Experimentación interactiva con algoritmos voraces

J. Ángel Velázquez Iturbide, Ouafae Debdi
Departamento de Lenguajes y Sistemas Informáticos I
Universidad Rey Juan Carlos
C/ Tulipán s/n
28933 Móstoles, Madrid
{angel.velazquez, ouafae.debdi}@urjc.es

Resumen

Aunque pueden establecerse algunos objetivos de aprendizaje comunes a todas las técnicas de diseño de algoritmos, cada técnica particular tiene peculiaridades que obligan a variar ligeramente sus objetivos educativos. Uno de los elementos clave de los algoritmos voraces es una función de selección de candidatos que garantiza un resultado óptimo. Presentamos una colección de ayudantes interactivos diseñados para ayudar al alumno a identificar funciones de selección óptimas para problemas concretos. El proceso de identificación es un experimento (al estilo de las ciencias experimentales), en el que el alumno prueba de forma planificada posibles funciones de selección y decide cuáles son óptimas. La experimentación se realiza con ayuda de varios ayudantes interactivos, que hemos desarrollado para los siguientes problemas: mochila, selección de actividades y árbol de recubrimiento de coste mínimo. Los ayudantes interactivos se han utilizado durante los cursos académicos 2007-2008, 2008-2009 y 2009-2010, con resultados positivos.

1. Introducción

El trabajo aquí presentado es una contribución dentro de una línea de investigación sobre el desarrollo de aplicaciones educativas de animación de programas para técnicas de diseño de algoritmos. Dicha línea toma la taxonomía de Bloom [1] como marco para la especificación de aplicaciones educativas [5]. Con este enfoque, hemos desarrollado sistemas para la animación de la recursividad [12] y de algoritmos de divide-y-vencerás [13].

En esta comunicación de recurso docente se presentan tres ayudantes interactivos (es decir, aplicaciones educativas interactivas), diseñados para ayudar a los alumnos en el estudio de los algoritmos voraces. AMO está concebido para ayudar en el estudio del problema de la mochila, SEDA para el problema de la selección de actividades y TuMiST para el problema del árbol de recubrimiento de coste mínimo (resoluble por los conocidos algoritmos de Prim y Kruskal).

La estructura de la comunicación es la siguiente. En la sección siguiente incluimos un método experimental para el aprendizaje activo de los algoritmos voraces, así como una descripción de los ayudantes interactivos desarrollados para soportarlo por ordenador. La sección tercera reproduce una hipotética sesión de usuario. La cuarta sección comenta brevemente nuestra experiencia de uso y los esfuerzos realizados para facilitar su difusión y fomentar su uso docente. Por último, incluimos algunas líneas de trabajo actual y futuro.

2. Experimentación interactiva con algoritmos voraces

Organizamos esta sección en dos partes. Primero presentamos un método experimental para el aprendizaje activo de los algoritmos voraces y después, los tres ayudantes interactivos desarrollados para soportarlo por ordenador.

2.1. Aprendizaje activo de algoritmos voraces

Hemos comentado que usamos la taxonomía de Bloom [1] para especificar los objetivos educativos de nuestras aplicaciones. Se trata de un marco, muy conocido, para medir el grado de conocimiento de una materia por el alumno. En orden creciente de dificultad, pueden alcanzarse los niveles de: conocimiento, comprensión, aplicación, análisis, síntesis y evaluación.

Los objetivos de aprendizaje de los algoritmos voraces pueden alcanzarse de diversas formas. Hemos diseñado un método experimental para su aprendizaje activo. Lo presentamos con un ejemplo; puede encontrarse más información sobre el método en [11].

Sea el problema de la selección de actividades [2]. Dado un conjunto de n actividades, cada una con un instante de inicio c_i y un instante de fin f_i, se desea seleccionar un subconjunto de tamaño máximo de actividades que no se solapen.

