

UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR

Proyecto Fin de Carrera

# **Análisis de Técnicas Evolutivas para la Estimación de Curvas de Tipos de Interés**

Autor: Carlos Carrero Yubero

Directores: David Quintana Montero

Alejandro Cervantes Rovira

Octubre de 2011



# Resumen

---

La curva de tipos de interés trata de reflejar una estructura temporal continua de la relación entre rendimiento y plazo hasta vencimiento de los bonos para un día determinado a partir de un número finito de datos. Para ello hay que establecer una hipótesis sobre la forma funcional entre tipos de interés y plazo.

La mayoría de los bancos centrales se inclinan por el modelo de Nelson y Siegel (1987) o la versión extendida de Svensson (1995). Estas funciones constan de una serie de parámetros que hay que determinar para ajustar la curva a los datos reales.

En el caso del modelo de Svensson, la función consta de seis parámetros a determinar, y la función a minimizar para realizar el ajuste suele ser el error entre los precios ajustados por el modelo y los precios reales de los títulos. Para realizar este ajuste, necesitamos un algoritmo de optimización que explore el dominio de búsqueda.

Una de las cuestiones técnicas más importantes a la hora de ajustar la función de Svensson es la posibilidad de que los parámetros obtenidos correspondan a un óptimo local, es decir, el riesgo de falsa convergencia. Este problema deriva de la elevada no-linealidad de la función, así como de la alta sensibilidad de los resultados ante las condiciones iniciales con los algoritmos de optimización comúnmente utilizados.

La mayor parte de métodos tradicionales de optimización se mueven de un punto a otro del dominio de posibles soluciones utilizando reglas determinísticas. Uno de los problemas de estos métodos es la posibilidad de quedar estancado en un óptimo local.

Frente a esto, los paradigmas de la computación evolutiva utilizan una población de puntos para explorar el dominio. Habitualmente producen una nueva población con el mismo número de individuos en cada generación. De esta manera muchos posibles óptimos son explorados simultáneamente, disminuyendo la probabilidad de falsa convergencia.

El objetivo de este proyecto es explorar el comportamiento en el dominio de una serie de algoritmos propios de la computación evolutiva y compararlo con el de algunos métodos tradicionales. Para ello utilizamos algoritmos genéticos y estrategias evolutivas, así como métodos de optimización no-lineal para problemas con restricciones.

**Palabras clave:** Computación Evolutiva, Estrategias Evolutivas, Algoritmos Genéticos, Curva de Tipos de Interés, Nelson Siegel y Svensson, Estructura Temporal de los Tipos de Interés.



# Abstract

---

A yield curve is a continuous estimation of the relationship between the nominal interest rates of default-free zero-coupon bonds and time to maturity, based on a finite number of data. For this aim we need to establish an hypothesis about the functional form between interest rates and time to maturity.

Most central banks use the Nelson and Siegel model (1987) or the extended version of Svensson (1995). These functions have some parameters that need to be determined in order to adjust the curve to real data.

The extended version of Svensson has six parameters to determine, and the objective function to minimize could be the error between real prices and adjusted prices. In order to estimate these parameters, an optimization algorithm is needed to explore the domain of the variables.

One of the most important technical questions when it comes to estimating the parameters of the NSS function, is the possibility of obtaining a local optimum, i.e. the risk of false convergence. This problem is due to the high non-linearity of the function, and the great variability of the estimation with the traditional algorithms based on derivatives.

Most of the traditional optimization methods move from one point of the domain to another using deterministic rules. One of the consequences of this method is the possibility of getting stuck in a local optimum.

On the other hand, evolutionary computation paradigms use a population of points to explore the domain, generally producing a new population with the same number of individuals in every generation. Consequently, many possible optimums are explored simultaneously, reducing the risk of false convergence.

The objective of this project is to analyze the behavior of some evolutionary computation paradigms in the domain of the problem, and compare it to the behavior of the traditional methods. For this aim we used genetic algorithms and evolution strategies, and different constrained non-linear optimization methods.

**Keywords:** Evolutionary Computation, Evolution Strategy, Genetic Algorithm, Yield Curve, Nelson Siegel and Svensson, Term Structure of Interest Rates.



# Índice General

---

<b>1 INTRODUCCIÓN</b> .....	<b>9</b>
1.1 CONTEXTO .....	9
1.2 OBJETIVOS .....	11
1.3 FASES DE DESARROLLO .....	11
1.4 ESTRUCTURA DEL DOCUMENTO .....	13
<b>2 ESTADO DEL ARTE</b> .....	<b>15</b>
2.1 ESTRUCTURA TEMPORAL DE LOS TIPOS DE INTERÉS.....	15
2.2 LA FUNCIÓN DE NELSON-SIEGEL-SVENSSON.....	20
2.3 ESTIMACIÓN DE LOS PARÁMETROS DE LOS MODELOS DE NELSON Y SIEGEL Y SVENSSON .....	24
<b>3 ENFOQUE DEL PROBLEMA</b> .....	<b>27</b>
3.1 PLANTEAMIENTO DEL PROBLEMA .....	27
3.2 ALGORITMOS DE BÚSQUEDA.....	30
3.3 MEDIOS EMPLEADOS .....	33
<b>4 MÉTODOS TRADICIONALES</b> .....	<b>41</b>
4.1 INTERIOR POINT .....	42
4.3 ACTIVE SET.....	50
4.4 SQP .....	54
<b>5 ALGORITMOS GENÉTICOS</b> .....	<b>59</b>
5.1 INTRODUCCIÓN .....	59
5.2 CONFIGURACIÓN DE LOS PARÁMETROS DEL ALGORITMO. ....	61
5.3 EXPERIMENTACIÓN .....	70
<b>6 ESTRATEGIAS EVOLUTIVAS</b> .....	<b>73</b>
6.1 INTRODUCCIÓN .....	73
6.2 $(\mu,\lambda)$ -EE.....	75
6.3 COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY (CMA-ES).....	82
<b>7 COMBINACIÓN DE (3,60)-EE Y SQP</b> .....	<b>93</b>
7.1 INTRODUCCIÓN .....	93
7.2 CONFIGURACIÓN DE LOS PARÁMETROS DEL ALGORITMO .....	96
7.3 EXPERIMENTACIÓN .....	96
<b>8 EVALUACIÓN DE LOS RESULTADOS</b> .....	<b>97</b>
8.1 COMPARACIÓN DE LOS RESULTADOS GLOBALES .....	98
8.2 COMPARACIÓN DE LOS RESULTADOS DIARIOS.....	103
<b>9 CONCLUSIONES Y LÍNEAS FUTURAS</b> .....	<b>109</b>
9.1 CONCLUSIONES .....	109
9.2 LÍNEAS FUTURAS .....	110
<b>10 PRESUPUESTO</b> .....	<b>113</b>
10.1 FASES DEL PROYECTO .....	113
10.2 COSTES .....	115

# Índice de Figuras

---

Figura 1: Curva de tipos de interés de la Eurozona. _____	16
Figura 2: Soluciones de la función de Nelson y Siegel. _____	21
Figura 3: Soluciones de la función de Svensson. _____	23
Figura 4: Método de la ruleta _____	62
Figura 5: Selección estocástica universal _____	62
Figura 6: Recombinación intermedia _____	64
Figura 7: Recombinación lineal _____	65
Figura 8: Recombinación lineal extendida _____	66
Figura 9: Camino de evolución de la media _____	85
Figura 10: Resultados diarios SQP II, EE III _____	94
Figura 11: Resultados diarios SQP I, EE I. _____	95
Figura 12: Resultados globales. _____	100
Figura 13: Análisis de la significación de los resultados. _____	101
Figura 14: Medias diarias del error cuadrático. _____	104
Figura 15: Medias diarias del error cuadrático (orden media global $\leq 10^{-4}$ ) _____	105
Figura 16: Medias diarias del error cuadrático (SQP y (3,60)-EE) _____	106
Figura 17: Medias diarias del error cuadrático (SQP, EE y su combinación) _____	107
Figura 18: Diagrama de Gantt _____	114



# Capítulo 1

## Introducción

### 1.1 Contexto

Dada una cantidad de dinero y un plazo o término para su devolución o su uso, el tipo de interés indica qué porcentaje de ese dinero se obtendría como beneficio, o en el caso de un crédito, qué porcentaje de ese dinero habría que pagar. Se trata de un índice para medir la rentabilidad del dinero, generalmente expresado como porcentaje.

La función que desempeñan los tipos de interés en la asignación de fondos en los mercados financieros es análoga al papel que juegan los precios en la asignación de recursos en los mercados de bienes y servicios, es decir, son señales que sirven a los agentes económicos en la toma de decisiones sobre consumo, inversión y financiación.

La teoría económica sugiere que un importante factor que explica la discrepancia en los tipos de interés entre dos instrumentos financieros con características similares tales como riesgo, régimen fiscal, emisor y mercado, es la diferencia entre las fechas de vencimiento de cada uno de éstos. Esta relación entre la madurez de los instrumentos y sus tipos de interés de mercado es conocida como estructura temporal de los tipos de interés (ETTI).

La ETTI, para un punto en el tiempo, puede ser representada utilizando un diagrama que relaciona el rendimiento de estos instrumentos con su fecha de vencimiento conocido como curva de tipos de interés. Las curvas de tipos de interés pueden presentar una amplia variedad de formas y movimientos, cada una de los cuales aporta una explicación en sentido

económico y financiero. Los agentes económicos tratan de pronosticar dichos movimientos y formas con el fin de anticiparse al mercado buscando obtener los mayores beneficios. Para ello hay que establecer una hipótesis sobre la forma funcional entre tipos de interés y plazo.

La mayoría de los bancos centrales se inclinan por el modelo de Nelson y Siegel (1987) o la versión extendida de Svensson (1995). Estas funciones constan de una serie de parámetros que hay que determinar para ajustar la curva a los datos reales.

En el caso del modelo de Svensson, la función consta de seis parámetros a determinar, y la función a minimizar para realizar el ajuste suele ser el error entre los precios ajustados por el modelo y los precios reales de los títulos. Para realizar este ajuste, necesitamos un algoritmo de optimización que explore el dominio de búsqueda.

Una de las cuestiones técnicas más importantes a la hora de ajustar la función de NSS es la posibilidad de que los parámetros obtenidos correspondan a un óptimo local, es decir, el riesgo de falsa convergencia. Este problema deriva de la elevada no-linealidad de esta función, así como de la alta sensibilidad de resultados ante condiciones iniciales con los algoritmos de optimización comúnmente utilizados.

La mayor parte de métodos tradicionales de optimización se mueven de un punto a otro del dominio de posibles soluciones utilizando reglas determinísticas. Uno de los problemas de estos métodos es la posibilidad de quedar estancado en un óptimo local.

Frente a esto, los paradigmas de la computación evolutiva utilizan una población de puntos para explorar el dominio. Habitualmente producen una nueva población con el mismo número de individuos en cada generación. De esta manera muchos posibles óptimos son explorados simultáneamente, disminuyendo la probabilidad de falsa convergencia.

## 1.2 Objetivos

Partiendo de la premisa de la sensibilidad ante las condiciones iniciales a la hora de calcular los parámetros de la función de NSS con los métodos tradicionales, proponemos la utilización de técnicas evolutivas para solucionar los problemas de falsa convergencia.

El objetivo de este proyecto es hacer un estudio exploratorio del comportamiento en el dominio de una serie de algoritmos propios de la computación evolutiva:

- Algoritmos genéticos.
- Estrategias evolutivas.

y compararlo con el de algunos métodos tradicionales:

- Algoritmos de programación secuencial cuadrática.
- Algoritmos de punto interior.

## 1.3 Fases de Desarrollo

Las fases de desarrollo del proyecto comprenden:

### 1. Estudio previo:

- Estudio del contexto.
- Análisis del problema.

### 2. Selección del instrumental:

- Búsqueda y selección de librerías de computación evolutiva.
- Búsqueda y selección de software de programación no-lineal.
- Selección del entorno de trabajo.

### 3. Trabajo experimental:

- Configuración de los parámetros de los algoritmos.
- Experimentación.

### 4. Análisis de los resultados y documentación: redacción de la memoria paralelamente al análisis de los resultados .

A continuación describimos algunas partes relevantes para comprender la estructura del proyecto.

## **Búsqueda y selección de librerías de computación evolutiva y software de programación no lineal**

En esta fase realizamos un estudio previo a la selección del entorno de trabajo para la implementación de las técnicas y el trabajo experimental.

Las herramientas estudiadas y criterios de selección están descritos en el apartado 3.3 *Medios Empleados*.

## **Configuración de los parámetros de los algoritmos**

En esta fase previa a la experimentación, realizamos pruebas exploratorias para seleccionar los operadores y parámetros que configuran cada técnica.

El procedimiento experimental ha variado en función de la técnica y parámetros utilizados en la experimentación.

Con los algoritmos tradicionales, al no tratarse de métodos estocásticos, sólo hizo falta una ejecución para estimar la mejor opción, mientras que en las técnicas evolutivas realizamos varias ejecuciones para utilizar la media y desviación típica del error cuadrático como criterios de selección.

Realizamos una serie de pruebas preliminares de tipo exploratorio para determinar los parámetros de los algoritmos.

Una vez fijados determinados parámetros del algoritmo, realizamos las pruebas más relevantes con mayores tamaños de población para que se asemejara más a la configuración final, lo que aumenta mucho el gasto computacional. Con el fin de acotar el tiempo de ejecución, realizamos menor número de ejecuciones en estos casos.

Los diferentes procedimientos están explicados con mayor detalle en los apartados correspondientes a la experimentación de cada algoritmo.

## **Experimentación**

Para llevar a cabo la fase experimental hemos utilizado métodos diferentes en función del tipo de técnica (estocásticas/no estocásticas).

En los algoritmos tradicionales, al no tratarse de métodos estocásticos, ha consistido en una ejecución sobre un rango de treinta días utilizando dos métodos distintos que se explican en el apartado 4.1.3.

En las técnicas evolutivas ha consistido en 10 ejecuciones sobre un rango de treinta días utilizando tres métodos diferentes que se describen en el apartado 5.3.

## 1.4 Estructura del Documento

Este proyecto se divide en diez capítulos, siendo este primero una introducción al enfoque y desarrollo del mismo, exponiendo brevemente el contexto, causas y objetivos, así como las fases de desarrollo del proyecto y la estructura de la memoria.

En el segundo capítulo intentamos ahondar en el contexto, presentando una descripción detallada, así como la evolución y estado de los temas de estudio.

En el tercer capítulo exponemos el planteamiento del problema y los medios empleados, así como una introducción a las diferentes técnicas utilizadas: métodos tradicionales y técnicas evolutivas.

En los capítulos cuatro, cinco, seis y siete se presenta el trabajo realizado con cada una de las técnicas utilizadas, dividiendo cada una en una sección introductoria a la técnica, una para la configuración de los parámetros del algoritmo y otra para la experimentación final.

En el capítulo ocho analizamos los resultados obtenidos comparando las medias globales y diarias, y estudiando la significación estadística de los resultados.

El capítulo nueve contiene un resumen de las conclusiones alcanzadas, así como una breve propuesta de líneas futuras de estudio.

Los últimos apartados corresponden a las referencias, presupuesto del proyecto y un anexo con los resultados desagregados.



# Capítulo 2

## Estado del Arte

### 2.1 Estructura Temporal de los Tipos de Interés

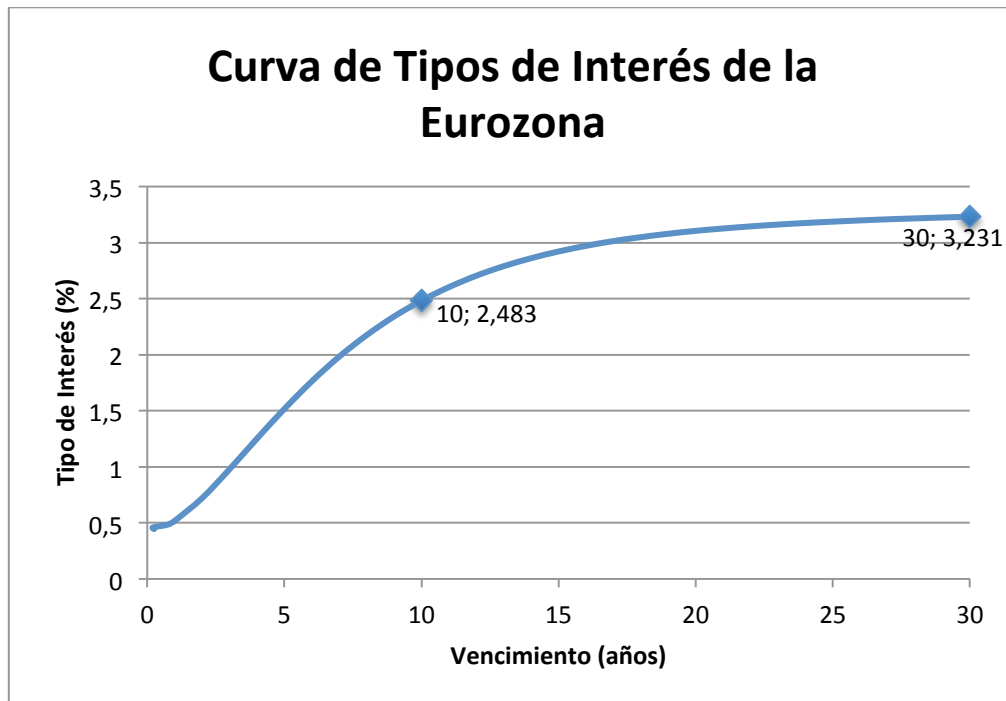
Como ya explicamos en la introducción al contexto, la función que desempeñan los tipos de interés en la asignación de fondos en los mercados financieros es análoga al papel que juegan los precios en la asignación de recursos en los mercados de bienes y servicios, es decir, son señales que sirven a los agentes económicos en la toma de decisiones sobre consumo, inversión y financiación. Cuando existe un precio relativamente alto de un bien en particular se tiende a destinar recursos a su producción, de la misma manera, cuando existe una tasa de interés relativamente alta de algún instrumento en particular se destinan fondos a las actividades que financia éste.

La teoría económica sugiere que un importante factor que explica la discrepancia en los tipos de interés entre dos instrumentos financieros con características similares tales como riesgo, régimen fiscal, emisor y mercado, es la diferencia entre las fechas de vencimiento de cada uno de éstos. Esta relación entre la madurez de los instrumentos y sus tipos de interés de mercado es conocida como estructura temporal de los tipos de interés (ETTI).

La ETTI, para un punto en el tiempo, puede ser representada utilizando un diagrama que relaciona el rendimiento de estos instrumentos con su fecha de vencimiento conocido como curva de tipos de interés. Las curvas de tipos de interés pueden presentar una amplia variedad de formas y movimientos, cada una de los cuales aporta una explicación en sentido

económico y financiero. Los agentes económicos tratan de pronosticar dichos movimientos y formas con el fin de anticiparse al mercado buscando obtener los mayores beneficios.

**Figura 1:** Curva de tipos de interés de la Eurozona.



En la figura 1 se presenta la curva de tipos para la Eurozona a 14 de septiembre de 2011<sup>1</sup>. En el eje de abscisas se muestra el plazo o vencimiento, y en el eje de las ordenadas los correspondientes tipos de interés vigentes en el momento de la elaboración de la curva. Atendiendo a lo anterior, el gráfico nos informa de que, en el día señalado, el rendimiento obtenido por un inversor por la compra y el mantenimiento hasta su vencimiento de un título de deuda pública a 10 años sería del 2,483 %, rentabilidad que se elevaría hasta el 3,231 % si el título de deuda adquirido fuera a 30 años.

Las curvas de tipos de interés generalmente muestran pendiente positiva, cuanto mayor es el tiempo de madurez de un instrumento, mayor se espera que sea su rendimiento. Esto se debe principalmente a que entre mayor es el tiempo de vencimiento de un instrumento mayores riesgos enfrenta el inversor. En otras palabras, se incrementan las posibilidades de que ocurra algún evento catastrófico que impacte sobre la inversión, por lo que es necesaria una prima que compense el riesgo al que se encuentra expuesto el inversor.

De igual manera, puede darse el caso en el que los tipos de interés de corto plazo sean superiores a las de largo plazo lo cual implica una pendiente negativa de la ETTI. Esta situación anormal y contradictoria ocurre cuando los inversores esperan que los rendimientos en el futuro sean menores a los de ahora, esto debido a que visualizan una

<sup>1</sup> Datos obtenidos de la página web del Banco Central Europeo.



posible recesión económica futura o por el contrario, implica que el mercado cree que la inflación permanecerá baja.

Se puede pensar pues, que la ETTI muestra un menú de precios de consumo futuro e inversión a distintos plazos. Es evidente, que los agentes económicos que ofertan en los mercados financieros tienen el propósito de posponer sus decisiones de consumo presente por consumo futuro, recibiendo a cambio, un tipo de interés que compense su espera. Por otra parte, algunos agentes pueden demandar esos fondos con el fin de utilizarlos en el desarrollo de proyectos de inversión que les permitan pagar el tipo de interés que los ahorradores demandan, de tal manera que diferentes percepciones acerca de los rendimientos de los proyectos en el futuro deberán verse reflejados en distintos tipos de interés según los plazos de cada uno de éstos.

## Aplicaciones de la ETTI

La ETTI es una importante herramienta en materia económica y financiera, ya que es capaz de proporcionar información de las condiciones macroeconómicas futuras de un país. Dentro de la información que transmite la ETTI encontramos expectativas sobre tipos de interés futuros, actividad económica, inflación y efectividad de la política monetaria.

Asimismo, el conocimiento de la ETTI contribuye en la evaluación de proyectos de inversión, valuación de activos e instrumentos financieros como productos derivados que impliquen la utilización de tipos de interés tales como contratos forwards, futuros o swaps.

## Construcción de la curva de tipos de interés

Se podría observar directamente del mercado de deuda pública una estructura temporal continua dada, en el caso que hubiese para cada plazo un título de cupón cero sin riesgo de crédito. No obstante, solo se dispone de un número finito de títulos y sus precios definen un número limitado de puntos; con estos datos se podría construir una curva de tipos de interés. Sin embargo, el primer problema es que no se observan tipos spot<sup>2</sup> directamente, es decir, no se dispone de precios obtenidos de operaciones simples, de modo que hay que estimarlos. La mayoría de los títulos que cotizan en el mercado pagan cupones periódicamente. Además, los tipos observados incluyen efectos como riesgo de crédito, fiscalidad, riesgo de liquidez, entre otros.

