

Experimentación con algoritmos – herramientas informáticas y evaluación de dificultades de los alumnos

BANCO DE BUENAS PRÁCTICAS DOCENTES

JOSÉ ÁNGEL VELÁZQUEZ

I. La práctica

- **Título:** Experimentación con algoritmos – herramientas informáticas y evaluación de dificultades de los alumnos
- **Curso Académico:** 2014-15
- **Asignatura:** Algoritmos Avanzados
- **Área/Titulación:** Grado en Ingeniería Informática
- **Grupo de Estudiantes:** Grupo presencial del campus de Móstoles y grupo *online*

Se presenta un método didáctico diseñado para una asignatura de algoritmos. La experiencia consta de dos elementos:

- La herramienta interactiva OptimEx, diseñada con un objetivo de aprendizaje: experimentar con algoritmos de optimización y comparar sus resultados según la función objetivo especificada en el enunciado del problema correspondiente.
- Una organización de las prácticas adecuada al objetivo de aprendizaje anterior. Cada práctica propone resolver sucesivamente el mismo problema de optimización con una técnica de diseño distinta, salvo la práctica final en la que se comparan, con OptimEx, los resultados de los algoritmos desarrollados en prácticas anteriores.

También presentamos la puesta en práctica y evaluación del método didáctico en el curso 2013-14. Por un lado, se ha evaluado la herramienta OptimEx con un cuestionario de usabilidad, lo que ha permitido identificar oportunidades de mejora. Por otro lado, se han analizado con técnicas cualitativas las memorias elaboradas por los alumnos para la práctica final, lo que ha permitido: (a) comprobar la gran aceptación de la organización de prácticas por los alumnos, y (b) identificar actitudes, dificultades y errores conceptuales de los alumnos. Los hallazgos del análisis de las memorias han sido sorprendentes en varios aspectos y muy útiles para comprender mejor los logros, capacidades y dificultades de los alumnos. Los resultados de la evaluación se han utilizado para mejorar la experiencia en el curso 2014-15 y se ha seguido usando, evaluando y mejorando en el curso 2015-15..

2. Justificación

Nuestro trabajo se inscribe dentro del campo de investigación en enseñanza de la informática, que ha sido muy activo en años recientes (Fincher & Petre, 2004), principalmente sobre el aprendizaje de la programación. Sin embargo, la atención dedicada al aprendizaje de los algoritmos ha sido escasa, y aún más lo es la dedicada al aprendizaje de los algoritmos de optimización (Velázquez-Iturbide *et al.*, 2012). Consultando libros de texto sobre algoritmos, puede observarse que las actividades propuestas suelen consistir en diseñar o analizar un algoritmo para un problema dado, sin distinguir niveles de aprendizaje (Anderson *et al.*, 2001) ni promover el uso de métodos didácticos activos. El método didáctico aquí presentado propone prácticas de carácter innovador y contiene varias contribuciones a la didáctica de los algoritmos (aunque buena parte de la propuesta y de la metodología de investigación puede adaptarse a otras disciplinas).

La experimentación es una de las tres tradiciones intelectuales de la informática, inspirada en las ciencias experimentales (las otras dos tradiciones son la matemática y la ingenieril, véase Denning *et al.*, 1989). Puede destacarse que la experimentación con algoritmos suele limitarse a las propiedades de corrección o eficiencia, sin medirse su optimidad (Velázquez *et al.*, 2012). Nuestra propuesta cuenta con el antecedente de la herramienta GreedEx, que permite experimentar con varios algoritmos voraces (Velázquez *et al.*, 2013a). GreedEx se utiliza en la asignatura de “Diseño y Análisis de Algoritmos” de segundo curso de los grados en Ingeniería Informática e Ingeniería del Software. La herramienta se acompaña de materiales didácticos y una organización precisa de las clases (Velázquez, 2013b). Este método didáctico se ha evaluado con respecto al rendimiento académico de los alumnos (Esteban, Pizarro & Velázquez, 2014), con resultados positivos, y se ha utilizado para identificar concepciones erróneas de los alumnos sobre optimidad (Velázquez, 2013a).

Hemos buscado generalizar el ámbito de aplicabilidad de GreedEx a cualquier algoritmo de optimización mediante el desarrollo de una nueva herramienta interactiva, llamada OptimEx. La herramienta tiene una interfaz de usuario muy parecida a la de GreedEx, de forma que los alumnos que han utilizado GreedEx

pueden aprender el uso de OptimEx con poco esfuerzo. Ambas herramientas pueden usarse de forma combinada, primero GreedEx y luego OptimEx.

Ambas herramientas proporcionan entornos para experimentar con algoritmos de optimización. Sus posibilidades didácticas son similares a las proporcionadas en otros campos por los micromundos (Laurillard, 2012), con la salvedad de que su componente gráfico es limitado. También pueden considerarse herramientas cognitivas (Kommers, Jonassen & Mayes, 1992) ya que permiten una interacción activa (la experimentación), con creación de productos (las medidas obtenidas) y el control lo tiene el usuario (el alumno).

