

UNIVERSIDAD DE GRANADA
E.T.S. DE INGENIERÍA INFORMÁTICA



Departamento de Ciencias de la
Computación e Inteligencia Artificial

Bioinformática

<http://sci2s.ugr.es/docencia/bioinformatica>

Guión de Prácticas

Práctica 3:
Algoritmos Genéticos Multi-Objetivo

Curso 2011-2012

Quinto Curso de Ingeniero en Informática

Práctica 3

Algoritmos Genéticos Multiobjetivo

3.1 Objetivos

El objetivo de esta práctica es estudiar el comportamiento de varios *Algoritmos Genéticos Multiobjetivo (AGMOs)*. Para ello, se requiere que el alumno use varios algoritmos de este tipo resolviendo distintos problemas de optimización continua de naturaleza multiobjetivo. Concretamente se considerarán dos algoritmos de última generación descritos en las transparencias de la asignatura, ambos con enfoque elitista: SPEA2 y NSGA-II. Con objeto de estudiar la influencia del operador de cruce se implementarán dos variantes de cada algoritmo: una versión utilizando el operador de cruce BLX- α y la otra utilizando el operador de cruce SBX (propio de NSGA-II y usado por SPEA2).

Dichos algoritmos se aplicarán sobre un conjunto de 5 funciones bien conocidas y estudiadas (todas ellas con dos objetivos): ZDT1, ZDT2, ZDT3, ZDT4 y ZDT6. Por lo tanto, la práctica consistirá en estudiar y comparar los conjuntos de Pareto (conjunto de soluciones no dominadas) obtenidos por los algoritmos indicados en las distintas funciones.

La práctica se califica sobre 3 puntos. Esta puntuación se distribuye de la siguiente forma: **SPEA2 1 punto, NSGA-II 1 punto y análisis de resultados 1 punto** (la puntuación del análisis de resultados bajará proporcionalmente si sólo se estudia un algoritmo o no se usan todas sus variantes). **La fecha tope de entrega es el Viernes 15 de Junio de 2012.**

3.2 Optimización de Funciones Multiobjetivo

En esta sección, se presentan los cinco problemas de prueba (funciones ZDT) que se han de utilizar para analizar el comportamiento de SPEA2 y NSGA-II. Cada problema consiste en una función multiobjetivo que se pretende optimizar con la ayuda de los AGMOs.

Dado un vector de valores reales $X = (x_1, x_2, \dots, x_n)$ y siendo n el número de variables de la función a optimizar, el problema consiste en encontrar los valores de X que optimizan los distintos objetivos de dicha función. Por lo tanto el número de variables de la función, determina el tamaño (número de genes) de un cromosoma.

El conjunto de estas cinco funciones se presenta en los siguientes apartados. Fueron propuestas por primera vez para la comparación de AGMOs en:

E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, 8(2), 2000, 173-195.

Dichas funciones consideran dos objetivos que hay que minimizar y cada una de ellas está estructurada de la misma forma:

$$\begin{array}{l} \text{Minimizar } f_1(x_1), \\ \text{Minimizar } f_2(X) = g(x_2, \dots, x_n) \cdot h(f_1(x_1), g(x_2, \dots, x_n)), \\ \text{siendo } X = (x_1, \dots, x_n). \end{array}$$

La diferencia entre las distintas funciones ZDT es cómo se definen las funciones f_1 , g y h , el número de variables n y el rango de valores válidos para cada una de ellas. **El código fuente de estas funciones se encuentra en el fichero *problemdef.c* del código fuente de NSGA-II disponible en la página *web* de la asignatura.**

3.2.1 Función ZDT1

Este problema presenta un frente de Pareto óptimo convexo. Las funciones que lo definen son:

$$\begin{aligned} f_1 &= f_1(x_1) = x_1, \\ g &= g(x_2, \dots, x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^n x_i}{(n-1)}, \\ h(f_1, g) &= 1 - \sqrt{f_1/g}, \\ \text{con } n &= 30 \quad \text{y} \quad x_i \in [0, 1]. \end{aligned}$$

El frente de Pareto óptimo se obtiene con $g=1$ para los distintos valores de f_1 . La figura 1 representa dicho frente.