Tanto en el ámbito académico como profesional, se realizan estimaciones a partir de los precios de títulos de deuda pública u otros con características similares. Se asume que los títulos de deuda pública de los países desarrollados no presentan riesgo de crédito y, el conjunto de plazos negociados es bastante amplio; aunque, en determinados países no existen suficientes datos para cubrir todos los plazos.

Para estimar una estructura temporal continua con el número limitado de datos con que contamos hay que establecer una hipótesis sobre la relación funcional entre el tipo de interés y el plazo.

---

<sup>2</sup> Los tipos spot o al contado son los tipos vigentes en el mercado en un momento dado para un plazo determinado.

Existen diversos modelos para estimar la curva de tipos de interés. La mayor parte de ellos se pueden dividir en dos grupos: Los que buscan caracterizar la curva de tipos de interés con un conjunto de parámetros reducido generando curvas parsimoniosas, llamados métodos paramétricos; y los métodos que buscan un mayor ajuste utilizando polinomios por intervalos, llamados métodos no paramétricos o de splines.

Los **modelos paramétricos** permiten replicar la forma funcional de la curva a partir de una muestra de datos, ajustando para ello una serie de parámetros que minimicen el error entre los datos ajustados y los reales.

Los más utilizados son los propuestos por **Nelson y Siegel** (1987) y la ampliación de **Svensson** (1994), aplicada en este proyecto.

Los **modelos no paramétricos o de splines** dividen los datos observados de los rendimientos o precios en segmentos, y se ajusta un polinomio a cada uno uniéndose entre sí de manera suavizada igualando las derivadas en cada nodo.

El trabajo pionero de este enfoque es el de **McCulloch** (1971), quien primero utilizó polinomios cuadráticos para la estimación de curvas de tipos de interés, aplicando en 1975 funciones cúbicas para mejorar tanto la flexibilidad de la curva como la forma de las curvas a plazo. Los splines cúbicos sirven como base para todos los métodos de splines encontrados en la literatura para la estimación de la estructura de tipos de interés.

Shea (1984) y Steeley (1991) recomiendan el uso de **b-splines**. Se trata de funciones a partir de las cuales se genera una base del espacio de splines, con lo que cualquier curva cúbica a trozos puede ser representada como combinación lineal de ellas.

Los **smoothing splines** son splines cúbicos que incorporan además una función de penalización por variabilidad de la curva. Estos nacieron ante la necesidad de algunos bancos centrales de contar con métodos que se ajustaran mejor a los datos que los métodos parsimoniosos, pero que a su vez mantuvieran estabilidad en el largo plazo. Los estudios realizados con este método se atribuyen a bancos centrales en EEUU e Inglaterra como se ve en los trabajos de Fisher, Nychka y Zervos (1995), Waggoner (1997) y Anderson y Sleath (2001).

Los **splines exponenciales** son una variante de los splines introducida por Vasicek y Fong (1982), que aplicaron un spline a una transformación exponencial del vencimiento.

Los **modelos dinámicos**, a diferencia de los modelos descritos, que son métodos de ajuste a los datos observados, estiman la estructura de tipos *spot* asumiendo una relación teórica entre los tipos a corto plazo y el resto de tipos mediante una función diferencial estocástica. A partir del tipo a corto plazo se puede inferir toda la curva de tipos de interés. Los modelos de este tipo más representativos son el de Vasicek (1977), el de Cox, Ingersoll y Ross (1985), y el de Duffie y Kan (1996).

Los modelos de splines y paramétricos son los más empleados en el mercado debido a que han demostrado un mejor desempeño. En el caso de los modelos de splines su principal desventaja es que no tienen una forma de curva predeterminada por lo que son muy sensibles a la muestra de datos disponible y al número de segmentos en que se divide la curva. Estos modelos son utilizados con mayor éxito en países que cuentan con un número de bonos suficientes para cada intervalo de la curva, lo que generalmente no ocurre en la

mayoría de países. A cambio de esta debilidad, los modelos de *splines* permiten un mejor ajuste de la curva cuando se dispone de datos suficientes, sobre todo del tramo largo de la curva, en comparación a los modelos paramétricos.

En el siguiente cuadro mostramos los métodos utilizados por diferentes bancos centrales para la estimación de la curva de tipos de interés<sup>3</sup>.

BANCO CENTRAL	MÉTODO	ERROR MINIMIZADO
Bélgica	NS <sup>4</sup> ó NSS <sup>5</sup>	Precios ponderados
Canadá	Splines exponenciales	Precios ponderados
Finlandia	NS	Precios ponderados
Francia	NS ó NSS	Precios ponderados
Alemania	NSS	Rendimientos
Italia	NS	Precios ponderados
Japón	Smoothing spline	Precios
Noruega	NSS	Rendimientos
España	NSS	Precios ponderados
Suecia	Smoothing spline	Rendimientos
Suiza	NSS	Rendimientos
Reino Unido	Anderson Sleath (spline)	Rendimientos
EEUU	Smoothing spline	Precios ponderados

<sup>3</sup> Datos de 2005 obtenidos de [www.bis.org](http://www.bis.org)

<sup>4</sup> Modelo de Nelson y Siegel.

<sup>5</sup> Modelo de Svensson.

## 2.2 La Función de Nelson-Siegel-Svensson.

### Función de Nelson y Siegel

Nelson y Siegel (1987) fueron los primeros en proponer un modelo basado en una propiedad financiera. Estos autores destacan que la curva de tipos forward debe ser asintótica para plazos muy largos. Cuando el plazo es suficientemente grande, los inversores no diferencian el tipo de interés forward de un año para otro, por lo que la función de ajuste debe incorporar esta propiedad.

Este modelo paramétrico, los tipos forward<sup>6</sup> instantáneos en un momento  $t$ , para una inversión iniciada  $m$  periodos en el futuro y que vence un número dado de periodos más allá de la fecha de inicio del contrato, siguen la siguiente forma funcional:

$$f_t(m) = \beta_{t,0} + \beta_{t,1} \exp\left(\frac{-m}{\tau_t}\right) + \beta_{t,2} \frac{m}{\tau_t} \exp\left(\frac{-m}{\tau_t}\right)$$

Donde  $(\beta_{t,0}, \beta_{t,1}, \beta_{t,2}, \tau_t)$  son los parámetros a estimar.

$\beta_{t,0}$  es el tipo de interés cuando  $m$  tiende a infinito, es decir, cuando los inversores ya no diferencian entre el tipo forward de un año para otro.

$\beta_{t,0} + \beta_{t,1}$  es el tipo de interés cuando  $m$  tiende a 0, o el tipo a muy corto plazo.

Los otros dos parámetros definen la forma de la curva:

$\beta_{t,2}$  captura cómo ocurre la transición entre el muy corto plazo y el muy largo plazo: La curva será cóncava, convexa o sinusoidal dependiendo de si  $\beta_{t,2}$  es negativa, positiva o cero.

$\tau_t$  determina la velocidad a la que esta transición tiene lugar.

Dada la curva de tipos forward instantáneos, podemos determinar los tipos spot de un bono de cupón cero con plazo  $m$  de vencimiento integrando esta función:

$$s_t(m) = \frac{1}{m} \int_0^m f_t(u) du$$

---

<sup>6</sup> Los tipos forward o implícitos son los tipos definidos para fechas futuras para un plazo determinado.

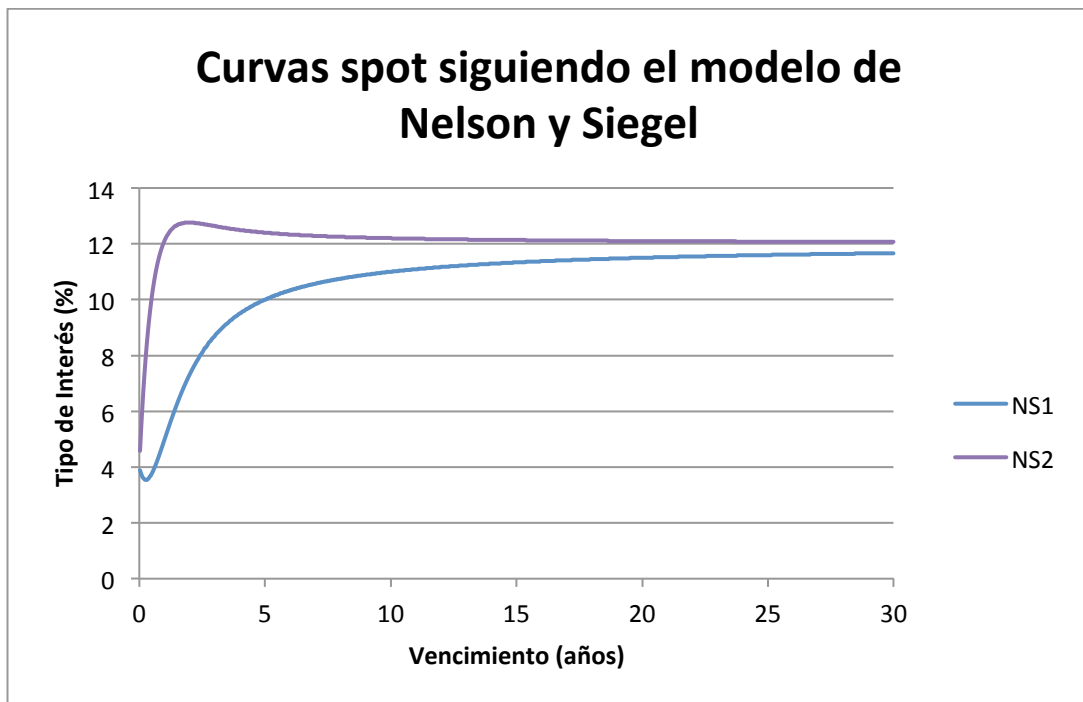
Obteniendo la siguiente expresión:

$$s_t(m) = \beta_{t,0} + (\beta_{t,1} + \beta_{t,2}) \left( \frac{\tau_t}{m} \right) \left( 1 - \exp\left( \frac{-m}{\tau_t} \right) \right) - \beta_{t,2} \exp\left( \frac{-m}{\tau_t} \right)$$

Dado que los tipos de interés spot deben ser estrictamente positivos, para que esta función tenga significado económico los parámetros deben tener las siguientes restricciones:

- $\beta_{t,0} > 0$
- $\beta_{t,0} + \beta_{t,1} > 0$
- $\tau_t > 0$

**Figura 2:** Soluciones de la función de Nelson y Siegel.



Donde:

NS1 corresponde a la siguiente solución:

- $\beta_{t,0} = 12$
- $\beta_{t,1} = -8$
- $\beta_{t,2} = -12$
- $\tau_t = 0,5$

NS2 corresponde a la siguiente solución:

- $\beta_{t,0} = 12$
- $\beta_{t,1} = -8$
- $\beta_{t,2} = 12$
- $\tau_t = 0,5$

En la figura 2 se puede observar la concavidad y convexidad de la curva en función de la positividad o negatividad del parámetro  $\beta_{t,2}$ . También se puede apreciar la tendencia hacia  $\beta_{t,0} + \beta_{t,1}$  cuando  $m$  tiende a 0, y hacia  $\beta_{t,0}$  cuando  $m$  tiende a infinito.

### **Función de Nelson-Siegel-Svensson**

Svensson (1994) amplía el modelo de Nelson y Siegel añadiendo un término que implementa una segunda curvatura a la curva de tipos forward instantáneos:

$$f_t(m) = \beta_{t,0} + \beta_{t,1} \exp\left(\frac{-m}{\tau_{t,1}}\right) + \beta_{t,2} \frac{m}{\tau_{t,1}} \exp\left(\frac{-m}{\tau_{t,1}}\right) + \beta_{t,3} \frac{m}{\tau_{t,2}} \exp\left(\frac{-m}{\tau_{t,2}}\right)$$

El significado de los nuevos parámetros  $\beta_{t,3}$  y  $\tau_{t,2}$  es análogo al de  $\beta_{t,2}$  y  $\tau_t$ , aplicado a la segunda curvatura.

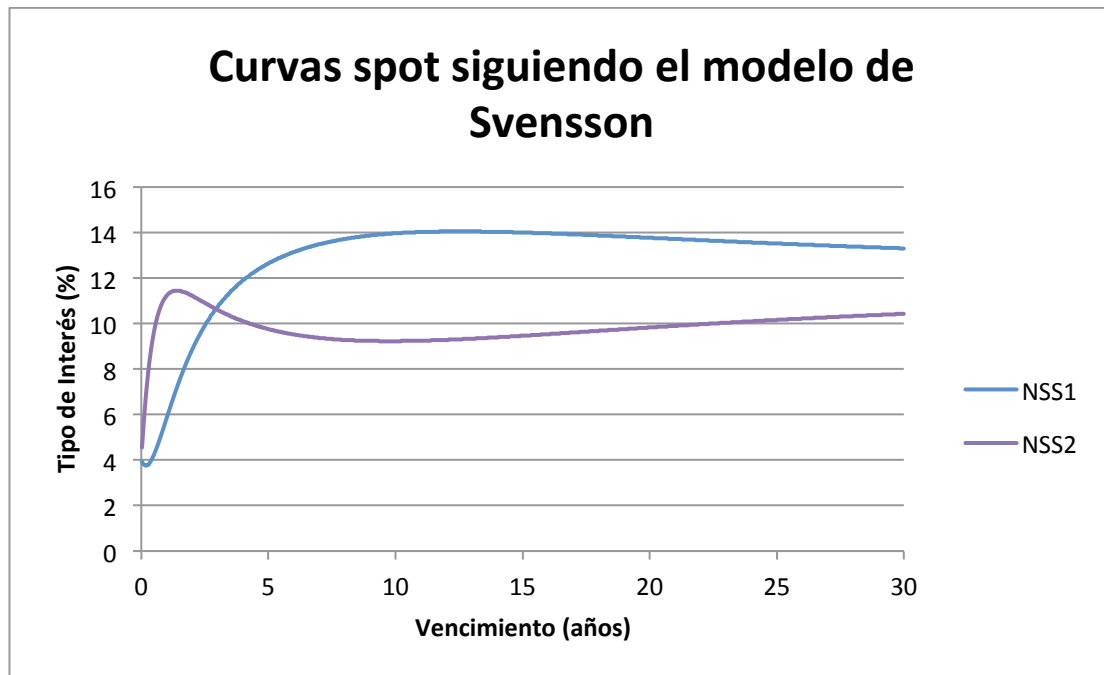
La curva spot también se calcula análogamente al modelo de Nelson y Siegel dando lugar a la siguiente expresión:

$$s_t(m) = \beta_{t,0} + (\beta_{t,1} + \beta_{t,2}) \left(\frac{\tau_{t,1}}{m}\right) \left(1 - \exp\left(\frac{-m}{\tau_{t,1}}\right)\right) - \beta_{t,2} \exp\left(\frac{-m}{\tau_{t,1}}\right) + \beta_{t,3} \left(\frac{\tau_{t,2}}{m}\right) \left(1 - \exp\left(\frac{-m}{\tau_{t,2}}\right)\right) - \beta_{t,3} \exp\left(\frac{-m}{\tau_{t,2}}\right)$$

Para que tenga sentido económico se deben tener en cuenta las siguientes restricciones:

- $\beta_{t,0} > 0$
- $\beta_{t,0} + \beta_{t,1} > 0$
- $\tau_{t,1}, \tau_{t,2} > 0$

Figura 3: Soluciones de la función de Svensson.



Donde:

NSS1 corresponde a la siguiente solución:

- $\beta_{t,0} = 12$
- $\beta_{t,1} = -8$
- $\beta_{t,2} = -12$
- $\beta_{t,3} = 10$
- $\tau_{t,1} = 0,5$
- $\tau_{t,2} = 0,5$

NSS2 corresponde a la siguiente solución:

- $\beta_{t,0} = 12$
- $\beta_{t,1} = -8$
- $\beta_{t,2} = 12$
- $\beta_{t,3} = -10$
- $\tau_{t,1} = 0,5$
- $\tau_{t,2} = 0,5$

En la figura 3 se puede observar como el modelo de Svensson añade una segunda curvatura frente al de Nelson y Siegel, teniendo la curvatura de ambas un comportamiento análogo respecto al valor de los parámetros  $\beta_{1,2}$  y  $\beta_{1,3}$ .

## 2.3 Estimación de los Parámetros de los Modelos de Nelson y Siegel y Svensson

A la hora de estimar los parámetros del modelo de Nelson y Siegel o la ampliación de Svensson existen dos cuestiones fundamentales: La elección de la función objetivo y la elección del algoritmo de optimización. Otras cuestiones que han mostrado importancia en diferentes estudios son la elección de las restricciones de los parámetros y la estimación del valor inicial para determinados algoritmos de optimización.

Nelson y Siegel (1987) ajustaban el modelo prefijando el parámetro  $\tau$ , causante de la no-linealidad, y estimando el resto mediante mínimos cuadrados ordinarios. El proceso era repetido para un rango de  $\tau$  desde 0,027 hasta 1. Esta estimación ha sido tachada de inestable por muchos investigadores como Barret et al. (1995), Cairns et al. (2001), Fabozzi et al. (2005), Diebold et al. (2006 y 2008), o de Pooter (2007).

En el artículo *Estimating the Yield Curve Using the Nelson-Siegel Model: A Ridge Regression Approach*, J. Annaert et al. propone un método iterativo, que denominan **ridge regression**, con el fin de solucionar los problemas de inestabilidad achacados a la multicolinealidad de los parámetros. En él estiman el parámetro  $\tau$  según el proceso utilizado por Nelson y Siegel, realizando iteraciones hasta que un indicador que mide la colinearidad de los parámetros está por debajo de un umbral.

Pero en la mayoría de los casos se estiman todos los parámetros utilizando técnicas de optimización no-lineal.

En el artículo *Estimating the Term Structure of Interest Rates: The Swiss Case*, I. Meier (1999) analiza la robustez (estudiando la variabilidad de los parámetros obtenidos y su significado económico) de la función de Nelson y Siegel para el ajuste de la curva de tipos de interés en el caso Suizo, utilizando el error cuadrático como función de objetivo. Para el ajuste de los parámetros utiliza un algoritmo de **programación cuadrática secuencial**<sup>7</sup>(SQP), añadiendo a las variables ciertas restricciones para afianzar la coherencia económica:  $\beta_0$  no puede superar en más de tres puntos porcentuales al bono con mayor tiempo hasta el vencimiento;  $\beta_1$  se encuentra como mucho a tres puntos porcentuales del rango entre el bono con mayor vencimiento y el interés a tres meses. Concluyen que los resultados son atractivos para el análisis en política monetaria, donde no es necesaria tanta precisión, mientras que para el cálculo de derivados de corto plazo o intereses de productos de largo plazo el método resulta problemático.

---

<sup>7</sup> En concreto el algoritmo SQP de Matlab.



En *Implementation of LM Algorithm and SQP for Yield Curve Optimization Using Nelson-Siegel-Svensson Model*, D. Prastyo et al. (2010) comparan el uso de los algoritmos **Levenberg-Marquardt (LM)** y **SQP** utilizando el error cuadrático como función de fitness para calcular la curva de tipos de interés de Indonesia.

En el artículo *Estimación de la Curva de Rendimiento para el Perú y su uso para el Análisis Monetaria*, J. Pereda (2009) analiza la estimación de la curva del Banco de la Reserva del Perú. Siguiendo éste el modelo de NSS, utilizando como función de fitness la WSSE<sup>8</sup>, y el método de **Newton-Raphson** como algoritmo de búsqueda.

Otros investigadores han estudiado el uso de métodos heurísticos que simplifican la exploración del dominio.

En el artículo *A Genetic Algorithm Estimation of the Term Structure of Interest Rates*, R. Gimeno y J. M. Nave (2009) proponen el uso de **algoritmo genéticos**. Para ello utilizan como función objetivo el error cuadrático del precio del bono ponderado según la duración, y dos métodos de estimación del valor inicial análogos a los utilizados en este proyecto. Partiendo de una muestra de datos del 9 de enero de 2004, comparan los ajustes de ambos modelos mediante el uso de un algoritmo genético y un algoritmo tradicional de mínimos cuadrados, obteniendo resultados prometedores que confirman la hipótesis de menor riesgo de falsa convergencia y mayor estabilidad de los valores en cuanto al sentido económico.

En el artículo *Calibrating the Nelson-Siegel-Svensson model* (2010), M. Grilli, S. Grobe y E. Shumann utilizan un método heurístico, **differential evolution** (DE; Storn and Price, 1997) para estimar los parámetros del modelo de NSS. Para ello utilizan como función de fitness el error cuadrático medio de los intereses de los bonos, y aplican a las variables una serie de restricciones más allá de su significado económico para añadir robustez al modelo, concluyendo que con el método DE se obtuvieron resultados significativamente más fiables que con los métodos tradicionales basados en gradientes de la función de fitness.

---

<sup>8</sup> Descrita en el apartado 3.1.2 *Función Objetivo*.



# Capítulo 3

## Enfoque del Problema

### 3.1 Planteamiento del Problema

Tal y como se deduce de los capítulo anteriores, nuestro problema consta de:

- Una función con seis parámetros a optimizar (NSS).
- Una función a minimizar que mida la bondad de la solución.
- Un algoritmo que explore el dominio de soluciones para alcanzar un valor cercano al óptimo.

### 3.1.1 Parámetros a optimizar

La función de NSS consta de seis parámetros a optimizar:  $(\beta_{t,0}, \beta_{t,1}, \beta_{t,2}, \beta_{t,3}, \tau_{t,1}, \tau_{t,2})$

En la implementación utilizada para nuestra búsqueda, las componentes del vector  $x$  adoptan los siguientes significados

$$x = (\beta_{t,0}, \beta_{t,1}, \beta_{t,2}, \beta_{t,3}, \tau_{t,1}, \tau_{t,2})$$

cuya descripción se encuentra en el apartado 2.2 *La Función de Nelson-Siegel-Svensson*.

El dominio de búsqueda basado en las propiedades financieras sería:

$$\text{LímiteInferior} = (0, -x_1, -Inf, -Inf, 0, 0)$$

$$\text{LímiteSuperior} = (Inf, Inf, Inf, Inf, Inf, Inf)$$

Sin embargo, teniendo en cuenta la experiencia y las limitaciones de computación, establecemos los siguientes límites:

$$\text{LímiteInferior} = (0, -x_1, -150, -150, 0, 0)$$

$$\text{LímiteSuperior} = (50, 50, 150, 150, 300, 300)$$

### 3.1.2 Función Objetivo

La función objetivo utilizada es una implementación del error cuadrático del precio del bono ponderado según la duración<sup>9</sup> (WSSE). Esta función es utilizada por los bancos centrales de España, Canadá, Bélgica, Francia y EEUU entre otros.

