

# THE SHORTEST NOT NECESSARILY THE BEST OTHER PATH ON THE BASIS OF THE OPTIMAL PATH

Tomasz Neumann<sup>1</sup>

<sup>1</sup>Gdynia Maritime University, Gdynia, Poland

## Abstract

*This paper presents a different perspective on the Dijkstra algorithm. In paper this algorithm is used to the further analyses to find another paths between nodes in maritime traffic. In many cases, the best solution for a single criterion is not sufficient. Finding more effective solutions can be the starting point to use for subsequent analysis or making decision by the captain of the vessel. Using advanced reasoning mechanisms it is possible to create a decision support system based on well-known Dijkstra algorithm.*

**Keywords:** Sea Transport, Networks and Graphs, Intelligent Transportation System

-----\*\*\*-----

## 1. INTRODUCTION

Rapid and accurate calculation of the estimated time of arrival at the port of destination of the ship at sea, is of great importance in many areas of the ocean shipping industry. For example, the route and schedule of ships on a day to day, it is important to know when the ship reached the port of destination and to be available for new loads. Estimated time of arrival can also be important in the planning of port operations. Give a reliable estimated time of arrival for ships entering the port, planning port operations, such as work assignments, loading/unloading equipment and harbours for ships can be more efficiently done.

The traditional way is to base the estimate of estimated time of arrival on ships. However, the new technology of satellite position reporting combined with electronic map allows users to land to check weather or update the master's estimated time of arrival at regular intervals, not interfering with the crew of the ship at all hours.

The paper presents an efficient algorithm for determining the estimated time of arrival in port the ship is at sea. The algorithm is implemented in a decision support system in planning the operation of ships, one of which is installed and used by several owners. By calculating the distance of the route between the ship and the port of destination, estimated time of arrival can be estimated by dividing the distance by the speed sailing. Calculating the distance between the ship and the port of destination may be considered to determine the shortest path between two points in the presence of polygonal obstacles, where one point corresponds to the vessel and the other end to the port of destination. Sections defining polygons are obstacles coast. Estimated time of arrival accuracy can be improved by defining special network structures in areas with limited speed, or are waiting for the pilots [3].

The vehicle routing problem lies in the design of optimal routes for a fleet of vehicles, usually in order to minimize

operating costs or the number of vehicles used. Several variations of this problem has been extensively studied in the literature optimization as efficient routing of vehicles have a great impact on logistics costs [7].

## 2. THE PROBLEM

Given that the total linear programming model of a simplified version of the problem of routing the vessel presents an unacceptable solution times for a typical daily planning process, taken a heuristic approach, deciding on your hand. Author decided on this approach for its implementation relatively simple calculation, as well as its record of good results with similar problems to the present.

There are several algorithms such as Dijkstra's algorithm, which is a single source-single destination shortest path algorithm, the Bellman-Ford algorithm to solve the shortest path algorithm with a free hand, A\* algorithm solves the single pair shortest path problems using a heuristic algorithm and Floyd Warshall algorithm to find all pairs of Johnson-perturbation and the shortest path algorithm to find the shortest path locally. Genetic algorithms are also used to finding shortest path [1]. In this paper to calculation will be used Dijkstra algorithm.

### 2.1 Dijkstra's Algorithm

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (ie the shortest path) between that vertex and every other vertex. It can also be used for finding the shortest cost path from one vertex to a destination vertex by stopping the algorithm is determined by the shortest path to the destination node. For example, if the vertices of the graph represent the city and are the costs of running paths edge distances between pairs of cities connected directly to the road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used

routing protocols in a network, in particular the IS-IS and Open Shortest Path First.

Short characteristic of Dijkstra algorithm [2].

- The input of the algorithm consists of a weighted directed graph  $G$  and a source vertex  $s$  in  $G$
- Denote  $V$  as the set of all vertices in the graph  $G$ .
- Each edge of the graph is an ordered pair of vertices  $(u,v)$
- This representing a connection from vertex  $u$  to vertex  $v$
- The set of all edges is denoted  $E$
- Weights of edges are given by a weight function  $w: E \rightarrow [0, \infty)$
- Therefore  $w(u,v)$  is the cost of moving directly from vertex  $u$  to vertex  $v$
- The cost of an edge can be thought of as (a generalization of) the distance between those two vertices
- The cost of a path between two vertices is the sum of costs of the edges in that path
- For a given pair of vertices  $s$  and  $t$  in  $V$ , the algorithm finds the path from  $s$  to  $t$  with lowest cost (i.e. the shortest path)

- It can also be used for finding costs of shortest paths from a single vertex  $s$  to all other vertices in the graph.

