SUMMARY OF DREYFUS' AN APPRAISAL OF SOME SHORTEST-PATH ALGORITHMS

Graph Theory
Uppsala University Ångströms Laboratory
Department of Math

William Paradell, Mira Åkerman, Max Callenmark

12-12-2023

Introduction

This paper is a short summary of Stuart E. Dreyfus' paper *An appraisal of some shortest-path algorithms*[3], where five different shortest paths problems are discussed and how good different algorithms are at solving the problems.

We strive with this summary to summarize the arguments presented pertaining to the different problems, as well as try to give an overview of the algorithms presented to simplify understanding.

Part 1: The shortest-path problem

To start off the shortest path problem Dreyfus presents the problem in greater detail as well as a computationally efficient algorithm solving the problem by Whiting and Hillier. The algorithm will be skipped since it's presented in relatively great detail in Dreyfus's paper; but it's worth mentioning the computational complexity of it, since it's referenced many times while he discusses other algorithms. The algorithm requires $\frac{N(N-1)}{2}$ additions and $N(N-1)$ comparisons to solve the problem; where N is the number of nodes. Next he presents other algorithms pertaining to the problem and starts off with Pollack and Wiebenson's paper[6], but does not present the algorithm, which will be done in this summary, to further simplify understanding of the points presented pertaining to it. In the paper a duality is presented with Minty's and the Ford-Fulkerson algorithm (where the shortest path is a special case of the Ford-Fulkerson algorithm for network flows), since the Ford-Fulkerson algorithm is relatively well known, only Minty's algorithm will be presented before the duality is explained. Minty's algorithm as it is presented in Pollack and Wiebenson's paper:

For the shortest-path problem between two specified cities (nodes). Label the starting city 'A' and the desired destination city 'B' (the terminating city).

1. Label the starting city A with the distance 'zero'

2. Look at all one-way streets with their 'tails' labeled and their 'heads' unlabeled. For each such street, form the sum of the label and the length. Select a street making this sum a minimum and place a check-mark beside it. Label the 'head' city of this street with the sum. Return to the beginning of 2

According to Dreyfus, Minty's algorithm is combined with the Ford-Fulkerson algorithm to make the *Minty-Ford-Fulkerson* algorithm which is stated to be easily programmed, but on the other hand significantly slower than the algorithm he started off by presenting, needing a total of $\frac{N^3}{6}$ additions and comparisons, versus the $\frac{N(N-1)}{2}$ additions and $N(N-1)$ comparisons of the previous algorithm. Dreyfus does present a paper by Whiting and Hillier that discusses a modification to the algorithm that speeds up the Minty-Ford-Fulkerson algorithm, but according to Dreyfus, it's not enough in order to compete with Dijkstra's algorithm.

Part 3: Determination of the second-shortest path

Dreyfus also talks about the problem of finding the second-shortest path through a network. To solve this problem he talks about different methods. Firstly Dreyfus refers to the method described by Hoffman and Pavley[4]. The method says that the second-shortest path between specified initial and terminal nodes is the deviation from the shortest path. The deviation from the shortest path is defined "to be a path that coincides with the shortest path (j may be the origin or the terminal node), then deviates directly to some node k not the next node of the shortest path, and finally proceeds from k to the fixed terminal node via the shortest path from k". Using this method will approximately use MK additions and comparisons, beyond the ones needed to find the shortest path, where M is the average node outgoing edges and K is the number of edges in the shortest path. Secondly Dreyfus refers to the method described by Bellman and Kalaba [1]. This method uses the equation:

$$v_i = \min \begin{bmatrix} \min_{i \neq j}{}_2 (d_{ij} + u_j), \\ \min_{j \neq i}{}_1 (d_{ij} + v_j) \end{bmatrix}, \qquad (i = 1, \dots, N-1) \tag{1}$$
$$v_N = \min_i{}_1 [d_{Ni} + u_i].$$

In this equation u_i is defined as the length of the shortest path from the node i to the terminal node N and v_i as the length of the second shortest path. The term $\min_k(x_1, \dots, x_n)$ is defined as the k:th smallest value of the quantities x_i. The term $\min_2(d_{ij} + u_j)$ determines the best path beginning in node i and deviating from the shortest path in node i and the term $\min_1(d_{ij} + v_j)$ evaluates the best path with any first edge and the second best continuation. Bellman and Kalaba recommends solving the equation above using an iterative procedure until convergence by elevating v_i on the left with $(k+1)$ and v_j on the right with (k). Using this method will need an average of NML additions and comparisons, where L is the number of iterations needed. Dreyfus compares this method to Hoffman and Pavley and says that Hoffman and Pavley's method is the better one because it requires less calculations. Dreyfus also writes about a method from Pollack's[5] paper to find the third-shortest path from all initial nodes to N. We get this path by following

$$w_i = \min \begin{bmatrix} d_{ik} + v_k \\ d_{im} + v_m \\ \min_{j \neq i}{}_3 [d_{ij} + u_j] \end{bmatrix}. \tag{2}$$

Let w_i represent the length of the third-shortest path from i, v_i the second-shortest path and u_i the shortest path. Let k be the verticy following i on the shortest path and m the vertices following i on the second shortest path. If u_i and v_i is determined then w_i can be computed node by node, this by first computing nodes that are one egde away from N, then two etc.