Una precaución importante al incorporar una nueva herramienta en la práctica docente es comprobar su usabilidad. Una herramienta puede estar diseñada según unos objetivos educativos claros pero, si su usabilidad es baja, los alumnos no la encontrarán adecuada para realizar las tareas docentes previstas. Si quieren evaluarse los efectos de la herramienta, como su eficiencia educativa, conviene que antes se haya evaluado su usabilidad y se hayan mejorado sus puntos débiles para que no interfieran en la medición pedagógica (Velázquez, Pérez & Debdí, 2013b).

Desde un punto de vista pedagógico, la experimentación es respaldada por el constructivismo. Esta teoría sostiene (Ben-Ari, 2001) que las personas no son pizarras en blanco donde se va grabando la información que reciben por sus sentidos, por ejemplo, en una clase magistral o leyendo un libro. Según el constructivismo, el conocimiento lo construye cada persona a partir de sus conocimientos previos y de la instrucción que recibe. Los métodos activos de aprendizaje buscan que el alumno tenga un papel más activo en la construcción del conocimiento que con las clases presenciales tradicionales. En particular, la experimentación proporciona a los alumnos la oportunidad de contrastar personalmente resultados prácticos con predicciones teóricas.

Según el constructivismo, podemos distinguir entre modelos conceptuales y mentales (Norman, 1983). Un modelo conceptual es una representación del conocimiento construida por el profesor para presentársela a los alumnos y transmitirles su conocimiento. Por tanto, un modelo conceptual debe ser preciso, completo y coherente. Por otro lado, un modelo mental es la representación que el alumno construye sobre un modelo conceptual, es decir, su comprensión real del mismo. Por tanto, un modelo mental con frecuencia es incompleto y ambiguo. No existen modelos mentales correctos, ya que varían con cada persona, pero sí podemos hablar de modelos viables o inviables, según sean capaces de explicar y comprender los fenómenos observados. Podemos hablar de malentendidos, errores conceptuales o malas concepciones (*misconceptions*) para referirnos a aspectos clave de un modelo mental que dificultan o imposibilitan su viabilidad. Desde el punto de vista del constructivismo las malas concepciones pueden ser aprovechadas por el profesor para ayudar a los alumnos a construir un conocimiento viable.

Uno de los resultados del trabajo aquí presentado es la identificación de dificultades y malas concepciones de los alumnos sobre algoritmos de optimización. Dentro de la enseñanza de la informática, se han realizado estudios muy variados sobre las dificultades de los alumnos, sobre todo en el aprendizaje de la programación (Clancy, 2004). Sin embargo, no existen apenas trabajos sobre las malas concepciones de los alumnos sobre algoritmos, destacando un trabajo nuestro sobre optimización y algoritmos voraces (Velázquez, 2013a).

La experiencia presentada propone una organización precisa de las prácticas, consistente en pedir a los alumnos la resolución del mismo problema mediante distintas técnicas de diseño. De esta forma, los alumnos no tienen que esforzarse en comprender un nuevo problema con cada práctica, facilitando que se concentren en la técnica de diseño correspondiente. Además, disponen de un intervalo de tiempo razonable para recopilar un conjunto de algoritmos diferentes. Esta revisita de contenidos ya conocidos tiene cierta relación con la técnica didáctica en espiral (Powers & Powers, 1999) y, sobre todo, con la teoría de la variación (Marton & Booth, 1997).

En nuestro caso, cada práctica permite estudiar el diseño e implementación de cada técnica, mientras que en la última práctica de la asignatura se estudia cada técnica solamente desde el punto de vista de su resultado, resaltando su carácter de técnica exacta (es decir, óptima) o inexacta (es decir, subóptima). La última práctica es distinta porque su objetivo principal no es desarrollar un algoritmo sino comparar los algoritmos desarrollados en prácticas anteriores. Se permite y fomenta que realicen la práctica en pareja para que disfruten de los beneficios del debate en el aprendizaje colaborativo. Una buena parte de la práctica consiste en la comparación de datos y obtención de conclusiones. Esta actividad permite de forma natural el debate entre los dos miembros de un equipo.

3. Desarrollo

Objetivos

El objetivo principal de la actividad didáctica propuesta es:

“Mejorar la comprensión de las técnicas de diseño de algoritmos de optimización más comunes mediante la experimentación.”

Los objetivos parciales de la realización de la actividad son:

- Diseñar y evaluar una organización adecuada de las prácticas de la asignatura.
- Evaluar la usabilidad de OptimEx.
- Evaluar la comprensión por los alumnos de las diferencias entre técnicas de diseño de algoritmos de optimización que permiten obtener soluciones óptimas y subóptimas.

Descripción de la experiencia educativa

La experiencia educativa se ha llevado a cabo en la asignatura “Algoritmos Avanzados”, de 4º curso del Grado en Ingeniería Informática. La asignatura se imparte en dos grupos: grupo presencial del campus de Móstoles y grupo on-line.

Organización de las prácticas

Los alumnos debían realizar cinco prácticas obligatorias y cuatro trabajos optativos de profundización sobre algunas partes del temario. Para cada práctica o trabajo debe entregarse una memoria escrita según un índice detallado en el enunciado. Las prácticas fueron:

- Reimplementación de un algoritmo voraz.
- Desarrollo de un algoritmo de vuelta atrás y otro de ramificación y poda.
- Eliminación de redundancia en un algoritmo con recursividad múltiple.
- Desarrollo de un algoritmo de programación dinámica.
- Desarrollo de un algoritmo heurístico y comparación con algoritmos exactos.

