CS602 Combinatorial Optimization via Linear and Convex Programming **Jan-Apr 2022**

End-semester exam — Total Marks: 60

1. Perfect matching polytope for general graphs can be described by polynomially many linear constraints.
   False
2. Spanning trees in a graph satisfy the matroid property.
   True
3. A graph has a perfect matching if and only if the determinant of its adjacency matrix is nonzero.
   False
4. The Steiner tree LP works exactly for minimum spanning tree.
   False
5. There is no upper bound on the number of iterations in the primal dual bipartite matching algorithm that is independent of the edge weights.
   False
6. Feedback vertex set problem can be solved exactly in polynomial time.
   False
7. By adding new variables, spanning tree polytope can be described by polynomially many linear constraints.
   True
8. The LP formulations for Max flow and Min cut are duals of each other.
   True
9. If an LP has a totally unimodular constraint matrix, then all optimal solutions must be integral.
   False
10. Multiway cut problem is a generalization of (s,t)-minimnum cut problem.
    True
11. In the multiplicative weight update algorithm, if an expert makes a wrong prediction, their weight is increased by a constant multiplicative factor.
    False
12. In the facility location problem, we minimize total cost of opening facilities plus the cost of assigning clients to facilities.
    True
13. Max cut is an example of submodular function maximization.
    True
14. A graph which is a cycle of length 5 is a perfect graph.
    False

15. The matroid intersection problem reduces to bipartite matching.

    False

16. In the preemptive scheduling problem, the sum of all completion times can be minimized if we always schedule the job with shortest remaining processing time.

    True

17. In the bin packing problem, we can achieve minimum number of bins, if we keep filling bins one by one by going over items in decreasing order of their sizes.

    False

18. An optimal solution for the bounded degree spanning tree problem can be found via a greedy algorithm.

    False

19. For the scheduling problem on unrelated parallel machines, the natural linear program has an integrality gap of 2.

    False

20. For the minimum knapsack problem, the integrality gap in the natural linear program cannot be bounded by any constant.

    True

Que 21. Write an integer linear program for the following problem. Given a graph, color its vertices using three colors so as to minimize the number of edges whose two endpoints get the same color. For example, in the complete graph on 4 vertices, the minimum value is 1. This is because whatever coloring scheme we use, two vertices will get the same color and the edge between them will be counted.

Ans 21. We will take three variables for each vertex u, say $\alpha_u, \beta_u, \gamma_u$, one for each color. We will put a constraint that a vertex has exactly one of the three colors. We will also have edge variables x_e. We will add a constraint that will force x_e to be 1 if the two endpoints of e have the same color.

$$
\begin{aligned}
&\min \sum_{e \in E} x_e \\
&\text{subject to} \\
\alpha_u, \beta_u, \gamma_u \in \{0,1\} &\text{ for each } u \in V \\
\alpha_u + \beta_u + \gamma_u = 1 &\text{ for each } u \in V \\
x_e \in \{0,1\} &\text{ for each } e \in E \\
x_e \geq \alpha_u + \alpha_v - 1 &\text{ for each edge } e = (u,v) \\
x_e \geq \beta_u + \beta_v - 1 &\text{ for each edge } e = (u,v) \\
x_e \geq \gamma_u + \gamma_v - 1 &\text{ for each edge } e = (u,v)
\end{aligned}
$$

Que 22. We want to minimize a linear function $f(x,y)$ over the set $\{(x,y) : x^2 + y^2 \leq 1, x \geq 0\}$. In other words, the feasible region is all the points in the disk of radius 1, centered at origin, whose x-coordinate is non-negative. Express this problem as a semidefinite program.

Hint: you may need the PSD constraint on a 2 by 2 or a 3 by 3 matrix.

Ans 22. There are multiple ways to express this as a semidefinite program. Here is one way.

$$\min f(x, y) \text{ subject to}$$

$$\begin{pmatrix} x & 0 & 0 \\ 0 & 1-x & y \\ 0 & y & 1+x \end{pmatrix} \succeq 0$$

Recall that positive semidefinite is equivalent to all principal minors being nonnegative. Hence, the above constraint is equivalent to

$$\begin{aligned} x &\geq 0 \\ 1-x &\geq 0 \\ 1+x &\geq 0 \\ (1-x)(1+x) - y^2 &\geq 0 \end{aligned}$$

This is equivalent to

$$\begin{aligned} x &\geq 0 \\ x &\leq 1 \\ x^2 + y^2 &\leq 1 \end{aligned}$$

Que 23. Consider the minimum weight Steiner tree problem with four terminals. That is, we want a Steiner tree that connects all four terminals with each other. Recall the primal dual algorithm for the Steiner tree problem discussed in the class. Show that the algorithm gives a solution with approximation factor 3/2.

