Network Algorithms - Fall 2002

Dr. Guy Even

`http://hyde.eng.tau.ac.il/CO02/`

Handout #2: Maximum Flow (the push-relabel algorithm)

Deadline: Wednesday 30/10/02.

[CCPS] refers to the book "Combinatorial Optimization" by Cook, Cunningham, Pulleyblank, and Schrijver.

[CLR] refers to the book Introduction to Algorithms by Corman, Leiserson, and Rivest.

[GT] refers to the article Andrew V. Goldberg and Robert E. Tarjan,
A new approach to the maximum-flow problem, Journal of the ACM (JACM), Volume 35 , Issue 4 (October 1988). (You can download the paper only from TAU computers)

1. (Arnon) Present, for every n, a network with n vertices and an execution of the generic push-relabel algorithm in which at least one vertex's label reaches $d(v) = 2n - 1$. Try to find in example in in which as many vertex labels as possible reach this value.

2. ([CLR, question 27.4-8]) Suggest a modification of the push-relabel algorithm so that the following invariant holds:

   (a) $d(u) < n \Rightarrow d(u) = D_{G_f}(u, t)$.

   (b) $d(u) \geq n \Rightarrow n - d(u) = D_{G_f}(u, s)$.

   Can you suggest an implementation so that the time spent on maintaining this invariant is $O(mn)$.

3. ([GT],[CCPS, exercise 3.49]) Suppose that we wish to find a minimum cut (rather than a maximum flow).

   (a) Show how to compute a min-cut using only the vertex labels once the push-relabel algorithm terminates.

   (b) Suggest a modification that does not compute a max-flow but does compute a min-cut. Try to improve the bounds on the number of relabels and pushes by roughly a factor of 2.

4. *(not easy!)* ([GT],[CCPS],[CLR]) Suggest an implementation of the generic push-relabel algorithm that runs in $O(n^3)$ time.

5. Suppose we wish to inform the user while the generic push-relabel algorithm runs how the algorithm is progressing. To be concrete, we want to report (occasionally) an approximation of the ratio $e(t)/maxflow$ (upon termination this ratio should be 1!). Suggest how this can be done.