Chutong Wu; Shiyu Tu; Minghao Liu
MAT 160 Spring 2018
Project 3
25 May 2018

1.1.a Let $a_{i,j}$ be the length from i to j (directed). Define

$$x_{i,j} = \begin{cases} 1, i \to j \text{ is included in the path} \\ 0, \text{ otherwise.} \end{cases}$$

Then the model will be:

$$\begin{aligned} \text{Max. } & \sum_{i,j} a_{i,j} x_{i,j} \\ \text{s.t. } & \sum_i x_{i,j} = \sum_i x_{j,i} \leq 1, \quad \forall j \neq s, t \\ & \sum_j x_{s,j} = \sum_i x_{i,t} = 1 \\ & \sum_i x_{i,s} = \sum_j x_{t,j} = 0 \\ & x_{i,j} \in \{0,1\}, a_{i,j} \in \mathbb{R} \end{aligned}$$

1.1.b Let $a_{i,j}$ be the length from i to j (directed). Define

$$x_{i,j} = \begin{cases} 1, i \to j \text{ is included in the path} \\ 0, \text{ otherwise.} \end{cases}$$

Then the model will be:

$$\begin{aligned} \text{Max. } & f \\ \text{s.t. } & f \geq a_{i,j} x_{i,j}, \forall i, \forall j \\ & \sum_i x_{i,j} = \sum_i x_{j,i} \leq 1, \quad \forall j \neq s, t \\ & \sum_j x_{s,j} = \sum_i x_{i,t} = 1 \\ & \sum_i x_{i,s} = \sum_j x_{t,j} = 0 \\ & x_{i,j} \in \{0,1\}, a_{i,j} \in \mathbb{R} \end{aligned}$$

1.2. Let $c_{i,j}$ be the cost on the path $i - j$. Define:

$$x_{i,j,k} = \begin{cases} 1, & \text{goes from city i to city j on the k-th leg} \\ 0, & \text{otherwise} \end{cases}$$

The model will be:

$$
\begin{aligned}
\text{Min. } & \sum_{i,j,k} x_{i,j,k} c_{ij} \\
\text{s.t. } & \sum_{i,j,k} x_{i,j,k} = n \\
& \sum_{i,j} x_{i,j,k} = 1, \forall k \\
& \sum_{j,k} x_{i,j,k} \leq 1, \forall i \\
& \sum_{i,k} x_{i,j,k} \leq 1, \forall j \\
& \sum_{j} x_{j,i,k \text{ mod n}} = \sum_{j} x_{i,j,k+1 \text{ mod n}}, \forall k, \forall i \\
& x_{i,j,k} \in \{0,1\}, c_{i,j} \in \mathbb{R}
\end{aligned}
$$

2. Denote the graph by G, the set of all vertices by V and the set of all edges by E. Define the *matching* of G to be

$$M := \{e \in E : |e \sim v| \leq 1, \forall v \in V\}$$

where $u \sim v$ means the edge e is incident to the vertex v. M is the *maximal* if there's no matching M' with $M \subset M'$. Denote the size of a maximum matching by OPT_G. Define $V(G)$ to be the *vertex cover* of G if

$$\forall e \in E, \exists v \in V(G) \text{ s.t. } v \sim e.$$

2.a Let M be a matching and define $V(M)$ to be the set of endpoints of edges in M:

$$V(M) := \{v \in V : \exists e \in M \text{ s.t. } v \sim e\}.$$

FSC, suppose M is a maximal matching of G but $V(M)$ is a not a vertex cover of G. It follows that there exists at least one edge $e' \in E$ which is not incident to any vertex in $V(M)$, so e' is not in M. We add e' to M, denoted by M', it's clear that $M' \supset M$ is also a matching since e' does not share endpoints with $V(M)$. Hence contradiction.

2.b Let M be a maximal matching of G. Since each edge is incident with 2 vertices and cannot share any vertex with other edges in M,

$$|V(M)| = 2|M| \leq 2OPT_G.$$

2.d Assume G has n nodes and m edges. Consider only the edges first. If no two edges share the same vertex, then the running time for the greedy algorithm will be m. Now consider only vectices. Since each vertex can be incident to at most one edge and each edge is incident to 2 vertices, the running time for the greedy algorithm will be $\lceil \frac{n}{2} \rceil$. Notice that there are at most $n = 2m$ vertices in G, so the running time is dominated by m.

