducen algunos individuos de la población resultado de período de tiempo $i - 1$. Estos nuevos miembros se copian en la primera generación de la nueva población, y de ahí en acto, como si fueran normales, generado aleatoriamente individuos.

En la práctica, esta técnica genera un sesgo en la búsqueda evolutiva hacia la región del espacio de búsqueda que contenía la solución en el período de tiempo anterior. A menos que los valores de retorno cambian muy drásticamente, los individuos sin semillas tendrán un valor de aptitud ligeramente más alto que la mayoría generados aleatoriamente individuos, y se reproducen más, lo que lleva a la población a enfocar su búsqueda en la zona de la ganadora anterior. De igual forma adicionan un parámetro de tamaño ligado al algoritmo; su valor indica el número de individuos de la población anterior que se copian en la nueva población.

Se termina con la segunda técnica introducida al algoritmo con el fin de mejorar la optimización de la cartera en el tiempo compartido; siendo esta la distancia euclidiana entre el individuo (solución actual), y la mejor solución anterior. La distancia euclidea entre dos soluciones aquí se calcula de la forma habitual: donde w_i es el peso del índice de orden i en la cartera, y l_i es el valor 0 o 1, que indica la presencia o ausencia de ese índice en particular en la cartera; con la obtención de esta nueva medición objetiva, se tiene que cambiar el proceso evolutivo a tenerla en cuenta. En cada generación, una de las dos medidas de acondicionamiento físico se utiliza para

evaluar la población, con una probabilidad dada por los parámetros $\rho > 1$. La medida de la aptitud principal, Sharpe, se elige con PSR probabilidad = $1 - (1/\rho)$, y la medida de la aptitud secundaria.

Para la experimentación se utilizaron los valores para los parámetros de evolución, que se encontraron para dar las tasas de convergencia aceptables. La diferencia entre las generaciones y tamaño de la población es debido al tamaño de datos mayor del conjunto de datos NIKKEI. El conjunto de datos estaba conformado por dos grupos de activos pertenecientes a índices conocidos. Ambos conjuntos de datos son muy grandes, lo que lleva al difícil problema de optimización de cientos de variables valores reales a la vez.

El objetivo fue ver si se podía usar el sistema propuesto para construir carteras que varían constantemente los índices durante un largo período de tiempo. Los valores de las acciones de estos activos fueron obtenidos de fuentes públicas, y la información de retorno se calcula a partir de los precios mensuales de cierre modificados con fórmula logarítmica que devuelve el precio del activo en el momento k y r_k es el retorno en el instante k .

El primer conjunto de datos es un subconjunto del índice NASDAQ 100, con 89 activos, de los 100 que componen el índice, que han estado en el mercado durante más de 72 meses (el periodo de 6 años a partir de Nov/2000 a Oct/2006), estando compuesto por una amplia variedad de industrias de la tecnología, con niveles muy dispersos de madurez y el comportamiento económico, con un gran número de activos, se convierte en difícil de aplicar técnicas de balanceo tradicionales.

El segundo conjunto de datos es un subconjunto del índice NIKKEI 225, con 205 activos, de los 225 que componen el índice, que han estado en el mercado durante el periodo de 106 meses (de Jan/1998 a DIC/2006), se compone de compañías en muchos diferentes campos de la bolsa japonesa, hay una amplia variedad de activos en este grupo que en el conjunto de datos NASDAQ - Que van desde la industria tecnológica hasta las empresas pesqueras.

Para ambos conjuntos de datos, utilizaron los últimos 53 meses del período como los datos de la evaluación; el parámetro de longitud indica cómo se tomarán en consideración muchos meses antes de la fecha actual en el cálculo de la rentabilidad esperada, probaron un rango de 3 a 30 meses, tanto en los conjuntos de datos del Nasdaq y el Nikkei, que ejecuta el algoritmo genético simple para decidir la mejor longitud de conocimiento para estos experimentos.

Establecen de igual manera una técnica de población de siembra introduciendo un nuevo parámetro el algoritmo genético, el tamaño de la herencia, que determina el número de individuos de la genética anterior de búsqueda que debe ser incluido en la nueva población; a medida que el número de activos disponibles para una cartera aumenta, el espacio de búsqueda para el portafolio óptimo también aumenta exponencialmente, haciendo el problema más difícil. La técnica de población siembra de acuerdo con García (García Sánchez, 2017), es una heurística que dice que "la última soluciones de tamaño de herencia es probablemente lo suficientemente buena para ser utilizado de nuevo", proporcionando una ventaja inicial para la nueva población. La nueva población comenzará su búsqueda donde la última población fue encontrada para tener éxito.

Para comprobar esta hipótesis, probaron un tamaño herencia de 0 a 30 % del tamaño de la población, tanto para datos NIKKEI y datos NASDAQ, y los resultados mostraron que la hipótesis inicial no es del todo correcta. Por tanto, en los datos NIKKEI y el NASDAQ, hay una diferencia significativa entre herencia de tamaño=0 y herencia de valor > 0 .

La ganancia media acumulada y ratio de Sharpe es mucho más bajo cuando no se utiliza la siembra. Sin embargo, cuando la siembra se utiliza, el efecto del valor del parámetro de la herencia estabiliza los resultados rápidamente; orientan igualmente un análisis de participación del objetivo, el cual introduce la probabilidad dada por los parámetros (POS) de parámetros para el algoritmo genético. Su valor determina la velocidad a la que uno de los dos objetivos será elegido para cada

generación. Esperando que un valor inferior dé lugar a una estrategia de cartera con las distancias y los retornos más pequeños, y que a medida que elevamos este parámetro, se pueden encontrar diferentes grados de compromiso entre las dos medidas hasta que se encuentre una solución aceptable.

Para el conjunto de datos NASDAQ, cuando Pos es 2, se tiene la mayor pérdida y la distancia más pequeña. A medida que se eleva el valor de Pos, se obtienen ganancias más pequeñas de Sharpe para las pérdidas más grandes en la distancia; los resultados reflejaron las expectativas dadas, se pudo ver en la compensación entre Sharpe y media distancia, a medida que aumentamos Pos, los resultados para el conjunto de datos NIKKEI fueron similares.

A partir de estos resultados concretos, observaron que al cambiar el valor de 1 a 2 dará lugar a un gran aumento en lugar de rentabilidad para un pequeño aumento en la diferencia. Después de eso, el aumento del valor del parámetro aumentará la diferencia entre las posiciones para menores incrementos en los rendimientos. Este tipo de información puede permitir que el analista de sintonizar el algoritmo de acuerdo con su estructura de costos en particular, mediante el equilibrio de las dos metas.

Con el fin de verificar la optimalidad del modelo propuesto, se hizo una comparación con el algoritmo genético simple y los valores históricos de los índices. El índice valor representa la evolución media de los activos que lo componen; de tal forma que se eligieron 5 combinaciones de las dos técnicas propuestas, y funcionó en el NASDAQ y el NIKKEI, con un conjunto de 30 datos de cada uno con diferentes semillas RNG.

Una vez probado el método con los datos históricos, se confirmó que pueden ser utilizados para reducir la distancia entre las posiciones mantenidas en el tiempo sin reducir el valor esperado demasiado. Mediante la manipulación de los parámetros del método de objetivo compartido, se

consiguieron estrategias con diferentes énfasis minimizando la distancia o la maximización de los retornos. Con este trabajo, se propuso una extensión al método algoritmos genéticos de optimización de la cartera, con el fin de añadir una medida de los costos de transacción en el modelo. Se tomó el modelo del costo como un objetivo de minimización de la distancia de la cartera a través del tiempo, y sugieren dos mecanismos para lograr esta reducción de costos: siembra y objetivo con recursos compartidos.

Un modelo de optimización de la cartera basado en el problema de selección de Markowitz incluyendo tres de las limitaciones más importantes, así: lotes mínimos de transacción, las restricciones de cardinalidad y el mercado de capitalización (que se propone en esta investigación por primera vez como un obstáculo para el modelo de Markowitz), es el que se presenta por parte de Soleimani (Hames Soleimani, 2009), en su artículo: *Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm*, considerado como un modelo ampliado; indispensable mencionar que ningún estudio ha propuesto y resuelto este modelo ampliado nunca. Se propone resolverlo a través de programación lineal entera mixta (NP-Hard) utilizando un algoritmo genético.

El modelo propuesto se basa en la media y la varianza del modelo selección carteras de Markowitz que no considera las situaciones reales del mercado como restricción cardinalidad y de transacción mínimo lotes.

El modelo ampliado se formula, con función objetivo que determina el riesgo a minimizar, asegura la consecución del mínimo esperado de retorno esperado al inversor, considera la limitación de presupuesto, la limitación práctica de lotes mínimos de transacción, con el fin de extender el modelo para cubrir las restricciones de cardinalidad, se limita la proporción del presupuesto que se invierte en la seguridad.

En el modelo propuesto, la capitalización del sector se considera por primera vez como un obstáculo para que la seguridad en lo seleccionado sea más razonable y segura.

En cuanto a la capitalización de mercado, se determina si se seleccionan algunos valores en los diferentes sectores, el peso total de cada acción seleccionada debe estar relacionado con su capitalización en el sector. En términos generales, el sector con mayor capitalización debe tener más proporción en la cartera definitiva, si está activada. Por lo tanto, es un punto importante que esta limitación se desactiva cuando no hay seguridad se elige de un sector.

El planteamiento que convencionalmente los problemas de selección de cartera se resuelven con modelos de programación cuadrática o lineal es abordado por Chun Lin (Chang-Chun Lin, Feb 2008), en su artículo: *Genetic Algorithms for Portfolio Selection Problems with Minimum Transaction Lots*, sin embargo, las soluciones obtenidas por estos métodos están en números reales y difíciles de implementar debido a que cada activo por lo general tiene su lote mínimo de transacción. Métodos considerando lotes mínimos de transacción se han desarrollado sobre la base de algunos modelos de optimización de carteras lineales. Sin embargo, ningún estudio ha investigado alguna vez el problema de lote mínimo de transacciones en la optimización de la cartera basada en el modelo de Markowitz, que es probablemente también el más conocido y ampliamente utilizado. Basado en el modelo de Markowitz, este estudio presenta tres posibles modelos para problemas de selección cartera con lotes mínimos de transacción y legados correspondientes algoritmos genéticos para obtener las soluciones. Los resultados del estudio muestran de forma empírica que las carteras que se obtuvieron utilizando los algoritmos propuestos están muy cerca de la frontera eficiente, lo que indica que el método propuesto puede obtener casi óptima y también soluciones prácticamente viables al problema de selección de cartera en un tiempo corto aceptable.

El modelo propuesto se basa en un enfoque de toma de decisiones multiobjetivo difuso, siendo muy recomendado debido a su capacidad de adaptación y la simplicidad.

Teniendo en cuenta según la propuesta en el artículo que la selección de cartera y los lotes mínimos de transacción, originalmente son un problema multi-objetivo de toma de decisiones que intenta maximizar la tasa de retorno y minimizar el riesgo de la cartera de forma simultánea; en la práctica, el problema es a menudo degeneró a un solo objetivo mediante la introducción de una estructura de preferencia a un compromiso entre los objetivos, o simplemente la optimización de uno de los dos objetivos mientras que delimita el restante, sin embargo, en la práctica, cada activo tiene su lote transacción mínima que debe ser tomado en consideración en la búsqueda de las carteras de riesgo mínimo. Las soluciones a los problemas de selección de la cartera por lo tanto deben ser enteros y las soluciones reales podrían ser difíciles de implementar. Como consecuencia, los modelos de selección de cartera necesitan ser modificado para tener en cuenta los lotes mínimos de transacción.

El estudio presentado utilizó datos de fondos de inversión de Taiwán desde el año 1997 hasta 2000 para poner a prueba los modelos propuestos a partir del modelo modificado de Markowitz, siendo estos una minimización de la distancia entre el objetivo y carteras obtenidas, y uno derivado de un enfoque de toma de decisión multi-objetivo difusa y el algoritmo genético. Se utilizaron las tasas mensuales de rendimiento para determinar las tasas medias de rentabilidad de los fondos de inversión y las covarianzas de retorno entre los activos en cada año. Las tasas mensuales de retorno se utilizan en lugar de los semanales porque horizonte mensual es más probable que sea el horizonte de inversión real que horizonte semanal cuando se invierte en fondos mutuos. Para medir y comparar la eficacia de los modelos propuestos, se seleccionan una serie de carteras objetivo repartidas en las fronteras eficientes para el algoritmo genético a abordar.

Los ajustes a los parámetros del algoritmo genético propuesto se dan en los datos de cada año, el número de la población y la longitud del cromosoma, las tasas de cruce y mutación, el número de iteraciones, el presupuesto y la unidad de precio; la desviación entre las carteras que se obtuvieron y las carteras objetivo se miden en términos de la distancia euclídea.

De igual forma el estudio propone modelos de decisión para los problemas de selección de cartera con lotes mínimos de transacción y utiliza algoritmos genéticos para resolver los modelos.

Los resultados de los estudios empíricos muestran que los algoritmos genéticos para estos modelos se pueden obtener casi óptimos en un tiempo razonablemente corto. Las soluciones obtenidas no sólo son aplicables en la práctica, sino que también muestran una alta eficiencia media-varianza. El modelo que se incorpora, recomienda un enfoque de toma de decisiones multiobjetivo difusa debido a su capacidad de adaptación y la simplicidad.

La adopción de un algoritmo genético como técnica de análisis, es abordado por Chung Fu (Tak-Chung Fu, Dic 2013), examinando dos aplicaciones diferentes de los algoritmos genéticos en la gestión de carteras; una se usa para determinar el ajuste de los parámetros de optimización de los diferentes indicadores técnicos y otro para la ponderación de la cartera. Diferentes algoritmos y el uso de diferentes números de los indicadores técnicos son evaluados en diferentes situaciones económicas. El algoritmo genético muestra su poder sobre la optimización de las diferentes tareas en la gestión de carteras.

Su fundamentación surge en que el número de personas que participan en el mercado bursátil crece rápidamente. Esto genera las necesidades de desarrollo de las herramientas para los inversores a gestionar sus carteras de inversión. Es difícil para los inversores para así distribuir sus acciones en la mano y que los analistas técnicos pueden emplear modelos y reglas de comercio, especialmente sobre la base de precio y de volumen transformaciones, tales como el índice de

fuerza relativa (RSI) y medias móviles (MA), cada indicador técnico tiene su propia característica. Sin embargo, es difícil ajustar los parámetros correspondientes de diferentes indicadores debido a su comportamiento único. Por lo tanto, una estrategia de optimización para ajustar los parámetros de los diferentes indicadores es necesaria.

Los objetivos de esta investigación son para hacer frente a los problemas que se presentan en la asignación incorrecta de la inversión que afecta a la rentabilidad y el análisis de la seguridad para la previsión de la dirección futura de precios de los valores a través del estudio de los datos de mercado en el pasado, el precio y sobre todo el volumen; es decir, la dificultad de asignar la ponderación de la cartera con base al uso de los algoritmos genéticos y la ambigüedad en la fijación de los parámetros de los indicadores técnicos, se propone un algoritmo genético para determinar los parámetros de los diferentes indicadores técnicos y la aplicación de este para la determinación de la ponderación de la cartera.

La optimización de los parámetros para el algoritmo y los indicadores técnicos se hacen sobre la codificación cromosoma, la evaluación de la aptitud y los operadores genéticos.

Para las estrategias de ponderación de cartera, se evaluaron tres diferentes, que incluyeron *equal weight*, el peso Markowitz y la propuesta de peso de algoritmo genético; teniendo en cuenta que para cada acción, el riesgo se puede calcular la varianza del precio de las acciones durante el período especificado, para las distintas poblaciones, su covarianza muestra la similitud de la fluctuación de precios de acciones y con el fin de medir el rendimiento de la cartera, se utiliza el ratio de Sharpe o coeficiente de medida.

Como resultado se realizan tres experimentos para evaluar las actuaciones del algoritmo genético, en resumen, aunque la rentabilidad de la optimización dada por el algoritmo genético propuesto no es tan alta como la estrategia de comprar y mantener en el mercado alcista, la

inversión realizada por la optimización ha demostrado que tienen un riesgo más bajo, ya que puede obtener beneficios en cualquier entorno de mercado (es decir, los mercados bajistas y alcistas) y diferentes sectores.

Otro enfoque principal es el uso de la programación matemática difusa, presentada por Dastkhan, Gharneh y Golmakani (Hossein Dastkhan, Naser Sams Gharneh, Hamid Reaz Golmakani, Sep 2011), bajo un enfoque de toma de decisiones multiobjetivo; de tal forma que los tomadores de decisiones puedan obtener la solución que satisfaga a su preferencia. La consideración propuesta es un modelo de max-min ponderado difuso para una cartera desviación media absoluta de problemas de selección con las características reales; para lo cual se propuso un algoritmo genético híbrido, comprobado en 75 activos de la bolsa de Nueva York (NYSE), considerados como la muestra objeto de estudio, ejecutándose el análisis para ilustrar la eficacia del modelo propuesto. Los resultados demuestran el alto rendimiento de las carteras difusas que comparan con el desempeño de las carteras nítidas y el índice S & P 500.

El modelo propuesto utiliza la formulación del problema de selección con base en la desviación absoluta como medida de riesgo, considerando el costo de la transacción, los límites en la explotación, la cardinalidad de las restricciones y se establece un modelo para evitar la concentración de la cartera en unos pocos números.

El modelo de selección de cartera multi-objetivo fuzzy, se fundamenta en la programación matemática fuzzy, donde, el principal operador difuso que aplica en la programación matemática fuzzy es el operador "max-min"; de manera que la relación entre los diversos objetivos y limitaciones en un entorno difuso es totalmente simétrica. Esto significa que todos los objetivos y restricciones difusas se negocian de forma equivalente.

Para la cartera es estable un problema de selección fuzzy, aplicando las funciones de pertenencia logística para expresar los niveles de aspiración de un inversor; con este modelo, los objetivos de rentabilidad y riesgo esperados son considerados como difusa.

El estudio empírico presentado por Dastkhan, Gharneh y Golmakani (Hossein Dastkhan, Naser Sams Gharneh, Hamid Reaz Golmakani, Sep 2011), realiza un análisis computacional compuesto de dos series de experimentos numéricos en grupos; el primero denominado "experimentos de muestra", incluye el cálculo de los portafolios óptimos para cada modelo; el segundo "fuera de las pruebas de muestra ", aplicando un entorno histórico simulado para demostrar el rendimiento; utilizando los precios semanales de 75 acciones ordinarias de bolsa de Nueva York (NYSE) en 103 semanas, desde enero de 2005 a diciembre de 2006 como en el conjunto de datos de muestra. Los precios semanales de las mismas poblaciones, desde enero 2007 hasta diciembre 2008 se utilizan para salir de las pruebas de muestra.

En el análisis de muestras, se establece que el uso de la programación matemática fuzzy como un método de toma de decisiones multi-objetivo, el inversor puede incorporar sus características subjetivas/intuitivas en la inversión. Siendo este tipo de características los elementos principales de los inversionistas; la principal ventaja de este método es que, si el inversor no está satisfecho con los resultados obtenidos, más carteras se pueden generar con la variación de los parámetros del modelo. Para poner de relieve los efectos de los pesos y los parámetros de la función de pertenencia, se consideraron tres situaciones de toma de decisiones: inversionista agresivo, inversor indiferente e inversor conservador.

La aplicación de la eficiencia de Pareto es propuesta por (He Ni, Yonggiao Wang, Dic 2013), se propone un enfoque de búsqueda heurística en la construcción de una cartera de seguimiento,

que es capaz de conseguir el retorno promedio del mercado y puede incluso superar a algunos fondos de cobertura que se gestionan de forma activa.

El modelo matemático propuesto se basa en un algoritmo genético híbrido con un mecanismo de evolución auto-adaptativo; con el fin de mejorar la eficiencia del modelo, se optimizó el algoritmo genético original mediante la aplicación de la eficiencia de Pareto como medida de la utilidad y la programación de metas para los inevitables conflictos de múltiples objetivos/intereses.

El enfoque propuesto ofrece una solución integral al problema de seguimiento de índice al considerar como muchas cuestiones prácticas como sea posible. La cartera de construcción, con un desempeño satisfactorio en los experimentos basados en CSI300, FTSE100 y los datos de HSI. Por consiguiente, la formulación propuesta de seguimiento de índice se cree que es una buena alternativa a muchas técnicas actuales.

La principal contribución de este trabajo consiste en reforzar la capacidad de aprendizaje del algoritmo genético convencional en la formulación del problema de seguimiento de índices mediante la integración de la programación por metas, medición de la eficiencia de Pareto y una serie de limitaciones prácticas.

La cartera de seguimiento se construyó mediante un algoritmo genético híbrido. El aprendizaje heurístico original está optimizado por la directriz dada por la eficiencia de Pareto y la programación por metas. La obtención de los datos se desarrolla de manera estocástica frente a los precios de las acciones, los datos de mercado de bienes fueron procesados cuidadosamente para adaptarse a los supuestos del modelo, por su parte los métodos elegidos fueron verificados para ajustar los datos; se obtuvieron un total de 1380 precios de cierre diarios de las acciones cotizadas en CSI300, 1434 de FTSE100 y 1428 de HSI.

Para la optimización multi-objetivo, se realizó una selección por torneo mediante la asignación de un valor a la calidad de un individuo con respecto a un objetivo determinado, logrando de todos los objetivos el superior, para contar con una Pareto eficiente; cuando el evento de conflicto se presenta por el gran número de objetivos se aplicó la programación por metas; siendo bastante útil cuando la función de utilidad no se puede definir de manera explícita o se generan intereses de conflicto, que son imposible de tener una representación matemática confiable de preferencia de la gente. Se utiliza la programación de metas lexicográfica, siendo una minimización secuencial, teniendo en cuenta que el objetivo principal de la cartera de seguimiento es también un seguimiento de la cartera de referencia con un costo limitado de tal forma que el rendimiento sea estable en términos de mantener una varianza bajo error de seguimiento, teniendo en cuenta el tracking error, rentabilidad acumulada y varianza del error, las limitaciones relacionadas principalmente con la cartera de seguimiento son la posición de cada población y los costos de transacción .

De igual forma se hace la asignación de la riqueza convencional, esto es que las poblaciones ponderadas fuertemente son las que tienen mayor valor de mercado, el aumento de la liquidez y el potencial de crecimiento significativo en la teoría, mientras que las empresas pequeñas o medianas escala que se ponderan a la ligera se solía tener influencia relativamente pequeña del mercado, escasa liquidez; con esto se propone primero el enfoque de búsqueda heurística: algoritmo genético, que permita encontrar la estrategia óptima de inversión sujetos a algunas restricciones dadas por las técnicas de búsqueda heurística; segundo se contempla la definición de *fitness* o el objetivo maximizar la utilidad de un inversor, tercero se realiza la selección: muestreo estocástico universal, utilizando una variante de la aptitud de selección proporcional llamada muestreo estocástico universal, ya que disminuye el sesgo y es la más eficiente, el muestreo estocástico es

una técnica utilizada en algoritmos genéticos para la selección de soluciones potencialmente útiles para la recombinación; cuarto se hace el cruzamiento con restricción de peso y recombinación intermedia, agregando elementos estocásticos en el proceso de aprendizaje, quinto se lleva a cabo la mutación u operación de cruce con elementos estocásticos, sexto se realiza el disparador de la evolución, algunos cambios ocasionales causadas por diversas razones, por ejemplo, una fusión o adquisición de una sociedad cotizada, se producen en cualquier momento, séptimo generación del cromosoma, la formación de una cartera de seguimiento candidato se representa por un cromosoma que evoluciona en cada generación. El cromosoma es consta de una serie de genes cuyos valores están hechos proporcional a la cantidad de dinero invertido las acciones componentes.

La medición del rendimiento se hace por medio de *La media de los cuadrados de error*, *el Coeficiente de Theil* y *el Test Diebold-Mariano*; con la configuración de los parámetros del modelo necesarios de la evolución en términos generales se pueden dividir en tres grupos, que son la definición de la aptitud, la selección y evolución, que incluye las operaciones de cruce y mutación.

El uso de los algoritmos genéticos, como un esquema de optimización de la cartera de gestión de fondos indexados se propone por (Kyong Joo Oh, Tae Yoon Kim, Sungky MIn, Feb 2005), siendo estos fondos una de las estrategias populares de gestión de la cartera que tiene como objetivo la búsqueda de la rentabilidad del índice de referencia, como el S & P 500 en Nueva York y en el FTSE 100 de Londres lo más fielmente posible. Esta estrategia es tomada por los gestores de fondos sobre todo cuando no están seguros acerca superando al mercado y adaptarse al rendimiento promedio. Recientemente, se observa que los rendimientos de los fondos indexados son mejores que los de muchos otros fondos de inversión de gestión activa. El objetivo principal de este trabajo es informar que el fondo de índice podría mejorar su rendimiento en gran medida

con el esquema propuesto cartera GA, que se demostró para el fondo de índice diseñado para realizar un seguimiento de Corea Stock Price Index (KOSPI) 200.

Es importante aclarar que los fondos indexados son instrumentos de inversión más populares que se utilizan en la gestión moderna de la cartera, están diseñados para imitar el comportamiento de los índices de los mercados de referencia son generalmente considerados como herramienta de inversión relativamente estable y eficiente en comparación con otros fondos de inversión y se componen de relativamente pequeño número de poblaciones.

Se propone un esquema de selección de cartera soportado en algoritmo genético para la optimización de fondos indexados, el esquema del algoritmo proporciona la selección óptima de las reservas utilizando variables de error fundamental estándar de la cartera, la cantidad de cotización promedio, y la capitalización media del mercado. Estas variables fundamentales son factores básicos bien conocidos de uso frecuente en el análisis y la predicción del mercado de valores. En términos generales, el esquema del algoritmo consta de dos pasos. En primer lugar, se seleccionan las reservas para el fondo de índice mediante el trabajo con las variables fundamentales en cada sector de la industria del índice de referencia. En segundo lugar, los pesos relativos de las acciones seleccionadas se optimizan a través del proceso del algoritmo. Se demostró que el esquema de la cartera replica eficientemente el índice de referencia con un número relativamente pequeño de las poblaciones.