Hint: look closely into the analysis done for the approximation factor

Ans 23. Recall the analysis done in the class. In an iteration when there are q active sets and all the active dual variables are increased by ϵ, the change in the dual objective was shown to be $\Delta D = q\epsilon$, while the change in the primal objective was shown to be at most $\Delta P = 2(q-1)\epsilon$. Hence the ratio of the change in primal and change in dual is upper bounded as

$$\frac{\Delta P}{\Delta D} \leq \frac{2(q-1)}{q} = 2 - 2/q.$$

The number q of active sets can be different in different iterations, but we claim that q is always bounded by 4. This is because by definition, active set means it must separate some terminal from another terminal. Since there are only four terminals, and the active sets are disjoint, the number of active sets can be at most 4. Hence,

$$\frac{\Delta P}{\Delta D} \leq 2 - 2/q \leq 2 - 2/4 = 3/2.$$

Since this is true in every iteration, we can get the same bound on the final primal and dual objectives.

$$\frac{P}{D} \leq 3/2.$$

This means the algorithm gives a solution with approximation factor 3/2.

Que 24. To apply interior point methods for semidefinite programming, we need an appropriate barrier function. Let's consider the feasible region to be all $n \times n$ symmetric PSD matrices, i.e., $\{X \in \mathbb{R}^n : X \succeq 0\}$. Let's define the barrier function as

$$B(X) = -\log(\det(X)).$$

a) (5 marks) Argue that the barrier function takes finite values for points strictly inside the feasible region and goes towards infinity as we move towards the boundary.

b) (5 marks) Let W be a symmetric matrix with (i,j) entry being $w_{i,j}$. Let's say we want to minimize a combination of a linear function (given by W) and the barrier function, say

$$B(X) + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} x_{i,j},$$

where $x_{i,j}$ is the (i,j) entry in the matrix X. Prove that the function is minimized at $X = W^{-1}$. If it helps, you can just prove this for $n = 3$and get full marks.

Hint: You might need to use the definitions of determinant and inverse in terms of minors. Let's say $W_{i,j}$ is the submatrix of W obtained by deleting row i and column j. Then,

- (i,j) entry of W^{-1} is $(-1)^{i+j} \det(W_{j,i}) / \det(W)$.
- for any i, we have $\det(W) = \sum_{j=1}^{n} (-1)^{i+j} w_{i,j} \det(W_{i,j})$.

Ans 24. a) Recall that an equivalent characterization of PSD matrices is that all eigenvalues are non-negative. For X strictly inside the feasible region, the eigenvalues of X are all strictly positive and hence the determinant is positive. Hence, $B(X)$ is finite. On the other hand, when we move towards the boundary, one of the eigenvalues of X moves towards zero. This means $\det(X)$, which is product of eigenvalues, moves towards zero. Then $-\log(\det(X))$ will move towards $+\infty$.

b) Recall that the barrier function is required to be convex. Indeed, $B(X) = -\log(\det(X))$ turns out to be convex. This was not required to be proved. Assuming that $B(X)$ is convex, its combination with a linear function

$$\phi(X) = B(X) + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} x_{i,j}$$

will remain convex. Hence, to find the minimizing point, we need to put the gradient to zero. We have $n(n+1)/2$ variables, we will take partial derivative with respect to each of them.

$$\frac{\partial B(X)}{\partial x_{i,j}} = \frac{-1}{\det(X)} \frac{\partial \det(X)}{\partial x_{i,j}} = \begin{cases} -\det(X_{i,i})/\det(X) & \text{if } i = j \\ -2(-1)^{i+j} \det(X_{i,j})/\det(X) & \text{if } i \neq j \end{cases}$$

Here $X_{i,j}$ is the submatrix of X after removing row i and column j. The factor 2 comes because $x_{i,j}$ appears twice in X, at (i,j) and at (j,i). Similarly, for the linear function $L(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} x_{i,j}$, we have

$$\frac{\partial L(X)}{\partial x_{i,j}} = \begin{cases} w_{i,i} & \text{if } i = j \\ 2w_{i,j} & \text{if } i \neq j \end{cases}.$$

