

J. Ángel Velázquez Iturbide

**Una Tercera Evaluación Cualitativa
de la Comprensión de la
Optimalidad**

Número 2016-01

Serie de Informes Técnicos DLSI1-URJC

ISSN 1988-8074

**Grupo Docente de Lenguajes y Sistemas Informáticos I
Universidad Rey Juan Carlos**

Índice

1	Introducción.....	1
2	Protocolo y Análisis.....	2
3	Resultados de la Práctica 2.....	3
3.1	Resultados de la Experimentación.....	18
3.2	Otras Dificultades y Malentendidos, y Terminología.....	22
3.3	Otros Comentarios.....	23
4	Resultados de las Prácticas 3b y 5b.....	24
4.1	Resultados de la Práctica 3b.....	24
4.2	Resultados de la Práctica 5b.....	26
5	Evolución de Resultados.....	28
6	Resumen de Resultados.....	31
7	Conclusiones.....	33
	Agradecimientos.....	33
	Referencias.....	33
	Apéndice A: Enunciado de la Práctica 2.....	35
	Apéndice B: Enunciado de la Práctica 3b.....	38
	Apéndice C: Enunciado de la Práctica 5b.....	40

Una Tercera Evaluación Cualitativa de la Comprensión de la Optimalidad

J. Ángel Velázquez Iturbide

Grupo Docente de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos,
C/ Tulipán s/n, 28933, Móstoles, Madrid
angel.velazquez@urjc.es

Resumen. OptimEx es un sistema para la experimentación interactiva con algoritmos de optimización. Este informe presenta un análisis de la comprensión de conceptos de optimización con OptimEx durante una experiencia de uso docente continuada realizada en otoño de 2015. Se incluyen los enunciados de prácticas usados, el método de análisis, los resultados detallados y comentados, así como una discusión de los mismos. El rendimiento de los alumnos mejora con respecto a la experiencia de los dos cursos anteriores.

Palabras clave: Algoritmos de optimización, experimentación interactiva, evaluación cualitativa, análisis de documentos.

1 Introducción

OptimEx es el acrónimo de “OPTIMization EXperimentation” y da nombre a un sistema interactivo para la experimentación con algoritmos de optimización. El objetivo genérico de OptimEx es permitir al alumno experimentar con distintos algoritmos para un mismo problema de optimización. El alumno debe comparar distintos algoritmos, determinando su optimalidad o suboptimalidad. En este último caso, el alumno podrá determinar su desviación de la solución óptima, tanto en porcentaje de casos como en porcentaje del valor de la solución óptima.

Muchas características de OptimEx [1] están inspiradas en GreedEx [2]. Conviene usar ambos sistemas de forma combinada, primero GreedEx como introducción a los algoritmos voraces (y, en menor medida, a otras técnicas de diseño de algoritmos de optimización), después OptimEx como sistema para la experimentación con algoritmos aproximados y quizá con técnicas de diseño exactas.

En cursos pasados se realizaron una evaluación de usabilidad [3] y dos evaluaciones cualitativas [4][5] de los informes presentados por los alumnos para una práctica desarrollada con OptimEx. Al igual que GreedEx [6][7], el uso de OptimEx reveló numerosas dificultades, que aconsejaron modificar la práctica docente. En este informe se presenta una tercera evaluación cualitativa realizada en un plazo de tiempo mayor mediante tres prácticas.

La estructura del informe es la siguiente. El apartado 2 describe el protocolo y método de análisis utilizados. El apartado 3 presenta en detalle los resultados

obtenidos con la primera práctica y el apartado 4, resultados menos detallados de las prácticas segunda y tercera. El quinto apartado analiza la evolución de los resultados de las tres prácticas. Los apartados 6 y 7 contienen, respectivamente, un resumen de resultados y unas conclusiones. Finalmente, tres apéndices contienen el enunciado de las prácticas.

2 Protocolo y Análisis

Esta evaluación se realizó en los dos grupos (presencial y online) de la asignatura optativa “Algoritmos Avanzados”, de cuarto curso del Grado en Ingeniería Informática, en octubre-diciembre del curso académico 2015-16.

El temario de la asignatura incluía varias técnicas algorítmicas para la resolución de problemas de optimización: técnica voraz, algoritmos heurísticos y aproximados, vuelta atrás, ramificación y poda, y programación dinámica. Se proponía la realización de 5 prácticas:

1. Técnica voraz. Dado el problema de selección de actividades y pseudocódigo para su resolución voraz, se identifican dos funciones de selección óptima: orden creciente de instantes de fin (en adelante, representado $F\uparrow$) y orden decreciente de instantes de inicio (en adelante, representado $C\downarrow$). Se pide completar el algoritmo y ampliarlo para que no dependa de que las actividades estén ordenadas.
2. Algoritmos heurísticos. Se plantea el problema del plan de sedes de coste mínimo y se esbozan dos algoritmos heurísticos. Se pide desarrollarlos, comprobar su optimidad con OptimEx y justificar su optimidad o suboptimidad (en el primer caso, con un razonamiento intuitivo y en el segundo, con un contraejemplo de su optimidad).
3. Técnicas de búsqueda. En realidad, la práctica se realizó desglosada en dos:
 - a. Dado el mismo problema del plan de sedes de coste mínimo, se pide desarrollar un algoritmo de vuelta atrás y otro de ramificación y poda.
 - b. Se pide comprobar la optimidad de estos dos algoritmos y los dos algoritmos heurísticos de la práctica 2.
4. Eliminación de recursividad múltiple redundante. Se da una función recursiva f y se pide analizar su redundancia y eliminarla mediante memorización y tabulación.
5. Técnica de programación dinámica. De nuevo, se plantea desglosada:
 - a. Dado el mismo problema del plan de sedes de coste mínimo, se pide diseñar un algoritmo recursivo y convertirlo en un algoritmo de programación dinámica (tabulado).
 - b. Se pide comprobar la optimidad de estos dos algoritmos y los cuatro algoritmos de la práctica 3b.

Obsérvese que la experimentación con optimidad se realizó en las prácticas 2, 3b y 5b. Se permitió realizar las prácticas 2 y 3b tanto individualmente como en parejas, pero la práctica 5b debía realizarse individualmente (por razones ajenas a esta evaluación).

A una hora y día convenidos, el enunciado de cada práctica estaba disponible en el campus virtual para que los alumnos se lo descargaran. Desde la práctica 2, también tenían disponible OptimEx y una pequeña guía de usuario. Los alumnos del grupo presencial asistían a una sesión de prácticas de 2 horas de duración. Los alumnos disponían de una semana de plazo para entregar el informe de cada práctica por medio del campus virtual.

Todos los informes debían incluir:

- Resultados: identificación de los algoritmos “óptimos”, si existían.
- Evidencias de los resultados de la experimentación: las tablas histórica y de resumen, comentadas.

La práctica 2 también pedía codificar los dos algoritmos heurísticos bosquejados, así como indicar y justificar la optimalidad o suboptimalidad de cada algoritmo. Asimismo, las prácticas 3b y 5b pedían identificar los algoritmos comparados y comentar las incidencias surgidas (por ejemplo, modificación de algún algoritmo).

Tras la corrección de cada práctica y el envío de comentarios por parte del profesor, se les permitía corregir errores y reenviar la práctica en el plazo de varios días.

Incluimos los enunciados de las tres prácticas como apéndices de este informe.

Los datos se analizaron utilizando los resultados de la evaluación de los dos cursos anteriores [4][5]. Se construyó una tabla donde aparecieran los elementos que se consideraron importantes para este análisis. Se realizaron dos rondas de análisis:

1. Ronda primera. Se construyó una tabla parecida a la elaborada en la evaluación anterior, con algunas diferencias menores derivadas del distinto planteamiento de las prácticas.
2. Rondas posteriores. Hubo que revisar diversos resultados de la primera ronda al escribir el informe, sobre todo para comprobar su completitud.

Conviene comentar que cada ronda no fue secuencial sino que implicó numerosas consultas a los informes y modificaciones de las tablas construidas.

3 Resultados de la Práctica 2

Se recogieron un total de 37 informes en el primer plazo de entrega: 34 del grupo presencial (23 individuales y 11 parejas) y 3 del grupo on-line (todos individuales). Solamente se presentaron 12 informes en el segundo plazo: 10 del grupo presencial (6 individuales y 4 parejas) y 2 del grupo on-line. Por último, se recibió una tercera entrega del grupo presencial. Por tanto, el número total de informes recogidos es 50.

Presentamos los resultados del cuestionario en la Tabla 1. La tabla incluye una fila por cada informe entregado. Cada columna tiene el siguiente significado:

- “Grupo”. Asigna un número a cada grupo, indicando también si es individual “I” o una pareja “P”.

- “Algoritmos”. Identifica los algoritmos aportados por el grupo. El enunciado sólo pedía dos algoritmos heurísticos que, por brevedad, los llamamos “voraz 1” y “voraz 2”. Sin embargo, varios grupos aportaron más algoritmos.
- “Condiciones experimento”. Da alguna información específica sobre el experimento realizado por el grupo. Al menos se incluye el número de datos de entrada utilizados. Otras informaciones son: si algún algoritmo estaba mal, si la experimentación se realizó marcando algún algoritmo como óptimo, los valores proporcionados en el diálogo de generación de datos aleatorios, etc.
- “Propuesto óptimo”. Si los alumnos han propuesto algún algoritmo como óptimo.
- “Resultados”. Se indica qué algoritmos son óptimos dadas las condiciones del experimento realizado.
- “¿Bien?”. Utilizamos el símbolo ✓ para indicar que el grupo ha realizado bien la experimentación, tanto los algoritmos usados, como el procedimiento y las conclusiones.
- “Dificultades y malentendidos”. Explicación mía de los problemas o malentendidos que parecen tener.
- “Contraej.”. Indicación de si han aportado contraejemplos de la optimidad de los dos algoritmos heurísticos.
- “Terminología”. Uso de la terminología relacionada con la optimización.
- “Comentarios abiertos”. Comentarios realizados por los grupos en el apartado de conclusiones del informe.

