28 Aug 2023

Min-Cost Bipartite Perfect Matching

Announcements.

1. Improved lecture notes on matchings to be posted tonight.
2. Enrollment capacity to be increased s.t. everyone who was on the waitlist yesterday will have a chance to enroll.

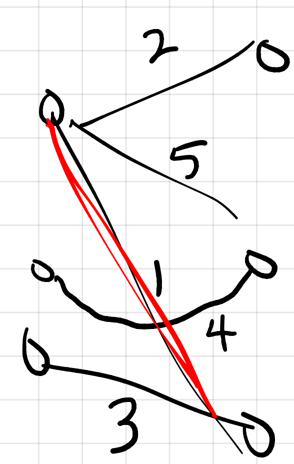
Min-Cost Bipartite Perfect Matching Problem.

Given: bipartite $G = (L, R, E)$ $|L| = |R| = \frac{n}{2}$

costs $c(u,v) \in \mathbb{R} \cup \{\infty\}$

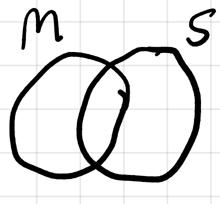
notation: $c(u,v) = +\infty$ if $(u,v) \notin E$

E.g.



Notation. If M is a matching and S is any edge set, the "incremental cost of S relative to M" is

$$\Delta c(S; M) = c(M \oplus S) - c(M)$$
$$= c(S \setminus M) - c(S \cap M)$$



(Quasi-)greedy Min-Cost Matching

initialize $M = \emptyset$
while M is not a perfect matching
    let P be an M-aug't path that minimizes $\Delta c(P; M)$
    $M \leftarrow M \oplus P$
endwhile
output M

Correct?

Yes, because of the following invariant.

After k iterations of the main loop, the matching M has minimum cost among all matchings with k edges.

Proof is by induction on k. (Surprisingly subtle.)

Why does G_M have no negative-cost cycles?

If C were a cycle in G_M with $\Delta c(C; M) < 0$ that would mean $c(M \oplus C) - c(M) < 0$

$$\Rightarrow \underline{c(M \oplus C)} < \underline{c(M)}$$

same # edges

This ineq. can never hold because of our loop invariant, and hence G_M has no negative cost cycles.

Running time: $\frac{n}{2}$ loop iterations

$O(mn)$ Bellman-Ford to find P in each iter.

Dijkstra would be $O(m + n \log n)$ if we could use it.

$\therefore$ $O(mn^2)$.

Our algorithm will be maintaining (and adjusting) vertex labels, y_u for each $u \in V$.

The reduced cost of edge $e = (u,v)$ will be

$$c^y(u,v) = \begin{cases} c(u,v) - y_u - y_v & \text{if } e \notin M \\ y_u + y_v - c(u,v) & \text{if } e \in M \end{cases}$$

Def. Labeling y and matching M are compatible if:

1. $y_u + y_v \leq c(u,v) \quad \forall (u,v)$
2. $y_u + y_v = c(u,v) \quad \forall (u,v) \in M$
3. $y_u = \max_{w \in L} \{y_w\} \quad \forall u \in L \cap F$
4. $y_v = \max_{w \in R} \{y_w\} \quad \forall v \in R \cap F$

If M is a matching and y is a compatible labeling:

1. We can use Dijkstra's algorithm to find an M-augmenting path P that minimizes $\sum_{e \in P} c^y(e)$

   (Min-cost path from $L \cap F$ to $R \cap F$ in G_M with respect to edge costs $c^y(u,v)$ which are non-negative.)

2. Using this path preserves the cost-minimization loop invariant.