El algoritmo propuesto se aplica al índice de precios de acciones de Corea (KOSPI) 200 desde enero 1999 hasta diciembre 2001. KOSPI 200 incluye 200 grandes empresas de 22 sectores de la industria, que figuran actualmente en la Bolsa de Corea. Las 200 compañías cubren el espectro general de la Bolsa de Valores y el Kospi 200 de Corea es también el índice de base del contrato de futuros del KOSPI 200, que es el contrato de futuros más activo en la Bolsa de Corea. Este

documento ofrece una breve encuesta acerca de la teoría de la cartera, los fondos de índice, y los algoritmos genéticos, discute el procedimiento detallado del sistema y presenta los resultados de experimentos empíricos. Las especificaciones del modelo incluyen como variable el beta portafolio, la cantidad de comercio y la capitalización de mercado

Algoritmo cultural

Las aplicaciones de la optimización con incontables, muchas aplicaciones en las ciencias, la economía y la industria pueden ser formuladas como procesos de optimización; los métodos evolutivos representan un área de la optimización en las ciencias de la computación y la matemática aplicada (Erik V. Cuevas Jiménez, José V. Osuna Enciso, Diego A. Oliva Navarro, Margarita A. Díaz Cortés, 2016); los últimos 10 años han sido testigo del desarrollo de numerosos enfoques evolutivos que permiten la interacción de diferentes disciplinas que incluyen estudios económicos; la mayoría de estos usan como inspiración fenómenos sociales, los cuales con cierto nivel de abstracción pueden ser considerados como modelos de optimización.

Recientemente, un número de algoritmos de cómputo evolutivo se han popularizado en la academia y en la industrial, motivados en grupos sociales; los cuales se han utilizado para solucionar problemas complejos de optimización. Algunos de estos algoritmos incluyen Optimización por enjambre de Partículas (PSO), Evolución Diferencial (DE), Búsqueda Armónica (HS), Sistemas Inmunológicos Artificiales (AIS), Optimización Basada en Electromagnetismo (EMO), Algoritmo Colonia Artificial de Abejas (ABC) y los Algoritmos Culturales propuestos por Robert G. Reynolds.

Estos algoritmos utilizan un modelo basado en una población como la base del algoritmo y solucionan problemas compartiendo la información vía la interacción social entre agentes en la población (Zezzatti, 2008).

De igual forma Zezzatti (Zezzatti, 2008), establece para los algoritmos culturales que en parte permiten que los agentes actúen recíprocamente de diversas maneras usando varias formas de información simbólica reflexivas de sistemas culturales complejos.

El algoritmo cultural básico permite que los individuos se comuniquen vía un espacio compartido de creencias. El espacio compartido almacena cinco tipos básicos de información que se puedan compartir cognitivamente o simbólicamente. Es bien sabido que la escala de interacción dentro de los sistemas complejos afecta la naturaleza de la estructura que emergen de la interacción de agentes dentro de ese sistema.

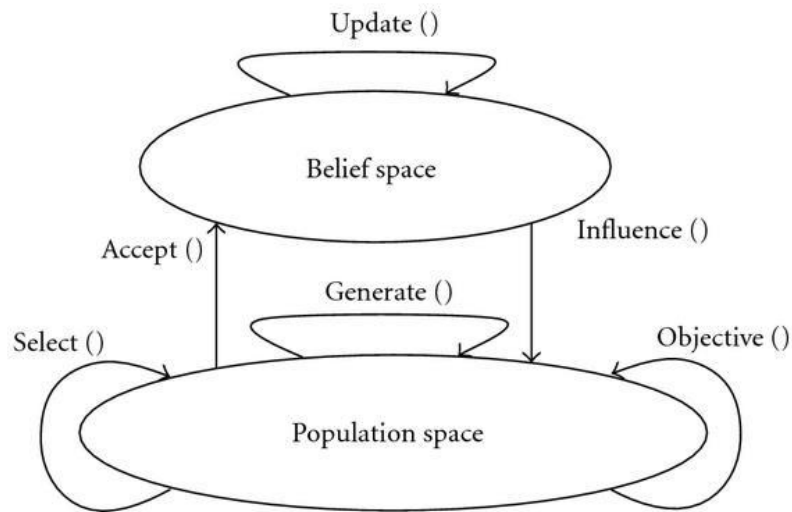
Los algoritmos culturales (ACs) son una clase de modelos computacionales derivados de la observación del proceso de evolución cultural en la naturaleza; los cuales tiene tres componentes principales:

un espacio de población,

un espacio de creencias,

y un protocolo

que describe como el conocimiento es intercambiado entre los dos primeros componentes. El espacio de población puede soportar cualquier población basada en el modelo computacional, tales como los algoritmos genéticos y la programación evolutiva. La Figura 1 muestra el diagrama conceptual básico.

Figura 1 Marco de referencia de los algoritmos culturales

Fuente: Tomado de <https://www.hindawi.com/journals/mse/2011/239743/fig1/>

Los algoritmos culturales son un sistema dual de la herencia que caracteriza la evolución en cultura humana en el nivel macro-evolutivo, que ocurre dentro del espacio de creencias, y en el nivel microevolutivo, que ocurre en el espacio de población. El conocimiento producido en la población que el espacio en el nivel microevolutivo se acepta o que se pase al espacio de creencias y que se utilice selectivamente para ajustar las estructuras de conocimiento allí. Este conocimiento se puede entonces utilizar para influenciar los cambios realizados por la población en la generación siguiente.

Lo que diferencia de los algoritmos culturales es el hecho de que usan cinco tipos de conocimiento básicos en el proceso de resolución de la solución del problema más que solamente el valor transmitido. Hay evidencia que en ciencia cognoscitiva cada uno de estos tipos de conocimiento es demostrado por varias especies animales, asumen que los sistemas sociales humanos apoyan como mínimo cada uno de estos tipos de conocimiento.

Las fuentes del conocimiento incluyen el conocimiento normativo (gamas de comportamientos aceptables), el conocimiento circunstancial (los ejemplares o las memorias de soluciones acertadas y fracasadas entre otros), el conocimiento del dominio (conocimiento de los objetos del dominio, de las relaciones de ellos, y sus interacciones), el conocimiento histórico (patrones temporales del comportamiento), y el conocimiento topográfico (patrones espaciales del comportamiento). Este conjunto de categorías es visto como completo para un dominio dado en el sentido que todo conocimiento disponible puede expresarse en términos de una combinación de una de estas clasificaciones.

Se pueden poner números arbitrarios de los conos de tamaños que varían en el terreno para producir superficies de complejidad diversa. La distribución de conos puede ser estática, dinámica, y engañosa (debido a la colocación de algunos conos ocultos en mejores áreas desde la perspectiva de los agentes). Los agentes entonces interactúan socialmente vía estas varias fuentes de conocimiento para encontrar el grado óptimo, y en el ambiente dinámico no se pierda de vista su posición que cambia en un cierto plazo. Entonces se investiga la aparición de patrones sociales en el espacio de la población y el espacio de creencias cuando el problema se soluciona con éxito.

Los algoritmos culturales se han estudiado con problemas de prueba patrón, también se han aplicado con éxito en un número diverso de áreas de aplicación tales como modelar la evolución de la agricultura, concepto de aprendizaje, optimización de la función de valor-real y reingeniería del conocimiento base para el proceso de ensamblaje de manufactura, y para modelar los sistemas de incentivo de precios basado en agentes, la distribución de elementos en un Diorama, obtención de un ranking en Eurovisión, y simulación de un modelo social en un juego inteligente y combinado con Predator/Prey Game para problemas culturales entre otros.

Bajo la propuesta de Esquivel, (Esquivel, 2017), los algoritmos culturales operan en dos espacios. Primero, el espacio de la población, como en todos los métodos de computación evolutiva, en el que se tiene 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, tales individuos podrán ser reemplazados por algunos de sus descendientes, obtenidos a partir de un conjunto de operadores aplicados a la población. El segundo espacio es el de creencias, donde se almacenarán 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.

Para unir ambos espacios se establece un protocolo de comunicación que dicta las reglas del tipo de información que se debe intercambiar entre los espacios. Por lo anterior, el pseudo-código de un algoritmo cultural, es:

P0=Generar población inicial (Siguiendo una distribución uniforme, con la restricción que la suma de las probabilidades de los portafolios sea igual a 1)

Mientras (Iteraciones < Total de Iteraciones)

F=Evaluar usando la Matriz de Markowitz (P0)

P1= Aceptar Individuos (Se utiliza un numero aleatorio de selección, sobre una población normalizada)

P1= Llevar población al espacio de creencias (Influenciar mediante métodos estocásticos P1)

P0=Reemplazo generacional (P1)

Iteraciones ++

Fin Mientras

Aceptar Individuos;

Fuente: Construcción Propia, bajo la propuesta por Robert G. Reynolds

En la técnica para optimización con restricciones, se utiliza el espacio de creencias para generar una especie de mapa del espacio de búsqueda, en el que se distinguen las regiones factible y no factible. La información se obtiene de los individuos generados con anterioridad, y se utiliza para generar nuevos individuos dentro de la región factible, y de preferencia en la frontera de la región factible y la región no factible.

Aplicación de los algoritmos culturales

La propuesta realizada por Becerra (Becerra, Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Miltiobjetivo, 2000), en su tesis, es la de conceptualizar la aplicabilidad del uso de los Algoritmos Culturales, definiéndolos como “técnicas de computación evolutiva que operan en dos espacios: el espacio de la población (común entre los algoritmos evolutivos), y el espacio de creencias, en el que se almacenan experiencias (positivas o negativas) que la población ha adquirido a lo largo del proceso de búsqueda.”.

Al igual Becerra (Becerra, Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Miltiobjetivo, 2000), comenta que sin embargo, los algoritmos culturales fueron desarrollados por Robert G. Reynolds, como un complemento a la metáfora que usan los algoritmos de computación evolutiva, que se habían concentrado en conceptos genéticos, y de selección natural. Los algoritmos culturales están basados en las teorías de algunos sociólogos y arqueólogos, que han tratado de modelar la evolución cultural.

Retomando a Esquivel (Esquivel, 2017), los algoritmos culturales operan en dos espacios. Primero, el espacio de la población, como en todos los métodos de computación evolutiva, en el que se tiene 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, tales individuos podrán ser reemplazados por algunos de sus descendientes, obtenidos a partir de un conjunto de operadores aplicados a la población. El segundo espacio es el de creencias, donde se almacenarán 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.

En la técnica para optimización con restricciones, se utiliza el espacio de creencias para generar una especie de mapa del espacio de búsqueda, en el que se distinguen las regiones factible y no factible. La información se obtiene de los individuos generados con anterioridad, y se utiliza para generar nuevos individuos dentro de la región factible, y de preferencia en la frontera de la región factible y la región no factible.

Los resultados de esta técnica se muestran competitivos frente a otros algoritmos evolutivos, requiriendo una menor cantidad de evaluaciones de la función objetivo.

Por otro lado, la técnica para optimización multiobjetivo utiliza un espacio de creencias, que también es una especie de mapa, pero en este caso en el espacio fenotípico (el espacio de las funciones objetivo). Con este espacio de creencias se tratan de identificar las zonas menos pobladas del frente de Pareto, y la influencia se ejerce sobre la selección, para conservar a los individuos que conformen un frente de Pareto mejor distribuido.

El trabajo desarrollado por Macedo, Almeida y González (Natalli Macedo Rodriguez, Carolina Paula de Almeida, Richard A. Gonzalez, 2007), en el cual se enfatiza no solamente en la

disminución de costos en los despachos para el cual se utiliza un caso con 6 unidades, el desarrollo en la fábrica termoeléctrica con tres y trece unidades generadoras de energía; sino que además señalaron que se trata de una aplicación de los algoritmos culturales con el fin de obtener resultados eficientes en el proceso de optimización de la generación de energía, y la amortiguación de contaminación en el medio ambiente.

El desarrollo de dicho trabajo generó una interacción entre un algoritmo genético, que se basó en la teoría de la evolución de las especies, y un algoritmo cultural.

Se tiene una aplicación en simulación de carrera de autos, realizada por Hernández (Hernández, 2012), la cual mediante la generación de un vector de parámetros nuevos controladores que basados en los pilares como el control difuso que se encarga de recibir las variables que definen el ambiente del simulador, procesarlas, y decidir el siguiente movimiento que será comunicado al simulador; y un algoritmo cultural que se encarga de evaluar el rendimiento de cada controlador (representado como un vector de parámetros para las funciones de pertenencia), reflejen eficazmente su funcionalidad.

Muestra una simulación multi-agente y que se define como: “un paradigma añejo donde agentes se entienden como entidades autónomas que tienen perfectamente definidas sus funciones, que interactúan entre sí y que están posicionadas en algún ambiente simulado al cual deben adaptarse, con el objetivo de resolver algún problema en conjunto”.

La aplicación de dicho caso se basa en los videojuegos de simulación y se toma como ejemplo de aplicación para este paradigma, los cuales han resultado muy atractivos porque permiten reproducir situaciones reales de algo que no está sucediendo sin poner en riesgo recursos valiosos.

El simulador de carreras de autos de código abierto TORCS toma en cuenta muchos de los aspectos de una carrera de autos real, el cual está lanzando retos constantemente a programadores

e investigadores. El problema actual que plantea el videojuego de simulación es el diseño de un control que sea capaz de adquirir las habilidades para conducir, y posteriormente memorizar y explotar las características de pistas desconocidas, contando con tiempo de simulación limitado.

El pseudocódigo del algoritmo cultural propuesto para la simulación de videojuegos, permite optimizar el control difuso y cuyo objetivo es evolucionar una población inicial que inicie sin conocimiento previo a diferencia de otros enfoques reportados en el estado del arte, que si bien utilizan menos tiempo de simulación parten de una población pre entrenada.

Tabla 2. Pseudocódigo del Algoritmo Cultural propuesto para la simulación de videojuegos

```
para i=1 popSize hacer
```

```
    Generar el i-ésimo individuo aleatoriamente
```

```
    Evaluar el i-ésimo individuo (ejecutar simulación en TORCS)
```

```
fin para
```

```
Inicializar el espacio de creencias
```

```
repetir
```

```
    para i=1 popSize hacer
```

```
        Generar el i-ésimo hijo de acuerdo a la fuente del conocimiento  
        elegida aleatoriamente
```

```
        Evaluar el i-ésimo individuo (ejecutar simulación en TORCS)
```

```
        Reemplazar el i-ésimo padre por el hijo, si el hijo es mejor
```

```
Fin para
```

Actualizar el espacio de creencias (con los individuos que pasan a la siguiente generación

hasta que el tiempo de simulación permitido haya terminado

Fuente: Guillermo Zamora Hernández, (Hernández, 2012)

La evaluación de los individuos en este problema implica la prueba de los controles generados en el simulador. El espacio de creencias en algoritmo cultural propuesto está conformado por tres fuentes:

Dos fuentes que promueven los controles más veloces y que recorren distancias largas (situacional y normativa) y una fuente de diversidad (que es la versión estándar de la evolución diferencial). La condición de paro se da al alcanzar los 500, 000 *gametics*, tiempo que se consideró suficiente para generar controles competitivos iniciando sin conocimiento alguno.

Bajo la propuesta de tesis doctoral (Becerra, Uso de Información del Dominio para Mejorar el Desempeño de un Algoritmo Evolutivo, 2003), desarrolla un nuevo algoritmo cultural para optimización multiobjetivo con o sin restricciones, un algoritmo que mejorara también el algoritmo cultural mono-objetivo a través de conocimiento la mejora principal de será la inclusión de un mecanismo de auto adaptación o de adaptación, así como el algoritmo cultural para optimización multiobjetivo se diseñará de tal forma que el espacio de creencias afecte los operadores genéticos.

El planteamiento del problema de optimización de un objetivo con restricciones puede escribirse de la siguiente manera:

Dada una función $f: \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $\Omega \neq \emptyset$, encontrar el vector $x^* = [x^*_1, x^*_2, \dots, x^*_n]^T$, tal que $x^* \in \Omega$, y que:

$$\forall x \in \Omega: f(x^*) \leq f(x)$$

Donde Ω es la *región factible*, definida por un conjunto de restricciones:

$$g_i(x) \geq 0 \quad i = 1, 2, \dots, m \quad h_i(x) = 0 \quad i = 1, 2, \dots, p$$

Este es el planteamiento del problema de minimización global. El problema de maximización es equivalente, y puede ser transformado utilizando la identidad

$$\text{máx} (f(x)) = -\text{mín} (-f(x))$$

En el problema multiobjetivo, el planteamiento es muy similar. Se requiere encontrar el vector $x^* = [x^*_1, x^*_2, \dots, x^*_n]^T$ que optimice el vector de funciones

$$f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T$$

En el vector de funciones pueden existir funciones por minimizar y funciones por maximizar. Sin embargo, suelen convertirse todas las funciones a una misma forma, ya sea maximización o minimización.

La diferencia fundamental entre los problemas mono-objetivo y multiobjetivo es la noción de “óptimo”. En el caso mono-objetivo se tiene un punto óptimo indiscutible, que tiene el mejor valor de la función objetivo, mientras que en el caso multiobjetivo suele ocurrir que las funciones objetivo se encuentren en conflicto, y para mejorar el desempeño en una de ellas se tenga que empeorar en otra.

Entre los primeros trabajos de algoritmos culturales para manejo de restricciones, está la modificación del algoritmo llamado GENOCOP, hecha por Reynolds en 1978, para incorporarle un espacio de creencias. La limitante más importante de dicho trabajo es que, tanto al algoritmo original como la versión cultural sólo pueden manejar restricciones lineales. El trabajo más reciente de manejo de restricciones con algoritmos culturales, se ha hecho con programación evolutiva (CAEP, *Cultural Algorithms with Evolutionary Programming*).

Funcionamiento del Algoritmo Cultural

El funcionamiento del Algoritmo Cultural se tratará desde la propuesta de Duarte y Pantrigo (Abraham Duarte Muñoz, Juan Jose Pantrigo Fernandez, 2007), quienes establecen:

El principio fundamental de funcionamiento de esta metaheurística es: Los Algoritmos Culturales establecen que la toma de decisiones depende de la cultura heredada, así como de las creencias de la población actual.

En las sociedades humanas, la cultura puede ser considerada como un contenedor de información pasada, en el sentido de una “memoria histórica”. La cultura es independiente de los individuos que la generaron y potencialmente accesible por todos los miembros de la sociedad. Este esquema se puede utilizar para desarrollar estrategias que permiten encontrar soluciones a un problema dado, modelando las relaciones entre los individuos, la sociedad y la cultura de dicha sociedad.

Los sistemas culturales se pueden interpretar como un mecanismo de herencia en dos niveles:

Nivel micro-evolutivo: el contenido genético para de los padres a los hijos

Nivel macro-evolutivo: el contenido cultural pasa de una generación a otra.

En cada iteración, la mejora genética de la población y las creencias adquiridas se transmiten a la siguiente generación. El algoritmo cultural utiliza esta información para guiar a los individuos, de modo que pueda distinguirse entre soluciones de alta y de baja calidad.

En el contexto de los algoritmos culturales, se produce una evolución tanto en la sociedad, compuesta por individuos, como en la cultura, compuesta por creencias. En este sentido, se

puede considerar que los algoritmos culturales están estructurados en las siguientes componentes:

Población: que evoluciona mediante los mecanismos de selección, reproducción y evaluación.

Espacio de creencias: permite guiar la búsqueda mediante el conocimiento adquirido durante la resolución del problema.

Protocolo de comunicación: describe la interacción entre la población y sus creencias; considera dos tipos de relaciones: establecimiento del conjunto de individuo (aceptables) que pueden influir en la modificación de las creencias adquiridas y el establecimiento del mecanismo de influencia de las creencias sobre los individuos.

Reynolds establece de acuerdo con Duarte y Pantrigo (Abraham Duarte Muñoz, Juan Jose Pantrigo Fernandez, 2007), que un Algoritmo Cultural viene descrito por un conjunto de ocho componentes: $CA = [P, S, Vc, f, B, Aceptar, Ajustar, Influir]$

Donde:

P es la población de individuos que componen la sociedad

S es el operador de selección

Vc es un operador de variación

f es la función objetivo

B es el espacio de creencias

Aceptar es la función de aceptación y determina el protocolo de comunicación entre el espacio de creencias y la sociedad (en el sentido sociedad → creencia).

Influir es un conjunto de funciones de influencia que afectan al operador de variación Vc. Determinan el protocolo de comunicación entre el espacio de creencias y la sociedad (en el sentido sociedad ← creencias).

Pseudocódigo del Algoritmo Cultural

```
{Xbest: TipoSolucion] = CA(N:integer; f:TipoFuncionObjetivo)
```

```
Var
```

```
  P: array [1...N] of TipoSolucion; //Soluciones con peso
```

```
  B: <Creencias>;
```

```
  g: integer; // generacion
```

```
begin
```

```
  g:=1; // Iniciar el número de generacion
```

```
  {P}:= Iniciar Población(N); //Iniciar los elementos de la población
```

```
  {B}:= Iniciar Creencias; // Iniciar las creencias de la población
```

```
  Repita
```

```
    {P}:= Evaluar(P,f); //Evaluar la calidad de los individuos
```

```
    {B}:= Ajustar(B,Aceptar(P)); // Ajustar las creencias con los  
    individuos influyentes
```

```
{P} := Modificar(P,influir(B)); // Modificar los individuos con las
creencias

g := g + 1;

{P} := Seleccionar (P,f);

until terminación

{Xbest} := SeleccionarMejor(P,f); // Seleccionar la mejor solución
encontrada

end
```

En las versiones más sencillas del algoritmo cultural, la primera componente se corresponde con una implementación de un algoritmo cualquiera, de tal forma que el mecanismo de selección S y de variación (cruce y mutación) V_c equivalen a sus correspondientes genéticos. En estos primeros modelos, el espacio de creencias contiene únicamente un operador cultural que aprende la idoneidad de usar un determinado operador genético sobre la población, de forma que la probabilidad de aplicarlo aumenta o disminuye en función del efecto producido sobre la calidad de los individuos. En este modelo, la función de aceptación es del 100%, de modo que se acepta cualquier mejora. Además, se permite que cualquier individuo pueda modificar el espacio de creencias.

En una segunda evolución de la metaheurística, se introduce en el espacio de creencias un conocimiento sobre las estructuras de alta calidad que aparecen en la población. En ese modelo, la función de aceptación es del orden del 50%, con lo que no todos los individuos

pueden modificar el espacio de creencias. La función de influencia modifica la calidad del individuo en lugar de su estructura.

En la tercera versión se generaliza el algoritmo para poder abordar problemas de optimización de funciones reales. En esta versión, la implementación del espacio de soluciones es bastante compleja.

En la cuarta y última generación del algoritmo cultural, tanto la población como el espacio de creencia utilizan estructuras de subgrafos. Éstas, se interpretan como estructuras de decisión que interactúan entre sí para encontrar una solución al problema. El espacio de creencias está formado por estructuras de subgrafo, que se caracterizan por aparecer en los individuos válidos. En este caso, la función de aceptación es del orden del 20%. La función de influencia se diseña de tal forma que se reduzca la probabilidad de que el cruce pudiera destruir sub-estructuras que aparecen en el espacio de creencias.

Composición de la aplicación

Con el fin de dar cumplimiento a los objetivos general y específicos trazados, se describe el funcionamiento de cada uno de los algoritmos que permiten modelar la optimización del portafolio de inversión.

Funcionamiento del Algoritmo Cultural

Al basarse en una estrategia evolutiva cooperativa, el proceso de adaptación del conjunto de soluciones al problema se basa en un espacio de creencias, el cual se encarga de dirigir la búsqueda hacia un óptimo global.

Pseudocódigo

P0=Generar población inicial (Siguiendo una distribución uniforme, con la restricción que la suma de las probabilidades de los portafolios sea igual a 1)

Mientras (Iteraciones < Total de Iteraciones)

F=Evaluar usando la Matriz de Markowitz (P0)

P1= Aceptar Individuos (Se utiliza un numero aleatorio de selección, sobre una población normalizada)

P1= Llevar población al espacio de creencias (Influenciar mediante métodos estocásticos P1)

P0=Reemplazo generacional (P1)

Iteraciones ++

Fin Mientras

Aceptar Individuos

La función para aceptar individuos, inicialmente ordena los individuos de mayor a menor (en el caso de maximización y viceversa en el caso de minimización), una vez ordenados los individuos

se genera un número aleatorio con distribución uniforme y este selecciona la cantidad de individuos aceptados en la nueva generación. Los que no son aceptados se realizan diferentes operaciones en el espacio de creencias.

Ejemplo

Condiciones normales

Tabla 1 Condiciones normales - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.2	0.3	0.5	0.1
0.8	0.1	0.1	0.4
0.7	0.15	0.15	0.01
0.5	0.5	0	0.09
0	0	1	0.25
0	1	0	0.15

Fuente, Elaboración propia

Vector ordenado (en caso de Maximizar)

Tabla 2 Vector ordenado – algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.8	0.1	0.1	0.4
0	0	1	0.25
0	1	0	0.15
0.5	0.5	0	0.09
0.7	0.15	0.15	0.01

Fuente, Elaboración propia

Selección

Se selecciona una posición aleatoria y todos los individuos anteriores a esa posición son aceptados (rojos), el resto son rechazados (azules);

Tabla 3 Aceptación y rechazo – algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Estado
0.8	0.1	0.1	0.4	Aceptados
0	0	1	0.25	
0	1	0	0.15	
0.5	0.5	0	0.09	Rechazados
0.7	0.15	0.15	0.01	

Fuente, Elaboración propia

Espacio de creencias;

Cuando los individuos son rechazados estos se ven influenciados mediante operadores de mutación y recombinaciones, entre los cuales se encuentran;

Permutación (la misma función del algoritmo genético).

