Métodos Algorítmicos en Resolución de Problemas II

Grado en Ingeniería Informática

Doble Grado en Ingeniería Informática y Matemáticas

Hoja de ejercicios 2 — Curso 2025–2026

Ejercicios de ramificación y poda

Ejercicio 1 Sean n programas $P_1, P_2, \ldots, P_n$ que hay que almacenar en un disco. El programa P_i requiere s_i kilobytes de espacio de disco, y la capacidad del disco es D kilobytes, donde $D < \sum_{i=1}^{n} s_i$. Desarrollar un algoritmo que elija el subconjunto de los programas que tienen que ser almacenados de forma que se maximice la capacidad de disco utilizado.

Ejercicio 2 Se tiene un sistema monetario formado por un conjunto finito $M = \{m_1, m_2, \ldots, m_n\}$ de tipos de monedas. Suponiendo que la cantidad disponible de monedas del tipo m_i es c_i, se quiere pagar una cantidad C utilizando un número total de monedas mínimo. Desarrollar un algoritmo que devuelva la mejor solución.

Ejercicio 3 Aplicar ramificación y poda al problema del coloreado óptimo de un grafo no dirigido. Se han de pintar los nodos de un grafo con una serie de colores, de manera que ningún arco del mismo una dos nodos del mismo color. Se trata de hacerlo utilizando el mínimo número de colores posible.

Ejercicio 4 Tenemos que distribuir a n invitados en una mesa redonda con espacio para n comensales. Se dispone de una función *afinidad*(i, j), $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$, que devuelve un valor positivo según el grado de afinidad que los invitados i y j tienen entre sí (a mayor valor, mayor afinidad). Diseñar un algoritmo que calcule la distribución de los invitados en la mesa de forma que se maximice el bienestar general. El bienestar general se calcula sumando las afinidades de los comensales sentados en posiciones adyacentes.

Ejercicio 5 Se desea organizar el banquete de una boda para n comensales. Se dispone de un comedor con m mesas de capacidad n/m, siendo n múltiplo de m. Se conocen las afinidades $a_{ij}, 1 \leq i, j \leq n$ entre cada par de comensales. La matriz de afinidades no es necesariamente simétrica, aunque sabemos que $a_{ii} = 0$, para todo i. Definimos la satisfacción global S de una distribución de comensales en mesas como la suma $S = \sum_{k=1}^{m} \sum_{i,j \in mesa_k} a_{ij}$. Diseñar un algoritmo que encuentre una asignación de comensales a mesas que maximice la satisfacción global. Se prestará especial atención a no generar soluciones equivalentes.

Ejercicio 6 Tenemos n trabajos que ejecutar y m procesadores iguales que trabajan en paralelo (con $m < n$). El tiempo necesario para ejecutar el trabajo i en cualquiera de los procesadores es t_i. Teniendo en cuenta que el orden en el que se ejecuten los trabajos dentro de un mismo procesador no es significativo, escribe un algoritmo utilizando la técnica de ramificación y poda que determine qué trabajos se deben ejecutar en qué procesadores de tal forma que se minimice el tiempo de finalización del último trabajo que se ejecuta. Hay que evitar encontrar soluciones equivalentes: dos soluciones se consideran equivalentes si una se consigue a partir de la otra simplemente mediante la permutación de los procesadores.

Ejercicio 7 Deseamos decorar una pared de L metros de ancho. Hemos tenido la innovadora idea de hacerlo colgando una hilera de cuadros pegados lado con lado. Nos disponemos a comprar los cuadros en la feria de arte moderno, donde tenemos la posibilidad de elegir entre n cuadros. Cada cuadro tiene un prestigio p_i, y unas dimensiones de a_i metros de alto por b_i metros de ancho, $1 \leq i \leq n$. Dado lo peculiar de los cuadros, podemos elegir colgar cada cuadro tanto en horizontal como en vertical sin que por ello se vea afectado su prestigio. Lo que no podemos hacer es trocear un cuadro. Diseñar un algoritmo que determine qué cuadros comprar de forma que la longitud de la hilera de cuadros sume exactamente L metros y se maximice el prestigio acumulado en la pared.

Ejercicio 8 Dado un grafo valorado no dirigido $G = \langle V, A \rangle$ y un entero r, escribir un algoritmo que encuentre, si existe, un cliqué de tamaño r y de mínimo coste en G. Un cliqué es un subgrafo completo, es decir, un subconjunto $V' \subseteq V$ de vértices tal que existe una arista entre cada par de vértices en V'. El tamaño de un cliqué es su número de vértices, y su coste es la suma de los costes de sus aristas. Si hubiese empates basta dar uno de ellos.

