CS345 : Assignment 3

Dhruv Gupta (220361)

October 2024

Dhruv Gupta: 220361

Question 1

Reduction to Max-Flow Problem:

We will reduce this problem to Max-Flow Problem by creating an appropriate directed graph $G = (V, E)$. Let B be the set of all buses, S be the set of all stations, s and t be special vertices not in B and S (these will be the source and sink vertices), $V = B \cup S \cup \{s, t\}$ be the set of vertices, E be the set of edges. For each pair $(u, v) \in B \times S$, if distance between u and $v \leq r$, we will add the edge (u, v) in E. Also we will add the edges (s, u) and (v, t) $\forall u \in B$ and $\forall v \in S$ in E. Note that there is no edge between any two vertices in B or between any two vertices in S.
We will assign capacities $c(u, v)$ to all the edges $(u, v) \in E$.

- $c(s, u) = 1 \ \forall u \in B$
- $c(v, t) = L \ \forall v \in S$
- $c(u, v) = 1$ otherwise

We will apply Max-Flow algorithm for this graph considering s and t as the source and sink respectively. We claim that the max flow will be equal to the maximum number of buses that can be connected simultaneously to a station while following the given constraints. If this number is less than n then it is not possible to connect every bus simultaneously. Otherwise if it is equal to n, it is possible to connect every bus simultaneously. It is easy to observe that max flow can not be more than n.

Proof of correctness:

We will define a *mapping* M as a set of edges of the form $(u, v) \in E$ where $u \in B, v \in S$ such that $\forall u \in B$, there is atmost one outgoing edge in M, and $\forall v \in S$ there are atmost L incoming edges in M. Note that for every *mapping* we will get a valid way of connecting the buses to the stations. Let us define the "size" of a *mapping* M as the number of elements in the set M, i.e., $|M|$. It is easy to see that every *mapping* corresponds to a valid way of connecting buses to stations while following all the constraints. Therefore, if we get a *mapping* of size k, it means that k buses can be connected.

Theorem: **There exists a flow of value k in G iff there exists a *mapping* of size k in G.**

Proof:
Part I: flow of value $k \Rightarrow$ mapping of size k.
Consider any flow with value k in G. By integrality of max flow, we can say that there exists an integral flow f of value k in G. We will create a *mapping* M of size k from this flow as follows -

$$M = \{(u, v) \mid u \in B,\ v \in S,\ (u, v) \in E, f(u, v) = 1\}$$

Note that flow can only be 0 or 1 for each edge from B to S (because flow is integral) and we are taking all those edges along which flow is 1. We will prove that M is a valid *mapping* of size k.

- First, we will prove that $|M| = k$.
  Consider the $s - t$ cut with $A = B \cup \{s\}$ and $\bar{A} = S \cup \{t\}$ as the two sets. Note that $k = value(f) = f_{out}(A) - f_{in}(A)$. Note that there are no edges from $\bar{A}$ to A. Therefore, $f_{in}(A) = 0$. So, $k = value(f) = f_{out}(A)$. Also, all the edges going from A to $\bar{A}$ are of the form (u, v) where $u \in B, v \in S$. Since, flow values along these edges can only be 0 or 1, $f_{out}(A) =$ total number of edges going from A to $\bar{A}$ which have a flow of $1 =$ total number of edges going from B to S which have a flow of $1 = |M|$. Therefore, $|M| = k$.

- Second, we will prove that there is atmost one outgoing edge in M for each vertex in B.
  Consider any $u \in B$. Since, f is a flow, by conservation constraint $f_{out}(u) - f_{in}(u) = 0 \Rightarrow f_{out}(u) = f_{in}(u)$ (since u is an intermediate vertex). Now, only incoming edge on u is from s. So, $f_{in}(u) = f(s, u) \le c(s, u) = 1$, (by capacity constraint) $\Rightarrow f_{in}(u) \le 1 \Rightarrow f_{out}(u) \le 1$. Since, flow along any edge from B to S can only be 0 or 1, $f_{out}(u)$ is the number of outgoing edges from u with flow of 1, there can be atmost one outgoing edge from u with a flow of 1. Therefore, atmost one outgoing edge from u will be included in M.