Again factor 2 comes because $x_{i,j} = x_{i,j}$. We need to put $\frac{\partial \phi(X)}{\partial x_{i,j}} = 0$, which will imply

$$w_{i,j} = (-1)^{i+j} \det(X_{i,j})/\det(X),$$

for each i, j. Observe that what we got is the matrix inverse formula (note that $X_{i,j} = X_{j,i}$). Hence,

$$W = X^{-1} \implies X = W^{-1}.$$

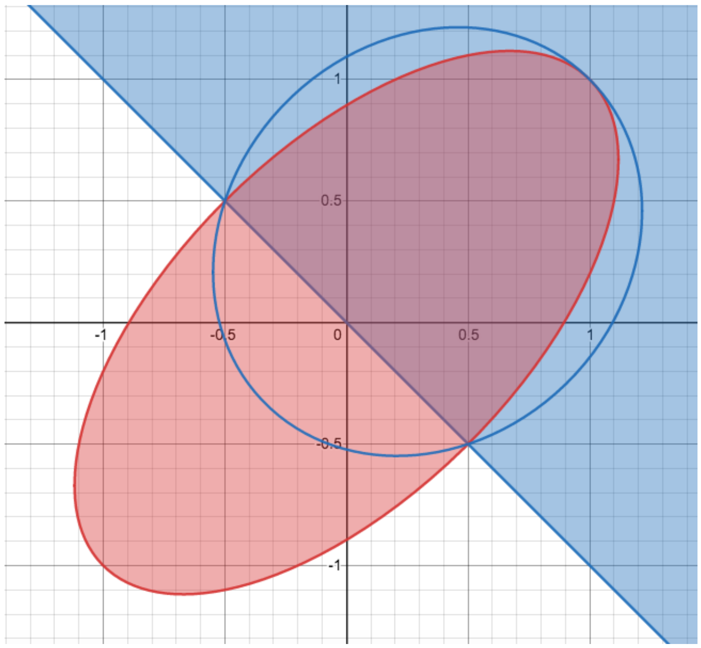

Figure 1: The given ellipse is shown in red, the halfspace in blue. Also shown the ellipse containing the half ellipse.

Que 25. Find the smallest ellipse which contains the following half ellipse

$$\{(x, y) : 5x^2 + 5y^2 - 6xy \leq 4, x + y \geq 0\}.$$

See Figure 1.

Ans 23. From the figure, we observe that the major and minor axes of the ellipse are along vectors $(1, 1)$ and $(1, -1)$. Hence, we must be able to rewrite the ellipse equation as

$$a(x + y)^2 + b(x - y)^2 \leq c.$$

Comparing the coefficients, we get $a = 1$, $b = 4$ and $c = 4$. The half ellipse constraints can be rewritten as

$$\frac{(x + y)^2}{2^2} + (x - y)^2 \leq 1, \; x + y \geq 0.$$

Now, the idea is to apply a linear transform to bring it to the standard form seen in the class. Set

$$u = (x + y)/2$$
$$v = (x - y)$$

The half ellipse description becomes

$$u^2 + v^2 \leq 1, u \geq 0.$$

We had seen in the class that the smallest ellipse containing this half ellipse is given by (we need to take the dimension $n = 2$)

$$\frac{(u - 1/3)^2}{4/9} + \frac{v^2}{4/3} \leq 1.$$

Equivalently,

$$(3u - 1)^2 + 3v^2 \leq 4.$$

Now, let us substitute $u = (x+y)/2$ and $v = x - y$. We get

$$(3x + 3y - 2)^2/4 + 3(x-y)^2 \le 4.$$

Or,

$$21x^2 + 21y^2 - 6xy - 12x - 12y + 4 \le 16.$$

Que 26. Consider the following polytope, where k is some natural number smaller than n.

$$0 \le x_i \le 1 \text{ for } 1 \le i \le n$$

$$\sum_{i=1}^{n} x_i \le k.$$

a) (5 marks) Argue that the polytope does not satisfy the non-degeneracy condition required in the simplex algorithm.

b)(5 marks) We can run the simplex algorithm even when non-degeneracy is not satisfied. The running time will depend on the maximum number of edges on any vertex. For the above polytope, prove that the maximum number of edges on any vertex is $O(n^2)$. For $n = 3$ and $k = 2$, the maximum number is 4.

