NATIONAL UNIVERSITY OF SINGAPORE
MATHEMATICS SOCIETY

PAST YEAR PAPER SOLUTIONS
with credits to Zheng Shaoxuan

MA4254 Discrete Optimization
AY 2009/2010 Sem 1

Question 1

(i) Consider the following relaxed linear program (LP1) with slack variables added to the inequality constraints to form equality constraints, and with variables not necessarily integer:

$$\begin{array}{lll} \text{(LP1)} & \max & x_1 + x_2 \\ & \text{s.t.} & 7x_1 + x_2 + x_3 = 15 \\ & & -x_1 + x_2 + x_4 = 1 \\ & & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

We solve (LP1) by primal simplex.

	x_1	x_2	x_3	x_4	
$\bar{c}$	1	1	0	0	0
x_3	7	1	1	0	15
x_4	-1	1	0	1	1
$\bar{c}$	0	6/7	$-1/7$	0	$-15/7$
x_1	1	1/7	1/7	0	15/7
x_4	0	8/7	1/7	1	22/7
$\bar{c}$	0	0	$-1/4$	$-3/4$	$-9/2$
x_1	1	0	1/8	$-1/8$	7/4
x_2	0	1	1/8	7/8	11/4

An optimal solution of $(LP1)$ is $x = (7/4\ 11/4\ 0\ 0)^T$.
However, $x = (7/4\ 11/4)^T$ is not a feasible solution to (ILP).
Consider the first row of the final primal simplex tableau:

$$(1+0)x_1 + \left(0 + \frac{1}{8}\right) x_3 + \left(-1 + \frac{7}{8}\right) x_4 = \left(1 + \frac{3}{4}\right).$$

We obtain our new Gomory cut constraint:

$$-\frac{1}{8}x_3 - \frac{7}{8}x_4 \leq -\frac{3}{4},$$

and we now consider the following linear program (LP2), which consists of constraints from the final primal simplex tableau together with our new Gomory cut constraint:

$$\begin{array}{llllllll} \text{(LP2)} & \max & x_1 + x_2 & & & & & \\ & \text{s.t.} & x_1 & & +\frac{1}{8}x_3 - \frac{1}{8}x_4 & & = & \frac{7}{4} \\ & & & x_2 & +\frac{1}{8}x_3 + \frac{7}{8}x_4 & & = & \frac{11}{4} \\ & & & & -\frac{1}{8}x_3 - \frac{7}{8}x_4 & +x_5 & = & -\frac{3}{4} \\ & & x_1, x_2, x_3, x_4, x_5 \geq 0. & & & & & \end{array}$$

We solve (LP2) by dual simplex.

	x_1	x_2	x_3	x_4	x_5	
$\bar{c}$	0	0	$-1/4$	$-3/4$	0	$-9/2$
x_1	1	0	$1/8$	$-1/8$	0	$7/4$
x_2	0	1	$1/8$	$7/8$	0	$11/4$
x_5	0	0	$-1/8$	$-7/8$	1	$-3/4$
$\bar{c}$	0	0	$-1/7$	0	$-6/7$	$-27/7$
x_1	1	0	$1/7$	0	$-1/7$	$13/7$
x_2	0	1	0	0	1	2
x_4	0	0	$1/7$	1	$-8/7$	$6/7$

An optimal solution to (LP2) is $x = (13/7\ 2\ 0\ 6/7\ 0)^T$.

However, $x = (13/7\ 2)^T$ is not a feasible solution to ILP.

Consider the first row of our final dual simplex tableau:

$$(1+0)x_1 + \left(0 + \frac{1}{7}\right)x_3 + \left(-1 + \frac{6}{7}\right)x_5 = \left(1 + \frac{6}{7}\right).$$

We obtain our second Gomory cut constraint:

$$-\frac{1}{7}x_3 - \frac{6}{7}x_5 \leq -\frac{6}{7}.$$

(ii) We use (LP1) and the corresponding primal simplex tableau from (i). $x = (7/4\ 11/4)^T$ is not a feasible solution to (ILP) since x_1 is required to be integer.