La experiencia que presentamos está formada por las prácticas 1, 2, 4 y 5. (La práctica 3 era introductoria a los conceptos necesarios para realizar la práctica 4.) En la práctica 1 se proporcionaba un algoritmo voraz para resolver el problema de selección de actividades (Cormen *et al.*, 2009, pp. 415-422; Kleinberg & Tardos, 2006, pp. 116-122), que debía modificarse según varios requisitos. Las prácticas 2 y 4 proponían resolver un problema relacionado, el problema de selección de actividades ponderadas (Kleinberg & Tardos, 2006, pp. 252-258), mediante varias técnicas de diseño de algoritmos exactos. Finalmente la práctica 5 proponía experimentar con los algoritmos exactos y heurísticos desarrollados en las prácticas anteriores para el problema de selección de actividades ponderadas y comparar sus resultados usando OptimEx. En términos de los objetivos de aprendizaje de la taxonomía revisada de Bloom (Anderson *et al.*, 2001), las primeras prácticas pretenden que los alumnos alcancen los niveles cognitivos de aplicación, análisis o diseño, mientras que la práctica 5 se sitúa en el nivel de evaluación.

La práctica 5 es la más interesante ya que cierra la asignatura dando una visión global de la misma, por lo que a continuación nos centramos en ella. El enunciado de la práctica repetía la especificación del problema de selección de actividades ponderadas, ya conocido por los alumnos. A los alumnos se les exigía que utilizaran OptimEx para comparar la optimidad de al menos 3 algoritmos. Se les proporcionaba un algoritmo voraz basado en orden creciente de duración. Un segundo algoritmo debían tomarlo los alumnos de su práctica 1 y adaptarlo al nuevo problema. Por último, podían tomar otros algoritmos desarrollados en las prácticas 2 y 4. Suponiendo que los algoritmos estuvieran correctamente desarrollados, la respuesta esperada era que los algoritmos de vuelta atrás, ramificación y poda o programación dinámica eran óptimos mientras que los algoritmos voraces, no.

La práctica podía hacerse individualmente o en parejas. La sesión presencial duró 2 horas.

Materiales

Los alumnos se podían descargar del campus virtual el material necesario:

- Enunciado de la práctica.
- Herramienta interactiva OptimEx y una pequeña guía de usuario (de 2 páginas).
- Parte de los materiales necesarios para la experimentación: una clase Java con un algoritmo voraz y un fichero con 100 datos de entrada.

La memoria de la práctica debía entregarse en el plazo de una semana y debía tener la siguiente estructura:

- Identificación de la técnica de diseño y código de los algoritmos utilizados.
- Tabla con los resultados de la experimentación.
- Conclusiones.

A los alumnos del grupo presencial también se les entregó al final de la sesión de prácticas un cuestionario de usabilidad sobre la herramienta OptimEx. Su cumplimentación era voluntaria y anónima. Por brevedad, omitimos esta parte de la evaluación de la descripción de esta buena práctica.

La herramienta interactiva Optimex

OptimEx es el acrónimo de “OPTIMization EXperimentation”. El objetivo de OptimEx es permitir al alumno experimentar con distintos algoritmos para un mismo problema de optimización, comparándolos y determinando su optimalidad o suboptimalidad. En este último caso, también se calcula su desviación respecto a la solución óptima, tanto en porcentaje de casos como en desviación media. OptimEx permite experimentar con cualquier algoritmo codificado en Java, siempre que los algoritmos a comparar estén codificados en una sola clase y que su cabecera sólo contenga tipos de datos predefinidos.

OptimEx se ha diseñado de forma que sea una aplicación sencilla y con una interfaz de usuario intuitiva, fácil de aprender y de usar. La Fig. 1 muestra una captura de pantalla de OptimEx. Puede observarse que su interfaz de usuario contiene dos paneles: código y tablas. El panel de código está situado a la izquierda e incluye un editor que permite cargar, editar y almacenar el código de los algoritmos a ejecutar. En la figura se ha cargado un fichero con cuatro algoritmos para resolver el problema de la selección de actividades ponderadas (Kleinberg & Tardos, 2006, pp. 252-258): tres algoritmos voraces y uno de vuelta atrás. El panel de tablas alberga tres tablas para almacenar y comparar los resultados de los algoritmos ejecutados. La tabla visible en la figura es la tabla histórica, que almacena los resultados obtenidos por todos los algoritmos seleccionados en todas las ejecuciones realizadas. Gracias al resaltado de celdas con colores, puede observarse que los algoritmos voraces no son óptimos mientras que el de vuelta atrás sí lo es (todas las celdas de su columna aparecen en color gris).

