Advanced Algorithms. Home Exam 2022/23

The Problem

This exam problem connects several of the course topics.
We are given a bipartite graph $G = (X, Y, E)$ with two node sets X and Y, and a set E of edges, each connecting two nodes from X and Y. It has the special property that every node in X has a larger degree than every node in Y (which also implies $|X| < |Y|$). The problem is to compute a bipartite matching that covers all nodes of X, that is, every node in X should have some matching partner in Y.
Of course, we may solve any such problem instance by the known flow-based algorithm for maximum bipartite matchings. However, in real-time applications that require quick decisions, it may be a reasonable alternative to let the nodes in X randomly, uniformly, independently (and simultaneously!) choose their partners in Y, and if several nodes incidentally choose the same partner, to resolve these conflicts afterwards in some way, perhaps even without changing the rest of the random matching. If Y is much larger than X, such conflicts should be rare, such that random choices followed by minor adjustments may be simpler and faster than the deterministic standard algorithm.
The exam subject is to investigate this question quantitatively and in detail, and to judge to what extent the above intuition is right.
The scenario may appear in applications where, generally speaking, several agents (in X) must divide certain resources (in Y) among them. The differences to the randomized load balancing problem from the course are that only the pairings specified by the given graph are permitted, and that collisions must be avoided in the end.
You may, for example, work along these lines (this is only a suggestion): Can you compute the expected number of collisions (and perhaps also bound the

probabilities of large deviations from it) in a nice way? How does this number bahave, in relation to the minimum and maximum degree in X and Y, respectively? Is it always possible to resolve collisions without changing those selected edges that are not involved in collisions? Can the collisions simply be resolved in parallel, or do you have to work with augmenting paths? Is it easier to find an approximate matching, i.e., one that does not cover all of X but only the vast majority? etc.
Another possible direction to look at: Hall's marriage theorem guarantees, under some conditions, the existence of a bipartite matching which does cover X. Can it be applied here somehow? – If you consider this question, a little bit of self-study (e.g., on the web) is needed, as Hall's marriage theorem did not appear in the course.

Submission

Mail your final report to `ptr@chalmers.se` as PDF attachment (no other formats please!). Given the complexity of the task and the need to rewrite drafts, handwriting is discouraged. Write your name and study programme on the title page. The final submission deadline is announced on the course web page. Do not wait until the last minute, but submit when you are done. See also the Instructions below.
Quality is more important than quantity. However, as a rule of thumb, your report should have at least 5 pages of text (with usual font size, spacing, and margins), plus the title page and possible references.

Criteria for a Good Report

- Correctness: There are no major factual mistakes. In particular, you avoid invalid calculations with probabilities and random variables. The final “product” does not have to be perfect or contain brilliant results, but what you write must be sound.
- Depth: You provide some solid, substantial results that are fully worked out, not only some trivial observations or vague heuristic guesses.
- Clarity: Algorithms, as well as proofs of their properties, are well described. One can follow your arguments step by step. (See also the general grading criteria on the course web page.)
- Negative statements (that something is probably not possible to do) are motivated by good reasons.

Instructions

- The given problem description is *deliberately* vague (not by mistake). You have some freedom to choose your specific working directions, as long as you stick to the given problem and goal.
- Only a certain selection of concepts and tools from the course are suitable for this problem while most others are probably not, and it is part of the task to figure out what fits here – which would also be the typical situation in practice. (We have already had assignments that covered the main course contents. The exam does *not* have the ambition to cover everything again.)
- Do not hesitate to discuss, and to send questions or drafts. But take availability times into account; see the course web page. Note that feedback on the drafts cannot go too much into technical details; the purpose is rather to provide guidance and check whether you are on the right track.
- You can submit arbitrarily many drafts, at any time. Only the last version submitted before the deadline will count for your grade. In fact, **drafts are strongly encouraged**. The worst strategy would be to start thinking only a few days before the deadline – this would

most likely result in a very poor report, because the details may be more tricky than anticipated at first glance.

- Utmost academic honesty is expected. The words about cheating (see the course web page) apply also here. In particular, you must cite all literature you have used, acknowledge all sources of help, and always describe the contents in your own way. Also mark very clearly in the text what is taken from other sources and what are your own thoughts.