CS 220: Synthesis of Digital Systems
University of California, Riverside

Assignment #2

Integer Linear Programming

1. Project Overview

In this project, you will use an ILP to solve a variant of the Graph Coloring Problem that modern compilers use to remove redundant move instructions. The input is a graph $G = (V, E, A)$ with two disjoint sets of edges (E is called the set of *interference edges*, and A is called the set of *affinity edges*; $E \cap A = \emptyset$), along with a positive integer K.

When cast as a Decision Problem, Graph Coloring asks whether or not there exists a function $f: V \to \{1, 2, \ldots, K\}$ which ensures that $\forall e = (u, v) \in E, f(u) \neq f(v)$. In the context of coloring, an affinity edge $a = (x, y) \in A$ is *satisfied* if $f(x) = f(y)$. The optimization problem that we aim to solve is to compute a K-coloring of G that *maximizes* the number of satisfied affinity edges.

Given a color assignment f, let $S_f \subseteq A$ be the subset of affinity edges that have been satisfied.

The problem will be solved in two parts, both of which should be formulated as an ILP:

1. Compute the chromatic number $\chi(G)$ of G. (This ignores affinity edges)
2. Compute the $\chi(G)$-coloring of G that maximizes $|S_f|$ the number of satisfied affinity edges.

A solution to this problem is a program whose input is G (specified as a text file) and whose output is a list of positive integers that comprises the chromatic number $\chi(G)$, the number of satisfied affinity edges $|S_f|$, and the assignment of colors to the vertices of G. The program should invoke the Gurobi ILP solver twice, to solve the two problems stated above. Your program should be written in one of the following four languages: Python, C, C++, or Java.

2. Input File Format

The input file is a text file that specifies the graph using the following format:

- The first line will contain three positive integers: the number of vertices, the number of interference edges, and the number of affinity edges
- $|E|$ lines, each of which specifies an interference edge (two integer vertex IDs)
- $|A|$ lines, each of which specifies an affinity edge (two integer vertex IDs)

Vertex IDs will be $1, 2, \ldots, |V|$. You can assume that the input file will enforce the property that $E \cap A = \emptyset$. Additionally, you can assume that the input file contains no errors, such as a vertex ID (for an edge) larger than $|V|$.

Let $E = \{(u_1, v_1), (u_2, v_2), \ldots, (u_{|E|}, v_{|E|})\}$.
Let $A = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{|A|}, y_{|A|})\}$

The input file for a specific graph would have the following syntax. Please note that there are no commas, parentheses, etc. The file format consists of (positive) integers, spaces, carriage returns, and the End-of-File character at the end (not shown).

$|V|\ \ |E|\ \ |A|$
$u_1\ \ v_1$
$u_2\ \ v_2$
…
$u_{|E|}\ \ v_{|E|}$
$x_1\ \ y_1$
$x_2\ \ y_2$
…
$x_{|A|}\ \ y_{|A|}$

In other words, the input specification file consists of exactly $|E| + |A| + 1$ lines:

- Line 1 contains 3 positive integers
- The remaining $|E| + |A|$ lines contain 2 positive integers each

3. Output File Format

The output file is a text file that specifies the solution quality and color assignment to the graph using the following format:

- The first line contains one positive integer: $\chi(G)$, the chromatic number of G.
- The second line contains one positive integer: $|S_f|$, the number of satisfied affinity edges.
- |V| lines, each of which contains one positive integer: $f(v_i)$, the color assigned to vertex v_i.

The syntax of the output file format is as follows:

$\chi(G)$
$|S_f|$
$f(v_1)$
$f(v_2)$
…
$f(v_{|V|})$

4. Submission Instructions

Submit a .tgz file to iLearn which contains the directory (and any subdirectories) along with your source code files. If there are any difficulties evaluating your assignment, the instructor will contact you to request a short in-person demo.

5. Details/Suggestions

You may use any ILP formulation of the Graph Coloring problem that you wish. It is perfectly OK to formulate both Problems 1 and 2 using appropriate variants of the Assignment ILP formulation. It is perfectly acceptable to adapt the more advanced Representatives or Partial Order (POP/POP2) formulations if you so desire, but they will be more difficult to design and debug.

You do not need to test your solver on exceptionally large graphs; the test cases used for evaluation will be small.

Several sample input files will be posted on iLearn to help guide you.

Remember, there may be multiple $\chi(G)$-colorings that maximize $|S_f|$.

While it is technically possible to design a single ILP that solves Problems 1 and 2 together, it will be very difficult to do so and it will have a very large search space. Decomposing this project into two separate problems will be much simpler and will not sacrifice optimality.

If you choose to use the representatives ILP formulation (even if just for Problem #1), I would be interested to know if modifying the following constraint has any impact on solution accuracy

$\sum_{u \in adj^c[v] \cup \{v\}} x_{uv} \geq 1$ (original constraint)

$\sum_{u \in adj^c[v] \cup \{v\}} x_{uv} = 1$ (proposed change)