2.e

Consider the graph above, the greedy algorithm fails to generate the maximum matching when we start from the edge $b - c$ instead of $a - b$ or $c - d$. Although the greedy algorithm cannot always give us the maximum matching, we claim it yields a solution that has at least half as many edges as a true maximum matching. To prove the statement, assume the algorithm ends and the solution N has less than half as many edges as a true maximum matching M. We know that $N \not\subset M$, since if N is contained in M then the algorithm will keep generating new edges. Since $N \not\subset M$, both $M \setminus (M \cap N)$ and $N \setminus (M \cap N)$ are nonempty we we denote them by M' and N', respectively. We know that $|M'| \geq \frac{1}{2}|M|$ and $|N'| < \frac{1}{2}|M|$. Since both M' and N' are parts of a matching, edges in M' do not share vertices, as well as N'. Since $|M'|$ in strictly greater than $|N'|$, there exists at least one vertex $v' \in V(M')$ which is not incident with any edges in N', hence N. And we can add such $e' \in M$ with $e' \sim v'$ to N which leads to a contradiction.

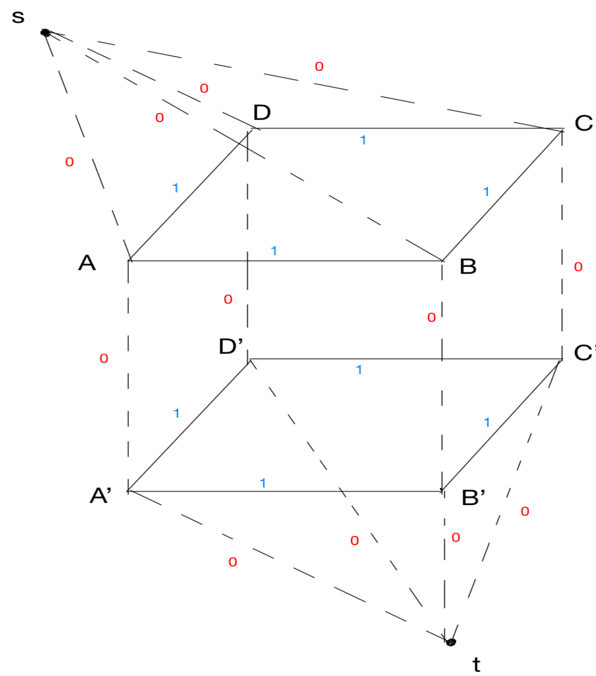
2.f Define $\delta(v) := \{e \in E : v \sim e\}$ to be the set of edges incident to a vertex $v \in V$. Define the binary variable x to be:

$$x_{i,j} = \begin{cases} 1, & \text{edge i-j is in M} \\ 0, & \text{otherwise.} \end{cases}$$

Then the model for the maximum matching problem will be:

$$\begin{aligned} \text{Max.} \ & \sum_{i,j} x_{i,j} \\ \text{s.t.} \ & \sum_{\{i,j\} \in \delta(v)} x_{i,j} \leq 1, \forall v \in V \\ & x_{i,j} \in \{0,1\} \end{aligned}$$

3.



Let G be a graph with two distinguished vertices s, t. Following from the hint, we make a copy of G, name it G', remove vertex t from G and vertex s from G', and joining every vertex v different from s, t with its copy in G'. Name the new graph H. Let $V(G)$ be to set of all vertices in $G \setminus \{s\}$, $V(G')$ the set of all vertices in $G' \setminus \{t\}$, and $E(G)$ the set of all edges of G. We first give weight to each edge: define

$$c_{i,j} = \begin{cases} 1, & \{i,j\} \in E(G) \text{ or } \{i,j\} \in E(G') \\ 0, & \text{otherwise.} \end{cases}$$

The graph above gives an example. Then we define the binary variable x:

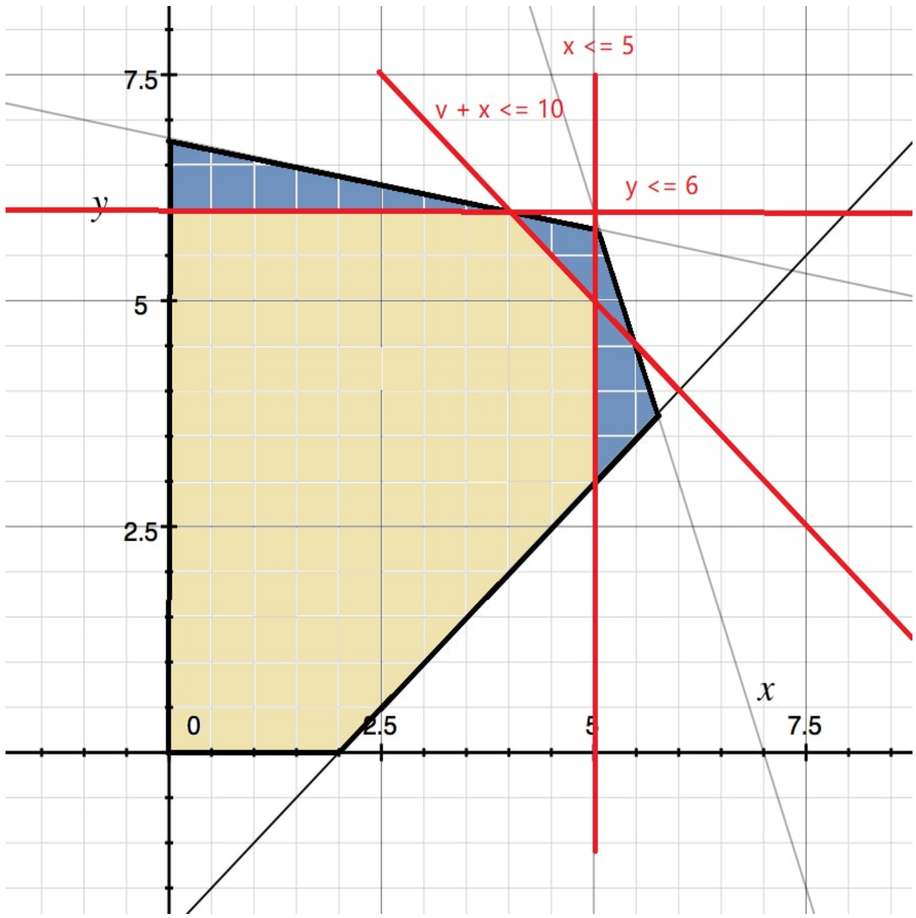
$$x_{i,j} = \begin{cases} 1, \text{ edge i-j is in the path} \\ 0, \text{ otherwise} \end{cases}$$

Then the model will be:

$$\begin{aligned} \text{Max.} \quad & \sum_{\{i,j\} \in E(G)} c_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_i x_{i,j} = \sum_i x_{j,i} \leq 1, \forall j \neq s, t \\ & \sum_{i \in V(G)} x_{s,i} = \sum_{j \in V(G')} x_{j,t} = 1 \\ & x_{i,j} \in \{0,1\}, c_{i,j} \in \mathbb{R}_{\geq 0} \end{aligned}$$

4.2.a Inequality description of convex hull of S:

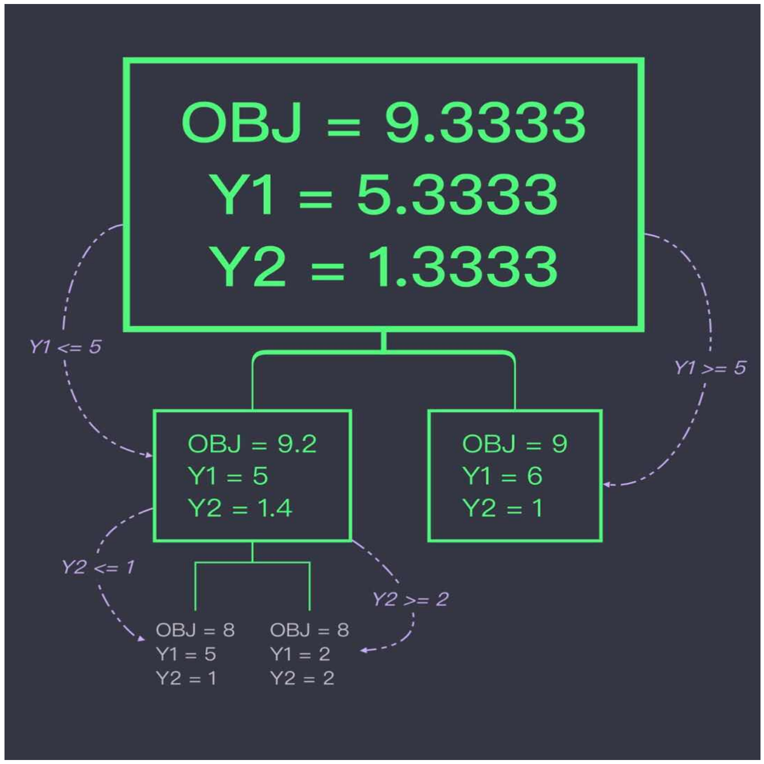
$$\begin{aligned} y_2 &\leq 6 \\ y_1 + y_2 &\leq 10 \\ y_1 &\leq 5 \end{aligned}$$