Consider the row in the final primal simplex tableau which corresponds to x_1:

$$x_1 + \frac{1}{8}x_3 - \frac{1}{8}x_4 = \frac{7}{4}.$$

By using Gomory's MILP method (see (iii) for more details), we have $J^+ = \{3\}$, $J^- = \{4\}$, $\beta_k = \frac{3}{4}$, and our Gomory mixed cut constraint is:

$$\begin{aligned} -\frac{1}{8}x_3 - \frac{\frac{3}{4}}{\frac{3}{4}-1}\left(-\frac{1}{8}\right)x_4 &\leq -\frac{3}{4} \\ \Leftrightarrow \qquad -\frac{1}{8}x_3 - \frac{3}{8}x_4 &\leq -\frac{3}{4}, \end{aligned}$$

and we now consider the following linear program (LP3), which consists of constraints from the final primal simplex tableau together with our new Gomory mixed cut constraint:

$$\begin{array}{llllllll} \text{(LP3)} & \max & x_1 + x_2 & & & & & \\ & \text{s.t.} & x_1 & +\frac{1}{8}x_3 & -\frac{1}{8}x_4 & & = & \frac{7}{4} \\ & & x_2 & +\frac{1}{8}x_3 & +\frac{7}{8}x_4 & & = & \frac{11}{4} \\ & & & -\frac{1}{8}x_3 & -\frac{3}{8}x_4 & +x_5 & = & -\frac{3}{4} \\ & & \multicolumn{6}{l}{x_1, x_2, x_3, x_4, x_5 \geq 0.} \end{array}$$

We solve (LP3) by dual simplex.

	x_1	x_2	x_3	x_4	x_5	
$\bar{c}$	0	0	$-1/4$	$-3/4$	0	$-9/2$
x_1	1	0	$1/8$	$-1/8$	0	$7/4$
x_2	0	1	$1/8$	$7/8$	0	$11/4$
x_5	0	0	$-1/8$	$-3/8$	1	$-3/4$
$\bar{c}$	0	0	0	0	-2	-3
x_1	1	0	0	$-1/2$	1	1
x_2	0	1	0	$1/2$	1	2
x_4	0	0	1	3	-8	6

An optimal solution to (LP3) is $x = (1\ 2\ 6\ 0\ 0)^T$.

Since $x = (1\ 2)^T$ is feasible in (MILP), it is hence an optimal solution to (MILP).

(iii) Let x_k be an arbitary integer variable in (MILP). Using the usual notation, from the simplex tableau of the relaxed (LP), we obtain the following source constraint which must be satisfied for any feasible solution to (MILP):

$$x_k = \bar{a}_{k_0} - \sum_{j\in N} \bar{a}_{k_j} x_j.$$

Let $\beta_k = \bar{a}_{k_0} - \lfloor \bar{a}_{k_0} \rfloor$, $J^+ = \{j \in N | \bar{a}_{k_j} \geq 0\}$ and $J^- = \{j \in N | \bar{a}_{k_j} < 0\}$. The above source constraint can be rewritten as:

$$x_k - \lfloor \bar{a}_{k_0} \rfloor = \beta_k - \sum_{j\in N} \bar{a}_{k_j} x_j.$$

Since x_k is integer for any feasible solution to (MILP),

$$\begin{array}{rrcl}
 & x_k \leq \lfloor \bar{a}_{k_0} \rfloor & \text{or} & x_k \geq \lfloor \bar{a}_{k_0} \rfloor + 1 \\
\Rightarrow & \displaystyle\sum_{j\in N} \bar{a}_{k_j} x_j \geq \beta_k & \text{or} & \displaystyle\sum_{j\in N} \bar{a}_{k_j} x_j \leq \beta_k - 1 \\
\Rightarrow & \displaystyle\sum_{j\in J^+} \bar{a}_{k_j} x_j \geq \beta_k & \text{or} & \displaystyle\sum_{j\in J^-} \bar{a}_{k_j} x_j \leq \beta_k - 1 \\
\Rightarrow & \displaystyle\sum_{j\in J^+} \bar{a}_{k_j} x_j \geq \beta_k & \text{or} & \displaystyle\frac{\beta_k}{\beta_k - 1}\sum_{j\in J^-} \bar{a}_{k_j} x_j \geq \beta_k > 0 \\
\Rightarrow & \multicolumn{3}{l}{\displaystyle\sum_{j\in J^+} \bar{a}_{k_j} x_j + \frac{\beta_k}{\beta_k - 1}\sum_{j\in J^-} \bar{a}_{k_j} x_j \geq \beta_k} \\
\Rightarrow & \multicolumn{3}{l}{\displaystyle -\sum_{j\in J^+} \bar{a}_{k_j} x_j - \frac{\beta_k}{\beta_k - 1}\sum_{j\in J^-} \bar{a}_{k_j} x_j \leq -\beta_k \qquad ,}
\end{array}$$

the resulting Gomory mixed cut constraint, representing a necessary condition for feasiblity of solution in (MILP).