Por ejemplo, dado el conjunto de actividades de la Tabla 1, el subconjunto formado por las actividades {5,0,8} es una solución válida, pero el subconjunto {0,3,4,5} es una solución de tamaño máximo.

i	0	1	2	3	4	5	6	7	8	9
s_i	13	12	10	24	18	0	9	22	18	10
f_i	17	23	23	29	23	10	18	30	25	19

Tabla 1. Un ejemplo del problema de selección de actividades

Podemos concebir distintas funciones de selección de las actividades para la solución de este problema mediante un algoritmo voraz:

- Orden creciente/decreciente de índice.
- Orden creciente/decreciente de inicio.
- Orden creciente/decreciente de fin.
- Orden creciente/decreciente de duración.

Si probamos a resolver el problema con estas funciones de selección, se obtienen los siguientes resultados:

Función de selección	Actividades seleccionadas	Nº actividades
Índice, crec.	{0, 3, 4, 5}	4
Índice, decr.	{9, 7, 5}	3
Inicio, crec.	{5, 9, 7}	3
Inicio, decr.	{3, 4, 0, 5}	4
Fin, crec.	{5, 0, 4, 3}	4
Fin, decr.	{7, 9, 5}	3
Durac., crec.	{0, 4, 3, 5}	4
Durac., decr.	{2, 5, 3}	3

Tabla 2. Resultado de aplicar distintas funciones de selección

Esta ejecución nos permite descartar cuatro funciones de selección que no han producido un resultado óptimo, quedando todavía otras cuatro funciones candidatas: orden creciente de índice, orden decreciente de inicio, orden creciente de fin, y orden creciente de duración. (Parece obvio que la primera función de selección no es una candidata seria, pero tenemos que descartarla experimentalmente.)

Si realizamos más ejecuciones, podemos ir descartando más funciones de selección:

Función de selección	1ª ejec.	2ª ejec.	3ª ejec.	4ª ejec.	5ª ejec.
Índice, creciente	4	2	2	–	–
Índice, decrec.	3	–	–	–	–
Inicio, creciente	3	–	–	–	–
Inicio, decrec.	4	2	5	4	3
Fin, creciente	4	2	5	4	3
Fin, decrec.	3	–	–	–	–
Durac., creciente	4	2	5	4	3
Durac., decrec.	3	–	–	–	–

Tabla 3. Resultados sobre distintos datos de entrada

Si realizamos un número suficiente de ejecuciones, las funciones de selección candidatas se reducen a las verdaderamente óptimas, que en este problema son: orden decreciente de inicio y orden creciente de fin.

2.2. Ayudantes interactivos

Hemos diseñado tres ayudantes interactivos, cada uno específico para un problema resoluble de forma óptima mediante algoritmos voraces. AMO [4][8] está concebido para ayudar en el estudio del problema de la mochila, SEDA [7] para el problema de la selección de actividades y TuMiST [3] para el problema del árbol de recubrimiento de coste mínimo. Puede encontrarse más información en otras publicaciones sobre cómo satisfacen los niveles propuestos de la taxonomía de Bloom [11] y sobre las propias aplicaciones y sus evaluaciones de usabilidad [10].