- Los pasos que el alumno debe seguir para realizar un experimento pedido en una práctica son:
- Editar o cargar de fichero una clase de Java y compilarla.
- Seleccionar el problema que resuelven los algoritmos a comparar.
- Introducir los datos de entrada.
- Ejecutar los algoritmos sobre los datos de entrada.
- Comparar los resultados mostrados en la tabla de resumen y, quizá en la tabla histórica. Exportar tablas en ficheros gráficos para documentar la memoria de la práctica.

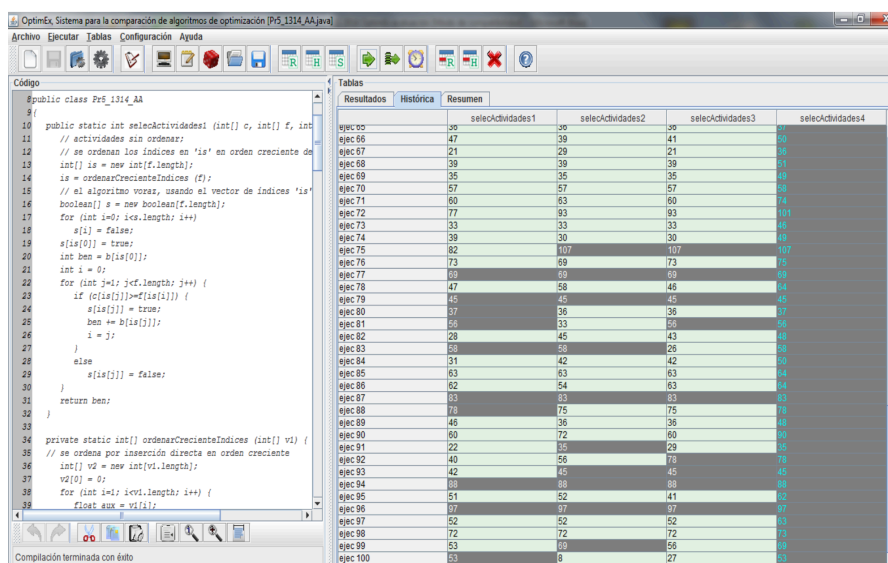


Fig 1. Interfaz de usuario de OptimEx tras experimentar con cuatro algoritmos para el problema de la selección de actividades ponderada

Los resultados se presentan en tres tablas: de resultados, histórica y de resumen. La primera muestra el resultado de ejecutar los algoritmos seleccionados sobre el último juego de datos de entrada. La tabla histórica (véase la Fig 1) muestra el resultado obtenido por cada algoritmo para cada juego de datos de entrada. Por último, la tabla de resumen da ciertos estadísticos que resumen el comportamiento de cada algoritmo. La Fig 2 muestra la tabla de resumen correspondiente a la ejecución mostrada (parcialmente) en la Fig 1.

Tablas				
Resultados Histórica Resumen				
Medida	selecActividades1	selecActividades2	selecActividades3	selecActividades4
Núm. ejecuciones	100	100	100	100
% subóptimas	80,00 %	77,00 %	82,00 %	0,00 %
% óptimas	20,00 %	23,00 %	18,00 %	100,00 %
% superóptimas	0,00 %	0,00 %	0,00 %	0,00 %
% desviación media	24,29 %	22,18 %	22,79 %	0,00 %
% desviación máxima superóptima	0,00 %	0,00 %	0,00 %	0,00 %
% desviación máxima subóptima	77,78 %	84,91 %	71,88 %	0,00 %

Fig 2. Tabla de resumen tras experimentar con 4 algoritmos y 100 datos de entrada para el problema de la selección de actividades ponderada

4. Resultados

Metodología de análisis

Las memorias entregadas se analizaron de forma cualitativa según los principios de la teoría fundamentada (Glaser & Strauss, 1967). Se realizaron tres rondas de análisis:

1. Ronda exploratoria. Se analizaron solamente los cinco informes del grupo on-line para identificar los elementos destacados de cada práctica.
2. Se diseñó una tabla donde registrar los elementos que se consideraron importantes para el análisis. Aunque el análisis procedió de forma secuencial, de vez en cuando se retrocedía a informes anteriores para comprobar que un elemento nuevo o importante se había tenido en cuenta, o para consultar en qué columna de la tabla se había anotado algún elemento.
3. A partir de la información contenida en la tabla, se elaboró una lista de preguntas que podían tener respuesta. También se reestructuró la tabla. Por último, se hizo una última ronda de análisis. Al igual que en la segunda ronda, el proceso de análisis fue secuencial pero con frecuentes retrocesos.

Los aspectos analizados finalmente fueron: errores y dificultades de los alumnos, actitudes y autoconocimiento, y malas concepciones sobre la materia (algoritmos de optimización)..

Resultados

Se recogieron un total de 30 memorias de la práctica 5. Dos memorias (correspondientes a dos individuos del grupo presencial) aclaraban que habían tenido dificultades graves para utilizar OptimEx, por lo que no pudieron contestar a lo pedido en la práctica. Los 28 grupos restantes se distribuyen de la siguiente forma: 22 del grupo presencial, 5 del grupo on-line y 1 de alumnos "oyentes".

Presentamos los resultados obtenidos desglosados en tres apartados: errores y dificultades, actitudes y autoconocimiento, y malas concepciones.