Question 2

(i) Any extreme point x to the feasible set of (1) has n linearly independent active constraints, p coming from $Ax = b$ and q coming from $x \geq 0$, with $p + q = n$.

We have the following corresponding linearly independent active vectors:

$$a^T_{p(1)}, \ldots, a^T_{p(p)}, e^T_{q(1)}, \ldots, e^T_{q(q)},$$

and the following square, non-singular matrix:

$$M = \begin{bmatrix} a_{p(1)}^T \\ \vdots \\ a_{p(p)}^T \\ e_{q(1)}^T \\ \vdots \\ e_{q(q)}^T \end{bmatrix}.$$

Since the n constraints are active, we have $Mx = \hat{b}$, where $\hat{b}$ is a submatrix of $(b\ 0)^T = (1\ 0\ \ldots\ 0)^T$. Since M is non-singular, $x = M^{-1}\hat{b}$.

Now, $\hat{b}$ cannot be the zero vector, otherwise $x = 0$ and $Ax = b$ implies that $b = 0$, a contradiction. Without loss of generality, $\hat{b} = (1\ 0\ \ldots\ 0)^T$.

By Cramer's Rule,

$$M^{-1} = \frac{\text{adj}(M)}{\det(M)}.$$

Since M is TU, $|\det(M)| = 1$ and cofactors of M are $\{-1, 0, 1\}$ scalars. Hence, adj(M) is a $\{-1, 0, 1\}$ matrix. This means that M^{-1} is a $\{-1, 0.1\}$ matrix.

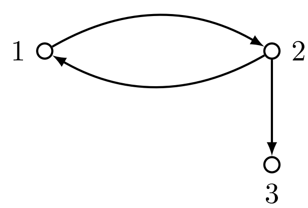
We have $x = M^{-1}\hat{b}$, which is just the first column of M^{-1}. So, x is a $\{-1, 0, 1\}$ vector. Since $x \geq 0$, we have x is a $\{0, 1\}$ vector.

(ii) No, not all optimal feasible solutions to (1) are optimal feasible solutions to (2). We construct the following counterexample:

Let c be the zero vector (possible since the question states that $c \geq 0$), and

$$\tilde{A} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix}, \tilde{b} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}.$$

It is easy to see that A is the node-arc incidence matrix of the following digraph:



(1) and (2) are now respectively the following linear programs:

$$\begin{array}{lll} (1) & \min & 0 \\ & \text{s.t.} & \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{pmatrix} x = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\ & & x \geq 0. \end{array}$$

$$\begin{array}{lll} (2) & \min & 0 \\ & \text{s.t.} & \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & -1 \end{pmatrix} x = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\ & & 0 \leq x \leq e. \end{array}$$

Consider $x^* = (2\ 1\ 1)^T$. x^* is feasible in (1) and hence optimal feasible in (1). But $x^* \not\leq e$, so x^* is not feasible in (2), and hence not optimal feasible in (2).

Question 3

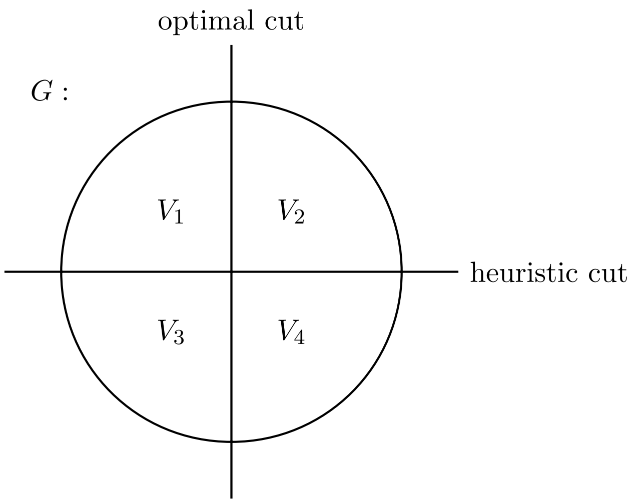
(i) The size of input in bits, L, is the number of bits used to represent G in terms of its vertex-edge incidence matrix, i.e., $L = O(|V||E|)$.