La ponderación penaliza los bonos a largo plazo a favor de los bonos a corto plazo, con el fin de solucionar el hecho de que un mismo error cuadrático en el precio afecta de maneras muy diferentes al ajuste dependiendo de su duración: Una variación pequeña en el precio de un bono a largo plazo representa un cambio pequeño en los tipos de interés; mientras que supone una variación más significativa si pertenece a un bono a corto plazo.

---

<sup>9</sup> Duración de Macaulay: Media de los distintos vencimientos de los flujos de caja ponderados por el valor actual de cada uno.

Dado el conjunto de datos con los precios, duración, fecha de inicio y vencimiento del cupón y una posible solución  $x = (x_1, x_2, x_3, x_4, x_5, x_6)$  del modelo de NSS, esta función realiza el siguiente proceso:

1. Cálculo de los precios teóricos<sup>10</sup> de los bonos utilizando el modelo de NSS con los parámetros  $x = (x_1, x_2, x_3, x_4, x_5, x_6)$ .

$$precio\_teórico = (p_{t1}, p_{t2}, \dots, p_m)$$

Donde  $n$  es el número de bonos para el día evaluado.

2. Cálculo de la diferencia entre los precios reales y los precios teóricos de los bonos.

$$error = precio\_real - precio\_teórico = (e_1, e_2, \dots, e_n)$$

3. Estimación del peso de cada bono según su duración, utilizando la siguiente fórmula:

$$w = \frac{(d_1^{-1/2}, d_2^{-1/2}, \dots, d_n^{-1/2})}{\sum_{i=1}^n d_i^{-1/2}}$$

Donde  $d$  es la duración del bono.

4. Cálculo del error ponderado.

$$error\_ponderado = (e_1 w_1, e_2 w_2, \dots, e_n w_n) = (ep_1, ep_2, \dots, ep_n)$$

5. Cálculo del error cuadrático

$$error\_cuadrático = (ep_1^2 + ep_2^2 + \dots + ep_n^2)$$

---

<sup>10</sup> El segmento de código que calcula el precio teórico a partir del modelo de Svensson ha sido suministrado por Ricardo Gimeno del Banco de España.

## 3.2 Algoritmos de búsqueda

La función del algoritmo de búsqueda es explorar el dominio de los parámetros del modelo de NSS, alcanzando un valor próximo al óptimo con un gasto computacional razonable.

Para ello utilizamos varias técnicas heurísticas, propias de la computación evolutiva, así como algunos métodos clásicos de optimización no-lineal.

### 3.2.1 Métodos Tradicionales

Estudiamos el comportamiento de algoritmos tradicionales de búsqueda de mínimos para funciones multivariable con restricciones. Esto es lo que comúnmente se denomina programación no-lineal.

Utilizamos dos métodos de programación no-lineal diferentes:

- Algoritmos de punto interior
- Programación secuencial cuadrática

Estos métodos se centran en la resolución de las ecuaciones de Karush-Kuhn-Tucker (KKT). Las ecuaciones de KKT son condiciones necesarias para la optimalidad en un problemas de optimización con restricciones, y suponen la base de muchos algoritmos de programación no-lineal.

### 3.2.2 Técnicas Evolutivas

La computación evolutiva (CE) es una rama de la inteligencia artificial que se inspira en los mecanismos de la evolución biológica para resolver problemas de optimización.

La mayor parte de paradigmas tradicionales de optimización se mueven de un punto a otro del dominio de posibles soluciones utilizando reglas determinísticas. Uno de los problemas de estos métodos es la posibilidad de quedar estancado en un óptimo local.

Por otro lado, los paradigmas de la CE utilizan una población de soluciones. Habitualmente producen una nueva población con el mismo número de miembros en cada generación. De esta manera se exploran simultáneamente muchas soluciones alternativas, disminuyendo la probabilidad de falsa convergencia. Operadores como mutación o recombinación habilitan la capacidad de búsqueda paralela a través de diferentes zonas del dominio de búsqueda.

Los paradigmas de la CE no requieren información auxiliar al problema, como cálculo de derivadas. Utilizan una función que mide la calidad de la solución, cuyo valor se desea optimizar.

El hecho de que los paradigmas de la CE utilicen métodos estocásticos no significa que realicen una búsqueda aleatoria. Por el contrario, los operadores estocásticos dirigen la búsqueda hacia zonas del dominio con alta probabilidad de alcanzar buenos resultados. Por ejemplo, los operadores de selección asignan mayor probabilidad de reproducirse a los individuos con mejor fitness.

Los paradigmas de la CE generalmente difieren de los métodos de búsqueda y optimización tradicionales de tres maneras:

- Utilizan una población de posibles soluciones en su búsqueda.
- Utilizan funciones de fitness en lugar de técnicas derivativas o similares.
- Utilizan procesos estocásticos en lugar de reglas determinísticas.

La población es un conjunto de individuos (posibles soluciones), que en cada iteración son evaluados mediante la función de fitness, y sometidos a una serie de operaciones para formar la siguiente generación.

La función de fitness se utiliza para evaluar la idoneidad de cada individuo, siendo el valor que devuelve el objetivo a optimizar.

Generalmente para implementar una técnica evolutiva se utiliza una combinación de las siguientes operaciones:

- **Selección:** es la operación mediante la cual se seleccionan los padres de la siguiente generación a partir de la población actual.
- **Recombinación:** esta operación combina los valores de las variables de los padres para formar los descendientes.
- **Mutación:** la operación de mutación tiene lugar asignando cambios aleatorios sobre los descendientes para formar la siguiente generación.

A continuación mostramos el esquema de un paradigma genérico de CE:

1. **Inicialización** de la población.
2. **Evaluación** de la idoneidad de cada individuo mediante la función de fitness.
3. **Selección** de los individuos con mejor salud siguiendo algún criterio.
4. **Recombinación** de los padres seleccionados para crear los descendientes.
5. **Mutación** de cada descendiente.
6. **Reinserción** de los nuevos individuos para formar la nueva generación.
7. Volver al paso 2 hasta que se alcanza un **criterio de finalización**.

Los distintos paradigmas de CE se diferencian en la forma de llevar a cabo los pasos y en la cantidad de ellos que realizan, pudiendo algunos de ellos ser omitidos en función de la técnica.

La **inicialización** generalmente se realiza creando una población aleatoria. Si los parámetros están representados en cadenas binarias, simplemente se generan valores aleatorios de ceros y unos con igual probabilidad. También es común incluir en la población inicial algunos individuos prometedores, dirigiendo la búsqueda hacia un espacio del dominio cercano al óptimo. El número de individuos seleccionado para formar la población depende del problema y del paradigma seleccionado. Suele oscilar desde unas decenas hasta varios cientos de individuos.

El valor de **fitness** es normalmente proporcional a la salida de la **función de fitness**, aunque también puedes ser una combinación de varios parámetros, y depende del método de selección que utilizemos.

La **selección** de los individuos para formar la siguiente generación se basa normalmente en el valor de fitness. De este modo, los individuos con mejor fitness suelen tener mayor probabilidad de ser elegidos para formar la siguiente generación.

La reproducción tiene lugar siguiendo diferentes criterios, como la **recombinación** de los individuos seleccionados, el paso directo de los mejores individuos a la siguiente generación (elitismo), o la **mutación** directa de los padres.

Después de producir los descendientes, éstos deben ser introducidos en nueva la población. La **reinserción** determina cuantos descendientes se introducen en la nueva población y bajo qué criterio.

Existen mucho **criterios de finalización** dependiendo del algoritmo. En general se utilizan varios de ellos simultáneamente para terminar la búsqueda.

Para la consecución de los objetivos del proyecto, utilizamos dos paradigmas de CE:

1. Algoritmos Genéticos.
2. Estrategias Evolutivas.

Ambas técnicas están orientadas a la búsqueda de óptimos numéricos, y se describen detalladamente en los apartados 5 y 6 respectivamente.



## 3.3 Medios Empleados

Como ya hemos mencionado, el objetivo de este proyecto es analizar el comportamiento de diferentes paradigmas de CE para explorar el dominio de búsqueda del problema., realizando para ello un estudio exploratorio de carácter científico para el que necesitamos seleccionar el instrumental necesario.

En este apartado explicamos los puntos más importantes del desarrollo de la fase de selección del instrumental:

- Muestra de datos.
- Librerías de computación evolutiva.
- Software de programación no-lineal.
- Entorno de trabajo.

La selección del entorno de trabajo fue muy ligada a la búsqueda de librerías de CE con las que implementar los algoritmos de búsqueda.

También barajamos la posibilidad de la implementación propia de algunas técnicas, pero tras encontrar numerosas herramientas que permiten configurar cualquier paradigma con multitud de opciones, descartamos esta opción.

### 3.3.1 Muestra de datos

Para realizar el ajuste de la curva y posterior análisis, hemos utilizado una muestra de datos de 30 días entre el 31 de mayo de 2000 y el 26 de julio de 2000, que contiene información de valores en circulación y operaciones de compra venta de deuda pública del Banco de España<sup>11</sup>.

### 3.3.2 Librerías de computación evolutiva

Realizamos una búsqueda y análisis de librerías, tras la que barajamos las siguientes opciones:

- Genetic Algorithm Toolbox (Matlab)
- Global Optimization Toolbox (Matlab)
- Genetic and Evolutionary Algorithm Toolbox (Matlab)

---

<sup>11</sup> Datos suministrados por Ricardo Gimeno del Banco de España.

- Evolving Objects (C++)
- Java Genetic Algorithms Package (Java)
- Evolutionary Computation Toolbox (Matlab)
- Diferentes implementaciones de CMA-ES obtenidas de la página web de los diseñadores (ANSI C, C++, Java, Matlab y Octave, Python, R...)

## Genetic Algorithm Toolbox

Genetic Algorithm Toolbox (Department of Automatic Control and Systems Engineering of The University of Sheffield, 1994) contiene una serie de funciones y rutinas implementadas para Matlab, con las que configurar un amplio rango de algoritmos genéticos.

Algunas de sus características son:

- Desarrollado para Matlab.
- Gratuito (GNU general license<sup>12</sup>).
- Sólo contiene algoritmos genéticos, no estrategias evolutivas.

## Global Optimization Toolbox

La Global Optimization Toolbox es una herramienta de optimización desarrollada por Mathworks que contiene varios métodos de búsqueda de soluciones globales para problemas con múltiples máximos y mínimos, en los que la función de fitness puede ser continua, discontinua, estocástica, no-derivable, no-lineal...

Entre otros métodos, incluye algoritmos genéticos con multitud de opciones de configuración.

Algunas de sus características son:

- Desarrollado para Matlab
- Coste de la licencia individual: 22 € (versión para estudiantes adquiriendo la licencia de Matlab).
- Sólo contiene algoritmos genéticos, no estrategias evolutivas.
- Opciones de visualización y progreso de la ejecución.

---

<sup>12</sup> Cualquier persona puede copiar y distribuir el producto sin realizar cambios.

## **Genetic and Evolutionary Algorithm Toolbox (GEATbx)**

GEATbx es un paquete de algoritmos evolutivos estructurado por módulos, que permite implementar cualquier paradigma de computación evolutiva con múltiples opciones.

Algunas de sus características son:

- Desarrollado para Matlab
- Coste de la licencia individual: 476 euros para empresa/297,5 para estudiantes o uso universitario.
- Contiene tanto algoritmos genéticos como estrategias evolutivas.
- Opciones de visualización y progreso de la ejecución.

## **Evolving Objects (EO)**

EO es una librería de computación evolutiva estructurada por módulos, que permite implementar fácilmente diferentes técnicas evolutivas.

Algunas de sus características son:

- Desarrollado para C++
- Gratuito (GNU general license).
- Permite implementar tanto algoritmos genéticos como estrategias evolutivas.
- Opciones de visualización y estadísticas.

## **Java Genetic Algorithms Package (JGAP)**

JGAP es un framework Java de algoritmos genéticos y programación genética.

Algunas de sus características son:

- Desarrollado para Java
- Gratuito (GNU general license)
- Sólo permite implementar algoritmos genéticos, no estrategias evolutivas

## **Evolutionary Computation Toolbox (EVOCOM)**

EVOCOM es una librería de optimización para Matlab que proporciona un framework para implementar diferentes algoritmos evolutivos. Se puede utilizar como un algoritmo genético tradicional, o para resolver problemas de programación genética, o como combinación de otras técnicas de inteligencia artificial.

Algunas de sus características son:

- Desarrollado para Matlab.
- Gratuito (los términos de la licencia no se especifican en su página web).
- Sólo permite implementar algoritmos genéticos, no estrategias evolutivas.

## **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)**

CMA-ES es una estrategia evolutiva compleja para problemas de optimización no-lineales, no-convexos en dominios continuos.

En la página web de los diseñadores, se pueden encontrar varias implementaciones fieles al paradigma original (Hansen, 2006).

Algunas de sus características son:

- Desarrollada para ANSI C, C++, Java, Matlab y Octave, Python, R...
- Gratuito (GNU general license)
- Se trata sólo de una estrategia evolutiva muy compleja.
- Opciones de visualización y progreso de la ejecución.

Entre las opciones descritas, y tras realizar una experimentación previa con algunas de ellas, escogimos las siguientes opciones:

- GEATbx
- CMA-ES (Implementación de los autores del algoritmo para Matlab)

Escogimos GEATbx porque a pesar del coste de la licencia y de que requiera la utilización del entorno Matlab, cuenta con numerosas posibilidades de configuración para implementar tanto algoritmos genéticos como estrategias evolutivas, además de amplia documentación y soporte.

GEATbx ha sido desarrollado por H. Pohlheim, y es un paquete muy completo con múltiples funciones para realizar cada una de las operaciones que componen una técnica evolutiva.

El motivo de escoger la implementación de CMA-ES facilitada por los autores, fue su flexibilidad y facilidad de uso, así como la disponibilidad en Matlab, que encaja con la elección de GEATbx para implementar el resto de paradigmas.

CMA-ES ha sido desarrollado por N. Hansen, uno de los referentes en el campo de la computación evolutiva. Utilizamos la implementación CMA-ES que se puede descargar en su página web, donde también se pueden encontrar amplia documentación y referencias.

### 3.3.3 Software de programación no-lineal

Una vez seleccionadas las librerías de computación evolutiva y decidido el uso del entorno Matlab para realizar el trabajo experimental, realizamos una búsqueda y análisis de software de programación no-lineal para Matlab, tras la que barajamos las siguientes opciones:

- Interior-Point Optimizer
- Optimization Toolbox de Matlab
- TOMLAB Optimization

#### IPOPT (Interior-Point Optimizer)

IPOPT es un paquete de software de programación no-lineal de gran escala, diseñado para encontrar soluciones locales en problemas de optimización con restricciones, donde la función objetivo puede ser no-lineal y no-convexa.

Algunas de sus características son:

- Desarrollado para C++.
- Incluye interfaces para Matlab, R y Java entre otros.
- Gratuito (Eclipse public license<sup>13</sup>).
- Sólo incluye un método de punto interior.

---

<sup>13</sup> Licencia de software libre de código abierto que permite utilizar, modificar, copiar y distribuir el trabajo.

## Optimization Toolbox de Matlab

La Optimization Toolbox de Matlab contiene un amplio número de algoritmos para resolver diferentes problemas de optimización.

Algunas de sus características son:

- Coste de la licencia variable, en torno a 30 euros con la licencia para estudiantes de Matlab.
- Incluye un algoritmo de punto interior y dos de programación secuencial cuadrática.

## TOMLAB

TOMLAB es un entorno de Matlab para problemas de optimización que agrupa en módulos múltiples librerías externas de optimización en una sola interfaz.

Algunas de sus características son:

- Coste variable en función de los módulos seleccionados. El coste de algunos módulos de optimización no-lineal está en torno a 300 euros para uso académico.
- Incluye múltiples métodos de programación no-lineal, entre los que cuenta con varios algoritmos de programación secuencial cuadrática.

Seleccionamos la Optimization Toolbox de Matlab, ya que contiene varias implementaciones de los métodos requeridos con un coste mucho menor que TOMLAB, así como por su facilidad de uso.

### 3.3.4 Entorno de trabajo

#### **MATLAB**

El nombre Matlab es una abreviatura de las palabras Matrix Laboratory. Matlab es un sistema interactivo para cálculos científicos y de ingeniería basado en las matrices. Con él se pueden resolver complejos problemas numéricos sin necesidad de escribir un programa específico para ello, aunque también es posible programar. Además, el programa Matlab dispone, dependiendo de la versión, de diferentes módulos que permiten resolver problemas específicos.

Como ya explicamos en el apartado anterior, la selección del entorno fue muy ligada a la búsqueda de herramientas para implementar las técnicas. A pesar del coste de la licencia (66 euros la versión para estudiantes), escogimos Matlab por la mayor variedad y calidad de las herramientas encontradas, así como por varias ventajas con las que contaba frente a otros entornos:

- El ahorro de tiempo y esfuerzo en implementar la función de fitness y dar formato a los datos utilizados, con los que contábamos preparados para el entorno de Matlab.
- La familiaridad con el lenguaje de Matlab.
- Facilidad de integración del código ajeno con la parte propia que se encarga de programar la configuración del algoritmo y gestionar los resultados.

#### **R**

Para llevar a cabo parte del análisis de los resultados, utilizamos el entorno de programación para análisis estadístico R.

Se trata de un proyecto de software libre resultado de la implementación GNU del lenguaje S, y actualmente uno de los lenguajes más utilizados en investigación estadística.

R se distribuye bajo la licencia GNU general license y está disponible para los sistemas operativos Windows, MacOS, Unix y GNU/Linux.





# Capítulo 4

## Métodos Tradicionales

En este capítulo estudiamos el comportamiento de algoritmos tradicionales de optimización para funciones multivariable con restricciones. Esto es lo que comúnmente se denomina programación no-lineal.

Utilizamos tres algoritmos diferentes:

- Interior-Point
- SQP
- Active-Set

Todos ellos se encuentran implementados en la función *fmincon* de Matlab.

## 4.1 Interior Point

### 4.1.1 Introducción

Los métodos de punto interior están inspirados por el algoritmo de Karmakar, desarrollado por N. Karmakar (1984) para la programación lineal.

Éstos basan su estrategia en la búsqueda del óptimo a través de caminos que recorren la zona interior de una región factible. Para ello transforman el problema de optimización con restricciones en otro sin ellas, y fijan una función barrera que evita que se salga de la región factible.

A continuación se muestra el desarrollo del método de punto interior utilizado, el algoritmo Interior-Point de Matlab.

El enfoque para la minimización con restricciones del algoritmo Interior-Point, se basa en resolver una secuencia de problemas de minimización aproximados.

Siendo el problema original:

$$\min_x f(x), \text{ sujeto a } h(x) = 0 \text{ y } g(x) \leq 0 \quad (1)$$

Para cada  $\mu > 0$ , el problema aproximado es:

$$\min_{x,s} f(x,s) = \min_{x,s} f(x) - \sum_i \mu \ln(s_i), \text{ sujeto a } h(x) = 0 \text{ y } g(x) + s = 0 \quad (2)$$

Hay tantos  $s_i$  como restricciones de desigualdad  $g$ . A medida que  $\mu$  se acerca a cero, el mínimo de  $f_\mu$  se debería aproximar al mínimo de  $f$ . El término logarítmico se denomina función barrera.

El problema aproximado (2) es una secuencia de problemas de igualdad que resultan más fáciles de resolver que (1).

Para resolver el problema aproximado, el algoritmo utiliza dos tipos de pasos en cada iteración:

1. Paso directo en  $(x,s)$ .
2. Paso de gradiente conjugado.

Por defecto el algoritmo intenta tomar el *paso directo*. Si no puede, intenta el *paso de gradiente conjugado*.

En cada iteración, el algoritmo disminuye una función de mérito:

$$f_{\mu}(x,s) + v \|(h(x), g(s) + s)\|$$

El parámetro  $v$  puede incrementarse con el número de iteraciones para forzar la solución hacia la factibilidad. Si el intento de paso no disminuye la función de mérito, el algoritmo lo descarta y realiza uno nuevo.

## Paso directo

Las siguientes variables son utilizadas para definir el paso directo:

- $H$  denota el Hessiano del Lagrangiano de  $f_{\mu}$  :

$$H = \nabla^2 f(x) + \sum_i \lambda_j \nabla^2 h_j(x)$$

- $J_g$  denota el Jacobiano de la función de restricciones  $g$ .
- $J_h$  denota el Jacobiano de la función de restricciones  $h$ .
- $S = \text{diag}(s)$ .
- $\lambda$  denota el vector de multiplicadores de Lagrange asociado a las restricciones  $g$ .
- $\Lambda = \text{diag}(\lambda)$ .
- $y$  denota el vector de multiplicadores de Lagrange asociado a  $h$ .
- $E$  denota el vector de unos del mismo tamaño que  $g$ .

La siguiente ecuación define el paso directo  $(\Delta x, \Delta s)$ :

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & S\Lambda & 0 & -S \\ J_h & 0 & I & 0 \\ J_g & -S & 0 & I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ -\Delta y \\ -\Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f - J_h^T y - J_g^T \lambda \\ S\lambda - \mu e \\ h \\ g + h \end{bmatrix}$$

Esta ecuación proviene del intento de solucionar las ecuaciones de *KKT* (*Karush-Kuhn-Tucker*) utilizando un Lagrangiano linealizado.

Para resolver esta ecuación para  $(\Delta x, \Delta s)$ , el algoritmo realiza una factorización *LDL* de la matriz. Un resultado de esta factorización es determinar si el Hessiano proyectado es definido positivo o no; si no, el algoritmo utiliza el paso de gradiente conjugado.

## Paso de gradiente conjugado

Con el enfoque del gradiente conjugado para resolver el problema aproximado (2), el algoritmo ajusta  $x$  y  $s$  manteniendo  $s$  positivo. Minimiza una aproximación cuadrática al problema aproximado en una región factible sujeta a restricciones lineales.

