Name:

MATP6620/DSES6770
Combinatorial Optimization and Integer Programming
Spring 2009

Midterm Exam, Thursday, March 19, 2009.

Please do all three problems. Show all work. No books or calculators allowed. You may use any result from class, the homeworks, or the texts, except where stated. You may use one sheet of handwritten notes. The exam lasts 110 minutes.

SOLUTIONS

Q1	/20
Q2	/30
Q3	/20
Q4	/20
Q5	/10
Total	/100

1. (20 points)
Recall the following material on clustering problems from Homework 4:

Given the complete graph $K_n =: (V, E)$ on n vertices, a **clustering** *of the vertices is obtained by choosing an integer p and a partition of the vertices into p sets $V_1, \ldots, V_p$ satisfying:*

$$V_s \cap V_t = \emptyset \text{ for } 1 \leq s < t \leq p \qquad \text{and} \qquad \cup_{s=1}^{p} V_s = V.$$

Note that p is not fixed. *The* incidence vector *of this clustering is defined by*

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same set } V_s \\ 0 & \text{otherwise.} \end{cases}$$

Let Q be the set of incidence vectors of clusterings for K_n. Let edge (i, j) have weight w_{ij}. The **clustering problem** *for this set of edge weights is then*

$$\begin{array}{ll} \max & z := \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_{ij} \\ \textit{subject to} & x \text{ is the incidence vector of a clustering} \end{array}$$

The edge weights w_{ij} can be positive or negative. Valid triangle inequalities for this problem for three distinct vertices i, j, k are

$$x_{ij} + x_{jk} - x_{ik} \leq 1.$$

Let $U \subseteq V$ and let $k = |U|$. Assume $k \geq 5$ and k is odd. Let $E(U)$ be the edges with both endpoints in U. Let $C \subseteq E(U)$ be a Hamiltonian tour through the vertices U. Denote the order of the vertices in the tour as $i_1 - i_2 - \ldots - i_k - i_1$. Prove that the following constraint is valid for the clustering problem:

$$\sum_{e \in C} x_e - \sum_{e \in E(U) \setminus C} x_e \leq \frac{k-1}{2}. \qquad (*)$$

Valid triangle inequalities:

$$x_{i_k i_1} + x_{i_1 i_2} - x_{i_2 i_k} \leq 1$$

$$x_{i_{k-1} i_k} + x_{i_k i_1} - x_{i_{k-1} i_1} \leq 1$$

$$x_{i_s i_{s+1}} + x_{i_{s+1} i_{s+2}} - x_{i_s i_{s+2}} \leq 1 \quad \text{for } s = 1, \ldots, k-2$$

Valid bounds:

$$-x_{i_r i_t} \leq 0 \text{ if } |r-t| \geq 2$$

Add and divide by 2:

$$\sum_{e \in C} x_e - \frac{1}{2} \sum_{e \in E(U) \setminus C} x_e \leq k/2$$

C-G rounding gives $(*)$.

2. (30 points)
Solve the following binary knapsack problem *using branch-and-bound*:

$$\begin{array}{lrcrcrcrcl} \text{max} & 35x_1 & + & 27x_2 & + & 15x_3 & + & 8x_4 & & \\ \text{subject to} & 10x_1 & + & 9x_2 & + & 6x_3 & + & 4x_4 & \le & 16 \\ & & & & & x_1,\ x_2,\ x_3,\ x_4 & & & & \text{binary.} \end{array}$$

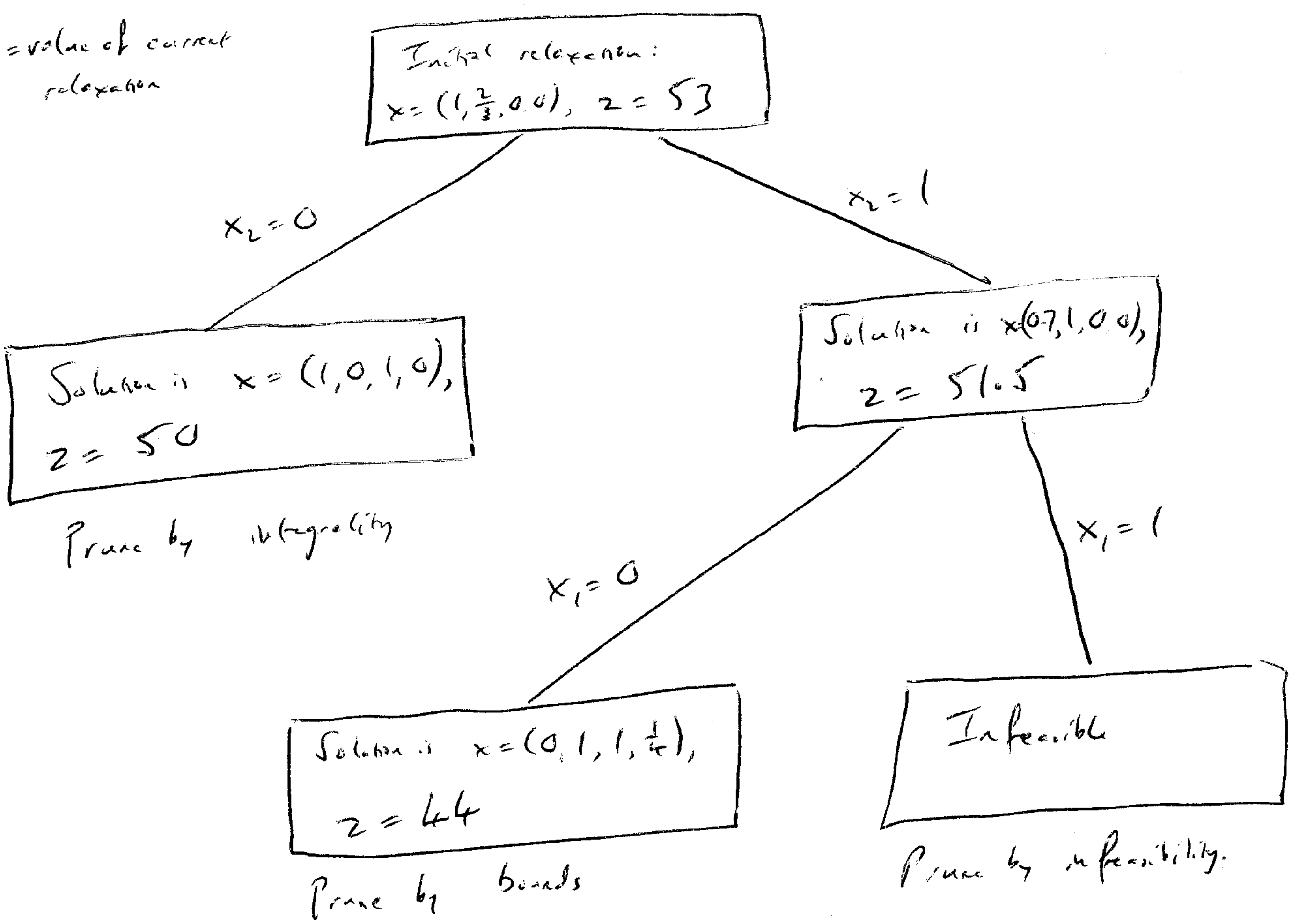
(Note: Linear programming relaxations of knapsack problems can be solved very easily. In particular, here the variables are ordered so that the ratios of the objective function coefficients c_i to constraint coefficients a_i satisfy

