Journal of Advanced Computing Research Vol. 1, Issue 1 (2016) 20-21

Floyd's Shortest-Path Algorithm Theory

Aini Hayati binti Anuar [#1], Mohamed Faidz Mohamed Said [#2]

[#] *Universiti Teknologi MARA*
70300 Seremban Negeri Sembilan, MALAYSIA
[1] ainihayatianuar@gmail.com
[2] faidzms@ieee.org

***Abstract*— These studies are parallel algorithms for the Floyd's shortest-path algorithm theory for solving the all pairs shortest-path problem which can be categorized into two cases. For a better understanding, we would like to explain the algorithm theory which has the most unfavorable scenario runtime of O(n^3) for graphs with n vertices. Researchers have put a lot of effort in order to improve the algorithms and at the same time, developing the other algorithms also. Based on these studies, the important, advantages and disadvantages of the Floyd's shortest-path algorithm theory have been identified.**

***Keywords*: Parallel Algorithm, Shortest-Path Algorithm, Floyd's**

I. Introduction

Shortest-path problem can be prescribed on a directed, weighted graph, where the distances are referred as weights. New contributions and ideas keep arising in the scientific publications due to the problem itself is quite understandable and is being extensively studied (for example, paper by [1]). In operational research especially related to find a route between two nodes (point), the shortest path problem is considered as vital and a well-known problem [2].

Primarily, there exist two types of the shortest path problem:

- **Single source shortest path problem (SSSP):**
  Establishing the briefest paths from an origin node to every single other nodes of the graph.

- **All pairs shortest path problem (APSP):**
  The computation of the briefest paths for individual pair of vertices in the graph.

Normally, the shortest path problem are classified into cases without cycle(s) (Figure: 1(a)) and cases with cycle(s) (Figure: 1(b)) [3]. There are algorithms for both cases where an excellent result is assured [4, 5]. Every node can be an origin or sink in the cases with cycles there is no origin, or if there a sink (last terminal).

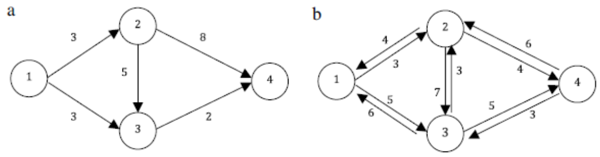

Figure 1. Two straightforward networks, (a) without a cycle and (b) with cycles.

In order to solve an all pairs shortest-path problem, many different algorithms can be exerted as well as the discrete optimization methods (for example, typical dynamic programming), customary heuristic search ways (for examples, genetic algorithm and simulated annealing), and typical graph algorithms (for examples, Johnson, Dijkstra and Floyd-Warshall [6]). A basic and commonly used algorithm, the Floyd-Warshall algorithm is implemented in order to evaluate the briefest paths in the middle of entire pairs of vertices in an edge weighted directed-graph.

This algorithm has the most unfavorable scenario runtime of $O(n^3)$ for graphs with n vertices. There exist a few algorithms with an improved most unfavorable scenario runtime [7-16], the terrific of these algorithms presently obtained a runtime of $O(n^3 \log \log n / \log 2\ n)$ [12] subsequently $O(mn + n^2 \log \log n)$ [17]. However, the Floyd–Warshall algorithm is less sophisticated compared to these algorithms which these algorithms also engage troublesome data structures. Therefore, the Floyd–Warshall algorithm is nevertheless the first option in many cases.

The purpose of this paper is to explain the Floyd's shortest-path algorithm theory for a better understanding which could be helpful for educational grounds. The arrangements for rest of this work are as follows. Section 2 provides an overview on the Floyd's algorithm. Section 3 discusses the Floyd's algorithm itself. Section 4 would be conclusion.

II. Background

Floyd algorithm is also well-known as the Roy-Warshall algorithm, Roy-Floyd algorithm or WFI algorithm. In 1962, this algorithm was issued by Robert Floyd. In spite of that, this algorithm is basically identical with the prior algorithm issued in 1959 by Bernard Roy and also by Stephen Warshall in 1962 to obtain the transitive closure of the graph. Also in 1962, Peter Ingerman was the first person to express the modern formulation of Warshall algorithm as three nested for-loop [18].

For the improvement of these algorithms, researchers have done quite a lot of effort and numerous other algorithms have been formed by bringing together these algorithms. For example, an excellent exploration of traditional algorithms and their applications [4] and a comprehensive comparative analysis on numerous shortest path algorithms [5]. Next, a recent APSP algorithm is introduced, for real-weighted directed graphs that run in $O(mn + n^2 \log\log n)$ time [17]. Other than that, is the awareness about adopting Floyd-Warshall algorithm with graphs that have negative weight cycles [3]. The applications of this algorithm also can be seen in discovering a ordinary expression indicating the ordinary language by restricted automaton, inversion of real matrices and optimal routing [19].