Tabla 1. Análisis de los informes de la práctica 2

Grupo	Algoritmos	Condiciones experimento	Propuesto óptimo	Resultados	Evidencias	¿Bien?	Dificultades o malentendidos	Contraej.	Terminología	Comentarios abiertos
1. I	Voraz 1 Voraz 2 Algoritmo “óptimo” (dos bucles anidados)	– Voraces mal – Marcado algoritmo 3 – 50	nº 3	Voraz 2, superóptimo	Tabla histórica		– Desde el principio asume sin duda que el algoritmo 3 es óptimo y lo toma como referencia	– Contraej. del 2 (long. 4) – Voraz 1 va a tener “un resultado menor, no el esperado”	– Habla de “pesos” para el coste del traslado – Algoritmo exacto u óptimo, solución mejor, mejor o más próxima	
	Voraz 2 Algoritmo “óptimo”	– 30 – Maximiza	nº. 3	nº. 3 (maximizar)	Tabla histórica Tabla de resumen		– No observa que ha invertido la f. objetivo – Tablas contradictorias entre sí	– ¡Igual! (no hay alg. 1)	– Igual	
2. I	Voraz 1 Voraz 2	– 100	Ninguno	Ninguno óptimo	Tabla histórica (parte) Tabla de resumen	✓		– 4 contraej. con 4 situaciones (generados por OptimEx, tablas de resultados, long. 5)	– Exacto, óptimo, más óptimo	– Sugiere el diseño de algoritmo de vuelta atrás o de fuerza bruta – Práctica sencilla, pero con ciertas complicaciones del problema

3. I	Voraz 1 Voraz 2	<ul style="list-style-type: none"> - 3x20 - Muestra datos de entrada - 3 experimentos: sin marcar, marcando voraz 1, y voraz 2 	Voraz 2	Ninguno Resultados superópts.	3 tablas histórica 3 tablas de resumen	<ul style="list-style-type: none"> - Asume el caso en que cada algoritmo fuera óptimo, pero no detecta el problema de resultados superóptimos 	<ul style="list-style-type: none"> - Sin contraej. 	<ul style="list-style-type: none"> - Óptimo, más óptimo - Por su uso del lenguaje, afirma que ninguno es óptimo y también que voraz 2 lo es 	
	Voraz 1 Voraz 2	<ul style="list-style-type: none"> - 100 	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	<ul style="list-style-type: none"> - Contraej. de los dos (generados por OptimEx, tablas de resultados, long. 12) 	<ul style="list-style-type: none"> - Exacto, óptimo 	<ul style="list-style-type: none"> - Los heurísticos se usan cuando no hay solución óptima - Argumentos mal expresados
4. P	Voraz 1 Voraz 2	<ul style="list-style-type: none"> - Ambos mal (f. objetivo) - Muestra diálogo generación - 90 	Voraz 1	<ul style="list-style-type: none"> - Resultado ej. del enunciado - V oraz 1 	Tabla histórica (parte) Tabla de resumen	<ul style="list-style-type: none"> - Aunque comprueba que voraz 1 es óptimo, su conclusión se limita a voraz 2 	<ul style="list-style-type: none"> - Contraej. del 2 (generados por OptimEx, long. 5) 	<ul style="list-style-type: none"> - Óptimo al 100%, óptimo 	<ul style="list-style-type: none"> - Técnica poco útil - Práctica útil - Sugiere el diseño de algoritmo de vuelta atrás
	Voraz 1 Voraz 2	<ul style="list-style-type: none"> - Algoritmos arreglados - Muestra diálogo generación - 38 	Ninguno	Ninguno	Tabla histórica Tabla de resumen	✓	<ul style="list-style-type: none"> - Contraej. de los dos (generados por OptimEx, long. 5) 	<ul style="list-style-type: none"> - Más óptimo, óptimo al 100%, óptimo en el 76'32% de las ejecuciones 	<ul style="list-style-type: none"> - Práctica útil - Poco tiempo de realización - Mejorar enunciado

5. I	Voraz 1 Voraz 2	– 5.000 – Muestra diálogo generación	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	– Suma porcentajes por filas	– Contraejs. de los dos – Generados por Optimex (tablas de resultados, long. 10) – Diseñados manualmente (long. 4)	– Óptimo	– Contraejs. complicado – Elogia OptimEx – Propone mejora de IA (búsqueda heurística)
6. I	Voraz 1 Voraz 2	– Voraz 1 mal – 1.000	Voraz 1	Voraz 1	Tabla histórica (completa, editada) Tabla de resumen		– Habla de los algoritmos como si fueran para problemas distintos: el 1 como un “problema relajado” del 2 – Sugiere el uso del primer algoritmo como “cota mínima” del segundo	– Contraej. del 2 (long. 2)	– Exacto, óptimo	
	Voraz 1 Voraz 2	– Voraz 1 revisado – 1.000	Ninguno	Ninguno	Tabla histórica (completa, editada) Tabla de resumen	✓	– Primer comentario igual	– Contraej. igual de los dos (long. 2)	– Igual	

7. P	Voraz 1 Voraz 2	- 100 - Muestra diálogo generación	Ninguno	Ninguno	Tabla histórica (parcial, editada) Tabla de resumen	✓	- Contraej. del 2 (long. 6, generado por OptimEx)	- Exacto, óptimo, porcentaje mayor	- Utilidad OptimEx
8. P	Voraz 1 Voraz 2	- Voraz 1 mal - 100	Voraz 1	Voraz 1	Tabla histórica (completa, editada) Tabla de resumen		- Contraej. del 2 (long. 3)	- Exacto, óptimo	- Insatisfacción por problemas con OptimEx al cargar clase
	Voraz 1 Voraz 2	- Voraz 1 revisado - 100	Ninguno	Ninguno	Tabla histórica (completa, editada) Tabla de resumen	✓	- Contraejs. del 1 y 2 (longs. 3 y 5)	- Igual	- Igual - Sugieren mejorar el enunciado con dos ejemplos
9. I	Voraz 1 Voraz 2	- 5.000	Ninguno	Ninguno	Tabla histórica (completa, editada) Tabla de resumen	✓	- Contraejs. para los dos (long. 4)	- Exacto, porcentaje de soluciones óptimas	- Utilidad y facilidad de uso de OptimEx, salvo al principio - "Largo" de copiar la tabla de resultados en el informe

10. I	Voraz 1 Voraz 2	– 273	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	– Contraej. del 2 (long. 5)	– Exacto, óptimo, algoritmo o resultado 100% óptimo	– Satisfacción con práctica – Da a entender que no hay algoritmo exacto para este problema
11. I	Voraz 1 Voraz 2	– Voraz 1 mal – 100 – Muestra diálogo generación	Voraz 1	Voraz 1	Tabla histórica (parte) Tabla de resumen	– Esperaba que voraz 2 fuera óptimo – “Los algoritmos con datos de precondition resultan más óptimos y efectivos que los que tienen más datos (coste f)”	– Sin contraej.	– Algoritmo exacto, resultado óptimo	
	Voraz 1 Voraz 2	– Voraz 1 revisado – 100 – Muestra diálogo generación	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	– Sugiere que es el segundo pero luego reconoce que ninguno es exacto	– Muestra filas de la tabla de resultados pero sin mostrar los datos asociados	– Exacto, óptimo al 100%, resultados óptimos o mejores – Sugiere diseñar algoritmo de vuelta atrás
12. P	Voraz 1 Voraz 2	– 847 (parece que los ha generado por tiempo)	Ninguno	Ninguno	Tabla de resumen	✓	– Contraej. del 1 (long. 2) – Contraej. del 2 (long. 3)	– Algoritmo óptimo, 100% de soluciones óptimas	– Sugiere diseñar algoritmo de vuelta atrás – Informa de problema con OptimEx

13. P	Voraz 1 Voraz 2	- 5.000	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Sin contraej.	- Algoritmo óptimo, 100% de ejecuciones óptimas	- Asocia esta técnica con iteración y la recursividad con vuelta atrás
	Igual	- Igual	Igual	Igual	Igual	Igual	- Contraej. del 2 (long. 5)	- Igual, "solución más válida"	- Igual
14. I	Voraz 1 Voraz 2	- 100, 500	Ninguno	Ninguno	17 tablas de resumen 3 gráficas propias	✓	- Contraej. del 2 (long. 5)	- Algoritmo exacto, porcentaje de soluciones óptimas	- Utilidad de OptimEx - Sugiere mejora de algoritmo heurístico
15. I	Voraz 1 Voraz 2	- 168	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Se remite a la tabla histórica para el 1 - Contraej. del 2 (long. 4)	- Algoritmo exacto u óptimo, porcentaje de soluciones óptimas	- Satisfacción con la práctica - Utilidad y facilidad de uso de OptimEx - Sugiere uso de otras técnicas y mejora de algoritmo heurístico
16. I	Voraz 1 Voraz 2	- 65	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. del 2 (long. 4, generado por OptimEx)	- Algoritmo óptimo, coste óptimo o mínimo	- Utilidad de OptimEx - Problema inicial con OptimEx