Observe that the size of the largest possible cut is $|E|$, so since each step of the algorithm strictly increases the size of the cut, hence the step will be repeated at most $|E|$ times.

For each step, given the current iteration of S and T, we first count the number of edges between S and T. This can be done in $O(|V||E|)$ bits by observing how S and T partitions the rows of the vertex-edge incidence matrix of G and counting how many columns have a vertex in S and a vertex in G. We next consider each individual vertex in G, to see if transferring it from S to T or vice versa increases the cut or not. We perform this at most $|V|$ times, and each time, counting the cut again takes $O(|V||E|)$ bits.

Hence the entire step takes $O(|V|^2|E|)$ bits. Since the step is repeated at most $|E|$ times, the algorithm takes $O(|V|^2|E|^2) = O(L^2)$ bits to execute.

(ii) Let (S, T) be the partition obtained from the algorithm and (S^*, T^*) be an optimum partition. Let $V_1 = S \cap S^*$, $V_2 = S \cap T^*$, $V_3 = T \cap S^*$ and $V_4 = T \cap T^*$. Let m_{ij} be the number of edges between V_i and V_j, for $i, j = 1, 2, 3, 4$.



By considering the fact that (S, T) cannot be improved by transferring a single vertex from one set to another, and by considering all vertices in V_1, we have:

$$2n_{11} + n_{12} \leq n_{13} + n_{14}.$$

This implies that $n_{12} \leq n_{13} + n_{14}$. Similarly, $n_{12} \leq n_{23} + n_{24}$, $n_{34} \leq n_{13} + n_{23}$, and $n_{34} \leq n_{14} + n_{24}$. By adding all these equations up and halving the result, we get

$$n_{12} + n_{34} \leq n_{13} + n_{14} + n_{23} + n_{24}.$$

Since $n_{14} + n_{23} \leq n_{13} + n_{14} + n_{23} + n_{24}$, so we have

$$n_{12} + n_{34} + n_{14} + n_{23} \leq 2(n_{13} + n_{14} + n_{23} + n_{24}).$$

Let τ be the size of the cut of (S, T) and τ^* be the size of the cut of (S^*, T^*). Observe that $\tau \leq \tau^*$, $\tau = n_{13} + n_{14} + n_{23} + n_{24}$ and $\tau^* = n_{12} + n_{34} + n_{14} + n_{23}$. From the above, we thus have $\tau^* \leq 2\tau$, which implies $\frac{1}{2}\tau^* - \tau \leq 0$, which implies that $(\tau^* - \tau) \leq \frac{1}{2}\tau^*$, and hence,

$$\frac{\tau^* - \tau}{\max\{\tau^*, \tau\}} \leq \frac{1}{2},$$

proving that the stated approximation algorithm is indeed a $\frac{1}{2}$-approximation algorithm for solving the maximum cut problem.

Question 4

(i) Given $a^1, \ldots, a^k \in \mathbb{R}$, let $P = \text{cone}\{a^1, \ldots, a^k\}$, $S = \text{span}\{a^1, \ldots, a^k\}$, and $k' = \dim(S)$.

Consider the intersection of the following half-spaces:

(a) Consider all possible $k' - 1$ linearly independent vectors in $\{a^1, \ldots, a^k\}$. For each of these, consider a hyperplane which passes through them, for which the half-space defined on one side of this hyperplane contains all of $a^1, \ldots, a^k$. For each case, if this hyperplane exists, then take the corresponding half-space into our consideration.

(b) If $k' < n$, then consider the $n - k'$ orthogonal hyperplanes which contains all of $a^1, \ldots, a^k$. Each of these hyperplanes is the intersection of two half-spaces, so take all of these $2(n - k')$ half-spaces into our consideration.

For any point $x \in P$, x is a non-negative linear combination of $a^1, \ldots, a^k$. Observe that x lies inside every half-space defined in (a), since x is also a linear combination of the corresponding $k' - 1$ vectors together with a positive sum of a vector linearly independent with these $k' - 1$ vectors. Also, observe that x lies in every half-space defined in (b).