Errores y dificultades

La primera cuestión que hemos analizado es si los grupos realizaron la experimentación de forma satisfactoria. La Tabla I muestra los resultados globales de la práctica, distinguiendo entre quienes identificaron correctamente los algoritmos óptimos, quienes lo hicieron parcialmente y quienes no lo consiguieron. Decimos que un grupo identificó parcialmente los algoritmos óptimos si identificaron algunos pero no todos (normalmente identificaron a uno entre dos).

Resultado	Número de grupos	% de grupos
Correcto	8	28'6%
Parcial	2	7'1%
Incorrecto	18	64'3%

Tabla I. Resultados globales de la práctica

Puede observarse que la mayoría de los grupos realizaron insatisfactoriamente la experimentación. Veamos qué hicieron mal los grupos de la categoría "incorrecto":

- Un grupo realizó bien la experimentación pero no identificó ningún algoritmo como óptimo, ni siquiera el algoritmo que resultó mejor en el 100% de los casos.
- Cuatro grupos marcaron como óptimo un algoritmo que es subóptimo. Como consecuencia, otros algoritmos produjeron resultados mayores ("superóptimos") para algunos juegos de datos.
- Trece grupos han realizado una experimentación abierta en la que ningún algoritmo ha producido resultados óptimos.

Analizando las condiciones de experimentación utilizadas por algunos grupos, identificamos dos factores que influyeron en algunos resultados negativos:

- Seis grupos no usaron exclusivamente los datos de entrada suministrados. Dadas las limitaciones de OptimEx para generar datos aleatorios, algunos datos generados eran inválidos.
- Dos grupos no marcaron la casilla de maximizar.

Actitudes y autoconocimiento

Si analizamos las actitudes de los grupos ante las dificultades encontradas, hay que señalar primero que no ha sido frecuente encontrar grupos que hayan advertido problemas. Entre estos, sólo estamos seguros de un grupo que haya corregido los problemas. Otros dos grupos informan de que han realizado mejoras tras detectar problemas, pero no hay constancia de estas mejoras en sus memorias.

Otros cuatro grupos detectan resultados erróneos pero los atribuyen a un algoritmo suyo, al del profesor o incluso cambian lo aprendido para adaptarse al nuevo hallazgo.

El enunciado de la práctica sugería a los alumnos que realizaran comentarios de valoración de la práctica. La Tabla II muestra el número de grupos que indicaron que la práctica era fácil, que expresaron su

satisfacción por realizar esta práctica como colofón de la asignatura (al englobar todas las técnicas de diseño anteriores) o que fueron conscientes de algunas dificultades con la práctica.

Resultado	Práctica fácil	Satisfacción por la práctica	Conciencia de dificultades
Correcto	2	3	1
Parcial	–	1	–
Incorrecto	4	5	2

Tabla II. Valoración de la práctica

En cuanto a la apreciación de la dificultad de la práctica, cuatro grupos de los 18 que hicieron mal la práctica (22%) declaran que era fácil, revelando un escaso autoconocimiento. Una tercera parte de los grupos (9 de 28) manifiestan su satisfacción por realizar una práctica que permite relacionar casi todo lo estudiado en la asignatura. La satisfacción es independiente del resultado de la práctica. Por último, solamente tres grupos identificaron algunas dificultades para realizar bien la práctica.

Malas concepciones

Una dificultad frecuente es el propio concepto de “óptimo”. Conviene que distingamos dos usos del término:

- “**Resultado óptimo**” significa el máximo valor que puede obtenerse para unos datos dados (o valor mínimo si el problema es de minimización).
- “**Algoritmo óptimo**” es un algoritmo que devuelve el resultado máximo (“óptimo”) siempre, es decir, para unos datos de entrada válidos cualesquiera.

Ambos significados los han usado correctamente diversos grupos de las tres categorías de resultados globales. Otros grupos no comentan sus hallazgos o los describen de manera trivial, de forma que no es posible conocer su uso del lenguaje.

Sin embargo, hay algunos grupos que utilizan el término “algoritmo óptimo” para referirse al algoritmo que calcula un resultado óptimo en un porcentaje mayor de casos. Este uso suele encontrarse en grupos que hicieron la práctica mal (6 grupos), pero también en algunos grupos que la hicieron bien (dos). Por ejemplo: “Según se puede ver en la tabla, el algoritmo recursivo (con poda) es un 94% óptimo en comparación con los otros dos algoritmos, (...)”. Este uso de la palabra “óptimo” suele venir expresado como “más óptimo” o “menos óptimo”.

Tres grupos alternan usos correctos e incorrectos del término en la memoria. Quizá esta imprecisión sea la razón de algunas de las propuestas erróneas detectadas: que un algoritmo voraz obtenga resultados mejores que un algoritmo exacto para el mismo problema o que ningún algoritmo obtenga resultados mejores en el 100% de los casos.