17. P	Voraz 1 Voraz 2	- 100	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraejs. para los dos (long. 4)	- Algoritmo óptimo, 100% de optimalidad, óptimo al 100%	- Utilidad de OptimEx
18. P	Voraz 1 Voraz 2	- Voraz 1 mal - 1.000	Voraz 1	Voraz 1	Tabla histórica (parte) Tabla de resumen	- Considera que los algoritmos resuelven problemas distintos	- Contraej. del 2	- Algoritmo exacto u optimo, ejecución óptima	- Satisfacción por el plazo de la práctica - Sugieren el uso de vuelta atrás - Dificultad en encontrar contraej. - Dificultad para entender la terminología de las evidencias
	Voraz 1 Voraz 2	- Voraz 1 revisado - 4x500	Ninguno	Ninguno	Tabla histórica (parte) 4 tablas de resumen	✓	- Contraej. para los dos (long. 3)	- Igual	- Igual los 2 primeros comentarios

19. P	Voraz 1 Voraz 2	- 2, 3	Voraz 2	Voraz 2	2 tablas de resumen	- 2 ó 3 ejecuciones son evidencia insuficiente	- Sin contraejs.	- "Solución óptima al 100%" (algoritmo)	- Práctica sencilla - Facilidad de uso de OptimEx, salvo al principio - Utilidad de su guía - Problemas con OptimEx (funcionamiento y error de compilación)
	Voraz 1 Voraz 2	- 400 - Voraz 2 marcado óptimo	Voraz 2	Voraz 1 superópt.	Tabla histórica (parte) Tabla de resumen	- A pesar de leer la guía, no han sabido interpretar la tabla	Igual	- Algoritmo óptimo, "100% de resultado" (de casos óptimos)	- Igual
	Voraz 1 Voraz 2	- 2.077 - Muestra diálogo generación - Probado con datos más pequeños y variando el coste del traslado	Voraz 2	Ninguno	Tabla histórica (parte) Tabla de resumen	- Conclusión incoherente con la tabla de resumen (sólo 94% de casos mejores)	- Contraej. del 1	- Porcentaje de optimalidad	- Tutoría - Explicación poco clara de la forma de probar datos - Cambios menores en el código

20. I	Voraz 1 Voraz 2	– 2×500 – Muestra 2 diálogos generación	Ninguno	Ninguno	2 tablas de resumen	✓	– Observa distintos % entre ambos dependiendo según coste del traslado	– Contraej. 1	– “Más óptimo”, “punto de optimalidad más alto”	– Utilidad de OptimEx
21. I	Voraz 1 Voraz 2	– 10.000	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓		– Contraej. los dos (long. 2)	– Algoritmo óptimo, “óptimo al 100%”	
22. I	Voraz 1 Voraz 2	– 100	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓		– Contraej. 1 y 2 (long. 2)	– Algoritmo exacto, solución óptima	– Utilidad de la práctica y OptimEx
23. I	Voraz 1 Voraz 2	– 70	Ninguno	Ninguno	Tabla histórica Tabla de resumen	✓		– Contraej. 1 y 2 (long. 4)	– Algoritmo exacto u óptimo, resultados óptimos	– Utilidad de OptimEx – Dificultad de diseñar contraejs.
24. I	Voraz 1 Voraz 2	– 1.000, 2.500, 4.600 – Muestra 3 diálogos generación	Ninguno	Ninguno	3 tablas de resumen	✓		– Contraej. 1 (long. 2)	– “Rango de optimalidad”, “porcentaje mayor de optimalidad”	– Elogio de OptimEx

25. I	Voraz 1 Voraz 2	- 81	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. 1 y 2 (long. 4) - Entiende "conrej." al revés, al ser el problema de minimización: caso donde el algoritmo devuelve un valor menor (subóptimo) que el otro	- Exacto	- Utilidad de OptimEx, a pesar de problemas iniciales - Práctica sencilla
26. I	Voraz 1 Voraz 2	- 108	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. de los dos (long. 4)	- Algoritmo exacto, resultado óptimo	- Utilidad de la práctica y OptimEx
27. I	Voraz 1 Voraz 2	- 917 - Muestra diálogo generación	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. 1 y 2 (long. 3)	- Algoritmo exacto, algoritmo o resultado "más óptimo"	- Sorpresa por el tipo de problemas y de práctica - Utilidad de OptimEx

28. P	Voraz 1 Voraz 2	– 22 (1 seg.) – Muestra diálogo de generación, resto de diálogos y datos generados	Ninguno	Ninguno	Tabla histórica Tabla de resumen	✓	– Contraej. 1 (long. 4)	– Exacto, “90,91% óptimo”, “más ‘subóptimo’”, “óptimo al 100%”	– Parece que ha generado datos con diversas variaciones – Utilidad de OptimEx
	Igual	Igual	Igual	Igual	Igual	Igual	– Contraej. 2	– Igual	– Igual
29. P	Voraz 1 Voraz 2	– 1.000	Ninguno	Ninguno	Tabla de resumen	✓	– Sin contraejs.	– “100% del valor óptimo”	– Utilidad práctica y OptimEx
30. I	Voraz 1 Voraz 2	– 5.127 – Muestra diálogo generación	Ninguno	Ninguno	Tabla de resumen	✓	– Contraej. de 1 y 2 (long. 3)	– Algoritmo o solución óptima, algoritmo “más eficiente”	– Aporta un algoritmo heurístico mejorado (código y resultados)
31. I	Voraz 1 Voraz 2	– Voraz 1 mal – 921	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓		– Algoritmo exacto, valor óptimo	
32. I	Voraz 1 Voraz 2	– 100	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	– Contraej. de 1 y 2 (long. 2)	– Algoritmo exacto, resultado óptimo	– Utilidad de OptimEx – Práctica sencilla

33. I	Voraz 1 Voraz 2	- 100	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. de 1 y 2 (long. 2)	- Algoritmo exacto, valor óptimo	- Utilidad de la práctica y OptimEx - Sugiere desarrollar algoritmo de vuelta atrás
34. I	Voraz 1 Voraz 2 Otros 6 voraces Vuelta atrás	- 10, 6x200, 1.200 - Especifica rangos de generación	Vuelta atrás	Vuelta atrás	1 tabla histórica (completa, editada) 6 tablas de resumen 4 gráficas propias de 2 formatos	✓	- Contraejs. de todos los voraces (longs. 2, 3 y 4)	- Algoritmo óptimo	- Utilidad de OptimEx - Práctica sencilla - Sugiere mejora del algoritmo
35. P	Voraz 1 Voraz 2	- 237	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓	- Contraej. 1 no lo dan por obvio - Contraej. 2 (long. 10)	- "Óptimo al 100%", solución óptima	- Satisfacción con OptimEx, con problemas iniciales
36. I	Voraz 1 Voraz 2	- Voraz 1 mal - 1.000	Voraz 1	Voraz 1	Tabla histórica (parte) Tabla de resumen	- Incoherencia de conclusión con resultados: "Aunque los algoritmos desarrollados no son óptimos queda clara la diferencia entre los algoritmos aproximados y los exactos."	- Sin contraejs.	- Algoritmo óptimo	

	Voraz 1 Voraz 2	– Voraz 1 revisado – Voraz 2 = 1 (será error) – 1.000	Ninguno	Ninguno	Tabla de resumen	✓	– Misma conclusión con resultados correctos	– Contraej. para los dos (long. 3)	– Algoritmo exacto, resultado óptimo	
37. I	Voraz 1 Voraz 2	– Voraz 1 mal – 1.198	Voraz 1	Voraz 1	Tabla de resumen			– Contraej. 2 (long. 2)	– Algoritmo exacto, solución óptima	– Satisfacción con OptimEx – Sugiere mejora de voraz 2
	Voraz 1 Voraz 2	– Voraz 1 revisado – 1.007	Ninguno	Ninguno	Tabla histórica (parte) Tabla de resumen	✓		– Contraej.1 (long. 5) – Contraej.2 (long. 2)	– Igual	– Satisfacción con OptimEx – Aporta otros 3 algoritmos (código) – Sugiere uso de vuelta atrás

En el resto del informe, utilizamos la notación G_{nn} para referirnos al grupo con número identificativo nn y la notación G_{nn-m} para referirnos a la entrega m -ésima del grupo G_{nn} . Solamente distinguimos las características de las distintas entregas cuando la parte correspondiente ha variado algo (p.ej. no se considera variación de la parte experimental si sólo han completado la parte de contraejemplos pero sí si han variado los algoritmos presentados).

3.1 Resultados de la Experimentación

Recordemos que se recogieron 50 informes (37 informes en la primera entrega, 12 de la segunda y 1 entrega tercera). Si se distinguen todas las entregas en las que varían los experimentos, se han recogido 48 informes.

Veamos primero los resultados globales para la primera entrega (Tabla 2).

Tabla 2. Resultados de la primera entrega de la práctica 2 (N=37)

Resultado	# (%) grupos	Grupos
Bien	27 (73%)	G02, G05, G07, G09, G10, G12, G13, G14, G15, G16, G17, G20, G21, G22, G23, G24, G25, G26, G27, G28, G29, G30, G31, G32, G33, G34, G35
Mal	10 (27%)	G01, G03, G04, G06, G08, G11, G18, G19, G36, G37

Es intrigante el grupo G05, que realizó bien su práctica pero sumaba los porcentajes de optimidad por filas en lugar de columnas.