Siendo  $R$  el radio de la región factible, y las otras variables iguales a las definidas en el paso directo, el algoritmo obtiene los multiplicadores de Lagrange resolviendo aproximadamente las ecuaciones de *KKT*

$$\nabla_x L = \nabla_x f(x) + \sum_i \lambda_i \nabla g_i(x) + \sum_j y_j \nabla h_j(x) = 0$$

mediante mínimos cuadrados, sujeto a  $\lambda \geq 0$ . Después toma un paso  $(\Delta x, \Delta s)$  para resolver aproximadamente

$$\min_{\Delta x, \Delta s} \nabla f^T \Delta x + \frac{1}{2} \Delta x^T \nabla_{xx}^2 L \Delta x + \mu e^T S^{-1} \Delta s + \frac{1}{2} \Delta s^T S^{-1} \Lambda \Delta s \quad (3)$$

sujeto a las restricciones linealizadas

$$g(x) + J_g \Delta x + \Delta s = 0, \quad h(x) + J_h \Delta x = 0 \quad (4)$$

Para resolver la ecuación (4), el algoritmo intenta minimizar una norma de las restricciones linealizadas dentro de la región factible de radio  $R$ . Luego la ecuación (3) se resuelve con las restricciones obtenidas al resolver la ecuación (4), permaneciendo en la región factible de radio  $R$ , y manteniendo  $s$  estrictamente positivo.

## 4.1.2 Configuración de los Parámetros del Algoritmo

Al no tratarse de un método estocástico, realizamos pruebas de una ejecución sobre el primer día del rango total utilizado para ajustar algunos parámetros del algoritmo.

Preparamos pruebas para las siguientes opciones:

1. Método de cálculo del Hessiano
2. Método estimación de gradientes
3. Método de cálculo del subproblema

Asimismo estudiamos los criterios de finalización para realizar pruebas con aquellos que sean relevantes.

### 4.1.2.1 Método de Cálculo del Hessiano

El algoritmo Interior-Point contiene varias opciones para el cálculo del Hessiano, de las cuales probamos las que no requieren de la estimación de un gradiente:

1. Cálculo del Hessiano mediante una aproximación quasi-Newton densa (*bfgs*).
2. Cálculo Hessiano mediante una aproximación quasi-Newton de gran-escala (*lbfgs*).

Realizamos pruebas con cada una de las opciones obteniendo los siguientes resultados:

Cálculo Hessiano	Fmin
bfgs	0,00194
lbfgs	0,00338

Donde *Fmin* es el valor mínimo encontrado de la función objetivo.

Seleccionamos la opción *bfgs*.

### 4.1.2.2 Método de Estimación de los Gradientes

Las diferencias finitas utilizadas para estimar los gradientes pueden ser *forward* o *centradas*. *Centradas* toma el doble de evaluaciones de la función objetivo pero es más exacta.

Realizamos pruebas con cada una de las opciones obteniendo los siguientes resultados:

Estimación Gradientes	Fmin
forward	0,00194
centrada	0,00287

Escogemos la opción *forward*.

### 4.1.2.3 Número Máximo de Evaluaciones

Este parámetro define el máximo número de evaluaciones de la función objetivo.

En las anteriores pruebas éste fue el criterio de finalización. Realizamos pruebas con 3000, 5000, 7000 y 10000 evaluaciones, obteniendo los siguientes resultados:

MaxFunEvals	Fmin
3000	0,00194
5000	0,00192
7000	0,001901
10000	0,001900

A partir de 10000 el criterio de finalización pasa a ser el número máximo de iteraciones. Fijamos el número máximo de evaluaciones en 10000.

#### 4.1.2.4 Número Máximo de Iteraciones

Realizamos pruebas con número máximo de iteraciones 1000 y 2000 obteniendo los siguientes resultados:

Número Máximo de Iteraciones	Fmin
1000	0,00190
2000	0,00188

Con 2000 iteraciones el criterio de finalización pasa a ser otra vez el número máximo de iteraciones.

El resto de criterios de finalización no ha mostrado ninguna relevancia, y las mejoras obtenidas ampliando el número máximo de iteraciones y el número máximo de evaluaciones han sido mínimas.

#### 4.1.2.5 Método de Cálculo del Subproblema

Determina cómo es calculado el paso de la iteración. Por defecto utiliza *factorización-ldl*, normalmente más rápida que el método del gradiente conjugado (*cg*), aunque *cg* puede resultar más rápida para problemas grandes con Hessianos densos.

Realizamos pruebas con gradiente conjugado, obteniendo los siguientes resultados.

Cálculo Subproblema	Fmin
factorización-ldl	0,00188
cg	0,00339

Seleccionamos la opción *factorización-ldl*.

### 4.1.3 Experimentación

Tras ajustar los parámetros del algoritmo, preparamos pruebas de una ejecución (al no ser un método estocástico no es necesario realizar más ya que siempre devolvería el mismo valor) sobre un rango de treinta días .

Realizamos la experimentación utilizando dos métodos diferentes:

1. Aportando el mismo valor inicial en cada ejecución para cada día.
2. Aportando la mejor solución del día anterior como valor inicial para el siguiente día.

El valor inicial aportado en todas las ejecuciones del primer método y en la primera del segundo corresponde a una función de los datos de los bonos del día evaluado. Realiza la siguiente asignación de valores:

- $\beta_{t,0}$  = rendimiento del bono de máxima duración contenido en los datos.
- $\beta_{t,0} + \beta_{t,1}$  = rendimiento del bono de mínima duración contenido en los datos.
- $\beta_{t,2} = 0$
- $\beta_{t,3} = 0$
- $\tau_{t,1} = \tau_{t,2}$  = vencimiento del bono con rendimiento más cercano al rendimiento medio del conjunto de los datos.

En el segundo método introducimos este valor inicial para el primer día, continuando en orden ascendente hasta completar el rango de treinta días utilizando la solución del día anterior para los siguientes. La ETTI no suele cambiar bruscamente entre días cercanos, por lo que la solución del día anterior debería suponer una buena aproximación para el siguiente.



A continuación se presenta un resumen de los resultados.

**Aportando el mismo valor inicial en cada ejecución para cada día**

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00165	0,00123
<b>T. Ejecución</b>	39,20838	29,18308

**Aportando la mejor solución del día anterior como valor inicial para el siguiente día.**

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00111	0,00080
<b>T. Ejecución</b>	35,88450	19,60015

Los resultados desagregados se detallan en el *Anexo*.

## 4.3 Active Set

### 4.3.1 Introducción

El algoritmo Active Set de la Optimization Toolbox de Matlab, es en realidad un método de programación cuadrática secuencial (SQP), que utiliza un método de conjunto activo para solucionar los subproblemas QP aproximados.

El primer método SQP fue propuesto en su tesis doctoral por Wilson (1963). A partir de este momento, los métodos SQP han evolucionado hasta convertirse en herramientas muy efectivas para un amplio rango de problemas de optimización.

El algoritmo Active-Set de Matlab es un método iterativo para optimización no-lineal. Basándose en los trabajos de Biggs (1975), Han (1977) y Powell (1978) permite emular el método de Newton para la optimización con restricciones como si se tratara de un problema sin restricciones. En cada iteración, se realiza una aproximación del Hessiano del Lagrangiano utilizando un método quasi-Newton. Ésta aproximación se utiliza para generar un subproblema QP, cuya solución es utilizada para determinar la dirección de búsqueda para el procedimiento lineal.

Matemáticamente podemos expresar el problema de la siguiente forma:

$$\text{Minimizar } f(x)$$

Donde  $f(x)$  es una función no lineal del tipo  $\mathfrak{R}^n \rightarrow \mathfrak{R}$ , y  $x = [x_1, x_2, \dots, x_n]$  sujeta a  $m$  restricciones no lineales, de las que  $m_e$  son de igualdad y el resto de desigualdad:

$$c_i(x) = 0, i=1, 2, \dots, m_e$$

$$c_i(x) \geq 0, i= m_e + 1, \dots, m$$

El método SQP en cada iteración aproxima localmente la función  $f$  no lineal a una función  $q$  cuadrática, y las restricciones no lineales a restricciones lineales.

Para ello utiliza un método quasi-Newton que realiza una aproximación definida positiva del Hessiano del Lagrangiano. Por lo tanto, se genera un subproblema de programación cuadrática (QP) con la aproximación cuadrática del Lagrangiano.

El subproblema QP es resuelto mediante un método de conjunto activo. Éste método es similar al de Gill et al. (1984 y 1991), pero modificado para problemas de programación lineal y programación cuadrática. Consta de dos fases: en la primera fase calcula un punto factible (si existe); la segunda consiste en la generación de una secuencia iterativa de puntos factibles que converjan a la solución.

La solución del problema QP produce un vector, que es utilizado para realizar una búsqueda lineal iterativa que minimice la función de mérito.

## 4.3.2 Configuración de los Parámetros del Algoritmo

Al igual que en el resto de métodos no estocásticos, realizamos pruebas de una ejecución sobre el primer día del rango, atendiendo al criterio de finalización para completar las pruebas.

### 4.3.2.1 Método de Estimación de los Gradientes

Seleccionamos entre los métodos de estimación de gradientes explicados en el apartado 4.1.2.2.

Realizamos pruebas obteniendo los siguientes resultados:

Estimación Gradientes	Fmin
forward	0,00445
center	0,00445

El resultado no cambia, por lo que seleccionamos la opción *forward*, al igual que en el resto de métodos tradicionales.

### 4.3.2.2 TolX

Este parámetro define la tolerancia en el valor de  $x$ , criterio mediante el que se termina la ejecución si las diferencias entre los valores de  $x$  de iteraciones sucesivas son menores a la tolerancia.

Ha sido el criterio de finalización en las anteriores pruebas, por lo que realizamos pruebas disminuyendo este valor, obteniendo los siguientes resultados:

TolX	Fmin
1,00E-06	0,00445
1,00E-07	0,00445
1,00E-08	0,00338
1,00E-09	0,00131

A partir de tolerancia 1,00E-09 el criterio de finalización pasa a ser el número máximo de evaluaciones.

### 4.3.2.3 Número Máximo de Evaluaciones

Realizamos pruebas aumentando el número hasta que cambia el criterio de finalización, obteniendo los siguientes resultados:

MaxFunEvals	Fmin
600	0,00131
1000	0,00125

Con número máximo de evaluaciones igual a 1000 el criterio de finalización pasa a ser la tolerancia en el valor de  $x$ , por lo que volvemos a este criterio obteniendo los siguientes resultados:

TolX	Fmin
1E-10	0,00125
1E-11	0,00125

A partir de este punto los resultados no mejoran, por lo que deducimos que a convergido a un óptimo local.

### 4.3.3 Experimentación

Repetimos el procedimiento descrito en el apartado 4.1.3, realizando una ejecución con los parámetros fijados sobre el rango de treinta días.

A continuación mostramos el resumen de los resultados obtenidos siguiendo cada uno de los métodos.

#### Aportando el mismo valor inicial en cada ejecución para cada día

	MEDIA	DESVIACIÓN
Fmin	0,00210	0,00134
T. Ejecución	0,78002	0,44627

#### Aportando la mejor solución del día anterior como valor inicial para el siguiente día

	MEDIA	DESVIACIÓN
Fmin	0,00209	0,00134
T. Ejecución	0,78923	0,47013

Los resultados desagregados se detallan en el *Anexo*.

## 4.4 SQP

### 4.4.1 Introducción

El algoritmo SQP de la Optimization Toolbox de Matlab es tan solo una variación del método Active-Set descrito en la sección anterior.

Ambos aproximan localmente la función  $f$  no lineal a una función  $q$  cuadrática en cada iteración, y las restricciones no lineales a restricciones para luego resolver este subproblema QP y utilizar el vector obtenido para realizar una búsqueda lineal minimizando la función de mérito.

Las principales diferencias frente a Active Set del algoritmo SQP son:

1. Satisface las restricciones en todas las iteraciones.
2. Puede recuperarse de valores Inf y NaN, realizando un nuevo intento con un paso más pequeño.
3. Utiliza un método diferente al de conjunto activo para la resolución de los subproblemas QP.
4. Utiliza un enfoque diferente cuando las restricciones del subproblema QP no son satisfechas.

## 4.4.2 Configuración de los Parámetros del Algoritmo

Al no tratarse de un método estocástico, comenzamos realizando pruebas de una ejecución sobre el primer día del rango de datos.

Preparamos pruebas para seleccionar el método de estimación de los gradientes. Asimismo estudiamos los criterios de finalización para decidir posteriormente cuales son relevantes para continuar con los ajustes de las opciones.

### 4.4.2.1 Método de Estimación de los Gradientes

Las diferencias finitas utilizadas para estimar los gradientes pueden ser forward o centradas, cuya diferencia ya explicamos en el apartado 4.1.2.2.

Realizamos pruebas obteniendo los siguientes resultados:

Estimación Gradientes	Fmin
forward	0,00125
center	0,00173

Seleccionamos la opción *forward*.

### 4.4.2.2 Número Máximo de Evaluaciones

En las anteriores pruebas este ha sido el criterio de finalización, por lo que realizamos pruebas aumentando el número máximo de evaluaciones hasta que cambie el criterio de finalización, obteniendo los siguientes resultados:

MaxFunEvals	Fmin
600	0,00194
1000	0,00125

Con número máximo de evaluaciones igual a 1000 el criterio de finalización pasa a ser la tolerancia en el valor de  $x$ , por lo que pasamos a las pruebas con este criterio.

### 4.4.2.3 TolX

En SQP, la tolerancia de  $x$  sigue el mismo mecanismo explicado en el apartado. 4.3.2.2

Realizamos pruebas aumentando este valor, obteniendo los siguientes resultados:

TolX	Fmin
1,00E-06	0,00125
1,00E-07	0,00125
1,00E-08	0,00125

Los resultados son iguales aunque el criterio de finalización sigue siendo el mismo, por lo que deducimos que ha convergido a un mínimo local.



### 4.4.3 Experimentación

Seguimos el mismo procedimiento descrito en el apartado 4.1.3 realizamos pruebas de una ejecución sobre el rango de treinta días.

A continuación mostramos el resumen de los resultados obtenidos siguiendo cada uno de los métodos.

#### **Aportando el mismo valor inicial en cada ejecución para cada día**

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00052	0,00043
<b>T. Ejecución</b>	7,50522	2,73291

#### **Aportando la mejor solución del día anterior como valor inicial para el siguiente día.**

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00038	0,00041
<b>T. Ejecución</b>	4,73301	2,98301

Los resultados desagregados se detallan en el *Anexo*.



# Capítulo 5

## Algoritmos Genéticos

### 5.1 Introducción

Un algoritmo genético (AG) es un método para resolver problemas de optimización basado en la selección natural, el proceso que impulsa la evolución biológica.

La primera mención del término y la primera publicación sobre una aplicación del mismo se deben a Bagley (1967), que diseñó algoritmos genéticos para buscar conjuntos de parámetros en funciones de evaluación de juegos.

Pero es otro científico el considerado creador de los algoritmos genéticos: John Holland, que los desarrolló junto con algunos de sus alumnos durante las décadas de los 60 y 70.

En 1975 Holland publicó *Adaptation in Natural and Artificial Systems*, uno de los libros más importantes en este campo. La mayor innovación de Holland (1975) fue introducir un algoritmo basado en poblaciones con cruces y mutaciones.

Un AG utiliza tres operaciones en cada paso para crear la siguiente generación:

- **Selección:** escoge siguiendo algún criterio los individuos de la población para la reproducción.
- **Recombinación:** combina dos padres para formar un descendiente.
- **Mutación:** efectúa cambios aleatorios en los individuos.

Generalmente, en este paradigma de CE la operación de cruce se realiza asignándole una probabilidad alta, mientras que la operación de mutación suele tener una probabilidad muy pequeña.

El AG implementado lleva a cabo el siguiente proceso:

1. Inicialización de la población.
2. Evaluación de cada individuo utilizando la función de fitness.
3. Selección de los individuos para la reproducción.
4. Cruce de los individuos seleccionados.
5. Mutación de los descendientes.
6. Reinserción de los descendientes en la población para formar la siguiente generación.
7. Volver al paso 2 hasta alcanzar un criterio de finalización.

## 5.2 Configuración de los Parámetros del Algoritmo.

Para la selección de los diferentes parámetros y funciones que configuran el AG hemos realizado pruebas siguiendo diferentes procedimientos según la configuración utilizada y la desviación del error cuadrático.

Partimos de una población de 100 individuos, creando 100 descendientes en cada generación y con un número máximo de 100 generaciones. Modificamos los parámetros correspondientes a medida que avanza la experimentación.

Para las primeras pruebas realizamos 10 ejecuciones para cada opción, ya que con 100 generaciones el tiempo de ejecución es bajo. Una vez fijados algunos parámetros modificamos este procedimiento a medida que aumenta el tiempo de ejecución y disminuye la desviación del valor de fitness.

El criterio de selección ha sido la media del valor de fitness.

### 5.2.1 Inicialización de la Población

Al iniciar la ejecución introducimos un valor inicial para enfocar la búsqueda hacia una zona del dominio prometedora.

La EE implementada crea el 80 % de la población inicial mediante una distribución uniforme en el dominio de búsqueda. El resto de individuos son creados mediante una distribución normal en torno al valor inicial introducido previamente.

De este modo la población inicial consta de un 20 % cercano a una solución prometedora, mientras que explora todo el dominio mediante el 80 % restante.

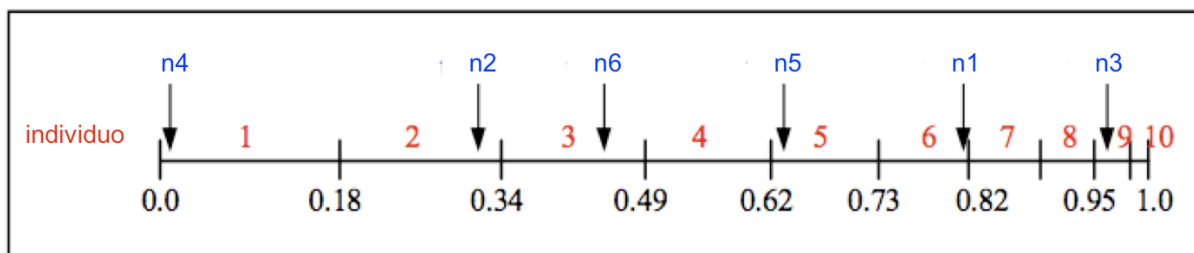
## 5.2.2 Selección

Escogimos entre las funciones de selección descritas a continuación.

### Ruleta (selrws)

Se ubican los individuos en una línea de rango [0;1], ocupando cada uno un segmento proporcional a su valor de fitness. A continuación se genera un número aleatorio entre 0 y 1, y el individuo en cuyo segmento se encuentra el número es seleccionado. Se repite este proceso hasta que la población intermedia está completa.

Figura 4: Método de la ruleta

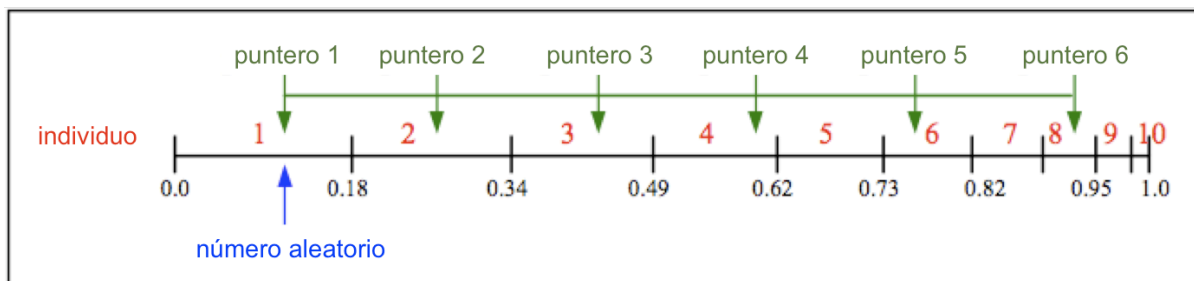


La figura 4 muestra la selección de seis individuos sobre diez mediante la generación de números aleatorios ( $n$ ).

### Selección Estocástica Universal (selsus)

Se ubican los individuos en una línea de rango [0;1], ocupando cada uno un segmento proporcional a su valor de fitness. A continuación se genera un número aleatorio entre 0 y 1, colocando sobre la línea  $N$  punteros equiespaciados (siendo  $N$  la cantidad de individuos a seleccionar y  $1/N$  la distancia entre punteros) comenzando en este punto. Se seleccionan los individuos en cuyos segmentos han caído los  $N$  punteros.

Figura 5: Selección estocástica universal



La figura 5 muestra la selección de seis individuos sobre diez mediante la generación de un número aleatorio y la colocación de seis punteros equiespaciados .

## Torneo (seltour)

Este método escoge aleatoriamente un número de individuos (en nuestro caso 3) de la población para llevar a cabo el torneo, seleccionando como padre el individuo con mejor fitness. Este proceso se repite hasta formar una población intermedia de padres.

## Truncamiento (seltrunc)

Se forma la población intermedia con los mejores  $T$  individuos, siendo  $T$  el umbral de truncamiento (en nuestro caso 59%).

Todas estas funciones crean una población intermedia, para a continuación seleccionar los individuos para la recombinación de manera aleatoria uniforme con reemplazo.

Realizamos pruebas de 10 ejecuciones sobre un solo día, obteniendo los siguientes resultados:

Función Selección	FMEDIA	FDESV
selsus	0,10729	0,09320
selrws	0,13413	0,07690
seltour	0,05561	0,25809
seltrunc	0,16905	0,02443

Donde:

- **FMEDIA:** promedio del error cuadrático de las soluciones de todas las ejecuciones.
- **FDESV:** desviación típica del error cuadrático.

Basándonos en estos resultados escogemos la selección por torneo.

### 5.2.3 Recombinación

Escogimos entre los métodos de recombinación descritos a continuación.

