AMS 553.766: Combinatorial Optimization  
Homework Problems - Week VII

1. Let \( G = (V, E) \) be a graph. Consider variables \( x_e \) associated with each edge \( e \in E \). Let \( P(G) \) be the polytope given by:

\[
\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in V \\
x_e \geq 0 \quad \forall e \in E
\]

We showed in class that if \( G \) is bipartite, then all vertices of \( P(G) \) are integral by the TUM property of the constraint matrix. Show the converse that if \( G \) is not bipartite, then there exists a non integral vertex.

2. In class, we showed how the “minimum odd cut” problem can be used to implement a separation oracle for the perfect matching polyhedron, i.e.,

\[
\sum_{e \in \delta(v)} x_e = 1 \quad \forall v \in V \\
\sum_{e \in E[S]} x_e \leq \frac{|S|-1}{2} \quad \forall S \subseteq V \text{ such that } |S| \geq 3 \text{ and is odd} . \\
x_e \geq 0 \quad \forall e \in E
\]

This means that we can optimize over this polyhedron by making a polynomial number of queries to this oracle. Now we reduce a general maximum matching problem to a maximum perfect matching problem.

Design a procedure that takes as input an undirected graph \( G = (V, E) \) with arbitrary weights on the edges, and explicitly constructs a new graph \( G' = (V', E') \) with weighted edges such that the following all hold:

(a) \(|V'|\) and \(|E'|\) are both bounded by a polynomial in \(|V|\) and \(|E|\).
(b) Every matching in \( G \) corresponds to a perfect matching in \( G' \) of the same weight and vice versa.

Thus, given a graph \( G \) where we want to solve the maximum matching problem, one would run the perfect matching algorithm on \( G' \), and then return the corresponding matching in \( G \).

Bonus question: Can you see a relationship between the polytope obtained by the convex hull of matching vectors in \( G \) and the polytope obtained by the convex hull of perfect matching vectors in \( G' \)? Does a separation oracle for the perfect matching polytope for \( G' \) give a separation oracle for the matching polytope for \( G \)?

3. Given an undirected graph, a clique is a subgraph that is complete, i.e. all pairs of vertices in the subgraph have an edge between them. In other words, a clique is given by a subset of the vertices that are all connected to each other by edges.

Write an integer program to find the largest clique (in terms of number of vertices) in a graph \( G = (V, E) \). Let \( G' = (V', E') \) be the graph on the same set of vertices \( V \) such that \( ij \in E' \) if and only if \( ij \notin E \). Interpret the dual of the LP relaxation of the largest clique IP for \( G \) in terms of \( G' \).

4. A company sets an auction for \( N \) objects. Bidders place their bids for some subsets of the \( N \) objects that they like. The auction house has received \( n \) bids, namely bids \( b_j \) for subset \( S_j \), for \( j = 1, \ldots, n \). The auction house is faced with the problem of choosing the winning bids so that profit is maximized and each of the \( N \) objects is given to at most one bidder. Formulate the optimization problem faced by the auction house as an integer programming problem.

[Adapted from Problem 2.16 from the “Integer Programming” textbook by Conforti, Cornuéjols and Zambelli.]
5. Jobs \( \{1, \ldots, n\} \) must be processed on a single machine. Each job is available for processing after a certain time, called release time. For each job we are given its release time \( r_i \), its processing time \( p_i \) and its weight \( w_i \). Formulate as an integer linear program the problem of sequencing the jobs without overlap or interruption so that the sum of the weighted completion times is minimized.

[Problem 2.17 from the “Integer Programming” textbook by Conforti, Cornuérjols and Zambelli.]

6. A firm is considering project \( A, B, \ldots, H \). Using binary variables \( x_a, \ldots, x_h \) and linear constraints, model the following conditions on the projects to be undertaken.

(a) At most one of \( A, B, \ldots, H \).
(b) Exactly two of \( A, B, \ldots, H \).
(c) \( A \) or \( B \).
(d) \( A \) and \( B \).
(e) If \( A \) then \( B \).
(f) If \( A \) then not \( B \).
(g) If not \( A \) then \( B \).
(h) If \( A \) then \( B \), and if \( B \) then \( A \).
(i) If \( A \) then \( B \) and \( C \).
(j) If \( A \) then \( B \) or \( C \).
(k) If \( B \) or \( C \) then \( A \).
(l) If \( B \) and \( C \) then \( A \).
(m) If two or more of \( B, C, D, E \) then \( A \).
(n) If \( m \) or more than \( n \) projects \( B, \ldots, H \) then \( A \).

[Problem 2.19 from the “Integer Programming” textbook by Conforti, Cornuérjols and Zambelli.]

7. For the following subsets of edges of an undirected graph \( G = (V, E) \), we view the following sets as \((0, 1)\) vectors in \( \mathbb{R}^{|E|} \) in the standard way. Find an integer programming formulation and prove its correctness:

(a) The family of Hamiltonian paths of \( G \) with endnodes \( u, v \). (A Hamiltonian path is a path that goes exactly once through each node/vertex of the graph.)
(b) The family of all Hamiltonian paths of \( G \).
(c) The family of edge sets that induce a triangle of \( G \).
(d) Assuming that \( G \) has \( 3n \) nodes, the family of \( n \) node-disjoint triangles.
(e) The family of odd cycles of \( G \).

Research question: For each problem above, is it possible to find a formulation using polynomially many inequalities (in the size of the graph \( G \)), or show that no such formulation exists?

[Problem 2.21 from the “Integer Programming” textbook by Conforti, Cornuérjols and Zambelli.]

8. (Playing with \((0,1)\)-vectors). Find integer programming formulations for the following integer sets.
(a) The set of all $(0, 1)$-vectors in $\mathbb{R}^4$ except \[
\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.
\]

(b) The set of all $(0, 1)$-vectors in $\mathbb{R}^6$ except \[
\left\{ \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ \\
1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \\
0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\
\end{pmatrix} \right\}.
\]

(c) The set of all $(0, 1)$-vectors in $\mathbb{R}^6$ except all the vectors having exactly two 1s in the first 3 components and one 1 in the last 3 components.

(d) The set of all $(0, 1)$-vectors in $\mathbb{R}^n$ with an even number of 1s. You don’t have to find a system with $\text{poly}(n)$ inequalities.

(e) The set of all $(0, 1)$-vectors in $\mathbb{R}^n$ with an odd number of 1s. You don’t have to find a system with $\text{poly}(n)$ inequalities.

Research question: For problems (d) and (e), is it possible to find a formulation using $\text{poly}(n)$ many inequalities, or show that no such formulation exists?

[Adapted from Problem 2.27 from the “Integer Programming” textbook by Conforti, Cornuéjols and Zambelli.]