- Third, we will prove that there are atmost L incoming edges in M for each vertex in S.
  Consider any $v \in S$. Since, f is a flow, by conservation constraint $f_{out}(v) - f_{in}(v) = 0 \Rightarrow f_{out}(v) = f_{in}(v)$ (since v is an intermediate vertex). Now, only outgoing edge from v is to t. So, $f_{out}(v) = f(v, t) \le c(v, t) = L$, (by capacity constraint) $\Rightarrow f_{out}(v) \le L \Rightarrow f_{in}(v) \le L$. Since, flow along any edge from B to S can only be 0 or 1, $f_{in}(v)$ is the number of incoming edges on v with flow of 1, there can be atmost L incoming edges on v with a flow of 1. Therefore, atmost L incoming edges on v will be included in M.

Hence we have proven that M is a valid *mapping* of size k. Therefore, we have proven that for every flow of value k there exists a *mapping* of size k in G.

Part II: mapping of size $k \Rightarrow$ flow of value k.
Consider a *mapping* M in G, with $|M| = k$. We will construct a flow f, in G such that $value(f) = k$.

- For all edges $(s, u) \in E,\ u \in B :\ f(s, u) = 1$ if there exists an edge of the form (u, v) in M, 0 otherwise.

- For all edges $(u, v) \in E,\ u \in B,\ v \in S :\ f(u, v) = 1$ if $(u, v) \in M$, 0 otherwise.

- For all edges $(v, t) \in E,\ v \in S :\ f(v, t) = number\ of\ edges\ of\ the\ form\ (u, v)\ in\ M.$

Now we need to prove that this is a valid flow of value k.

- First, we will prove that $value(f) = k$.
  Consider the $s - t$ cut with $A = B \cup \{s\}$ and $\bar{A} = S \cup \{t\}$ as the two sets. Note that $k = value(f) = f_{out}(A) - f_{in}(A)$. Note that there are no edges from $\bar{A}$ to A. Therefore, $f_{in}(A) = 0$. So, $value(f) = f_{out}(A)$. Also, all the edges going from A to $\bar{A}$ are of the form (u, v) where $u \in B, v \in S$. Since, flow values along these edges can only be 0 or 1, $f_{out}(A) =$ total number of edges going from A to $\bar{A}$ which have a flow of $1 =$ total number of edges going from B to S which have a flow of $1 = |M| = k$ (since, for every edge $(u, v) \in B \times S$, flow is 1 iff $(u, v) \in M$). Therefore, $value(f) = k$..

- Second, we will prove the capacity constraint, i.e., for every edge $(u, v) \in E, f(u, v) \leq c(u, v)$.

  - For all edges $(s, u) \in E,\ u \in B:\ f(s, u) = 1$ or $0 \leq 1 = c(s, u)$.
  - For all edges $(u, v) \in E,\ u \in B,\ v \in S:\ f(u, v) = 1$ or $0 \leq 1 = c(u, v)$.
  - For all edges $(v, t) \in E,\ v \in S:\ f(v, t) = number\ of\ edges\ of\ the\ form\ (u, v)\ in\ M =$ number of incoming edges on v in $M \leq L = c(v, t)$.

- Third, we will prove the conservation constraint, i.e., for every vertex $w \in V \backslash \{s, t\},\ f_{out}(w) - f_{in}(w) = 0$. Take any vertex $w \in V \backslash \{s, t\} = B \cup S$.