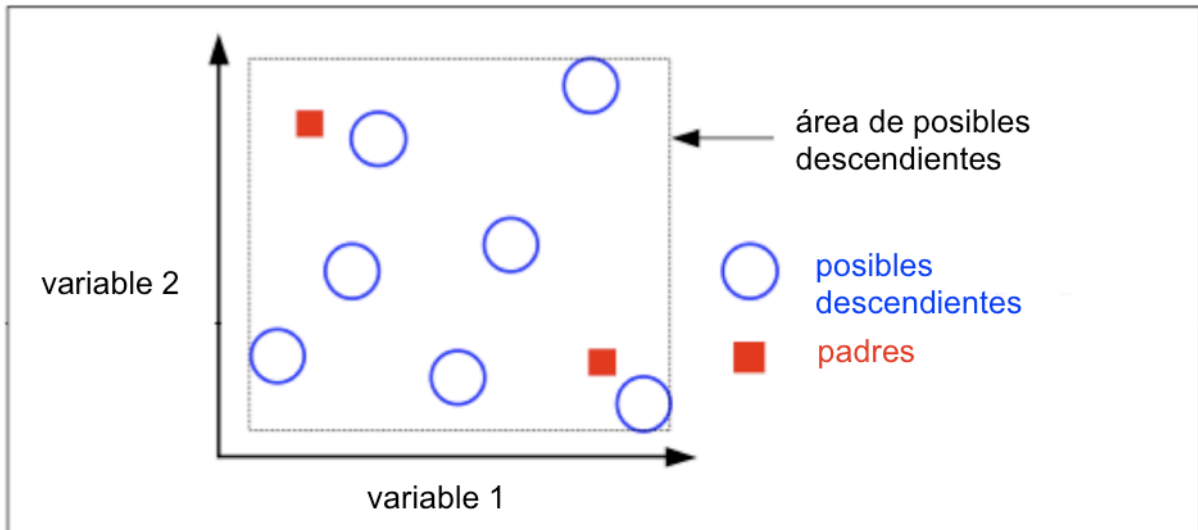
#### Recombinación intermedia (recint)

La recombinación intermedia es un método aplicable a variables reales donde dada variable del descendiente  $x$  se crea a partir de los padres  $(x^1, x^2)$  siguiendo el siguiente esquema:

$$x'_i = x_i^1 * a_i + x_i^2 * (1 - a_i) \text{ donde } i \in (1, 2, \dots, n \text{ var})$$

$$a_i \in [-d, 1 + d] \text{ siguiendo una distribución aleatoria uniforme con } d=0,25$$

**Figura 6:** Recombinación intermedia



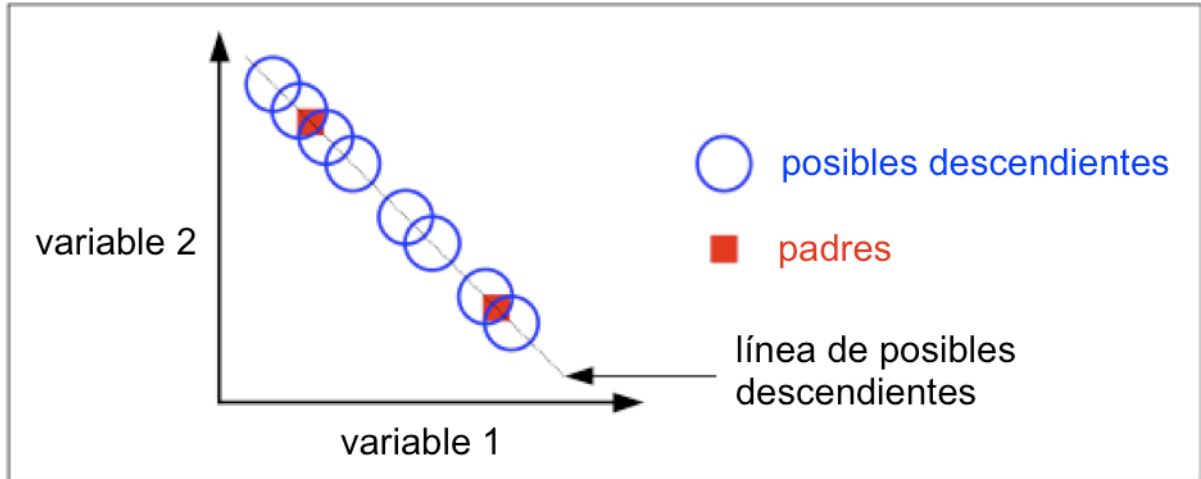
La figura 6 muestra el área de posibles valores del descendiente comparada con los padres en la recombinación intermedia.



## Recombinación lineal (reclin)

La recombinación lineal es sólo una variante de la recombinación intermedia en la que el parámetro  $a$  es igual para todas las variables.

**Figura 7:** Recombinación lineal



La figura 7 muestra el área de posibles valores del descendiente comparada con los padres en la recombinación lineal.

## Recombinación lineal extendida (reclinex)

La recombinación lineal extendida es una ampliación de la anterior. En este caso la línea que definen las variables de los padres es sólo la zona de mayor probabilidad para el descendiente, reduciéndose la probabilidad a medida que nos alejamos de esta zona.

$$Var_i^0 = Var_i^{P1} + s_i r_i a_i \frac{Var_i^{P2} - Var_i^{P1}}{\|Var_i^{P2} - Var_i^{P1}\|} \quad i \in (1, 2, \dots, N \text{ var})$$

Donde

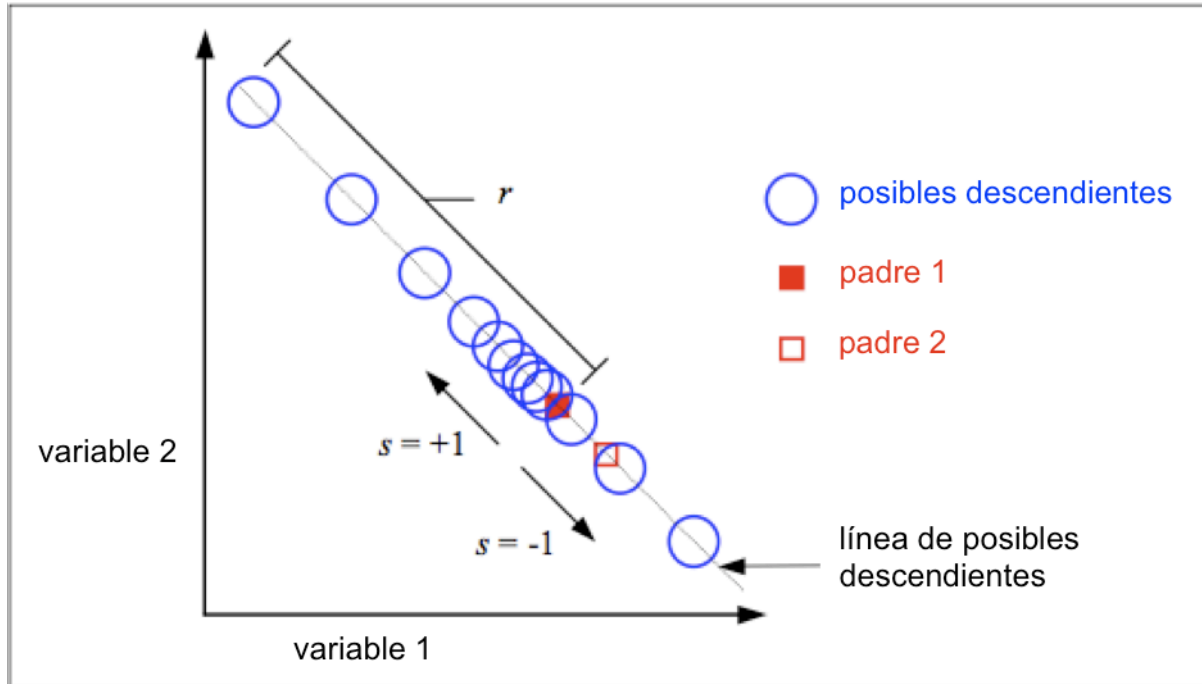
$Var_i$ : Variable  $i$  del individuo.

$a_i = 2^{-ku}$   $k$ : Precisión mutación,  $u \in [0, 1]$  aleatoria uniforme.

$r_i = r * dominio_i$ , donde  $r$  es el rango del paso de recombinación.

$s_i \in \{-1, +1\}$  aleatoria uniforme

**Figura 8:** Recombinación lineal extendida



La figura 8 muestra el área de posibles valores del descendiente comparada con los padres en la recombinação lineal extendida.

En primer lugar realizamos una batería de 10 ejecuciones sobre un día con cada función, obteniendo los siguientes resultados:

Función de Recombinación	FMEDIA	FDESV
recint	0,25566	0,17538
reclin	1,14406	1,08054
reclinx	0,25032	0,08770

Posteriormente (una vez fijado el elitismo), realizamos pruebas para seleccionar conjuntamente tamaño de población, número máximo de generaciones y función de recombinación. Basándonos en los resultados de las anteriores pruebas descartamos la *recombinación lineal*.

Las pruebas consistieron en 5 ejecuciones sobre un mismo día para cada configuración. A continuación se muestran las medias del valor error cuadrático obtenidas para cada configuración:

		Población					
		25		50		100	
		Reclinux	Recint	Reclinux	Recint	Reclinux	Recint
Generaciones	2500	0,00498	0,01055	0,01142	0,02136	0,00821	0,01512
	5000	0,01187	0,00375	0,01147	0,01942	0,00289	0,01245
	10000	0,00986	0,00312	0,00321	0,00358	<b>0,00267</b>	0,00561

En las pruebas preliminares comprobamos que con 20000 generaciones el AG no mostraba mejoría significativa, mientras que doblaba el tiempo de ejecución, por lo que fijamos el número de generaciones en 10000 y escogimos la *recombinación lineal extendida*.

## 5.2.4 Mutación

Utilizamos mutación para valores reales, de modo que valores aleatorios son añadidos a cada variable con baja probabilidad.

Existen dos parámetros que definen la mutación:

- **Tasa de mutación:** probabilidad de mutación de cada variable.
- **Tamaño de paso:** magnitud de la mutación.

Seleccionamos una tasa de mutación de 0,5. Esto implica que existe una probabilidad de mutación de  $0,5/N$  por cada variable (siendo  $N$  el número de variables), es decir,  $1/12$ .

El tamaño del paso es independiente de esta probabilidad. Existen dos parámetros que definen el tamaño de paso, según los cuales se pueden configurar estrategias muy diferentes:

- **Precisión:** define el menor tamaño de paso posible.
- **Rango de Mutación:** define el radio del área de posibles pasos.

Para estos dos parámetros hemos tomado los siguientes valores:

- Precisión ( $k$ )=12
- Rango ( $r$ )=1

De modo que cada paso de mutación tiene lugar siguiendo la siguiente fórmula:

$$x'_i = x_i + s_i * r_i * 2^{-u*k} \text{ donde } i \in (1, 2, \dots, n \text{ var})$$

$$s_i \in \{-1, 1\} \text{ siguiendo una distribución aleatoria uniforme.}$$

$$r_i = r * dom_i \text{ siendo } dom_i \text{ el dominio de la variable } i.$$

$$u \in \{0, 1\} \text{ siguiendo una distribución aleatoria uniforme.}$$

### 5.2.5 Reinserción

Mediante las operaciones de recombinación y mutación creamos tantos descendientes como individuos hay en la población.

La reinserción se lleva a cabo sustituyendo todos los padres o un porcentaje de ellos por los mejores descendientes.

Así que llevamos a cabo pruebas con elitismo del 20%, 10% y 0%. De modo que se conservaría el mejor 20%, 10% ó 0% de los padres respectivamente para la siguiente generación, ocupando porcentaje restante los mejores descendientes.

Realizamos pruebas de 10 ejecuciones sobre un solo día para cada opción, obteniendo los siguientes resultados:

Elitismo	FMEDIA	FDESV
1	0,15525	0,08158
0,9	0,03545	0,04518
0,8	0,07799	0,06939

Basándonos en estos resultados escogimos un 10% de elitismo.

### 5.2.6 Criterios de Finalización

El número máximo de generaciones fue el único criterio de finalización relevante en las pruebas preliminares para fijar el resto de parámetros, por lo que realizamos una batería de pruebas para fijar el tamaño de la población y el número máximo de generaciones con diferentes funciones de recombinación.

Basándonos en estos resultados que mostramos en el apartado 5.2.3, escogimos un número máximo de generaciones de 10000.

### 5.2.7 Tamaño de Población

Basándonos en las mismas pruebas que el número máximo de generaciones y la función de recombinación, fijamos el tamaño de población en 100 individuos.

## 5.3 Experimentación

Tras ajustar los parámetros del algoritmo, preparamos pruebas de diez ejecuciones sobre un rango de treinta días .

Realizamos la experimentación utilizando tres métodos diferentes:

1. Aportando el mismo valor inicial en cada ejecución para cada día.
2. Aportando la mejor solución del día anterior como valor inicial para el siguiente día.
3. Realizando cuatro rearranques aportando la mejor solución del día anterior como valor inicial para el siguiente día.

A continuación se describen los tres métodos utilizados y se presenta un resumen de los resultados.

### Aportando el mismo valor inicial en cada ejecución para cada día

Realizamos una ejecución para el rango completo de días en orden ascendente, repitiendo este proceso hasta alcanzar treinta ejecuciones para cada día.

En este método introducimos el valor inicial descrito en el apartado 4.1.3. Es decir, cada día tiene un valor inicial diferente basado en los datos de los bonos, pero igual al del resto de ejecuciones del mismo día.

	MEDIA	DESVIACIÓN
Fmin	0,00114	0,00086
T. Ejecución (seg)	590,43432	130,75279

### Aportando la mejor solución del día anterior como valor inicial para el siguiente día.

Seguimos el mismo procedimiento que en el anterior método, pero utilizamos la solución del día anterior como valor inicial para el siguiente.

En este caso realizamos las ejecuciones para dos opciones diferentes de inicialización de la población:

	MEDIA	DESVIACIÓN
Fmin	0,00090	0,00079
T. Ejecución (seg)	715,17674	100,41044

## Realizando cuatro rearranques aportando la mejor solución del día anterior como valor inicial para el siguiente día.

En esta variación seguimos el mismo procedimiento que en el anterior método, pero realizando cuatro rearranques para cada ejecución. Es decir, para cada día se realiza la ejecución introduciendo el valor inicial correspondiente, se utiliza la solución como valor inicial para realizar otra ejecución sobre el mismo día, y así sucesivamente hasta alcanzar cuatro rearranques.

El objetivo de este método es evitar que la población se estanque en un óptimo local, reiniciando la exploración del dominio en cada rearranque con un 20% distribuido normalmente en torno a la solución de la ejecución anterior y un 80% distribuido uniformemente en el dominio.

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00076	0,00064
<b>T. Ejecución (seg)</b>	2748,22772	1047,17682

Los resultados desagregados se detallan en el *Anexo de Resultados*.





# Capítulo 6

## Estrategias Evolutivas

### 6.1 Introducción

Las estrategias evolutivas (EE) fueron desarrolladas en 1964 por un grupo de estudiantes alemanes de ingeniería encabezado por Ingo Rechenberg y Hans-Paul Schwefel. Su objetivo era resolver problemas hidrodinámicos de alto grado de complejidad. En 1973 I. Rechenberg publicó un libro considerado la fundación de este paradigma.

Las EE se centran en la operación de mutación como motor fundamental para la búsqueda, aunque también pueden utilizar recombinación. La representación de los individuos consta de variables objeto y componentes adaptativas o parámetros de estrategia, que almacenan información sobre la cercanía al óptimo.

La operación de mutación se lleva a cabo sobre variables reales continuas, realizando perturbaciones mediante una distribución normal de media cero y cuya desviación define tamaño de paso, parámetro que determina la magnitud de la siguiente mutación en función de la cercanía al óptimo.

Una EE genérica sigue el siguiente proceso:

1. Inicialización de la población.
2. Evaluación de cada individuo utilizando la función de fitness.
3. Selección de los individuos para formar la población intermedia de padres.
4. Mutación de los descendientes.
5. Reinserción de los descendientes para formar la siguiente generación.
6. Vuelta al paso 2 hasta alcanzar algún criterio de finalización.

Dependiendo de la EE implementada, se pueden utilizar diferentes tipos de mutación. El punto común de este paradigma es la inclusión de cierta información sobre la cercanía al óptimo en la representación de las variables para determinar la magnitud de la mutación para formar la siguiente generación.

## **EE utilizadas en la experimentación**

Para llevar a cabo la experimentación hemos seleccionado dos EE distintas:

1.  $(\mu, \lambda)$ -EE.
2. CMA-ES (Covariance Matrix Adaptation Evolution Strategy).

Estos métodos se describen en los apartados 6.2 y 6.3 respectivamente.

## 6.2 ( $\mu,\lambda$ )-EE

La herramienta seleccionada para llevar a cabo esta parte de la experimentación es la Genetic and Evolutionary Algorithm Toolbox, descrita en el apartado 3.3.2.

En una ( $\mu,\lambda$ )-EE, la población consta de  $\mu$  individuos, y se crean  $\lambda$  descendientes. La siguiente generación se forma escogiendo a los mejores  $\mu$  descendientes (si  $\lambda > \mu$ ) o sustituyendo a los peores  $\lambda$  padres por los descendientes (si  $\lambda < \mu$ ).

A continuación describimos los parámetros que definen el algoritmo, así como las pruebas de selección realizadas.

### 6.2.1 Configuración de los Parámetros del Algoritmo.

Para la selección de los diferentes parámetros y funciones que configuran la EE hemos realizado pruebas de 5 ejecuciones sobre un rango de 8 días utilizando el mejor individuo del día anterior como valor inicial para el siguiente, y con un número máximo de 500 generaciones.

El objetivo de esta metodología es eliminar resultados negativos que obtuvimos en pruebas preliminares utilizando este método en el rango completo de treinta días tras configurar los parámetros siguiendo un procedimiento similar al del AG. Observamos resultados fuera del orden normal (fitness de orden  $10^2$  frente a los resultados normales de  $10^{-3}$ ), especialmente en los primeros 8 días del rango total de 30 que abarcamos en la experimentación. Por lo que repetimos la configuración de los parámetros realizando pruebas sobre los días más problemáticos, escogiendo 500 como número máximo de generaciones, ya que la EE suele alcanzar en 500 generaciones en torno al 90% del valor que alcanzaría con 10000 y supone un coste computacional mucho menor.

El criterio de selección ha sido la media del valor de fitness.

#### 6.2.1.1 Inicialización de la Población

Al iniciar la ejecución introducimos un valor inicial para enfocar la búsqueda hacia una zona del dominio prometedora.

La EE implementada crea la mitad de la población inicial mediante una distribución uniforme en el dominio de búsqueda. El resto de individuos son creados mediante una distribución normal en torno al valor inicial introducido previamente.

### 6.2.1.2 Mutación, Tamaño de la Población y Número de Descendientes

Realizamos la selección de la función de mutación conjuntamente con el número de padres y descendientes, es decir, tamaño de población ( $\mu$ ) y número de descendientes ( $\lambda$ ), utilizando un número máximo de 500 generaciones en las pruebas.

Escogimos entre las siguientes ( $\mu,\lambda$ )-EE:

- (3,30)-EE
- (3,60)-EE
- (5,30)-EE
- (5,60)-EE

Con cada una de las opciones anteriores realizamos pruebas con las siguientes funciones de mutación:

1. Derandomized ES-algorithm with individual step-sizes (mutes1).
2. ES-algorithm with derandomized mutative step-size control using accumulated information (mutes2).

Ambas funciones implementan técnicas desarrolladas por A. Ostermeier et al., y presentadas en los artículos *A Derandomized Approach to Self-Adaptation of Evolution Strategies* (1994) y *Step-size adaptation based on non-local use of selection information* (1994) respectivamente.

A continuación describimos cada una de las funciones de mutación utilizadas.

#### Derandomized ES-algorithm with individual step-sizes

Este método emplea tamaños de paso individuales para cada variable, y una única dirección para cada individuo, según el siguiente proceso:

1. Creación de los descendientes

$$\vec{x}_{N_k}^g = \vec{x}_E^g + \xi^k \vec{\delta}^g \vec{z}^k \quad \text{con } k=1,\dots,\lambda$$

2. Selección

$$\vec{x}_E^{g+1} = \vec{x}_{N_{sel}}^g$$

3. Evolución de las variables adaptativas

$$\vec{\delta}^{g+1} = \left( \xi^{sel} \right)^\beta \left( \vec{\xi}_{z_{sel}} \right)^{\beta_{scal}} \vec{\delta}^g$$

Donde

$\vec{x}_{N/E}^g$  : variables objeto de los padres ( $N$ ) y descendientes ( $E$ ) de la generación  $g$ .

$\vec{\delta}^g$  : vector de tamaños de paso individuales de la generación  $g$ .

$\xi^k = \alpha; 1/\alpha$  distribuido uniformemente con  $\alpha = 1,4$  : Variación global del tamaño de paso del descendiente  $k$ .

$\vec{\xi}_{z_{sel}} = (\xi_1, \dots, \xi_n)$  con  $\xi_i = \exp\left\{\left|z_i^{sel}\right| - \sqrt{2/\pi}\right\}$  : Adaptación individual del tamaño de paso.

$\vec{z}^k = (z_1, \dots, z_n)$  donde  $z_i$  sigue una distribución normal  $(0,1)$ .

*sel*: individuos seleccionados.

$n$ : número de variables.

$$\beta = \sqrt{1/n}$$

$$\beta_{scal} = 1/n$$

### ES-algorithm with derandomized mutative step-size control using accumulated information.

Esta función analiza las variaciones de mutación acumuladas en varias generaciones para adaptar el tamaño de paso, ya que estas están generalmente correlacionadas, según el siguiente proceso:

1. Creación de los descendientes

$$\vec{x}_{N_k}^g = \vec{x}_E^g + \vec{\delta}^g \vec{\delta}_{scal}^g \vec{z}^k \text{ con } k=1, \dots, \lambda$$

2. Selección

$$\vec{x}_E^{g+1} = \vec{x}_{N_{sel}}^g$$

3. Acumulación de mutaciones seleccionadas

$$\vec{Z}^g = (1-c)\vec{Z}^{g-1} + c\vec{Z}_{sel} \text{ con } \vec{Z}^0 = \vec{0}$$

4. Adaptación de los tamaños de paso general e individual

$$\delta^{g+1} = \delta^g \left( \exp \left( \frac{|\bar{z}^g|}{\sqrt{n} \sqrt{\frac{c}{2-c}}} - 1 - \frac{1}{5n} \right) \right)^\beta$$

$$\bar{\delta}_{scal}^{g+1} = \bar{\delta}_{scal}^g \left( \frac{|\bar{z}^g|}{\sqrt{\frac{c}{2-c}}} + 0,35 \right)^{\beta_{scal}}$$

Donde

$\bar{x}_{N/E}^g$  : variables objeto de los padres ( $N$ ) y descendientes ( $E$ ) de la generación  $g$ .

$\delta^g$  : tamaño de paso general de la generación  $g$ .

$\bar{\delta}_{scal}^g$  : vector de tamaños de paso individuales de la generación  $g$ .  $\bar{\delta}_{scal}^0 = (1, \dots, 1)$

$\bar{z}^k = (z_1, \dots, z_n)$  donde  $z_i$  sigue una distribución normal  $(0,1)$ .

$sel$ : individuos seleccionados.

$n$ : número de variables.