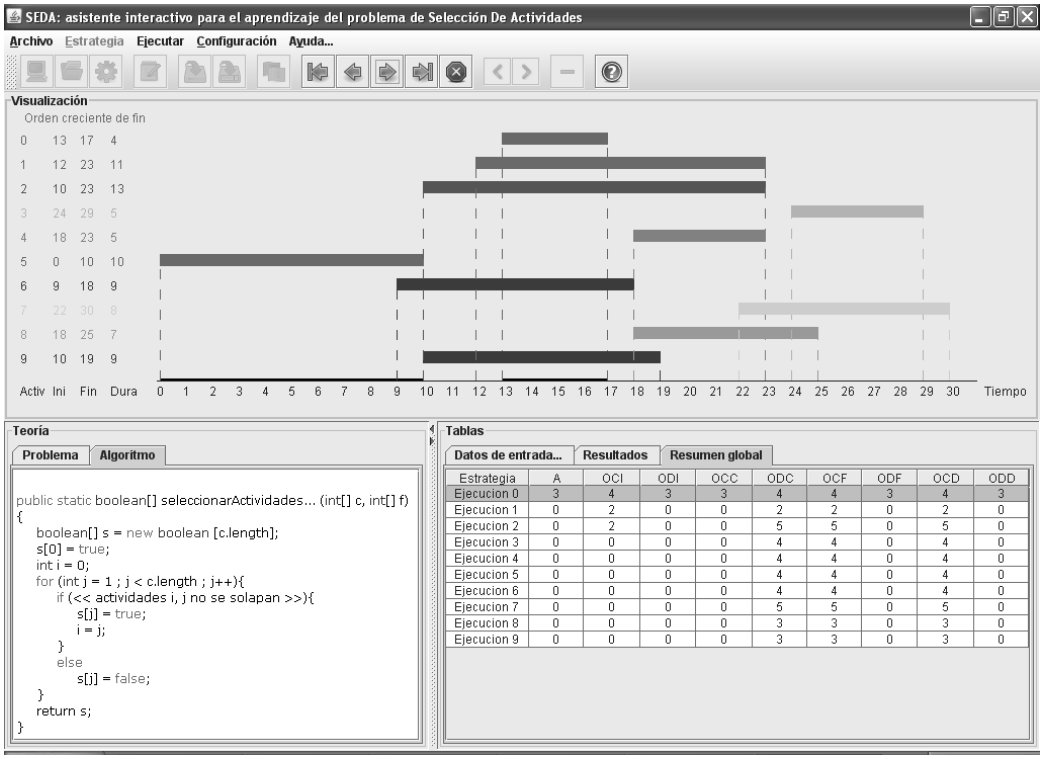

Figura 1. Una captura de pantalla del ayudante interactivo SEDA

Cada asistente se ha diseñado para ayudar en los siguientes niveles educativos:

- Comprensión. El alumno comprenderá el problema planteado y el algoritmo voraz que lo resuelve. El algoritmo será independiente de la función de selección elegida, por lo que puede contener fragmentos de pseudocódigo.
- Análisis. El alumno analizará el efecto de aplicar a unos datos de entrada el mismo algoritmo voraz, pero con diversas funciones de selección.
- Evaluación. El alumno evaluará el efecto de cada función de selección y seleccionará (empíricamente) las óptimas.

Los tres ayudantes interactivos tienen una interfaz similar. Por concreción, en el artículo, utilizamos el ayudante SEDA, desarrollado para el problema de selección de actividades [2]. La Figura 1 muestra la interfaz de usuario de SEDA. Se muestra un estado intermedio de la ejecución del algoritmo, para el primer conjunto de datos antes dado y con selección en orden creciente de fin.

Se distinguen claramente tres zonas, aparte del menú principal y la barra de iconos. En la parte superior se encuentra el panel de visualización, que muestra gráficamente los datos de entrada. En la parte inferior izquierda se encuentra el panel de teoría. Consta de dos pestañas: la pestaña del problema contiene su enunciado, y la del algoritmo (visible en la figura), su codificación en Java. Finalmente, la parte derecha contiene el panel de tablas, que a su vez contiene tres pestañas, cada una con una tabla: datos de entrada, resultados y resumen (visible). Las tres tablas se corresponden con las presentadas en la subsección 2.1.

3. Una sesión de usuario

Al arrancar la aplicación, el usuario sólo encuentra contenido en el panel de teoría. Como puede verse en la Figura 1, consta de dos

pestañas (problema y algoritmo), que el usuario debe leer para comprender el problema planteado.

