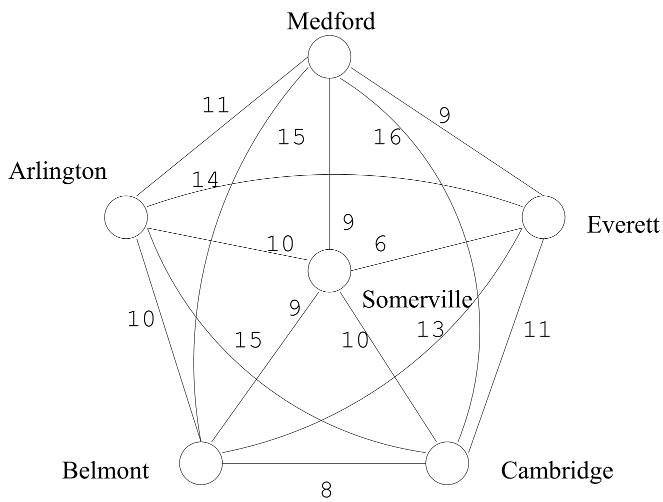
# CS580 Fall 2024: Homework Set #2

Q1. (8p) The Traveling Salesman Problem (TSP) is a challenge to the salesman who wants to visit every location exactly once and return home, as quickly as possible. Each location can be reached from every other location, and for each pair of locations, there is metric that defines the time between them. Given the following graph




your task is to use gradient descent to improve on the original loop, i.e. find a loop with a lower cost.

Note that at each step the method should identify the pair of edges whose endpoints can be switched to improve the loop. The switching is done in the following manner. Given the loop $a_1,a_2,a_3,a_4,a_5\ldots,a_n,a_1$ the edges which do not share a node can be switched, for example switching the edges $(a_1,a_2)$ and $(a_4,a_5)$ we obtain a new loop $a_1,a_4,a_3,a_2,a_5,\ldots a_n,a_1$.

Start with the loop *(Medford Belmont Everett Arlington Cambridge Somerville Medford)* [or *(M B E A C S M)* for short]. You should stop when the loop cannot be improved anymore.

# Homework-2

① Travelling salesman problem

→ Initial Loop :-

$M \rightarrow B \rightarrow E \rightarrow A \rightarrow C \rightarrow S \rightarrow M$

$M \rightarrow B$ : 15

$B \rightarrow E$ : 13

$E \rightarrow A$ : 14

$A \rightarrow C$ : 15

$C \rightarrow S$ : 10

$S \rightarrow M$ : 9

Initial cost $\Rightarrow 15+13+14+15+10+9 \Rightarrow 76$

→ Swapping $(B \rightleftharpoons E)$ and $(A \rightleftharpoons C)$

After swapping, new loop would be :-

$M \rightarrow A \rightarrow E \rightarrow B \rightarrow C \rightarrow S \rightarrow M$

$M \rightarrow A$ : 11

$A \rightarrow E$ : 14

$E \rightarrow B$ : 13

$B \rightarrow C$ : 8

$C \rightarrow S$ : 10

$S \rightarrow M$ : 9

New cost $\Rightarrow 11+14+13+8+10+9 \Rightarrow 65$

∴ New cost is improved, compared to Initial cost

→ Swapping (A→E) and (E→B)

After swapping, new loop would be :-

$M \rightarrow A \rightarrow B \rightarrow E \rightarrow C \rightarrow S \rightarrow M$

New cost $\Rightarrow 11 + 10 + 13 + 11 + 10 + 9 \Rightarrow 64$

Hence, New cost is improved when compared to previous cost i.e 65

→ Swapping (B→E) and (C→S)

After swapping, new loop would be :-

$M \rightarrow A \rightarrow B \rightarrow S \rightarrow C \rightarrow E \rightarrow M$

New cost $\Rightarrow 11 + 10 + 9 + 10 + 11 + 9$

$\Rightarrow 60$

Hence, New cost is improved when compared to previous cost i.e 64

→ Swapping (B→S) and (C→E)

After swapping, new loop would be :-

$M \rightarrow A \rightarrow B \rightarrow C \rightarrow S \rightarrow E \rightarrow M$

New cost $\Rightarrow 11 + 10 + 8 + 10 + 6 + 9$

$\Rightarrow 54$

Hence, New cost is improved when compared to previous cost i.e 60

Therefore, we managed to reduce the initial cost of 76 to 54 by swapping the appropriate pairs of edges.

Q2. (8p) A map of a part of central Europe is shown in the figure.


Use backtracking search with *most-constrained-variable* heuristics and *Arc Consistency Checking* to color this map using four colors *(R,G,B,Y)*. Is it possible to color the map with three colors? Can you prove that it is impossible? Hint: Color Austria first.

**Ans** : Here, the goal is to see if we can color the countries on the map using just three colors (let's say Red, Green, and Blue) in such a way that no neighbouring countries share the same color .

**Step 1: Representing the Map as a Graph**

First, we think of this map as a network, where each country is like a node (or point), and an edge (or line) connects two nodes if those countries share a border.

For example:

Austria borders Germany, Czech Republic, Slovakia, Hungary, Slovenia, Switzerland, and Italy.
Germany borders Poland, Czech Republic, Austria, Switzerland, and Italy.

**Step 2: Choosing a Strategy (Most-Constrained Variable Heuristic)**

To make the coloring process more efficient, we'll use a strategy called the Most-Constrained Variable Heuristic. This strategy suggests starting with the country that has the most neighbors (or connections)

In this case, Austria has the most neighbors (seven in total), making it the most constrained. So, we'll start by coloring Austria first.

**Step 3: Color the Map Step by Step**

Let's start with Austria and assign Red to it. Then we color the neighboring countries one by one, ensuring that no two connected countries have the same color:

- Austria is colored Red.
- Germany, a neighbor of Austria, needs a different color, so we color it Green.
- Czech Republic borders both Austria (Red) and Germany (Green), so we color it Blue.
- Poland only borders the Czech Republic (Blue), so we can safely color it Red.
- Switzerland borders Germany (Green) and Austria (Red), so it gets Blue.
- Italy borders Austria (Red) and Switzerland (Blue), so we color it Green.
- Slovakia borders Austria (Red) and the Czech Republic (Blue), so it gets Green.
- Hungary borders Austria (Red), Slovakia (Green), and Slovenia (which we haven't colored yet). We color Hungary Blue.

**The Issue with Slovenia**

Slovenia is surrounded by Austria (Red), Hungary (Blue), and Italy (Green). Unfortunately, all three colors are already in use by its neighbors, leaving no available color for Slovenia without causing a conflict!

So, after trying different combinations, we've proven that it's impossible to color this map with only three colors without causing a conflict. We would need a fourth color to make it work.