Veamos las razones de que los resultados de algunos grupos sean incorrectos:

- El caso más frecuente es haber diseñado mal el algoritmo primero (G06, G08, G11, G18, G36, G37) o incluso los dos (G01, G04). El error de estos grupos procede de confundir la función objetivo (del problema) y la función de selección (del algoritmo heurístico). La consecuencia de esta confusión es que el algoritmo primero obtiene un 100% de resultados óptimos. Estos grupos hacen varias propuestas de algoritmo óptimo:
 - El algoritmo primero (G06, G08, G11, G18, G36, G37).
 - Ningún algoritmo (G04).
- Se proporciona un algoritmo “óptimo”, sin más justificación (G01; además, sus algoritmos heurísticos están mal). Este experimento produce resultados superóptimos del algoritmo heurístico 2.
- Se obtiene que ninguno de los algoritmos es exacto pero se propone el segundo porque obtiene resultados mejores en un porcentaje mayor de casos (G03). También se realiza el experimento marcando como óptimo sucesivamente a uno y a otro algoritmo sin mencionar que resultan resultados superóptimos.
- Los algoritmos son correctos pero realizan un número muy pequeño de ejecuciones, concluyendo que es óptimo el algoritmo segundo (G19).

La confusión antes señalada entre problema y algoritmo destaca especialmente en algunos grupos:

- Un grupo identifica el primer algoritmo voraz como óptimo pero su conclusión sobre la suboptimalidad de los algoritmos heurísticos se limita al algoritmo 2 (G04).
- Varios grupos expresan, a veces de forma confusa, que no resuelven el mismo problema:
 - “Ya que el primer algoritmo tiene una restricción menos, este caso no existe el coste de traslación, y sería un problema relajado” (G06). Este grupo aporta una sugerencia potencialmente constructiva: en su primer informe “El primer algoritmo puede servir como cota mínima del segundo algoritmo”.
 - “Los algoritmos con datos de precondition resultan más óptimos y efectivos que los que tienen más datos (coste f)” (G11).
 - “Por ello, no nos parece del todo lógico comparar la optimalidad de estos 2 algoritmos entre sí puesto que, realmente, no resuelven exactamente el mismo problema pero aun así vamos a hacer esta comparativa ya que es lo que se nos ha pedido”. “Como ya hemos explicado anteriormente, la introducción del coste de traslado cambia completamente el problema ya que para obtener un resultado óptimo para cualquier caso es necesario el uso de algoritmos más complejos. Es por eso que no encontramos el sentido de tener que comparar ambos algoritmos cuando son en gran parte distintos” (G18).
 - “El primer algoritmo según OptimEx es óptimo en el 100% de los casos, pero como hemos dicho antes el resultado de este no es realista al no incluir el coste de traslado de sede” (G36).

Los 10 grupos que realizaron mal la práctica la repitieron. El resultado es que:

- 8 grupos han corregido los problemas señalados (G03, G04, G06, G08, G11, G18, G36, G37). En 6 casos revisaron el primer algoritmo, en otro caso los dos (G04) y en otro, la interpretación de los resultados (G03).
- 2 grupos no han corregido los problemas señalados (G01, G19). El grupo G-19 realizó dos entregas, ambas fallidas.

Es interesante notar que estos dos grupos no interpretaron bien la información contenida en sus tablas:

- G01-2 aportaba evidencias contradictorias: mientras la tabla histórica resaltaba el algoritmo segundo como óptimo, la tabla de resumen daba el 100% de resultados óptimos al algoritmo primero. Además, no observó que los resultados de la tabla histórica correspondían a una función objetivo de maximización.
- G19-2 marcó el algoritmo segundo como óptimo pero no advirtió que el algoritmo primero producía resultados superóptimos.

Hay que añadir la primera entrega del grupo G03, que probó con distintas situaciones de marcado de algoritmos óptimos, sin advertir los resultados superóptimos obtenidos. Estos tres grupos son los únicos que han marcado algún algoritmo como óptimo.

Veamos ahora algunas características de la experimentación (véase Tabla 3). Marcamos en cursiva las entregas que están mal.

Tabla 3. Números de datos de entrada usados en la práctica 2 (N=48)

# datos entrada	# (%) grupos	Grupos
2	1 (2%)	<i>G19-1</i>
3	1 (2%)	<i>G19-1</i>
10	1 (2%)	G34
20	1 (2%)	<i>G03-1</i>
22	1 (2%)	G28
30	1 (2%)	<i>G01-2</i>
38	1 (2%)	G04-2
50	1 (2%)	<i>G01-1</i>
65	1 (2%)	G16
70	1 (2%)	G23
81	1 (2%)	G25
90	1 (2%)	<i>G04-1</i>
100	12 (25%)	G02, G03-2, G07, <i>G08-1</i> , G08-2, <i>G11-1</i> , G11-2, G14, G17, G22, G32, G33
108	1 (2%)	G26
200	1 (2%)	G34
237	1 (2%)	G35
273	1 (2%)	G10
400	1 (2%)	<i>G19-2</i>
500	3 (2%)	G14, G18-2, G20
847	1 (2%)	G12
917	1 (2%)	G27
921	1 (2%)	G31
1.000	7 (15%)	<i>G06-1</i> , G06-2, <i>G18-1</i> , G24, G29, <i>G36-1</i> , G36-2
1.007	1 (2%)	G37-2
1.198	1 (2%)	<i>G37-1</i>
1.200	1 (2%)	G34
2.077	1 (2%)	<i>G19-3</i>
2.500	1 (2%)	G24
4.600	1 (2%)	G24
5.000	3 (2%)	G05, G09, G13
5.127	1 (2%)	G30
10.000	1 (2%)	G21

Puede observarse que los únicos números que han sido repetidos por más de un grupo son: 100, 500, 1.000 y 5.000, siendo 1.000 y sobre todo 100 los más usados. Estas cuatro decisiones suman el 52% de las entregas. Bastantes números son sorprendentes por su falta de sencillez. En algunos casos podría deberse a la ejecución repetida y no planificada de los algoritmos. En otros grupos parece haber evidencia en sus informes de haber generado datos por tiempo (G12, G28).

Algunos grupos aportan tablas correspondientes a varias ejecuciones. Sus objetivos son variados:

- Un grupo realiza 3 experimentos, variando el marcado del algoritmo óptimo: ninguno, algoritmo primero y algoritmo segundo (G03-1).
- Un grupo ha realizado varias pruebas con pocos casos y aportan dos tablas con 2 y 3 ejecuciones (G19-1).
- Cuatro grupos realizan varios experimentos variando el valor máximo del coste del traslado (en el rango de generación de datos aleatorios) (G18-2, G20, G24) o su valor (siendo constante al generar el resto de costes) (G34). Otro grupo presenta 16 variaciones del número de meses y del rango de los costes mensuales y de traslado (G14).

Otro grupo (G28) comenta que ha realizado varias ejecuciones, aunque sólo aporta resultados de una.

En cuanto a las evidencias presentadas, se presentan resumidas en la Tabla 4.

Tabla 4. Evidencias presentadas en la práctica 2 (N=48)

Evidencia	# (%) informes
Tabla de resumen	47 (98%)
Tabla histórica completa	12 (25%)
Tabla histórica parcial	27 (56%)
Gráficas propias	2 (4%)

Las tablas históricas completas se han presentado en la mayor parte de los casos con un número de datos relativamente bajo, aunque se da un caso de 1.000 datos y otro de 5.000. En 8 informes (17%, G12, G19-1, G20, G24, G29, G30, G36-2, G37-1) sólo se han incluido tablas de resumen, mientras que solamente el grupo (2%, G01-1) no ha presentado tabla de resumen. La mayor parte de las tablas históricas se han obtenido mediante capturas de pantalla, aunque en 7 casos se ha exportado a Excel e incluido en el informe (6 tablas completas y 1 parcial). Las gráficas propias han sido incluidas en informes muy completos, que presentan más resultados de los pedidos (G14, G34).

Por último, en 12 informes se han presentado los diálogos de generación de datos (véase Tabla 5), lo cual permite conocer mejor las condiciones del experimento. Asimismo, un grupo (G34) ha especificado los rangos de generación.

Tabla 5. Informes de la práctica 2 con el diálogo de generación aleatoria de datos

Resultado	Grupos
Bien	G04-2, G05, G07, G11-2, G20, G24, G27, G28, G30
Mal	G04-1, G11-1, G19-3

Dos grupos (G03-1, G28) muestran los datos usados. Este último grupo también muestra todos los diálogos realizados en su experimento.

3.2 Otras Dificultades y Malentendidos, y Terminología

Hay varias fuentes de dificultad en la práctica. Ya hemos comentado en la subsección anterior algunas relacionadas con la propia experimentación. Veamos aquí las relativas a los propios algoritmos heurísticos y al uso de la palabra “óptimo”. Dejamos sin analizar el apartado de contraejemplos de la práctica, ya que no está relacionado directamente con la experimentación.

Veamos primero algunas afirmaciones incorrectas:

- Utilización de terminología propia de otros problemas, como “peso” para el coste de traslado (G01).
- Los algoritmos heurísticos se usan cuando no hay algoritmo exacto (G03-2, G10).
- Técnica poco útil, al no producir resultados óptimos (G04-1).
- Asociación de las técnicas con características que no son exclusivas de ellas: “No obstante este algoritmo no nos muestra en el 100 % de los casos la solución más óptima, dado que es un método iterativo, si se hiciera de forma recursiva mediante *backtracking* su efectividad podría llegar a ser total” (G13).
- Argumentación confusa: “Aunque los algoritmos desarrollados no son óptimos queda clara la diferencia entre los algoritmos aproximados y los exactos” (G36).

Conocemos de evaluaciones anteriores [4][5] el frecuente mal uso del término “óptimo”, diciendo que un algoritmo es “más óptimo” o “menos óptimo”. Esta expresión se utiliza como sinónimo de “con mayor porcentaje de resultados óptimos” o simplemente de “mejor”.