Después, el usuario debe producir datos de entrada para ejecutar el algoritmo (con distintas estrategias). La aplicación permite al usuario que los datos procedan de tres fuentes distintas:

- Teclado. Se introducen mediante un diálogo, como se muestra en la Figura 2. Es útil para introducir un ejemplo concreto.
- Generador de números aleatorios. Es útil para producir rápidamente un ejemplo cualquiera.
- Fichero. Es útil para reproducir unos datos de entrada ya usados anteriormente, normalmente de tamaño grande.

Las dos primeras opciones piden al usuario que concrete el número de actividades (hasta un máximo de 12). El generador también pide que se especifique el instante máximo admisible (hasta 30). Los límites se han fijado para que los ejemplos sean manejables (y sus visualizaciones legibles).

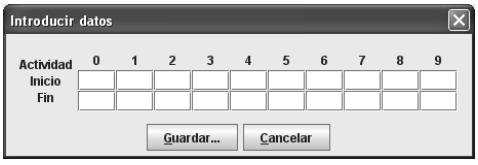

Figura 2. Diálogo para entrada de datos

Una vez producidos los datos, se muestran gráficamente en el panel de visualización. SEDA visualiza las actividades en formato bidimensional sobre un eje temporal horizontal. La visualización muestra tantas filas como actividades hay. Cada actividad se representa mediante una barra comprendida entre los instantes de comienzo y fin.

Las actividades se colorean con tonos según la función de selección activa, correspondiendo los tonos más oscuros a las actividades que antes se seleccionarán con dicha función. Obsérvese que al aumentar el número de objetos resulta más difícil distinguir la gama de tonos.

Por ejemplo, la Figura 3 muestra los datos del ejemplo dado en la subsección 2.1 para la función de selección en orden creciente de índice. Asimismo, la Figura 4 los muestra para la función de selección en orden creciente de duración.

Una vez seleccionada una función de selección, podemos examinar su ejecución de forma más o menos detallada. SEDA proporciona cuatro acciones de ejecución:

- Avanzar un paso en la ejecución. Muestra en el panel de visualización el resultado de seleccionar el siguiente objeto, según la función de selección activa.
- Realizar la ejecución completa. Ejecuta el algoritmo de forma completa utilizando la función de selección activa y muestra el resultado.
- Retroceder un paso. Es la acción complementaria de la primera.
- Retroceder al comienzo. Es la acción complementaria de la segunda.

Obsérvese que estas cuatro acciones permiten examinar el efecto de la función de selección activa sobre los datos de entrada, con el grado de detalle que el usuario desee. Incluso si hay pasos que se quieren volver a examinar, puede volverse atrás en la ejecución. Técnicamente, es una animación discreta del algoritmo controlada manualmente [9].

En las Figuras 5-8 se muestra el resultado de aplicar la función de selección en orden creciente de duración sobre los datos anteriores. La Figura 5 muestra el resultado de avanzar un paso, durante el cual se ha seleccionado la actividad más corta (actividad 0). Para resaltar la actividad válida seleccionada, se cambia su color y se le aplica un efecto de parpadeo. Finalmente, se muestra en gris para indicar que no se puede volver a seleccionar.

La Figura 6 muestra el resultado de avanzar otro paso más, en que se selecciona la actividad 4. La Figura 7 muestra el resultado de avanzar dos pasos más: selección de la actividad 3 y descarte de la actividad 8. La actividad 8 se descarta porque se solapa con las actividades 3 y 4, ya seleccionadas.

Finalmente, la Figura 8 muestra el resultado tras acabar la ejecución del algoritmo con la función de selección activa. Puede observarse que la duración de las actividades seleccionadas se ha proyectado sobre el eje horizontal, para apreciarlas mejor.

