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

Assignment #3

NSGA-II

1. Project Overview

In this project, you will use an ILP to solve the same variant of the Graph Coloring Problem that we studied in Assignment #2. This time, we will treat this graph coloring variant as a multi-objective optimization problem and solve it using NSGA-II, which is a multi-objective genetic algorithm.

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. We seek to color the graph in a manner that optimally satisfies two objectives:

1. Minimize the number of colors used to color G.
2. Maximize the number of satisfied affinity edges.

Recall that a legal coloring satisfies the property that $\forall e = (u, v) \in E, f(u) \neq f(v)$. Also recall that an affinity edge $a = (x, y) \in A$ is *satisfied* if $f(x) = f(y)$.

Please note that it is not always possible to satisfy both constraints. For example, it may not be possible to satisfy the maximal number of affinity edges using the minimum number of colors.

2. NSGA-II

NSGA-II has been written and released many times by many authors. The original, which was written in C, can be found at the following URL:

https://www.iitk.ac.in/kangal/codes.shtml

Other implementations can be found via Google search. For example, here is a Python implementation:

https://pythonhealthcare.org/2019/01/17/117-code-only-genetic-algorithms-2-a-multiple-objective-genetic-algorithm-nsga-ii/

Feel free to use any implementation that you wish; however, the instructor only has experience with the original release listed above.

3. 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

4. Output File Format

The output file should contain all of the Pareto-optimal solutions that are found by NSGA-II. (Please note that this does not mean to output all of the solutions in the population). The output file is a text file that specifies the solutions and their color assignments using the following format:

- The first line contains one positive integer: N, the number of solutions
- The next N lines contains a pair of vertices, $k_i\ a_i$, where k_i is the number of colors used in the i^{th} solution and a_i is the number of satisfied affinity edges.
- The next $N|V|$ lines list the color assignments for each of the solutions. Let $f_i(v_j)$ denote the color assigned to vertex v_j in the i^{th} solution.

The syntax of the output file format is as follows:

N
$k_1\ a_1$
$k_2\ a_2$
…
$k_N\ a_N$
$f_1(v_1)$
$f_1(v_2)$
…
$f_1(v_{|V|})$
$f_2(v_1)$
$f_2(v_2)$
…
$f_2(v_{|V|})$
…
$f_N(v_1)$
$f_N(v_2)$
…
$f_N(v_{|V|})$

5. 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.