La Tabla 6 muestra dos malos usos del término óptimo referidos a un algoritmo: “más óptimo” y “óptimo a 100%”. Este segundo uso a veces toma formas peculiares: “solución óptima al 100%” (G19-1).

Tabla 6. Malos usos del término “óptimo” en la práctica 2

Resultado	“Más óptimo”	“Óptimo al 100%”
Bien	G02, G20, G24, G27, G35	G04-1, G10, G11-2, G17, G21, G28
Mal	G03-1, G04-2	G04-2, G19-1

Este mal uso es menos frecuente referido al resultado de una ejecución, aunque también aparece ocasionalmente: “resultado 100% óptimo” (G10), “solución más válida” (G13-2), “resultado más óptimo” (G27).

Incluso en algunos casos en que se usa una terminología correcta, ésta se encuentra mal expresada, debiendo entenderse a partir del contexto. Así, para referirse a la fila de resultados óptimos: “100% de resultados” (G19-2), “rango de optimalidad” y “porcentaje mayor de optimalidad” (G24), “100% del valor óptimo” (G29), “algoritmo más eficiente” (G30).

3.3 Otros Comentarios

El enunciado de la práctica contemplaba que el alumno hiciera comentarios abiertos. Algunos comentarios se han tenido en cuenta en el análisis realizado en las dos subsecciones anteriores. Otros son triviales, ya que simplemente describen el trabajo realizado. Los comentarios restantes pueden agruparse en cuatro categorías: sugerencias de mejores algoritmos, comentarios sobre la práctica, valoraciones de OptimEx, y otros.

La Tabla 7 muestra la variedad de sugerencias realizadas sobre mejores algoritmos.

Tabla 7. Propuestas de mejoras de los algoritmos de la práctica 2

Algoritmo propuesto	Grupos
Algoritmos heurísticos	G14, G15, G30, G37-1, G37-2
Considerar meses anteriores	G34
Vuelta atrás	G02, G04-1, G11-2, G12, G18-1, G18-2, G33, G37-2
Otras técnicas exactas	G15
Fuerza bruta	G02
Algoritmo de IA	G05

En total se han recogido 17 propuestas (35% de las entregas). La mayor parte de las sugerencias proponían desarrollar algoritmos de vuelta atrás (8 grupos, 17%) o diversas mejoras a realizar sobre los algoritmos heurísticos del enunciado, normalmente sobre el segundo (4 grupos, 8%). Se ha propuesto un algoritmo de inteligencia artificial, la búsqueda heurística. Tres grupos aportan algoritmos nuevos en la propia práctica (G34) o en las conclusiones (G30, G37); los dos primeros también aportan los resultados de experimentar con ellos.

También se recogieron 16 comentarios sobre las prácticas (1/3 de las entregas, véase Tabla 8).

Tabla 8. Comentarios sobre la práctica 2 (N=48)

Comentarios	# (%) grupos	Grupos
Práctica sencilla	5 (10%)	G02, G19-1, G25, G32, G34
Práctica útil	9 (19%)	G04-1, G04-2, G10, G15, G22, G26, G27, G29, G33
Mejoras	2 (4%)	G04-2, G08-2

A pesar de considerar sencilla la práctica, G02 comenta sus dificultades para codificar los algoritmos esbozados. Un grupo comenta su sorpresa por el tipo de práctica, donde no hay una solución única (G27). Las sugerencias de mejora piden “anunciados más explícitos” (G04-2, probablemente se refieren a una explicación más detallada que incluya ejemplos) o añadir otro ejemplo adicional (G08-2).

Por último, se recibieron 31 comentarios sobre OptimEx (65% de las entregas, véase Tabla 9).

Tabla 9. Comentarios sobre OptimEx en la práctica 2

Comentarios	# (%) grupos	Grupos
Utilidad	19 (40%)	G05, G07, G09, G14, G15, G16, G17, G20, G22, G23, G24, G25, G26, G27, G28, G29, G32, G33, G34
Facilidad de uso	4 (8%)	G09, G15, G19-1, G35
Dificultades encontradas	8 (17%)	G08-1, G08-2, G09, G12, G16, G18-1, G19-1, G35

Entre los grupos que comentan la utilidad de OptimEx, pueden destacarse dos elogios: “Lo mejor de la práctica ha sido conocer el programa OptimEx ya que es bastante útil, no solo para esta asignatura y el ámbito académico, sino para cualquier situación en la que queramos comprobar resultados de distintos programas.” (G05), “Utilizaré esta herramienta más a menudo, ya que para esta asignatura y para el mundo laboral veo que es bastante útil” (G24). Otro grupo agradece la guía de uso de OptimEx, sobre todo por su explicación de errores (G19-1, G19-2, aunque no reconocieron sus propios errores).

Las dificultades de uso de OptimEx son de varias clases:

- La primera versión proporcionada no funcionaba bien (G16, G35).
- Imposibilidad de uso de OptimEx por no haber especificado la ruta (G12).
- Uso inicial (G09, G19-1).
- Problemas de compilación, ya que el sistema no reconocía la clase a cargar. Un grupo la copió sobre una clase proporcionada en la asignatura (G08-1, G08-2), mientras que otro eliminó una cláusula *package* (G19-1).
- Copiar la tabla de resultados en el informe (G09, con 5.000 datos).
- Familiaridad con la terminología necesaria para aportar evidencias (G18-1).

Un grupo criticó la obligatoriedad de utilizar OptimEx para realizar la práctica, dados los problemas de acabado que presenta (G08-1, G08-2).

4 Resultados de las Prácticas 3b y 5b

Las prácticas 3b y 5s eran continuación de la práctica 2. En cada una debían añadirse algunos algoritmos exactos y comprobar su optimidad: algoritmos de vuelta atrás y ramificación y poda en la práctica 3b, y las versiones recursiva e iterativa de un algoritmo de programación dinámica en la práctica 5b. En este apartado presentamos los resultados obtenidos. El análisis no es tan detallado como en la práctica 2, sino que se limita a comprobar si es correcto y a algunas otras cuestiones interesantes.

4.1 Resultados de la Práctica 3b

Se recogieron 35 informes (31 en la primera entrega y 4 en posteriores). La Tabla 10 muestra los resultados obtenidos con la primera entrega de cada grupo. Mantenemos la

misma numeración de grupos que en la práctica 2, para poder seguir mejor su evolución.

Tabla 10. Resultados de la primera entrega de la práctica 3b (N=31)

Resultado	# (%) grupos	Grupos
Bien	27 (87'1%)	G02, G04, G05, G06, G07, G08, G09, G10, G11, G12, G13, G17, G19, G20, G21, G22, G23, G24, G25, G26, G28, G30, G32, G33, G34, G36, G37
Mal	4 (12'9%)	G16, G27, G29, G38

Dos incidencias destacadas entre los grupos que realizaron bien la práctica son los siguientes:

- Seis grupos (19'3%) ya habían realizado la práctica 3b como parte final de la práctica 3a (G24 ¡sólo comparó cada algoritmo consigo mismo!; G02, G05, G30, G37; G34 además ¡modificó los algoritmos para contar el número de nodos y comparar los algoritmos con OptimEx!).
- Tres grupos (9'7%) incluyen más algoritmos de los pedidos (G05: 2 algoritmos de vuelta atrás; G30: 3 algoritmos heurísticos; G34, 8 algoritmos heurísticos, 2 de vuelta atrás y 2 de ramificación y poda).

Las entregas mal realizadas presentan las siguientes características:

- Sólo se comparan los algoritmos de búsqueda, bien cada uno consigo mismo (G38) bien entre sí (G16, G29).
- Se obtienen resultados superóptimos (G27).

Los cuatro grupos que realizaron la práctica mal en la primera entrega entregaron bien la segunda. Cabe destacar que G38 no había realizado la práctica 2, lo cual quizá explica parcialmente que la experimentación de su primera entrega no tenga sentido.

Varios grupos han necesitado modificar algunos algoritmos. La Tabla 11 muestra los cambios realizados.

Tabla 11. Cambios realizados para la experimentación incluyendo algoritmos de búsqueda (N=35)

Cambio	# (%) grupos	Grupos
Desconocido	2 (5'7%)	G30, G36
Ninguno	24 (68'6%)	G04, G05, G06, G09, G10, G11, G13, G16-1, G16-2, G19, G20, G21, G22, G23, G25, G26, G27-1, G28, G29-2, G32, G33, G34, G38-1, G38-2
Cabecera de algoritmos de búsqueda	1 (2'9%)	G12
Algoritmo de ramificación y poda	7 (20%)	G02, G07, G08, G17, G24, G29-1, G37
Algoritmos heurísticos	1 (2'9%)	G27-2

Cabe destacar que al menos el 68'6% de los informes no necesitaron realizar ninguna modificación en los algoritmos. Como en el resto de prácticas, se había permitido modificar la práctica 3a atendiendo a los comentarios del profesor. Sin embargo, el 25'7% de los informes declararon alguna modificación en los algoritmos, normalmente en los de búsqueda. El algoritmo de ramificación y poda se ha modificado normalmente en la actualización de la cota, aunque también se informa de la modificación de su estructura completa (G37).

Algunos grupos realizaron una extensión de la práctica (no contabilizada arriba) consistente en contar el número de nodos explorado por cada algoritmo de búsqueda (G26, G34).