4.2.b Extreme points:

$$(0,6), (4,6), (5,5), (5,3), (2,0), (0,0)$$

4.3. The optimal solution is $y_1 = 6, y_2 = 1$ and the maximum value of the objective function is 9.



4.4. We first write inequalities in the linear programming as linear equations by adding slack variables y_4, y_5, and y_6. We get:

$$\text{Max } y_1 + 3y_2$$

$$\text{s.t.} \begin{bmatrix} 3 & 5 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 2 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 12 \\ 7 \\ 9 \end{bmatrix}, y \in \mathbb{Z}_{\geq 0} \tag{1}$$

Solving the linear system using SCIP, we get:

$$y_1 + 3y_2 = 10 \text{ and } y = (0, \frac{11}{3}, \frac{19}{3}, 0, \frac{2}{3}, 0)^T$$

which tells us that the nonbasic variables are y_2, y_3, y_5. We hence take the 2nd, 3rd, and 5th column of the first matrix in (1), compute its inverse and multiply the both sides of (1) on the L.H.S.. We get:

$$\begin{bmatrix} 5 & -1 & 0 \\ 0 & 1 & 1 \\ -1 & 2 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 3 & 5 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 2 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{7}{9} & 1 & 0 & \frac{2}{9} & 0 & \frac{1}{9} \\ \frac{8}{9} & 0 & 1 & \frac{1}{9} & 0 & \frac{5}{9} \\ \frac{1}{9} & 0 & 0 & -\frac{1}{9} & 1 & -\frac{5}{9} \end{bmatrix}$$

and

$$\begin{bmatrix} 5 & -1 & 0 \\ 0 & 1 & 1 \\ -1 & 2 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 12 \\ 7 \\ 9 \end{bmatrix} = \begin{bmatrix} \frac{11}{3} \\ \frac{19}{3} \\ \frac{2}{3} \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \frac{7}{9} & 1 & 0 & \frac{2}{9} & 0 & \frac{1}{9} \\ \frac{8}{9} & 0 & 1 & \frac{1}{9} & 0 & \frac{5}{9} \\ \frac{1}{9} & 0 & 0 & -\frac{1}{9} & 1 & -\frac{5}{9} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} \frac{11}{3} \\ \frac{19}{3} \\ \frac{2}{3} \end{bmatrix}.$$

We pick a row which its y_1, y_4 and y_6 are fractions. Wlog, we pick *row* 1. Then:

$$\begin{aligned} & \frac{7}{9}y_1 + y_2 + \frac{2}{9}y_4 + \frac{1}{9}y_6 = \frac{11}{3} \\ \Leftrightarrow \quad & y_2 - 3 = \frac{2}{3} - \frac{7}{9}y_1 - \frac{2}{9}y_4 - \frac{1}{9}y_6 \in \mathbb{Z}_{\geq 0} \\ \Leftrightarrow \quad & \frac{2}{3} - \frac{7}{9}y_1 - \frac{2}{9}y_4 - \frac{1}{9}y_6 \equiv 0 \bmod(1) \end{aligned}$$

Since $\frac{2}{3} > 0$, the inequality

$$\frac{2}{3} - \frac{7}{9}y_1 - \frac{2}{9}y_4 - \frac{1}{9}y_6 \leq 0$$

yields a new cut to LP.

5. Code:

```
data_DNA1 = fopen('DNA_data1.txt','r');
s = fscanf(data_DNA1,'%c');
data_DNA2 = fopen('DNA_data2.txt','r');
t = fscanf(data_DNA2,'%c');
n = length(s);
m = length(t);

Matrix = zeros(n + 1, m + 1);

v = zeros(1,3);

a = n + 1;
b = m + 1;

for i = 2:b
    Matrix(1,i) = Matrix(1,i-1) + 2;
end

for j = 2:a
```

```
    Matrix(j,1) = Matrix(j-1,1) + 2;
end


for k = 2:a
    for f = 2:b
       if s(k-1) == t(f-1)
          v(1,1) = Matrix(k-1,f-1);
       else
          v(1,1) = Matrix(k-1,f-1) + 1;
       end

            v(1,2) = Matrix(k-1,f) + 2;
            v(1,3) = Matrix(k,f-1) + 2;
        Matrix(k,f) = min(v);

    end
end

Matrix(a,b)

fclose(data_DNA);
```

5.e The cost of the DNA matching is 223, divided by the length of the sequence is 2.23% which is less than 5%. Therefore, the two DNA sequences in the datafile have the same biological function.