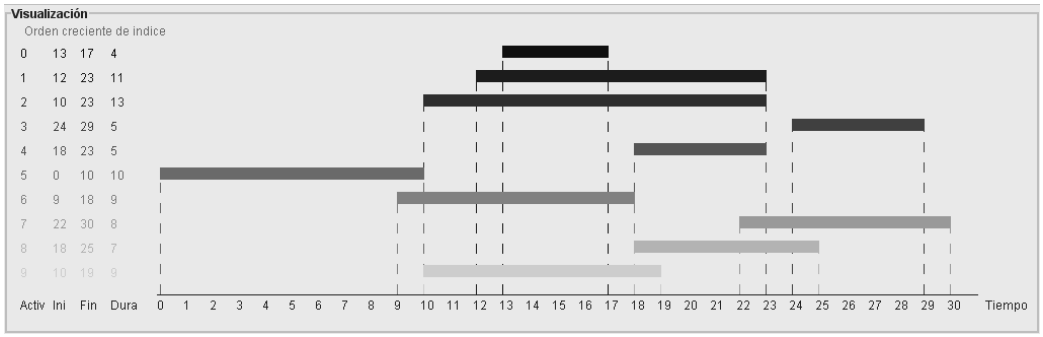

Figura 3. Visualización de actividades en orden creciente de índice

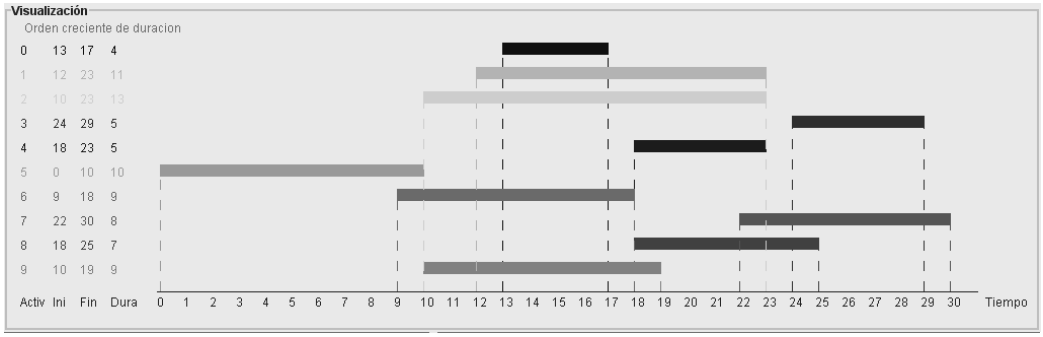

Figura 4. Visualización de actividades en orden creciente de duración

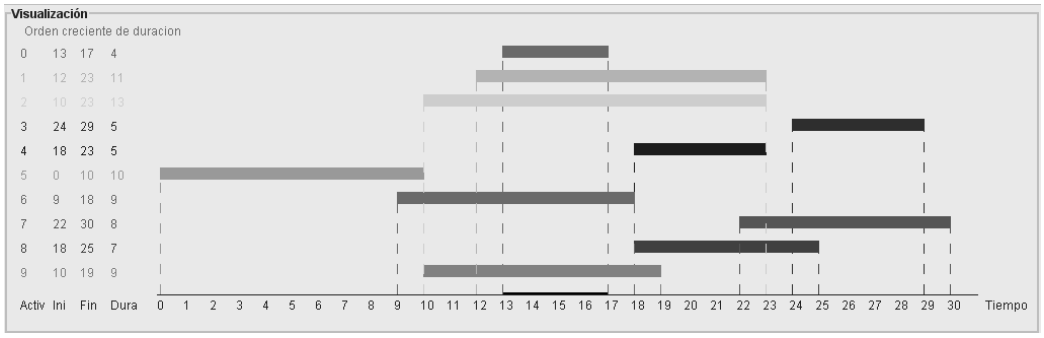

Figura 5. Visualización de actividades en orden creciente de duración tras un paso de ejecución

Las facilidades de ejecución descritas ayudan al alumno a seguir y comprender el efecto de cada estrategia. Su efecto también se refleja en la tabla de resultados, que tiene el mismo formato que la Tabla 1.