Mutación Gaussiana (La misma del algoritmo genético).

Apuesta; Este operador modifica un individuo de la población de tal manera que una sola empresa tiene el control del portafolio.

Ejemplo;

Población en general

Tabla 4 Población General – Algoritmo Cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Estado
0.8	0.1	0.1	0.4	Aceptados
0	0	1	0.25	
0	1	0	0.15	
0.5	0.5	0	0.09	Rechazados
0.7	0.15	0.15	0.01	

Fuente, Elaboración propia

Tomar individuos rechazados**Tabla 5** Individuos rechazados – algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Estado
0.5	0.5	0	0.09	Rechazados
0.7	0.15	0.15	0.01	

Fuente, Elaboración propia

Se aplica la función sobre un individuo de la población y quedaría de la siguiente forma;

Tabla 6 Aplicación de función - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.5	0.5	0	0.09
1	0	0	0.01

Fuente, Elaboración propia

Y este individuo generado se incorpora de nuevo a la población

Tabla 7 Nueva población - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.8	0.1	0.1	0.4
0	0	1	0.25
0	1	0	0.15
0.5	0.5	0	0.09
1	0	0	0.01

Fuente, Elaboración propia

Renacer; Esta función se encarga de cambiar un individuo rechazado por uno nuevo.

Ejemplo;

Tomar individuos Rechazados**Tabla 8** Individuos Rechazados - Algoritmo Cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Estado
0.5	0.5	0	0.09	Rechazados
0.7	0.15	0.15	0.01	

Fuente, Elaboración propia

Aplicar modificador sobre un individuo de la población y quedaría de la siguiente forma.

Tabla 9 Aplicación modificador - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.25	0.25	0.5	0.09
0.7	0.15	0.15	0.01

Fuente, Elaboración propia

Flip; Esta función invierte los elementos que componen el individuo como se muestra abajo.

Población Rechazada

Tabla 10 Población rechazada - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Estado
0.5	0.5	0	0.09	Rechazados
0.7	0.15	0.15	0.01	

Fuente, Elaboración Propia

Aplicando operador en un individuo

Tabla 11 Aplicación operador - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>
0.5	0.5	0	0.09
0.15	0.15	0.7	0.01

Fuente, Elaboración propia

No hacer Nada; Este operador deja el individuo tal y como estaba.

Selección del operador e influencia sobre la población

La modificación de los individuos se realiza de manera ordenada y descendente, es decir, se toman los individuos rechazados y a cada uno de ellos se aplica un operador de recombinación o mutación según sea el caso.

Ejemplo

Población rechazada

Tabla 12 Población rechazada - algoritmo cultural

Empresa 1	Empresa 2	Empresa 3	<i>Fitness</i>	Aplicar Operador	Estado
0.8	0.1	0.1	0.4	Apuesta	Operación 1
0	0	1	0.25	No hacer nada	Operación 2
0	1	0	0.15	Flip	.
0.5	0.5	0	0.09	Renacer	.
1	0	0	0.01	Permutación	.

Fuente, Elaboración propia

La selección del operador de recombinación o mutación se realiza de manera aleatoria, asignando a cada operador una probabilidad de ser elegido de manera aleatoria y aplicando el método ruleta para su selección.

Ejemplo

Tomar operadores

Tabla 13 Operador a aplicar - algoritmo cultural

Operador a aplicar	Probabilidad aleatoria
Apuesta	0.1
No hacer nada	0.2
Flip	0.25
Renacer	0.15
Permutación	0.1
Mutación Gauss	0.2
<i>Suma</i>	<i>1.0</i>

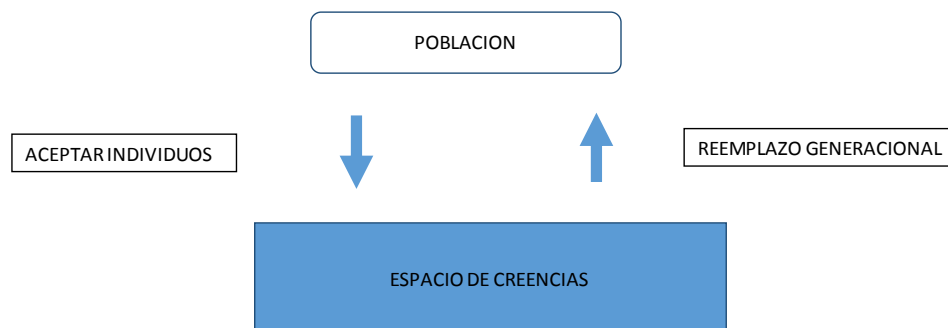
Fuente, Elaboración propia

Seguidamente se aplica el método ruleta para seleccionar el operador y este es asignado al individuo como se muestra en la tabla anterior.

Por último, los individuos modificados se incorporan a la nueva población y se evalúa la función de fitness aplicando la matriz de Markowitz. Así se asigna cambia el vector de aptitud y se procede a realizar el reemplazo generacional.

Reemplazo generacional

Se utilizó un reemplazo generacional usando la técnica *State Stade* la cual consiste en comparar el individuo generado y el anterior, si este es mejor el individuo anterior será reemplazado de lo contrario seguirá igual.

Figura 2 Diagrama algoritmo cultural propuesto

Fuente, Elaboración propia

Funcionamiento del algoritmo genético convencional

Al basarse en una estrategia evolutiva requiere la generación de una población (Combinaciones aleatorias de portafolios), las cuales serán recombinadas y mutadas a medida que avanza el algoritmo.

Pseudocódigo

P0=Generar población inicial (siguiendo una distribución uniforme, con la restricción que la suma de las probabilidades de los portafolios sea igual a 1)

Mientras (Iteraciones < Total de Iteraciones)

 F=Evaluar usando la Matriz de Markowitz (P0)

 P1=Seleccionar Individuos (Se utiliza el método ruleta y una Z normalización)

 Si (Numero Aleatorio [0,1] < Probabilidad de Cruce) entonces

 Seleccione un individuo al azar y realice una permutación

 Fin Si

 Si (Numero Aleatorio [0,1] < Probabilidad de Mutación) entonces

 Seleccione un individuo al azar y realice una Mutación Gaussiana

 Fin Si

 P0=Reemplazo generacional (P1)

Si (Contador ==50)

 Si (Mejoro el Mejor individuo) entonces

 //Aumenta Recombinaciones para realizar más explotación

 Probabilidad de Cruce= Probabilidad de Cruce/0.2

 Probabilidad de Mutación= Probabilidad de Mutación*0.2

Si no

//Aumenta Mutaciones para realizar más exploración

Probabilidad de Cruce= Probabilidad de Cruce*0.2

Probabilidad de Mutación= Probabilidad de Mutación/0.2

Fin Si

Contador=0

Fin Si

Fin Mientras

Selección usando ruleta

Se crea para esta selección una ruleta con los cromosomas presentes en una generación, de acuerdo con De la Peña y Truyol (Jorge Arranz de la Peña, Antonio Truyol, 2016), cada cromosoma tendrá una parte de esa ruleta mayor o menor en función a la puntuación que tenga cada uno. Se hace girar la ruleta y se selección el cromosoma en el que se para la ruleta. Obviamente el cromosoma con mayor puntuación saldrá con mayor probabilidad. En caso de que las probabilidades difieran mucho, este método de selección dará problemas puesto que, si un cromosoma tiene un 90% de posibilidades de ser seleccionado, el resto apenas saldrá lo que reduciría la diversidad genética.

A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad (Pose, 2017).

Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente la población está ordenada en base al ajuste por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo [0, 1] y devolver el individuo situado en esa posición de la ruleta. Esta

posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Recordando que anterior a esto se realiza una Z normalización para ajustar todos los valores de portafolio en la misma unidad de medida.

Cruce;

Es una permutación La cual desordena todos los elementos que componen al individuo.

Ejemplo

Tabla 14 Recombinación - Algoritmo Genético

Individuo sin recombinar		
1	2	3

Individuo recombinado		
2	1	3

Fuente, Elaboración propia

Mutación;

Selecciona un individuo al azar, posteriormente lo descompone y selecciona uno de sus componentes al azar y lo multiplica por un número aleatorio entre cero y uno (siguiendo una distribución gaussiana). Seguidamente selecciona su componente contiguo y reduce o aumenta la proporción asignada por el numero aleatorio.

Ejemplo

Tabla 15 Mutación - algoritmo genético

Individuo Normal				
Componente	1	2	3	4
Valor	0.1	0.2	0.3	0.4

Fuente, Elaboración propia

Para este caso el componente seleccionado fue el numero 2

Así que este se multiplica por un número aleatorio entre cero y uno

$$\text{Componente 2} = 0.2 * \text{random} = 0.35$$

Ahora como el nuevo valor de esa celda será 0.3, se toma la diferencia entre el valor anterior y el nuevo.

$$\text{Diferencia} = 0.2 - 0.35 = -0.15$$

Y esta diferencia se suma a la celda contigua

$$\text{Componente 3} = 0.3 + (-0.15) = 0.15$$

De tal manera que

Tabla 16 Individuo mutado - algoritmo genético

Individuo Mutado				
Componente	1	2	3	4
Valor	0.1	0.35	0.15	0.4

Fuente, Elaboración propia

Para el caso del ultimo valor se toma el componente anterior a este.

Reemplazo generacional;

Se utilizó un reemplazo generacional usando la técnica *State Stade* la cual consiste en comparar el individuo generado y el anterior, si este es mejor el individuo anterior será reemplazado de lo contrario seguirá igual.

Funcionamiento del Algoritmo de Hill Climbing

En ciencia de la computación, Hill Climbing (ascenso de colinas, en alguna literatura) es una técnica de optimización matemática que pertenece a la familia de los algoritmos de búsqueda local.

Es un algoritmo iterativo que comienza con una solución arbitraria a un problema, luego intenta encontrar una mejor solución variando incrementalmente un único elemento de la solución.

Si el cambio produce una mejor solución, otro cambio incremental se le realiza a la nueva solución, repitiendo este proceso hasta que no se puedan encontrar mejoras.

Pseudocódigo

portafolio=generar combinación de portafolio inicial

mientras (iteracion<total de interacciones)

 solucion nueva=evaluar portafolio usando la matriz de markowitz

 si (solucion nueva es mejor que solucion anterior) entonces,

 portafolio=nuevo potafolio

 si no,

 mantengase como esta

 nuevo portafolio=generar combinacion de portafolio

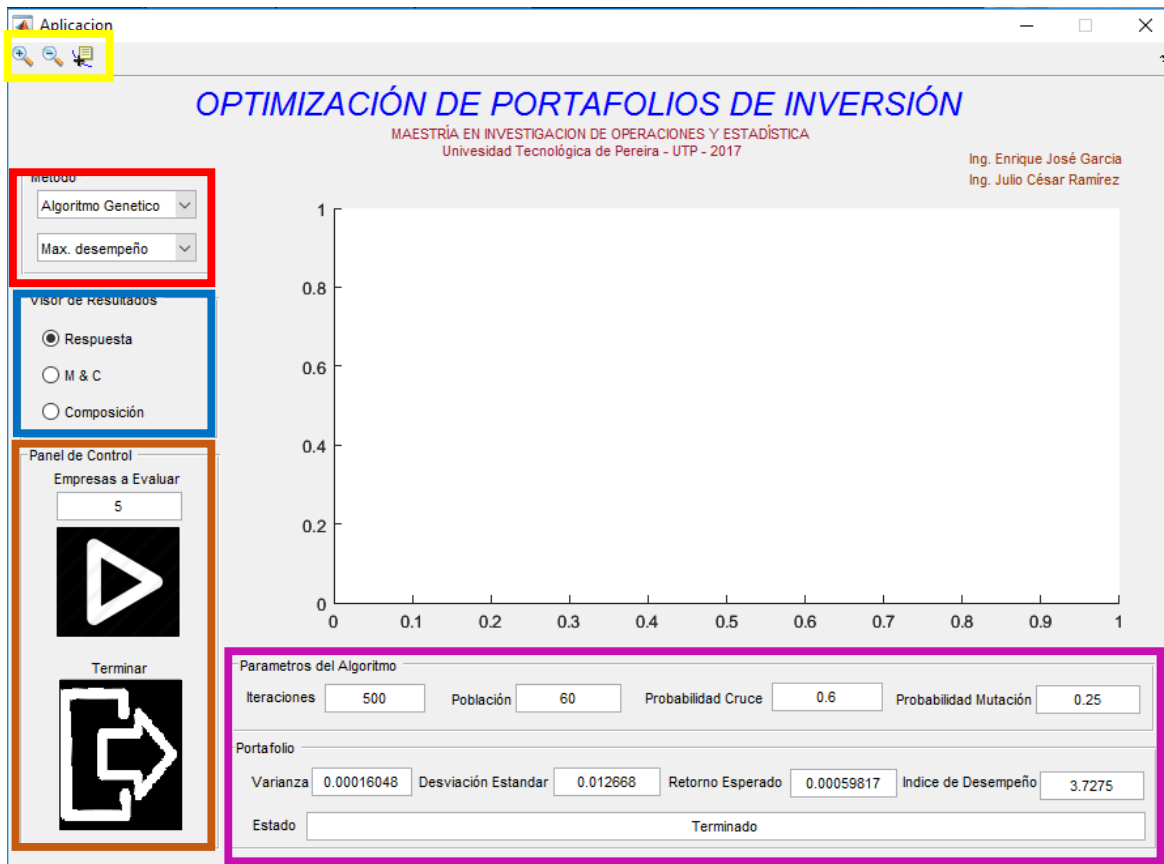
iteracion ++;

fin mientras

Aplicación en MATLAB

Inicialmente la aplicación se compone de una interfaz gráfica de usuario la cual funciona de la siguiente manera:

Figura 3 Interface Gráfica en Matlab



Fuente, Elaboración propia

La barra de herramientas (marcada de color amarillo), tiene tres botones, las lupas dan la opción de zoom sobre el área de trabajo y la cruz permite seleccionar un punto sobre la gráfica para conocer su valor.

La opción de método (marcada de color rojo): permite seleccionar el algoritmo mediante el cual se podrá optimizar el portafolio, las opciones habilitadas son las siguientes:

- **Algoritmo cultural:** es una metodología basada en los algoritmos evolutivos donde hay un componente de conocimiento dentro de la población llamado espacio de creencia el cual se encarga de recordar experiencias. Es utilizado para optimizar una población de programas de acuerdo a una función de ajuste o aptitud (en inglés: *fitness function*) que evalúa la capacidad de cada programa para llevar a cabo la tarea en cuestión (Borradopedia, 2016).

- **Algoritmo genético convencional;** Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados (Wikipedia, 2016).

- **Algoritmo de Hill Climbing;** es un algoritmo de búsqueda aleatoria, el cual permite acercarse a un óptimo global rápidamente, generando números aleatorios siguiendo una distribución uniforme.

Dentro de esta opción también se puede seleccionar si se desea maximizar el rendimiento del portafolio o minimizar el riesgo de inversión.

El visor de resultados (enmarcado en color azul); permite seleccionar entre tres opciones diferentes para mostrar los resultados del algoritmo.

- La primera opción muestra los siguientes parámetros de la población;
 - El mejor resultado obtenido durante la ejecución de la aplicación
 - El peor resultado obtenido durante la ejecución de la aplicación

- La segunda opción muestra la mutación y cruces realizados por el algoritmo (Aplica en el caso del algoritmo genético y cultural, para el caso de Hill Climbing no tiene importancia porque este método no se basa en una estrategia poblacional), durante el tiempo de ejecución.
- La tercera muestra la composición del portafolio.

El panel de control (marcado de color marrón); permite ingresar el número de empresas con las cuales se va a realizar el estudio y tiene los botones con la función de ejecutar la aplicación y cerrar el programa.

Por último, se encuentran los parámetros de entrada y salida del algoritmo (color morado), entre los cuales están;

- Cantidad de Iteraciones (aplica para todos); número de veces entre las cuales la aplicación intenta encontrar una solución al problema.
- Población (aplica cultural y genético); número de individuos que van a componer el conjunto de soluciones al problema planteado.
- Probabilidad cruce y probabilidad mutación (aplica cultural y genético); Indica la probabilidad de cruce inicial que tendrá el programa para que un individuo se recombine o mute durante la ejecución del programa.

Por último, el algoritmo entrega la varianza (incertidumbre del retorno a la inversión), desviación estándar (medida de riesgo de la rentabilidad), retorno esperado e índice de desempeño. También se encuentra un aviso que indica el estado de la aplicación (Procesando Espere... o Terminado).

Análisis y Resultados

Para validar la propuesta del algoritmo cultural, se utilizan en total 16 funciones las cuales son incluidas como anexo.

Condiciones generales

Ajustes de bases de datos: Los ajustes de las bases de datos se hacen teniendo en cuenta la información disponible en Yahoo-Finance (<https://finance.yahoo.com/>). De allí se obtiene la información relacionada con los mercados accionarios reales de la industria, la tecnológica y los mercados.

Las bases de datos utilizadas fueron:

Aceto Corporation (ACET), Alps Industries Limited, Camden National Corporation, Cisco, Dow Jones Industrial Average, IBM, JPMorgan Chase & Co, Kellogg Company (K), Microsoft, Nike, S&P 500, Terex Corporation, United Technologies Corporation (UTX), Visa Inc. (V), Zynga Inv. (ZNGA)

Ajustes de validación: El aplicativo utilizado fue el desarrollado en MATLAB 2015.

Hardware: Para todas las pruebas de las diferentes bases de datos, se utilizó una máquina con las siguientes características:

Procesador: Intel(r) Core(TM) i7-2670QM CPU @ 2.20GHz 2.20 GHz

Memoria Instalada (RAM): 4.00 GB

Sistema Operativo: Windows 10 Pro N

Lenguaje de Programación: MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M)

Ajustes de parámetros

Los parámetros establecidos fueron:

Número de Iteraciones: 500

Número de individuos: 60

Probabilidad de cruce: 0.6

Probabilidad de mutación: 0.25

Número de acciones que conforman el portafolio: se hará el análisis con 2, 5 y 8 diferentes tipos de acciones, bajo las condiciones de máximo desempeño y mínimo riesgo.

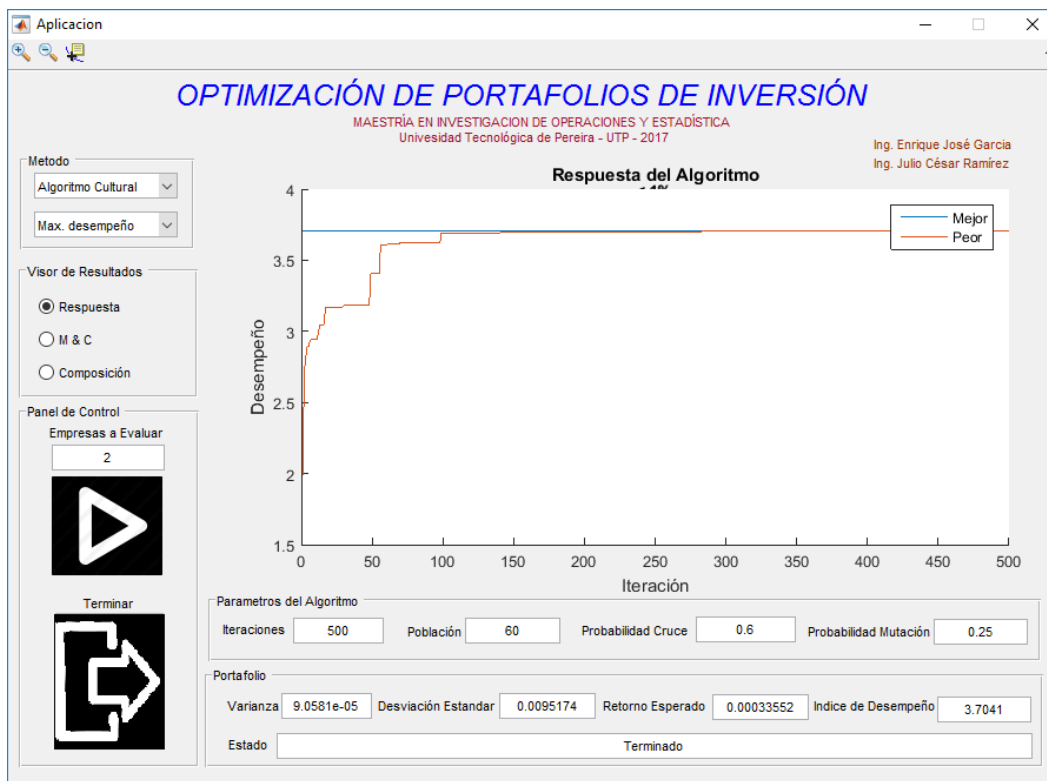
Resultados

Caso 1: 2 Acciones – Máximo Desempeño

Se toman de manera aleatoria las acciones de IBM y Kellogg Company (K), con los parámetros establecidos.

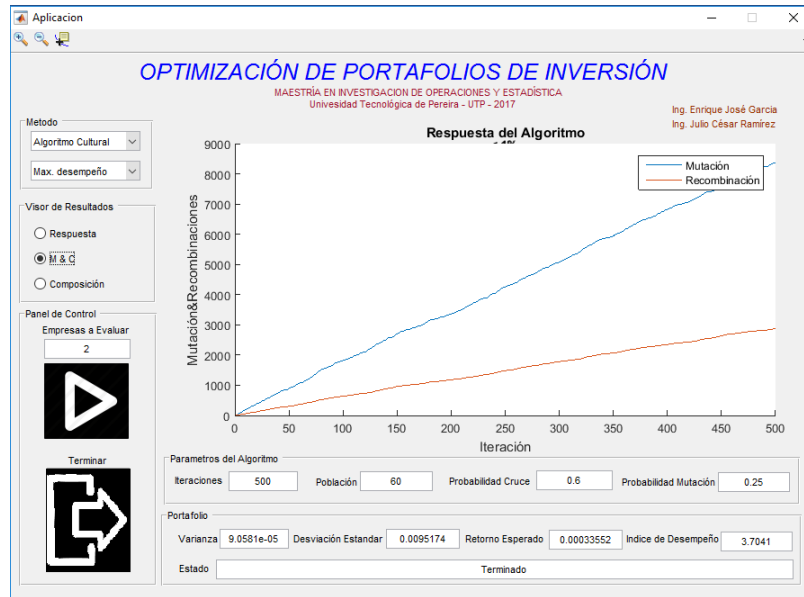
Algoritmo Cultural

Figura 4 Respuesta Algoritmo Cultural - 2 Acciones - Máximo Desempeño



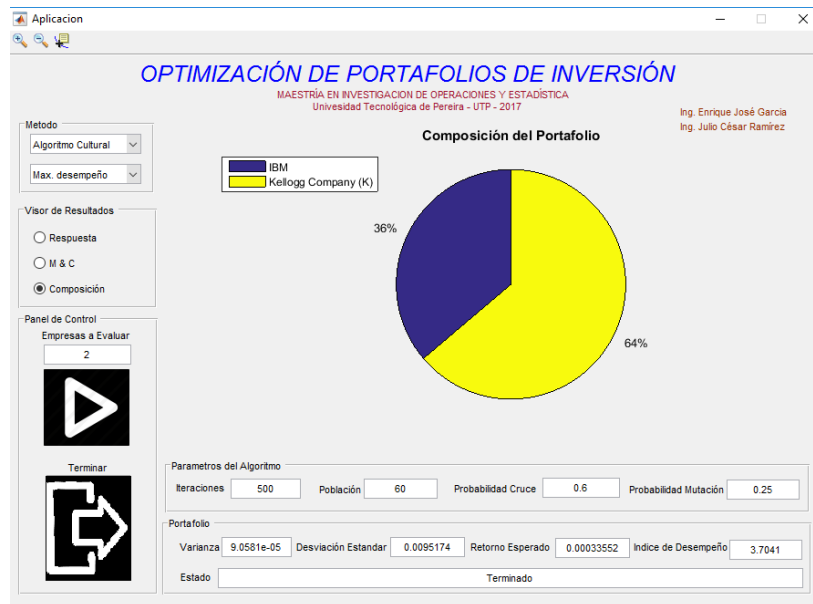
Fuente, Elaboración propia

Figura 5 Mutación y Recombinación Algoritmo Cultural - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

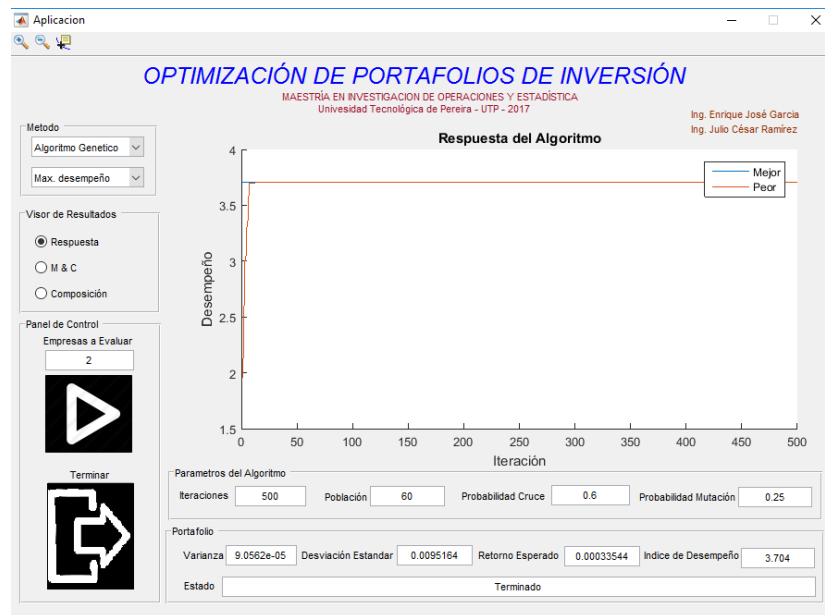
Figura 6 Composición Algoritmo Cultural - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

