W2U3:
All Pairs Shortest Path -1
Part 5

C Pandu Rangan

Recap

- k paths
- Algorithm 4 (Floyd-Warshall Algorithm)

Outcome

- Johnson's Algorithm

Johnson's Algorithm

- We will now look at yet another algorithm that is faster for sparse graph
- Floyd-Warshall's algorithm is $O(n^3)$ and the complexity is independent of the number of edges of the graph.
- The algorithm by Johnson runs in $O(n^2 \log n + nm)$ time and when the graph is sparse, this is asymptotically better than $O(n^3)$ algorithm.
- For dense graph with $m = O(n^2)$, the complexity is $O(n^3)$, which is same as Warshalls Algorithm. This algorithm uses a clever transformation technique to achieve improvements.

Johnson's Algorithm (contd)

If all weights are positive, we may apply Dijkstra's algorithm n times (once from each vertex as the source) and the complexity for this algorithm would be in

$$O(n[n \log n + m]) = O(n^2 \log n + nm)$$

However, this approach is not applicable if G has some negative edges.

If G has negative edges but no negative cycles, we may apply n times the Bellman-Ford algorithm and the complexity would be

$O(n.n.m) = O(n^2 m)$. For Dense graph this may go as high as $O(n^4)$.

Johnson's algorithm deploys a transformation of weights that allowed him to use both Dijkstra's and Bellman-Ford algorithms to exploit the best in both methods.

Weight Transformation

Let $G = (V, E)$ be a directed graph and w be the weight function from edge set to integers.

Let $V = \{1,2,\cdots,n\}$ and h be any function from V to integers.

Define a new weight function w' by

$$w'(u,v) = w(u,v) + h(u) - h(v)$$

1. P is a shortest path from i to j under w if it is a shortest path under w'
2. For any cycle c in $G, w(c) = w'(c)$
3. For any path P from i to j

$$w(P) = w'(P) + h(j) - h(i)$$

Basic Idea

- Thus, instead of working on G with weight function w, we may work on G with weight function w'.
- If $w(e) > 0 \ \forall\, e \in E$, we need not transform the weights. We apply Dijkstra's algorithm n times and obtain as algorithm for APSP with complexity $o(n^2 \log n + nm)$.
- If $w(e)$ is negative for some edges, using Bellman and Ford n times leads to a very inefficient $o(n^2 m)$ algorithm. This is the case that requires a transformation of weights.

Basic Idea (contd)

- The trick is,
  Use Bellman-Ford algorithm ONCE and find a
  $h: V \rightarrow I$ such that $w'(e) > 0 \ \forall \ e \ \in E$.
- Now, Dijkstra's algorithm can be applied n times om G with weight function w' and solve APSP with respect to w'. The same physical paths determined by w' can be used for w, by (1).
- Since $\delta(i,j) = \delta'(i,j) + h(j) - h(i)$ by (3), the APSP weight matrix under w can be constructed from APSP weight matrix under w' in $O(n^2)$ time.

Johnson's Algorithm

The Algorithm at a high level is as follows:

1. Use Bellman-Ford algorithm to determine a $h: V \to I$ such that $w'(e) > 0 \; \forall \; e \; \in E$ where $w'(i,j) = w(i,j) + h(i) - h(j)$
2. For $i, j \;\; \forall V, \;\; i \neq j.$
 $w'(i,i) = w(i,i) = 0 \forall i \in V$
3. Solve APSP problem by using Dijkstra's Algorithm n times on G with weight function w'. Let D' be the shortest path weight Matrix obtained.

Johnson's Algorithm (contd)

4. Construct the shortest path weight Matrix D by using the formula

$$\delta(i,j) = \delta'(i,j) + h(j) - h(i)$$

5. Return D.

The total complexity is

$O(mn) + O(n^2 \log n + nm) + n^2$

which is $O(n^2 \log n + nm)$

Thus, Johnson's Algorithm solves APSP problem for a G with no negative cycles in $O(n^2 \log n + nm)$ time.

Construction of h

We now focus on the construction of h and prove that

$w'(i,j) = w(i,j) + h(i) - h(j) \geq 0$

Let $s \notin V$ and construct G' by adding s to V and adding directed edges

$(s,i)\ \forall\ i \in V$ with $w(s,i) = 0.$

That is $G' = (v', E')$ where $v' = V \cup \{s\}$

$E' = E \cup \{(s,i)|i\epsilon V\}$

Solve SSSP problem on G' with s as a source and define

$h(i) = \delta(s,i)$ in E, (which is also in E').

Weight Transformation

We know that

$$\delta(j) \le \delta(i) + w(i,j)$$

Hence,

$$w(i,j) + \delta(i) - \delta(j) \ge 0$$

Implying

$$w(i,j) + h(i) - h(j) \ge 0$$

Thus

$w'(i,j) = w(i,j) + h(i) - h(j) \ge 0$ for all $(i,j) \in E$

This completes our discussions on Johnson's Algorithm

Remark

- If G has a cycle with negative weight then G' also will have the same cycle as a negative weight cycle. Thus, if Bellman-Ford algorithm working on G' reports a negative cycle in G', we report G has a negative cycle and simply terminate the algorithm at this point. We will proceed with further steps only when we know that G has no negative cycles.

W3U1

- Minimum Spanning Trees

Thank You