$c = \sqrt{1/n}$ : constante que establece el peso de las mutaciones de generaciones anteriores, siendo  $c \in (0,1]$ . El factor  $\sqrt{\frac{c}{2-c}}$  normaliza las variaciones medias.

$\beta = \sqrt{1/n}$  : la precisión y velocidad de la adaptación dependen de estos dos exponentes.  
 $\beta_{scal} = 1/n$

A continuación se muestran las medias del valor de fitness obtenidas para cada configuración:

		Función de Mutación	
		Mutes1	Mutes2
EE	(3,30)	0,00174	0,00138
	(3,60)	0,00091	0,00098
	(5,30)	6,52687	0,01541
	(5,60)	0,04957	20,62427

Los resultados con  $\mu = 5$  son muy inferiores al resto.

Las elevadas medias de (5,30)-EE con *mutés1* y (5,60)-EE con *mutés2* se deben a unos pocos resultados de rango muy superior al resto ( $10^3$ ), pero muy recurrentes en diferentes pruebas realizadas con estas técnicas. Aun eliminando estos resultados las medias de ambas técnicas siguen siendo de orden inferior a las obtenidas con  $\mu = 3$ , por lo que no barajamos otras opciones de experimentación para tratar de solventar este problema.

Seleccionamos la función *mutés1* y la (3,60)-EE.

El tamaño de paso inicial lo fijamos en un 0,01% del dominio de cada variable.

### 6.2.1.3 Selección

Para escoger la función de selección realizamos pruebas conjuntas con la presión selectiva (*SP*), parámetro con diferente significado según la función de selección. Las funciones de selección están definidas en el apartado 5.2.2. A continuación explicamos el significado de la presión selectiva en cada una:

- **Ruleta (selrws)/ Selección Estocástica Universal (selsus):** en ambos métodos la presión selectiva define la pendiente de la curva de escalado de fitness. A mayor *SP*, más probabilidad de ser elegidos tienen los mejores individuos.
- **Torneo (seltour):** la presión selectiva define el tamaño del torneo.
- **Truncamiento (seltrunc):** la presión selectiva define el umbral de truncamiento, siendo este  $T=1/SP$ .

A continuación se muestran las medias del valor de fitness obtenidas para cada configuración:

		Función Selección			
		Seltrunc	Seltour	Selrws	Selsus
SP	1	0,00091	0,00131	0,00102	0,00092
	2	0,00103	0,00098	0,00101	0,00097
	3	0,00096	0,00093	0,00104	0,00094
	4	0,00094	0,00103	0,00092	0,00111

Basándonos en estos resultados escogemos la selección por truncamiento, y realizamos otra nueva batería de pruebas con 5000 generaciones para escoger la presión selectiva:

SP	FMEDIA	FDESV
1	0,00088	0,00040
2	0,00087	0,00041
3	0,00088	0,00041
4	0,00088	0,00041
5	0,00094	0,00054

Escogemos presión selectiva 2, lo que implica un umbral de truncamiento del 50%, es decir, el mejor 50% de los individuos son seleccionados para formar la población intermedia.

#### 6.2.1.4 Método de Reinserción

Este parámetro define el tipo de EE, en este caso  $(\mu, \lambda)$ . Los mejores descendientes sustituyen a los padres para formar la siguiente generación.

#### 6.2.1.5 Criterios de Finalización

En los experimentos realizados para fijar el resto de parámetros, el único criterio de finalización determinante fue el número máximo de generaciones.

Seleccionamos un número máximo de 10000 generaciones al no apreciar cambios significativos a partir de este punto.



## 6.2.2 Experimentación

Siguiendo el mismo procedimiento que con el AG, realizamos pruebas de diez ejecuciones sobre un rango de treinta días utilizando las tres variantes descritas en el apartado 5.3.

A continuación mostramos el resumen de los resultados obtenidos siguiendo cada uno de los métodos.

### Aportando el mismo valor inicial en cada ejecución para cada día

	MEDIA	DESVIACIÓN
<b>Fmin</b>	0,00053	0,00063
<b>T. Ejecución (seg)</b>	101,70155	6,86698

### Aportando la mejor solución del día anterior como valor inicial para el siguiente día

	MEDIA	DESVIACIÓN
<b>Fmin</b>	0,00041	0,00045
<b>T. Ejecución (seg)</b>	99,07406	6,33476

### Realizando cuatro rearranques aportando la mejor solución del día anterior como valor inicial para el siguiente día

	MEDIA	DESVIACIÓN
<b>Fmin</b>	0,00039	0,00041
<b>T. Ejecución (seg)</b>	497,48327	32,86654

Los resultados desagregados se detallan en el *Anexo de Resultados*.

## 6.3 Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

### 6.3.1 Introducción

CMA-ES es un método estocástico para la optimización de funciones no-lineales, no-convexas, con parámetros reales y dominio continuo, desarrollado por N. Hansen (2006).

Se trata de una EE más compleja que el paradigma canónico, una  $(\mu/\mu_w, \lambda)$ -ES. Esto quiere decir que es una EE no elitista con recombinación intermedia ponderada, en la que se obtiene cada nueva generación muestreando una distribución normal multivariante.

La principal característica que diferencia CMA-ES de otras EE, es la utilización de una distribución normal multivariante de media  $m$  y matriz de covarianzas  $C$  (que determina la forma del elipsoide de distribución) para implementar el modelo de aprendizaje. Ésta explota no sólo la relación entre éxitos y fracasos de cada variable en los anteriores pasos, sino también la relación entre variables mediante la matriz de covarianzas.

La exploración del dominio tiene lugar utilizando los mismos parámetros para determinar todos los nuevos individuos:

$$x_i \sim m + \sigma N_i(0, C) \text{ para } i=1, \dots, \lambda$$

Donde

- El vector media  $m$  representa la media de las mejores soluciones.
- El tamaño de paso  $\sigma$  controla la longitud de cada paso.
- La matriz de covarianzas  $C$  determina la forma del elipsoide de distribución.

A continuación se describe detalladamente el cálculo y adaptación de estos parámetros.

### 6.3.1.1 Fórmula General

La fórmula general para la generación  $g+1$  de  $\lambda$  descendientes sería:

$$x_k^{(g+1)} \sim m^{(g)} + \sigma^{(g)} N(0, C^{(g)}) \text{ para } k=1, \dots, \lambda$$

Donde:

$$x_k^{(g+1)} \in \mathbb{R}^n : k\text{-ésimo individuo de la generación } g+1.$$

$$m^{(g)} \in \mathbb{R}^n : \text{media de la generación } g.$$

$$\sigma^{(g)} \in \mathbb{R}_+ : \text{desviación típica global o tamaño de paso de la generación } g.$$

$C^{(g)} \in \mathbb{R}^{n \times n}$  : matriz de covarianzas de la generación  $g$ , que determina la forma del elipsoide de la distribución normal multivariante.

$$\lambda \geq 2 : \text{número de descendientes.}$$

La siguiente cuestión es cómo calcular  $m$ ,  $C$  y  $\sigma$  para la siguiente generación.

### 6.3.1.2 Selección y Recombinación: Cálculo de la Nueva Media

La nueva media  $m^{(g+1)}$  es la media ponderada de los  $\mu$  puntos seleccionados de la población  $x_k^{(g+1)}$ :

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)}$$

Donde:

$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq w_2 \geq \dots \geq w_{\mu} > 0$$

La selección de los  $\mu$  mejores individuos ( $x_{i:\lambda}^{(g+1)}$ ), tal que  $f(x_{1:\lambda}^{(g+1)}) \geq f(x_{2:\lambda}^{(g+1)}) \geq \dots \geq f(x_{\mu:\lambda}^{(g+1)})$  puede equipararse a la *selección por truncamiento* definida en el apartado 5.2.2, mientras que la adaptación de la media asignando un peso a cada individuo se denomina *recombinación ponderada*.

Utilizando los pesos asignados se calcula otro parámetro, que denominamos *variance effective selection mass*, y que será utilizado en adelante.

$$\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$$

### 6.3.1.3 Adaptación de la Matriz de Covarianzas

La adaptación de la matriz de covarianzas incrementa la probabilidad de que aparezcan pasos exitosos en las siguientes generaciones. Aprende las dependencias dos a dos de las variables utilizando el análisis de componentes principales (PCA) de los pasos  $y$ .

La actualización se lleva a cabo siguiendo el siguiente principio:

$$C^{(g+1)} = (1 - c_{\text{cov}})C^{(g)} + c_{\text{cov}} \sum_{i=1}^{\mu} w_i y_w y_w^T$$

Donde:

- $c_{\text{cov}}$ : tasa de aprendizaje de la matriz de covarianzas.  $c_{\text{cov}} \approx 2/n^2$
- $y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda}$  siendo  $y_i = N_i(0, C)$

Esta actualización se realiza combinando dos métodos:

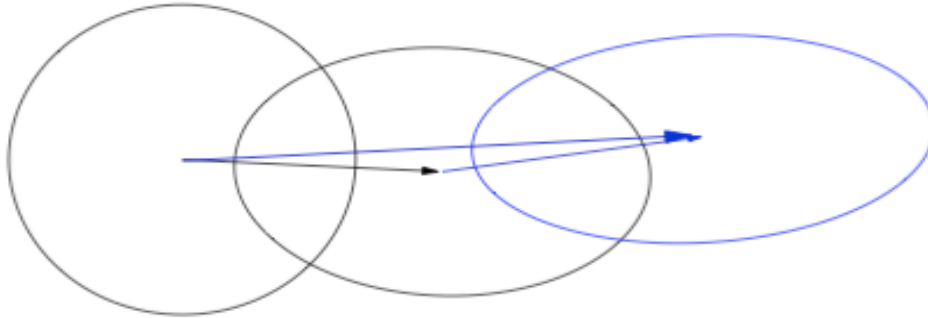
1. **Actualización Rank- $\mu$** : importante para poblaciones grandes.
2. **Actualización Rank-one**: importante para poblaciones pequeñas.

### Actualización Rank-one

Utiliza el *camino de evolución (evolution path)* para reducir el número de evaluaciones de la función de fitness.

El *camino de evolución* suma los pasos consecutivos de la media  $m$  almacenando información histórica acumulada.

**Figura 9:** Camino de evolución de la media



Utiliza una suma de los pasos  $y_w$  ponderada exponencialmente:

$$p_c \propto \sum_{i=1}^g (1 - c_c^{g-i}) y_w^{(i)}$$

La actualización del *camino de evolución* tiene lugar de la siguiente manera:

$$p_c^{(g+1)} = (1 - c_c) p_c^g + \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w} y_w$$

Donde  $c_c \ll 1$

De modo que la actualización *Rank-one* tiene lugar según la siguiente fórmula:

$$C^{(g+1)} = (1 - c_{\text{cov}}) C^{(g)} + c_{\text{cov}} p_c p_c^T$$

## Actualización Rank- $\mu$

Rank- $\mu$  extiende la regla de adaptación a poblaciones grandes, utilizando  $\mu$  vectores para actualizar  $C$  en cada generación.

Incrementa la tasa de aprendizaje en poblaciones grandes, reduciendo el número de generaciones necesarias.

$$C_\mu = c_{\text{cov}} \sum_{i=1}^{\mu} w_i y_w y_w^T$$

Siendo la actualización Rank- $\mu$  de la siguiente manera:

$$C^{(g+1)} = (1 - c_{\text{cov}}) C^{(g)} + c_{\text{cov}} C_\mu$$

## Combinación de Rank-one y Rank- $\mu$

La matriz de covarianzas se actualiza combinando ambos métodos, dando lugar a la siguiente ecuación:

$$C^{(g+1)} = (1 - c_1 - c_\mu)C^{(g)} + c_1 P_c^{(g+1)} P_c^{(g+1)T} + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)} (y_{i:\lambda}^{(g+1)})^T$$

Donde:

$c_1 \approx 2/n^2$  : tasa de aprendizaje de Rank-one.

$c_\mu \approx \min(\mu_w/n^2, 1 - c_1)$  : tasa de aprendizaje de Rank- $\mu$ .

$y_{i:\lambda}^{(g+1)} = (x_{i:\lambda}^{(g+1)} - m^{(g)})/\sigma^{(g)}$ .

$P_c^{(g+1)}$  : *camino de evolución* de la matriz de covarianzas.

$c_c$  : tasa de aprendizaje del *camino de evolución* de la matriz de covarianzas.

### 6.3.1.4 Tamaño del paso

La matriz de covarianzas no controla explícitamente el tamaño del paso global. Controla la importancia relativa que tendrán la información nueva y la antigua, atenuando el peso de esta última con el factor  $1 - c_1 - c_\mu$ , dirigiéndonos hacia una nueva zona del dominio en busca de mejores soluciones.

La principal razón para que es necesario un control del tamaño de paso además de la matriz de covarianzas es que la tasa de aprendizaje de actualización de la matriz de covarianzas es muy lenta para alcanzar tasas competitivas para el paso global.

En otras palabras, si los pasos sucesivos se van cancelando, es decir, damos vueltas en torno a una misma zona, es que estamos cerca de la solución óptima y por lo tanto acortaremos el paso global. Si por el contrario los pasos sucesivos nos llevan en una misma dirección es que estamos lejos de la solución óptima y por tanto necesitamos alargar la longitud de los pasos.

Para controlar el tamaño de paso utilizamos el “camino de evolución del tamaño de paso”. Esta variable no es más que la suma de los pasos sucesivos, es decir, cuánto y en qué dirección hemos avanzado en los últimos pasos.

Para decidir si el camino de evolución es largo o corto lo comparamos con la longitud del camino bajo selección aleatoria.

Para calcular el camino de evolución utilizamos la siguiente ecuación:

$$p_{\sigma}^{(g+1)} = (1 - c_{\sigma})p_{\sigma}^{(g)} + \sqrt{c_{\sigma}(2 - c_{\sigma})}\mu_w C^{(g)-\frac{1}{2}} \frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$$

Donde:

$p_{\sigma}^{(g)} \in \mathbb{R}^n$  : camino de evolución de la generación  $g$ .

$c_{\sigma} < 1$  : tasa de aprendizaje del camino de evolución.

$\sqrt{c_{\sigma}(2 - c_{\sigma})}\mu_w$  : constante de normalización.

Entonces, para actualizar  $\sigma^{(g)}$  comparamos  $p_{\sigma}^{(g+1)}$  con su valor esperado bajo selección aleatoria y nos queda la siguiente expresión:

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_{\sigma}}{d_{\sigma}} \left(\frac{\|p_{\sigma}^{(g+1)}\|}{E\|N(0, I)\|} - 1\right)\right)$$

Donde:

$d_{\sigma} \approx 1$  : factor de amortiguamiento del cambio de magnitud del paso.

### 6.3.2 Configuración de los Parámetros del Algoritmo.

CMA-ES no requiere una configuración de parámetros demasiado exhaustiva. Los autores consideran parte del diseño del algoritmo la configuración de los parámetros de estrategia, por lo que recomiendan la implementación por defecto en la mayor parte de éstos.

Siguiendo las recomendaciones de los diseñadores, realizamos pruebas exploratorias para fijar tamaño de población, sigma inicial y algunos criterios de finalización. Para ello, utilizamos una población de 14 individuos y 28 descendientes (configuración de los diseñadores).

Con CMA-ES, el tiempo de ejecución supuso una limitación grande, al ser de orden mucho mayor que con el resto de algoritmos, por lo que el procedimiento experimental para fijar los parámetros consistió en pruebas de 10 ejecuciones para cada opción, analizando posteriormente la media y desviación típica del valor de fitness.

#### 6.3.2.1 Sigma: Desviación Típica Inicial

CMA-ES recomienda una sigma inicial de valor 1/3 del dominio de cada variable. Dado que el dominio de nuestro problema es en algunas variables muy superior a los resultados habituales, hemos probado varias fórmulas para comprobar cual arrojaba los mejores resultados empíricos.

A continuación se muestran los principales resultados:

SIGMA	FMEDIA	FDESV
[]	0,00263	1,71669E-10
[16; 16; 100; 100; 100; 100]	0,00263	1,6014E-10
[16; 16; 20; 20; 20; 20]	0,01699	0,02591
10	0,00235	0,00079
15	0,02086	0,03033
20	0,00856	0,01341

Donde

- $Sigma = [ ]$  corresponde a la configuración de los diseñadores, y devuelve la raíz cuadrada de la varianza de las componentes del valor inicial de  $x$ .
- $Sigma = [\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6]$  corresponde a una sigma inicial diferente para cada variable.
- $Sigma = [\sigma]$  corresponde a la misma sigma inicial para todas las variables.

Escogimos sigma=10, por obtener los mejores resultados empíricos.



### 6.3.2.2 Tamaño de la Población

En CMA-ES el tamaño de población indica el número de descendientes, es decir, la población intermedia sobre la cual se escogerán los padres de la siguiente generación.

En la configuración aconsejada por los diseñadores, el tamaño de la población es una función dependiente del número de variables, que en nuestro caso devuelve un valor de 28 individuos:

$$4 * [4 + \text{floor}(3 * \log(N))]$$

Donde  $N$  es el número de variables (en nuestro caso 6) y  $\text{floor}$  indica la parte entera del resultado del interior del paréntesis.

Realizamos pruebas diferentes tamaños de población, obteniendo los siguientes resultados:

Tamaño de Población	FMEDIA	FDESV
$4 * (4 + \text{floor}(3 * \log(N)))$	0,00235	0,00079
50	0,00264	0,000282
100	0,00254	0,000475

En los resultados empíricos no se observa una relación clara con el tamaño de población con los parámetros fijados hasta el momento.

Fijamos el tamaño de la población en 50 individuos.

### 6.3.2.3 Número de Padres

Para este parámetro utilizamos el valor aconsejado por los diseñadores del algoritmo, una función del número de descendientes ( $PopSize$ ):

$$Padres = \min(PopSize / 2)$$

### 6.3.2.4 Criterios de Finalización

Durante el resto de la experimentación, hubo dos criterios de finalización que mostraron importancia en determinadas ejecuciones que devolvieron resultados peores a los habituales:

- *TolHistFun*: detiene la ejecución si el rango de cambios en el valor de fitness de los mejores individuos de anteriores iteraciones es menor que este parámetro.
- *TolFun*: detiene la ejecución si el rango de cambios en el valor de fitness entre la nueva generación y los mejores individuos de anteriores iteraciones es menor que este parámetro.

Tras comprobar que al aumentar estos valores desaparecían estos resultados fijamos estas opciones en:

- *TolHistFun*=  $1e-16$
- *TolFun*=  $1e-15$

### 6.3.3 Experimentación

Siguiendo el procedimiento descrito en el apartado 5.3, realizamos diez ejecuciones sobres un rango de treinta días.

A continuación mostramos el resumen de los resultados obtenidos siguiendo cada uno de los métodos.

#### Aportando el mismo valor inicial en cada ejecución para cada día

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00226	0,00642
<b>T. Ejecución (seg)</b>	1117,19192	823,67268

#### Aportando la mejor solución del día anterior como valor inicial para el siguiente día.

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00107	0,00079
<b>T. Ejecución (seg)</b>	618,80773	435,66437

#### Un rearranque doblando la población aportando la mejor solución del día anterior como valor inicial para el siguiente día.

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00097	0,00069
<b>T. Ejecución (seg)</b>	1649,12767	1029,96163

Los resultados desagregados se detallan en el *Anexo de Resultados*.



# Capítulo 7

## Combinación de (3,60)-EE y SQP

### 7.1 Introducción

Tras analizar la experimentación realizada hasta este punto, observamos que dos técnicas sobresalen respecto al resto:

1. SQP aportando la mejor solución del día anterior como valor inicial para el siguiente día.
2. (3,60)-EE con cuatro rearranques aportando la mejor solución del día anterior como valor inicial para el siguiente día.

Las medias globales del error cuadrático en ambos métodos son de orden inferior al resto ( $10^{-4}$ ) y muy parecidas entre sí, siendo las medias diarias similares en todos los días como se puede apreciar en la figura 10.

Figura 10: Resultados diarios SQP II, EE III



La figura 10 muestra la media de los resultados diarios de la experimentación con los métodos SQP II y EE III, donde

- **EE III:** estrategia evolutiva con cuatro rearranques aportando la mejor solución del día anterior como valor inicial para el siguiente día.
- **SQP II:** SQP aportando la mejor solución del día anterior como valor inicial para el siguiente día.

Ambos métodos obtienen resultados muy similares para todos los días, siendo los de SQP II ligeramente superiores en la mayoría de los puntos.

Figura 11: Resultados diarios SQP I, EE I.



La figura 11 muestra la media de los resultados diarios de la experimentación con los métodos SQP I y (3,60)-EE I, es decir, utilizando el valor inicial calculado a partir de los datos de los bonos para cada día.

Si comparamos estos resultados con los obtenidos con los métodos SQP II y (3,60)-EE III, se puede apreciar la sensibilidad de ambos algoritmos ante las condiciones iniciales.

El algoritmo SQP converge rápidamente hacia buenas soluciones partiendo de un valor inicial adecuado. Ya que la (3,60)-EE suele alcanzar en 500 generaciones valores de fitness próximos al 90% del valor que alcanzaría con 5000, consideramos que esta podría ser una buena aproximación como valor inicial para el método SQP.

De modo que probamos una combinación de ambos algoritmos en la que se busca un buen valor inicial con la (3,60)-EE y utilizamos el algoritmo SQP para hacer una optimización local. La EE haría en buena medida el trabajo de exploración y SQP el de explotación.

## 7.2 Configuración de los Parámetros del Algoritmo

Las configuraciones de parámetros de la (3,60)-EE y el algoritmo SQP son las mismas que en los apartados 6.2.2 y 4.4.3 respectivamente, exceptuando el número de generaciones de la EE que fijamos en 500.

## 7.3 Experimentación

El método implementado para esta parte de la experimentación es análogo al explicado en el apartado 4.1.3 (aportando la mejor solución del día anterior como valor inicial para el siguiente día), pero adaptado a la combinación de algoritmos.

De modo que introducimos el valor inicial calculado con los datos de los bonos para la (3,60)-EE en el primer día, utilizando posteriormente la solución como valor inicial para SQP. Para el resto de días utilizamos la solución del día anterior como valor inicial para la EE, y la solución de esta como valor inicial para SQP. Repetimos este proceso hasta alcanzar diez ejecuciones sobre el rango de treinta días.

A continuación se muestra el resumen de los resultados:

	<b>MEDIA</b>	<b>DESVIACIÓN</b>
<b>Fmin</b>	0,00039	0,00043
<b>T. Ejecución (seg)</b>	13,80515	2,49543

Los resultados desagregados se detallan en el *Anexo de Resultados*.