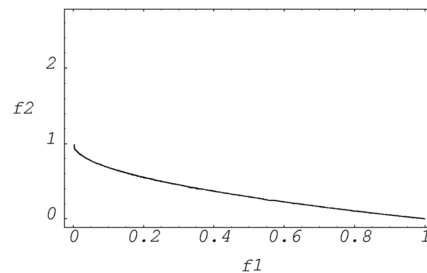


Figura 1. Frente de Pareto óptimo de ZDT1

3.2.2 Función ZDT2

Este problema presenta un frente de Pareto óptimo cóncavo y contrapuesto al de ZDT1. Las funciones que lo definen son:

$$\begin{aligned}
 f_1 &= f_1(x_1) = x_1, \\
 g &= g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n x_i / (n-1), \\
 h(f_1, g) &= 1 - \left(f_1 / g \right)^2, \\
 \text{con } n &= 30 \quad \text{y} \quad x_i \in [0, 1].
 \end{aligned}$$

El frente de Pareto óptimo se obtiene con $g=1$ para los distintos valores de f_1 . La figura 2 representa dicho frente.

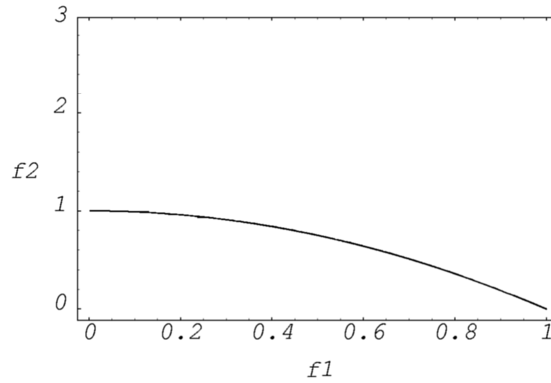


Figura 2. Frente de Pareto óptimo de ZDT2

3.2.3 Función ZDT3

En este problema el frente de Pareto óptimo es discontinuo y está compuesto por varias zonas convexas. Las funciones que lo definen son:

$$\begin{aligned}
 f_1 &= f_1(x_1) = x_1, \\
 g &= g(x_2, \dots, x_n) = 1 + 9 \cdot \sum_{i=2}^n x_i / (n-1), \\
 h(f_1, g) &= 1 - \sqrt{f_1 / g} - \left(f_1 / g \right) \cdot \text{sen}(10 \cdot \pi \cdot f_1), \\
 \text{con } n &= 30 \quad \text{y} \quad x_i \in [0, 1].
 \end{aligned}$$

El frente de Pareto óptimo se obtiene con $g=1$ para los distintos valores de f_1 . El uso de la función seno en h provoca discontinuidad en dicho frente. Sin embargo, no la hay en el dominio de los parámetros. La figura 3 representa dicho frente.

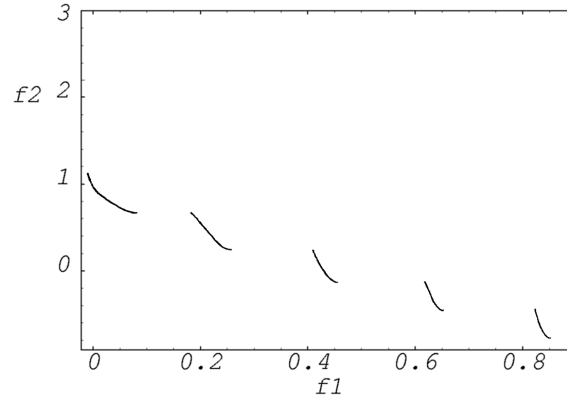


Figura 3. Frente de Pareto óptimo de ZDT3

3.2.4 Función ZDT4

Este problema tiene 21⁹ frentes locales de Pareto óptimos, y por lo tanto, sirve para probar la habilidad de los algoritmos al resolver problemas multimodales. Las funciones que lo definen son:

$$f_1 = f_1(x_1) = x_1,$$

$$g = g(x_2, \dots, x_n) = 1 + 10 \cdot (n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cdot \cos(4 \cdot \pi \cdot x_i)),$$

$$h(f_1, g) = 1 - \sqrt{f_1 / g},$$

$$\text{con } n = 10, \quad x_1 \in [0, 1] \quad \text{y} \quad x_2, \dots, x_n \in [-5, 5].$$