$$\frac{c_1}{a_1} \ge \frac{c_2}{a_2} \ge \frac{c_3}{a_3} \ge \frac{c_4}{a_4}.$$

Then the solution is to take x_1 as large as possible, then x_2 as large as possible, then x_3 as large as possible, and then x_4 as large as possible. For the initial LP relaxation, this gives a solution of $x = (1, \frac{2}{3}, 0, 0)$.

It may be useful to know that $0.7 \times 35 = 24.5$.)

Let z = value of current relaxation



Optimal solution: $x = (1, 0, 1, 0)$, value $= 50$.

3. (20 points)
The binary variables x_i, $i = 1, \ldots, 4$ satisfy

$$\begin{array}{lllllll} x_1 & + & x_2 & + & x_3 & & \leq 2 \\ x_1 & & & + & (1 - x_3) & + \; x_4 & \leq 2 \end{array}$$

The constraint $x_2 + x_4 \leq 2$ follows trivially from the upper bounds on the binary variables. By thinking of this as a constraint when $x_1 = 0$, lift it to give a stronger valid constraint of the form

$$\alpha x_1 + x_2 + x_4 \leq 2.$$

Lifting subproblem:

$\zeta = \max \quad x_2 + x_4$

s.t. $\quad 1 + x_2 + x_3 \leq 2$

$\quad 1 + (1-x_3) + x_4 \leq 2$

$\quad x_2, x_3, x_4$ binary

Two cases (i) $x_3 = 0 \Rightarrow x_4 = 0 \Rightarrow x_2 + x_4 \leq 1$

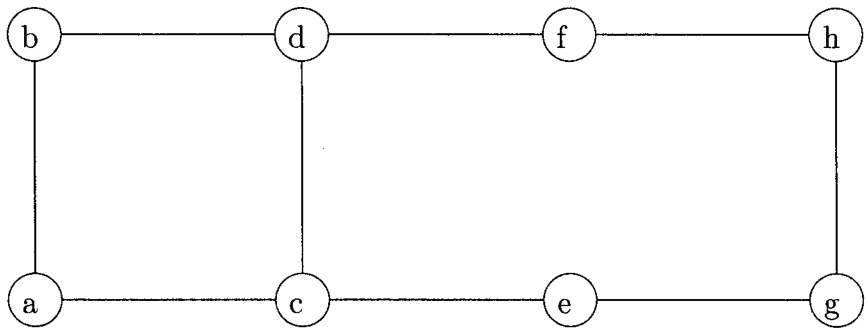
(ii) $x_3 = 1 \Rightarrow x_2 = 0$

$\Rightarrow x_2 + x_4 = 1.$

So $\zeta = 1$. So coefficient $\alpha = 2 - \zeta = 1$,

Cut constraint: $\boxed{x_1 + x_2 + x_4 \leq 2}$

4. (20 points)
   We want to solve the Max Cut problem on the graph $G = (V, E)$ has edge weights w_e. Part of the graph has the following structure:



Prove that the valid inequality

$$x_{ab} + x_{bd} + x_{df} - x_{fh} - x_{gh} - x_{eg} - x_{ce} - x_{ac} \leq 2 \qquad (*)$$

does not define a facet of the convex hull of the set of incidence vectors of cuts.

Have valid inequalities:

$$x_{ab} + x_{bd} + x_{cd} - x_{ac} \leq 2$$

$$x_{df} - x_{fh} - x_{gh} - x_{eg} - x_{ce} - x_{cd} \leq 0.$$

Adding gives $(*)$, so $(*)$ is implied by two other inequalities, so it cannot be facet defining.

5. (10 points)
Let c and a be integral positive n-vectors and let b be a positive integer. The knapsack problem $\max\{c^T x : a^T x \le b,\ x \in \mathbb{B}^n\}$ can be solved in time polynomial in n and b, yet the knapsack problem is NP-Hard. Why is this not a contradiction?

It requires $O(\log_2 b)$ ~~sto~~ space to store b.

So the running time is exponential in $\log_2 b$,

since $b = 2^{\log_2 b}$