Ejercicio 9 Tenemos un conjunto de n componentes electrónicas para colocar en n posiciones sobre una placa. Se nos dan dos matrices de dimensiones $n \times n$, N y D, donde $N[i, j]$ indica el número de conexiones necesarias entre la componente i y la componente j, y $D[p, q]$ indica la distancia sobre la placa entre la posición p y la posición q (ambas matrices son simétricas y con diagonales nulas). Un cableado $(x_1, \ldots, x_n)$ de la placa consiste en la colocación de cada componente i en una posición distinta x_i. La longitud total de este cableado viene dada por la fórmula $\sum_{i<j} N[i, j]D[x_i, x_j]$. Escribir un algoritmo para encontrar el cableado de longitud mínima.

Ejercicio 10 Una de las pruebas habituales del concurso *Supervivientes* es la construcción de una cabaña rudimentaria cuyo techo es una simple lona soportada por cuatro pilares. Los concursantes dispondrán de n de fragmentos de caña de longitudes enteras $l_1, \ldots, l_n$ ensamblando los cuales deberán obtener los cuatro pilares. El objetivo es que sus alturas queden razonablemente equilibradas y sean lo más altas posibles. Precisando, desarrollar el problema buscando maximizar el pilar más bajo de los cuatro.

Ejercicio 11 Alonso Rodríguez tiene que hacer la compra de la semana consistente en n productos. En su barrio hay m supermercados, cada uno de los cuales dispone de todos esos productos. Pero no quiere comprar más de tres productos en cada uno de los supermercados, ya que así pasa más tiempo comprando (se puede suponer que $n \leq 3m$). Dispone de una lista de precios de los productos en cada uno de los supermercados. Escribir un algoritmo de ramificación y poda, aplicando el esquema optimista-pesimista en caso de ser posible, que le permita decidir cómo hacer la compra de forma el coste total sea mínimo.

Ejercicio 12 Juanita afronta sus exámenes finales del curso con precipitación. Para cada asignatura A_i del curso conoce los días f_i que quedan hasta la fecha de su examen, y sabe la cantidad de días d_i que debería dedicar a su estudio (lógicamente, antes del correspondiente examen) para poder aprobarla. Juanita se teme que no podrá con todas las asignaturas, así que ha dado un valor v_i a cada una de ellas, y ha de determinar qué asignaturas estudiará, y en qué orden, hasta la finalización de los exámenes, a sabiendas de que si dedica a una asignatura menos de los d_i estipulados la suspenderá irremediablemente. Debéis ayudar a Juanita a elegir las asignaturas que estudiará, de manera que maximice el valor total de las asignaturas que ha decidido estudiar con tiempo suficiente para aprobarlas. Describir el algoritmo de ramificación y poda apropiado para este problema, e indicar con detalle las cotas inferior y superior empleadas.

Ejercicio 13 Para un conjunto de n hombres y otro de n mujeres, nos dan dos matrices de números naturales $n \times n$, H y M, en las que H_{ij} representa la afinidad del hombre i hacia la mujer j y M_{ij} la afinidad de la mujer i hacia el hombre j. Un emparejamiento es cualquier aplicación biyectiva entre ambos conjuntos. Un emparejamiento es estable si en el mismo no existe un hombre i y una mujer j tales que i tenga más afinidad por j que por su pareja y a la vez j tenga más afinidad por i que por la suya. El valor de un emparejamiento estable es la suma de los productos de las afinidades mutuas de cada pareja. Se pide desarrollar un algoritmo que devuelva el emparejamiento estable de mayor valor.

Ejercicio 14 La Facultad de Informática debe planificar sus exámenes de septiembre de 2020 necesariamente entre los días 1 y n de ese mes. Tiene que realizar m exámenes, $m > n$, y puede planificar cada día cuantos exámenes desee, siempre que no sean incompatibles entre sí, y para ello dispone de una matriz I, $m \times m$, de booleanos, donde $I_{ij} = \mathit{true}$ si los exámenes i, j son incompatibles e $I_{ij} = \mathit{false}$ en caso contrario.