Part 4: Time-dependent length of arcs

Dreyfus refers to a paper by Cooke and Halsey[2] that studied the problem of finding the fastest paths between cities, where travel time depends on departure from the starting city. In the paper they refer to two cities as i and j and denote $d_{ij}(t)$ as the travel time where t is the time of departure and defines a function $f_i(t)$ as the minimum travel time to N.

$$\begin{aligned} f_i(t) &= \min_{j \neq i}[d_{ij}(t) + f_j(t + d_{ij}(t))] \\ f_N(t) &= 0 \end{aligned} \tag{3}$$

The authors present an iterative algorithm for finding the quickest paths from all cities to city N. Assume all cities are connected, even if it takes an infinite time. The algorithm can be summarized as follows:

1. Start at city i at time 0
2. Define T as the maximum time taken over all initial cities
3. Use tentative and permanent labels for nodes (cities)
4. Permanently label the initial node i as 0 and all the others nodes as infinity
5. Tentatively label nodes with the minimum of the current label and the sum of the current label and the time taken to travel to that node.
6. Find the minimum, non-permanent node label and declare it permanent.
7. Use the selected node to update labels at other tentatively labeled cities.
8. Repeat steps 6-7 until city N is permanently labeled.

The procedure requires most N^2T^2additions and comparisons. After most N^2 comparisons and $\frac{N^2}{2}$ additions, the quickest paths to all nodes, including N, are determined. A similar result is achieved by using a variant of the Dijkstra algorithm, where $\frac{N^2}{2}$ additions and $2N^2$comparisons are required. If the quickest path from all cities to N is wanted the algorithm must be repeated $N-1$ times. This procedure compares favorably in computation and required assumptions with Cooke and Halsey's algorithm.

Part 5: Shortest path visiting specified nodes

The last problem Dreyfus discusses in the paper is the problem of finding the shortest path between two specified nodes 1 and N that pass through $k-1$ nodes $(2, 3, \ldots, k \leq N-1)$. Dreyfus critiques a solution given to the problem by Saksena and Kumar[7]. The solution proposed by Saksena and Kumar assumes that the shortest path from a specified node i to the destination node N, passing through at least p specified nodes can be broken down into two problems:

1. Finding the shortest path from i to some specified node j.
2. Finding the shortest path from j to N, passing through at least $p-r-1$ specified nodes, where r is the number of specified nodes on the shortest unrestricted path from i to j.

Dreyfus claims that this procedure is flawed. If a specified node exists on the shortest path from i to j it also exists on the subsequent path from j to N and is therefore counted twice. If any potential path associated with node j does not allow node duplication the scenario is not allowed, and the option of initially traveling from i to j via the shortest path is eliminated from consideration. Dreyfus further claims that the procedure overlooks the possibility that a slightly longer continuation from j, passing through at least $p-r-1$ specified nodes and avoiding node duplication, could result in a better path than the best remaining path that satisfies the conditions outlined above. Dreyfus asserts that the problem can be correctly solved as follows assuming paths with loops are admissible:

1. Solve the shortest path problem for the N-node network for all pairs of initial and final nodes. Let d_{ij} represent the length of the shortest path from node i to j.
2. Solve the $k+1$ "traveling-salesman" problem for the shortest path from 1 to N passing through nodes $2, 3, \ldots, k$ where the distance from node i to j is d_{ij}.

Notes

Some notes that didn't fit with the flow of the summary. In part 1 it's worth mentioning that it's not clear what algorithm from Pollack and Wiebenson's paper Dreyfus has named the Minty-Ford-Fulkerson algorithm, as it's not explicitly stated in the paper that there's a combination of Minty's algorithm and the Ford-Fulkerson algorithm; but in their section *Dual Solutions* they do discuss a changed Ford-Fulkerson algorithm. Dreyfus's paper originally covers 5 parts but because of group problems we decided to only cover 4 out of the 5 parts. The part we skipped was part 2, *The shortest paths between all pairs of nodes of a network.*

References

[1] R. BELLMAN and R. KALABA. "On kth Best Policies". In: *J. SIAM* 8 (1960), pp. 582–588.

[2] K. L. COOKE and E. HALSEY. ""The Shortest Route through a Network with Time-Dependent Internodal Transit Tim". In: *J. Math. Anal. and Appl.* 14 (1966), pp. 493–498.

[3] Stuart E. Dreyfus. "An appraisal of some Shortest-Path algorithms". In: *Operations Research* 17.3 (June 1969), pp. 395–412. DOI: 10.1287/opre.17.3.395. URL: https://doi.org/10.1287/opre.17.3.395.

[4] W. HOFFMAN and R. PAVLEY. "A Method for the Solution of the Nth Best Path Problem". In: *J. ACM* 6 (1959), pp. 506–514.

[5] M. POLLACK. "Solutions of the kth Best Route through a Network-A Re-view". In: *J. Math. Analánd and Appl.* 3 (1961), pp. 547–559.

[6] M. POLLACK and W. WIEBENSON. "Solution of the Shortest-Route Problem-A indent Review". In: Opns. Res. 8, 224-230 (1960).

[7] J. P. SAKSENA and S. KUMAR. ""The Routing Problem with 'K' Specified Nodes". In: *Opns. Res.* 14 (1966), pp. 909–913.