Una vez que el alumno ya entiende las distintas estrategias, suele interesarle ejecutarlas sobre otros datos de entrada, pero más rápidamente. Con este fin, SEDA proporciona otras dos acciones de ejecución:

- Ejecutar todas las estrategias.
- Ejecutar un subconjunto de estrategias.

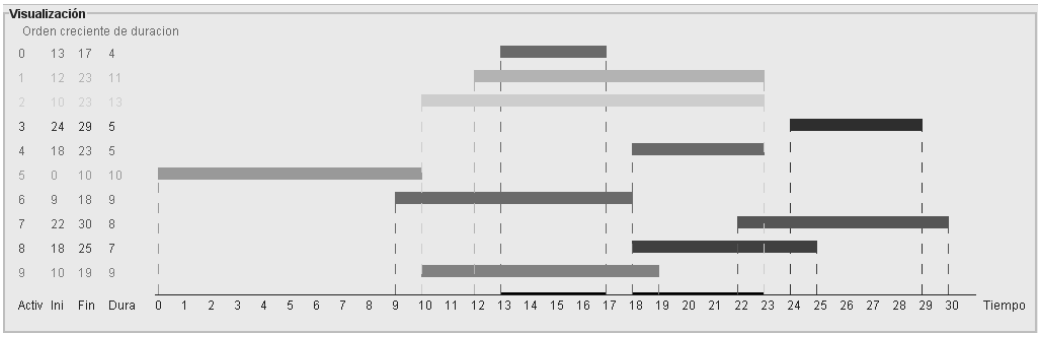

Figura 6. Visualización de actividades en orden creciente de duración tras dos pasos de ejecución

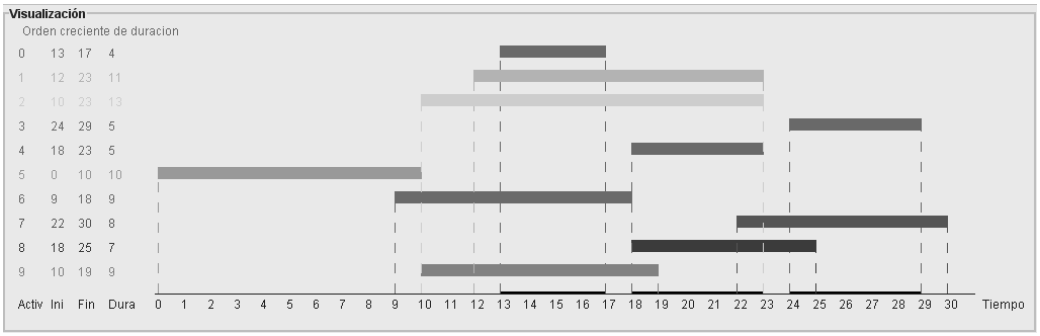

Figura 7. Visualización de actividades en orden creciente de duración tras cuatro pasos de ejecución

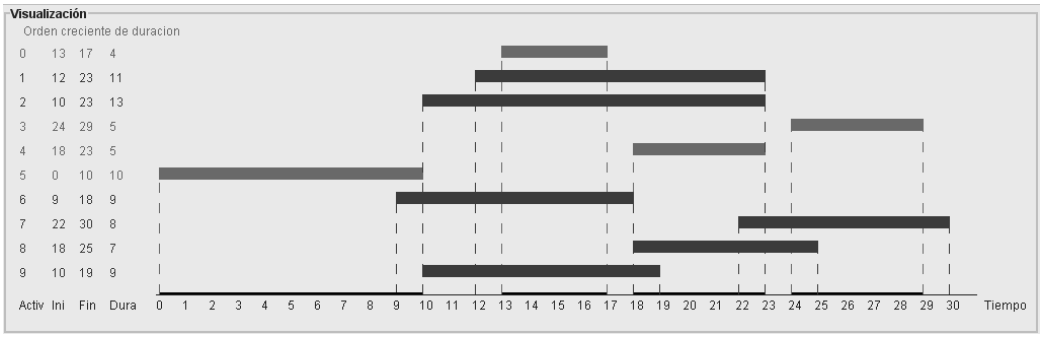

Figura 8. Visualización de actividades en orden creciente de duración al final de su ejecución

La primera opción es la más sencilla de usar para ir rápidamente. Sin embargo, la segunda es más adecuada ya que permite limitar las ejecuciones a las estrategias que han dado resultados óptimos hasta el momento, como se aprecia en la Tabla 3.