Algunos grupos aportan algunos comentarios abiertos de interés:

- Satisfacción con la práctica (G10, G24).
- Satisfacción con el conjunto de prácticas (G19, G22, G27-1, G27-2, G38-2).
- Realización conjunta de las prácticas 3a y 3b (G21).
- Conveniencia de ejercitar más la definición de funciones de cota (G11).
- Interés en medir los tiempos de ejecución de los algoritmos de búsqueda (G22, G33), quizá por medio de OptimEx (G27-1, G27-2).
- Lanzamiento de excepciones por el algoritmo de vuelta atrás con datos muy grandes (G05).
- Llaman "eficiencia" a la optimidad (G13).
- No les parece bien que se les obligue a usar OptimEx, por los problemas de ejecución que les da (G08).
- Mala coordinación, en OptimEx, de las operaciones de cambio de columnas y coloreado de la tabla histórica (G09).

4.2 Resultados de la Práctica 5b

La práctica 5b era de carácter individual. Se recogieron 28 informes (25 en la primera entrega y 3 en la segunda, de los que 2 se entregan por primera vez, G12b, G36).

La Tabla 12 muestra los resultados obtenidos con la primera entrega de cada grupo. Mantenemos la misma numeración de grupos que en las prácticas anteriores. Cuando un grupo *Gnn* estaba formado por dos alumnos, ahora desglosamos sus informes en *Gna* y *Gnb*, en orden alfabético de sus apellidos.

Tabla 12. Resultados de la primera entrega de la práctica 5b (N=27)

Resultado	# (%) grupos	Grupos
Bien	24 (88'9%)	G04a, G05, G07a, G07b, G08a, G08b, G09, G10, G11, G20, G21, G22, G23, G24, G25, G26, G28a, G29a, G29b, G30, G34, G36, G37, G38a
Mal	3 (11'1%)	G01, G12b, G16

Cuatro grupos (15%) ya habían realizado la práctica 5b como parte final de la práctica 5a (G05, G08b, G24, sólo los comparan con algoritmos de vuelta atrás; G34

con todos). G21 lo afirma, aunque no aparece en el informe entregado de la práctica 5a.

Algunos experimentos correctos presentaban peculiaridades:

- No incluyeron algoritmos heurísticos. Hay varios casos: no incluyó el segundo algoritmo heurístico (G01-1), ninguno (G22, G29a) o ningún algoritmo heurístico ni el algoritmo recursivo de programación dinámica (G08a).
- No incluyeron el algoritmo recursivo de programación dinámica en la tabla de resumen, aunque sí en la tabla histórica (G09).
- Cuatro grupos (14'8%) incluyen más algoritmos de los pedidos (G05: 2 algoritmos de vuelta atrás y 2 tabulados; G07a: 2 algoritmos tabulados; G30, 3 algoritmos heurísticos; G34, 8 algoritmos heurísticos, 2 de vuelta atrás, 2 de ramificación y poda, 2 recursivos y 2 tabulados).
- Han realizado (relativamente) pocas ejecuciones (G10, 27; G12b, 49).
- Dos columnas de la tabla histórica tienen su coloreado intercambiado (G20).
- Los porcentajes de los dos algoritmos heurísticos son ligeramente distintos en la explicación y en la tabla de resumen (G23).

Las entregas mal realizadas presentan los siguientes problemas:

- Los algoritmos de programación dinámica dan resultados subóptimos, bien ambos (G01-1) bien el algoritmo tabulado (G12b, G16).
- Las tablas histórica y de resumen presentan datos contradictorios (G01-1). Además, la tabla histórica presenta mal el coloreado de sus celdas.

Resulta curioso el comentario de un grupo con la práctica mal (G16): “Haciendo esta práctica te das cuenta de la sencillez de los algoritmos de programación dinámica comparados con los algoritmos de técnicas exactas”.

Solamente realizó la segunda entrega uno de los tres grupos que tenían mal la primera entrega (G01), corrigiéndola. Cabe destacar que había realizado mal la práctica 2 y no había realizado la práctica 3b.

Varios grupos han necesitado modificar sus algoritmos de diversas formas. La Tabla 13 muestra los cambios realizados.

Tabla 13. Cambios realizados para la experimentación incluyendo algoritmos de programación dinámica (N=28)

Cambio	# (%) grupos	Grupos
Desconocido	4 (14'3%)	G01-2, G05, G29a, G30
Ninguno	19 (67'9%)	G04a, G07b, G08a, G08b, G09, G10, G12b, G16, G20, G21, G22, G23, G24, G25, G26, G29b, G34, G37, G38a
Borrar detalles de la clase (comentarios, <i>main</i> , etc.)	1 (3'6%)	G28a
Algoritmos heurísticos y cabecera de algoritmos de búsqueda	1 (3'6%)	G01-1
Algoritmo tabulado	3 (10'7%)	G07a, G11, G36

Cabe destacar que al menos 2/3 de los informes no necesitaron realizar ninguna modificación en los algoritmos. Como en el resto de prácticas, se había permitido modificar la práctica 5a siguiendo los comentarios del profesor. Sin embargo, el 15% de los alumnos declaró alguna modificación en los algoritmos, sobre todo de los tabulados.

Algunos grupos realizaron una ampliación de la práctica al medir experimentalmente el rendimiento de los algoritmos, bien midiendo tiempos (G05) bien contando nodos del árbol de búsqueda (G34), bien contando nodos del árbol de búsqueda o iteraciones (G26).

Algunos grupos aportan algunos comentarios abiertos de interés:

- Práctica útil (G07a, G07b).
- Práctica sencilla (G07a, G07b, G08a, G37).
- Satisfacción con el conjunto de prácticas relacionadas (G08b, G10, G11, G20, G22, G24, G38a).
- Satisfacción con el uso conocido de OptimEx (G04a).
- Utilidad de OptimEx (G07b, G08b, G11, G26), incluyendo un elogio: “Creo que es una buena herramienta para saber si los programas que realizamos, tanto en esta como en otras asignaturas están bien, no estaría mal incluir OptimEx en otras asignaturas de la carrera, aparte de AA y DAA” (G11).
- Interés en medir los tiempos de ejecución (G22).

5 Evolución de Resultados

La organización de prácticas se concibió como un intento de mejorar el aprendizaje de la experimentación y de reforzar los conocimientos de la teoría. La Tabla 14 presenta la evolución de las entregas realizadas por los alumnos. Se utilizan los símbolos ✓ y ✗ para indicar respectivamente una experimentación correcta o incorrecta.

Tabla 14. Corrección de las entregas realizadas de todas las prácticas

Grupo	Pr.2-1	Pr. 2-2	Pr.3b-1	Pr.3b-2	Pr.5b-1	Pr.5b-2
1.	✗	✗			✗	✓
2.	✓		✓			
3.	✗	✓				
4.	✗	✓	✓		✓	
5.	✓		✓		✓	
6.	✗	✓	✓			
7.	✓		✓		✓	✓
8.	✗	✓	✓		✓	✓
9.	✓		✓			

10.	✓		✓	✓	
11.	x	✓	✓	✓	
12.	✓		✓	x	
13.	✓		✓		
14.	✓				
15.	✓				
16.	✓		x	✓	x
17.	✓		✓		
18.	x	✓			
19.	x	x	✓		
20.	✓		✓	✓	
21.	✓		✓	✓	
22.	✓		✓	✓	
23.	✓		✓	✓	
24.	✓		✓	✓	
25.	✓		✓	✓	
26.	✓		✓	✓	
27.	✓		x	✓	
28.	✓		✓	✓	
29.	✓		x	✓	✓
30.	✓		✓	✓	
31.	✓				
32.	✓		✓		
33.	✓		✓		
34.	✓		✓	✓	
35.	✓				
36.	x	✓	✓		✓
37.	x	✓	✓	✓	
38.			x	✓	✓

Para analizar su evolución, vamos a considerar la evolución en la corrección de la primera entrega de cada práctica, dado que para la segunda ya tienen la guía de los comentarios del profesor. Encontramos 13 clases de evolución, que aparecen reflejadas en la Tabla 15. Cada clase de evolución se caracteriza por una secuencia de

tres símbolos, que representan el resultado de las tres prácticas. El símbolo – representa una práctica no entregada.

Tabla 15. Evolución de las primeras entregas de todas las prácticas (N=50)

Cambio	# (%) grupos	Grupos
x—	3 (6%)	G03, G18a, G18b
✓—	5 (10%)	G14, G15, G31, G35a, G35b
–x–	1 (2%)	G38b
x–x	1 (2%)	G01
–x✓	1 (2%)	G38a
x✓–	4 (8%)	G04b, G06, G19a, G19b
✓x–	1 (2%)	G27
✓✓–	11 (22%)	G02, G09, G12a, G13a, G13b, G17a, G17b, G28b, G30, G32, G33
x✓✓	6 (12%)	G04a, G08a, G08b, G11, G36, G37
✓xx	1 (2%)	G16
✓x✓	2 (4%)	G29a, G29b
✓✓x	1 (2%)	G12b
✓✓✓	13 (26%)	G05, G07a, G07b, G10, G20, G21, G22, G23, G24, G25, G26, G28a, G34

Podemos agrupar estas clases de evolución en 8 categorías:

- Sin información de cambio: x—, –x–, ✓—
- Se mantienen mal: x–x
- Se mantienen bien: ✓✓–, ✓✓✓
- Evolución variable: ✓x✓
- Empeoran algo: ✓x–, ✓✓x
- Empeoran mucho: ✓xx
- Mejoran algo: –x✓, x✓–
- Mejoran mucho: x✓✓

Veamos su frecuencia en la Tabla 16.

