COMP1201

Assignment 3

Alberto Tamajo

Student ID: 30696844

May 2020

Electronics and Computer Science Department
University of Southampton

1 Questions

Q1 **CLAIM:** in any Facebook community, there must exist two people who have the same number of friends (assuming there cannot be dual-links in a Facebook community).

PROOF:

It is possible to model a Facebook community by using a simple and connected graph where the community members are the vertices of the graph and the friendship relations are the edges of the graph. The graph that models the Facebook community is simple because there cannot be loops as no member can be friend with himself/herself and there cannot be multiple edges between 2 vertices because one edge is enough to represent the friendship relation between two community members. The graph is connected because of the simple fact that each community member must be friend with at least another community member. Therefore, given that x and y are two vertices of the graph that models a Facebook community, there must exist a path from x to y.

Let $G(V, E)$ be a simple and connected graph that models a Facebook community.
Let $|V|$ be the number of nodes in G.
Let d(v) indicate the degree of vertex v such that v $\in$ G.
It follows that $0 < d(v) \leq (|V| - 1)$ for every vertex v $\in$ G.
Let there be a function $f : V \to D$ where $D = \{1, 2, ..., |V| - 1\}$ which maps every vertex of G to its vertex degree. By the pigeonhole principle we know that if there exist a function $g : X \to Y$ where $|X| > |Y|$ then at least 2 elements of X are mapped to the same element of Y.
Therefore, since in the function $f : V \to D$, $|V| > |D|$ because $|D| = |V| - 1$, then at least 2 nodes of G have the same vertex degree. In other words, at least 2 members of the community must have the same number of friends.

A MORE GENERAL PROOF
Let G(V,E) be a simple graph.
Let S be the set containing the COMPONENTS of G such that $S = \{C_1, C_2, ..., C_n\}$.
Let d(v) indicate the degree of a vertex v.
It follows that:
$((\forall v \in C \text{ s.t } C \in S,\, 0 < d(v) \leq (|C| - 1)) \Rightarrow \forall\, C \in S,\, d : C \to D\, where\, D = \{1, 2, ..., |C| - 1\})$
which implies that $\forall\, C \in S$, there exist at least 2 vertices v_1 and v_2 such that $d(v_1) = d(v_2)$ by the Pigeonhole Principle.

Q2 (a) **CLAIM:**
Let T be a MST of a graph G. If we remove an edge {u,v} of T we obtain two trees T* and T**. Both trees, T* and T**, are MSTs on their respective sets of nodes and {u,v} is a least-weight crossing edge between T* and T**.

PROOF T* and T are MSTs on their respective sets of nodes:**

The proof will prove that T** must be a MST on its respective set of nodes and for the same reason also T*.
There are 2 possible cases that can occur:

- **CASE 1:** T** contains just one node and so either u or v was a leaf. In this case T** is trivially a MST on its set of nodes.
- **CASE 2:** T** contains more than one node.

PROOF BY CONTRADICTION CASE 2:

Let w be a function that takes as input a set of weighted edges and outputs their weight sum.
Let there be a a tree P on the set of vertices of T** such that $w(P) < w(T^{**})$.
Therefore $w(P \cup \{u, v\} \cup T^*) < w(T)$ where $T = T^* \cup \{u, v\} \cup T^{**}$.
However, T is a MST of G and so there cannot be a tree P such that $w(P) < w(T^{**})$.
In conclusion, T** must be a MST on its set of vertices and for the same reason T* must be a MST on its set of vertices as well.

PROOF {u,v} is a least-weight crossing edge between T* and T:**

Let {z,q} be a crossing edge between T* and T** such that $w(\{ \{z, q\} \}) < w(\{ \{u, v\} \})$.
It follows that $w(T^* \cup \{z, q\} \cup T^{**}) < w(T)$ where $T = (T^* \cup \{u, v\} \cup T^{**})$.
However, T is a MST of G and so there cannot be a crossing edge {z,q} between T* and T** such that $w(\{ \{z, q\} \}) < w(\{ \{u, v\} \})$.