Resumen de resultados

Podemos resumir los hallazgos realizados:

- Son mayoría los grupos que han realizado mal la experimentación, aunque la frontera entre prácticas bien y mal realizadas es difusa. Algunos grupos que la hicieron bien utilizan un lenguaje equivoco o algoritmos que no son correctos del todo, otros grupos han encontrado un algoritmo óptimo pero no otro que también lo es, y algunos grupos cuyos resultados son erróneos tienen soluciones cercanas a la correcta.
- Algunos grupos realizaron mal la experimentación por no dedicar suficiente atención a las condiciones de realización del experimento indicadas en el enunciado.
- La mayor parte de las prácticas mal realizadas no han interpretado bien los resultados mostrados en las tablas o han utilizado algoritmos subóptimos. Lo primero puede deberse al propio formato de las tablas o a un conocimiento superficial y frágil de la materia. Cuando se detectan errores, no se corrigen, e incluso se hacen propuestas contradictorias con lo explicado en la asignatura.

- Numerosos grupos declaran su satisfacción con la organización de prácticas y su percepción de la práctica como sencilla.
- Es bastante frecuente un uso impreciso del término “óptimo”.

Hay que añadir que OptimEx es valorado de forma muy positiva, aunque se han recogido numerosas sugerencias de mejora.

Evaluación

Al evaluar la experiencia, primero valoramos de los resultados obtenidos (desglosados según los 3 objetivos parciales identificados) y después hacemos algunas recomendaciones de mejora de la actividad.

Valoración de la organización de las prácticas

La percepción de los alumnos sobre la organización de las prácticas y sobre la práctica 5 como englobadora de prácticas anteriores ha sido muy positiva. Así lo expresaron libremente la tercera parte de los alumnos en la sección de conclusiones de memoria, cuando esta sección animaba genéricamente a incluir comentarios sobre la práctica.

También ha permitido al profesor conocer algunos conceptos que los alumnos adquieren de forma deficiente o superficial:

- Algoritmos exactos y heurísticos, junto con las técnicas de diseño estudiadas que corresponden a cada categoría.
- Dificultades de los alumnos para interpretar los resultados de una experimentación, sobre todo aquellos resultados que son inadmisibles si un algoritmo es exacto o inexacto.

Diversos usos del término “óptimo”.

Valoración de la comprensión de los alumnos

Claramente, las calificaciones han sido peores de lo esperado. Solamente un tercio de los grupos han realizado la práctica bien, total o parcialmente. Hay que tener en cuenta que la práctica era “sencilla” porque no exigía apenas diseñar ni programar (solamente había que adaptar el algoritmo desarrollado en la práctica 1 a un enunciado ligeramente distinto). Además, se les proporcionaba uno de los algoritmos a comparar y los datos a utilizar para la experimentación. Por tanto, su labor casi se reducía a experimentar con los algoritmos e interpretar los datos obtenidos. Hubiera sido deseable, aunque no se pedía, que, en caso de obtener resultados inaceptables, buscaran las razones e incluso depuraran sus algoritmos.

Esta alta tasa de suspensos puede explicarse (al menos parcialmente) por la comprensión deficiente de varios conceptos de la asignatura antes señalados: algoritmos exactos e inexactos, planificación de un experimento sobre optimidad e interpretación de resultados, y usos del término “óptimo”.

Una mayoría de grupos han sido incapaces de distinguir los datos relevantes de la tabla de resumen, así como inconsistencias dentro de la propia tabla o entre la tabla y lo explicado en la asignatura. En términos de la taxonomía revisada de Bloom (Anderson et al., 2001), la primera actividad se sitúa en el nivel de análisis mientras que la segunda se sitúa en el nivel de evaluación. Puede mejorarse la guía de uso de OptimEx para tratar ambas cuestiones.

Por último, algunas dificultades de los alumnos tienen que ver con su autoconocimiento. El autoconocimiento es la categoría más alta de conocimiento que puede alcanzar un alumno según la taxonomía revisada de Bloom (Anderson et al., 2001). Su mejora no es sencilla y debe fomentarse a lo largo de toda la asignatura. En todo caso, podría ampliarse el tiempo dedicado a realizar la práctica, permitiendo que eliminen los errores detectados por el profesor y que realicen una segunda entrega.

En todo caso, todas estas dificultades son coherentes con hallazgos de la psicología de la programación. Winslow (1996) afirma que se tarda diez años en pasar de aprendiz a experto en cualquier materia. Nuestros alumnos ya están en cuarto curso pero se aprecia esta falta de madurez.

Mejoras realizadas y futuras

En el curso 2014-15 se sigue una planificación parecida de la asignatura, con la misma organización de prácticas obligatorias. Se tiene previsto volver a realizar la experiencia completa, incorporando varias mejoras. Ya se han incorporado varias mejoras en la práctica 5:

- **Enunciado de la práctica.** El enunciado da porcentaje de resultados óptimos esperados para el algoritmo voraz proporcionado, de forma que sirvan a los alumnos de control de sus resultados. También indica más claramente qué deben incluir en la memoria, de forma que se disponga de más información sobre el trabajo realizado por los alumnos (p.ej. si han debido depurar algún algoritmo), que se identifiquen los algoritmos óptimos y que incluyan los resultados mediante las tablas histórica y de resumen.
- **Documentación.** La guía de usuario de OptimEx se ha estructurado en secciones, se ha mejorado la descripción de la tabla de resumen y se ha añadido una sección donde se identifican dificultades. Se identifican 3 situaciones inaceptables en una experimentación con algoritmos exactos (errores de ejecución, resultados “superóptimos” o todos los algoritmos subóptimos), explicando en cada caso su justificación y su solución.
- **Segunda entrega.** En la asignatura se permite mejorar cualquier práctica obligatoria a partir de los comentarios del profesor y realizar una segunda entrega. Esta posibilidad es muy apreciada por los alumnos porque, aparte de subir su nota, les permite aprender de sus propios errores. En el curso 2013-14, esto no se hizo con la práctica 5 por la falta de tiempo. Este curso hemos modificado la planificación temporal para permitir esta segunda entrega.