For any point $x \notin P$, if x is not a linear combination of $a^1, \ldots, a^k$, then x does not lie in one of the hyperplanes defined in (b). Otherwise, take a set of linearly independent vectors $a^{i_1}, \ldots, a^{i_{k'}}$, where $a^{i_2}, \ldots, a^{i_{k'}}$ corresponds to one of the hyperplanes described in (a), and when x is expressed as a linear combination of $a^{i_1}, \ldots, a^{i_{k'}}$, the coefficient of a^{i_1} is negative. Observe that x does not lie in the corresponding half-space defined by this hyperplane.

Hence, P is an intersection of finitely many half-spaces and thus, is a polyhedral cone.

(ii) Consider $S = \{x \in \mathbb{R}^n \mid x_1 + \ldots + x_n = 1,\ x_i \geq 0,\ i = 1, \ldots, n\}$.

Any extreme point x^* of S has n linearly independent active constraints in S. We observe that selecting any n out of the $n+1$ constraints in S yields us a set of n linearly indepedent constraints.

Case (1). Suppose the only constraint not selected is $x_1 + \ldots + x_n = 1$. Then $x^* = 0$. However, this is not feasible in S. Hence, $x^* = 0$ is not an extreme point of S.

Case (2). Suppose the only constraint not selected is $x_i \geq 0$ for some $i \in \{1, \ldots, n\}$. Then x^* is such that $x_i = 1$, $x_j = 0$ for all $j = 1, \ldots, n$, $j \neq i$. We can check x^* is feasible in S. Hence, $e_1, \ldots, e_n$ are the extreme points of S.

Question 5

(i) We first model the stated problem into a minimum spanning tree problem by letting vertices represent stations, edges repesent links between stations, and edge costs representing rental costs of links.

Let M be a large number, larger than all rental costs of links. The problem now becomes:

$$
\begin{array}{rrl}
 & \min & \sum_{ij\in E} c_{ij}x_{ij} \\
 & \text{s.t.} & x \text{ represents a spanning tree} \\
\Leftrightarrow & \min & \sum_{ij\in E} ((c_{ij}-M)+M)x_{ij} \\
 & \text{s.t.} & x \text{ represents a spanning tree} \\
\Leftrightarrow & -\max & \sum_{ij\in E} ((M-c_{ij})-M)x_{ij} \\
 & \text{s.t.} & x \text{ represents a spanning tree} \\
\Leftrightarrow & -\max & \sum_{ij\in E} (M-c_{ij})x_{ij} + M(n-1) \\
 & \text{s.t.} & x \text{ represents a spanning tree}
\end{array}
$$

By replacing each link cost c_{ij} with $d_{ij} = M - c_{ij} > 0$, the minimum spanning tree problem is transformed into a maximum spanning tree problem.

(ii) We let $G = (V, E)$, $S = E$, $X \in \Xi \Leftrightarrow X \subseteq E$, X has no cycles. $M = (S, \Xi)$ is known as a graphic matriod.

For $X \subseteq E$, let $W(X) = \sum_{ij\in X} d_{ij}$. Then the maximum spanning tree problem can be formulated as:

$$
\begin{array}{rl}
\max & W(X) \\
\text{s.t.} & X \in \Xi
\end{array}
$$

(iii) Let $c_{max} = \max_{ij\in E}\{c_{ij}\}$. Then the size of input is $L = O(|V||E| + |E|\log|c_{max}|)$ bits.

The greedy algorithm for solving this matriod problem is as follows, assuming $W(e_1) \geq \ldots \geq W(e_n) \geq 0$:

- Step (0). Let $X = \emptyset$.
- Step (k). For k =1,...,n, if $X + e_k \in \Xi$, let $X := X + e_k$.

Step (k) is repeated $|E|$ times. In each execution of Step (k), checking $X + e_k \in \Xi$ takes $O(|V||E|)$ bits by observation of the vertex-edge incidence matrix, and letting $X := X + e_k$ is in constant number of bits. Hence, each iteration of Step (k) takes $O(|V||E|)$ bits.

So, the algorithm takes $O(|V||E|^2) = O(L^2)$ bits to execute.