(b) **CLAIM:**
It is not guaranteed that splitting arbitrarily the set of nodes V of a graph G into two nearly equal-sided sets R and S, finding a MST on R and S and connecting the MST of R and the MST of S with the least-cost edge between them gives a MST of G. Therefore the algorithm proposed in the assignment is **NOT CORRECT.**

PROOF:

Let G(V,E) be a complete graph such that V={a,b,c}.
Let T(V,Q) be a MST of G such that $Q = \{ \{a, b\}, \{b, c\} \}$.
Let w be a function that takes as input a set of weighted edges and outputs their weight sum.
Let $w(\{ \{a, c\} \}) > w(\{ \{a, b\} \})$.
Let V be split into two nearly equal-sided sets R and S such that R={a,c} and S={b}.
Consequently, the MST of R must include the edge {a,c}.
From the MST T we know that the least weight edge between R and S is the edge {b,c}.
Therefore, we get a spanning tree T*(V,Z) such that Z={ $\{a, c\}, \{b, c\}$ }.
However, T* is not a MST of G because $w(T^*) > w(T)$ which follows from the fact that $w(\{ \{a, c\} \}) > w(\{ \{a, b\} \})$.

Q3 (a) In order to find the optimal allocation of manufacturing capacity between the factories I have transformed the given problem into the following linear program:

Let x be the number of vehicles produced by Factory 1.
Let y be the number of vehicles produced by Factory 2.
Let z be the number of vehicles produced by Factory 3.

- **OBJECTIVE FUNCTION:** $x + y + z$
- **CONSTRAINTS:**
  - $18x + 14y + 11z \leq 4000$
  - $6x + 5y + 7z \leq 4000$
  - $x \geq 100$
  - $10x + 17y + 20z \leq 3000$
  - $y, z \geq 0$

However, this linear program above is not in the Standard Form. With Standard Form I mean a linear program that has the following characteristics:

- Maximisation of an OBJECTIVE FUNCTION
- All constraints are inequalities of the form $x_1 + x_2 + ... + x_n \leq b$
- Non-negative variables

The Standard form of the Linear program above is the following:

- **OBJECTIVE FUNCTION:** $x + y + z$
- **CONSTRAINTS:**
  - $18x + 14y + 11z \leq 4000$
  - $6x + 5y + 7z \leq 4000$
  - $-x \leq -100$
  - $10x + 17y + 20z \leq 3000$
  - $y, z \geq 0$

Additionally, I also write down the Slack-Form of the Linear Program above.
Let p,q,r,s be 4 slack variables. It follows that the Slack-Form of the linear program above is:

- **OBJECTIVE FUNCTION:** $x + y + z$
- **CONSTRAINTS:**
  - $p = 4000 - 18x - 14y - 11z$
  - $q = 4000 - 6x - 5y - 7z$
  - $r = -100 + x$
  - $s = 3000 - 10x - 17y - 20z$

- $x, y, z, p, q, r, s \geq 0$

By using the **intlinprog** (since the problem is an INTEGER LINEAR PROGRAMMING PROBLEM) function of MatLab, I have obtained the following results:

- $x = 188$
- $y = 0$
- $z = 56$

(b) In order to find the optimal allocation of manufacturing capacity between the factories I have transformed the given problem into the following linear program:

Let x be the number of vehicles produced by Factory 1.
Let y be the number of vehicles produced by Factory 2.
Let z be the number of vehicles produced by Factory 3.

- **OBJECTIVE FUNCTION:** $x + y + z$
- **CONSTRAINTS:**

  - $18x + 14y + 11z \leq 4000$
  - $6x + 5y + 7z \leq 4000$
  - $10x + 17y + 20z \leq 3000$
  - $x \geq 75$
  - $y \geq 75$
  - $z \geq 75$

As regards the linear program above, it has no feasible solution as it is evident that the company would violate the environmental regulations if it had to produce 75 vehicles in each factory.