

Homework 1

Wenting LI

liw14@rpi.edu

September 12, 2017

1.

a.

Given a $(\Delta + 1)$ -colorable graph, where Δ is the maximum degree of any vertex in $G = (V, E)$,

1. Find the vertex v in G with the maximum degree Δ and its neighbor $N(v) = \{w | (v, w) \in E, i = 1, \dots, \Delta\}$;
2. First color v with one color, then color its neighbors $w_i \in N(v)$ one by one. If w_i is also connected to $w_j, i \neq j$, then color them with different colors. In this way, we need at most $\Delta + 1$ colors to color v and its neighbor $N(v)$;
3. Then find another v' not colored yet and its neighbor $N(v') = \{w | (v', w) \in E, i \leq \Delta\}$. Color them in the same way as step (2) using the $\Delta + 1$ colors;
4. Continue to color all other vertexes with less than $\Delta + 1$ colors until the graph G is completely colored.

As we can color G in poly-time with at most $(\Delta + 1)$ colors, then G is $(\Delta + 1)$ colorable.

b.

Bipartite graph is an undirected graph with nodes partitioned into group X and Y . For each line, its one ending node is in group X and the other ending node is in Y .

Then we can color all the nodes in group X with one color and the nodes in group Y with the other color. Therefore, bipartite graphs are 2-colorable.

c.

Given undirected 3-colorable graph $G = (V, E)$, $|V| = n$. For each $v \in V$ vertex, let $d(v)$ be the degree of v .

If $d(v) \geq \sqrt{n}$, then remove v and its neighbors $N(v) = \{w_i | (v, w_i) \in E, i \geq \sqrt{n}\}$, and then we can first assign v with the first color and then color those points that are only connected with v by the second color. Then remove all the points that are colored already together with the line between them.

If there are still points not colored, then these points can be colored with at most two colors, since G is a 3 colorable graph and all the remaining points are connected with v . Thus we can form a bipartite graph with them and color them with at most two colors. Specifically, for each of the remaining line, one ending point of the line is in one group while the other point is in the other group. Then color these two groups with two colors. As a result, we can color each v and its neighbor with at most 3 colors in this way in a poly-time.

We at most remove \sqrt{n} times. The explanation is following: each time we remove at most $\sqrt{n} + 1$ vertex from G , and there are n vertex in the graph, and thus we need to remove $O(\sqrt{n})$ times.

As we need at most 3 colors each time and totally $O(\sqrt{n})$ times, we need $O(\sqrt{n})$ colors to color those vertex having degree $\geq \sqrt{n}$ and their neighbors.

Until all the vertex in G have degree less than \sqrt{n} , we can apply the algorithm in (a) to color them with $O(\sqrt{n})$ colors with a poly-time.

In summary, we need $O(\sqrt{n})$ colors.

d.

Given undirected 4-colorable graph $G = (V, E)$, $|V| = n$. For each $v \in V$ vertex, let $d(v)$ be the degree of v .

If $d(v) \geq n^{\frac{2}{3}}$, then remove v and its neighbors $N(v) = \{w_i | (v, w_i) \in E, i \geq \sqrt{n}\}$, and we need remove $n^{\frac{1}{3}} = n/n^{\frac{2}{3}}$ times.

We first color v with one color. Then the subgraph of its neighbors is a 3 colorable graph since all its neighbors are connected with v and G is a 4 colorable graph.

According to the conclusion of (c) that the best known algorithm for coloring 3-colorable graphs uses $O(n^{0.19996})$ colors, we can color all the v and the subgraphs of their neighbors with $O(n^{0.19996} n^{\frac{1}{3}}) < O(\frac{2}{3})$ colors.

Until all the vertex in G have degree less than $n^{\frac{2}{3}}$, we can apply the algorithm in (a) to color them with $O(n^{\frac{2}{3}})$ colors with a poly-time.

In summary, we need $O(n^{\frac{2}{3}})$ colors.

e.

Claim: Suppose VERTEX COLORING is approximable to a factor better than $\frac{4}{3}k$ for a k -colorable graph when $P \neq NP$, then determining if a graph is 3-colorable is solvable in poly-time.

Proof: \Rightarrow

If VERTEX COLORING is approximable to a factor better than $\frac{4}{3}k$ when $P \neq NP$, then we can find an approximation algorithm that using less than $\frac{4}{3}k$ colors for a k -colorable graph.

For the 3-colorable graph ($k = 3$) and $\frac{4}{3}k = 4$, this approximation algorithm can determine whether this graph can be colored by less than 4 colors and thus it can determine whether this is a 3-colorable graph in poly-time.

This is contradict to the fact that determining if a graph is 3-colorable is NP-Complete problem and $P \neq NP$. Therefore, VERTEX COLORING is not approximable to a factor better than $\frac{4}{3}$ unless $P = NP$.

1.5

(a)

We will show that if any extreme point not in the set of $\{0, \frac{1}{2}, 1\}$, then it is not an extreme point, in other word, we will show that it can be expressed as $\lambda x^1 + (1 - \lambda)x^2$ for $0 < \lambda < 1$ and the x^1 and x^2 are distinct feasible solutions.

Proof: \Rightarrow

Suppose there is one extreme point $x_i \notin \{0, \frac{1}{2}, 1\}$, then there are three possibilities of the range of x_i .

- (1) If $x_i \in (0, \frac{1}{2})$ is an extreme point, then $x_j > \frac{1}{2}, \forall (i, j) \in E$, since $x_i + x_j \geq 1$. \Rightarrow we can find a point x_i^1

$$x_i^1 = x_i - \varepsilon \geq 0 \tag{1}$$

for some small positive number ε , and we can also find another point x_i^2

$$x_i^2 = x_i + \varepsilon \geq 0 \quad (2)$$

since $x_i \geq 0$ and $\varepsilon > 0$.