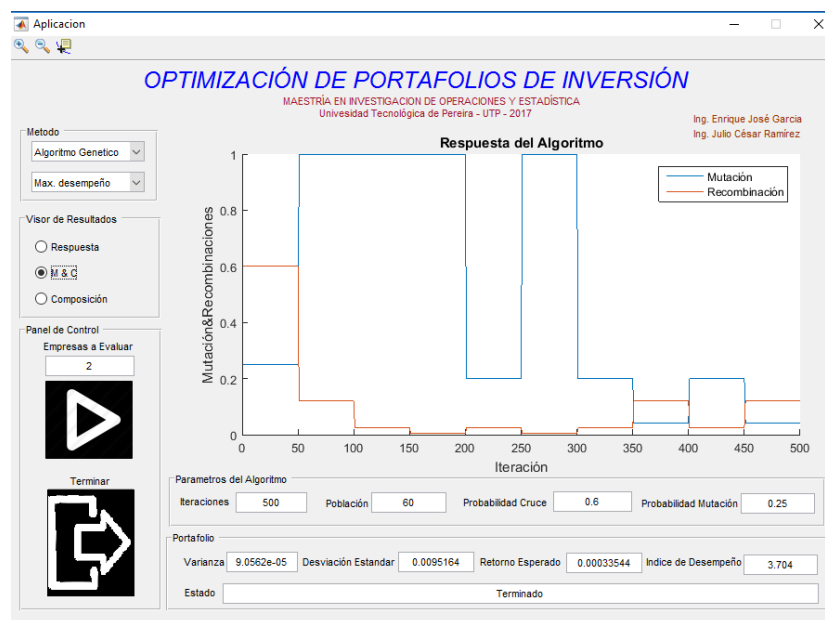
Algoritmo Genético

Figura 7 Respuesta Algoritmo Genético - 2 Acciones - Máximo Desempeño



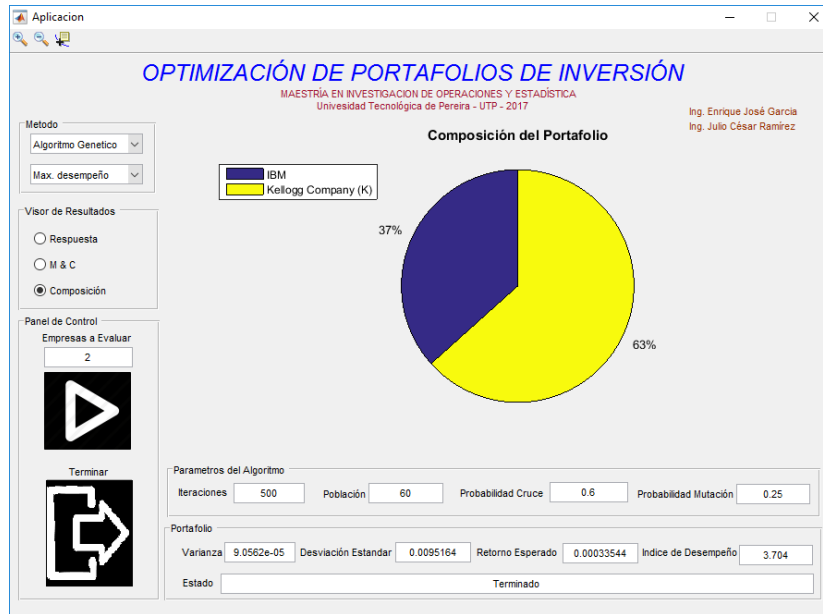
Fuente, Elaboración propia

Figura 8 Mutación y Recombinación Algoritmo Genético - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

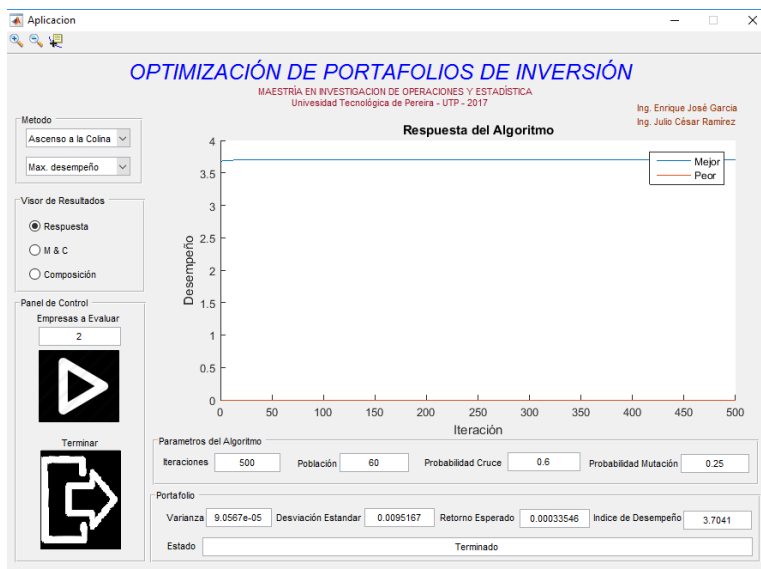
Figura 9 Composición Algoritmo Genético - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

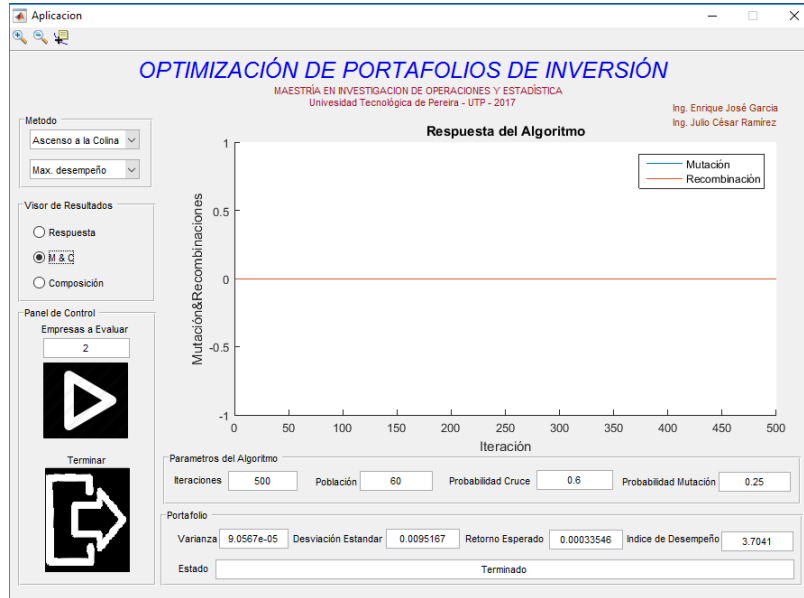
Algoritmo Ascenso A la Colina

Figura 10 Respuesta Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño



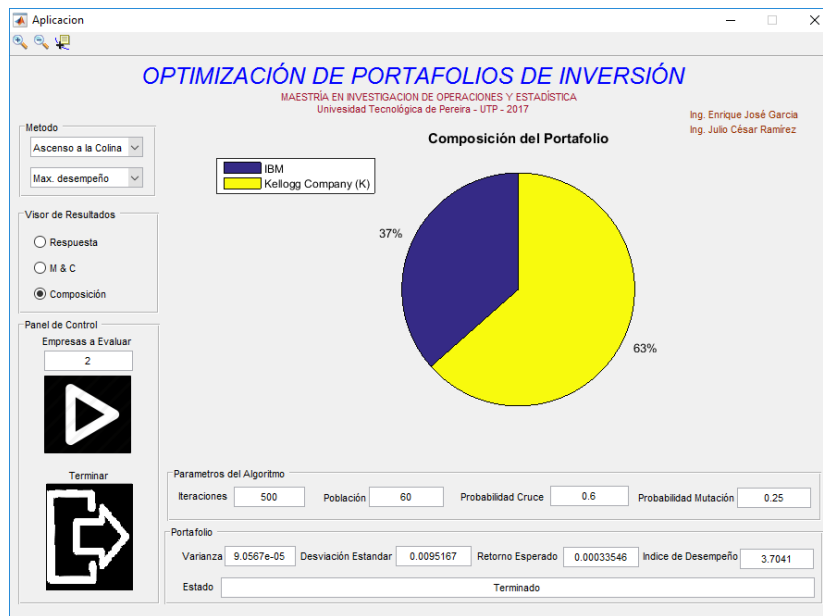
Fuente, Elaboración propia

Figura 11 Mutación y Recombinación Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Figura 12 Composición Algoritmo Ascenso A la Colina - 2 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Tabla 17 Resultados Máximo Desempeño - 2 Acciones

	Máximo Desempeño			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Indice de desempeño
Algoritmo Cultural	9,0581 E-5	0,0095174	0,00033552	3,7041
Algoritmo Genético	9,0562 E-5	0,0095164	0,00033554	3,704
Algoritmo de Ascenso a la Colina	9,0567 E-5	0,0095167	0,00033546	3,7041

Fuente, Elaboración propia

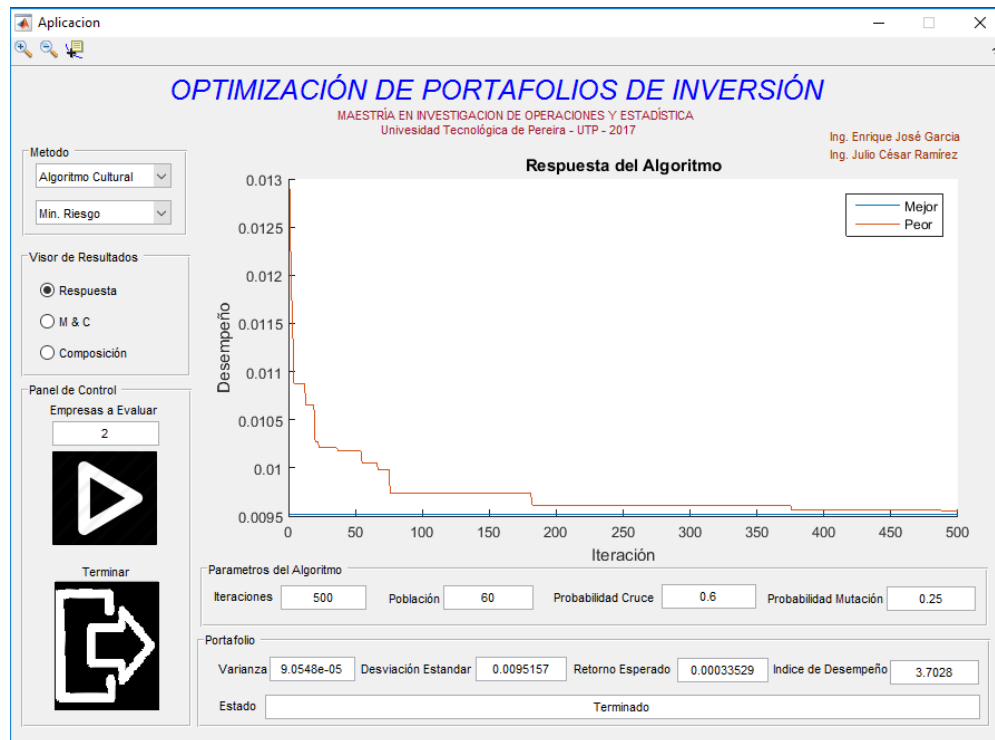
Tabla 18 Composición Portafolio de Inversión 2 Acciones – Máximo Desempeño

Portafolio	
IBM	Kellogg Company (K)
36%	64%
37%	63%
37%	63%

Fuente, Elaboración propia

Caso 1: 2 Acciones – Mínimo Riesgo

Se toman de manera aleatoria las acciones de IBM y Kellogg Company (K), con los parámetros establecidos.

Algoritmo Cultural**Figura 13** Respuesta Algoritmo Cultural - 2 Acciones - Mínimo Riesgo

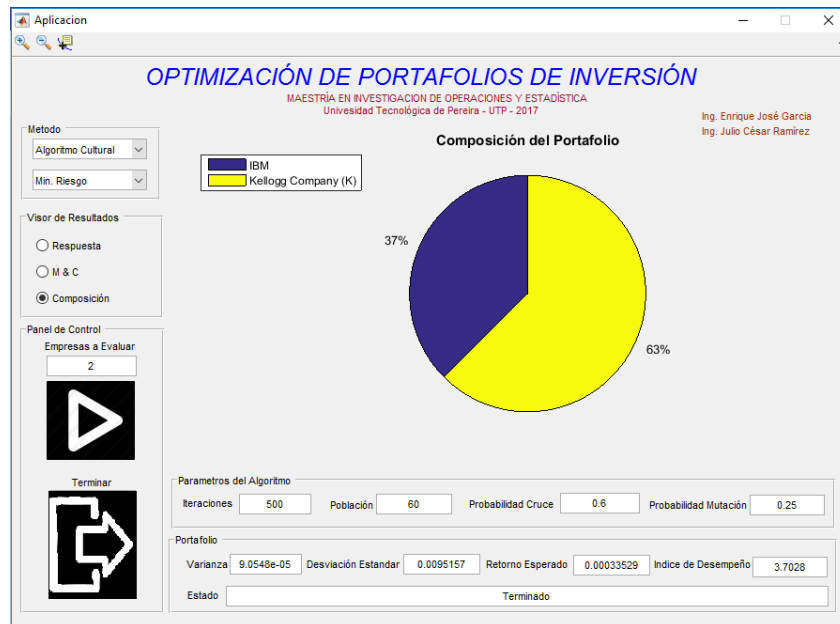
Fuente, Elaboración propia

Figura 14 Mutación y Recombinación Algoritmo Cultural - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 15 Composición Algoritmo Cultural - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

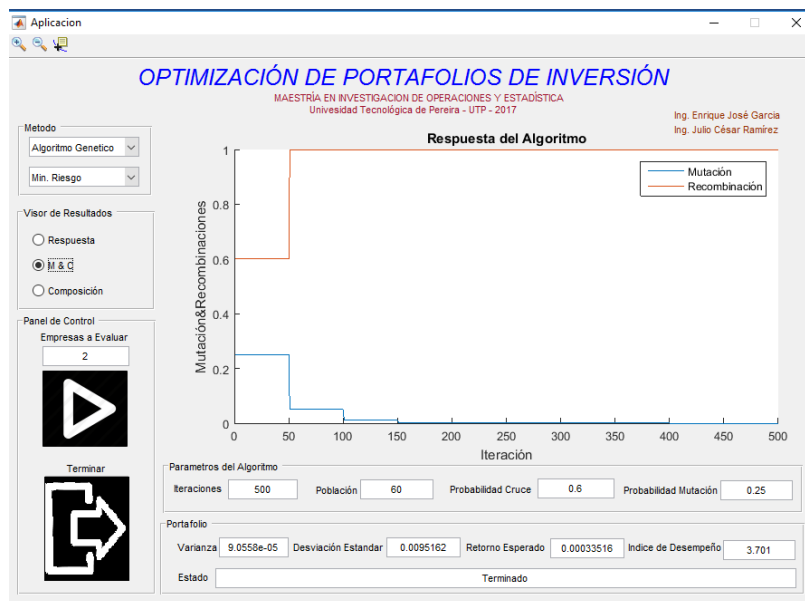
Algoritmo Genético

Figura 16 Respuesta Algoritmo Genético - 2 Acciones - Mínimo Riesgo



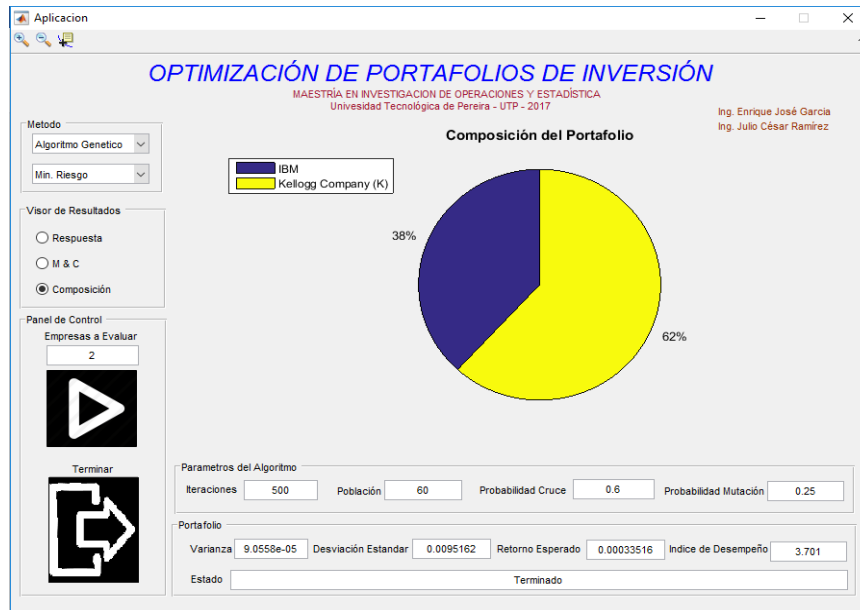
Fuente, Elaboración propia

Figura 17 Mutación y Recombinación Algoritmo Genético - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 18 Composición Algoritmo Genético - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

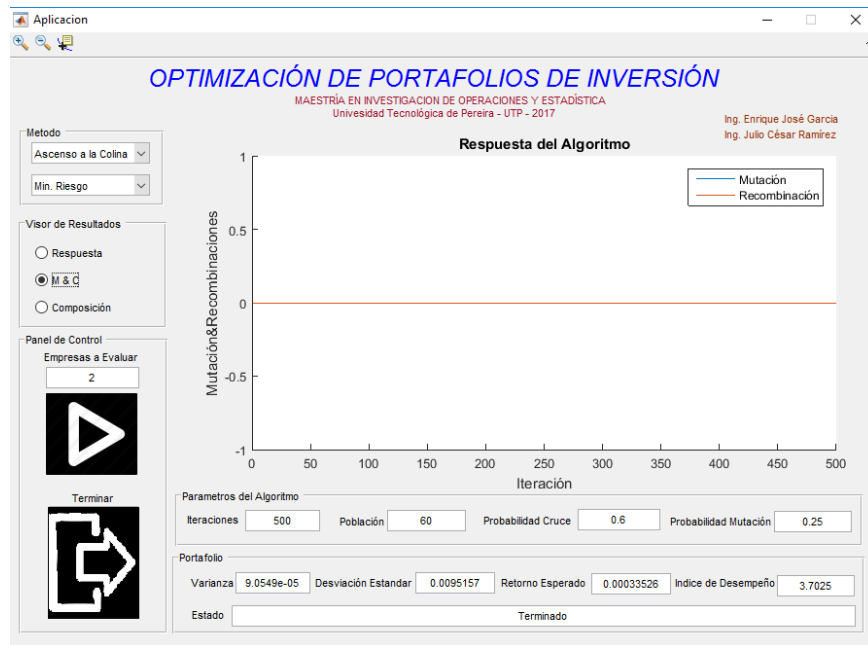
Algoritmo Ascenso A la Colina

Figura 19 Respuesta Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo



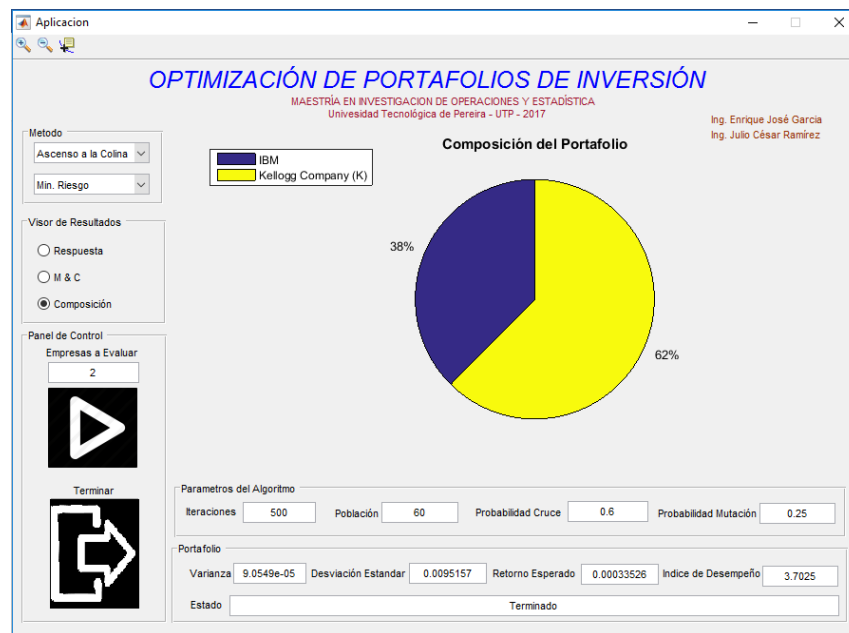
Fuente, Elaboración propia

Figura 20 Mutación y Recombinación Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 21 Composición Algoritmo Ascenso A la Colina - 2 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Tabla 19 Resultados Mínimo Riesgo - 2 Acciones

	Mínimo Riesgo			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Indice de desempeño
Algoritmo Cultural	9,0548 E-5	0,0095157	0,00033529	3,7028
Algoritmo Genético	9,0558 E-5	0,0095162	0,00033516	3,701
Algoritmo de Ascenso a la Colina	9,0549 E-5	0,0095157	0,00033526	3,7025

Fuente, Elaboración propia

Tabla 20 Composición Portafolio de Inversión 2 Acciones – Mínimo Riesgo

Portafolio	
IBM	Kellogg Company (K)
37%	63%
38%	62%
38%	62%

Fuente, Elaboración propia

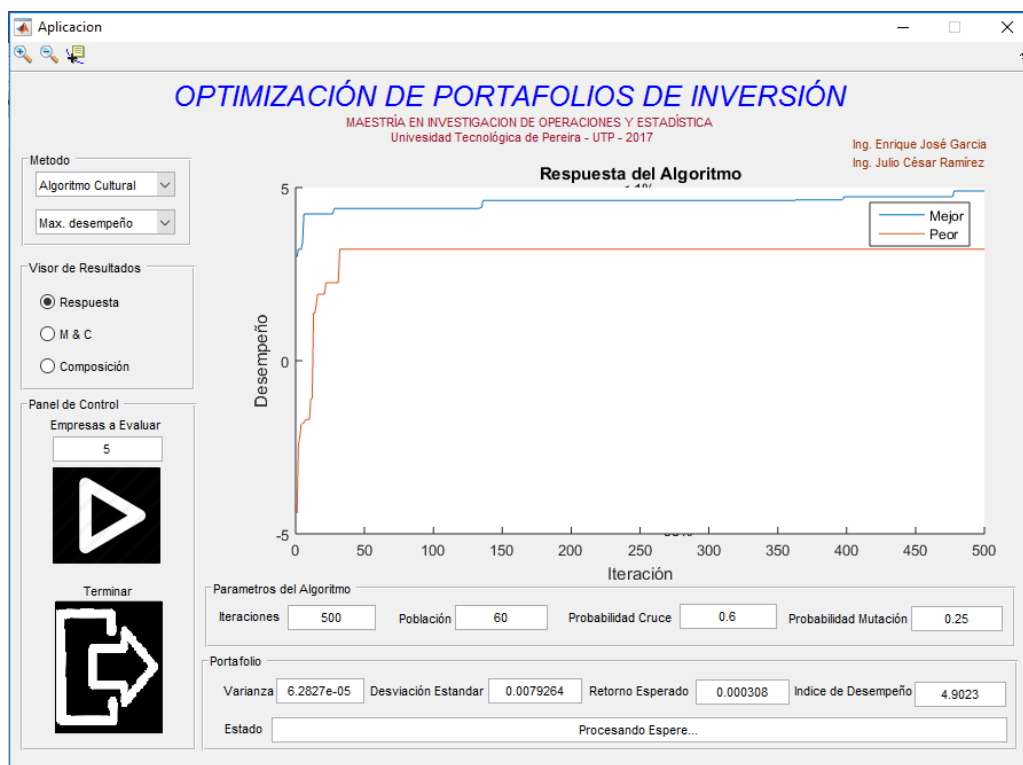
Tabla 21 Consolidado Portafolio Inversión 2 Acciones

	Portafolio Máximo Desempeño		Portafolio Mínimo Riesgo	
	IBM	Kellogg Company (K)	IBM	Kellogg Company (K)
Algoritmo Cultural	36%	64%	37%	63%
Algoritmo Genético	37%	63%	38%	62%
Algoritmo Ascenso a la Colina	37%	63%	38%	62%

Fuente, Elaboración propia

Caso 2: 5 Acciones – Máximo Desempeño

Se toman de manera aleatoria las acciones de Alps Industries Limited, Dow Jones Industrial Average, IBM, Nike y Zynga Inv. (ZNGA), con los parámetros establecidos.