# Capítulo 8

## Evaluación de los Resultados

Recordamos que el objetivo de este proyecto es analizar el comportamiento de técnicas evolutivas en el dominio de búsqueda del problema, así como compararlo con el de algunos métodos tradicionales.

La sensibilidad ante las condiciones iniciales supone un aspecto clave en este análisis, ya que partimos de la premisa de riesgo de falsa convergencia con los métodos tradicionales.

Para ello realizamos el siguiente análisis:

1. **Comparación de los resultados globales:** comparamos las medias globales de todo el rango de días y analizamos la significación estadística de los resultados.
2. **Comparación de los resultados de cada día:** comparamos las medias de cada día por separado.

## 8.1 Comparación de los Resultados Globales

Recordamos que la experimentación consistió en 10 ejecuciones sobre un rango de 30 días para los métodos estocásticos, y en una ejecución sobre un rango de 30 días para los métodos tradicionales.

Asimismo utilizamos varios métodos con cada algoritmo para analizar la sensibilidad ante las condiciones iniciales<sup>14</sup>.

A continuación se muestran los resultados globales obtenidos en la parte experimental para cada algoritmo:

	<b>FMEDIA</b>	<b>FDESV</b>	<b>TMEDIO</b>	<b>TDESV</b>
<b>AG I</b>	0,00114	0,00086	590,43432	130,75279
<b>AG II</b>	0,00090	0,00079	715,17674	100,41044
<b>AG III</b>	0,00076	0,00064	2748,22772	1047,17682
<b>(3,60)-EE I</b>	0,00053	0,00063	101,70155	6,86698
<b>(3,60)-EE II</b>	0,00041	0,00045	99,07406	6,33476
<b>(3,60)-EE III</b>	0,00039	0,00041	497,48327	32,86654
<b>CMA-ES I</b>	0,00226	0,00642	1117,19191	823,67268
<b>CMA-ES II</b>	0,00107	0,00079	618,80774	435,66437
<b>CMA-ES III</b>	0,00097	0,00069	1649,12767	1029,96163
<b>IP I</b>	0,00165	0,00123	0,00165	29,18308
<b>IP II</b>	0,00111	0,00080	0,00111	19,60015
<b>AS I</b>	0,00210	0,00134	0,00210	0,44627
<b>AS II</b>	0,00209	0,00134	0,00209	0,47014
<b>SQP I</b>	0,00052	0,00043	0,00052	2,73291
<b>SQP II</b>	0,00038	0,00041	0,00038	2,98301
<b>Combinación</b>	0,00039	0,00043	13,80515	2,49543

Donde:

- **FMEDIA, FDESV, TMEDIO y TDESV** son la media y desviación típica de la función de fitness, y la media y desviación típica del tiempo de ejecución respectivamente.
- **EE:** (3,60)-EE.
- **IP:** Interior-Point.
- **AS:** Active-Set.

<sup>14</sup> Los métodos utilizados están descritos en el apartado 4.1.3 para los métodos tradicionales, y en el apartado 5.3 para las técnicas evolutivas.

- **Método I:** utilizando los datos de los bonos del día para estimar el valor inicial.
- **Método II:** utilizando la solución del día anterior como valor inicial para el siguiente.
- **Método III:** realizando rearranques utilizando la solución del día anterior como valor inicial para el siguiente.

Atendiendo a la media global, tanto los métodos tradicionales como las técnicas evolutivas han mostrado cierta sensibilidad ante las condiciones iniciales, obteniendo mejores resultados utilizando la mejor solución del día anterior como valor inicial para el siguiente día<sup>15</sup>.

En la siguiente tabla se muestran los resultados obtenidos con cada algoritmo y método, así como el porcentaje de mejora de los métodos II y III respecto al primero:

	MÉTODO I	MÉTODO II		MÉTODO III	
	MEDIA	MEDIA	MEJORA (%)	MEDIA	MEJORA (%)
Interior Point	0,00165	0,00111	32,4	NA	NA
Active Set	0,00210	0,00209	0,4	NA	NA
SQP	0,00052	0,00038	26,6	NA	NA
AG	0,00114	0,00090	21,0	0,00076	33,4
(3,60)-EE	0,00053	0,00041	22,8	0,00039	26,1
CMA-ES	0,00226	0,00107	52,5	0,00097	57,1
Combinación	NA	0,00039	NA	NA	NA

Donde:

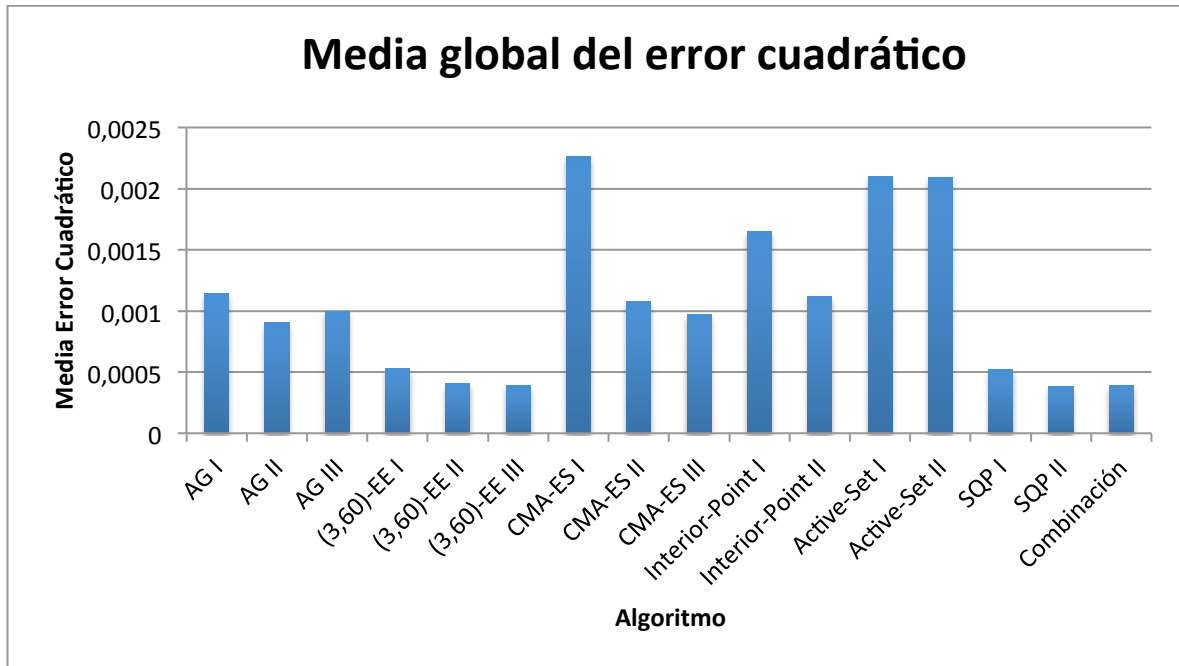
- **Técnicas evolutivas:** AG, (3,60)-EE, CMA-ES.
- **Métodos tradicionales:** Interior Point, Active Set y SQP.

Como podemos observar, las mayores diferencias al variar las condiciones iniciales corresponden a CMA-ES, siendo muy inferiores y similares entre sí con la (3,60)-EE, el AG, SQP e Interior-Point, mientras que en Active Set son mínimas. **Estos resultados contradicen la tesis de mayor sensibilidad ante las condiciones iniciales de los métodos tradicionales.**

Tanto en las tres técnicas evolutivas como en el algoritmo SQP y en el AG III se han alcanzado errores globales del orden de  $10^{-4}$  utilizando el mejor individuo del anterior día anterior como valor inicial del siguiente. No obstante, los mejores resultados globales corresponden el algoritmo SQP, con un coste computacional ínfimo en comparación con las técnicas evolutivas.

<sup>15</sup> Recordamos que la mejor solución del día anterior supone un buen valor inicial para el siguiente día, ya que los tipos de interés no suelen variar bruscamente entre días cercanos.

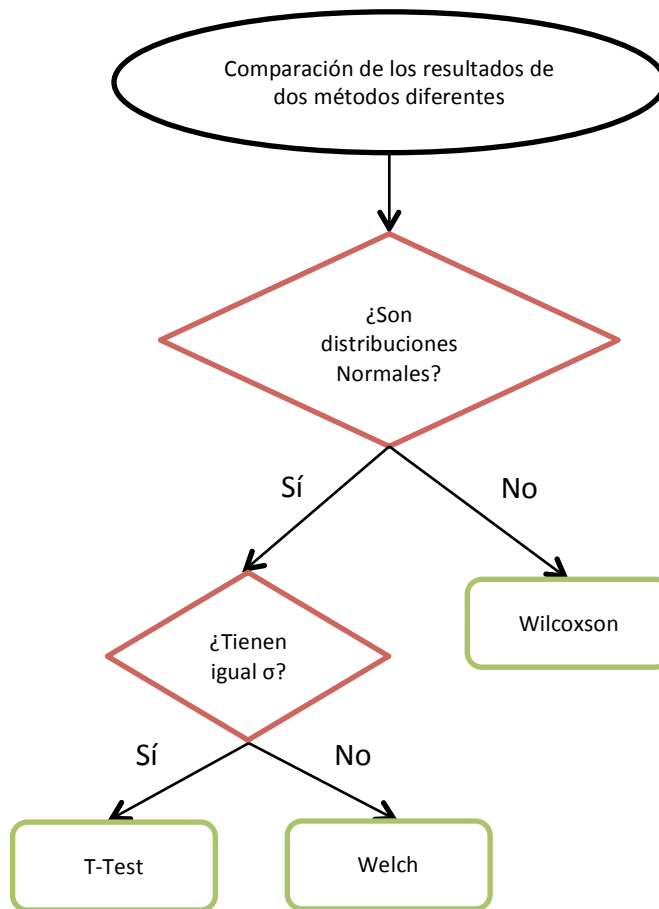
Figura 12: Resultados globales.



Para estudiar la significación estadística de los resultados utilizamos el entorno de programación R, descrito en el apartado 3.3.4 *Medios Empleados*.

Realizamos una serie de contrastes para comprobar si los resultados de los distintos métodos son diferentes significativamente según el siguiente proceso:

**Figura 13:** Análisis de la significación de los resultados.



Para determinar si se trata de distribuciones normales, utilizamos la **prueba de normalidad de Lilliefors**, un contraste no paramétrico que se utiliza para determinar la bondad de ajuste con una distribución normal.

Para contrastar las dos muestras cuando la prueba de normalidad es negativa, utilizamos la **prueba de los signos de Wilcoxon**, un contraste no paramétrico para comparar la mediana de dos muestras relacionadas y determinar si existen diferencias entre ellas. Se utiliza como alternativa a la prueba t de Student cuando no se puede suponer la normalidad de dichas muestras.

Cuando la prueba de normalidad es positiva, determinamos si las varianzas de las muestras se pueden considerar iguales mediante el **test de Levene de homogeneidad de varianzas**.

Cuando la prueba de normalidad es positiva y las varianzas son iguales, utilizamos **test-T o prueba t de Student**. Se trata de un contraste en el que el estadístico utilizado tiene una distribución t de Student si la hipótesis nula es cierta. Se aplica cuando la población se asume normal pero el tamaño muestral es demasiado pequeño como para que el estadístico en el que está basada la inferencia esté normalmente distribuido, utilizándose una estimación de la desviación típica en lugar del valor real.

Si la prueba de normalidad es positiva pero las varianzas de las dos muestras son diferentes, realizamos la **prueba t de Welch**, que se utiliza para contrastar hipótesis en función de la media aritmética cuando dada la diferencia de las varianzas, no es aplicable la prueba t de Student.

	AG I	AG II	AG III	EE I	EE II	EE III	CMAES I	CMAES II	CMAES III	IP I	IP II	SQP I	SQP II	AS I	AS II
AG II	--														
AG III	--	=													
EE I	--	--	=												
EE II	--	--	-	--											
EE III	--	--	--	--	--										
CMAES I	++	++	++	++	++	++									
CMAES II	--	=	=	++	++	++	--								
CMAES III	--	=	=	++	++	++	--	--							
IP I	++	++	++	++	++	++	-	++	++						
IP II	--	+	=	++	++	++	--	=	++	--					
SQP I	--	--	=	=	++	++	--	--	--	--	--				
SQP II	--	--	-	--	--	-	--	--	--	--	--	--			
AS I	++	++	++	++	++	++	=	++	++	++	++	++	++		
AS II	++	++	++	++	++	++	=	++	++	++	++	++	++	--	
C	--	--	=	=	=	=	--	--	--	--	--	=	=	--	--

Donde

++: El estadístico de la muestra del método de la fila es mayor que el del método de la columna con un nivel de significación del 1%.

+: El estadístico de la muestra del método de la fila es mayor que el del método de la columna con un nivel de significación del 5%.

--: El estadístico de la distribución del método de la fila es menor que el del método de la columna con un nivel de significación del 1%.

-: El estadístico de la muestra del método de la fila es mayor que el del método de la columna con un nivel de significación del 5%.

=: No podemos descartar que los errores de predicción medios sean iguales.

Si comparamos el comportamiento de cada algoritmo con los diferentes métodos comprobamos que en todos los resultados son mejores con el método II que con el I con un

nivel de significación del 1%. Según este contraste, todos los algoritmos muestran sensibilidad ante las condiciones iniciales.

Asimismo se puede considerar que las estrategias evolutivas mejoran con el método III respecto al II con un nivel de significación del 1%, mientras que con el AG no podemos considerar significativa la diferencia.

Por último, si comparamos los resultados de los diferentes algoritmos, podemos concluir que el método SQP II se puede considerar mejor que el resto de algoritmos con un nivel de significación del 1%, a excepción de (3,60)-EE III y la combinación de ambos. Respecto al método (3,60)-EE III se puede considerar mejor con un nivel de significación del 5%, mientras que respecto a la combinación no muestra diferencia significativa.

## 8.2 Comparación de los Resultados Diarios

La figura 13 muestra los resultados obtenidos en la parte experimental para cada día y algoritmo.

Los métodos CMA-ES I y AG I y los algoritmos Interior-Point y Active-Set en todas sus variantes obtienen resultados de orden inferior al resto.

Figura 14: Medias diarias del error cuadrático.

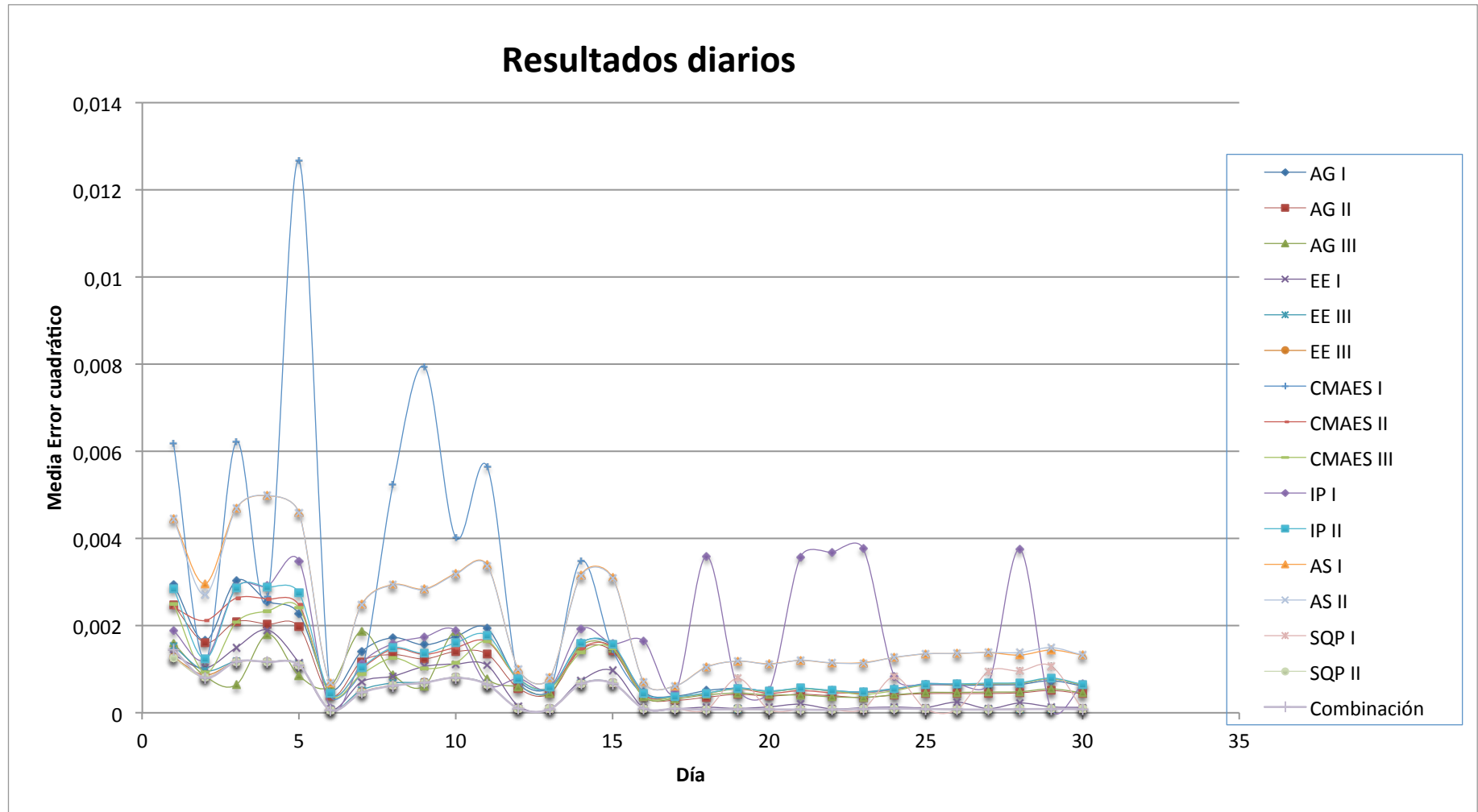
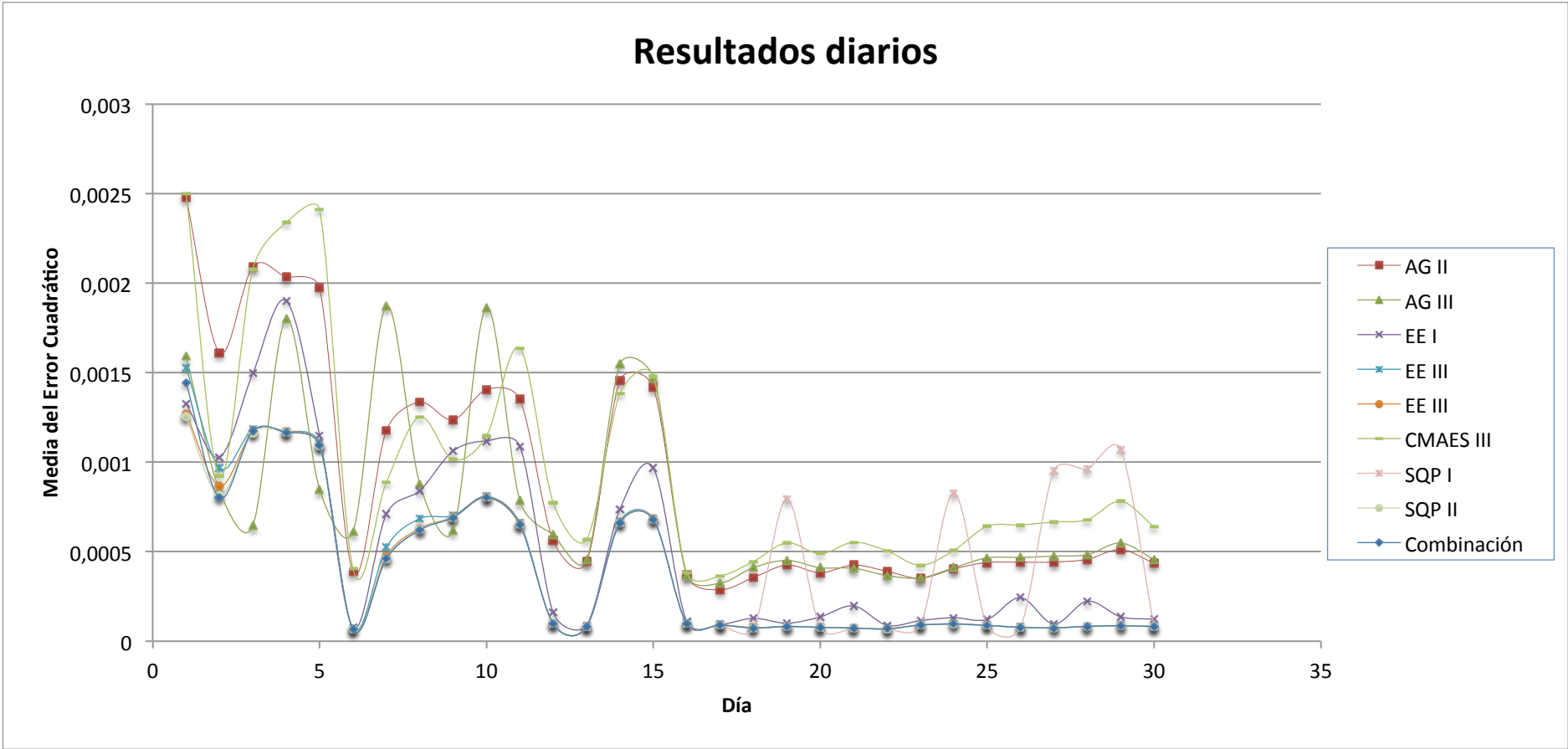




Figura 15: Medias diarias del error cuadrático (orden media global  $\leq 10^{-4}$ )



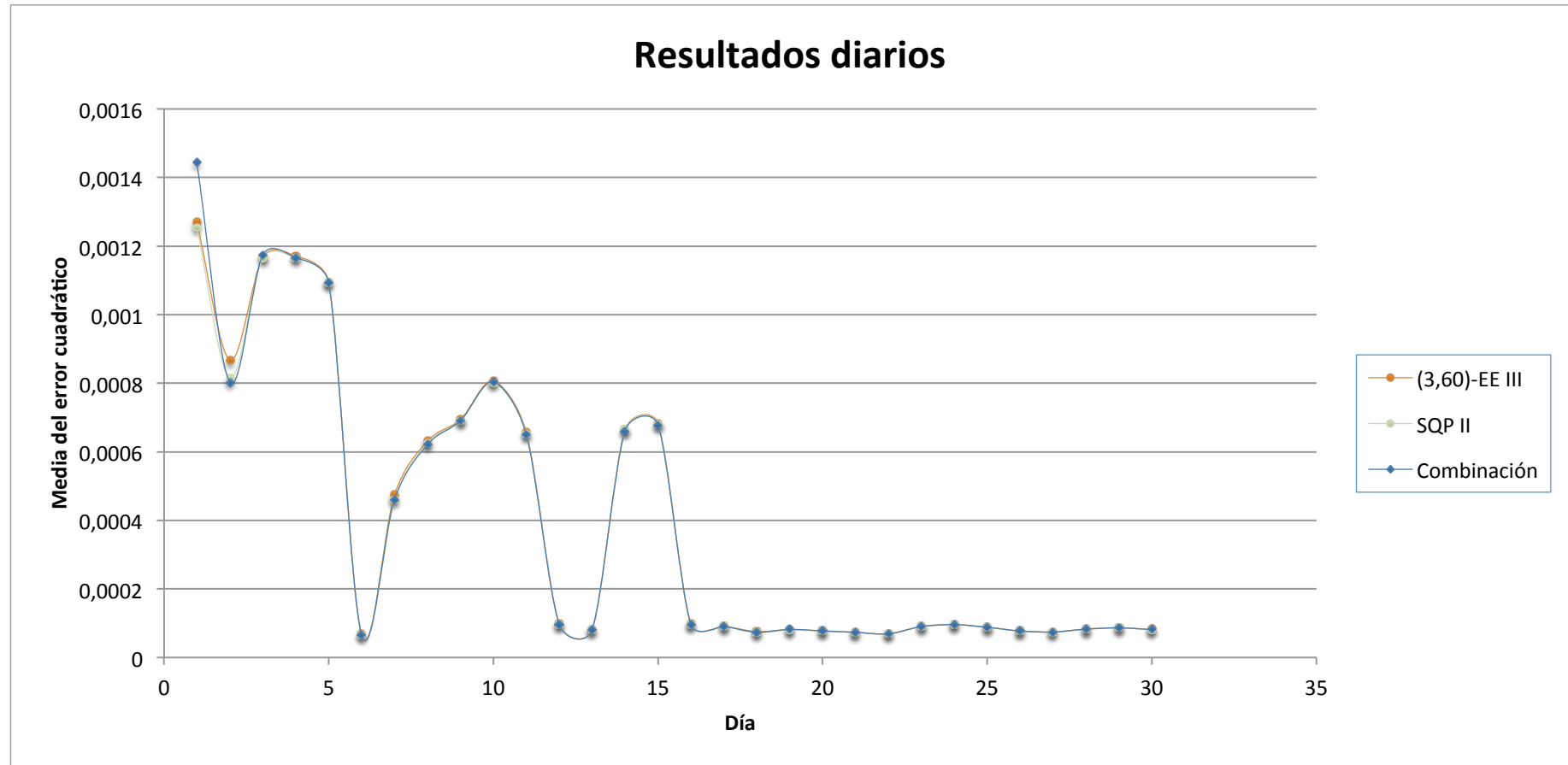
En la figura 14 eliminamos los métodos con medias globales de orden  $10^{-3}$ : AG I, CMA-ES I, CMA-ES II, Interior-Point I, Interior-Point II, Active-Set I y Active-Set II. En él se observa mejor cómo el algoritmo SQP y las variantes de la (3,60)-EE obtienen resultados muy similares para casi todos los días, probablemente cercanos al óptimo, mientras que el AG y CMA-ES obtienen resultados más irregulares.

