CS4234: Optimisation Algorithms Tutorial 5

MAX-FLOW Part 2 + GRAPH-MATCHING Part 1

V1.7: A/Prof Steven Halim *October 02, 2023*

Discussion Points

Q1: Please read `https://onlinejudge.org/external/128/12873.pdf` and try to reduce this problem into a max flow problem, solve it using $O(n^2 \times m)$ Dinic's algorithm (assuming that you have such implementation ready), and analyze its time complexity. Follow-up question: Can we solve this problem as a pure MCBM problem and solve it with Augmenting Path algorithm++ (the one with randomized greedy pre-processing step)?

Q2: In Lecture 6, we have learned that there are some Bipartite Matching problem that admits Greedy solution. As an exercise, try solving `https://nus.kattis.com/problems/froshweek2` using a Greedy algorithm. Can we hope to pass the time limit if we use any MCBM (or Max Flow) algorithm?

Q3: Back in Lecture 1, we have learned about the MIN-VERTEX-COVER (MVC) problem. In T03, we also have learned about the MAX-INDEPENDENT-SET (MIS) problem. In Lecture 6, we have seen the special cases of solving MVC if it is asked on *Bipartite Graph*, but only the easier part of Konig's theorem, i.e., "In any bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.".

In problem `https://nus.kattis.com/problems/bilateral`, you are asked to find MVC and *one possible solution* too... In short, given a Bipartite Graph $G = (V_L, V_R), E$ of approximately 2000 vertices and up to 10 000 edges, show how to find the MVC (the optimal solution, not just the cardinality of the optimal solution) on G by reducing those problems into MCBM (you can then use a MCBM-specific algorithm or further reduce them into Max Flow problems and use a Max Flow algorithm).

Follow up question: How to find MIS instead of MVC? Then, what if the MVC/MIS problems asked are the weighted variants?

Q4: Discuss the solution for `https://nus.kattis.com/problems/taxicab`. Is this (a special case of) an NP-hard problem? Which PS4 task is this one?

Q5: Discuss the general idea of PS4 A.

OPTIONAL: In 06.MatchingLecture1.pdf and 07.MatchingLecture2.pdf (later), we use *Berge's theorem* for Augmenting Path(++), Kuhn-Munkres (Hungarian), and also Edmonds' Matching algorithm.

From `https://en.wikipedia.org/wiki/Berge%27s_theorem`, the wording of Berge's theorem is as follows: "In graph theory, Berge's theorem states that a matching M in a graph G is maximum (contains the largest possible number of edges) if and only if there is no augmenting path (a path that starts and ends on free (unmatched) vertices, and alternates between edges in and not in the matching) with M".

In 06.MatchingLecture1.pdf, Prof Halim has shown the proof. Students who are already comfortable with this proof can leave the tutorial. TA can re-present the proof one more time for the rest.