Algoritmo Cultural**Figura 22** Respuesta Algoritmo Cultural - 5 Acciones - Máximo Desempeño

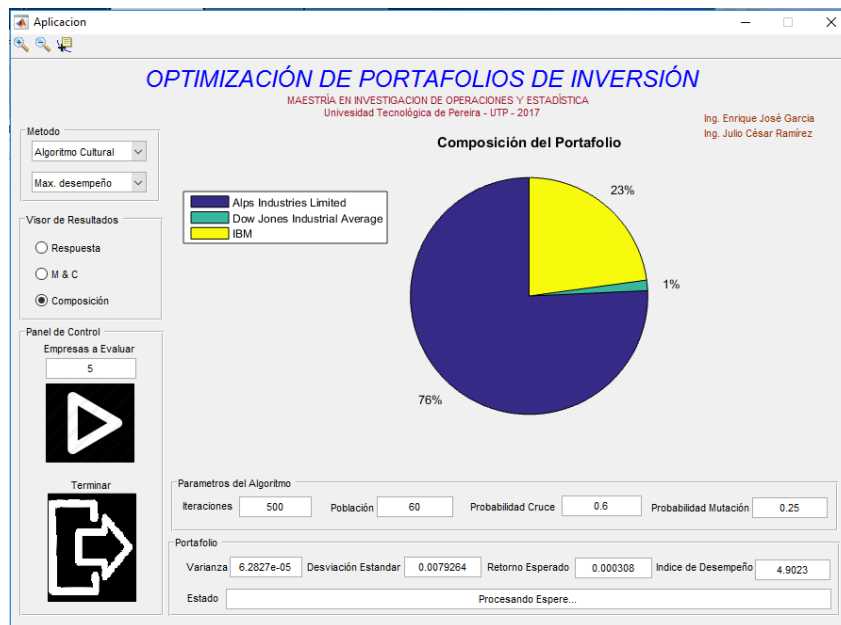
Fuente, Elaboración propia

Figura 23 Mutación Y Recombinación Algoritmo Cultural - 5 Acciones - Máximo Desempeño



Fuente, Elaboración propia

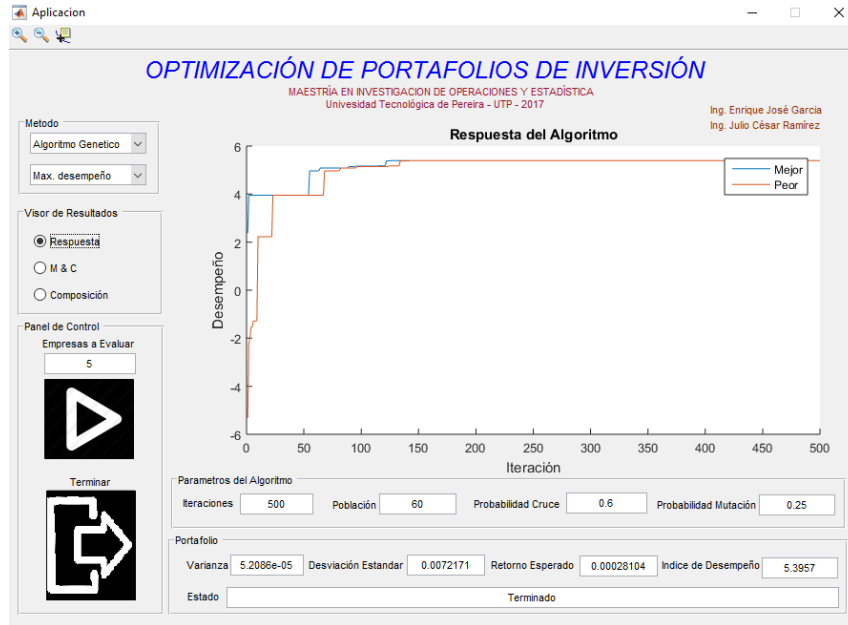
Figura 24 Composición Algoritmo Cultural - 5 Acciones - Máximo Desempeño



Fuente, Elaboración propia

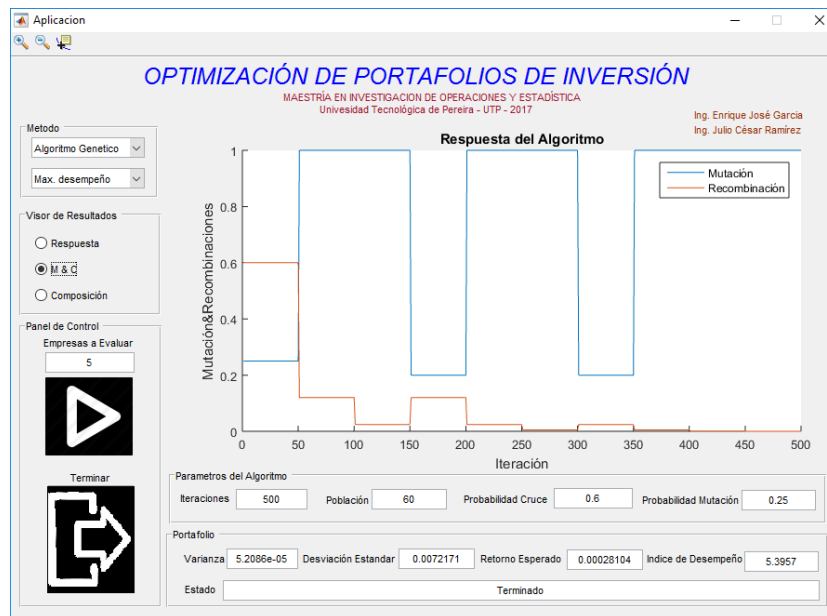
Algoritmo Genético

Figura 25 Respuesta Algoritmo Genético - 5 Acciones - Máximo Desempeño

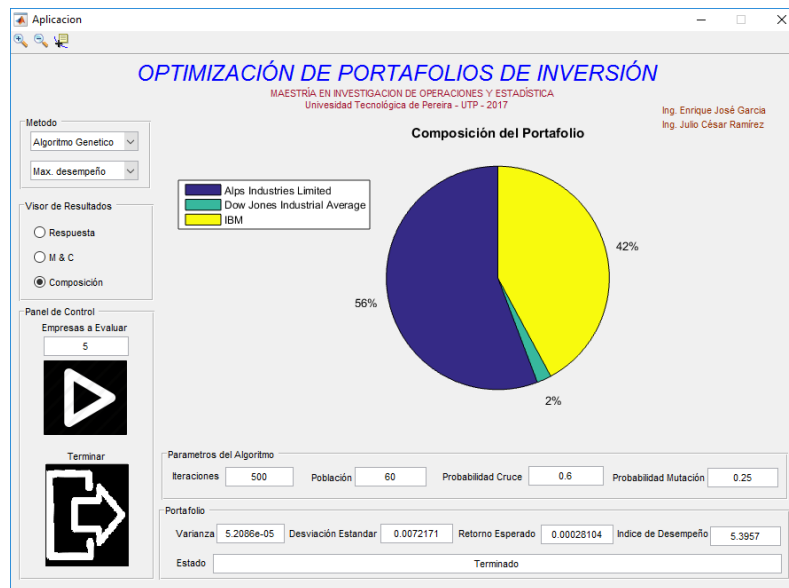


Fuente, Elaboración propia

Figura 26 Mutación y Recombinación Algoritmo Genético - 5 Acciones - Máximo Desempeño

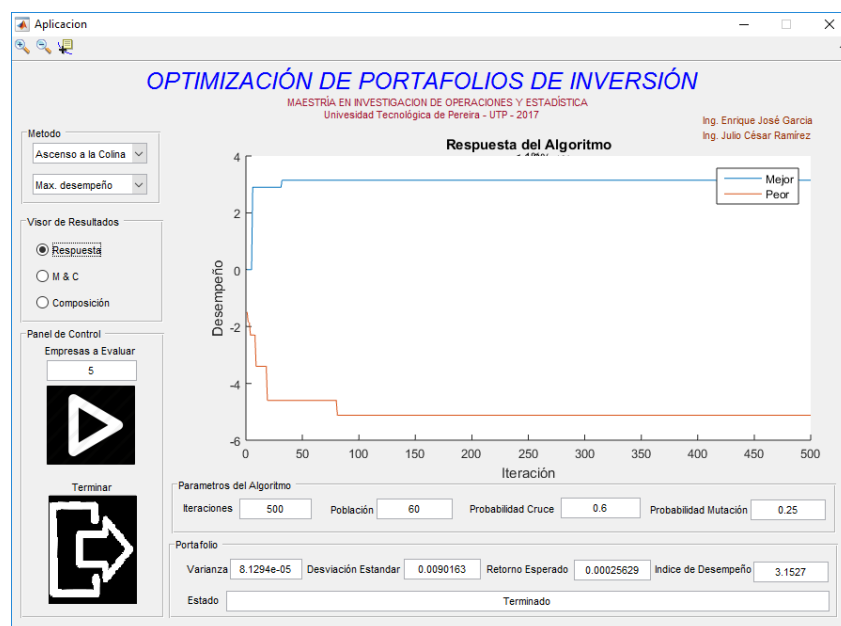


Fuente, Elaboración propia

Figura 27 Composición Algoritmo Genético - 5 Acciones - Máximo Desempeño

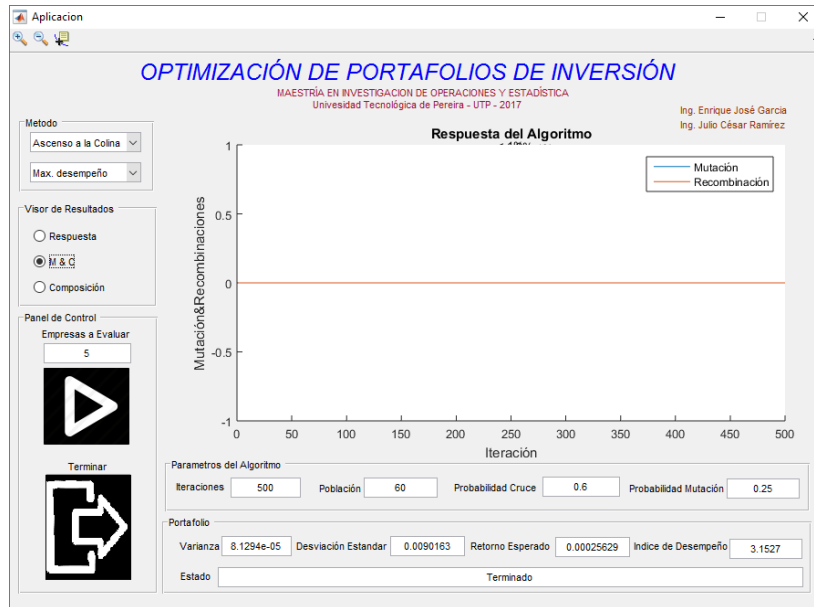
Fuente, Elaboración propia

Algoritmo Ascenso A la Colina

Figura 28 Respuesta Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño

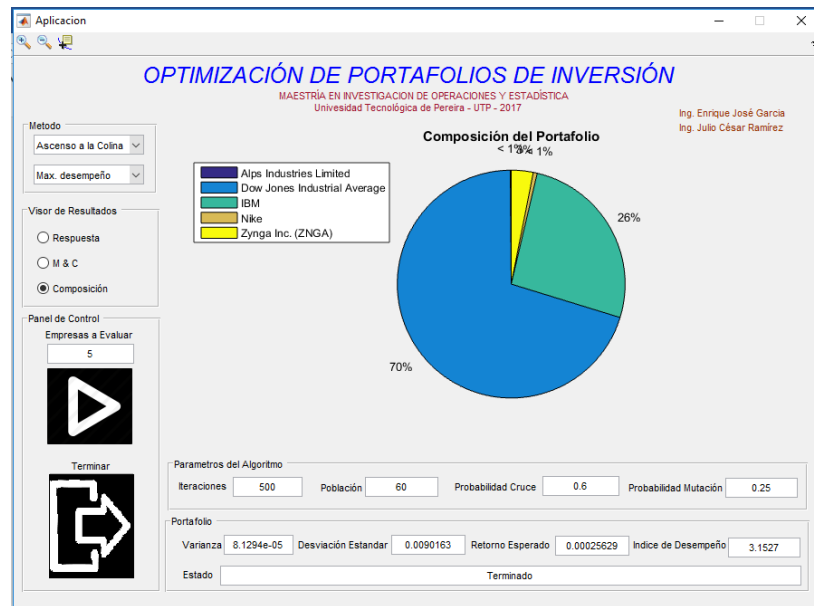
Fuente, Elaboración propia

Figura 29 Mutación y Recombinación Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Figura 30 Composición Algoritmo Ascenso A la Colina - 5 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Tabla 22 Resultados Máximo Desempeño - 5 Acciones

	Máximo Desempeño			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Índice de desempeño
Algoritmo Cultural	6,2872 e-5	0,0079624	0,000308	4,9023
Algoritmo Genético	5,2086 e-5	0,007217	0,00028104	5,3957
Algoritmo de Ascenso a la Colina	8,1249 e-5	0,0090163	0,00025629	31.527

Fuente, Elaboración propia

Tabla 23 Composición Portafolio de Inversión 5 Acciones – Máximo Desempeño

Portafolio - Máximo Desempeño				
Alps Industries Limited	Dow Jones Industrial Average	IBM	Nike	Zynga Inv. (ZNGA),
76%	1%	23%	0%	0%
56%	2%	42%	0%	0%
< 1%	70%	26%	< 1%	3%

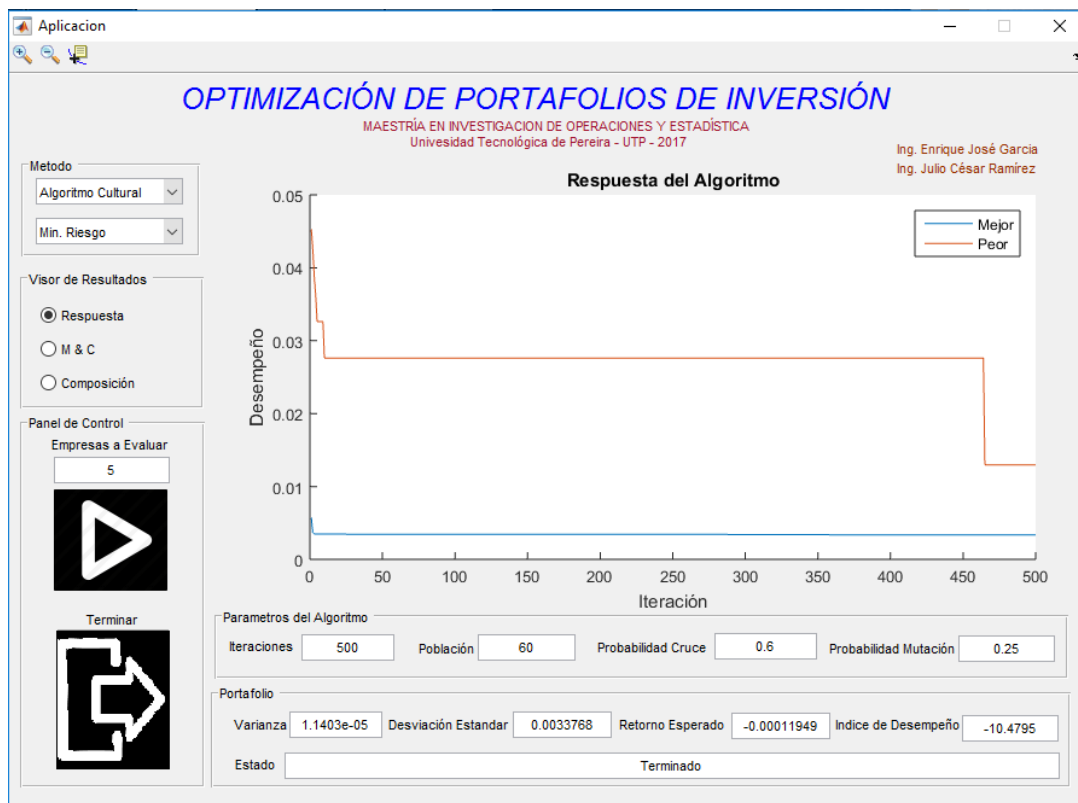
Fuente, Elaboración propia

Caso 2: 5 Acciones – Mínimo Riesgo

Se toman de manera aleatoria las acciones de Alps Industries Limited, Dow Jones Industrial Average, IBM, Nike y Zynga Inv. (ZNGA), con los parámetros establecidos.

Algoritmo Cultural

Figura 31 Respuesta Algoritmo Cultural - 5 Acciones - Mínimo Riesgo



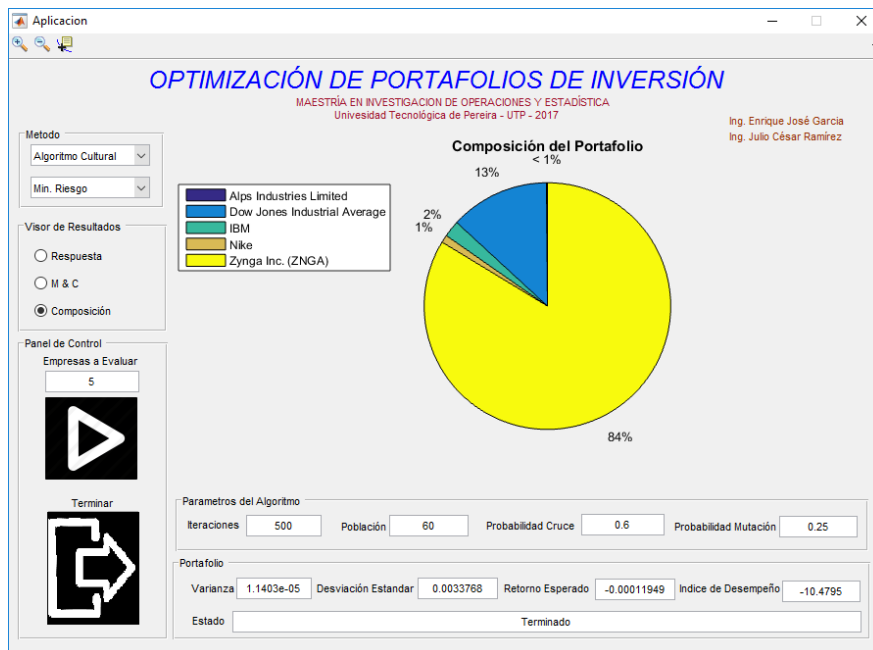
Fuente, Elaboración propia

Figura 32 Mutación y Recombinación Algoritmo Cultural - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 33 Composición Algoritmo Cultural - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

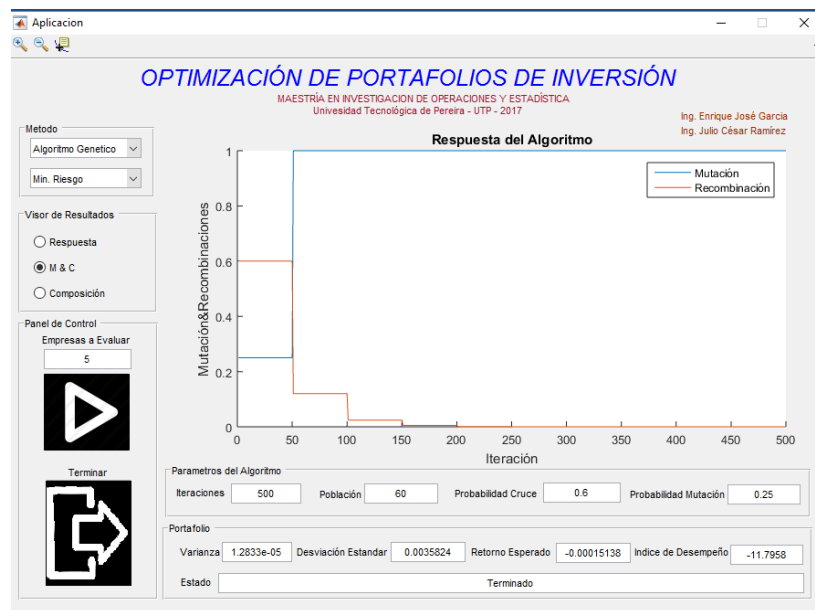
Algoritmo Genético

Figura 34 Respuesta Algoritmo Genético - 5 Acciones - Mínimo Riesgo



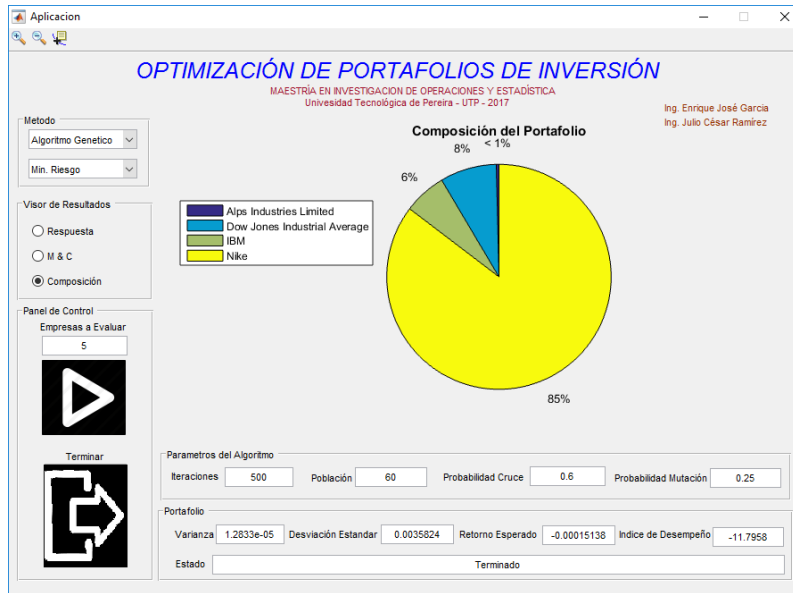
Fuente, Elaboración propia

Figura 35 Mutación y Recombinación Algoritmo Genético - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

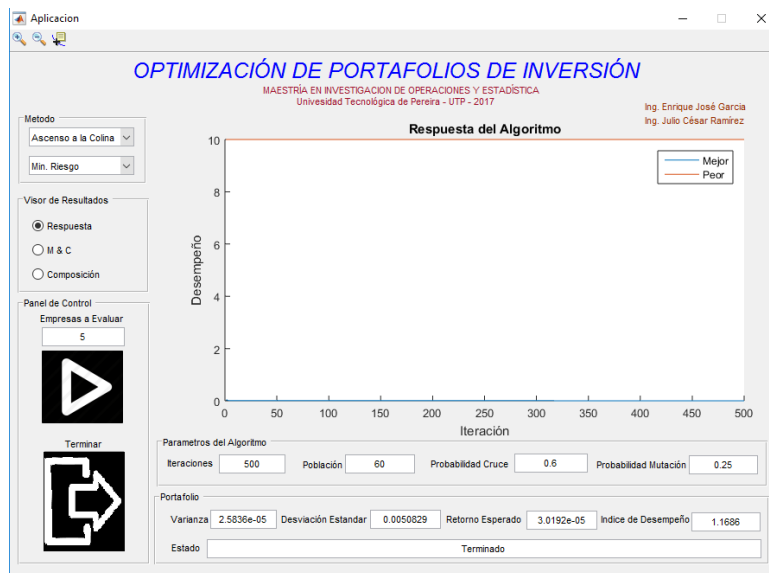
Figura 36 Composición Algoritmo Genético - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

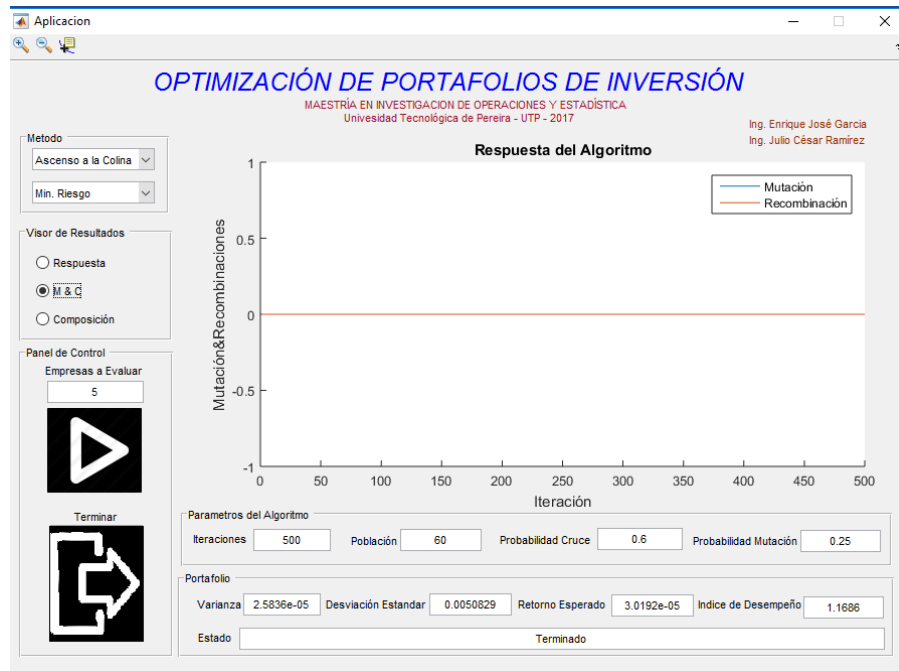
Algoritmo Ascenso A la Colina

Figura 37 Respuesta Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo



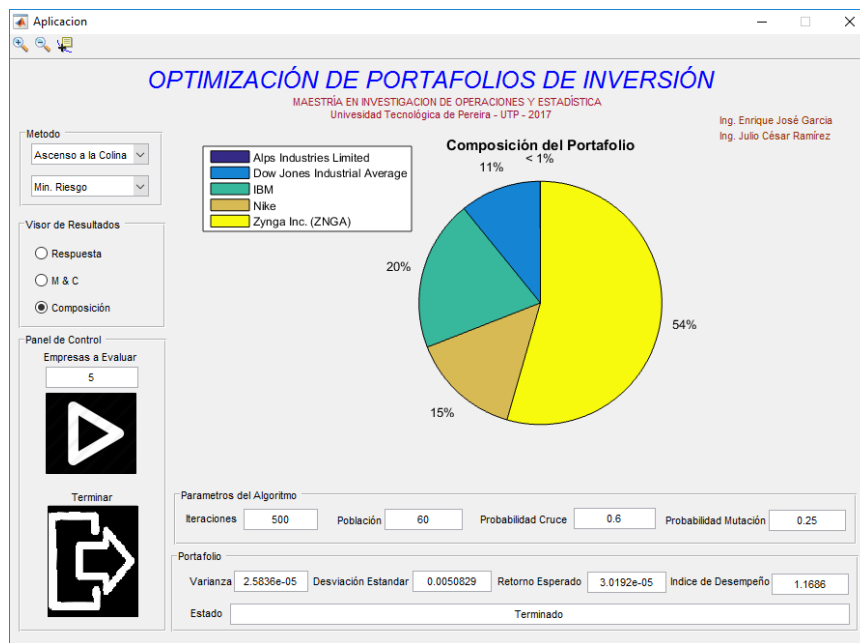
Fuente, Elaboración propia

Figura 38 Mutación y Recombinación Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 39 Composición Algoritmo Ascenso A la Colina - 5 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Tabla 24 Resultados Mínimo Riesgo - 5 Acciones