Accordingly, we can find the points x_j^1 and x_j^2

$$x_j^1 = x_j + \varepsilon \geq 0 \quad (3)$$

$$x_j^2 = x_j - \varepsilon \geq 0 \quad (4)$$

We can show that positive numbers x_i^1 and x_i^2 are distinct and feasible points, because

$$x_i^1 + x_j^1 = (x_i - \varepsilon) + (x_j + \varepsilon) = x_i + x_j \geq 1 \quad (5)$$

Thus x_i^1 is a feasible point. Similarly, x_i^2 is also a feasible point. As $\varepsilon > 0$, they are distinct.

Then we can demonstrate that x_i can be expressed by x_i^1 and x_i^2 as follows:

$$x_i = \lambda(x_i^1) + (1 - \lambda)x_i^2 \quad (6)$$

$$= \lambda(x_i - \varepsilon) + (1 - \lambda)(x_i + \varepsilon) \quad (7)$$

$$= x_i - \varepsilon\lambda + (1 - \lambda)\varepsilon \quad (8)$$

Let $-\varepsilon\lambda + (1 - \lambda)\varepsilon = 0$, then $\lambda = 0.5 \in (0, 1)$, therefore, x_i can be expressed by the two distinct feasible points;

- (2) If $x_i \in (\frac{1}{2}, 1)$ is an extreme point, then $x_j > 0, \forall (i, j) \in E$, then the x_i^1 and x_i^2 in (1) and (2) are still positive respectively, the points x_j^1 and x_j^2 in (3) and (4) are also positive. Meanwhile (5) still no less than 1 and thus x_i^1 and x_i^2 are feasible points. Similarly, we can express x_i with x_i^1 and x_i^2 according to the equations (6)-(8);
- (3) If $x_i > 1$, then let $x_j \geq 0, \forall (i, j) \in E$. Then we can find $\varepsilon > 0$ such that the $x_i^1 > 1$ and $x_i^2 > 1$ from (1) and (2), and then x_j^1 and x_j^2 can any number no less than 0 to make $x_i^1 + x_j^1 \geq 1$ and $x_i^2 + x_j^2 \geq 1$, thus $\exists x_i^1, x_i^2$ are feasible points, and then by applying (6)-(8), we can also find $\lambda = 0.5$ to express x_i by x_i^1 and x_i^2 .

Therefore these discussions indicate the contradictions to the assumption that x_i is an extreme point. Hence, all extreme points have the property that $x_i \in \{0, \frac{1}{2}, 1\}$.

(b)

I first explain the algorithm, then I will show that it is a $\frac{3}{2}$ -approximation algorithm.

My algorithm is as following:

1. Apply the LP solver to return the extreme points and find the extreme point that achieves the minimum value of the object function. Then all the points of the graph are assigned with the values in the set of $\{0, \frac{1}{2}, 1\}$;
2. Take all the points assigned with 1;
3. For the subgraph of the points assigned with $\frac{1}{2}$, we apply the poly-time algorithm to 4-color it. Among these four colors, find the one color that is assigned to the maximum number of points and the set of these points is denoted as S ;

4. Take those points in the set of $\bar{S} = \{i | x_i = \frac{1}{2} \text{ and } i \notin S\}$ and round each of them to be 1, in other words, we take the points assigned with the other three colors;
5. Then all these points taken in step 2 and 4 form our solution.

Then I will demonstrate that our solution is a $\frac{3}{2}$ -approximation algorithm.

Proof: \Rightarrow

1. As running the 4-color algorithm and the LP solvers can be finished in poly-time, this algorithm is a poly-time method;
2. As the extreme points have the property $x_i \in \{0, \frac{1}{2}, 1\}$, then the points taken in the Step 2 can cover those lines that one of its two points is assigned with 1; for those lines that two of their ending points are assigned with $\frac{1}{2}$, we apply the 4-color algorithm to them and take the points not belong to the set S . As each line is assigned with distinct colors, there is at least one color belonging to the three colors we take and thus our algorithm takes at least one of the points of each line. Therefore, our solution is a vertex cover;
3. We will show that our algorithm has a approximation factor $\frac{3}{2}$.

Let x_i^* be the extreme points from LP solver and achieve the minimum weight. Then all the points of the graph are assigned with the value in the set $\{0, \frac{1}{2}, 1\}$. Our solution take all the points assigned with 1, and the points in the set of $\bar{S} = \{i | x_i^* = \frac{1}{2} \text{ and } i \notin S\}$. If all the points assigned with $\frac{1}{2}$ form the set $N_{\frac{1}{2}} = \{i | x_i^* = \frac{1}{2}\}$, then the size of S is more than $\frac{1}{4}|N_{\frac{1}{2}}|$, otherwise the total number of the points colored by the 4 colors will be less than $|N_{\frac{1}{2}}|$. Then the weight of our solution W satisfy the following relations:

$$W = \sum_{x_i^*=1} w_i + \sum_{i \in \bar{S}} w_i \tag{9}$$

$$\leq (\sum_{x_i^*=1} w_i x_i^* + \frac{3}{4} \times 2 \sum_{i \in N_{\frac{1}{2}}} w_i) (\text{since } |\bar{S}| \leq \frac{3}{4}|N_{\frac{1}{2}}|) \tag{10}$$

$$\leq \frac{3}{2} (\sum_{x_i^*=1} w_i x_i^* + \sum_{i \in N_{\frac{1}{2}}} w_i) \tag{11}$$

$$= \frac{3}{2} \text{LP-OPT} \tag{12}$$

$$\leq \frac{3}{2} \text{OPT} \tag{13}$$

Therefore, our algorithm is a $\frac{3}{2}$ -approximation algorithm.