15.083: Integer Programming and Combinatorial Optimization
Problem Set 2 Solutions

Due 9/30/2009

Problem (2.1) We are just checking to see if you demonstrated that you are comfortable with the methods that we covered.

Problem (2.9)

(a) We will separate into 2 cases:

C1: n even, n=2k+2:
We look at the subgraph on nodes $\{1, ..., n-1\}$. This is a complete graph on $2k+1$ nodes, so its edge-set may be written as the union of k edge-disjoint Hamiltonian tours: $E_{n-1} = \bigcup_{i=1}^{k-1} HC_i$. For each HC_i, we construct a n-1 distinct Hamiltonian tours on the complete graph of n nodes through the following procedure: for each edge $(u, v) \in HC_i$ we drop the edge (u, v) and replace it with the edges (u, n) and (v, n). This gives us a total of $(n-1) * k = (n-1) * (n-2)/2 = n * (n-1)/2 - n + 1 = d$ tours. Since each HC_i was edge disjoint, the incidence matrix of all our d tours contains a block diagonal $d \times d$ submatrix B in which the diagonal blocks can be rearranged by row and column operations to $E_{n-1} - I_{n-1}$ which is non-singular. Thus B is non-singular and our tours are linearly independent, thus they are affinely independent.

C2: n odd, n=2k+1:
We look at the subgraph on nodes $\{1, ..., n-1\}$. This is a complete graph on $2k$ nodes, so its edge-set may be written as the union of $k-1$ edge-disjoint Hamiltonian tours and one perfect matching: $E_{n-1} = PM \cup \bigcup_{i=1}^{k-1} HC_i$. For each HC_i, we construct a n-1 distinct Hamiltonian tours on the complete graph of n nodes through the same procedure as C1. This gives us $(n-1) * (k-1) = (n-1) * (n-2)/2 = n * (n-1)/2 - n + 1 - (n-1) = d - (n-1)$ linearly independent tours. We then order the edges on PM arbitrarily $\{(u_1, v_1), (u_2, v_2), ..., (u_k, v_k)\}$ and add the edges $\{(v_k, u_1)\} \cup \bigcup_{i=1}^{k-1}\{(v_i, u_{i+1})\}$ to obtain another tour HC_k on n-1 nodes. We create $n-1$ tours on n nodes from HC_k in the same manner as before. Now if we look at the incidence matrix of all our d tours, this matrix contains a a block triangular $d \times d$ submatrix B with $k-1$ blocks equivalent to $E_{n-1} - I_{n-1}$ and an additional block $E_k - I_k$.

(b) We will separate this problem into 3 cases:

C1: $n = 4, 5$: these cases can easily be checked graphically (if you omitted them for your proof not realizing that they are not covered by cases 2 and 3, you will still receive full credit).

C2: $n \geq 6, 3 \leq |S| \leq n/2$:
Suppose $S = \{1, 2, ..., k\}$. We will apply theorem A.2 in the book. We write the subtour elimination inequality as $fx \leq g$ and let $F = \{x \in conv(\mathcal{F}) : fx = g\}$, $(A^=, b^=)$ the equality set of $conv(\mathcal{F})$. Let $hx = d$ be any equality that holds for $x \in F$ To prove that F is a facet using this theorem, we need to find multipliers (α, u) such that $h = \alpha f + uA^=$ [note that this implies $d = \alpha g + ub$]. The first step in applying this theorem is to find certain components of h that we can fix the value of without loss of generality. To do this, we will select a set of columns of $A^=$ that is non-singular. If we take the edge set $E' = \{2, 3\} \cup \bigcup_{i=1}^{n}\{1, i\}$ and look at the columns of $A^=$ corresponding to the edges in E', we have such a

non-singular matrix. Thus there exists a multiplier $\bar{u}$ such that $\bar{h} = h + \bar{u}A^{=}$ where $\bar{h}_e = f_e \; \forall e \in E'$. Therefore, without loss of generality, we may restrict our attention to equalities $hx = d$ where:

$$\begin{aligned} h_{1i} = f_{1i} = 1 &\qquad i = 2, ..., k \\ h_{1i} = f_{1i} = 0 &\qquad i = k+1, ..., n \\ h_{23} = f_{23} = 1 & \end{aligned}$$

Now, if we pick any 2 tours x^1, x^2 in F, we must have that $hx^1 = hx^2$. We will use this property to derive the other components of h:

For any $i \in \{4, ..., k\}$ consider the tours:

$$\begin{aligned} x^1 &= (1, i, i-1, ..., 4, 2, 3, i+1, ...n) \\ x^2 &= (1, 2, 4, ..., i-1, i, 3, i+1, ...n) \end{aligned}$$

We have $0 = hx^1 - hx^2 = h_{1i} + h_{23} - h_{12} - h_{3i} = 1 - h_{3i} \Rightarrow h_{3i} = 1$. By iterating this argument we can obtain that $h_{ij} = 1 = f_{ij} \; \forall (i,j) \in E(S)$.

