COMP 362 - Winter 2017 - Assignment 4

Due: 6pm Mar 23th

General rules: In solving these questions you may consult your book; you can discuss high level ideas with each other. But each student must find and write his/her own solution. You should drop your solutions in the assignment drop-off box located in the Trottier Building on the 2nd floor.

1. (10 Points) Show that the following problem is in PSPACE:

   - Input: An undirected graph G and a positive integer m.
   - Question: Is the number of proper m-vertex colorings of G divisible by $(m+1)$?

   Solution: One can generate all the possible m^n colorings, one by one (reusing the memory). Each such coloring takes $n\log(m)$ space. We will also have a variable A that is equal to the number of proper colorings found so far ($\mod m+1$). Every time that a new coloring is generated, we check whether it is proper or not, and then update the variable A accordingly. Note that A takes only $O(\log m)$ bits of memory. Hence in total the required space is going to be $O(n^2 + n\log m)$.

2. (15 Points) Problem 10 of Chapter 11: Suppose you are given an $n \times n$ grid graph G. Associated with each node v is an integer weight $w(v) \geq 0$. You may assume that all the weights are distinct. Your goal is to choose an independent set S of nodes of the grid, so that the sum of the weights of the nodes in S is as large as possible. (The sum of the weights of the nodes in S will be called its total weight.) Consider the following greedy algorithm for this problem.

   - Start with $S := \emptyset$.
   - While some node remains in G:
     - Pick a node v of maximum weight.

- Add v to S.
- Delete v and its neighbors from G

- Endwhile.

Show that this algorithm returns an independent set of total weight at least $\frac{1}{4}$ times the maximum total weight of any independent set in the grid graph G.

Solution: Since for every node v picked we remove the neighbors, the algorithm will not output any connected nodes thus the algorithm gives an independent set. Suppose that we pick a node v at some point in the algorithm. Let $v_1, \ldots, v_4$ be its neighbours. Note that none of $v_1, \ldots, v_4$ have been picked at this point (otherwise v would have been deleted). Since v has the maximum weight among the remaining vertices, we have $\text{weight}(v) \geq \text{weight}(v_i)$ for $i = 1 \ldots 4$. So

$$4 \times \text{weight}(v) \geq \sum_{i=1}^{4} \text{weight}(v_i).$$

If the optimal algorithm doesn't choose v and chooses a subset (or all four) of the neighbors instead, then it could be at most 4 times better.

3. (15 Points) Given a set P of n points on the plane, consider the problem of finding the smallest circle containing all the points in P. Show that the following is a 2-factor approximation algorithm for this problem. Pick a point x in P, and set r to be the distance of the farthest point in P from x. Output the circle centered at x with radius r.

   Solution: First we show that all the points will be in the circle outputted by algorithm: If y is the point farthest from x, then $r = d(x, y)$. If there's a point p outside the circle, then $d(p, x) > r = d(x, y)$ which cannot be because y is the farthest point from x.

   Now we show that this is a 2-factor approximation: Suppose x' and r' are the center and radius of the optimal solution. Since x and y are inside the circle we have $d(x, c) \geq r'$ and $d(y, c) \geq r'$ so $d(x, c) + d(y, c) \geq 2r'$. From the triangle inequality we have $d(x, c) + d(y, c) \geq d(x, y) = r$ from combining these two we have $2r' \geq r$.

4. Consider a directed bipartite graph $G = (V, E)$. We want to eliminate all the directed cycles of length 4 by removing a smallest possible set of <u>vertices</u>.

(a) (5 points) Let $\mathcal{C}_4$ denote the set of all cycles of length 4 in the graph. Show that the following integer program models the problem:

$$\begin{array}{lll} \min & \sum_{v\in V} x_v & \\ \text{s.t.} & \sum_{u\in C} x_u \geq 1 & \forall C \in \mathcal{C}_4 \\ & x_u \in \{0,1\} & u \in V \end{array}$$

Solution: For each vertex v, we have a variable x_v. These variables are $0/1$ valued. The meaning of $x_v = 1$ is that we remove vertex v from the graph. The meaning of $x_v = 0$ is that we keep vertex v. Let OPT denote the optimum value for the original problem. Let OPT_{ip} denote the optimum value for the integer program. Let x^* be an optimum solution of the integer program. By the inequality constraint, the integer program will pick at least one vertex from each 4-cycle. Thus removing the vertices corresponding to $x^* = 1$ will remove all the 4-cycles. Therefore we have $\text{OPT} \leq \text{OPT}_{ip}$. On the other hand, take a minimum set of vertices whose removal kills all the 4-cycles. Setting $x_v = 1$ for these vertices clearly produces a feasible solution for the integer program. Therefore $\text{OPT}_{ip} \leq \text{OPT}$.

(b) (5 points) Why does the optimal solution to the following relaxation provides a lower bound for the optimal answer to the above integer linear program? In other words why it is not necessary to have the constraints $x_u \leq 1$ in the relaxation?

$$\begin{array}{lll} \min & \sum_{v\in V} x_v & \\ \text{s.t.} & \sum_{u\in C} x_u \geq 1 & \forall C \in \mathcal{C}_4 \\ & x_u \geq 0 & \forall u \in V \end{array}$$

Solution: We claim that in any optimum solution x^*, $x_u^* \leq 1$ for all u. Suppose there exists some u such that $x_u^* > 1$. Round down the value of this variable to 1. Note that all the inequality constraints will still be satisfied. So we still have a feasible solution. On the other hand, the optimum value will go down, which is a contradiction.