Implementar un algoritmo que indique el día a realizar cada examen, de forma que el máximo número de aulas necesario sea mínimo. Se supondrá que cada examen ocupa un aula durante todo el día. Se valorará negativamente el que se generen soluciones equivalentes.

Ejercicio 15 Se quiere grabar en vídeo un concierto al aire libre desde tres puntos distintos, para lo que se cuenta con tres cámaras de la misma marca y modelo. Debido a que no hay puntos de electricidad a los que enchufar las mismas, los organizadores se han visto obligados a utilizar n baterías. Como las han comprado en diversas tiendas, sus duraciones d_i, $1 \leq i \leq n$, varían de unas a otras. El problema consiste en decidir cómo distribuir las baterías entre las tres cámaras de manera que se maximice el tiempo que las tres estén simultáneamente grabando.

Ejercicio 16 La hija de Mariano ha llegado ya a una edad en la que ha pedido su primer móvil, y Mariano se lo ha comprado. Sabe que, de ahora en adelante, no querrá pasar sin él y ha llegado a la conclusión de que, para que su hija esté contenta, los próximos T años tendrá que renovarlo como mucho cada M años (siendo $T > M$). Por otra parte, sabe que a medida que crezca, sus exigencias en cuanto a prestaciones o marcas irán cambiando, por lo que ha hecho una estimación del coste de renovar el móvil en cada uno de los próximos T años: $c_1, \ldots, c_T$. Se pide:

1. Diseñar e implementar un algoritmo de ramificación y poda que ayude a Mariano a decidir en qué años renovar el móvil, para minimizar la suma de los costes de las renovaciones.
2. Explicar al menos dos cotas optimistas y dos pesimistas, e implementar una de cada tipo, justificando la elección realizada.

Ejercicio 17 La pandemia Covid-19 ha forzado a muchos restaurantes a digitalizarse. El restaurante *Come Sano* dispone de n plazas y conoce las distancias d_{ij} entre cada dos plazas i y j, $1 \leq i, j \leq n$. Tiene reservas para $m \leq n$ comensales y una matriz de booleanos c_{ij} indica si dos comensales i y j, $1 \leq i, j \leq m$ son o no convivientes. Dicha matriz refleja una relación simétrica, no reflexiva y no necesariamente transitiva. Las parejas convivientes puede disponerlas con cualquier distancia entre los dos miembros, pero en las no convivientes, los dos miembros han de estar separados al menos dos metros entre sí. Se pide un algoritmo de ramificación y poda que asigne cada comensal a una plaza, de forma que se respeten las distancias de seguridad y se maximice el número de parejas de comensales convivientes cuyos miembros están a menos de dos metros de distancia.

Ejercicio 18 Para detectar eficazmente los contagios de la pandemia, el Gobierno de Buenrollitolandia ha entrenado a n equipos de rastreadores, que han de ser asignados a n comarcas. Por razones logísticas, algunos equipos no pueden ir a algunas comarcas y, en aquellas a las que pueden acudir, tienen una eficacia estimada, que varía de 1 a 10 dependiendo de la comarca asignada. Dada una matriz $\{e_{ij}\}_{1 \leq i,j \leq n}$, con dichas eficacias, donde se indica con $e_{ij} = -\infty$ que el equipo i no puede ir a la comarca j, desarrollar y escribir un algoritmo de ramificación y poda que encuentre una asignación cuya suma de eficacias sea máxima.

Ejercicio 19 Un ratón de laboratorio tiene disponibles en su jaula n teclas para pulsar. Cuando pulsa una tecla por primera vez pone en marcha un mecanismo de recompensas y castigos. Cada vez que pulsa de nuevo una tecla recibe una recompensa o un castigo dependiendo de la última tecla que pulsó y la actual. Esta información viene dada en una matriz T en la que para cada par de teclas i, j ($1 \leq i, j \leq n$) $T[i][j]$ indica la recompensa (un valor ≥ 0) o el castigo (un valor < 0) que recibe el ratón al pulsar la tecla j inmediatamente después de la i.

Se pide diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema de encontrar una secuencia de teclas de longitud m que maximice la suma de recompensas obtenidas por el ratón, teniendo en cuenta que la suma de los castigos en valor absoluto no puede ser superior a un determinado valor dado $C \geq 0$.