Next, for any $i \in \{k+1, ..., n\}$ consider the tours:

$$\begin{aligned} x^1 &= (1, 2, ..., i-1, i+1, ..., n, i) \\ x^2 &= (2, 1, ..., i-1, i+1, ..., n, i) \end{aligned}$$

We have $0 = hx^1 - hx^2 = h_{13} + h_{1i} - h_{23} - h2i = h_{2i} \Rightarrow h_{2i} = 0$. By iterating this argument we can obtain that $h_{ij} = 0 = f_{ij} \; \forall (i,j) \in \delta(S)$.

Finally, for any $i < j : i, j \notin S$, consider the tours:

$$\begin{aligned} x^1 &= (1, 2, ..., i-1, i+1, ..., j-1, j+1, ..., n, i, j) \\ x^2 &= (1, 2, ..., i-1, i+1, ..., j-1, j+1, ..., n, j, i) \end{aligned}$$

We have $0 = hx^1 - hx^2 = h_{in} + h_{1j} - h_{jn} - h1i = h_{in} - h_{jn} \Rightarrow h_{in} = h_{jn}$. By iterating this argument we can obtain that there exists some number γ:$h_{ij} = \gamma \; \forall (i,j) \in E(V \backslash S)$.

Together this implies that $hx = x(E(S)) + \gamma x(E(V \backslash S))$. Thus by picking $\alpha = 1 + \gamma$, $u_i = -\gamma/2$ for $i = 1, ..., k$ and $u_i = \gamma/2$ for $i = k+1, ..., n$ we obtain $h = \alpha f + uA^{=}$.

C3: $n \geq 6, |S| = 2$:

Suppose $|S| = \{n-1, n\}$. Then our subtour elimination constraint is equivalent to $x_{n-1,n} \leq 1$. We will use a similar construction to our proof of part (a), ensuring that the edge $(n-1, n)$ is included in each tour we construct. We again consider odd and even cases:

(i) $n = 2k+1$:

To ensure that $(n-1, n)$ is included in everything we construct, we begin with the subgraph on the nodes $\{1, ..., n-2\}$. The edgeset of this complete graph can be written as $E_{n-2} = \bigcup_{i=1}^{k-1} HC_i$.

For $HC_1 = (u_1, u_2, ..., u_{n-2})$, we construct 2(n-3) tours on n nodes by dropping each edge $(u_j, u_{j+1}) \in HC_1$ and replacing it with the edges $\{(u_j, n-1), (u_{j+1}, n), (n-1, n)\}$ and $\{(u_{j+1}, n-1), (u_j, n), (n-1, n)\}$ respectively to create 2 distinct tours containing $(n-1, n)$. We create one additional tour from HC_1 by dropping (u_{n-2}, u_1) and adding $\{(u_{n-2}, n), (u_1, n_{n-1}), (n-1, n)\}$. For HC_i $i = 2, ..., k-1$, we construct $n-2$ tours on n nodes by replacing each edge $(u, v) \in HC_i$ [labeled in such a way that when we look at HC_1 $u = u_{j_1}, v =_{u_{j_2}}$ with $j_1 < j_2$] by the edges $\{(u, n-1), (v, n), (n-1, n)\}$. This gives us a total of $2(n-3) + 1 + (k-2) * (n-2) = n * (n-1)/2 + n$ tours. Let B be the incidence matrix of the tours we created. We can use a similar approach to part (a) to prove that the rows of B are linearly independent, but here we will show you an alternative way based on linear programming duality. Now suppose that the rows of B are not linearly independent. Then $\exists \lambda \neq 0 : \sum_i B_i \lambda_i = 0$; for such a λ we have $\lambda_\tau \neq 0$ for some arbitrary tour τ. Then the following LP is unbounded for any choice of $\alpha \neq 0$:

$$\boxed{\begin{array}{l} \max \alpha \lambda_\tau \\ \text{subject to} \\ \quad \lambda' B \; = \; 0 \end{array}}$$

The above LP is unbounded if and only if its dual is infeasible:

$$\boxed{\begin{array}{l}\min 0\\ \text{subject to}\\ \quad By \;=\; \alpha e_\tau\end{array}}$$