Tabla 16. Categorías de cambio de las primeras entregas de todas las prácticas (N=45)

Categoría de cambio	# (%) grupos
Sin información de cambio	9 (18%)
Se mantienen mal	1 (2%)
Se mantienen bien	24 (48%)
Evolución variable	2 (4%)
Empeoran algo	2 (4%)
Empeoran mucho	1 (2%)
Mejoran algo	5 (10%)
Mejoran mucho	6 (12%)

Agrupando en cuatro categorías todavía más generales, podemos concluir que:

- No tenemos información de cambio en 18% de los grupos. Además, otro 4% tiene una evolución variable, difícil de interpretar.
- Se mantienen igual la mitad de los grupos.
- Empeoran el 6% de los grupos.
- Mejoran el 22% de los grupos.

6 Resumen de Resultados

Resumimos en primer lugar los hallazgos de la práctica 2:

- La mayor parte de los grupos realizaron bien la práctica en la primera entrega (aproximadamente 3/4), porcentaje que con la segunda entrega subió hasta el 95%. Se puede destacar que el 13'5% de los grupos hicieron el experimento de forma concienzuda, variando las condiciones de generación aleatoria de los datos de entrada.
- La mayor parte de las prácticas incorrectas se deben a haber diseñado mal los algoritmos heurísticos (8 prácticas, 80%). Los alumnos confundieron los conceptos de función objetivo (del problema) y función de selección (del algoritmo). Solamente 2 grupos obtuvieron resultados superóptimos y 1 grupo realizó un experimento de 2 ó 3 ejecuciones.
- La evidencia utilizada casi universalmente ha sido la tabla de resumen (98%), siendo la tabla histórica presentada "sólo" por un 81%.
- Los términos "algoritmo más óptimo" u "óptimo al 100%" son usados por la tercera parte de los grupos.
- Casi un tercio de los grupos proponen mejoras en los algoritmos del enunciado o técnicas de diseño alternativas. Tres grupos codificaron estos algoritmos y dos midieron su optimidad.
- La sexta parte de los grupos muestra su interés por la eficiencia de los algoritmos, bien sugiriendo que se incorpore en la práctica bien tomando medidas ellos mismos.
- La mitad de los grupos ensalzan la utilidad de OptimEx (en dos casos, de forma elogiosa). Una quinta parte, comentan diversas dificultades.

Los resultados de la práctica 3b son los siguientes:

- El porcentaje de grupos que realizó bien la práctica es aún mayor que para la práctica 2 (87'1%), ya que la realizaron bien el mismo número de grupos (27) pero la entregaron menos (31 frente a 37). Tras la segunda entrega, todos los grupos la hicieron bien.
- Al igual que con otras prácticas, los alumnos tuvieron la oportunidad de realizar una segunda entrega de la práctica 3a para corregirla siguiendo los comentarios del profesor. Por tanto, en al menos dos tercios de las entregas no se tuvo que modificar ningún algoritmo. No obstante, la cuarta parte de las

entregas tuvieron que realizar alguna modificación, normalmente del algoritmo de ramificación y poda.

- Un quinto de los alumnos ya habían realizado la experimentación por propia iniciativa en la práctica 3a.
- La mayor parte de las prácticas mal realizadas se debe a no incluir algoritmos heurísticos, lo cual fue corregido fácilmente.
- Un quinto de los grupos muestran su satisfacción con la práctica, bien individualmente o, sobre todo, en relación con otras.

Asimismo, los resultados de la práctica 5b son:

- El porcentaje de alumnos que realizó bien la práctica es ligeramente mayor que para la práctica 3b (88'9%), realizándola bien algunos alumnos menos (24 frente a 27) y entregándola también algunos alumnos menos (27 frente a 31).
- Los alumnos que han realizado bien la práctica parecen presentar más incidencias en la experimentación que en las prácticas anteriores. Puede deberse a trabajar individualmente o a la mayor dificultad de los algoritmos.
- Los tres alumnos que hicieron mal la práctica presentan resultados subóptimos en el algoritmo tabulado. Solamente la corrigió un alumno.
- Realizan alguna modificación al menos un sexto de los alumnos, principalmente del algoritmo tabulado.
- Se repite la satisfacción con las prácticas y OptimEx, así como el interés por el rendimiento de los algoritmos.

Por último, si consideramos la evolución de los resultados, observamos que:

- La disminución de entregas puede deberse al menor peso de las prácticas 3b y 5b en la nota final, a la creciente dificultad de los algoritmos o, en el caso de 5b, al ser ya el final del cuatrimestre y estar los alumnos agobiados por la entrega de prácticas y exámenes. El 15% de los alumnos sólo ha realizado una práctica.
- La mitad no experimentan variación en la corrección de varias entregas y el número de mejoras es mayor que empeoramientos (22% frente al 6%). Hay que destacar este comportamiento, dado que los algoritmos probados son de dificultad creciente.

Los resultados son bastante mejores que los obtenidos en las experiencias de los dos cursos anteriores [4][5]. Sin embargo, pueden extraerse algunas conclusiones y recomendaciones:

- La organización actual de prácticas de experimentación con la optimidad produce mejores resultados que en cursos anteriores y goza de aceptación por los alumnos.
- Es preferible dar descripciones más completas de los algoritmos que el bosquejo dado de los algoritmos heurísticos, por ejemplo incluyendo varios ejemplos. También conviene ejercitar más la definición de las funciones de cota de los algoritmos de ramificación y poda y resaltar que los algoritmos tabulados deben tener el mismo comportamiento que los algoritmos recursivos redundantes.

- A pesar del alto porcentaje de alumnos que realizan bien las prácticas 2 y 3b, que afirman que es sencilla, y que incluso han realizado la práctica 3b por propia iniciativa, parece conveniente mantenerlas. Por un lado, ganan familiaridad con el proceso experimental, ya que no es raro que no comparen los nuevos algoritmos con los anteriores. Por otro lado, es una oportunidad para que algunos grupos corrijan los algoritmos desarrollados. Sin embargo, la práctica 5b debería unirse a la 5a. Aparte de que los alumnos ya están familiarizados con la experimentación, con lo que su realización es trivial para una amplia mayoría, se asegura un mayor número de entregas.
- La relativamente baja complejidad de las prácticas deja espacio para pedir más a los alumnos. Por ejemplo, según las características del problema, puede pedirse que la experimentación se haga con distintos rangos de valores de algunos datos de entrada. La práctica 3b podría incluir un apartado dedicado a contar los nodos del árbol de búsqueda generados por cada algoritmo. Siguiendo el ejemplo de un alumno (G34), basta con modificar ligeramente los algoritmos correctamente construidos para que cuenten nodos expandidos del árbol de búsqueda y compararlos con OptimEx.
- Sería deseable mejorar OptimEx a partir de las recomendaciones recogidas en estas evaluaciones y otras anteriores [3][5].

7 Conclusiones

Hemos presentado de forma detallada un análisis de la comprensión de conceptos de optimización. La evaluación se realizó con OptimEx en otoño de 2015. Se ha incluido el procedimiento y los enunciados de prácticas usados, los resultados detallados y comentados, así como una discusión de los mismos. Los resultados mejoran con respecto a las evaluaciones realizadas en los dos cursos anteriores [4][5].

Agradecimientos. Este trabajo se ha financiado con los proyectos TIN2011-29542-C02-01 y TIN2015-66731-C2-1 del Ministerio de Economía y Competitividad de España y S2013/ICE-2715 de la Comunidad Autónoma de Madrid.

Referencias

1. Velázquez-Iturbide, J.Á. Design and evaluation of OptimEx, an experimentation system for optimization algorithms. En: *ICT in Education - Multiple and Inclusive Perspectives*, M.J. Marcelino, A.J. Mendes y C. Azevedo Gomes (eds.) (en imprenta) Springer
2. Velázquez Iturbide, J.Á., Debdi, O., Esteban Sánchez, N., Pizarro, C.: GreedEx: A visualization tool for experimentation and discovery learning of greedy algorithms. *IEEE Transactions on Learning Technologies* 6, 2 (abril-junio 2013) 130-143, DOI [10.1109/TLT.2013.8](https://doi.org/10.1109/TLT.2013.8)
3. Velázquez Iturbide, J.Á. Una evaluación de usabilidad de OptimEx. En: *Serie de Informes Técnicos DLSII-URJC*, no. 2014-02, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2014.

4. Velázquez Iturbide, J.Á. Difficulties, attitudes and misconceptions on experimenting with optimization algorithms. En: Proceedings of the 2014 International Symposium on Computers in Education (SIIE'14), IEEE Xplore (2014) 17-22, DOI [10.1109/SIIE.2014.7017698](https://doi.org/10.1109/SIIE.2014.7017698)
5. Velázquez Iturbide, J.Á. Una segunda evaluación cualitativa de la comprensión de la optimalidad. En: Serie de Informes Técnicos DLSII-URJC, no. 2015-02, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2014.
6. Velázquez Iturbide, J.Á. An experimental method for the active learning of greedy algorithms. ACM Transactions on Computing Education 13, 4 (octubre 2013) artículo 18, DOI [10.1145/2534972](https://doi.org/10.1145/2534972)
7. Velázquez Iturbide, J.Á. Identification and removal of misconceptions on optimization concepts underlying greedy algorithms. Journal of Research and Practice in Information Technology 45, 3/4 (2013) 203-217

Apéndice A: Enunciado de la Práctica 2

Grado en Ingeniería Informática Asignatura *Algoritmos Avanzados* Curso 2015/2016 Práctica nº 2

Objetivo

El objetivo de la práctica es que el alumno practique con los algoritmos heurísticos.

Carácter

La práctica es voluntaria, pero es un requisito para poder entregar la segunda parte de las prácticas 3 y 5. Puede realizarse individualmente o en pareja.

