Theory Assignment-5: ADA Winter-2024

Himanshu Raj (2022216)   Tanish Verma (2022532)

April 21, 2024

1 Flow Network construction

Add two additional vertices s and t, where s is a source vertex and t is a sink vertex. Let each box b_i be represented by two vertices u_i and v_i.
All edges are directed and have capacity 1. Add an edge from s to u_i for all i. Add an edge from v_i to t for all i.
Add an edge from u_i to v_j, iff $v_j(b_j)$ can be nested into $u_i(b_i)$. Note that the box b_j fits into b_i iff dimensions of b_j are strictly smaller than dimensions of b_i in a specific configuration (or rotation) as stated in the problem itself.

2 Claims

Claim 1: For u_i for all i, there is at most one outgoing edge having flow 1.
Reasoning: At most one unit of inflow is possible from s for any u_i.

Claim 2: For v_i for all i, there is at most one incoming edge having flow 1.
Reasoning: At most one unit of outflow is possible from any v_i to t.

For Claims 3 and 4, since they deal with maximum flow and edges cases, let us discuss a Case which we'll refer to as Case 1 below.
Case 1: Let's suppose there exists a set of boxes where every box can be nested into some box except the biggest one. Let the smallest box be b_1, the second smallest box be b_2, and so on, the largest box b_n.

Claim 3: The maximum flow can be at most n-1, where n is the number of boxes.
Reasoning: Let the smallest box be b_k, in Case 1. According to our construction, no incoming edge exists for v_k, and hence there will be no outflow from v_k.
Given n boxes, each box can have maximum flow 1 as per Claim 1 and Claim 2. Now, if for each box b_i in this case which is not b_k, there is a maximum flow 1 through u_i .
For the maximum network flow, there is a case where each u_i is connected to at least one v_i having a valid flow from u_i to v_i. (Refer the Case 1). Since, all boxes from b_2, b_3 ... b_n (but not the smallest box, b_1 in this case) , have a valid flow of 1, the maximum flow is n - 1.

Claim 4: The maximum number of edges can be at most $(n^2+3n)/2$, where n is the number of boxes.
Reasoning: In Case 1,
For b_n, the maximum number of boxes which can be nested inside it can be n-1 in the worst case, that is, n-1 edges from u_n to v_1, v_2 ... v_{n-1}.
For b_{n-1} this maximum number of edges is n-2 , that is, from u_{n-1} to v_1, v_2 ... v_{n-1}. Therefore, for b_i , this number is n - (i+1).
The summation of edges becomes (n - 1) + (n - 2) + (n - 3) + ... + 2 + 1 which reduces to n*(n-1)/2.
Also, there are n edges from the source s to all u_i, and another n edges from all v_i to sink t.
Therefore, maximum number of edges possible is 2*n + ((n*(n-1)/2) which is $(n^2 + 3n)/2$.

Claim 5: Minimum Number of Visible Boxes = Number of Boxes - Max Flow from s to t.
Reasoning: This is because the Max Flow from s to t gives the total number of boxes that can be nested. Since each flow of value 1 from u_i to v_j represents one nesting, and each nesting can be seen as reduction of one box from the total number of boxes.

3 Proof of Correctness

The constructed network is a maximum matching problem in bipartite graph (from A to B where A represents all u_i and B represents all v_j) .
After Flow Network construction, run the Ford-Fulkerson algorithm to find the maximum flow from s to t.
The Ford-Fulkerson algorithm gives a maximum-bipartite matching. This means that for every u_i matched with some v_j, v_j can be nested in u_i. This matching ensures that each box is nested inside only one other box at max
Finally, the max flow in this maximum matching is the maximum number of times one box can be nested into a bigger one. Since every time a flow from s to t exists (during the execution of Ford-Fulkerson), the number of visible boxes reduces by 1.

Therefore, Minimum Number of Visible Boxes = Number of Boxes - Max Flow from s to t

4 Runtime Analysis

Let n be the total number of boxes. For the flow network construction, $|V| = 2*n + 2$
and the maximum number of edges can be of order n^2 at the worst case. The maximum number of edges in a directed graph can be at most $|V|*(|V| - 1)$, but here this number is even less -> $|E| = 2*n + n(n-1)/2$ (following from Claim 4)
Therefore, the flow network construction takes $O(|V| + |E|)$ time in adjacency list representation, which is simply $O(n^2)$.

The Ford-Fulkerson algorithm used to check maximum matching takes O(value(flow)(|V| + |E|)-time.
The value(flow) can be n-1 in the worst case. Therefore, this takes $O(n-1)*O(n^2)$ time, which is $O(n^3)$ time in the worst case.