  - If $w \in B$ and there exists an edge of the form (w, v) in M, by construction, $f(s, w) = 1$. Also, since M is a *mapping*, there cannot be more than one edge of the form (w, v) in M. Let this edge be (w, v_0). Again by construction, $f(w, v_0) = 1$ and $f(w, v) = 0$, where $(w, v) \in E,\ \forall v \in S \backslash \{v_0\}$.
    Therefore, $f_{out}(w) - f_{in}(w) = \sum_{v \in S} f(w, v) - f(s, w) =$
    $f(w, v_0) + \sum_{v \in S \backslash \{v_0\}} f(w, v) - f(s, w) = 1 + 0 - 1 = 0$.
  - If $w \in B$ and there does not exist an edge of the form (w, v) in M, by construction, $f(s, w) = 0$. Also, by construction $f(w, v) = 0$ for all $v \in S$. Therefore, $f_{out}(w) - f_{in}(w) = \sum_{v \in S} f(w, v) - f(s, w) = 0 - 0 = 0$.
  - If $w \in S$, $f_{out}(w) = f(w, t) = number\ of\ edges\ of\ the\ form\ (u, w)\ in\ M =$ number of edges $(u, w),\ u \in B$ with $f(u, w) = 1$ (by construction all the edges of this form have a flow of 1 if they are in M, otherwise they have a flow of 0) $= \sum_{u \in B} f(u, w) = f_{in}(w)$.
    Therefore, $f_{out}(w) = f_{in}(w) \Rightarrow f_{out}(w) - f_{in}(w) = 0$.

Hence we have proven that f is a valid flow with value k. Therefore, we have proven that for every *mapping* of size k, there exists a flow of value k in G.
This completes the proof of the theorem.
Now, suppose max flow in G has a value p, i.e., we have a flow with value p, and there does not exist any flow with value more than p. From above theorem, we can say that we will have a *mapping* of size p, and there does not exist any *mapping* with size more that p. Thus size of maximal *mapping* must be of size p, i.e., the maximum number of buses that can be connected simultaneously to a station while following the given constraints must be p. This proves the correctness of our algorithm.

Pseudocode:

We will assume that we are given B (set of buses), S (set of stations), r (range parameter) and L (capacity parameter). Also, $u.x$ and $u.y$ are the x and y coordinates for the vertex u (bus or a station). We will also assume that a function **Max-Flow(G)** is available which returns the max flow value for a given input graph G. Also, we have a function $dist(u, v)$ which returns the distance between u and v in $O(1)$ time.

IsConnectionPossible(B, S, r, L**)**{
  $V \leftarrow B \cup S \cup \{s, t\}$;
  $E \leftarrow \phi$;
  for every $u \in B$ { $E \leftarrow E \cup \{(s, u)\}$; $c(s, u) = 1$}
  for every $v \in S$ { $E \leftarrow E \cup \{(v, t)\}$; $c(v, t) = L$}
  for every $(u, v) \in B \times S$ {
    if($dist(u, v) \leq r$**)** {$E \leftarrow E \cup \{(u, v)\}$; $c(u, v) = 1$}
  }
  Let $G = (V, E)$
  $x \leftarrow$ **Max-Flow(**G**)**;
  if($x < |B|$**) return** $false$;
  else return $true$;
}

Time Complexity Analysis:

Let $|B| = n, |S| = k$. We assume that insertion operation (union with a single element) in a set takes $O(log(N))$ time, where N is the size of the set (this is true for RBT implementation of set). Creating V takes $O((n + k)log(n + k))$ time. Creating E and assigning capacities to all the edges simultaneously takes $O((n + k + nk)log(n + k) = O(nklog(n + k))$ time. Note that $|V| = n + k + 2 = O(n + k)$, $|E| \leq nk + n + k = O(nk)$. **Max-Flow** will take $O(E^2V) = O(n^2k^2(n + k))$ time (using Edmond Karp algorithm). Therefore, overall time complexity is $O(n^2k^2(n + k) + nklog(n + k) + (n + k)log(n + k)) = O(n^2k^2(n + k))$, which is a polynomial in n and k.