Según se va interaccionando con la aplicación, ésta va cargando automáticamente los resultados en las distintas tablas. Sin embargo, puede suceder que en algún momento haya excesiva información. En estos casos, el usuario puede usar varias acciones de borrado sobre las tablas:

- Borrar una fila de la tabla de resultados. Permite borrar el efecto de aplicar una estrategia, normalmente cuando se trata de una estrategia que ya se sabe que no es óptima.
- Borrar una fila de la tabla de resumen. Permite borrar un juego de datos de entrada completo y sus resultados, normalmente cuando su ejecución no aporta nada nuevo con respecto a otras anteriores.
- Borrar los datos de toda la sesión. Permite comenzar desde cero sin tener que salirse de la aplicación.

4. Experiencia, difusión y fomento

Hemos desarrollado nuestros ayudantes interactivos para su uso docente. Se han utilizado durante los cursos académicos 2007-2008, 2008-2009 y 2009-2010 en la asignatura "Diseño y Análisis de Algoritmos", de tercer curso de Ingeniería Informática en la Universidad Rey Juan Carlos. En el primer curso se usó AMO para la práctica correspondiente al tema de algoritmos voraces. En los dos cursos siguientes, el profesor usó AMO y TuMiST en el aula y después se usó SEDA para la práctica del tema. Cada sesión de prácticas se aprovechó para realizar una evaluación de usabilidad de la aplicación. La experiencia ha sido muy positiva, tanto por los resultados de las evaluaciones de usabilidad [10] como por las calificaciones obtenidas por los alumnos en la práctica.

También hemos realizado algunas acciones para difundir la existencia de estas aplicaciones y fomentar su uso, principalmente entre la comunidad de profesores universitarios de Informática.

Para la difusión de las aplicaciones, hemos desarrollado un sitio web con información útil (http://www.lite.etsii.urjc.es/greedex/). Consta de seis secciones, inspiradas en las recomendaciones dadas por [6]:

- Página principal. Contiene el nombre de las aplicaciones, breve descripción, persona de contacto y una captura de pantalla.
- Descripción. Contiene una descripción breve pero completa del conjunto de funciones.
- Uso educativo. Contiene una descripción de la materia y los objetivos de aprendizaje para los que se han diseñado las aplicaciones, sugerencias de uso y descripción de algunas de sus facilidades educativas.
- Documentación técnica. Describe los requisitos *software* para poder ejecutar las aplicaciones, así como información detallada de las evaluaciones de usabilidad realizadas.
- Documentación académica. Contiene referencias y ficheros PDF de las publicaciones realizadas, así como las memorias de los proyectos bajo las que se han desarrollado
- Descarga. Incluye enlaces a las aplicaciones, así como una breve descripción de cómo descargarlas e instalarlas.

Para fomentar el uso de los ayudantes interactivos, hemos incluido algunas facilidades que permitirán al profesor encontrarse más cómodo en su uso:

- Internacionalización. Puede elegirse el idioma de interacción de la aplicación. Actualmente, está limitado al español y el inglés, pero su extensión es sencilla.
- Configuración. El usuario puede configurar algunas de las características de los gráficos, de forma que le resulten más agradables estéticamente o se adapten mejor al estilo de la asignatura.
- Exportación. Permiten exportar imágenes estáticas y animaciones en formatos estándar (JPG, GIF), de forma que puedan usarse con otras aplicaciones.