III. METHODOLOGY

The Floyd's algorithm is a basic algorithm for discovering out the briefest paths for each pair of vertices in a directed graph. It is a dynamic programming algorithm that solves a problem by combining the solutions to a sub-problems [20, 21]. The abstraction of intervening vertices is the base of this algorithm. Let d_{ij}^0 be the weight matrix, and d_{ij}^k be the briefest path from *i* to *j* with its intervening vertices in the set [6]. Then for *k* > 1,

$$d_{ij}^k = \min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1} \qquad (1)$$

Thus, the d_{ij}^n will give the briefest-paths matrix for the input graph. The algorithm is stated in Figure 2 beneath.

```
for i = 1 to n do
    for j = 1 to n do
        if there exists an edge from i to j then
            d[i, j] = w[i, j]
            r[i, j] = i
        else
            d[i, j] = infinity
            r[i, j] = -1
for k = 1 to n do
    for i = 1 to n do
        for j = 1 to n do
            if d[i, k] + d[k, j] < d[i, j] then
            d[i, j] = d[i, k] + d[k, j]
                r[i, j] = k
```

Figure 2. The Floyd-Warshall Algorithm.

The triple nested for loops decide the time complexity of the algorithm that obviously display *O*(*n*3) is the running time of this algorithm. The distance matrix consist of *d[i,j]* values yields the distance of briefest-path from *i* to *j* and the route of the briefest-paths can be quickly established from the former matrix containing *r[i, j]* values.

IV. CONCLUSION

In conclusion, the Floyd's shortest-path algorithm theory play an important role especially in solving the all pairs shortest-path problem. Other than that, it also has plentiful real-world applications such as in communications, transportation and electronics problems. While analyzing the Floyd's algorithm, the advantages, disadvantages and application of this algorithm also have been identified.

REFERENCES

[1] F. Glover, D. Klingman, and N. Phillips, "A new polynomially bounded shortest path algorithm," *Operations Research,* vol. 33, pp. 65-73, 1985.

[2] A. Aini and A. Salehipour, "Speeding up the Floyd–Warshall algorithm for the cycled shortest path problem," *Applied Mathematics Letters,* vol. 25, pp. 1-5, 2012.

[3] S. Hougardy, "The Floyd–Warshall algorithm on graphs with negative cycles," *Information Processing Letters,* vol. 110, pp. 279-281, 2010.

[4] G. Gallo and S. Pallottino, "Shortest path algorithms," *Annals of Operations Research,* vol. 13, pp. 1-79, 1988.

[5] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Mathematical programming,* vol. 73, pp. 129-174, 1996.

[6] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM (JACM),* vol. 24, pp. 1-13, 1977.

[7] T. M. Chan, "All-pairs shortest paths with real weights in O (n 3/log n) time," *Algorithmica,* vol. 50, pp. 236-243, 2008.

[8] T. M. Chan, "More algorithms for all-pairs shortest paths in weighted graphs," *SIAM Journal on Computing,* vol. 39, pp. 2075-2089, 2010.

[9] M. L. Fredman, "New bounds on the complexity of the shortest path problem," *SIAM Journal on Computing,* vol. 5, pp. 83-89, 1976.

[10] Y. Han, "A note of an O (n3/logn) time algorithm for all pairs shortest paths," *Information Processing Letters,* vol. 105, pp. 114-116, 2008.

[11] Y. Han, "An O (n 3 (log log n/log n) 5/4) Time Algorithm for All Pairs Shortest Path," *Algorithmica,* vol. 51, pp. 428-434, 2008.

[12] Y. Han and T. Takaoka, "An O (n 3 log log n/log 2 n) time algorithm for all pairs shortest paths," 2009.

[13] T. Takaoka, "A new upper bound on the complexity of the all pairs shortest path problem," in *Graph-Theoretic Concepts in Computer Science*, 1991, pp. 209-213.

[14] T. Takaoka, "A faster algorithm for the all-pairs shortest path problem and its application," in *Computing and Combinatorics*, ed: Springer, 2004, pp. 278-289.

[15] T. Takaoka, "An O (n3loglogn/logn) time algorithm for the all-pairs shortest path problem," *Information Processing Letters,* vol. 96, pp. 155-161, 2005.

[16] U. Zwick, "A slightly improved sub-cubic algorithm for the all pairs shortest paths problem with real edge lengths," *Algorithmica,* vol. 46, pp. 181-192, 2006.

[17] S. Pettie, "A new approach to all-pairs shortest paths on real-weighted graphs," *Theoretical Computer Science,* vol. 312, pp. 47-74, 2004.

[18] E. W. Weisstein, "Floyd-Warshall Algorithm," 2008.

[19] S. Sanan, L. jain, and B. Kappor, "Shortest Path Algorithm," *International Journal of Application or Innovation in Engineering & Management (IJAIEM),* vol. 2, July 2013.

[20] R. Bellman, "Dynamic programming and Lagrange multipliers," *Proceedings of the National Academy of Sciences,* vol. 42, pp. 767-769, 1956.

[21] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM,* vol. 5, p. 345, 1962.