Assignment-01

December 13, 2015

1 Maximum Matchings for Vertex Cover

1.1 Understand the Definition

1.1.1 Q1. Consider the following graph. Is the set (A,B), (C,E), (D,I), (G,H) a matching? Is it maximal?

A1.The set is a matching. It is also a maximal matching.

1.1.2 Q2. Consider the following graph. Is the set A,B,C,E,D,I,G,H a vertex cover? Is it optimal?

A2. The set is a vertex cover. It isn't an optimal one since the set B,E,G,I can do better.

1.2 Correctness

1.2.1 Q3.a If the VC returned by the algorithm is not a correct vertex cover, what does this imply for the vertex cover?

A3.a This implies that there exists an edge (u, v) such that none of its vertices is inside the set generated by the maximal matching M.

1.2.2 Q3.b What does this imply for the maximal matching M?

A3.b This implies that the edge can be added to the maximal matching generated by the algorithm. This does conflict with the definition of maximal matching.

1.2.3 Q3.c What does this imply for the correctness of the algorithm?

A3.c The above conflict says that the set generated by the algorithm must be a correct vertex cover.

1.3 Approximation Guarantee

1.3.1 Q4.a What is the size of intersection between vertex set defined by two edges in M?

A4.a From the definition of matching we know that

$$|\{v_i, v_j\} \cap \{v_k, v_l\}| = 0$$

1.3.2 Q4.b What is the relation between the expression and M?

A4.b From the definition of matching we know that

$$\sum_{\{u,v\} \in M} |\{u, v\}| = 2|M|$$

1.3.3 Q4.c What is the size of intersection between the set defined by one edge in M and S?

A4.c From the definition of vertex cover we know that

$$|\{v_i, v_j\} \cap S| = 1$$

and

$$|\cup_{\{u,v\}\in M} \{u, v\} \cap S| = |M|$$

since

$$\cup_{\{u,v\}\in M}\{u, v\} \subseteq V$$

in which V is the set of vertex of the given graph G, so we also have

$$|M| \leqslant |S|$$

1.3.4 Q4.d What is the size of solution VC compared to M and an optimal vertex cover S?

A4.d From the above deduction we have

$$|S| \leqslant |VC| = 2|M| \leqslant 2|S|$$

1.4 Running Time Analysis

1.4.1 Q5.a Is it possible to compute a maximal matching in time *O(n+m)* in which *n* is the number of vertices and *m* is the number of edges?

A5.a Yes it's possible. The following algorithm can generate a maximal matching with the required time complexity.

```
In [1]: def compute_maximal_matching(V, adj_list):
            ''' Compute one maximal matching in O(n+m)
            :param V:
                     Map with vertex as key and bool flag as value.
                     All the values should be initialized to False
            :param adj_list:
                     Adjacency list of the graph.
                     It is implemented as a map with vertex as key and list as value
            :return maximal_matching:
                     Implemented as a list of vertex tuple
            '''
            # Initialize:
            maximal_matching = []
            # Compute one maximal matching:
            for one_vertex, flag in V.iteritems():
                if not flag:
                    V[one_vertex] = True
                    for another_vertex in adj_list[one_vertex]:
                        if not V[another_vertex]:
                            V[another_vertex] = True
                            maximal_matching.append((one_vertex, another_vertex))
            # Return result:
            return maximal_matching
```

1.4.2 Q5.b What's the overall complexity of the proposed algorithm?

A5.b The overall complexity of the proposed algorithm is O(1.5n+m)

