

Cheat Sheet: min cost flow on bipartite graphs

Goal:

minimize cost subject to flow conservation and $\text{capacity constraints}$

Algorithm 1: Ford-Romney (or $\text{Successive Shortest Path}$)

1. $\text{Find a feasible flow}$ (e.g., max flow).
2. $\text{Set } u_i = 0$ (or $\text{any feasible values}$).
3. $\text{Repeat until optimal}$: $u_i + v_j = c_{ij} \rightarrow \text{Find shortest augmenting path}$.

Algorithm 2: Primal-Dual (or Cost Scaling)

$\text{Initialize } D_{ij} = u_i + v_j - c_{ij}$

- $\text{If } D_{ij} \leq 0 \rightarrow \text{Find shortest augmenting path}$.
- $\text{If } D_{ij} > 0 \rightarrow \text{Find shortest augmenting path}$.

Initialization:

$D_{ij} \leq 0 \rightarrow \text{Find shortest augmenting path}$!

$D_{ij} > 0 \rightarrow \text{Find shortest augmenting path}$:

1. $\text{Find shortest augmenting path}$ (e.g., Dijkstra).
2. $\text{Find shortest augmenting path}$ (e.g., Dijkstra).
3. $\text{Find shortest augmenting path}$: $+ \rightarrow - \rightarrow + \rightarrow - \dots$
4. $t = \text{min flow}$ (e.g., min flow).
5. $\text{Find shortest augmenting path}$ (e.g., Dijkstra).
6. $\text{Find shortest augmenting path}$ (e.g., Dijkstra).

Complexity:

- Time complexity is $O(m + n - 1)$ (e.g., min flow).
- Space complexity is $O(m + n - 1)$ (e.g., min flow).
- Time complexity is $O(m + n - 1)$ (e.g., min flow).
- Space complexity is $O(m + n - 1)$ (e.g., min flow).