El frente de Pareto óptimo se obtiene con $g=1,25$ para los distintos valores de f_1 . La figura 4 representa dicho frente. Véase como los frentes locales de Pareto óptimos no son distinguibles en dicha figura.

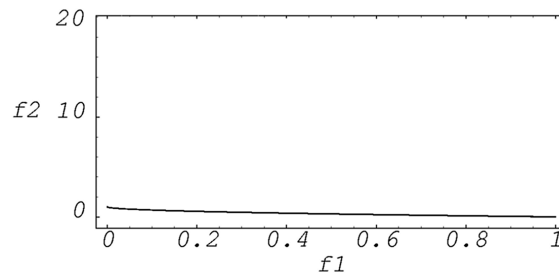


Figura 4. Frente de Pareto óptimo de ZDT4

3.2.5 Función ZDT6

Este problema presenta dos dificultades causadas por la falta de uniformidad del espacio de búsqueda: las soluciones Pareto óptimas se distribuyen de forma no uniforme a lo largo del frente de Pareto global (el frente está sesgado por soluciones para las que f_1 está cerca de 1) y la densidad de las soluciones es menor cuanto más cerca se está del frente de Pareto óptimo y mayor cuanto más lejos. Las funciones que definen el problema son,

$$f_1 = f_1(x_1) = 1 - \exp(-4 \cdot x_1) \cdot \sin^6(6 \cdot \pi \cdot x_1),$$

$$g = g(x_2, \dots, x_n) = 1 + 9 \cdot \left(\left(\sum_{i=2}^n x_i \right) / (n-1) \right)^{0.25},$$

$$h(f_1, g) = 1 - \left(f_1 / g \right)^2,$$

con $n = 10$ y $x_i \in [0, 1]$.

El frente de Pareto óptimo se obtiene con $g=1$ para los distintos valores de f_1 y es cóncavo. La figura 5 representa dicho frente.

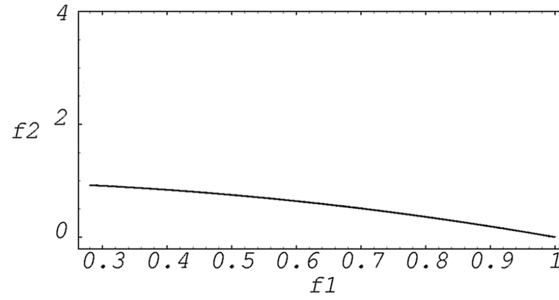


Figura 5. Frente de Pareto óptimo de ZDT6

3.3 Representación

El cromosoma estará formado por un conjunto de n genes (siendo n el número de variables de la función que se esté resolviendo). Cada gen de un cromosoma será un número real que tomará un valor dentro del intervalo de variación de la variable asociada, y que dependerá de la función con la que estemos trabajando. De esta manera un cromosoma será un vector con la siguiente forma:

$$Crom = [x_1, x_2, \dots, x_n].$$

3.4 Función Objetivo

Para evaluar la calidad de un cromosoma se calcularán el valor de cada objetivo de la función que se esté optimizando mediante las fórmulas que la definen, proporcionadas en las secciones 3.2.1 a 3.2.5. De esta forma, la evaluación de un cromosoma dará como resultado el siguiente vector de objetivos (que en nuestro caso hay que minimizar):

$$F(Crom) = [O_1, O_2].$$

3.5 Operadores Genéticos

Se considerarán dos variantes para cada algoritmo, cada una de ellas utilizando operadores distintos. A continuación se describen los operadores para cada una de las dos variantes:

Variante 1. Operador de Cruce *BLX- α* y mutación aleatoria:

Dados dos padres $C_1 = (c_1^1, \dots, c_i^1, \dots, c_n^1)$ y $C_2 = (c_1^2, \dots, c_i^2, \dots, c_n^2)$, el operador *BLX- α* genera un hijo $H = (h_1, \dots, h_i, \dots, h_n)$ donde h_i es un número escogido de forma aleatoria (con una distribución uniforme) en el intervalo $[c_{min} - I\alpha, c_{max} + I\alpha]$, donde $c_{max} = \max(c_i^1, c_i^2)$, $c_{min} = \min(c_i^1, c_i^2)$ e $I = c_{max} - c_{min}$.

Puesto que h_i podría salirse del intervalo de variación del gen i , si h_i se sale por arriba habrá que ajustarlo al máximo valor posible para dicho gen y si h_i se sale por abajo habrá que ajustarlo al mínimo valor posible. Este operador se aplicará dos veces para cada par de padres usando $\alpha = 0.5$.

El operador de mutación aleatoria consiste en sustituir el valor de un gen por un valor seleccionado aleatoriamente de su conjunto de valores válidos. Se aplica a cada uno de los genes reales de los cromosomas con probabilidad P_m .

Variante 2. Operador de Cruce *SBX-15* y mutación polinomial:

Dados dos padres $C_1 = (c_1^1, \dots, c_i^1, \dots, c_n^1)$ y $C_2 = (c_1^2, \dots, c_i^2, \dots, c_n^2)$, el operador *SBX* genera dos hijos:

$$H_k = (h_1^k, \dots, h_i^k, \dots, h_n^k), \quad k = 1, 2, \text{ donde:}$$

$$h_i^1 = \frac{1}{2} \cdot [(1 - \beta_k) \cdot c_i^1 + (1 + \beta_k) \cdot c_i^2] \text{ y}$$

$$h_i^2 = \frac{1}{2} \cdot [(1 + \beta_k) \cdot c_i^1 + (1 - \beta_k) \cdot c_i^2]$$

donde $\beta_k (\geq 0)$ es una muestra del generador de números aleatorios con densidad:

$$p(\beta) = \begin{cases} \frac{1}{2} \cdot (\eta + 1) \beta^\eta, & \text{si } 0 \leq \beta \leq 1 \\ \frac{1}{2} \cdot (\eta + 1) \frac{1}{\beta^{\eta+2}}, & \text{si } \beta > 1. \end{cases}$$

Esta distribución puede obtenerse fácilmente a partir de una uniforme $u(0, 1)$, usando la transformación (con valor $\eta = 15$ en este caso):

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{si } u(0,1) \leq \frac{1}{2} \\ [2(1-u)]^{-\frac{1}{\eta+1}} & \text{si } u(0,1) > \frac{1}{2}. \end{cases}$$

El operador de mutación polinomial se basa en el mismo concepto que el cruce SBX aplicando una adaptación de las fórmulas sobre un único gen. **Dada la complejidad de ambos operadores, dichos operadores se dan implementados en los archivos *crossover.c* y *mutation.c* del código fuente de NSGA-II disponible en la página web de la asignatura (funciones *realcross* y *real_mutate_ind*). Así mismo, también están incluidos en el código fuente de SPEA2. Sólo hay que implementar los códigos del cruce BLX- α y mutación aleatoria descritos en la sección anterior para sustituir por éstos en la versión correspondiente de cada algoritmo.**

3.6 Algoritmos Evolutivos y Parámetros Considerados

Se utilizarán/adaptarán los siguientes algoritmos evolutivos:

- **SPEA2.** Se basa en el algoritmo SPEA descrito en las transparencias de teoría y los operadores genéticos descritos anteriormente. Se ha puesto a disposición de los alumnos el código del algoritmo SPEA2 y un ejemplo de ejecución sobre la función ZDT1 con una semilla. El código está dividido en dos algoritmos que se ejecutan en paralelo y utiliza el operador de cruce SBX y la mutación polinomial. El primer algoritmo (*dtlz*), contiene los operadores genéticos y los códigos de las funciones objetivo y el segundo (*spea2*), implementa el esquema de selección propio de SPEA2. Ambos algoritmos deben ejecutarse a la vez y comparten información a través de archivos (en el caso del ejemplo dejado en la web, hay que ejecutar *run_dtlz.bat* y acto seguido *run_spea2.bat*, y devuelve los resultados en el directorio principal).