Ejercicio 20 Dado un conjunto de n actividades con instante de comienzo y finalización conocidos $[c_i, f_i)$ y un beneficio $b_i \geq 0$ ($1 \leq i \leq n$), se pide diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema de encontrar el subconjunto de r (con $0 \leq r \leq n$) o más actividades (teniendo en cuenta que cada una solo se puede hacer una vez) que no solapen entre sí, cuya duración desde el comienzo de la primera (la que comienza más temprano) hasta el final de la última (la que termina más tarde) no supere un determinado tiempo T y cuyo beneficio sea máximo. Se valorarán todos todos los pasos: descripción del espacio de soluciones, definición y discusión de las cotas optimistas y pesimistas, implementación del algoritmo y análisis del coste en tiempo.

Ejercicio 21 Santiago, un alumno de la Facultad de Informática ha de decidir de qué asignaturas matricularse el año que viene, teniendo en cuenta que el número de créditos de las asignaturas en las que se matricule ha de ser mayor o igual que una cierta cantidad C. Algunas de las n asignaturas de la titulación que está cursando tienen un prerrequisito. Estos prerrequisitos vienen dados en un vector r de n componentes, donde $r[i]$ representa la asignatura que es necesario tener previamente aprobada para poder matricularse de la asignatura i (o -1 en caso de no tener prerrequisito). Por supuesto, Santiago sabe para cada asignatura si la tiene aprobada o no, el número de créditos y el coste.

Se pide diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema de decidir en qué asignaturas matricularse (de las aún no aprobadas) cuya suma de créditos supere C, respete los prerrequisitos y la suma de los costes de dichas asignaturas sea mínima. Describe claramente el espacio de soluciones, los marcadores utilizados y la poda de optimalidad realizada.

Ejercicio 22 Llega el verano y con él las brigadas antiincendios. Tenemos n montes que vigilar y para ello hemos contratado a m vigilantes. Ahora debemos formar las brigadas antiincendios, una por cada monte a vigilar. Numeraremos las brigadas según la numeración del monte al que son asignadas, así la brigada 1 irá al monte 1, la brigada 2 al monte 2, etc. Cada monte necesita un número mínimo de vigilantes, almacenado en un vector B, aunque la brigada encargada del monte puede estar formada por más vigilantes de los estrictamente necesarios.

Una matriz d almacena en $d[i][j]$ la distancia del vigilante i al monte j ($1 \leq i \leq m$, $1 \leq j \leq n$). Todos los vigilantes reciben el mismo salario, pero tienen derecho a un plus de movilidad, que es mayor cuanto mayor es el desplazamiento del vigilante desde su casa hasta el monte que le toque vigilar. Por ello, no permitiremos que un vigilante forme parte de una brigada asignada a un monte que esté más lejos de su casa que un cierto valor dado $D > 0$.

Se pide diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema de formar las brigadas de forma que todos los vigilantes estén asignados a una brigada y se minimice la suma de los desplazamientos que deban realizar los vigilantes. Describe claramente el espacio de soluciones, los marcadores utilizados y la poda de optimalidad realizada. Analiza el coste en tiempo de ejecución y en memoria del algoritmo implementado.

Ejercicio 23 La asociación "Pon una mascota en tu vida" tiene n mascotas para ser adoptadas y una lista de n candidatos que han visitado la sede de la asociación para conocerlas. En base a los gustos, situaciones personales y laborales de los candidatos y necesidades de las mascotas, la asociación ha elaborado una matriz de afinidades A. El valor $A[i][j]$ representa la afinidad entre la mascota i y el candidato j ($0 \leq i < n$, $0 \leq j < n$) si el valor es positivo, mientras que si es -1 representa que no son compatibles. Se desea asignar a cada persona exactamente una mascota compatible de forma que se maximice la suma de las afinidades de los emparejamientos realizados.

Diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema. Describir claramente el espacio de soluciones, las estimaciones y los costes del algoritmo.

Ejercicio 24 Dos compañeros de piso, Ibai y Manu, consideran que las tareas domésticas son sumamente tediosas. Pero puesto que han de realizarse, han hecho una lista de n tareas imprescindibles y a cada una le han asignado un coste $c_i > 0$. Quieren repartirse las n tareas entre ellos lo más equitativamente posible, es decir, minimizando el máximo entre c_I y c_M donde c_I representa la suma de los costes de las tareas asignadas a Ibai y c_M la de las tareas asignadas a Manu.

Diseñar e implementar un algoritmo de ramificación y poda que resuelva el problema de decidir qué tareas debe realizar cada uno. Describir claramente el espacio de soluciones, las estimaciones y los costes del algoritmo.