OPTIMIZACIÓN - 1er Cuatrimestre 2009

Práctica 3

Grafos

Ejercicio 1 Sea $G = (V, E)$ donde $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ y $E = \{(0, 1), (1, 2), (2, 3), (3, 4), (0, 4), (0, 5), (1, 6), (2, 7), (3, 8), (4, 9), (5, 6), (6, 7), (7, 8), (8, 9), (5, 9)\}$. Dibujar a G y hallar las matrices de incidencia y la tabla de adyacencia de G.

Ejercicio 2 Mostrar que un grafo $G = (V, E)$ con $|V| \geq 2$ tiene por lo menos 2 vértices cuyos grados son iguales.

Ejercicio 3 Mostrar que un grafo es Bipartito si y sólo si todo circuito tiene un número par de ramas.

Problemas y Algoritmos

Ejercicio 4 La expresión booleana $(x_1 + x_2 + x_3)(x_1' + x_2)(x_2' + x_3)(x_3' + x_1)$ $(x_1' + x_2' + x_3')$ no tiene una asignación que la satisfaga.

Ejercicio 5 Probar que SAT se reduce a ILP.

Ejercicio 6 Hallar un algoritmo polinomial para el problema $2SAT$.

Ejercicio 7 Sean los siguientes problemas:

- $CLIQUE$: Sea $G = (V, E)$ un grafo. Decimos que $A \subset V$ forma un *clique* si $u, v \in A \Rightarrow (u, v) \in E$. El problema es **Hallar un clique** A **tal que** $|A| = k$.
- $INDEPENDIENTE$: Decimos que $A \subset V$ es *independiente* si $u, v \in A \Rightarrow (u, v) \notin E$. El problema es **Hallar un conjunto independiente** A**, tal que** $|A| = k$.
- $CUBRIMIENTO$: Decimos que $A \subset V$ forma un *cubrimiento por vertices* si $(u, v) \in E \Rightarrow u$ ó $v \in A$. El problema es **Hallar un cubriemiento** A**, tal que** $|A| = k$.

1. Mostrar que los tres problemas anteriores equivalentes.
2. Mostrar que los tres son $\mathcal{NP}-$completos. Sugerencia, probar que $3SAT$ se reduce a $INDEPENDIENTE$.

Ejercicio 8 El problema $VAGABUNDO$ es similar al TSP, pero el viajante no tiene que retornar a la ciudad de origen.

Probar que $VAGABUNDO \equiv TSP$.

Backtraking y Branch and Bounding

Ejercicio 9 Sea G un grafo dirigido, u, v dos vértices y $c : E \to \mathbb{R}^+$ una funcion de costo. Se trata de hallar un camino dirigido de u a v de mínimo costo. Plantear este problema con Branch and Bounding.

Ejercicio 10 Se trata de colocar en un tablero de ajedrez de $n \times n$, n reinas que no se ataquen. Diseñar un algoritmo de Backtracking que resuelva este problema.

Ejercicio 11 Usando Backtraking listar todos los cliques de un grafo. Plantear un algoritmo basado en lo anterior para resolver el problema $CLIQUE$.

Programación Dinámica

Ejercicio 12 Sea el grafo dirigido $G = (V, E)$ con $V = \{1, 2, 3, 4, 5, 6\}$ y $E = \{(1, 2), (1, 4), (1, 3), (2, 4), (2, 5), (3, 4), (3, 6), (4, 5), (5, 6)\}$ con las siguientes longitudes de las ramas: $c_{12} = 2, c_{13} = 3, c_{14} = 6, c_{24} = 7, c_{25} = 8, c_{34} = 2, c_{36} = 9, c_{56} = 1$.

1. Para cada vertice u sea $v_0(u)$ el vértice más próximo a u. A v_0 la llamamos política miope. Hallar un camino del nodo 1 al nodo 6 usando la política miope.

2. Hacer un planteo de Programación Dinámica para encontrar un camino de longitud máxima entre el nodo 1 y el nodo 6.

Ejercicio 13 Se tienen monedas de 5,10,25,50 y 100 centavos. Sea x entero múltiplo de 5. Llamemos $f(x)$ al mínimo número de monedas que suman x. Escriba la ecuación funcional de f según la Programación Dinámica. Usando el algoritmo de la Programación Dinámica hallar la manera de juntar 235 centavos con un mínimo número de monedas.

Ejercicio 14 Hacer el planteo de Programación Dinámica para el problema de la mochila (knapsack).

Ejercicio 15 Para obtener el título de Doctor en Matemática uno de los requisitos es juntar por lo menos 20 puntos en materias optativas. Sea p_i el puntaje de la materia $i = 1, ...N$, donde N es el número de materias optativas. Los doctorandos han estimado un puntaje de dificultad de las materias que contiene aspectos como la modalidad de examen, carga horaria, entre otras cosas. Sea c_i el puntaje de la materia i. Hacer el planteo de Programación Dinámica para los siguieintes problemas, ámbos con el objetivo común de sumar al menos 20 puntos.

1. Que la suma de las dificultades sea mínima.

2. Que la dificultad más alta sea mínima.

Ejercicio 16 La Ciudad de Buenos Aires tiene N comunas con p_i habitantes en la comuna i. Se desea formar un concejo de R representantes $(R > N)$ y se pretende que cada comuna teng aun número de representantes proporcional a p_i. Sea $r_i = R\frac{p_i}{\sum_{k=1}^{N} p_k}$. Sea x_i el número de representantes de la comuna i. Hacer el planteo de Programación Dinámica con el objetivo de que sea mínima la mayor de las diferencias $|x_i - r_i|$. Calculear además el caso $N = 3$, $R = 4$, $r_1 = 0.4, r_2 = 2 - 4, r_3 = 1.2$.