El alumno debe realizar las siguientes tareas:

1. Estudiar y comprender cómo está implementado y cómo se ejecuta el algoritmo. Para ello es imprescindible leer detenidamente los archivos *spea2_documentation.txt* y *dtlz_documentation.txt* (en la web está disponible el artículo original de SPEA2). Ejecutarlo para cada una de las funciones con 10 semillas distintas.
2. Modificar el código para usar el operador de cruce BLX en sustitución de SBX y la mutación aleatoria en sustitución de la polinomial.
3. **Ejecutar cada una de las dos variantes del código 10 veces, considerando 10 semillas distintas, para cada una de las funciones.**

IMPORTANTE: puesto que el código de SPEA2 no está preparado para trabajar con valores negativos en los objetivos, para ZDT3 se ha sumado la constante 1 a la función $f_2(X)$. **Antes de utilizar los resultados para analizar el algoritmo en ZDT3, se debe restar 1 a los valores obtenidos en este objetivo ($f_2(X)=f_2(X)-1$).**

- **NSGA-II.** Usará todas las características descritas en las transparencias de teoría y los operadores genéticos descritos anteriormente para sus dos variantes. Se ha puesto a disposición de los alumnos el código del algoritmo NSGA-II y un

ejemplo de ejecución sobre la función ZDT1 con una semilla. El código utiliza el operador de cruce SBX y la mutación polinomial. En el caso del ejemplo dejado en la *web* hay que ejecutar *ejecuta.bat*, que devuelve los resultados en el directorio principal (además se incluyen los ficheros de parámetros para todas las funciones). Las funciones están implementadas en el fichero *problemdef.c*, siendo necesario recompilar el código cada vez que se quiera cambiar de función. Se recomienda modificar el código para poder utilizar cualquier función indicándolo como parámetro de entrada al algoritmo.

El alumno debe realizar las siguientes tareas:

1. Estudiar y comprender cómo está implementado y cómo se ejecuta el algoritmo. Para ello es imprescindible leer detenidamente los ficheros *Readme.txt* (en la web está disponible el artículo original de NSGA-II). Ejecutarlo para cada una de las funciones con 10 semillas distintas.
2. Modificar el código para usar el operador de cruce BLX en sustitución de SBX y la mutación aleatoria en sustitución de la polinomial.
3. **Ejecutar cada una de las dos variantes del código 10 veces, considerando 10 semillas distintas, para cada una de las funciones.**

En cuanto al formato de los ficheros de salida de SPEA2 y NSGA-II, se representa una solución por cada línea del fichero de salida. En cada línea se encuentran primero los valores obtenidos por los objetivos (f_1 y f_2) y a continuación los valores aprendidos para cada variable que dan lugar a dicho resultado (x_1, \dots, x_n). En el caso de SPEA2, antes de los objetivos se encuentra el identificador del individuo dentro de la población (ID, $f_1, f_2, x_1, \dots, x_n$).

Los parámetros que serán considerados para las ejecuciones se muestran en las siguientes tablas:

PARÁMETRO	VALORES
Probabilidad de Mutación por gen	1/n
Probabilidad de Cruce	1.0
Tamaño de la Población	100
Máximo de Evaluaciones	10000
Nº de Ejecuciones (semillas)	10

Parámetros Generales

PARÁMETRO	VALORES
TAMAÑO POBLACIÓN EXTERNA	25

Parámetros específicos de SPEA2

Los parámetros necesarios para el operador SBX, la mutación polinomial y los parámetros mostrados en las tablas están fijados en los ficheros de parámetros incluidos en los ejemplos de ejecución de SPEA2 y NSGA-II con ZDT1 disponibles en la página *web* de la asignatura (*NO hace falta modificarlos para la versión que usa SBX y mutación polinomial, sólo cambiar las semillas e ir variando la función*).

