Homework 3
IE418—Integer Programming
Dr. Ralphs
Due November 4, 2014

1. Let

\[ S = \{ x \in \mathbb{B}^N \mid \sum_{j \in N} a_j x_j \leq b, \sum_{j \in Q_i} x_j \leq 1 \forall i \in I \} \]  

with \( N = \bigcup_{i \in I} Q_i, a_j \in \mathbb{R}_+ \) for \( j \in N, \) and \( b \in \mathbb{R}_+ \)

(a) Show that if \( C \) is a minimal dependent set with respect to the knapsack constraint such that \( |C \cap Q_i| \leq 1 \) and \( C \cap Q_i = \{ j(i) \} \) when \( C \cap Q_i \neq \emptyset; \) and

\[ \tilde{E}(C) = E(C) \cup \{ j \in Q_i \mid a_j \geq a_j(i) \} \]  

then \( \sum_{j \in \tilde{E}(C)} x_j \leq |C| - 1 \) is a valid inequality for \( S. \)

(b) Specify conditions under which this valid inequality is facet-defining for \( \text{conv}(S). \)

2. Consider the following knapsack set.

\[ S = \{ x \in \mathbb{B}^{10} \mid 35x_1 + 27x_2 + 23x_3 + 19x_4 + 15x_5 + 15x_6 + 12x_7 + 8x_8 + 6x_9 + 3x_{10} \leq 39 \} \]

Find at least three facets of the convex hull of \( S \) by lifting the bound constraints optimally. Do these facets have any recognizable structure. What are the magnitudes of the coefficients?

3. Consider the set

\[ T = \left\{ x \in \mathbb{B}^n, y \in \mathbb{R}_+^n \mid \sum_{j \in N} y_j \leq b, l_j x_j \leq y_j \leq a_j x_j \forall j \in N \right\}, \]

where \( l_j, a_j \geq 0 \) are given bounds for each \( j \in J. \) Now consider the set

\[ T' = \left\{ x \in \mathbb{B}^n, y \in \mathbb{R}_+^n \mid \sum_{j \in N} (y_j + p_j x_j) \leq B, y_j \leq m_j x_j \forall j \in N \right\}. \]

(a) Show that there is an affine mapping \( f \) such \( f(T) = T' \) for certain chosen values of the parameters \( B \) and \( p_j, m_j \) for \( j \in N. \)

(b) Use this equivalence to derive valid inequalities for this problem.
4. Consider the linear ordering problem whose solution is a permutation of the set \( N = \{1, \ldots, n\} \). This problem can be formulated as

\[
\max \sum_{(i,j) \in N \times N} c_{ij} \delta_{ij} \\
\delta_{ij} + \delta_{ij} = 1 \quad \forall (i,j) \in N \times N \text{ with } i < j \\
\delta_{ii} = 0 \quad \forall i \in N \\
\delta_{j_1 j_2} + \cdots + \delta_{j_r j_1} \leq |C| - 1 \quad \forall C = \{j_1, \ldots, j_r\} \subset N \\
\delta \in \mathbb{B}^{N \times N}
\]

(a) Prove this formulation is correct (how do you do the mapping?)
(b) Show that the cycle-breaking inequalities with \(|C| \geq 4\) are unnecessary in the formulation.
(c) (Extra credit) Show that for \(|C| = 3\), these inequalities are facet-defining.

5. Using SYMPHONY, study the effect of generating valid inequalities (especially cover cuts) by solving the given instances with and without generating various classes of cuts. You may also experiment with parameters affecting the frequency of generation of cuts if you wish. Report on your findings. You will probably want to use performance profiles of both solution times and number of nodes generated, but I expect you to do some analysis of the results. Do not just say “turning off Gomory cuts resulted in an increase in solution times.” Say why you think this might have happened. How much difference was there? Was it uniform across all instances? Or were there certain instances more impacted than others? You can look at the structure of the instances to answer the questions (these are standard instances so you can probably find good information about them by searching on the Internet). Is there a theoretical reason you can give why certain classes might be ineffective for certain problems?

The tests are to be done with the instances available on the course Web site. To perform the experiments, you must use bash scripts to generate a condor submission file in order to run your job on the cluster. Afterwards, you will gather your results using a second set of scripts. The instructions are give in more detail on the course Web site.

I attempted to pick some instances that would not be too difficult to solve so the whole test set runs quickly, but with such moderately difficult instances, you can see anomalous behavior. You are welcome to try to choose some more difficult instances from MIPLIB if you like and I will perhaps provide a more difficult set in a few days for those who want something a bit more interesting. You may find more interesting results if you turn off the preprocessor, since the preprocessor already strengthens the reformulation to an extent. It would be interesting to see what happens in that case.