Improvement of the Ford–Fulkerson algorithm

Jose Chavez

October 2, 2019

1 Definition Ford–Fulkerson algorithm

The following algorithms are greedy aproaches. We are going to present two algorithms that outperform the Ford-Fulkerson. These algorithms are Edmonds–Karp algorithm and Dinic's algorithm. At this point we should know that the Ford-Fulkerson algorithm computes the maximum flow in a flow network. It uses two speciall nodes , with wich It finds a path, these nodes are called source node and sink node. So, our task in Ford–Fulkerson is to fill the edges until we are out of available paths(augmenting paths), then we will retrieve the max flow value at the sink node. The time complexity of Ford–Fulkerson is $O(mf * E)$ were "mf" is the maximum flow in the graph and "E" is the number of edges. We reach the maximum flow when no more augmenting paths can be found . However, there is no guarantee that this stage will be reached, so at most we can ensure that the answer will be correct if the algorithm terminates. Taking the case where It ends, we know that each augmented path can be found in $O(E)$ and, in worst case we will add 1 unit flow in every iteration. Then we get our mentioned $O(mf * E)$ time complexity.

Terminologies

- Residual Graph: Its a graph which indicates additional possible flow.
- Residual Capacity: It's original capacity of the edge minus flow.
- Minimal cut: Also known as bottle neck capacity, which deicides maximum possible flow from source to sink through an augmented path.
- Augmenting path: Augmented path can be done in two ways:
  - Non-full forward edges
  - Non-empty backward edges

2 Edmonds–Karp algorithm

The algorithm is identical to the Ford–Fulkerson algorithm, except that the search order when finding the augmenting path is defined. The path found must be a shortest path that has available capacity. This can be found by a breadth-first search, where we apply a weight of 1 to each edge.The running time of $O(VE^2)$ is found by showing that each augmenting path can be found in $O(E)$ time, that every time at least one of the E edges becomes saturated, that the distance from the saturated edge to the source along the augmenting path must be longer than last time it was saturated, and that the length is at most V. Another property of this algorithm is that the length of the shortest augmenting path increases monotonically

3 Dinic's algorithm

This algorithm is similar to the above algorithm(Edmonds–Karp),It uses shortest augmenting paths.
In Edmond's Karp algorithm, we use BFS to find an augmenting path and send flow across this path. In Dinic's algorithm, we use BFS to check if more flow is possible and to construct level graph. In level graph, we assign levels to all nodes, level of a node is shortest distance (in terms of number of edges) of the node from source. Once level graph is constructed, we send multiple flows using this level graph. This is the reason it works better than Edmond Karp. In Edmond Karp, we send only flow that is send across the path found by BFS.
A flow is Blocking Flow if no more flow can be sent using level graph
Doing a BFS to construct level graph takes $O(E)$ time. Sending multiple more flows until a blocking flow is reached takes $O(VE)$ time. The outer loop runs at-most $O(V)$ time. In each iteration, we construct new level graph and find blocking flow.
So the outer loop runs at most $O(V)$ times. Therefore overall time complexity is $O(EV^2)$.

4 References

- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 26.2: The Ford–Fulkerson method". Introduction to Algorithms (Second ed.). MIT Press and McGraw–Hill. pp. 651–664. ISBN 0-262-03293-7.
- Algorithms and Complexity (eBook).
- Yefim Dinitz (2006). "Dinitz' Algorithm: The Original Version and Even's Version" (PDF). In Oded Goldreich; Arnold L. Rosenberg; Alan L. Selman (eds.). Theoretical Computer Science: Essays in Memory of Shimon Even. Springer. pp. 218–240. ISBN 978-3-540-32880-3.