El total de la práctica consiste en **200 ejecuciones**. Este número resulta de combinar:

- Dos algoritmos: SPEA2 y NSGA-II.
- Dos variantes: BLX-aleatoria y SBX-polinomial.
- Cinco funciones: ZDT1, ZDT2, ZDT3, ZDT4 y ZDT6.
- **Diez ejecuciones con diez semillas distintas por función y variante del algoritmo (se utilizarán las mismas 10 semillas para todas las ejecuciones, aunque en el caso de NSGA-II serán distintas a las usadas en SPEA2 ya que recibe un número real como semilla).**

3.7 Resultados

Hay que obtener varias tablas de resultados con información sobre las soluciones encontradas por los algoritmos ejecutados. Para cada método (SPEA2 y NSGA-II), se creará una tabla con la misma estructura que la Tabla 3.1 comparando las dos variantes de cada método (una tabla por método). A la vista de los resultados de cada tabla se debe seleccionar la mejor versión para cada AGMO, justificando la elección.

La métrica C (aplicada en el espacio de los objetivos) está explicada en teoría y su fórmula se encuentra en las transparencias de la asignatura. La métrica C sirve como técnica de comparación entre algoritmos ya que permite comparar un conjunto de Pareto contra otro.

AGMO	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
Ejecución	C ₁₂	C ₂₁	C ₁₂	C ₂₁	C ₁₂	C ₂₁	C ₁₂	C ₂₁	C ₁₂	C ₂₁
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
Media										
Desv. Típica										

$$C_{12}=C(\text{AGMO-v1}, \text{AGMO-v2}) \text{ y } C_{21}=C(\text{AGMO-v2}, \text{AGMO-v1})$$

Tabla 3.1 Comparativa entre variantes.

Una vez seleccionada la mejor versión de cada AGMO, se creará una tabla con la misma estructura que la Tabla 3.2 para cada problema.

Ejecución	$C(\text{SPEA2-}v_i, \text{NSGA-II-}v_j)$	$C(\text{NSGA-II-}v_j, \text{SPEA2-}v_i)$
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
Media		
Desviación Típica		

Tabla 3.2 Comparativa entre la mejor variante de SPEA2 y NSGA-II sobre la función Y

También se creará una tabla global con los resultados medios de C para la variante seleccionada de cada AGMO en cada una de las funciones y con un valor promedio global sobre todas ellas. La estructura de dicha tabla es la siguiente (tabla 3.3):

Comparativa	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Media
Media $C(\text{SPEA2-}v_i, \text{NSGA-II-}v_j)$						
Desv. Típica $C(\text{SPEA2-}v_i, \text{NSGA-II-}v_j)$						----
Media $C(\text{NSGA-II-}v_j, \text{SPEA2-}v_i)$						
Desv. Típica $C(\text{NSGA-II-}v_j, \text{SPEA2-}v_i)$						----

Tabla 3.3 Comparativa global entre la mejor variante de NSGA-II y SPEA2

Finalmente, se creará una gráfica bidimensional por problema en la que se mostrará de manera conjunta el **conjunto de Pareto agregado** obtenido por cada AGMO en dicho problema. Para obtener el conjunto de Pareto agregado se fusionaran los diez conjuntos de Pareto obtenidos por el algoritmo en sus diez ejecuciones y se eliminarán las soluciones dominadas.

3.8 Documentación e Implementación a Entregar

La documentación de la práctica deberá incluir, al menos, el siguiente contenido:

1. Portada con el título de la práctica, curso académico, nombre y dirección e-mail del alumno, y grupo, horario y nombre del profesor de prácticas.
2. Índice del contenido de la documentación con la numeración de las páginas.
3. **Breve** descripción/formulación del problema (**máximo 2 páginas**): ¿en qué consiste, para qué sirve?
4. Breve descripción de la aplicación de los algoritmos evolutivos al problema (**máximo 2 páginas**): representación, función objetivo, operadores, etc.

