ADA-HW-5-Rubric

Kuber Budhija

April 2024

1 Formulating the Problem as a Flow-Network Problem (5 marks)

Following we give an algorithm for solving the problem. The have two main steps. Consider $\mathcal{B}$ denotes the set of the boxes. We assume that for any two boxes $b_i, b_j \in \mathcal{B}$ can not be contained inside another box say $b_k \in \mathcal{B}$ side-by-side. In other words, both b_i and b_j is contained inside box b_k if and only is either b_i is contained inside b_j or b_j is contained inside b_i. Following are the steps to the algorithm.

1. **Construct a flow graph.**
   (a) Initially consider a bipartite graph $\mathcal{G}(U \cup V, E)$ as follows.
      i. $U = V = \mathcal{B}$. This means that every node of U (and V) represents an unique box in $\mathcal{B}$ and each box of $\mathcal{B}$ corresponds to an unique node of U (and V).
      ii. For each pair of nodes (u, v) such that $u \in U$ and $v \in V$ where $u \neq v$, there exists an edge $(u, v) \in E$ if and only if the box corresponds to u can be contained inside the box corresponds to v.
   (b) Now we modify over $\mathcal{G}$ to generate a flow graph $\mathcal{F}(V_{\mathcal{F}}, E_{\mathcal{F}})$ as follows,
      i. For every edge $(u, v) \in E$ of $\mathcal{G}$ such that $u \in U$ and $v \in V$, set the weights of the edge $w(u, v)$ to be 1.
      ii. Introduce a new vertex s as source, and for every vertex $u \in U$ put an edge from s to u with edge weight $w(s, u) = 1$.
      iii. Similarly, introduce a new vertex t as sink, and for every vertex $v \in V$ put an edge from v to t with edge weight $w(v, t) = 1$.
2. **Find maximum flow of the graph.**
   - Find the maximum flow from s to t in the flow graph $\mathcal{F}$ using Ford-Fulkerson's algorithm.
3. **Output the nested boxes.**
   - For each edge $(u, v) \in E$ if (u, v) is saturated (or tight) in the maximum flow, then nest the box corresponds to u inside the box corresponds to v.

2 Justification of Maximum Flow Value (7 marks)

Lemma 1 *Let total number of boxes be $|\mathcal{B}| = n$. The flow-graph $\mathcal{F}$ has a max-flow of $k \leq n$, if and only if there exists a nested configuration of boxes with exactly $n - k$ visible boxes and there does not exist any nested configuration of boxes with $n - k'$ visible boxes where $k' > k$.*

Proof: if part: The if part is pretty straight forward. For any edge $(u, v) \in E$ consider $b_u \in \mathcal{B}$ and $b_v \in \mathcal{B}$ denotes the boxes corresponding to u and v respectively. For any saturated edge (u, v); b_u can be nested inside b_v. Thus by the solution of max-flow, at least, k distinct boxes can be put inside some other k distinct boxes, thus at most $n - k$ boxes will be visible in this configuration.

only if part: Let the nested configuration with minimized visible boxes has $(n - k)$ visible boxes. For contradiction let the max-flow be less than k. Let b_u is nested inside b_v; then we make the flow of the path $s - u - v - t$ as 1, making the flow of the edge (u, v) saturated. Observe that each of the k boxes are distinct and no two boxes are contained inside the same box side-by-side. This means if b_u is nested inside b_v and $b_{u'}$ is nested inside $b_{v'}$, then, the path $s - u - v - t$ and the path $s - u' - v' - t$ are disjoint, and thus each of such path contributes 1 to the flow. Since there are k nested boxes, thus the max-flow is at least k. □

The above lemma 1 leads to the following observation.

Observation 1 *Since at least one box must be visible in the solution, then the max-flow can be at most (n-1).*

3 Time Complexity Analysis (3 marks)

Assuming Ford-Fulkerson's algorithm takes $O(|E| \times f)$ time, $|E|$ is the number of edges in the graph and f be the max-flow of the graph, we analyze the time complexity of constructing the flow network and executing the algorithm.

Constructing the flow network involves creating $2n + 2$ nodes and at most $O(n^2)$ edges, resulting in a total of $O(n)$ nodes and $O(n^2)$ edges. Thus, the construction of the flow network takes $O(n^2)$ time.

The Ford-Fulkerson's algorithm runs in $O(|E| \times f)$ time. In our case, $f = O(n)$ and $|E| = O(n^2)$, so the algorithm takes $O(n^3)$ time.