También queremos acometer algunas acciones en el resto del curso para mejorar el aprendizaje de: concepto de óptimo, experimentación (planificación e interpretación de resultados) y las nociones de algoritmo exacto y aproximado. Basándonos en las recomendaciones dadas por la teoría de la variación, hemos identificado algunas modificaciones que deberían mejorar la percepción de los conceptos asociados. La primera está realizada y la segunda está en preparación:

- **Práctica 2 (algoritmos de vuelta atrás y de ramificación y poda).** Se motiva la necesidad de estas técnicas en una primera sección (nueva) donde se pide un ejemplo que muestre que los algoritmos voraces óptimos para el problema de selección de actividades no son óptimos para el problema de la selección de actividades ponderadas.
- **Sesión de clase (con aprendizaje activo) dedicada a conceptos relacionados con optimidad:** óptimo y optimal, adjetivos absolutos (p.ej. “óptimo”), problemas y algoritmos de optimización, mínimos y máximos, cotas, etc. Es preferible celebrarla hacia el final de la asignatura para poder compararse las distintas técnicas entre sí, tanto exactas como inexactas.

Conclusiones y trabajos futuros

Se ha presentado un método didáctico diseñado para una asignatura de algoritmos. La experiencia consta de dos elementos. Por un lado, la herramienta interactiva OptimEx, diseñada con el objetivo de aprendizaje de evaluar las características de distintas técnicas de diseño de algoritmos respecto a la propiedad de optimidad. Por otro lado, una organización de las prácticas adecuada al objetivo de aprendizaje anterior. Cada práctica propone resolver sucesivamente el mismo problema con una técnica de diseño distinta, salvo la práctica final en la que se comparan, con OptimEx, los resultados de los algoritmos desarrollados en prácticas anteriores.

El método didáctico se ha puesto en práctica y evaluado en el curso 2013-14. Por un lado, se ha evaluado la usabilidad de OptimEx con un cuestionario, lo que ha permitido identificar mejoras a realizar. Por otro lado, se han analizado con técnicas cualitativas las memorias de la práctica final, lo que ha permitido: (a) comprobar la gran aceptación por los alumnos de la organización de prácticas, y (b) identificar actitudes, dificultades y errores conceptuales de los alumnos. Los hallazgos del análisis de las memorias han sido sorprendentes en varios aspectos y muy útiles para comprender mejor el nivel de aprendizaje y las dificultades de los alumnos. Los resultados de la evaluación se están utilizando para mejorar las actividades y los materiales educativos en el curso 2014-15.

Quedan abiertas varias líneas de trabajo futuro. En primer lugar, sería interesante revisar la planificación de la asignatura usando la teoría de la actividad de forma más sistemática; algunos antecedentes que pueden orientarnos han abordado la enseñanza de las estructuras de datos o de la programación orientada a objetos.

En segundo lugar, debería mejorarse OptimEx con las sugerencias recogidas en la evaluación de usabilidad y con otras mejoras identificadas por el profesor. Aunque puede haber algunas dificultades técnicas, el problema principal es la falta de recursos humanos. En tercer lugar, sería interesante medir si OptimEx incrementa la motivación de los alumnos. Finalmente, quedan numerosas oportunidades para mejorar la didáctica de los algoritmos de optimización. Dos ejemplos de retos pendientes son: dar un planteamiento global al razonamiento de optimidad (que agrupe demostraciones formales y experimentos) o elaborar un inventario estandarizado de conceptos sobre la materia.

5. Equipo docente



J. Ángel Velázquez es Licenciado en Informática (1985) y Doctor en Informática (1990) por la Universidad Politécnica de Madrid, España. Ha sido profesor desde 1985 en la Facultad de Informática de la Universidad Politécnica de Madrid. En 1997 se incorporó a la Universidad Rey Juan Carlos. Actualmente está adscrito a la Escuela Técnica Superior de Ingeniería Informática, siendo Catedrático de Universidad y director del Laboratorio de Tecnologías de la Información en la Educación (LITE). Es autor de más de un centenar de publicaciones nacionales e internacionales en libros, revistas y congresos. Sus áreas de investigación son innovación docente en programación, software educativo para la enseñanza de la programación y visualización del software. El Prof. Velázquez es miembro de IEEE Computer Society, IEEE Education Society, ACM y ACM SIGCSE. Actualmente es Presidente de la Asociación para el Desarrollo de la Informática Educativa (ADIE), sociedad científica integrada en la Confederación de Sociedades Científicas de España (COSCE).