	Mínimo Riesgo			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Índice de desempeño
Algoritmo Cultural	1,1403 e-5	0,0033768	0,00011949	-10,4795
Algoritmo Genético	1,2833 e-5	0,0035824	0,00015138	-11,7958
Algoritmo de Ascenso a la Colina	1,5836 e-5	0,0050829	3,0192 e -5	1,1686

Fuente, Elaboración propia

Tabla 25 Composición Portafolio de Inversión 5 Acciones – Mínimo Riesgo

Portafolio				
Alps Industries Limited	Dow Jones Industrial Average	IBM	Nike	Zynga Inv. (ZNGA),
< 1%	13%	2%	1%	84%
< 1%	8%	6%	85%	< 1%
< 1%	11%	20%	15%	54%

Fuente, Elaboración propia

Tabla 26 Consolidado Portafolio Inversión 5 Acciones

	Portafolio - Máximo Desempeño				
	Alps Industries Limited	Dow Jones Industrial Average	IBM	Nike	Zynga Inv. (ZNGA),
Algoritmo Cultural	76%	1%	23%	0%	0%
Algoritmo Genético	56%	2%	42%	0%	0%
Algoritmo Ascenso a la Colina	< 1%	70%	26%	< 1%	3%

	Portafolio - Mínimo Riesgo				
	Alps Industries Limited	Dow Jones Industrial Average	IBM	Nike	Zynga Inv. (ZNGA),
Algoritmo Cultural	< 1%	13%	2%	1%	84%
Algoritmo Genético	< 1%	8%	6%	85%	< 1%
Algoritmo Ascenso a la Colina	< 1%	11%	20%	15%	54%

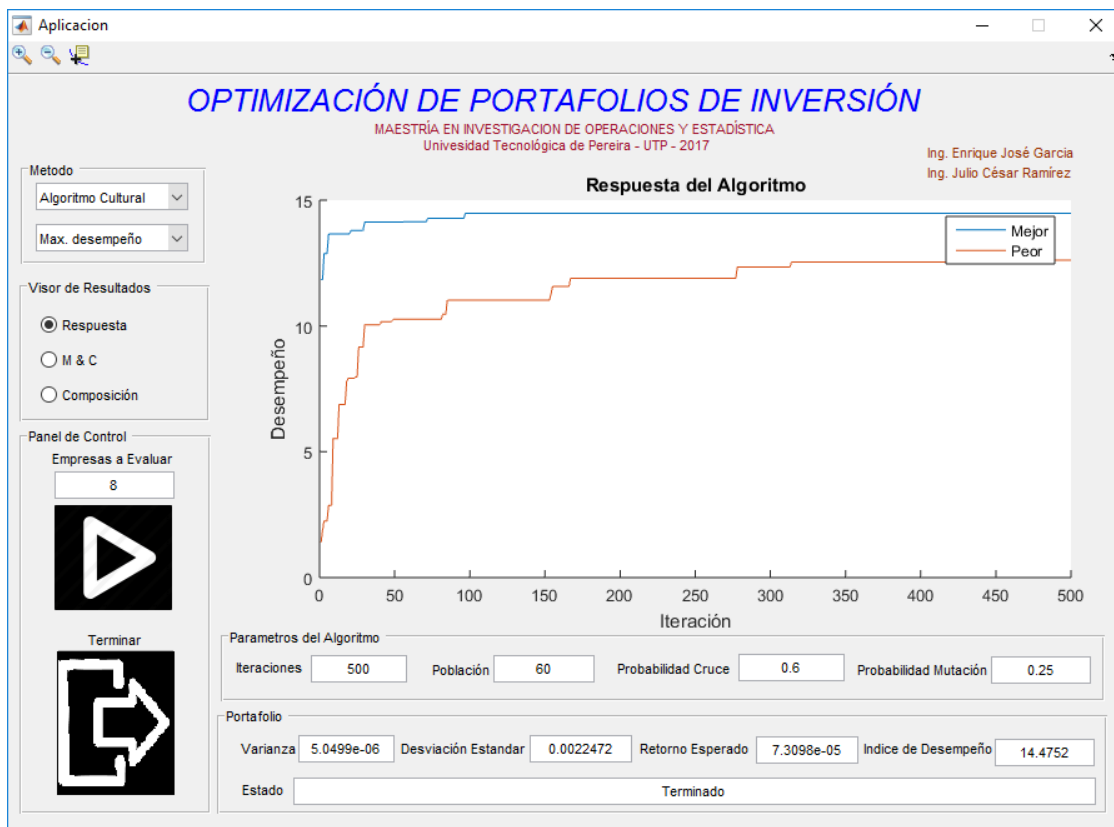
Fuente, Elaboración propia

Caso 3: 8 Acciones – Máximo Desempeño

Se toman de manera aleatoria las acciones de Aceto Corporation (ACET), Camden National Corporation, Cisco, Dow Jones Industrial Average, JPMorgan Chase & Co, Microsoft, Terex Corporation, United Technologies Corporation (UTX), Visa Inc. (V), con los parámetros establecidos.

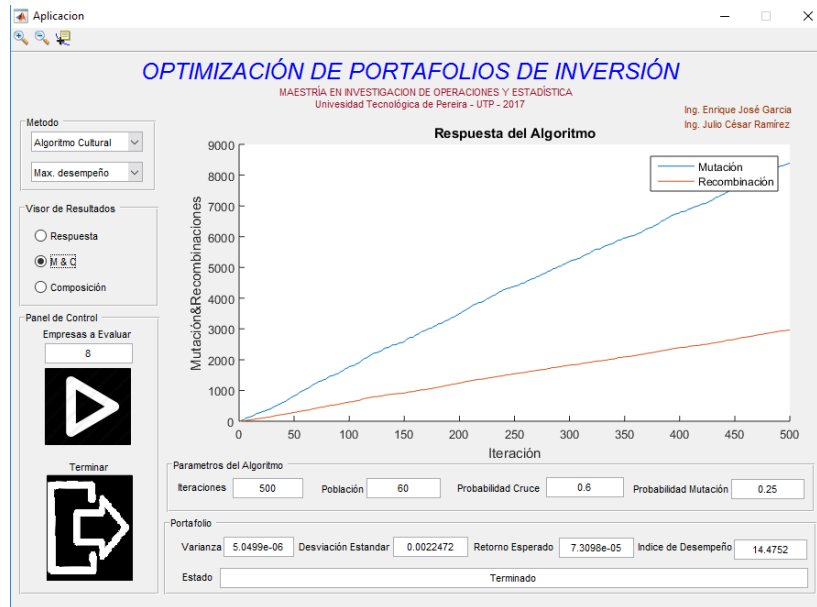
Algoritmo Cultural

Figura 40 Respuesta Algoritmo Cultural - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Figura 41 Mutación y Recombinación Algoritmo Cultural - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

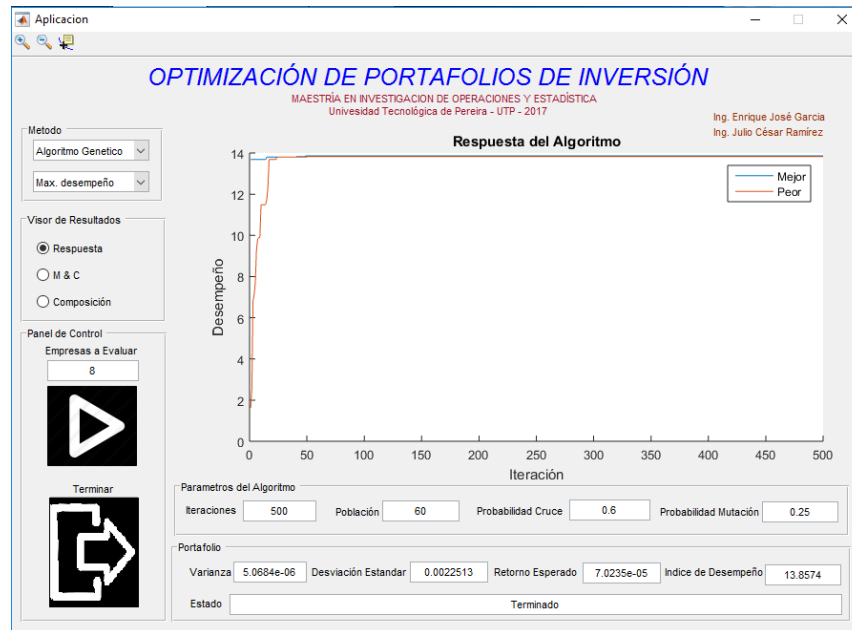
Figura 42 Composición Algoritmo Cultural - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

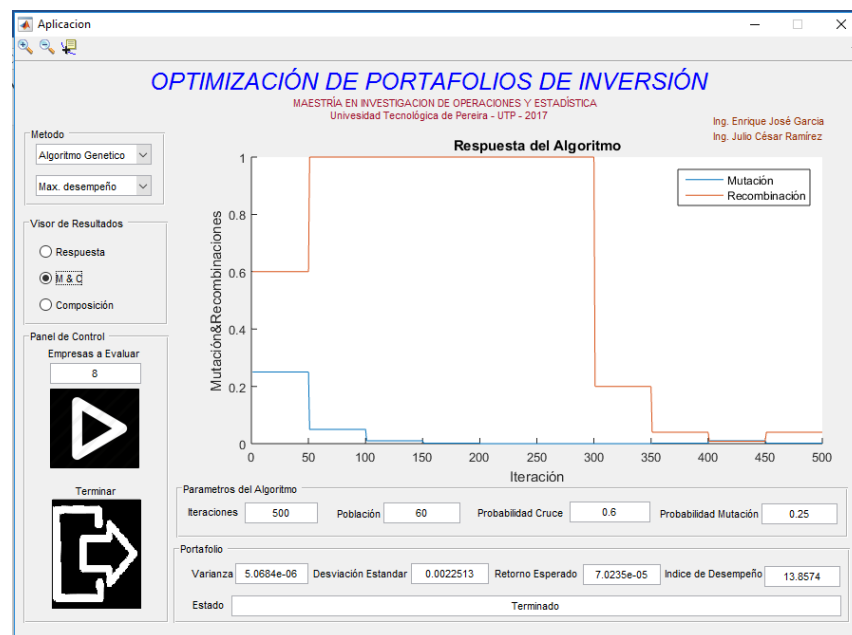
Algoritmo Genético

Figura 43 Respuesta Algoritmo Genético - 8 Acciones - Máximo Desempeño



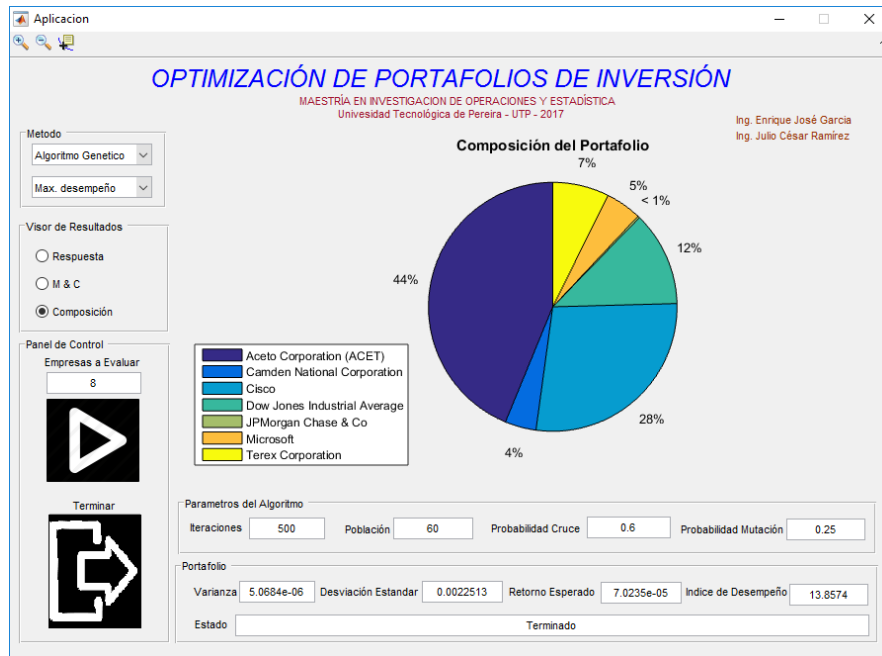
Fuente, Elaboración propia

Figura 44 Mutación y Recombinación Algoritmo Genético - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

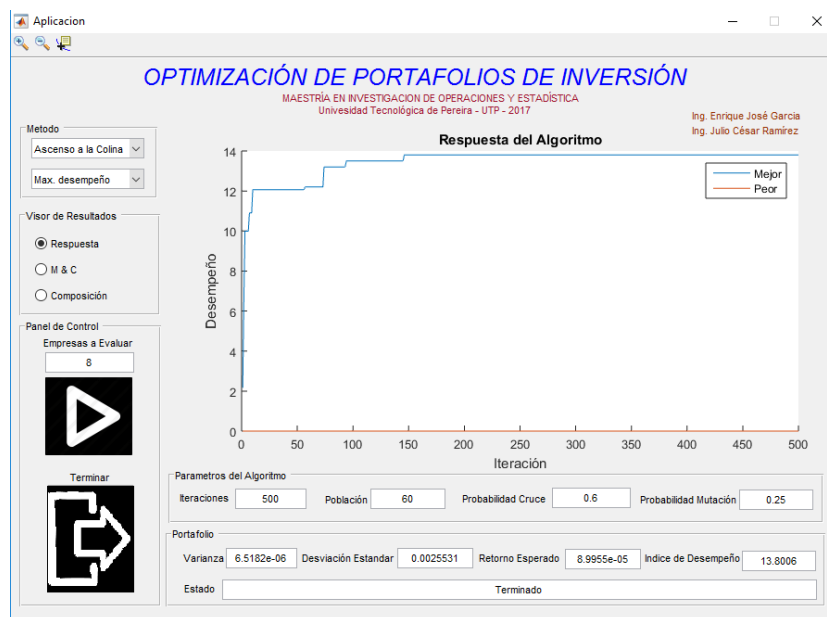
Figura 45 Respuesta Algoritmo Genético - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

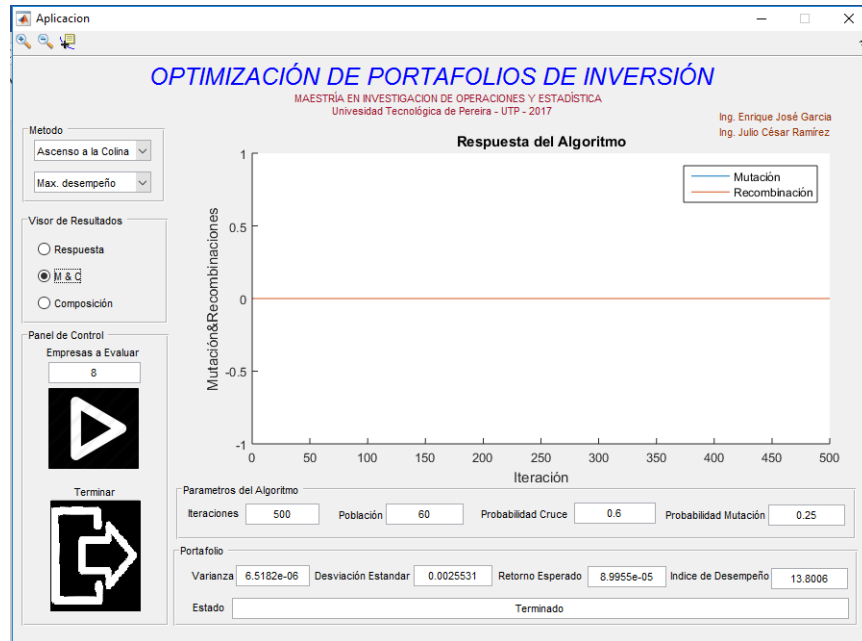
Algoritmo Ascenso A la Colina

Figura 46 Respuesta Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño



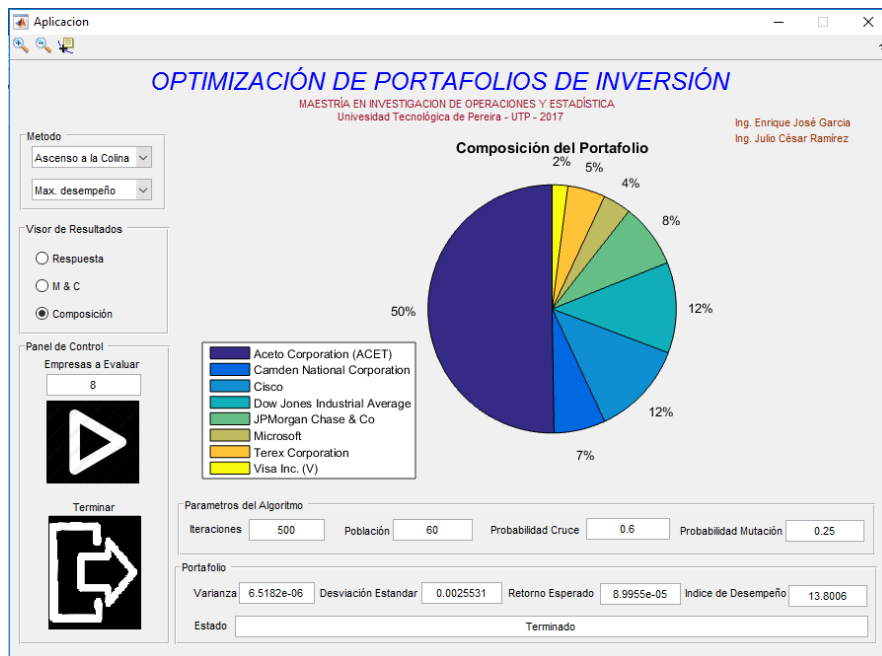
Fuente, Elaboración propia

Figura 47 Mutación y Recombinación Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Figura 48 Composición Algoritmo Ascenso A la Colina - 8 Acciones - Máximo Desempeño



Fuente, Elaboración propia

Tabla 27 Resultados Máximo Desempeño - 8 Acciones

	Máximo Desempeño			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Índice de desempeño
Algoritmo Cultural	5,0499 e-6	0,0022472	7,309 e-5	14,4752
Algoritmo Genético	5,0684 e-6	0,0022513	7,0235 e-5	13,8574
Algoritmo de Ascenso a la Colina	6,5182 e-6	0,0025531	8,9955 e-5	13,8006

Fuente, Elaboración propia

Tabla 28 Composición Portafolio de Inversión 8 Acciones – Máximo Desempeño

Portafolio - Máximo Desempeño							
Aceto Corporation (ACET)	Camden National Corporation	Cisco	Dow Jones Industrial Average	JPMorgan Chase & Co	Microsoft	Terex Corporation	Visa Inc. (V)
49%	5%	1%	27%	3%	3%	8%	4%
44%	4%	28%	12%	< 1%	5%	7%	< 1%
50%	7%	12%	12%	8%	4%	5%	2%

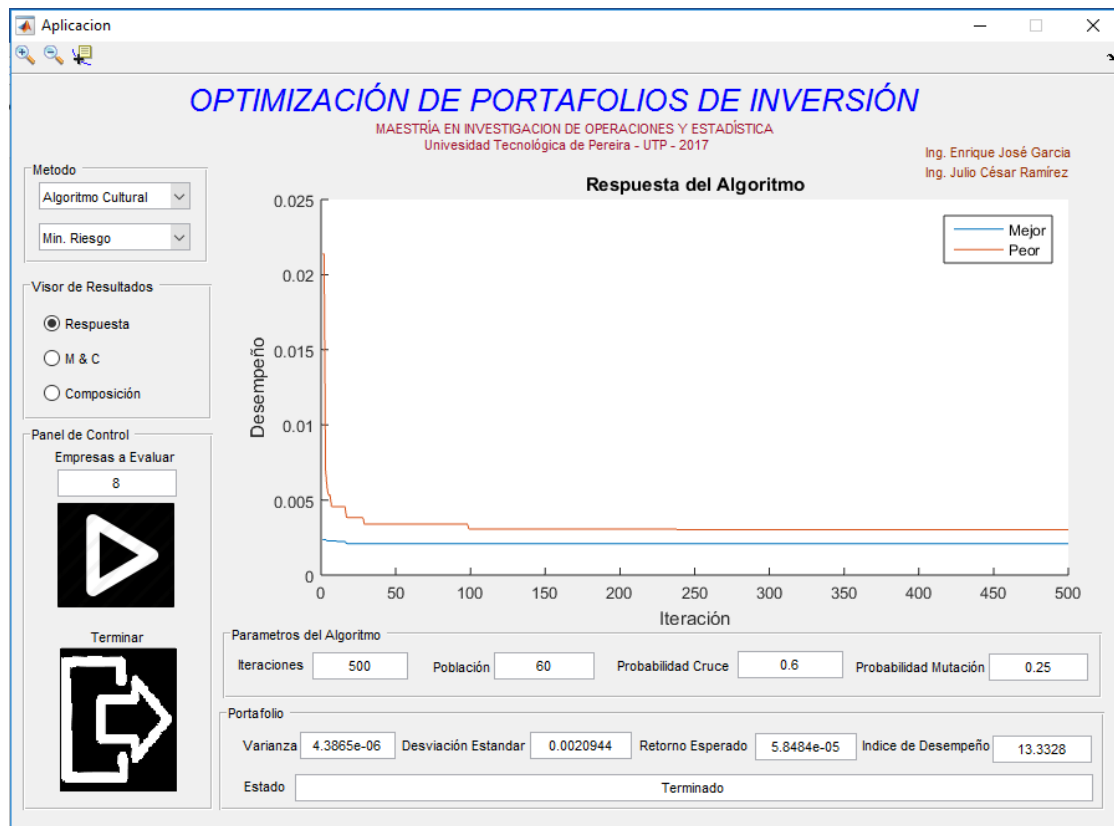
Fuente, Elaboración propia

Caso 3: 8 Acciones – Mínimo Riesgo

Se toman de manera aleatoria las acciones de Aceto Corporation (ACET), Camden National Corporation, Cisco, Dow Jones Industrial Average, JPMorgan Chase & Co, Microsoft, Terex Corporation, United Technologies Corporation (UTX), Visa Inc. (V), con los parámetros establecidos.