Ans 26. a) Non-degeneracy requires that every vertex (or corner) of the polytope has exactly n tight constraints. Here we will see a vertex which has $n+1$ tight constraints. Consider the following point

$$(\underbrace{1, 1, \ldots, 1}_{k \text{ times}}, \underbrace{0, 0, \ldots, 0}_{n-k \text{ times}}).$$

Clearly this point is feasible. It's a corner because there are n linearly independent tight constraints for it, namely,

$$\begin{aligned} x_1 &= 1, \\ x_2 &= 1, \\ &\vdots \\ x_k &= 1, \\ x_{k+1} &= 0, \\ &\vdots \\ x_n &= 0. \end{aligned}$$

But, there is one more constraint which is tight

$$\sum_{i=1}^{n} x_i = k.$$

Hence, we get $n+1$ tight constraints for a vertex.

b) Recall that an edge of a polytope can be characterized by $n-1$ tight constraints. In particular, if a vertex has $n+1$ tight constraints, to get an edge on this vertex we can loosen 2 out of $n+1$ constraints and keep remaining $n-1$ constraints tight. There are $\binom{n+1}{2}$ ways of doing this. Hence, there can be at most $\binom{n+1}{2}$ edges on a vertex. Similarly if a vertex has n tight constraints then there are only $\binom{n}{1}$ edges on it.

But, can a vertex have more than $n+1$ tight constraints here? No, it is not possible because a point can satisfy only one of $x_i = 0$ and $x_i = 1$.

Que 27. Semidefinite programming also has a duality thoery. Like LPs, the duality thoery gives us a way of obtaining upper bounds on the primal objective. The following program turns out to be the dual of the MAXCUT SDP program we saw in the class. The program has n variables: for each vertex u , we have a variable y_u . w_e denotes the given weight of the edge e.

$$\min (1/2) \sum_{e \in E} w_e + (1/4) \sum_{u \in V} y_u$$

$$\text{subject to } W + diag(y) \succeq 0.$$

Here W is the symmetric matrix with (i, j) entry being the weight of edge (i, j). By notation $diag(y)$, we mean an n-by-n diagonal matrix with (u, u) entry being y_u. Prove that for any feasible solution y of the above program, the objective value is an upper bound on the weight of any cut.

Hint: for any cut, you will need to construct an appropriate vector, say c. Then use the fact that for any PSD matrix M , we have $c^T Mc \geq 0$.

Ans 27. Let $U \subseteq V$ be any cut. As mentioned in the hint, we will construct a vector $c \in \mathbb{R}^n$ as follows:

$$c_u = \begin{cases} 1 & \text{if } u \in U \\ -1 & \text{otherwise.} \end{cases}$$

Now, take any dual feasible solution y. We know that

$$W + diag(y) \succeq 0.$$

Let's multiply c on both sides. By PSDness, we know

$$c^T (W + diag(y))c \geq 0.$$

Let's expand the above expression.

$$\sum_{u \in V} \sum_{v \in V} w_{(u,v)} c_u c_v + \sum_{u \in V} y_u c_u^2 \geq 0.$$

Now, we use the fact that $c_u c_v = -1$ if (u, v) is a cut edge, i.e., $u \in U, v \notin U$ (or vice-versa). And otherwise $c_u c_v = 1$.

$$-2 \sum_{e \in \delta(U)} w_e + 2 \sum_{e \notin \delta(U)} w_e + \sum_{u \in V} y_u \geq 0.$$

The factor 2 comes because the edge (u, v) appears twice in the summation. Now, use the simple fact that $\sum_{e \in E} w_e = \sum_{e \in \delta(U)} w_e + \sum_{e \notin \delta(U)} w_e$.

$$2 \sum_{e \in E} w_e + \sum_{u \in V} y_u \geq 4 \sum_{e \in \delta(U)} w_e.$$

Dividing by 4,

$$(1/2) \sum_{e \in E} w_e + (1/4) \sum_{u \in V} y_u \geq \sum_{e \in \delta(U)} w_e.$$

Thus, the dual objective value is an upper bound on the weight of any cut.