Figura 16: Medias diarias del error cuadrático (SQP y (3,60)-EE)



En la figura 15 se puede observar la sensibilidad de SQP y (3,60)-EE a las condiciones iniciales, obteniendo ambos resultados más irregulares utilizando los datos de los bonos para calcular el valor inicial para todos los días.

Figura 17: Medias diarias del error cuadrático (SQP, EE y su combinación)



En la figura 16 se observa que las técnicas SQP II, (3,60)-EE III y Combinación convergen hacia los mismos valores en casi todos los días, por lo que probablemente éstos sean muy cercanos al óptimo, aunque SQP II es ligeramente mejor en la mayoría de los días.

Tal y como cabía esperar en vista de los errores agregados presentados anteriormente, la estimación del valor inicial para SQP con la (3,60)-EE no supone mejoría respecto al método II con los datos utilizados. Dicho esto, la estrategia podría serlo potencialmente en un escenario en el que los tipos de interés pudieran no variar de manera continua de un día para otro a causa de algún shock en el mercado.

# Capítulo 9

## Conclusiones y Líneas Futuras

### 9.1 Conclusiones

Recordamos que el objetivo de este proyecto es analizar el comportamiento de técnicas evolutivas para la estimación de curvas de tipo de interés mediante el modelo de Svensson, y compararlo con el de algunos métodos tradicionales. Para ello ajustamos los parámetros del modelo en treinta días diferentes, utilizando un algoritmo genético, dos estrategias evolutivas, un método de punto interior y dos de programación secuencial cuadrática.

Tras llevar a cabo el trabajo experimental y análisis de los resultados, sacamos las siguientes conclusiones:

- Atendiendo a la media global del error cuadrático ponderado de las ejecuciones realizadas, tanto los métodos tradicionales como las técnicas evolutivas han mostrado cierta sensibilidad ante las condiciones iniciales, obteniendo mejores resultados utilizando la mejor solución del día anterior como valor inicial para el

siguiente<sup>16</sup> frente al método de utilizar una estimación basada en los datos de los bonos de cada día<sup>17</sup>.

- Las mayores diferencias porcentuales al variar las condiciones iniciales corresponden a CMA-ES, siendo muy inferiores y similares entre sí con la (3,60)-EE, el AG, SQP e Interior-Point, mientras que con Active-Set son mínimas. Estos resultados contradicen la tesis de mayor sensibilidad ante las condiciones iniciales de los métodos tradicionales.
- Las tres técnicas evolutivas han mejorado los resultados notablemente con el método de realizar rearranques, siendo el AG el más afectado atendiendo al porcentaje de mejora, mientras que según los contrastes estadísticos no podemos considerar esta diferencia significativa para el AG pero sí para las estrategias evolutivas con un nivel de significación del 1%.
- Los mejores resultados globales corresponden el algoritmo SQP, aunque los contrastes estadísticos no han mostrado una diferencia significativa frente la combinación de (3,60)-EE y SQP.
- Tanto el coste computacional de SQP como el de los otros algoritmos tradicionales es muy inferior al de las técnicas evolutivas.
- No hemos encontrado una diferencia general clara entre las técnicas evolutivas y los métodos tradicionales empleados, a parte de los tiempos de ejecución.
- Existe diferencia en la dificultad de ajuste entre los días evaluados, siendo los errores cuadráticos alcanzados muy superiores a la media en algunos días.

## 9.2 Líneas Futuras

En el desarrollo del proyecto hemos analizado el comportamiento de diferentes técnicas evolutivas para estimar la curva de tipos de interés mediante el modelo de Svensson, utilizando como función de fitness la WSSE<sup>18</sup>.

Hemos utilizado el valor del fitness y su desviación como únicos elementos de análisis. Una posible línea de continuación podría ser una ampliación comparando las estimaciones de los parámetros con su significado económico, así como estudiando su variabilidad en estimaciones de un mismo día.

Como ampliación también se podría intentar mejorar el análisis de técnicas evolutivas incluyendo subpoblaciones a los métodos implementados, para reducir así la sensibilidad

---

<sup>16</sup> Recordamos que la mejor solución del día anterior supone un buen valor inicial para el siguiente día, ya que la estructura temporal de los tipos de interés no suelen variar bruscamente entre días cercanos.

<sup>17</sup> Explicada en el apartado 4.1.3.

<sup>18</sup> Descrita en el apartado 3.1.2 *Función Objetivo*.

ante las condiciones iniciales, o intentar explotar la idea de combinar técnicas evolutivas con métodos tradicionales.

Otro factor de estudio que no hemos abordado es el comportamiento en función de las restricciones de los parámetros. Éste enfoque podría resultar interesante, ya que las investigaciones se suelen centrar en los modelos para ajustar las curvas y sus estimaciones, utilizando restricciones para procurar obtener un sentido económico fidedigno o acotar el espacio de búsqueda, pero sin analizar el comportamiento en función de éstas.

La selección de la función de fitness también juega un papel importante en el ajuste de la estimación de la curva, por lo que un análisis de la variabilidad de los parámetros ajustados y de su significado económico con diferentes implementaciones del error podría suponer otra línea interesante.

También puede resultar interesante la implementación de un framework interactivo que utilice diferentes modelos y técnicas para ajustar la curva de tipos de interés, realizando la selección de la opción más conveniente mediante interacción con el usuario.





# Capítulo 10

## Presupuesto

### 10.1 Fases del Proyecto

Como ya comentamos en el apartado 1.3 *Fases de Desarrollo*, el proyecto consta de cuatro fases:

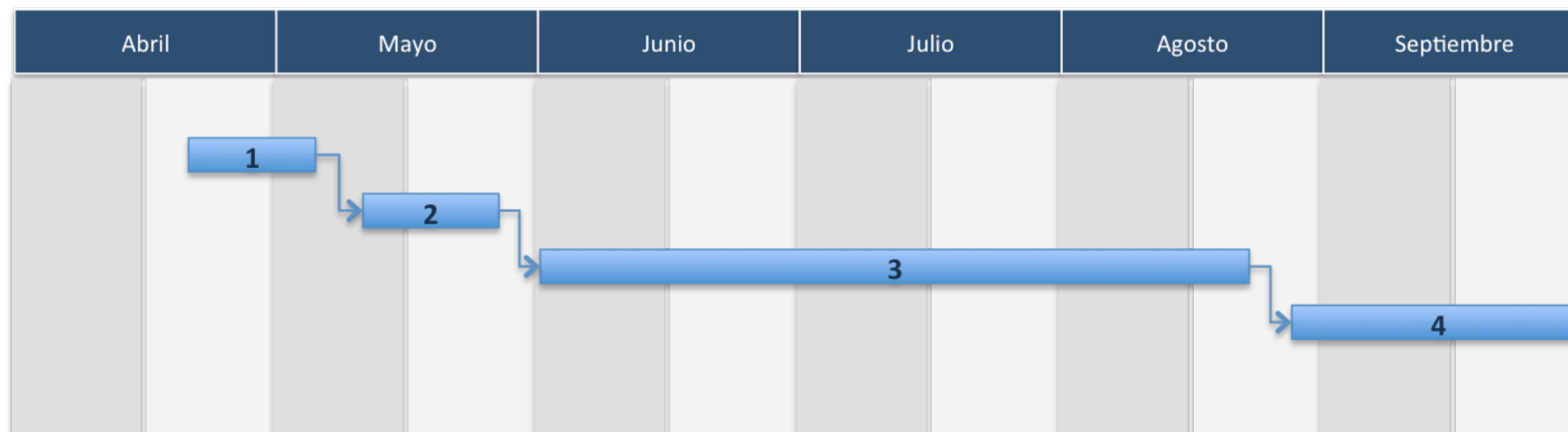
1. **Estudio previo:** estudio del contexto; análisis del problema.
2. **Selección del instrumental:** búsqueda y selección de librerías de CE; búsqueda y selección de software de programación no-lineal; selección del entorno de trabajo.
3. **Trabajo experimental:** configuración de los parámetros de los algoritmos; experimentación.
4. **Análisis de los resultados y documentación:** redacción de la memoria paralelamente al análisis de los resultados .

Dado que el proyecto ha estado estructurado de manera irregular en el tiempo, realizamos una estimación de la duración de cada fase para una jornada laboral de ocho horas diarias y cinco días a la semana.

La figura 18 muestra el diagrama de Gantt del desarrollo del proyecto.

**Figura 18:** Diagrama de Gantt

ID	Tarea	Comienzo	Fin	Trabajo
1	Estudio previo	18/04/11	06/05/11	120 horas
2	Selección del instrumental	09/05/11	27/05/11	120 horas
3	Trabajo experimental	30/05/11	19/08/11	480 horas
4	Análisis de resultados y documentación	22/08/11	30/09/11	240 horas



## 10.2 Costes

### Coste materiales

A continuación enumeramos el instrumental utilizado en la ejecución del proyecto y su coste:

- Licencia Matlab para estudiantes: 66 €
- Licencia GEATbx para uso universitario: 297,5 €
- Microsoft Office para Mac Home y Student 2011: 110 €
- MacBook Pro (13 pulgadas; 2,3 GHz): 1.149 €
- Material oficina: 30 €

El coste total de los materiales asciende a: 1.652,5 €

### Coste personal

Hemos estimado el coste de horas de ingeniero industrial superior en 32 €<sup>19</sup>.

El coste total de personal asciende a: 960 horas de ingeniero industrial x 32 €/hora=30.720 €.

### Coste total

El presupuesto total de este proyecto asciende a la cantidad de 32.372,5 euros.

---

<sup>19</sup> El precio medio/hora de ingenieros a partir de una muestra de 11 profesionales y empresas en toda España es de 31,78 € según el directorio de empresas de la página web (habitissimo).



# Referencias

---

- Annaert, J., G.P. Claes, A., J.K. De Ceuster, M., & Zhang, H. *Estimating the Yield Curve Using the Nelson-Siegel Model: A Ridge Regression Approach*. Working Paper, Universiteit Antwerpen.
- Diebold, F. X., Li, C., & Yue, V. Z. (2008). Global Yield Curve Dynamics and Interactions: A Generalized Nelson-Siegel Approach. *Journal of Econometrics*, 146, 351-363.
- Vasicek, O. A. (1997). An Equilibrium Characterization of the Term Structure. *Journal of Financial Economics*, 5 (2), 177-188.
- Vasicek, O. A., & Fong, H. F. (1982). Term Structure Modeling using Exponential Splines. *Journal of Finance* (37), 367-378.
- Waggoner, D. F. (1997). *Spline Methods for Extracting Interest Rate Curves from Coupon Bond Prices*. Working Paper, Federal Reserve Bank of Atlanta.
- Anderson, N., & Sleath, J. (2001). *New Estimates of the UK Real and Nominal Yield Curves*. Working Paper, Bank of England.
- Arango, L. E., Melo, L. F., & Vazquez, D. M. (n.d.). *Estimación de la Estructura a Plazo de las Tasas de Interés en Colombia*. Retrieved 2 de Noviembre de 2010 from Banco de la República, Colombia: <http://www.banrep.gov.co/docum/ftp/borra196.pdf>
- Bagley, J. D. (1967). *The Behavior of adaptive systems which employ genetic and correlation algorithms*. Ph. D. Dissertation University Microfilms, University of Michigan, Ann Arbor.
- Barret, W. R., Gosnell, T. F., Jr, T. F., & Heuson, A. J. (1995). Yield Curve Shifts and the Selection of Immunization Strategies. *Journal of Fixed Income*, 5 (2), 53-64.
- Biggs, M. C. (1975). Constrained Minimization Using Recursive Quadratic Programming. In L. D. Szegö (Ed.), *Towards Global Optimization* (pp. 341-349). North Holland.
- Cairns, A. J., & Pritchard, D. J. (2001). Stability of Descriptive Models for the Term Structure of Interest Rates with Applications to German Market Data. *British Actuarial Journal*, 7, 467-507.
- Cox, J. C., Ingersoll, J. E., & Ross, S. A. (1985). "A Theory of the Term Structure of Interest Rates. *Econometrica*, 53 (2), 385-407.
- EO Evolving Objects*. (n.d.). From <http://eodev.sourceforge.net/>
- Dwi Prastyo, D., Suhartono, & Sofyan, H. (2010). Implementation of LM Algorithm and SQP for Yield Curve Optimization using Nelson-Siegel-Svensson Model. *Korean Statistical Society Spring Conferece*. Daejeon.
- Duffie, D., & Kan, R. (1996). A Yield-Factor Model of Interest Rates. *Mathematical Finance*, 6, 379-406.

- de Pooter, M. (2007). *Examining the Nelson-Siegel Class of Term Structure Models*. Working Paper, Erasmus University Rotterdam, Econometric Institute and Tinbergen Institute.
- Department of Automatic Control and Systems Engineering of The University of Sheffield. (22 de Septiembre de 1994). *Genetic Algorithm Toolbox*. From The University of Sheffield: <http://www.shef.ac.uk/acse/research/ecrg/gat>
- Diebold, F. X., & Li, C. (2006). Forecasting the Term Structure of Government Bond Yields. *Journal of Econometrics*, 130, 337-364.
- Fabozzi, F. J., Martellini, L., & Priaulet, P. (2005). Predictability in the Shape of the Term Structure of Interest Rates. *Journal of Fixed Income*, 14 (1), 40-53.
- Fisher, M., Nychka, D., & Zervos, D. (1995). Fitting the Term Structure of Interest rates with Smoothing Splines. (B. o. (U.S.), Ed.) *Finance and Economic Discussion Series* (95-1).
- GEATbx. (n.d.). *Genetic and Evolutionary Algorithm Toolbox*. From <http://www.geatbx.com/>
- Gill, P. E., Murray, W., & Wright, M. H. (1991). *Numerical Linear Algebra and Optimization*. Addison Wesley.
- Gill, P. E., Murray, W., Saunders, M. A., & Wright, M. H. (1984). Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints. *ACM Trans. Math. Software*, 10, 282-298.
- Gilli, M., Grobe, S., & Shumann, E. (2010). *Calibrating the Nelson-Siegel-Svesson Model*. Working Paper, COMISEF.
- Gimeno, R., & Nave, J. M. (2006). *A Genetic Algorithm Estimation of the Term Structure of Interest Rates*. Working Paper, Banco de España.
- Isasi, P. (21 de Marzo de 2011). *Estrategias Evolutivas*. Retrieved 25 de Mayo de 2011 from OpenCourseWare de la Universidad Carlos III de Madrid: <http://ocw.uc3m.es/informatica/computacion-biologica/material-de-clase-1/EEvolutivasT.pdf>
- habitissimo*. (n.d.). Retrieved 2011 de Septiembre de 2011 from <http://www.habitissimo.es/>
- Hans, S. (1977). A Globally Convergent Method for Nonlinear Programming. In *J. Optimization Theory and Applications* (Vol. 22, p. 297).
- Hansen, N. (2006). The CMA Evolution Strategy: A Comparing Review. In J. A. Lozano, P. Larrañaga, I. Inza, & E. Bengoetxea (Eds.), *Towards a New Evolutionary Computation Advances in the Estimation of Distribution Algorithms* (pp. 75-102). Springer Berlin / Heidelberg.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.
- Karmarkar, N. (1984). A New Polynomial Time Algorithm for Linear Programming. *Combinatoria*, 4 (4), 373-395.

- Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm Intelligence*. San Francisco: Morgan Kaufmann.
- Nelson, C. R., & Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. *The Journal of Business*, 60 (4), 473-489.
- Mathworks. (n.d.). *fmincon Active Set Algorithm*. Retrieved 30 de Septiembre de 2011 from <http://www.mathworks.es/help/toolbox/optim/ug/brnoxzl.html#brnox01>
- Mathworks. (n.d.). *fmincon Interior Point Algorithm*. Retrieved 2 de Septiembre de 2011 from <http://www.mathworks.es/help/toolbox/optim/ug/brnoxzl.html#brnpd5f>
- Mathworks. (n.d.). *fmincon SQP Algorithm*. Retrieved 30 de Septiembre de 2011 from <http://www.mathworks.es/help/toolbox/optim/ug/brnoxzl.html#bsgppl4>
- Mathworks. (n.d.). *Global Optimization Toolbox*. From <http://www.mathworks.es/products/global-optimization/>
- McCulloch, J. H. (1971). Measuring the Term Structure of Interest Rates. *The Journal of Business* (44), 19-31.
- McCulloch, J. H. (1975). The Tax-adjusted Yield Curve. *The Journal of Finance* (3).
- Meier, I. (1999). *Estimating the Term Structure of Interest Rates: The Swiss Case*. Working Paper, Swiss National Bank, Study Center Gerzensee.
- Pereda, J. (2009). *Estimación de la Curva de Rendimiento Cupón Cero para el Perú: Aspectos Metodológicos y Aplicaciones*. Banco Central de Reserva del Perú.
- Powell, M. (1978). A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In G.A.Watson (Ed.), *Numerical Analysis* (Vol. 630). Springer Verlag.
- Powell, M. The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations. In *Nonlinear Programming 3* (p. 1978). O.L. Mangasarian, R.R. Meyer and S.M. Robinson.
- Pohlheim, H. (Diciembre de 2006). *GEATbx Tutorial*. Retrieved 21 de Mayo de 2011 from GEATbx: [http://www.geatbx.com/download/GEATbx\\_Tutorial\\_v38.pdf](http://www.geatbx.com/download/GEATbx_Tutorial_v38.pdf)
- Pohlheim, H. (Diciembre de 2006). *GEATbx: Introduction*. Retrieved 20 de Mayo de 2011 from GEATbx: [http://www.geatbx.com/download/GEATbx\\_Intro\\_Algorithmen\\_v38.pdf](http://www.geatbx.com/download/GEATbx_Intro_Algorithmen_v38.pdf)
- Pohlheim, H. (Diciembre de 2006). *GEATbx: Options*. Retrieved 20 de Mayo de 2011 from GEATbx: [http://www.geatbx.com/download/GEATbx\\_Options\\_v38.pdf](http://www.geatbx.com/download/GEATbx_Options_v38.pdf)
- Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). A Derandomized Approach to Self-Adaptation of Evolution Strategies. *Evolutionary Computation*, 2 (4), 369-380.
- Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). Step-size adaptation based on non-local use of selection information. *Parallel Problem Solving from Nature PPSN IV*. 866, pp. 189-198. Jerusalem: Springer.
- Ruiz Dotras, E. (12 de Diciembre de 2005). *Comparación de Curvas de Tipos de Interés. Efectos de la Integración Financiera*. Retrieved 5 de Noviembre de 2010 from Tesis Doctorales en Red: <http://tdx.cat/handle/10803/2132>

Rechenberg, I. (1973). *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Fromman-Holzboog.

Svensson, L. E. (1994). *Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994*. Working Paper, National Bureau of Economic Research, Cambridge.

Shea, G. S. (1984). Pitfalls in Smoothing Interest Rate Term Structure Data: Equilibrium Models and Spline Approximations. *Journal of Financial and Quantitative Analysis*, 19 (3), 253-269.

Steeley, J. M. (1991). Estimating the Gilt-edged Term Structure: Basis Splines and Confidence Intervals. *Journal of Business Finance and Accounting*, 18 (4), 513-529.

Storn, R., & Price, K. (1997). Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, 341-359.