5. Trabajo actual y futuro

Como se explica en el sitio web desarrollado, actualmente se están refundiendo las aplicaciones en una sola. Su nombre es GreedEx (de "GREEDy algorithms" y "EXperimentation"). En un primer paso, GreedEx soporta los algoritmos de la mochila y de la selección de actividades.

Asimismo, se están añadiendo nuevas facilidades que mejoren la usabilidad de la aplicación o que den mayor apoyo al método experimental. Por último, planeamos flexibilizar el diseño de GreedEx para que dé soporte a

problemas relacionados, resolubles con esta u otras técnicas de diseño.

Agradecimientos

Este trabajo se ha financiado parcialmente con el proyecto TIN2008-04103/TSI del MICINN.

Referencias

[1] Bloom, B., Furst, E., Hill, W., y Krathwohl, D.R. *Taxonomy of Educational Objectives: Handbook I, The Cognitive Domain*. Addison-Wesley, 1956.

[2] Cormen, T.H., Leiserson, C.E., y Rivest. R.L. *Introduction to Algorithms*. The MIT Press, 2ª ed., 2001.

[3] Debdi, O., y Granada, J.D. "Tutor interactivo para el aprendizaje del algoritmo voraz del árbol de recubrimiento de coste mínimo", proyecto de fin de máster, Universidad Rey Juan Carlos, Móstoles, Madrid, España, 2008.

[4] Gila Blázquez, J.A. "AMO: Asistente interactivo para el aprendizaje del algoritmo de la mochila", proyecto de fin de carrera, Universidad Rey Juan Carlos, Móstoles, Madrid, España, 2008.

[5] Hernán-Losada, I., Velázquez-Iturbide, J.Á., y Lázaro-Carrascosa, C.A. "Programming learning tools based on Bloom's taxonomy: Proposal and accomplishments". En *Proc. VIII International Symposium of Computers in Education, SIIE 2006*, pp. 325-334.

[6] Naps, T., Roessling, G., Cooper, S., Koldehofe, B., Leska, C. Dann, W., Korhonen, A., Malmi, L., Rantakokko, J., Ross, R.J., Anderson, J., Fleischer, R., Kuittinen, M., y McNally, M. "Evaluating the educational impact of visualization". En *ACM SIGCSE Bulletin*, 35(4):124-136, dic. 2003.

[7] Pablo García, M. "SEDA: Asistente interactivo para el aprendizaje del problema de selección de actividades", proyecto de fin de carrera, Universidad Rey Juan Carlos, Móstoles, Madrid, España, 2008.

[8] Segado Maillo, A. "Mejora de AMO: Asistente interactivo para el aprendizaje del algoritmo de la mochila", proyecto de fin de carrera, Universidad Rey Juan Carlos, Móstoles, Madrid, España, 2009.

[9] Stasko, J.T., Domingue, J., Brown, M.H., y Price, B.A. (eds.). *Software Visualization*. MIT Press, 1998.

[10] Velázquez Iturbide, J.Á., Lázaro Carrascosa, C.A., y Hernán Losada, I. "Asistentes interactivos para el aprendizaje de algoritmos voraces". En *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, IEEE-RITA*, 4(3):213-220, agosto 2009.

[11] Velázquez-Iturbide, J.Á., y Pérez-Carrasco, A. "Active learning of greedy algorithms by means of interactive experimentation". En *Proc. 14th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2009*. ACM Press, pp. 119-123.

[12] Velázquez-Iturbide, J.Á., Pérez-Carrasco, A., and Urquiza-Fuentes, J. "SRec: An animation system of recursion for algorithm courses". En *Proc. 13rd Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2008*. ACM Press, pp. 225-229.

[13] Velázquez-Iturbide, J.Á., Pérez-Carrasco, A., and Urquiza-Fuentes, J. "A design of automatic visualizations for divide-and-conquer algorithms". En *Electronic Notes in Theoretical Computer Science*, 224:159-167, enero 2009.