Algoritmo Cultural

Figura 49 Respuesta Algoritmo Cultural - 8 Acciones - Mínimo Riesgo



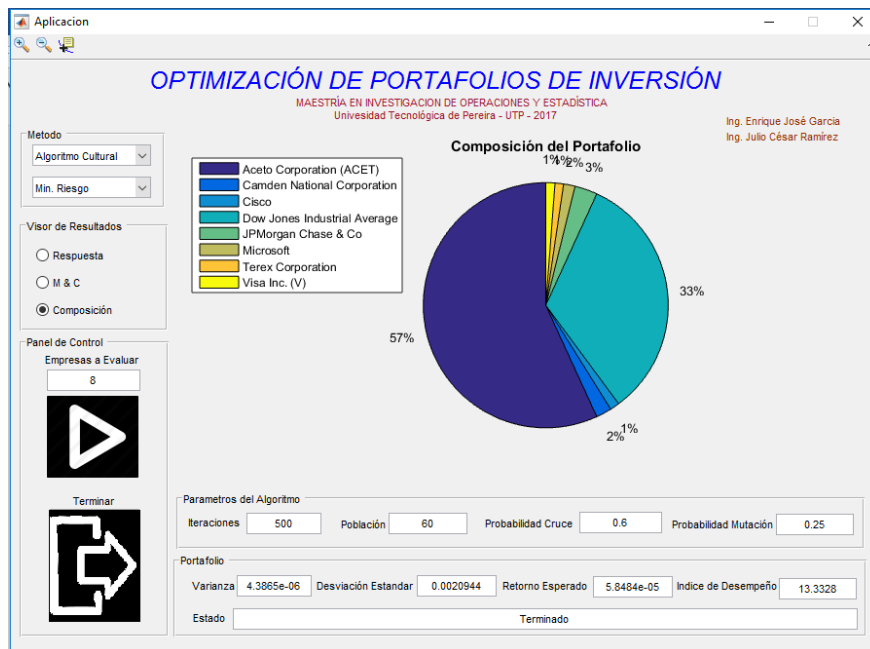
Fuente, Elaboración propia

Figura 50 Mutación y Recombinación Algoritmo Cultural - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 51 Composición Algoritmo Cultural - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

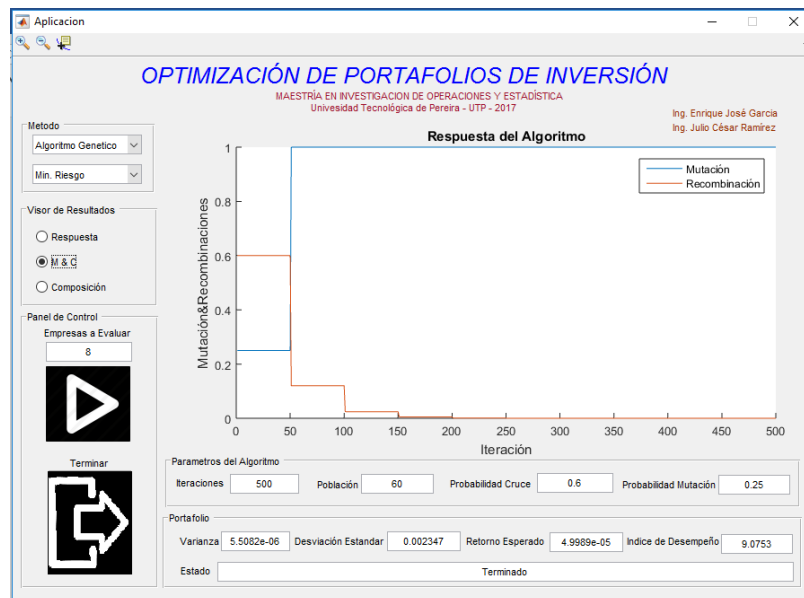
Algoritmo Genético

Figura 52 Respuesta Algoritmo Genético - 8 Acciones - Mínimo Riesgo



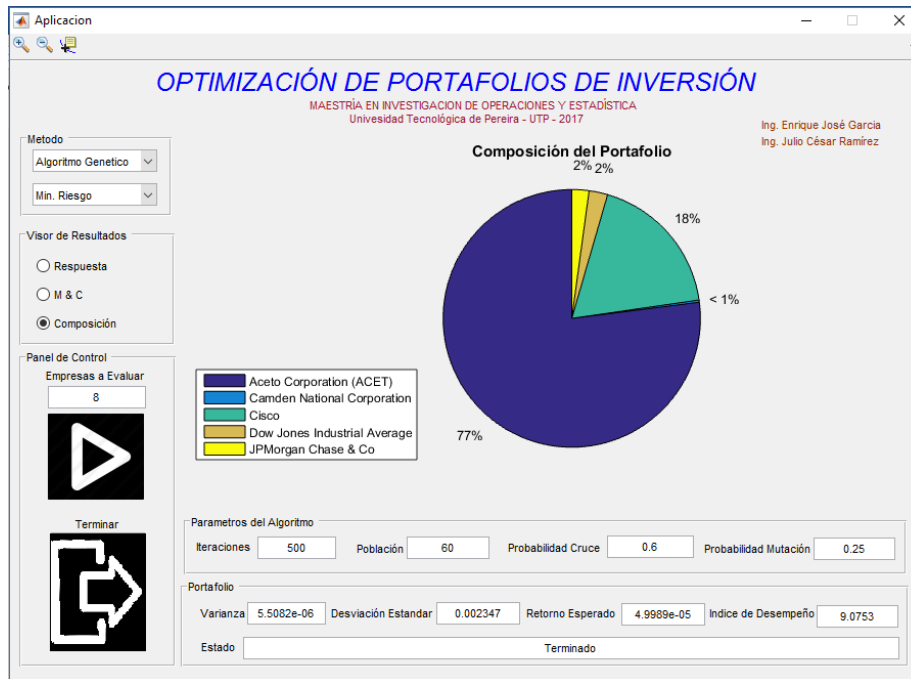
Fuente, Elaboración propia

Figura 53 Mutación y Recombinación Algoritmo Genético - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 54 Composición Algoritmo Genético - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

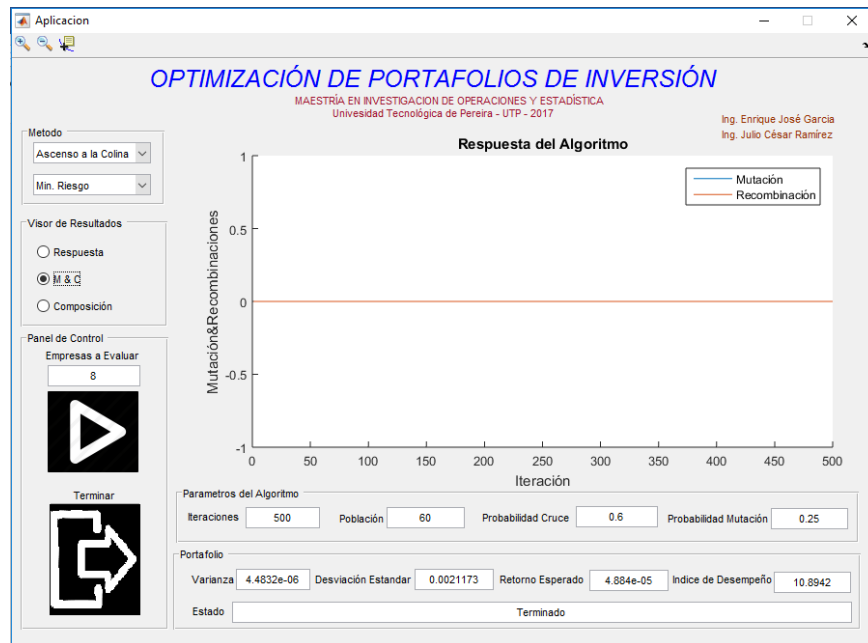
Algoritmo Ascenso A la Colina

Figura 55 Respuesta Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo Riesgo



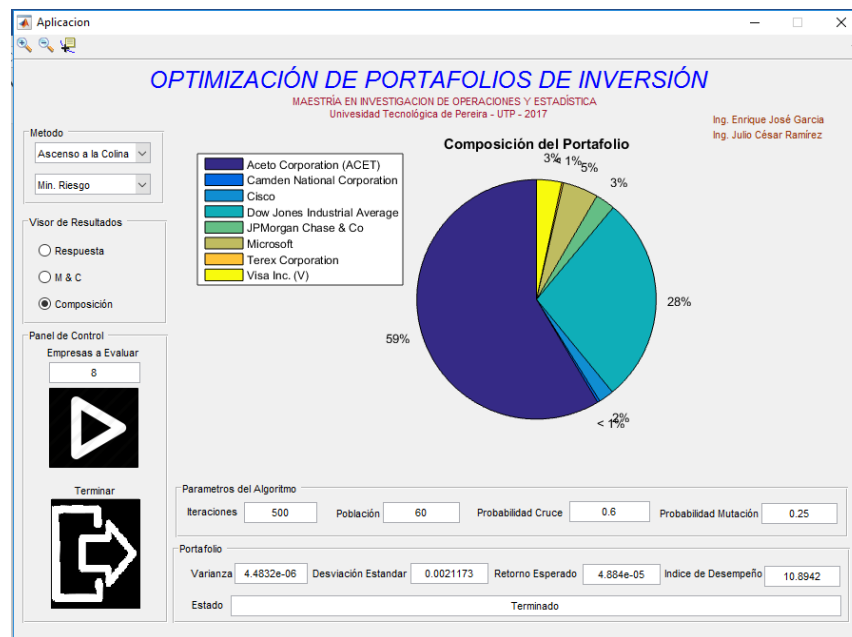
Fuente, Elaboración propia

Figura 56 Mutación y Recombinación Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Figura 57 Composición Algoritmo Ascenso A la Colina - 8 Acciones - Mínimo Riesgo



Fuente, Elaboración propia

Tabla 29 Resultados Mínimo Riesgo - 8 Acciones

	Mínimo Riesgo			
	Incertidumbre del retorno de la Inversión	Medida de riesgo de rentabilidad	Retorno Esperado	Indice de desempeño
Algoritmo Cultural	4,3865 e-6	0,0020944	5,8484 e-5	13,3328
Algoritmo Genético	5,5082 e-6	0,002347	4,9989 e-5	9,0753
Algoritmo de Ascenso a la Colina	4,4831 e-6	0,0021173	4,884 e-5	10,8942

Fuente, Elaboración propia

Tabla 30 Composición Portafolio de Inversión 8 Acciones – Mínimo Riesgo

Portafolio - Mínimo Riesgo							
Aceto Corporation (ACET)	Camden National Corporation	Cisco	Dow Jones Industrial Average	JPMorgan Chase & Co	Microsoft	Terex Corporation	Visa Inc. (V),
57%	2%	1%	33%	3%	2%	< 1%	< 1%
77%	< 1%	18,0%	2%	2%	< 1%	< 1%	< 1%
59%	< 1%	2%	28%	3%	5%	< 1%	3%

Fuente, Elaboración propia

Tabla 31 Consolidado Portafolio Inversión 8 Acciones

Portafolio - Máximo Desempeño							
Aceto Corporation (ACET)	Camden National Corporation	Cisco	Dow Jones Industrial Average	JPMorgan Chase & Co	Microsoft	Terex Corporation	Visa Inc. (V),
49%	5%	1%	27%	3%	3%	8%	4%
44%	4%	28%	12%	< 1%	5%	7%	< 1%
50%	7%	12%	12%	8%	4%	5%	2%

Portafolio - Mínimo Riesgo							
Aceto Corporation (ACET)	Camden National Corporation	Cisco	Dow Jones Industrial Average	JPMorgan Chase & Co	Microsoft	Terex Corporation	Visa Inc. (V),
57%	2%	1%	33%	3%	2%	< 1%	< 1%
77%	< 1%	18,0%	2%	2%	< 1%	< 1%	< 1%
59%	< 1%	2%	28%	3%	5%	< 1%	3%

Fuente, Elaboración propia

Análisis

Las diferentes aplicaciones que desde 1952, año en que Harry Markowitz presenta la Teoría de la Optimización de Portafolio de Inversión; siendo el principal aporte la utilidad para acumular los aspectos fundamentales que deben guiar a un inversionista racional en la elección de la composición de su portafolio; de tal forma, que le produzca la máxima rentabilidad, al controlar el riesgo; o en forma alternativa, minimizar el riesgo, controlando el rendimiento; han permitido presentar alternativas de inversión en torno a un conjunto de acciones disponibles en el mercado; autores como Herrera (Herrera, abril-junio 1999), en la aplicación del modelo de Markowitz al cálculo del valor del riesgo (VAR) a un portafolio de divisas, permite tener una aproximación a través de una explicación breve acerca de la teoría de portafolios de Markowitz, en el cálculo del valor del riesgo (VAR) a un portafolio de divisas y Kostas Metaxiotis (Kostas Metaxiotis, oct 2012) en su artículo *Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review*, que reúne el análisis de 91 documentos, en los cuáles el objetivo principal se centra en las medidas de riesgo aplicados en la formulación de modelos de gestión de cartera con el apoyo de los algoritmos evolutivos multiobjetivo (MOEA); otro autor como Landa (Becerra, Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo, 2002), propone dos algoritmos culturales: uno para optimización mono-objetivo con restricciones y otro para optimización multiobjetivo; ambos algoritmos están basados en la programación evolutiva, capaces de operar con cualquier tipo de función objetivo, que soportan la aplicación a la diversificación para portafolios de inversión.

Estos aspectos, al igual que las diferentes técnicas de computación evolutiva han mostrado que pueden ser un soporte para tener un buen desempeño en una gran variedad de problemas, pero se siguen explorando alternativas dentro de esta clase de heurísticas; si bien es cierto que

existen propuestas de aplicación de algoritmos culturales, este trabajo permitió por primera vez, que soportado en Matlab presentar la construcción y aplicación del Algoritmo Cultural propuesto por Robert G. Reynolds en 1978.

El aspecto fundamental de este algoritmo es la integración de un espacio de creencias como alternativa metodológica, para lograr mayores tasas de convergencia, diseñado para el problema de optimización del portafolio de inversión, de modo que codifique y almacene adecuadamente las experiencias de la población; conformado para este trabajo por: APORTAR, NO HACER NADA, FLIP, RENACER Y PERMUTAR; de esta forma se logró establecer un portafolio de inversión eficiente, estable y diversificado.

Se comprobó el portafolio generado a través del Algoritmo Cultural propuesto, con un Algoritmo Genético convencional y un Algoritmo de Ascenso a la Colina; el resultado de esta comparación es una propuesta para la diversificación del portafolio; lográndose al alcanzar la propuesta principal del Algoritmo Cultural mediante la incorporación del conocimiento del dominio adquirido durante el mismo proceso de búsqueda del portafolio.

El algoritmo cultural presentado, arrojó resultados comparables con el Algoritmo Genético y el Algoritmo de ascenso a la colina, logrando que los aspectos de mutación y recombinación sean acordes con las tasas de convergencia y la conformación de los portafolios; la propuesta de Algoritmo permite su funcionamiento con un número ilimitado de acciones para conformar el portafolio de Inversión.

Conclusiones

El desarrollo de este trabajo presenta una nueva manera de abordar el problema de selección de portafolio de inversión, formulado a través de la aplicación de un Algoritmo Cultural, la comparación de los resultados se hizo aplicando un Algoritmo Genético convencional y un Algoritmo de Ascenso a la Colina.

Como se puede observar a través de este documento el problema de selección de portafolio ha sido formulado y resuelto ampliamente; sin embargo, permitió abordar lo propuesto por Robert G. Reynolds en 1978, en cuanto a que los Algoritmos Culturales son un sistema dual de la herencia que caracteriza la evolución en cultura humana en el nivel macro-evolutivo, que ocurre dentro del espacio de creencias, y en el nivel micro-evolutivo, que ocurre en el espacio de población; diferenciando el hecho de que se usan cinco tipos de conocimiento básicos en el proceso de resolución de la solución del problema más que solamente el valor transmitido, siendo estos: APUESTA, NO HACER NADA, FLIP, RENACER, PERMUTAR; se puede establecer que este modelo 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.

En cuanto a los portafolios obtenidos como resultado de 2, 5 y 8 acciones son absolutamente factibles y prácticos desde el punto de vista de las mínimas diferencias obtenidas bajo los aspectos de Máximo Desempeño y Mínimo Riesgo; si se considera la aplicación del Algoritmo Cultural, del Algoritmo Genético o del Algoritmo de Ascenso a la Colina, los resultados obtenidos mejoran los que se pueden obtener con elementos menos dinámicos como podría ser el uso de una planilla

de Excel, la cual probablemente se vería seriamente afectada al momento de determinar un conjunto de portafolios como los aquí presentados.

Trabajo Futuro

Los trabajos a futuro que de este trabajo se puede desprender en el ámbito financiero y de modelos de optimización, serían:

1. El desarrollo de nuevos métodos de cómputo evolutivo, tal como se mencionan en (Erik V. Cuevas Jiménez, José V. Osuna Enciso, Diego A. Oliva Navarro, Margarita A. Díaz Cortés, 2016), así:
 - Algoritmo de optimización por enjambre de partículas (PSO)
 - Algoritmo de evolución diferencial (Differential Evolution – DE)
 - Algoritmo de optimización por búsqueda de armonías (HS)
 - Sistemas inmunes artificiales (AIS)
 - Algoritmo de optimización inspirado en principios de electromagnetismo (EMO)
 - Algoritmo colonia artificial de abejas (Artificial Bee Colony – ABC)
2. Desarrollar el enfoque alternativo de la optimización de portafolios en el espacio de retorno total-máximo drawdown; utilizando el marco conceptual y teórico propuesto por León y Laserna, (Carlos León, 2008), en lo referente con la asignación estratégica de activos para Fondos de Pensiones Obligatorias en Colombia; sustentado en que la políticas de inversión diferencia entre la definición de la composición del portafolio de referencia de largo plazo o benchmark y de los mecanismos de desviación en el corto plazo respecto a ese portafolio, en lo que se conoce como asignación estratégica de activos y asignación táctica de activos, respectivamente.

Anexos

Anexo 1 FUNCION Aplicación.m

```

%%La función Aplicación se encarga de generar la interfaz gráfica de
%%usuario y crear las condiciones iniciales de la aplicación, además de los
%%diferentes eventos de cada botón.
%%Cuando un evento es ejecutado ("Nombre_del_operador_Callback") este se
%%enlaza con las diferentes funciones contenidas dentro la carpeta de la
%%aplicación.
%%Las dos primeras funciones son definidas por defecto, con ayuda del
%%asistente de Matlab, razón por la cual no se deben editar, su uso es
%%exclusivo para la definición de las condiciones iniciales de las
%%variables globales para ser utilizadas dentro de todo el Script.

```

```

%Iniciar Programa

```

```

function varargout = Aplicacion(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Aplicacion_OpeningFcn, ...
                  'gui_OutputFcn',  @Aplicacion_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```

%Cargar Configuración Inicial del GUI

```

```

function Aplicacion_OpeningFcn(hObject, ~, handles, varargin)
global Metodo
global MN
handles.output = hObject;
guidata(hObject, handles);
set(handles.pushbutton1, 'CData', imread('Play.png'));
set(handles.pushbutton3, 'CData', imread('Salir.png'));
Metodo=1;
MN=1;

```

```

%Valores Por defecto

```

```

function varargout = Aplicacion_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```

```

%Función para el botón de Play

```

```

function pushbutton1_Callback(hObject, eventdata, handles)

```

```

%Variables Globales

```

```

global MN
global Mejor
global Peor

```

```

global Mut
global Cru
global Composicion
global Nombres
global Metodo
Empresas=str2double(get(handles.edit1, 'string'));
if (Empresas>=2)
    %Leer Parámetros del Algoritmo desde el GUI
    Iteracion=str2double(get(handles.edit7, 'string'));
    Poblacion=str2double(get(handles.edit10, 'string'));
    PC=str2double(get(handles.edit8, 'string'));
    PM=str2double(get(handles.edit9, 'string'));
    set(handles.edit2, 'string', 'Procesando Espere...');
    [F,Nombres,Retornos]=Retornos_Activos(Empresas);
    if(F==1)
        %Calcular Matriz de Covarianza
        Matriz_Covarianza=cov(Retornos);
        %Calcular Matriz de Correlación
        Matriz_Correlaciones=corrcoef(Retornos);
        %Iniciar Algoritmo Genetico

if (Metodo==2), [Parame, Composicion, Mut, Cru, Peor, Mejor]=AG(Iteracion, Poblacion,
PC, PM, Empresas, Retornos, Matriz_Covarianza, MN); end;

if (Metodo==3), [Parame, Composicion, Mut, Cru, Peor, Mejor]=Hill(Iteracion, Empresas
, Retornos, Matriz_Covarianza, MN); end;

if (Metodo==1), [Parame, Composicion, Mut, Cru, Peor, Mejor]=Cultural(Iteracion, Pobl
acion, PC, PM, Empresas, Retornos, Matriz_Covarianza, MN); end;
    pintar(hObject, eventdata, handles);
    set(handles.edit3, 'string', num2str(Parame(1)));
    set(handles.edit4, 'string', num2str(Parame(2)));
    set(handles.edit5, 'string', num2str(Parame(3)));
    set(handles.edit6, 'string', num2str(Parame(4)));
    %Exportar como archivo EXCEL

Exportar(Nombres, Retornos, Matriz_Covarianza, Matriz_Correlaciones, Parame, Compo
sicion, Mut, Cru, Peor, Mejor)
    end
    set(handles.edit2, 'string', 'Terminado')
else
    set(handles.edit2, 'string', 'El número de Empresas debe ser mayor de 2')
end

%Publicar Comportamiento del Algoritmo
function radiobutton1_Callback(hObject, eventdata, handles)
pintar(hObject, eventdata, handles);

function pintar(hObject, eventdata, handles)
global Mejor
global Peor
set(handles.axes1, 'Visible', 'on');
set(handles.axes2, 'Visible', 'off');
axes(handles.axes2)
legend('off')
title(' ')

```

```

axes(handles.axes1)
cla
legend('off')
hold on
title('Respuesta del Algoritmo')
xlabel('Iteración')
ylabel('Desempeño')
plot(Mejor)
plot (Peor)
legend('Mejor','Peor')
hold off
set(handles.radiobutton1,'Value',1);
set(handles.radiobutton2,'Value',0);
set(handles.radiobutton3,'Value',0);

%Publicar Comportamiento de la Mutación y Cruce si aplica
function radiobutton2_Callback(hObject, eventdata, handles)
global Mut
global Cru
set(handles.axes1,'Visible','on');
set(handles.axes2,'Visible','off');
axes(handles.axes2)
legend('off')
title(' ')
axes(handles.axes1)
cla
legend('on')
hold on
title('Respuesta del Algoritmo')
xlabel('Iteración')
ylabel('Mutación&Recombinaciones')
plot(Mut)
plot (Cru)
legend('Mutación','Recombinación')
hold off

%Publicar Diagrama de Pastel
function radiobutton3_Callback(hObject, eventdata, handles)
global Composicion
global Nombres
warning 'off'
axes(handles.axes1)
cla
legend('off')
set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','on');
axes(handles.axes2)
legend('on')
pie(Composicion)
title('Composición del Portafolio')
legend(Nombres,'Location','northwest')

%Seleccionar Metodo de Optimización
function popupmenu1_Callback(hObject, eventdata, handles)
global Metodo
Metodo=get(handles.popupmenu1,'Value');

```



```
%Seleccionar opción de maximizar o minimizar
function popupmenu3_Callback(hObject, eventdata, handles)
global MN
MN=get(handles.popupmenu3, 'Value');

%Boton de cerrar Aplicación
function pushbutton3_Callback(hObject, eventdata, handles)
clc
clear
close all;

%Funciones sin uso Creados por defecto para animar el GUI
function edit1_Callback(hObject, eventdata, handles)

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit5_Callback(hObject, eventdata, handles)

function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit6_Callback(hObject, eventdata, handles)

function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit4_Callback(hObject, eventdata, handles)

function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit7_Callback(hObject, eventdata, handles)

function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit8_Callback(hObject, eventdata, handles)

function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit9_Callback(hObject, eventdata, handles)

function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit10_Callback(hObject, eventdata, handles)

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function popupmenu3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

Anexo 2 FUNCION AG.m

```

%%La función algoritmo Genético "AG", se encarga de tomar el conjunto
%%población inicial (determinado por el número de empresas dentro del
%%portafolio), y realizar durante una cantidad de iteraciones determinada
%%(Generaciones), una búsqueda de la solución más adecuada al problema
%%utilizando para ello recombinaciones y mutaciones con el fin de realizar
%%explotación y exploración de las diferentes zonas de la función de
%%aptitud.
%%Como regulador de las mutaciones y cruces se encuentra integrado el
%%algoritmo de la regla de 1/5, dicho algoritmo se encarga de reducir la
%%exploración (Recombinaciones) si durante la búsqueda es necesario
%%realizar más explotación (Mutaciones), y viceversa.
%%Las recombinaciones se realizan utilizando una permutación de cada
%%individuo.
%%Y las mutaciones se realizan, realizando un random con una distribución
%%de probabilidad Gaussiana el cual se encarga de modificar dos componentes
%%dentro del individuo.

```

```

%Codigo Para Implementar un Algoritmo Genetico

```

```

function
[Par,Compo,Muta,Cruz,Peor,Mejor]=AG(Generaciones,Poblacion,PC,PM,Empresas,Ret
ornos,Matriz_Covarianza,MN)
%Contador para Actualizar la regla de 1/5
Conta=0;
%Generar Población Inicial
P0=zeros(Poblacion,Empresas);
for i=1:Poblacion
    P0(i,:)=Generar_Individuo(Empresas);
end
P1=P0;
if(MN==1)
    Norm=1;
    Col=4;
else
    Norm=0;
    Col=2;
end
for i=1:Generaciones
    %Inicia Algoritmo Genetico
    MA=Evaluar_f(Poblacion,Empresas,Retornos,P0,Matriz_Covarianza);
    %Función para Normalizar entre [0,1] los datos (Si Normalizacion(MA,0),
esta minimizando, si Normalizacion(MA,1) esta maximizando)
    Probabilidad=Normalizacion(MA(:,Col),Norm);
    %Funcion Para Selección Utilizando el método ruleta
    P1=Ruleta(Probabilidad,Poblacion,P1);
    %Se realiza un recombinación si la probablidad es mayor que el número
aleatorio
    if(rand<PC)
        a=round(random('unif',1,Poblacion,1,1));
        P1(a,:)=Cruzar(P1(a,:));
    end
end

```

```

    %Se realiza una Mutación si la probabilidad es mayor que el número
aleatorio
    if(rand<PM)
        a=round(random('unif',1,Poblacion,1,1));
        P1(a,:)=Mutar(P1(a,:));
    end
    %Reemplazo generacional STADE STATE "Si el nuevo individuo es mejor que
el padre se cambia si no NO"
    MB=Evaluar_f(Poblacion,Empresas,Retornos,P1,Matriz_Covarianza);
    for k=1:Poblacion
        if(MN==1)
            if(MB(k,Col)>MA(k,Col))
                P0(k,:)=P1(k,:);
            end
        else
            if(MB(k,Col)<MA(k,Col))
                P0(k,:)=P1(k,:);
            end
        end
    end
    end
    %Regla de 1/5 para el control de las recombinaciones y las mutaciones
    if(Conta==50)
        Factor=0.2;
        %Aqui si el peor de todos es mayor al actual aumenta las mutaciones
        %asi llevara a cabo mas exploracion para buscar individuos buenos
        if(max(Mejor)>max(MA(:,Col)))
            PM=PM/Factor;
            PC=PC*Factor;
        end;
        %Aqui si el peor de todos es menor al actual aumenta las
        %recombinaciones para llevar a cabo mas explotacion de la zona y
        %encontrar individuos buenos
        if(max(Mejor)<max(MA(:,Col)))
            PM=PM*Factor;
            PC=PC/Factor;
        end;
        %Aqui si el peor de todos es igual al actual, introduce ruido al
        %sistema para crear nuevos individuos y evitar el encierro en minimos
locales
        if(max(Mejor)==max(MA(:,Col)))
            [~,pos]=max(MA(:,Col));
            P1(pos,:)=Generar_Individuo(Empresas);
            if(rand<0.5),PM=PM/Factor;PC=PC*Factor;
            else PM=PM*Factor;PC=PC/Factor;end;
        end;
        if(PM>=1),PM=1;end;
        if(PC>=1),PC=1;end;
        Conta=0;
    end;
    %Acumulador de Valores
    if(MN==1)
        Mejor(i)=max(MA(:,Col));
        Peor(i)=min(MA(:,Col));
    else
        Mejor(i)=min(MA(:,Col));
        Peor(i)=max(MA(:,Col));
    end
end

```

```
Muta(i)=PM;
Cruz(i)=PC;
Conta=Conta+1;
end
if(MN==1)
    [~,Pos]=max(MA(:,Col));
else
    [~,Pos]=min(MA(:,Col));
end
Compo=P0(Pos,:);
Par=MA(Pos,:);
```

