The Escape Problem*

June 22, 2006

An $n \times n$ *grid* is an undirected graph consisting of n rows and n columns of vertices, as shown in figure below. We denote the vertex in the ith row and the jth column by (i, j). All vertices in a grid have exactly four neighbors, except for the boundary vertices, which are the points (i, j) for which $i = 1, i = n, j = 1$, or $j = n$.

Given $m \leq n^2$ starting points $(x_1, y_1), \ldots, (x_m, y_m)$ in the grid, *the escape problem* is to determine whether or not there are m vertex-disjoint paths from the starting points to any m different points on the boundary. For example, the left grid in the below figure has an escape, but the one on the right does not.

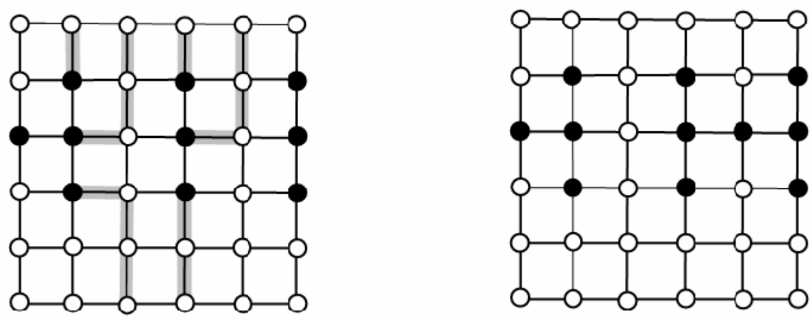

Figure 1. Starting points are black, and other grid vertices are white. **(left)** A grid with an escape, shown by shaded paths. **(right)** A grid with no escape.

1. Consider a flow network in which the vertices, as well as edges, have capacities. That is, the positive net flow entering any given vertex is subject to a capacity constraint. Show that determining the maximum flow in a network with edge and vertex capacities can be reduced to an ordinary maximum-flow problem on a flow network of comparable size.

2. Describe an efficient algorithm to solve the escape problem and analyze its running time.

*Taken from Cormen, Leiserson, Rivest, *Introduction to Algorithms*, MIT Press, 1989, Problem 27-1, pp. 625–626.

Solution.

1. For a flow network $G = (V, E)$ with both edge and vertex capacities we have edge capacity $c : E \to \mathbb{R}$ (as before) and vertex capacity $d : V \to \mathbb{R}$. The previous requirements of "(edge) capacity constraint", "skew symmetry", and "flow conservation" are the same. But there is a new requirement: for every $v \in V$ we require $\sum_{u \in V} \{f(u, v) \mid f(u, v) > 0\} \leq d(v)$.

   We construct a new flow network $G' = (V', E')$ with edge capacities only such that every maximum flow in G corresponds to a maximum flow in G', and vice-versa. Denote the edge capacity of G' by $c' : E' \to \mathbb{R}$. The construction of g' from G consists in splitting every vertex $v \in V$ into two vertices v_{in} and v_{out}, and inserting a new edge (v_{in}, v_{out}) with capacity $c'(v_{in}, v_{out}) = d(v)$ between the two parts of the split vertex. Formally,

   $$\begin{aligned} V' &= \{v_{in}, v_{out} \mid v \in V\} \\ E' &= \{(u_{out}, v_{in}) \mid (u, v) \in E\} \cup \{(v_{in}, v_{out}) \mid v \in V\} \\ c'(u_{out}, v_{in}) &= c(u, v), \text{ for every } (u, v) \in E \\ c'(v_{in}, v_{out}) &= d(v), \text{ for every } v \in V \end{aligned}$$

   If the source vertex in G is s and the sink vertex is t, then the source vertex in G' is s_{in} and the sink vertex is t_{out}.

   Let us determine the cost of constructing G' from G, and then running a maximum flow algorithm on G' - this is needed in part (2). The cost of splitting every vertex v into v_{in} and v_{out}, to obtain V' is $O(V)$. The cost of inserting a new edge (v_{in}, v_{out}) and assigning it capacity $d(v)$, repeated $|V|$ times, is also $O(V)$. The total cost of this construction is therefore $O(V)$. Note that $|V'| = 2 \cdot |V|$ and $|E'| = |E| + |V|$. If we run the Ford-Fulkerson method on G', the cost is $O(E' \cdot |f^*|) = O((E + V) \cdot |f^*|)$, where f^* is the maximum flow (assuming also that the capacities are integral values). If we choose to run the Edmonds-Karp implementation on G', its running time is $O(V'E'^2)$. Hence the total cost of finding a maximum flow in the original G is

   $$O(V) + O(V'E'^2) = O(V'E'^2) = O(2 \cdot V \cdot (E + V)^2).$$

2. We reduce the escape problem to the "maximum flow problem in a network with edge and vertex capacities". The reduction consists in transforming the input grid of the escape problem into such a flow network $G = (V, E)$. Initially there are m starting points on the grid and $4n-4$ boundary points. The construction of $G = (V, E)$ proceeds as follows:

   (a) Introduce an artificial source vertex s and connect s to each of the m starting points.

   (b) Introduce an artificial sink vertex t and connect each of the $4n-4$ boundary points to t.

(c) Each undirected edge between points u and v in the original grid is changed into two directed edges (u, v) and (v, u) in G.

(d) Assign edge capacity $= 1$ to each edge in G and every vertex capacity $= 1$ to each vertex in G.

This complete the construction of G. To determine whether there is an escape in the original grid, we find a maximum flow through G (using any of the algorithms studies in class): If the value of the maximum flow in G is m, then there is an escape in the original grid, else (if it is less than m) there is no escape in the original grid (make sure that you can prove that there is an escape in the original grid iff the maximum floe in $G = m$).

If you run Edmonds-Karp implementation of the Ford-Fulkerson method on G, the running time is $O(VE^2)$, as shown in the previous part. But we need to determine this running time in terms of the input size of the escape problem. The input grid has n^2 points and $2n^2 - 2n$ undirected edges. Hence, G has $2 + n^2 = O(n^2)$ vertices and $m + (4n-4) + 2(2n^2 - 2n) = m + 4n^2 - 4 = O(n^2)$ (directed) edges. Hence, the running time is $O(VE^2) = O(n^6)$. This upper bound is correct, but we can make it tighter. We showed earlier an upper bound $O((E+V) \cdot |f^*|)$ for all Ford-Fulkerson-based algorithms, and in the particular case of the escape problem, $|f^*| \leq 4n - 4$ (why?) and $|E| + |V| = O(n^2)$. Hence, a tighter bound for the running time is $O((E + V) \cdot |f^*|) = O(n^3)$.