Dhruv Gupta: 220361

Question 2

Reduction to Edge-Disjoint-Paths Problem:

We will reduce the given problem to the edge-disjoint-paths problem, by modifying the given graph G to create a new graph G'. Let $G = (V, E)$. We will create $G' = (V', E')$. For every vertex $v \in V$, add vertices v_{in} and v_{out} in V' and edge (v_{in}, v_{out}) in E'. Also, for every edge $(x, y) \in E$, add edge (x_{out}, y_{in}) in E'. We claim that maximum number of vertex disjoint paths in G is equal to the maximum number of edge disjoint paths in G', which we know how to compute from the edge-disjoint-paths problem.

Proof of correctness:

We will first prove the following theorems.

Theorem 1: **For every $s-t$ path in G there is a unique $s_{out} - t_{in}$ path in G' and vice-versa.**

Proof: Consider any $s-t$ path $P_1 = (s = v^0, v^1, ..., v^{k-1}, v^k = t)$ in G. Construct a path $P_1' = (s_{out} = v^0_{out}, v^1_{in}, v^1_{out}, ..., v^{k-1}_{in}, v^{k-1}_{out}, v^k_{in} = t_{in})$. Note that this is a valid path in G'.
Now, observe that all the edges in G' are of the form (v_{in}, v_{out}) or of the form (x_{out}, y_{in}). Therefore, any $s_{out} - t_{in}$ path P_2' must be of the form $(s_{out} = v^0_{out}, v^1_{in}, v^1_{out}, ..., v^{k-1}_{in}, v^{k-1}_{out}, v^k_{in} = t_{in})$. To construct an $s-t$ path P_2 from P_2', we can remove every consecutive pair v^i_{in}, v^i_{out} and replace it by the vertex v^i. Also, replace s_{out}, t_{in} by s, t respectively. This will give us an $s-t$ path $P_2 = (s = v^0, v^1, ..., v^{k-1}, v^k = t)$ in G.
Thus we have proven that for every $s-t$ path in G we have an $s_{out} - t_{in}$ path in G' and vice-versa. Now we need to prove that this path is unique.
Consider any two distinct $s-t$ paths P_1 and P_2 in G. If they have different number of vertices, by construction the number of vertices in the corresponding $s_{out} - t_{in}$ paths P_1' and P_2' in G' will also be different (note that the number of intermediate vertices becomes double). Hence, P_1' and P_2' must be distinct. Otherwise, they must be of the form $P_1 = (s = v^0, v^1, ..., v^{k-1}, v^k = t)$ and $P_2 = (s = u^0, u^1, ..., u^{k-1}, u^k = t)$. For these paths to be distinct, there must exist some i, such that $v^i \neq u^i$. This implies that $v^i_{in} \neq u^i_{in}$ and $v^i_{out} \neq u^i_{out}$ for that i. Thus P_1' and P_2' must be distinct.
Similarly, take any two distinct $s_{out} - t_{in}$ paths P_1' and P_2'. If they have different number of vertices, by construction the number of vertices in the corresponding $s-t$ paths P_1 and P_2 in G will also be different (note that the number of intermediate vertices becomes half). Hence, P_1 and P_2 must be distinct. Otherwise, they must be of the form
$P_1' = (s_{out} = v^0_{out}, v^1_{in}, v^1_{out}, ..., v^{k-1}_{in}, v^{k-1}_{out}, v^k_{in} = t_{in})$ and
$P_2' = (s_{out} = u^0_{out}, u^1_{in}, u^1_{out}, ..., u^{k-1}_{in}, u^{k-1}_{out}, u^k_{in} = t_{in})$. For these paths to be distinct, there must exist some i, such that $v^i_{in} \neq u^i_{in}$ or $v^i_{out} \neq u^i_{out}$. But since, there is only one outgoing edge from v^i_{in} which goes to v^i_{out}, and there is only one incoming edge to v^i_{out} which comes from v^i_{in}, $v^i_{in} \neq u^i_{in} \Leftrightarrow v^i_{out} \neq u^i_{out}$. This implies that $v^i \neq u^i$ for that i. Therefore, P_1 and P_2 must be distinct.
Thus we have proven a one-to-one correspondence between the $s-t$ paths in G and $s_{out} - t_{in}$ paths in G', i.e., for every $s-t$ path in G there is a unique $s_{out} - t_{in}$ path in G' and vice-versa.