Enunciado

Sea un pequeño negocio con dos sedes, s_0 y s_1 , en dos ciudades distintas. Según la clientela prevista en cada mes, le puede interesar realizar su actividad en una sede o en otra. Si durante el mes i está en la sede s_0 , sus operaciones le supondrán un coste $c_{0,i}$, pero si está en la sede s_1 , el coste será $c_{1,i}$ ($c_{s,i} > 0$, para $s \in \{0,1\}$, $0 \leq i \leq n-1$). El cambio de una sede a otra siempre conlleva un gasto fijo $f \geq 0$.

El *problema del plan de sedes de coste mínimo* consiste en decidir en qué sede debe el negocio concentrar su actividad durante cada mes de forma que se minimicen los costes totales de operación.

Por ejemplo, los costes de 4 meses pueden ser $c_0 = \{1,3,20,30\}$ y $c_1 = \{50,20,2,4\}$, con un coste de traslado $f=10$. Un plan con coste mínimo sería $\{s_0, s_0, s_1, s_1\}$, con un coste igual a $1+3+10+2+4=20$. Puede comprobarse que cualquier otro plan tendrá un coste mayor.

Un algoritmo heurístico consiste en seleccionar en cada mes la sede que tiene menos coste. Una variación de este algoritmo tomaría la decisión teniendo también en cuenta el coste del traslado f .

El objetivo de la práctica es implementar ambos algoritmos y usar OptimEx para comparar su optimalidad, medida como el porcentaje de casos en los que cada algoritmo calcula una solución óptima.

Entrega

El alumno debe entregar un informe elaborado siguiendo el índice detallado a continuación. El informe debe enviarse por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto “Práctica 2”. El plazo de entrega del informe es el domingo 11 de octubre de 2015, incluido.

Informe

El alumno debe entregar un informe con la siguiente estructura:

1. **Implementación de los algoritmos.** Implementar ambos algoritmos en una sola clase de Java y con la siguiente cabecera (obviamente, con identificadores distintos):

```
public static int sedes (int[] c0, int[] c1, int f)
```

donde $c0$ y $c1$ son los vectores de costes mensuales de las sedes $s0$ y $s1$, respectivamente, y f es el coste fijo de cada cambio de sede. Para el ejemplo anterior, la llamada será `sedes({1,3,20,30},{50,20,2,4},10)`. Cada algoritmo debe devolver el coste óptimo de un plan de sedes.

2. **Experimentación con la optimalidad de los algoritmos.** Se comparará la optimalidad de ambos algoritmos con OptimEx y se incluirá:
 - a) Resultado. Se dirá si algún algoritmo es exacto.
 - b) Evidencias. Deben aportarse los resultados recogidos en las tablas de resumen e histórica y explicar el significado de dichos resultados.

Puede usarse OptimEx de la siguiente forma: se carga la clase con los dos algoritmos a comparar, se selecciona su signatura, se realiza una ejecución intensiva y se comparan los resultados obtenidos en la tabla resumida. Hay que tener cuidado con respetar las restricciones del enunciado al generar datos aleatoriamente: los costes mensuales son mayores que cero, y el coste fijo f es no negativo. La guía de uso de OptimEx identifica diversos resultados que, si se obtienen, son contradictorios con los conceptos presentados en clase; conviene leerla atentamente para, en su caso, identificar y corregir estas situaciones.

3. **(Opcional.) Razonamiento sobre la optimalidad de los algoritmos.** En el apartado anterior ha podido resultar que algún algoritmo, los dos o ninguno parezca ser óptimo. En este apartado se pide que se intente razonar sobre los algoritmos a partir de los resultados obtenidos. Si algún algoritmo ha dado siempre resultados óptimos, se pide una demostración (o boceto) de su optimalidad. Si algún algoritmo da a veces resultados subóptimos, se pide aportar un contraejemplo de su optimalidad que sea lo más sencillo posible (es decir, que implique pocos meses).
4. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de los algoritmos heurísticos o cualquier comentario sobre la práctica. Por ejemplo, pueden

describirse las incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.

Evaluación

Se evaluará la calidad y claridad de todos los apartados del informe.

Apéndice B: Enunciado de la Práctica 3b

Grado en Ingeniería Informática Asignatura *Algoritmos Avanzados* Curso 2015/2016 Práctica nº 3.b

Objetivo

El objetivo de la práctica es que el alumno profundice en su conocimiento de las técnicas de diseño exactas y no exactas.

Carácter

La práctica es voluntaria pero sólo puede entregarse si se ha realizado la práctica 2. Debe realizarse en el mismo grupo que la práctica 2 (individualmente o con el mismo compañero).

Enunciado

En la primera parte de la práctica se pidió desarrollar de forma sistemática dos algoritmos de búsqueda que resuelvan el *problema del plan de sedes de coste mínimo*: uno de vuelta atrás y otro de ramificación y poda. Obsérvese que ambos algoritmos calcularán el mismo resultado, aunque previsiblemente de forma más eficiente en el caso de ramificación y poda. El objetivo de esta segunda parte es comprobar la optimalidad de ambos algoritmos.

Entrega

El alumno debe entregar un informe elaborado siguiendo el índice detallado a continuación. El informe debe enviarse por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto "Práctica 3b". El plazo de entrega del informe es el jueves 5 de noviembre de 2015 a las 15h.

Informe

El alumno debe entregar un informe con la siguiente estructura:

1. **Comparación de optimalidad de algoritmos.** Se añadirán los dos algoritmos de búsqueda en la clase utilizada en la práctica 2 para comparar algoritmos heurísticos. Obsérvese que los dos algoritmos de búsqueda están diseñados con técnicas exactas; por tanto, deberían ser óptimos en el 100% de los casos. Si no fuera así, deberían revisarse los 4 algoritmos para identificar y corregir los errores,

y repetir el experimento hasta que se obtengan los resultados predichos por la teoría. (En el manual de OptimEx puede consultarse un catálogo de situaciones conflictivas y su resolución.)

Se pide:

- a) Identificar cada algoritmo medido, indicando la técnica de diseño con la que se ha desarrollado y, si es necesario, algún elemento diferenciador (p.ej. la función de selección de cada algoritmo heurístico).
 - c) Resultado. Se dirá qué algoritmos son exactos según los resultados de la experimentación.
 - d) Evidencias. Deben aportarse los resultados recogidos en las tablas de resumen e histórica y explicar el significado de dichos resultados.
 - b) Incidencias. Se indicará si ha sido necesario revisar y modificar algún algoritmo durante esta práctica, explicando por qué se realizó dicha modificación y en qué consistió.
2. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de las técnicas de búsqueda o cualquier comentario sobre la práctica. Por ejemplo, pueden describirse las incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.

Evaluación

Se evaluará la calidad y claridad de todos los apartados del informe.

Apéndice C: Enunciado de la Práctica 5b

Grado en Ingeniería Informática Asignatura *Algoritmos Avanzados* Curso 2015/2016 Práctica nº 5.b

Objetivo

El objetivo de la práctica es que el alumno profundice en su conocimiento de las técnicas de diseño exactas y no exactas.

Carácter

La práctica es voluntaria pero sólo puede entregarse si se ha realizado la práctica 2. Debe realizarse individualmente.

Enunciado

En la primera parte de la práctica se pidió desarrollar de forma sistemática un algoritmo de programación dinámica que resolviera el *problema del plan de sedes de coste mínimo*. Se desarrolló un algoritmo recursivo, redundante e ineficiente, que posteriormente se convirtió en un algoritmo tabulado, más eficiente. El objetivo de esta segunda parte es comprobar la optimidad de ambos algoritmos.

Entrega

El alumno debe entregar un informe elaborado siguiendo el índice detallado a continuación. El informe debe enviarse por medio del apartado de Evaluación del campus virtual. Si se tienen dificultades, puede enviarse por el correo del campus virtual con el asunto "Práctica 5b". El plazo de entrega del informe es el jueves 10 de diciembre de 2015 a las 15h.

Informe

El alumno debe entregar un informe con la siguiente estructura:

1. **Comparación de optimidad de algoritmos.** Se añadirán los dos algoritmos de programación dinámica en la clase utilizada en la práctica 3b (o en la práctica 2 si la 3b no se realizó). Obsérvese que ambos algoritmos están diseñados con técnicas exactas; por tanto, deben ser óptimos en el 100% de los casos. Si no fuera así, deberían revisarse todos los algoritmos para identificar y corregir los errores, y repetir el experimento hasta que se obtengan los resultados predichos por la teoría.
Se pide:

- a) Identificar cada algoritmo medido, indicando la técnica de diseño con la que se ha desarrollado y, si es necesario, algún elemento diferenciador (p.ej. la función de selección de cada algoritmo heurístico).
 - b) Resultados. Se dirá qué algoritmos son exactos según los resultados de la experimentación.
 - c) Evidencias. Deben aportarse los resultados recogidos en las tablas de resumen e histórica y explicar el significado de los resultados mostrados.
 - d) Incidencias. Se indicará si ha sido necesario revisar y modificar algún algoritmo durante esta práctica, explicando por qué se realizó dicha modificación y en qué consistió e incluyendo el código fuente de los algoritmos modificados.
2. **Conclusiones.** Se explican las conclusiones obtenidas tras realizar la práctica. Estas conclusiones pueden consistir en una valoración de la técnica de programación dinámica, de su conveniencia frente a otras técnicas de diseño o cualquier otro comentario sobre la práctica (incidencias que han dificultado la realización de la práctica, sus aspectos más atractivos o más difíciles, sugerencias sobre cómo mejorar la práctica, etc.)

Evaluación

Se evaluará la calidad y claridad de todos los apartados del informe.