Que 28. Given a graph $G(V,E)$, we want to find a subset of vertices U with exactly k vertices that maximizes the number of cut edges $\delta(U)$. We can write the following linear program similar to what we saw in the class.

$$\max \sum_{e\in E} x_e$$
$$\text{subject to}$$
$$0 \le z_u \le 1 \text{ for } u \in V$$
$$0 \le x_e \le 1 \text{ for } e \in E$$
$$\sum_u z_u = k$$
$$x_e \le z_u + z_v \text{ for each edge } e = (u,v)$$
$$x_e \le 2 - z_u - z_v \text{ for each edge } e = (u,v)$$

Suppose we find an optimal solution for this LP , say (x^*, z^*).

(a) (5 marks) For any z , let us define $f(z)$ as the expected number of edges in the cut obtained by the following randomized rounding: put a vertex u in U with probability z_u. Show that when we do rounding for the optimal z_u^* , then expected number of cut edges is at least 1/2 times the LP optimal value.

(b) (5 marks) Unfortunately, randomized rounding does not guarantee that we will have exactly k vertices in U. Here is a deterministic rounding scheme: prove that for any z, and for any two non-integral coordinates $0 < z_u, z_v < 1$, we can increase one of them by ϵ and decrease the other by ϵ, while guaranteeing that $f(z)$ does not decrease.

If you prove the above then the algorithm is clear. Keep picking a pair of non-integral coordinates, increase/decrease by ϵ so that one of the coordinates becomes integral. In the end, everything will become integral, i.e., 0 or 1. In the whole process, we have maintained $\sum_u z_u$ as a constant. Hence, finally we have exactly k vertices in U. Since we never decrease $f(z)$, we maintain 1/2 approximation.

Ans 28. (a) Let's first compute the probability that an edge $e = (u,v)$ will in the cut. It will be in the cut when exactly one of u and v is put into U. The probability that this happens is

$$z_u(1-z_v) + z_v(1-z_u) = z_u + z_v - 2z_u z_v.$$

Hence, the expected number of edges is

$$f(z) = \sum_{(u,v)\in E} (z_u + z_v - 2z_u z_v).$$

When using the optimal solution z^*, the expectation is

$$f(z^*) = \sum_{(u,v)\in E} (z_u^* + z_v^* - 2z_u^* z_v^*).$$

Now, we need to compare this to the LP optimal value

$$\sum_{e\in E} x_e^*.$$

We will do the comparison for each edge separately. From the constraints in the LP we can say,

$$x_e^* \le \min\{z_u^* + z_v^*, 2 - z_u^* - z_v^*\}.$$

Case 1: $z_u^* + z_v^* \le 1$ for some $e = (u,v)$. Then,

$$z_u^* + z_v^* \ge (z_u^* + z_v^*)^2 \ge 4z_u^* z_v^*.$$

The above uses AM-GM inequality. This implies

$$z_u^* + z_v^* - 2z_u^* z_v^* \geq (1/2)(z_u^* + z_v^*) \geq (1/2)x_e^*.$$

Case 2: $z_u^* + z_v^* \geq 1$ for some $e = (u, v)$. Then, $1 - z_u^* + 1 - z_v^* \leq 1$. In the above inequality, replace z_u^* by $1 - z_u^*$ and z_v^* by $1 - z_v^*$. We get

$$(1 - z_u^*) + (1 - z_v^*) - 2(1 - z_u^*)(1 - z_v^*) = z_u^* + z_v^* - 2z_u^* z_v^* \geq (1/2)(2 - z_u^* - z_v^*) \geq (1/2)x_e^*.$$

Adding the inequality for all edges, we have

$$f(z^*) = \sum_{(u,v)\in E} (z_u^* + z_v^* - 2z_u^* z_v^*) \geq (1/2) \sum_e x_e^*.$$

b) Let $N(u)$ be the set of neighboring vertices of u. When we change z_u by $+\epsilon$ and z_v by $-\epsilon$, the change in $f(z)$ will be

$$\sum_{u'\in N(u)} \epsilon(1 - 2z_{u'}) - \sum_{v'\in N(v)} \epsilon(1 - 2z_{v'}) + \underbrace{2\epsilon^2}_{\text{this term appears only if } (u,v) \text{ is an edge}}.$$

We can check the sign of $\sum_{u'\in N(u)}(1 - 2z_{u'}) - \sum_{v'\in N(V)}(1 - 2z_{v'})$. If it's positive, we can take ϵ to be positive. Otherwise, we can take ϵ to be negative. The last term ϵ^2 is always positive. Thus, we can ensure that the change in $f(z)$ is always nonnegative. Hence, $f(z)$ does not decrease.

The rest of the argument is explained in the question.