Publicaciones y bibliografía

Publicaciones

La experiencia presentada había dado lugar, en el momento de presentar esta práctica docente, a varias publicaciones:

J. Ángel Velázquez Iturbide, Roberto Martín Torres y Noé González Rabanal, "OptimEx: un sistema para la experimentación con algoritmos de optimización", *SIIE'13 XV International Symposium on Computers in Education – Proceedings*, Maria José Marcelino, Maria Cristina Azebedo Gomes y António José Mendes (eds.), 2013 (ISBN 978-989-96261-3-3), pp. 30-35.

J. Ángel Velázquez Iturbide, "Difficulties, attitudes and misconceptions on experimenting with optimization algorithms", *Proceedings of the 2014 International Symposium on Computers in Education (SIIE'14)*, IEEE Xplore, 2014 (ISBN 978-1-4799-4428-6), pp. 17-22.

J. Ángel Velázquez Iturbide, "Una evaluación de usabilidad de OptimEx". En *Serie de Informes Técnicos DLSI1-URJC* (ISSN 1988-8074), nº. 2014-02, Universidad Rey Juan Carlos.

J. Ángel Velázquez Iturbide, "Una evaluación cualitativa de la comprensión de la optimidad". En *Serie de Informes Técnicos DLSI1-URJC* (ISSN 1988-8074), nº. 2014-03, Universidad Rey Juan Carlos.

Bibliografía

ACM and IEEE Computer Society, The Joint Task Force on Computing Curricula (2013). *Computer Science Curricula 2013*. <http://www.acm.org/education/CS2013-final-report.pdf>.

L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, P. R. Pintrich, J. Rath y M.C. Wittrock (2001). *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Nueva York, NY: Addison Wesley Longman.

M. Ben-Ari (2001). "Constructivism in computer science education". *Journal of Computers in Mathematics and Science Teaching*, 20(1):45-73.

M. Clancy (2004). "Misconceptions and attitudes that interfere with learning to program". En *Computer Science Education Research*, S. Fincher y M. Petre (eds.), Londres: Routledge, pp. 85-100.

- T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein, (2009). *Introduction to Algorithms*, 3ª ed. Massachusetts, MA: The MIT Press.
- P. J. Denning, D. E. Comer, D. Gries, M. C. Mulder, A.B. Tucker, A.J. Turner y P.R. Young (1989). "Computing as a discipline". *Communications of the ACM*, 32(1):9-23.
- N. Esteban-Sánchez, C. Pizarro y J. Á. Velázquez-Iturbide (2014). "Evaluation of a didactic method for the active learning of greedy algorithms", *IEEE Transactions on Education*, 57(2):83-91.
- S. Fincher y M. Petre (eds.) (2004). *Computer Science Education Research*. Londres: Routledge Falmer.
- B. Glaser y A. Strauss (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL: Aldine.
- J. Kleinberg y É. Tardos (2006). *Algorithm Design*. Cambridge, MA: Addison-Wesley.
- P. A. M. Kommers, D. H. Jonassen y T. M. Mayes (eds.) (1992). *Cognitive Tools for Learning*, Berlín: Springer-Verlag.
- D. Laurillard (2012). *Teaching as a Design Science*. Nueva York, NY: Routledge.
- F. Marton y S. A. Booth (1997). *Learning and Awareness*. Mahwah, NJ: Lawrence Erlbaum.
- D. Norman (1983). "Some observations on mental models". En *Mental Models*, D. Gentner y A. Stevens (eds.), Hillsdale, NJ: Erlbaum, pp. 7-14.
- K. D. Powers y D.T. Powers (1999). "Making sense of teaching methods in computer science". *29th ASEE/IEEE Frontiers in Education Conference – Conference Proceedings*, Stipes Publishing, pp. 11b3 30-35.
- J. Á. Velázquez-Iturbide (2013a). "Identification and removal of misconceptions about optimization concepts underlying greedy algorithms", *Journal of Research and Practice in Information Technology*, 45(3/4):203-217.
- J. Á. Velázquez-Iturbide (2013b). "An experimental method for the active learning of greedy algorithms", *ACM Transactions on Computing Education*, 13(4):artículo 18.
- J. Á. Velázquez-Iturbide, O. Debdi, N. Esteban-Sánchez y C. Pizarro (2013a). "GreedEx: A visualization tool for experimentation and discovery learning of greedy algorithms". *IEEE Transactions on Learning Technologies*, 6(2):130-143.
- J. Á. Velázquez-Iturbide, C. Pareja-Flores, O. Debdi y M. Paredes-Velasco (2012). "Interactive experimentation with algorithms". En *Computers in Education – Volume 2*, Sergei Abramovich (ed.), Nova Science Publishers, pp. 47-70.
- J. Á. Velázquez-Iturbide, A. Pérez-Carrasco y O. Debdi (2013b). "Experiences in usability evaluation of educational programming tools". En *Student Usability in Educational Software and Games: Improving Experiences*, Carina González (ed.), Hershey, PA: IGI Global, pp. 241-260.
- L. E. Winslow (1996). "Programming pedagogy – A psychological overview". *SIGCSE Bulletin*, 28(3):17-22 y 25.