(c) (15 points) Give a simple 4-factor approximation algorithm for the problem based on rounding the solution to the above linear program.

Solution: As before, let x^* be the optimum solution. The rounding is as follows. If $x_u^* \geq 1/4$, set $x_u^* = 1$, otherwise set $x_u^* = 0$.

First let's check that we get a feasible solution to our problem. In each inequality constraint, it must be the case that at least one of the variables has value $\geq 1/4$. Thus in our rounded solution, we pick at least one vertex from each 4-cycle. So we kill all the 4-cycles as required. Let OPT* be the optimum for the linear program, let OPT be the optimum for the original problem and let A be the value obtained by rounding the optimum of the linear program. Clearly OPT$^* \leq$ OPT. Also, by our rounding scheme, we have $A \leq 4$OPT*. Thus, $A \leq 4$OPT, i.e. our solution is within a factor 4 of the optimum.

(d) (15 points) Let L and R denote the set of the vertices in the two parts of the bipartite graph. (Every edge has one endpoint in L and one endpoint in R). Let x^* denote an optimal solution to the linear program in Part (b). We round x^* in the following way:

For every $u \in V$,

- if $u \in R$ and $x_u^* \geq 1/2$, set $\widehat{x}_u = 1$.
- if $u \in L$ and $x_u^* > 0$, set $\widehat{x}_u = 1$.
- Otherwise set $\widehat{x}_u = 0$.

Show that $\widehat{x}$ is a feasible solution to the integer linear program.

Solution: Observe that each 4-cycle contains two vertices from L and two vertices from R. Consider an inequality constraint of the linear program (so we are considering a fixed 4-cycle). If $x_u^* > 0$ for one of the two vertices in L, $\widehat{x}_u$ will be set to 1 and therefore this inequality will be satisfied. On the other hand, if $x_u^* = 0$ for both vertices in L, then it must be the case that $x_v^* \geq 1/2$ for one of the vertices in R. Thus this vertex will be rounded to 1 and the inequality will be satisfied.

(e) (10 points) Consider the dual of the relaxation:

$$\begin{array}{lll} \max & \sum_{C \in \mathcal{C}_4} y_C & \\ \text{s.t.} & \sum_{C \in \mathcal{C}_4, u \in C} y_C \leq 1 & \forall u \in V \\ & y_C \geq 0 & \forall C \in \mathcal{C}_4 \end{array}$$

and let y^* be an optimal solution to the dual. Use the complementary slackness to prove the following statement: For every $C \in \mathcal{C}_4$ either we have $|\{u : \widehat{x}_u = 1\}| \leq 3$ or $y_C^* = 0$.

Solution: Suppose $|\{u : \widehat{x}_u = 1\}| > 3$. Then all the variables for that cycle must be rounded to 1. For that to happen, it must be that $x_u^* \geq 1/2$ for the vertices in R and $x_u^* > 0$ for the vertices in

L. Thus, we must have $\sum_{u \in C} x_u^* > 1$, i.e. the constraint is not tight. By complementary slackness, this means $y_C^* = 0$.

(f) (10 points) Use the complementary slackness and the previous parts to show that our rounding algorithm is a 3-factor approximation algorithm.

Solution: As mentioned before, we have $\sum_{u \in V} x_u^* = \mathrm{OPT}^* \leq \mathrm{OPT}$. Thus, we are done once we show

$$\sum_{u \in V} \widehat{x}_u \leq 3\mathrm{OPT}^*.$$

Note that if $\widehat{x}_u = 1$, $x_u^* > 0$. Therefore, by complementary slackness, $\sum_{C \in \mathcal{C}_4, u \in C} y_C^* = 1$. The variables $\widehat{x}_u$ are 0/1 valued, so we can write

$$\sum_{u \in V} \widehat{x}_u = \sum_{u \in V} \widehat{x}_u \sum_{C \in \mathcal{C}_4, u \in C} y_C^* = \sum_{u \in V} \sum_{C \in \mathcal{C}_4, u \in C} \widehat{x}_u y_C^*.$$

We now change the order of the sums and get

$$\sum_{u \in V} \sum_{C \in \mathcal{C}_4, u \in C} \widehat{x}_u y_C^* = \sum_{C \in \mathcal{C}_4} \sum_{u \in C} \widehat{x}_u y_C^* = \sum_{C \in \mathcal{C}_4} y_C^* \sum_{u \in C} \widehat{x}_u.$$

From part (e) of the question, we know that if $y_C^* \neq 0$, then $\sum_{u \in C} \widehat{x}_u \leq 3$. Therefore the above quantity can be upper bounded by $3 \sum_{C \in \mathcal{C}_4} y_C^* = 3\mathrm{OPT}^*$ (the equality follows from duality). Putting things together, we have shown

$$\sum_{u \in V} \widehat{x}_u \leq 3\mathrm{OPT}^*$$

as required.