We will find a feasible solution y to this dual proving linear independence. Let HC_i be the n-2 cycle used to construct τ and let (u,v) be the edge dropped. We set $y_{(w_1,w_2)} = \frac{1}{n-3}\ \forall (w_1,w_2) \in HC_i \backslash \{(u,v)\}$ and $y_{(u,v)} = -\frac{n-4}{n-3}$. If $i > 1$ or $i = 1$ and $(u,v) = (u_{n-2}, u_1)$, we only created one tour by dropping (u,v); thus setting $y_e = 0$ for all other edges is sufficient for a feasible solution with $\alpha = 1$. If $i = 1$ and $(u,v) = (u_j, u_{j+1})$ then we created 2 tours from dropping (u,v) from HC_1 by adding $\{(u_j, n-1), (u_{j+1}, n), (n-1, n)\}$ and $\{(u_{j+1}, n-1), (u_j, n), (n-1, n)\}$ respectively. If $\{(u_j, n-1), (u_{j+1}, n), (n-1, n)\}$ was added to make τ we set $y_{(w,n-1)} = -\frac{n-4}{n-3}\ \forall w = u_p, p > j$ and $y_{(w,n)} = -\frac{n-4}{n-3} \forall w = u_p, p < j+1$ and $y_{n-1,n} = \frac{n-4}{n-3}$ and for all remaining edges $y_e = 0$, we then have a feasible solution for $\alpha = 1 + \frac{n-4}{n-3}$. Similarly, if $\{(u_{j+1}, n-1), (u_j, n), (n-1, n)\}$ was added to make τ we set $y_{(w,n-1)} = -\frac{n-4}{n-3}\ \forall w = u_p, p \geq j, p \neq j+1$ and $y_{(w,n)} = -\frac{n-4}{n-3} \forall w = u_p, p \leq j+1, p \neq j$ and $y_{n-1,n} = \frac{n-4}{n-3}$, we then have a feasible solution for $\alpha = 1 + \frac{n-4}{n-3}$.

(ii) $n = 2k+2$:

If we look at the subgraph on $\{1, ..., n-2\}$, we have $E_{n-2} = PM \cup \bigcup_{i=1}^{k-1} HC_i$. For each HC_i, we obtain the tours the same way as in (i), plus we obtain an additional $\frac{1}{2}(n-2)$ tours by ordering the edges in PM arbitrarily and adding missing edges to create a new tour on $\{1, ..., n-2\}$. We then replace each edge in PM as we did for HC_i $i > 1$. Using the same methodology as (i), we can show that these tours are affinely independent. If the tour τ for which $\lambda_\tau \neq 0$ is constructed from PM by removing (u,v), we set $y_{(u,v)} = -\frac{k-2}{k-1}$ and $y_e = \frac{1}{k-1}\ \forall e \in PM \backslash \{(u,v)\}$.

Problem (2.15) Just as in the case of the comb inequality proof in the book, it is easier to examine the clique tree inequalities through the edge-set representation rather than the boundary representation. For a clique tree CT, to obtain the edgeset representation we note that since $\sum_{e \in E(H_i)} x_e + \frac{1}{2} \sum_{e \in \delta(H_i)} x_e = |H_i|$ and $\sum_{e \in E(T_i)} x_e + \frac{1}{2} \sum_{e \in \delta(T_i)} x_e = |T_i|$ the boundary representation is equivalent to:

$$\begin{aligned}
\sum_{i=1}^{h} \sum_{e \in \delta(H_i)} x_e + \sum_{i=1}^{t} \sum_{e \in \delta(T_i)} x_e &\geq 3t + 2h - 1 \\
&\Leftrightarrow \\
\sum_{i=1}^{h} \sum_{e \in E(H_i)} x_e + \sum_{i=1}^{t} \sum_{e \in E(T_i)} x_e &\leq \sum_{i=1}^{h} |H_i| + \sum_{i=1}^{t} |T_i| - \frac{2h + 2t - 2}{2} - \frac{t+1}{2} \\
&= \sum_{i=1}^{h} |H_i| + \sum_{i=1}^{t} (|T_i| - t_i) - \frac{t+1}{2}
\end{aligned}$$

Where t_i is this number of handles that intersect T_i [note that $\sum_i t_i = h + t - 1$ since we have an underlying tree structure]. We define the quantities

$$\begin{aligned}
lhs(CT) &= \sum_{i=1}^{h} \sum_{e \in E(H_i)} x_e + \sum_{i=1}^{t} \sum_{e \in E(T_i)} x_e \\
rhs(CT) &= \sum_{i=1}^{h} |H_i| + \sum_{i=1}^{t} (|T_i| - t_i) - \frac{t+1}{2}
\end{aligned}$$

We will prove that the clique tree inequalities are valid by induction on the number of handles. For 1 handle, this reduces to the comb inequality which is proved valid in the book.
Suppose that the clique tree inequalities are valid for trees with $h-1$ handles and let CT be a clique tree with h handles. W.l.o.g. let $T_1, ..., T_p$ be the teeth intersecting H_h. We now consider two collections of clique trees:

(i) $CT \backslash H_h = \bigcup_{i=1}^{p} \{CT'_i\}$

(ii) $(CT \backslash H_h) \bigcup_{i=1}^{p} (T_i \cap H_h) = \bigcup_{i=1}^{p} \{CT_i\}$

Note that each CT_i and CT_i' has at most $h-1$ handles. We have by construction:

$$\begin{aligned} \sum_{i=1}^{p} rhs(CT_i) &= rhs(CT) - |H_h| + \frac{p+1}{2} \\ \sum_{i=1}^{p} rhs(CT_i') &= \sum_{i=1}^{p} (rhs(CT_i) - |H_h \cap T_i|) \end{aligned}$$

Therefore, examining the original clique tree CT, by the induction hypothesis we have:

$$\begin{aligned} 2lhs(CT) &\leq \sum_{i=1}^{p} \left(lhs(CT_i) + lhs(CT_i') + \sum_{e \in E(H_h \cap T_i)} x_e \right) + \sum_{v \in H_h} \sum_{e \in \delta(v)} x_e \\ &\leq \sum_{i=1}^{p} (rhs(CT_i) + rhs(CT_i') + |H_h \cap T_i| - 1) + 2|H| \\ &= 2 \sum_{i=1}^{p} (rhs(CT_i)) + 2|H| - p \\ &= 2rhs(CT) + 1 \\ &\Rightarrow \\ lhs(CT) &\leq rhs(CT) + \tfrac{1}{2} \\ &\underbrace{\Rightarrow}_{rounding} \\ lhs(CT) &\leq rhs(CT) \end{aligned}$$

Problem (2.20) For this question, we are checking that you are comfortable with AMPL output and CPLEX/AMPL interaction. You should be able to explain that gap tolerance measures the distance between the objective function value of the best found integer solution and the objective value of the best bound (linear relaxation or dual). In CPLEX, there are 3 relevant values relating to this gap; absmipgap is the absolute value of this difference, relmipgap is absmipgap/objective function value of best integer solution, and mipgap is is the gap tolerance which tells CPLEX to stop when
$abs((best\ bound) - (best\ integer)) < mipgap * (1 + abs(best\ bound))$. The default mipgap tolerence is 10^{-4}.
The 1-knapsack instances should solve within a few seconds. You should be able to set the timelimit parameter to 120 seconds for the N-knapsack problem. You can use a combination of mipdisplay=1 and mipdisplay=5 to count improving integral solutions and the number of nodes remaining.
With cover cuts disabled 2-knapsack should take much longer than 1-knapsack and 4-knapsack should timeout after finding 1 integer solution (since you are using random data, it is possible to find a number different than 1, but is unlikely). After cover cuts are enabled, CPLEX should prove optimality of the second integer solution it finds quickly.

Problem (Longest Path) Hamiltonian Circuit is a restriction of Hamiltonian path with $K = |V|$.

Problem (EXACT COVER BY 4-SETS) This problem is in NP since C' provides a certificate for a YES instance that can be checked in polynomial time. We use local replacement to transform 3DM to EXACT COVER BY 4-SETS. Given an instance of 3DM, w.l.o.g. assume $(X \cup Y \cup X) \cap \{j \in \mathbb{N} : 1 \leq j \leq q\} = \emptyset$. We create the set $\bar{X} = W \cup Y \cup X \cup \{j \in \mathbb{N} : 1 \leq j \leq q\}$. Let $m_i = (w_i, x_i, y_i)$ be the i^{th} element of M with $|M| = m$. We define $C = \bigcup_{j=1}^{q} \bigcup_{i=1}^{m} \{\{m_i, x_i, y_i, j\}\}$. Note that $|\bar{X}| = 4q$ and C is a collection of 4-sets.
Suppose our 3DM matching is a YES instance and let M' be a matching. Then each element in M', call it $m_i = (w_i, x_i, y_i)$ has exactly one distinct coordinate of W, X and Y. So if we take $C' = \{\{w_i, x_i, y_i, i\}\}$ we have an exact cover and thus a YES instance of EXACT COVER BY 4-SETS.
Suppose we have a YES instance of EXACT COVER BY 4-SETS and let C' be an exact cover. The we have $|C'| = q$ and each element of C', $C_i' = \{w_i, x_i, y_i, j_i\}$, contains exactly one distinct member of $\{j \in \mathbb{N} : 1 \leq j \leq q\}$. Thus each element of C' must contain exactly one element from each of W, Y and X by construction. Therefore if we take $M' = \{(w_i, x_i, y_i) | i = 1, .., q\}$ we have a matching and a YES instance of 3DM.