Anexo 3 FUNCION Generar_Indivuo.m

```
%%Para Generar un individuo se tuvo en cuenta la restricción, la cual
%%consiste en que la suma de sus componentes debe ser igual a 1.
%%Dicha función genera una suma de componentes de manera aleatoria
%%siguiendo una distribución uniforme.
```

```
function Individuo=Generar_Individuo(Empresas)
%Se genera un Individuo cuya suma sea=1 (Restricción del sistema).
Individuo=zeros(Empresas,1);
Uno=1;
for i=1:Empresas-1
    Individuo(i)=random('unif',0,Uno,1,1);
    Uno=Uno-Individuo(i);
end
Individuo(end)=1-sum(Individuo(1:end-1));
Individuo=Individuo';
```

Anexo 4 FUNCION Aceptar_Individuos.m

```
%Ordena los elementos según el resultado de la función de fitness
%(de Mayor a menor si está maximizando y de menor a Mayor si se está
%%minimizando), luego se genera un puntero de manera aleatoria el cual
%%indica cuantos individuos de la población pasaran a la siguiente
%%generación y cuantos serán rechazados.

function
[Vector_Aceptado,Aceptados]=Aceptar_Individuos(P1,MN,Individuos,Probabilidad,
Empresas)
    %Esta función ordena y selecciona los mejores individuos de la
    %población
    if(MN==1)
        A = [P1,Probabilidad];
        [~,k] = sort(A(:,Empresas+1),'descend');
        Vector_Aceptado = A(k,1:Empresas);
    else
        A = [P1,Probabilidad];
        [~,k] = sort(A(:,Empresas+1));
        Vector_Aceptado = A(k,1:Empresas);
    end
    Aceptados=round(random('unif',1,Individuos,1,1));
```

Anexo 5 FUNCION Cruzar.m

```
%Esta función realiza una permutación del vector de entrada.  
function C=Cruzar(Individuo)  
%Recombinación Basada en una permutación  
C=Individuo(randperm(length(Individuo)) );
```


Anexo 6 FUNCION Evaluar_f.m

```
%%Esta función genera una matriz de resultados basado en las operaciones de
%%la matriz de Markowitz, se retornan; Varianza, Desviación estándar,
%%Retorno esperado e índice de desempeño.
%%La matriz se genera dependiendo de la cantidad de individuos de la matriz
%%de entrada, es decir si la población es de 10, la cantidad de elementos
%%de salida será 10*4=40, en columnas ordenadas en una matriz de 10*4.

function
MA=Evaluar_f(Poblacion,Empresas,Retornos,Proporciones,Matriz_Covarianza)
%Evaluar la función (Ejecución de la Matriz de Markowitz)
for i=1:Poblacion

[Markowitz_Varianza,Markowitz_Desviacion_Estandar,Retorno_Esperado,Indice_des
empeno]=Markowitz(Empresas,Retornos,Proporciones(i,:),Matriz_Covarianza);

MA(i,:)=[Markowitz_Varianza,Markowitz_Desviacion_Estandar,Retorno_Esperado,In
dice_desempeno];
end
```

Anexo 7 FUNCION Mutar.m

```
%%Esta función toma un elemento aleatorio modifica su valor de manera
%%aleatoria siguiendo una distribución normal y para conservar la igualdad
%%(Recordando que la suma de sus valores debe ser igual a 1), toma el
%%elemento más próximo y aumenta o disminuye la diferencia con respecto a
%%la modificación hecha a su vecino.
%%Ejemplo
%%Vector=[0.1 0.2 0.3 0.4]
%%Si se toma de manera aleatoria la posición 2 y se modifica de manera
%%aleatoria por el valor 0.3, entonces para conservar la igualdad se
%%obtiene la diferencia con respecto a su valor anterior
%%diferencia=0.2-0.3=-0.1, y ese resultado se suma con su vecino mas
%%próximo, es decir la posición 3.
%%la cual quedaría Nuevo Vecino=0.3+(-0.1)=0.2
%%Como resultado el vector tendría los siguientes valores; [0.1 0.3 0.2 0.4]

function Individuo=Mutar(Individuo_Normal)
%Mutacion gausiana de una posición del vector al azar
Pos=round(random('unif',1,length(Individuo_Normal)-1,1,1));
Aux=sum(Individuo_Normal);
Aux=Aux-Individuo_Normal(Pos);
Numero_Al=random('Normal',0,Aux,1,1);
Aux=Individuo_Normal(Pos)-(Individuo_Normal(Pos)*Numero_Al);
Individuo_Normal(Pos)=(Individuo_Normal(Pos)*Numero_Al);
Individuo_Normal(Pos+1)=Aux+Individuo_Normal(Pos+1);
Individuo=Individuo_Normal;
```

Anexo 8 FUNCION Normalizacion.m

```
%%Esta función asigna una probabilidad a cada uno de los valores
%%resultado de la función de aptitud.
%%Primeramente se realiza una Z_Normalización, la cual consiste el
%%llevar todos los posibles valores de la función aptitud a uno donde
%%sean igualmente x_distantes y sean facilmente operables.
%%Seguidamente se realiza una sumatoria de todos los valores de los
%%resultados de la función de aptitud
%%y cada uno de estos valores es dividido entre el resultado de la sumatoria.
%%Así al finalizar todos los valores del vector de resultados de la
%%función de aptitud, se encontrarán normalizados con valores entre 0 y
%%1 cuya sumatoria será igual a 1.
```

```
function Prob=Normalizacion(fi,Minimo)
%Se realizo una Z normalización desplazada en 3-Sigma
fi=zscore(fi)+3;
if (Minimo==0)
    fi=1./fi;
    Prob=fi/sum(fi);
else
    Prob=fi/sum(fi);
end
```

Anexo 9 FUNCION Markowitz.m

```

%%Esta función calcula la Matriz de Markowitz a partir de los retornos
%%esperados (Los cuales ingresan a la función en un arreglo matricial,
%%calculados en la función Retornos_Activos).
%%Esta función realiza los cálculos de la varianza, Desviación estándar,
%%Retorno Esperado e Índice de desempeño, a partir de los resultados
%%proporcionados por la matriz de Markowitz, dichos valores los devuelve en
%%un arreglo vectorial.

function
[Markowitz_Varianza,Markowitz_Desviacion_Estandar,Retorno_Esperado,Indice_des
empeno]=Markowitz(Empresas,Retornos,Proporciones,Matriz_Covarianza)
Retorno_Esperado_Individual=zeros(1,Empresas);
Matriz_Markowitz=zeros(Empresas,Empresas);
%Calcular Activos Individuales
Retorno_Esperado_Individual(1,:)=mean(Retornos);
%Calcular Matriz de Markowitz
for i=1:Empresas
    A=Proporciones(i);
    for j=1:Empresas
        Matriz_Markowitz(i,j)=A*Proporciones(j);
    end
end
end
Matriz_Markowitz=Matriz_Markowitz.*Matriz_Covarianza;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Portafolio%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Markowitz_Varianza=sum(sum(Matriz_Markowitz));
Markowitz_Desviacion_Estandar=sqrt(Markowitz_Varianza);
Retorno_Esperado=sum(Retorno_Esperado_Individual.*Proporciones);
Indice_desempeno=(Retorno_Esperado/Markowitz_Varianza);

```

Anexo 10 FUNCION Retornos_Activos.m

```

%%Esta función se encarga de cargar el archivo con extensión "*.csv", y
%%extraer la información de las diferentes empresas. Dicha información
%%se dispone en un arreglo matricial y son guardadas en un variable
%%temporal.
%%También se extrae el nombre de cada empresa desde la etiqueta de cada
%%archivo y almacenada en un arreglo de caracteres de manera temporal.
%%Con la información de cada empresa se calculan los retornos esperados
%%y en caso de que la cantidad de datos entre empresas sean diferentes
%%en longitud, el programa completa los datos faltantes a partir de una
%%aproximación numérica utilizando como medida la mediana de los datos
%%anteriores.
%%Los retornos son calculados usando la siguiente formula.
%%Retorno=Valor_Ajuste (Actual)/Valor_Ajuste (Inmediatamente_Anterior)

function [Flag,N,R]=Retornos_Activos(Empresas)
    %Leer Nombre de Empresas desde el directorio
    Nombres=cell(1,Empresas);
    Flag=0;
    Long=[0,0];
    Long2=[0,0];
    for i=1:Empresas
        %Eliminar la extensión .csv de la etiqueta del directorio
        [Etiqueta,Dir] = uigetfile({'*.csv'});
        Nom=strrep(Etiqueta, '.csv', '');
        Valido=isequal(Etiqueta,0);
        if Valido==0
            Nombres(i)=cellstr(Nom);
            Precios=importdata(fullfile(Dir,Etiqueta),',');
            Precios=Precios.data;
            Long2=size(Precios);
            %%Corregir longitud de listas que entrar desde archivo
            if(Long(1)==Long2(1) || i==1)
                Datos_Empresas(:,i)=Precios(:,6);
            elseif(Long(1)<Long2(1))
                Datos_Empresas(Long(1):Long2(1),1:i-1)=median(Precios(:,6));
                Datos_Empresas(:,i)=Precios(:,6);
            elseif(Long(1)>Long2(1))
                Aux_Vec=Precios(:,6);
                Aux_Vec(end:Long(1))=median(Precios(:,6));
                Datos_Empresas(:,i)=Aux_Vec;
            end
            Long=size(Datos_Empresas(:,i));
            Long=Long(1);
            Flag=1;
        else
            Flag=0;
            N=0;
            R=0;
            break;
        end
    end
    if(Flag==1)
        L=size(Datos_Empresas);
        %Calcular Matriz de retornos;

```

```
for j=1:Empresas
    for i=1:L(1)-1
        Retornos(i,j)=(Datos_Empresas(i,j)/Datos_Empresas(i+1,j))-1;
    end
end
N=Nombres;
R=Retornos;
end
```

Anexo 11 FUNCION Ruleta.m

```
%%Esta función aplica el método ruleta a la función de aptitud, calculada
%%usando la Matriz de Markowitz para las diferentes combinaciones de
%%portafolios.
%%Ingresa un vector de valores normalizados, se genera un número aleatorio
%%siguiendo una distribución uniforme y dicho valor es buscado mediante la
%%suma acumulativa de los diferentes valores de probabilidad asignados a la
%%función de aptitud.
```

```
function Padres_ret = Ruleta(expectation,nPadres,P1)
%Vector expectativa o distribución de probabilidad
expectation = expectation(:,1);
rueda = cumsum(expectation);
Padres = zeros(1,nPadres);
Padres_ret = zeros(size(P1));
%Girar la ruleta
for i = 1:nPadres
    r = rand;
    for j = 1:length(rueda)
        if(r < rueda(j))
            Padres(i) = j;
            break;
        end
    end
end
for i=1:nPadres
    Padres_ret(i,:) = P1(Padres(i),:);
end
```

Anexo 12 FUNCION Exportar.m

```
%%Esta función genera un archivo *.xls, en el cual están contenidos los
%%parámetros de comportamiento del portafolio junto con los resultados.
%%Esta función permite guardar solamente la última corrida del algoritmo,
%%es decir si ejecuta el algoritmo con X empresas guardaría esos valores,
%%debe guardar el archivo desde el directorio si desea conservar esa copia,
%%sino el sobrescribirá los datos con la última corrida.
```

```
function
```

```
Exportar(Nombres,R,Matriz_1,Matriz_2,Parametros,Composicion,Mutacion,Cruce,Peor,Mejor)
Salida = [cellstr(Nombres)';num2cell(R)];
xlswrite('Salida',Salida,'Retornos');
Salida = [cellstr(Nombres)';num2cell(Matriz_1)];
xlswrite('Salida',Salida,'Matriz Covarianza');
Salida = [cellstr(Nombres)';num2cell(Matriz_2)];
xlswrite('Salida',Salida,'Matriz Correlacion');
Salida = [cellstr({'Varianza','Desviacion Estandar','Retorno Esperado','Indice desempeño'})';num2cell(Parametros)];
xlswrite('Salida',Salida,'Resumen Portafolio');
Salida = [cellstr(Nombres)';num2cell(Composicion)];
xlswrite('Salida',Salida,'Composicion Portafolio');
Salida = [cellstr({'Mutación','Cruce'})';num2cell([Mutacion',Cruce'])];
xlswrite('Salida',Salida,'M&C');
Salida = [cellstr({'Mejor','Peor'})';num2cell([Mejor',Peor'])];
xlswrite('Salida',Salida,'Comportamiento');
```


Anexo 13 FUNCION Cultural.m

```

%%Esta función permite ejecutar el algoritmo cultural, el cual se basa en la
%%influencia de un espacio de creencias sobre un conjunto de individuos para
%%encontrar la solución a un problema en particular.
%%Los individuos aceptados se escogen de manera aleatoria siguiendo una
%%distribución uniforme de datos normalizados.
%%Los individuos rechazados ingresan al espacio de creencias, los cuales son
%%modificados de acuerdo a la interacción con 6 operadores genéticos
%%seleccionado de manera aleatoria,
%%con acción sobre los individuos de manera ordenada y descendente.
%%Por último el reemplazo generacional se realiza aplicando una técnica de
%%STADE STATE, dicha técnica se encarga de seleccionar solamente los
%%individuos que son mejores que los que estaban inicialmente en la
%%población, sino se conserva el individuo anterior.

%Codigo Para Implementar un Algoritmo Cultural
function
[Par,Compo,Muta,Cruz,Peor,Mejor]=Cultural(Generaciones,Poblacion,PC,PM,Empres
as,Retornos,Matriz_Covarianza,MN)
%Generar Población Inicial
P0=zeros(Poblacion,Empresas);
for i=1:Poblacion
    P0(i,:)=Generar_Individuo(Empresas);
end
P1=P0;
%Selector para maximizar o minimizar la función
if(MN==1)
    Norm=1;
    Col=4;
else
    Norm=0;
    Col=2;
end
PM=0;
PC=0;
for i=1:Generaciones
    %Inicia Algoritmo Cultural
    MA=Evaluar_f(Poblacion,Empresas,Retornos,P0,Matriz_Covarianza);
    %Normaliza los valores cuya suma del fitness es 1
    Probabilidad=Normalizacion(MA(:,Col),Norm);
    %Ordena la población si esta minimizando de menor a mayor y viceversa,
    %retorna un valor de individuos aceptados
    [P1,Aceptados]=Aceptar_Individuos(P1,MN,Poblacion,Probabilidad,Empresas);
    %Función creencia determina que hacer con los individuos rechazados
    [P1,PM,PC]=Creencias(P1,Aceptados,Poblacion,PM,PC);
    %Reemplazo generacional STADE STATE "Si el nuevo individuo es mejor que
    el padre se cambia si no NO"
    MB=Evaluar_f(Poblacion,Empresas,Retornos,P1,Matriz_Covarianza);
    for k=1:Poblacion
        if(MN==1)
            if(MB(k,Col)>MA(k,Col))
                P0(k,:)=P1(k,:);
            end
        else

```

```
        if (MB(k, Col) < MA(k, Col))
            P0(k, :) = P1(k, :);
        end
    end
end
end
%Acumulador de Valores para ser graficados
if (MN==1)
    Mejor(i) = max(MA(:, Col));
    Peor(i) = min(MA(:, Col));
else
    Mejor(i) = min(MA(:, Col));
    Peor(i) = max(MA(:, Col));
end
Muta(i) = PM;
Cruz(i) = PC;
end
if (MN==1)
    [~, Pos] = max(MA(:, Col));
else
    [~, Pos] = min(MA(:, Col));
end
Compo = P0(Pos, :);
Par = MA(Pos, :);
```

Anexo 14 FUNCION Creencias.m

```

%%Esta función selecciona un operador genético al azar y modifica un
%%individuo de la población. Aplicando el operador seleccionado, la
%%selección de dicho operador se realiza mediante el método de ruleta.

function [Poblacion,M,C]=Creencias (P1,Numero_Aceptados,Individuos,PM,PC)
Poblacion=P1;
Puntero=1;
M=PM;
C=PC;
for i=Numero_Aceptados:Individuos
    %Se asigna una probabilidad a cada una de las operaciones que se puede
    %realizar sobre la población rechazada Selecciona Mediante el método
    %ruleta el operador (Ajuste del espacio de creencias)
    Operaciones=Generar_Individuo(6);
    Prob=Normalizacion(Operaciones,1);
    r=rand;
    acu=0;
    for j=1:6
        if(r<acu)
            break;
        end
        %El operador es aplicado sobre el individuo (Influencia sobre los
        %operadores)
        acu=acu+Prob(j);
        Puntero=Puntero+1;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Espacio de Creencias%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Modifica Poblacion con las creencias
    if(Puntero==1)
        %Operador de Recombinación Aleatoria
        Poblacion(i,:)=Cruzar(Poblacion(i,:));
        C=C+1;
    elseif(Puntero==2)
        %Operador de Mutación Gaussiana
        Poblacion(i,:)=Mutar(Poblacion(i,:));
        M=M+1;
    elseif(Puntero==3)
        %Operador de Recombinación Inversa
        Poblacion(i,:)=fliplr(Poblacion(i,:));
        C=C+1;
    elseif(Puntero==4)
        %Operador Apostar a un solo individuo
        Poblacion(i,:)=Apostar(Poblacion(i,:));
        M=M+1;
    elseif(Puntero==5)
        %Crear Nuevo individuo
        Poblacion(i,:)=Generar_Individuo(length(Poblacion(i,:)));
        M=M+1;
    else
        %Operador No hacer Nada
    end;
    Puntero=1;
end
End

```

Anexo 15. FUNCION Apostar.m

```
%%Esta función iguala a 1, cualquier posición del vector y las demás la  
%%igual a 0.
```

```
function C=Apostar(Individuo)  
%Escoge una posición al azar y le asigna el máximo valor  
L=length(Individuo);  
a=round(random('unif',1,L,1,1));  
for k=1:L  
    if(k~=a)  
        Individuo(k)=0;  
    else  
        Individuo(k)=1;  
    end  
end  
C=Individuo;
```

Anexo 15 FUNCION Hill.m

```

%%Código Para Implementar un Algoritmo de Ascenso a la Colina
%%El algoritmo de ascenso a la colina es un algoritmo de búsqueda aleatoria
%% el cual se basa en la generación de individuos de manera aleatoria
%% y la evaluación del mismo si es mejor que el anterior, si dicha
%%combinación
%% es mejor que la anterior, esta se conserva hasta que aparezca uno mejor,
%%así hasta que se terminen todas las iteraciones.

function
[Par,Compo,Muta,Cruz,Peor,Mejor]=Hill(Generaciones,Empresas,Retornos,Matriz_C
ovarianza,MN)
if(MN==1)
    col=4;
    P_aux=0;
    M_aux=0;
else
    col=2;
    P_aux=10;
    M_aux=10;
end
for i=1:Generaciones
    %Inicia Algoritmo Hill
    Individuo=Generar_Individuo(Empresas);
    MA=Evaluar_f(1,Empresas,Retornos,Individuo,Matriz_Covarianza);
    %Auxiliares para llenar el vector de peor y mejor valor "Maximizando"
    if((MA(:,col)>M_aux)&&(MN==1))
        M_aux=MA(:,col);
        Individuo_1=Individuo;
    end
    if((P_aux>=MA(:,col))&&(MN==1))
        P_aux=MA(:,col);
    end
    %Auxiliares para llenar el vector de peor y mejor valor "Minimizando"
    if((MA(:,col)<M_aux)&&(MN==2))
        M_aux=MA(:,col);
        Individuo_1=Individuo;
    end
    if((P_aux<=MA(:,col))&&(MN==2))
        P_aux=MA(:,col);
    end
    %Acumuladores de Valor
    Mejor(i)=M_aux;
    Peor(i)=P_aux;
    Muta(i)=0;
    Cruz(i)=0;
end
MA=Evaluar_f(1,Empresas,Retornos,Individuo_1,Matriz_Covarianza);
Compo=Individuo_1;
Par=MA;

```

Bibliografía

- Abraham Duarte Muñoz, Juan Jose Pantrigo Fernandez. (2007). *Metaheurísticas*. Madrid, España: Libros Dykinson.
- Algoritmos Genéticos, Campus Gipuzkoa*. (10 de 6 de 2015). Obtenido de <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>
- Becerra, R. L. (Noviembre de 2000). *Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Miltiobjetivo*. México D.F.
- Becerra, R. L. (2003). *Uso de Información del Dominio para Mejorar el Desempeño de un Algoritmo Evolutivo*. Mexico D.F: : Centro de Investigación y de Estudios Avanzados del IPN, Departamento de Ingeniería Eléctrica .
- Borradopedia. (20 de 12 de 2016). *Algoritmo Cultural*. Obtenido de http://borradopedia.com/index.php?title=Algoritmo_cultural
- Carlos León, J. M. (2008). *Asignación Estratégica de Activos para Fondos de Pensiones Obligatorias en Colombia: Un Enfoque Alternativo*. *Borradores de Economía, Banco de la República*, No. 523.
- Chang-Chun Lin, Y.-T. L. (Feb 2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operattional Research*, 393-404.
- Claus Aranha, H. I. (Enero de 2008). *Modelling cost into a genetic algorithm-based portfolio optimization system by seeding and objective sharing*. Obtenido de IEEE Xplore Digital Library: <http://ieeexplore.ieee.org/document/4424472/authors>
- Delfino, M. A. (1 de Abril de 2017). <http://marcelodelfino.net/files/CAPM.pdf>. Obtenido de <http://marcelodelfino.net/files/CAPM.pdf>: <http://marcelodelfino.net/files/CAPM.pdf>

- Erik V. Cuevas Jiménez, José V. Osuna Enciso, Diego A. Oliva Navarro, Margarita A. Díaz Cortés. (2016). *Optimización - Algoritmos programados con Matlab*. México: junio 2016.
- Esquivel, D. (3 de Abril de 2017). *Research Gate*. Obtenido de Research Gate: https://www.researchgate.net/publication/298464370_Algoritmos_Culturales
- Eugenia, M. (2013). *Apuntes de Regresión Lineal*. Buenos Aires. Argentina: Universidad de Buenos Aires.
- García Sánchez, Á. (25 de abril de 2017). *Técnicas Metaheurísticas*. Obtenido de www.iol.etsii.upm.es: <http://www.iol.etsii.upm.es/arch/metaheuristicas.pdf>
- Granada, M. (2013). *Algoritmos evolutivos y Técnicas Bio-Inspiradas*. Pereira: Universidad Tecnológica de Pereira.
- Hames Soleimani, H. R. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert System with Applications*, 5058-5063.
- He Ni, Yonggiao Wang. (Dic 2013). Stock Index Tracking by Pareto Efficient Genetic Algorithm. *Applied Soft Computing*, 4519-4535.
- Hernández, G. Z. (2012). *Simulación Multi-Agente mediante Algoritmos Culturales: Aplicación en Simulación de Carreras de Autos*. México D.F.: Centro de Investigaciones y de estudios avanzados del Instituto Politecnico Nacional.
- Herrera, F. L. (abril-junio 1999). Aplicación del enfoque de Markowitz al cálculo del Valor en Riesgo (VaR) a un portafolio de divisas. *Revista Contaduría y Administración No.193*, 53-60.

- Hossein Dastkhan, Naser Sams Gharneh, Hamid Reaz Golmakani. (Sep 2011). A Linguistic-Base Portfolio Selection Model Using Weighted Max-Min Operator and Hybrid Genetic Algorithm. *Expert Systems with Applications*, 11735-11743.
- Jorge Arranz de la Peña, Antonio Truyol. (5 de 10 de 2016). *Algoritmos Genéticos*. Obtenido de <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf>
- Kostas Metaxiotis, K. L. (oct 2012). Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literatura review. *Expert Systems with Applications*, 11865-11698.
- Kyong Joo Oh, Tae Yoon Kim, Sungky MIn. (Feb 2005). Using Genetic Algorithm to Support Portfolio Optimization For Index Fund Management. *Expert Systems with Applications*, 371-379.
- López, C. (2008). Mercado de Capitales y Gestión de Cartera. Buenos Aires, Buenos Aires, Argentina.
- López, C. (1 de 4 de 2017). http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf. Obtenido de http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf:
http://marcelodelfino.net/files/Teora_de_la_Cartera.pdf
- Luis C. Franco Arbelaez, C. T. (Junio de 2011). Modelo de Markowitz y Modelo Black-Litterman en la Optimización de Portafolios de Inversión. *Revista Tecno Lógicas No. 26*, 71-88.
- Mauricio Gutierrez Urzúa, E. T. (2014). Portfolio Optimization Using a Micro Genetic Algorithm. *UNMSM-Revistas*.
- Medina, L. Á. (2033). Aplicación de la teoría del portafolio en el mercado accionario Colombiano. *Scielo - Cuadernos de Economía*, Vol. 22 No.39.

- Natalli Macedo Rodriguez, Carolina Paula de Almeida, Richard A. Gonzalez. (2007). Um Algoritmo Cultural para o Problema do Despacho Economico e Ambiental. *XIV ERI-PR - Universidad Estadual do Centro-Oeste - Brasil*.
- Pose, M. G. (5 de 11 de 2017). *Introducción a los Algoritmos Genéticos*. Obtenido de <http://sabia.tic.udc.es/mgestal/cv/AAGGtutorial/node10.html>
- Tak-Chung Fu, C.-P. C.-L. (Dic 2013). Adopting genetic algorithms for technical analysis and portfolio management. *Computers & Mathematics with Applications*, 1743-1757.
- Tun-Jen Chang, S.-C. Y.-J. (Sep 2009). Portfolio Optimization Problems in Different Risk Measures Using Genetic Algorithm. *Expert Systems with Applications*, 10529-10537.
- Universidad EAFIT. (Sep de 2007). Mercado de Capitales. *Boletín 63*. Medellín.
- Wikipedia. (14 de 12 de 2016). *Algoritmo Genético*. Obtenido de https://es.wikipedia.org/wiki/Algoritmo_gen%C3%A9tico
- Zezzatti, A. O. (2008). Algoritmos Genéticos. *Gaceta Ideas CONCYTEG Año 3*.