1.5 Tightness

1.5.1 Q6.a What are the 2 possible maximal matchings?

A6.a

$$\{(v_0, v_1), (v_2, v_3)\}\&\{(v_1, v_2)\}$$

1.5.2 Q6.b Which is the maximal matching that will lead to an optimal vertex cover?

A6.b

$$\{(v_1, v_2)\}$$

1.5.3 Q6.b Which is the maximal matching that will lead to a factor 2 approximation?

A6.c

$$\{(v_0, v_1), (v_2, v_3)\}$$

2 Triangles of A Graph

2.1 Understand the Definition

2.1.1 Q1. Give an lower bound for the given expression.

A1 From the specification of S we know that

$$x_A + x_B + x_C \geqslant 1$$

2.1.2 Q2. Describe an integer program for the problem

A2 The required integer program is as follows:

$$Minimize : \sum\nolimits_{v \in V} x_v$$

$$s.t. \forall (a, b, c) \in T, x_a + x_b + x_c \geqslant 1$$

$$x_v \in \{0, 1\}, x_v = 1 \iff v \in S$$

2.2 Relaxation

2.2.1 Q3. Consider a linear relaxation of the integer program for the problem.

A3 The integer program in A.2 can be relaxed as follows

$$Minimize : \sum\nolimits_{v \in V} x_v$$

$$s.t. \forall (a, b, c) \in T, x_a + x_b + x_c \geqslant 1$$

$$x_v \in [0, 1]$$

The expression satisfies

$$max(x_a, x_b, x_c) \geqslant \frac{1}{3}$$

The expression's minimum value and its tight lower bound are both

$$\frac{1}{3}$$

2.3 Rounding

2.3.1 Q4. Describe a rounding procedure for the relaxation solution.

A4 The designed procedure for rounding is as follows

$$z_v = \begin{cases} 1, & \text{if } x_v^* \geqslant \frac{1}{3} \\ 0, & \text{otherwise} \end{cases}$$

2.4 Correctness

2.4.1 Q5.a Give a lower bound for the number of triangles in the given set.

A5.a It can be proved that the lower bound for the number of triangles in the set is 0.

2.4.2 Q5.b Show the contradiction.

A5.b Because it is a triangle, we have

$$x_a + x_b + x_c \geqslant 1$$

However, since the set is not a subset of S, we also have

$$x_a + x_b + x_c < 1$$

Because the two above-mentioned constraints conflict with each other, there does not exist a satisfactory value for

$$max(x_a, x_b, x_c)$$

to satisfy both constraints

2.4.3 Q5.c Prove the algorithm's correctness.

A5.c So the correctness of the proposed algorithm is proved.

2.5 Approximation

2.5.1 Q6.a By which factor can the value of the fractional solution be multiplied in the worst-case?

A6.a Because

$$z_v = \begin{cases} 1, & \text{if } x_v^* \geqslant \frac{1}{3} \\ 0, & \text{otherwise} \end{cases}$$

We have

$$z_v \leqslant 3x_v^*$$

So

$$\sum\nolimits_{v \in V} z_v \leqslant 3 \sum\nolimits_{v \in V} x_v^*$$

2.5.2 Q6.b Conclude about the approximation guarantee of the algorithm.

A6.b From the above derivation we have.

$$\sum\nolimits_{v \in V} x_v^* \leqslant OPT \leqslant \sum\nolimits_{v \in V} z_v \leqslant 3 \sum\nolimits_{v \in V} x_v^*$$

2.6 Running Time Analysis

2.6.1 Q7 What's the overall complexity of the algorithm.

A7 Firstly, we can find and list all triangles in a given graph in

$$O(|V|^2)$$

and the algorithm can scale to very large graph using distributed computing framework such as Hadoop MapReduce and Spark.

So the overall complexity of my algorithm will be

$$(|V|^2 + T)$$

2.7 Tightness

2.7.1 Q8 Give an example with 3 vertices that shows that the analysis is tight.

A8 Consider the graph G = (V, E) in which

V = {A, B, C}

E = {(A, B), (A, C), (B, C)}

The optimal fractional solution given by linear program relaxation is

$$x_a^* = x_b^* = x_c^* = \frac{1}{3}$$

The approximated integer solution given by the rounding procedure is

$$z_a = z_b = z_c = 1$$

And the loss objectives will satisfy

$$\sum\nolimits_{v \in V} z_v = 3 \sum\nolimits_{v \in V} x_v^* = 3$$

Which shows that the analysis is tight.