Theorem 2: **Two $s - t$ paths are vertex disjoint in G if the corresponding $s_{out} - t_{in}$ paths are edge disjoint in G' and vice-versa.**

Proof: Consider any two vertex disjoint $s - t$ paths P_1 and P_2 in G, i.e., if we take any pair of vertices $u, v(\neq s \text{ or } t)$ such that u lies in P_1 and v lies in P_2, then $u \neq v \Rightarrow u_{in} \neq v_{in}$ and $u_{out} \neq v_{out}$ in the corresponding $s_{out} - t_{in}$ paths P_1', P_2' in G'. Also, trivially any "in-vertex" can not be equal to any "out-vertex" in G' (by construction of G'). Hence, the paths P_1' and P_2' will also be vertex disjoint. Since for two paths to have a common edge, they must necessarily have a common vertex, we can conclude that P_1' and P_2' must be edge disjoint.
Now, consider two edge disjoint $s_{out} - t_{in}$ paths P_1' and P_2' in G'. We need to prove that the corresponding $s - t$ paths P_1 and P_2 in G must be vertex disjoint. We will prove this by contradiction. Suppose, P_1 and P_2 both have a common vertex v. Then, P_1' and P_2' must have a common edge (v_{in}, v_{out}). This is a contradiction.
This completes the proof of our theorem.

Now, suppose that the maximum number of edge disjoint paths in G' is k and let these paths be $P_1', ..., P_k'$. Also, since this is the maximum number we can get, it is not possible to get more than k edge disjoint paths in G'. By theorem 1, we can construct k paths, $P_1, ..., P_k$ in G. By theorem 2, all these paths will be vertex disjoint. Now, if it is possible to get more than k vertex disjoint paths in G, by theorem 1 and 2, we can construct more than k edge disjoint paths in G'. Thus it is not possible. Therefore, k is the maximum number of vertex disjoint paths that we can get in G. This completes our proof of correctness.

Pseudocode:

We will assume that we are given a graph $G = (V, E)$ and we have a function **Max-Edge-Disjoint-Paths(**G, s, t**)** which returns the maximum number of edge disjoint paths in a given input graph G from vertex s to vertex t.

Max-Vertex-Disjoint-Paths($G = (V, E), s, t$**)**{
  $V' \leftarrow \phi$; $E' \leftarrow \phi$;
  for every $v \in V$ { $V' \leftarrow V' \cup \{v_{in}, v_{out}\}$; $E' \leftarrow E' \cup \{(v_{in}, v_{out})\}$; }
  for every $(u, v) \in E$ { $E' \leftarrow E' \cup \{(u_{out}, v_{in})\}$; }
  Let $G' = (V', E')$
  $x \leftarrow$ **Max-Edge-Disjoint-Paths(**G', s_{out}, t_{in}**)**;
  return x;
}

Time Complexity Analysis:

Assuming that each insertion in set takes $O(log(N))$ time, where N is the number of elements in the set (RBT implementation of set), construction of G' will take $O(n(2log(n) + log(m)) + mlog(m) = O((m + n)(log(m) + log(n)))$ time (for each vertex, we have two insertions in V' and one insertion in E'; for each edge, we have one insertion in E'). We know that **Max-Edge-Disjoint-Paths** function, uses max-flow algorithm. Therefore, it should take $O(m^2n)$ time (Edmond Karp algorithm). Thus overall time complexity would be $O(m^2n + (m + n)(log(m) + log(n)) = O(m^2n)$, which is a polynomial in m and n.