5. Descripción en **pseudo-código** de todas aquellas operaciones relevantes de cada algoritmo: esquema de evolución seguido, operadores genéticos, etc. La descripción de cada algoritmo no podrá ocupar más de **3 páginas**.
6. Experimentos y análisis de resultados.
 - Valores de los parámetros considerados en las ejecuciones de cada algoritmo (**incluyendo las semillas utilizadas**).
 - Resultados obtenidos según el formato especificado.
 - Análisis de resultados. El análisis deberá estar orientado a **justificar** (según el comportamiento de cada algoritmo) **los resultados** obtenidos en lugar de realizar una mera “lectura” de las tablas. Se valorará la inclusión de otros elementos de comparación tales como gráficas de convergencia, análisis comparativo de las soluciones obtenidas, etc.
7. Referencias bibliográficas, si procede.
8. Un pequeño manual de usuario describiendo cómo ejecutar el software.

Para mayor claridad de la documentación, se recomienda comenzar cada sección o descripción en una página nueva. Superar el máximo de páginas establecido para cada sección/descripción o emplear un formato ilegible podrá influir negativamente (hasta en un 50 %) en la evaluación final de la práctica. En caso de que el comportamiento del algoritmo en la versión implementada no coincida con la descripción en pseudo-código, se podría reducir hasta en un 50% la calificación del algoritmo correspondiente.

3.9 Componentes de la práctica

- **Documentación:** La documentación será impresa e incluirá como mínimo las secciones indicadas en el apartado anterior. Aunque lo esencial es el contenido, también debe cuidarse la presentación y la redacción. No se requiere listado del código fuente.

El documento llevará una portada en la que se incluirá el título de la práctica, el curso académico actual, el nombre y DNI del alumno, su correo electrónico, el grupo de prácticas, su horario y el nombre del profesor de prácticas correspondiente.

- **Implementación:** Se entregará un disco que contenga el código fuente y una versión ejecutable de los programas. Los fuentes se organizarán en la estructura de directorios que sea necesaria y que deberá colgar del directorio FUENTES en el raíz. Junto con los fuentes hay que incluir los ficheros necesarios para construir los ejecutables, (por ejemplo, los ficheros proyecto *.prj o el fichero makefile). La versión ejecutable de los programas junto con los ficheros necesarios para ejecutarlos en las distintas instancias se incluirán en un subdirectorío del raíz de nombre BIN. En este mismo directorio se adjuntará un pequeño fichero de texto de nombre LEEME que contendrá breves reseñas sobre cada fichero incluido en el directorio, es decir, indicará a qué instancia corresponde cada fichero de entrada y a qué programa corresponde cada fichero ejecutable. El CD a entregar deberá incluir en ficheros ASCII las soluciones obtenidas por cada algoritmo en cada ejecución.

Adicionalmente podrán incluirse ficheros de configuración o inicialización de cualquiera de los algoritmos considerados para la práctica. Es importante que los programas realizados puedan leer los valores de los parámetros de los algoritmos desde fichero, es decir, que no tengan que ser recompilados para cambiar éstos ante una nueva ejecución. Por ejemplo, la semilla que inicializa la secuencia pseudoaleatoria debería poder especificarse como un parámetro más.

La práctica se entregará en dos formatos distintos:

- En formato **electrónico** a través de la web del departamento CCIA. Se subirá un fichero Zip que incluya un pdf con la documentación y los ficheros mencionados.
- En formato **impreso** entregado en mano al profesor/a de prácticas. Se le proporcionará una copia impresa de la documentación y un CD-ROM con los ficheros. Los discos irán etiquetados con la misma información descrita para la portada de la documentación y deberán estar pegados de algún modo a la documentación, no sueltos.