Graph Theory Applications EPFL, Spring 2014

Problem Set 11

Date: 13.05.2014 Not graded

Problem 1. Show how to use the Ford-Fulkerson algorithm to compute the maximum matching in a bipartite graph. Further, using the max-flow min-cut theorem, deduce a result on the sizes of the maximum matching and of the minimum vertex cover in a bipartite graph.

Problem 2. Consider an (l, r)-regular bipartite graph. This graph contains n variable nodes and nl/r check nodes. From each of the variable nodes depart l edges towards (not necessarily distinct) check nodes and from each of the check nodes depart r edges towards (not necessarily distinct) variable nodes. Take the code based on this graph. Show that the expansion of such a graph cannot be much larger than $1 - 1/l$ even for very small sets.

Hint. Start with one variable node and then include all its l neighbors as well as all the neighbors of its neighbors. Let $\mathcal{V}$ be all the variable nodes in this set and $\mathcal{C}$ all the check nodes in this set. Show that $\mathcal{V}$ expands by at most a factor $1 - 1/l + 1/(l(1 + l(r - 1)))$. Now generalize to larger neighborhoods and show that for increasing neighborhoods the maximum possible expansion approaches $1 - 1/l$.

Problem 3. Draw a $(3, 4)$-regular bipartite graph with $n = 12$ variable nodes. The edges are drawn at random, which means that for any edge of each variable node one selects uniformly at random the check node to be connected. Find a non-zero codeword for the corresponding code.

Assume that this codeword is transmitted. Pick uniformly at random two of the $n = 12$ bits and flip their value. Now run the flipping algorithm by hand and try to recover the error.

For these parameters (but a general length n) can you come up with a criterion on the graph which guarantees that any error pattern of weight two can be corrected?

Problem 4. Prove the triangle inequality in the Hamming space. More precisely, let x, y, z be elements of $\{0, 1\}^n$ for $n \geq 1$. Let $d(\cdot, \cdot)$ be the "distance" between two binary vectors, i.e., the number of positions in which the two vectors differ. Prove that $d(\cdot, \cdot)$ is indeed a distance, i.e., that it fulfills the following three requirements.

- $d(x, y) = 0$ if and only if $x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) + d(y, z) \geq d(x, z)$

Note. For error correction coding this has the following significance. If we construct a code of Hamming distance d, where d is odd, then a "minimum distance" decoder is guaranteed to decode correctly all error patterns of weight up to $(d - 1)/2$. Here, a minimum distance decoder proceeds as follows. Given the received word it checks if there is a codeword at distance $(d - 1)/2$ or less. If there is, it decides for this codeword. If not, then it declares an error. For suitably constructed codes, this procedure can be implemented in an efficient manner and this is why traditionally many codes were designed according to this criterion. Modern codes use very different principles based on sparse graphs and message passing.