### 2.2 Proposed to Use

As indicated above, the algorithm is well known and widely used. Finding the shortest path between two vertices in graphs is not difficult. But not always the shortest path is the best. It is proposed to fast search method inferior alternatives. In such situation, decision maker can choose one of the alternatives and take all the consequences.

After finding the shortest path is proposed to remove from the graph piece belonging to the founded path. Disposal should be carried out as long as there is a lot of segments in the founded path. For each new graph is necessary to find the shortest path. Founded new path should be saved to an array of solutions.

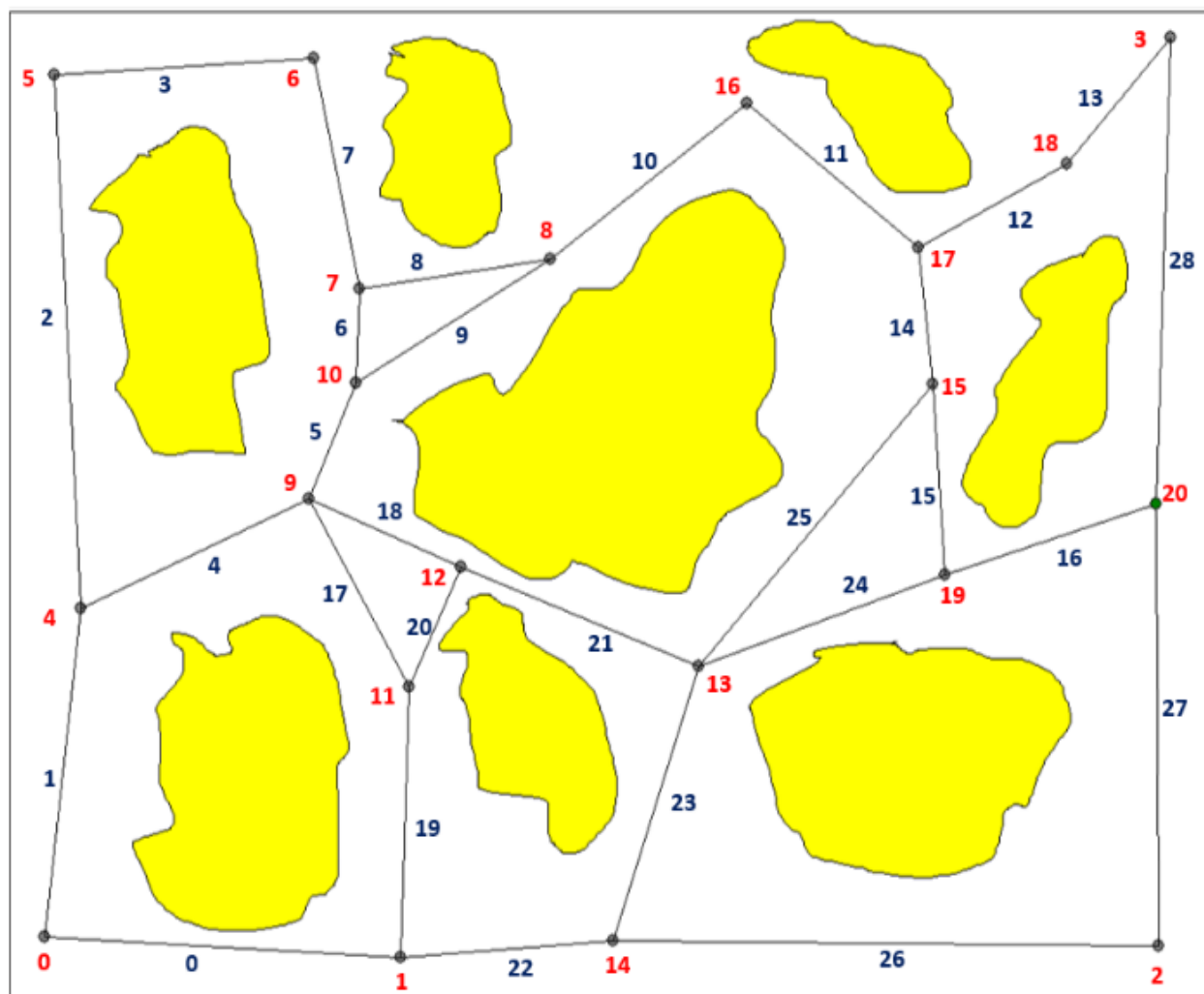


Fig 1: Scheme of traffic among islands

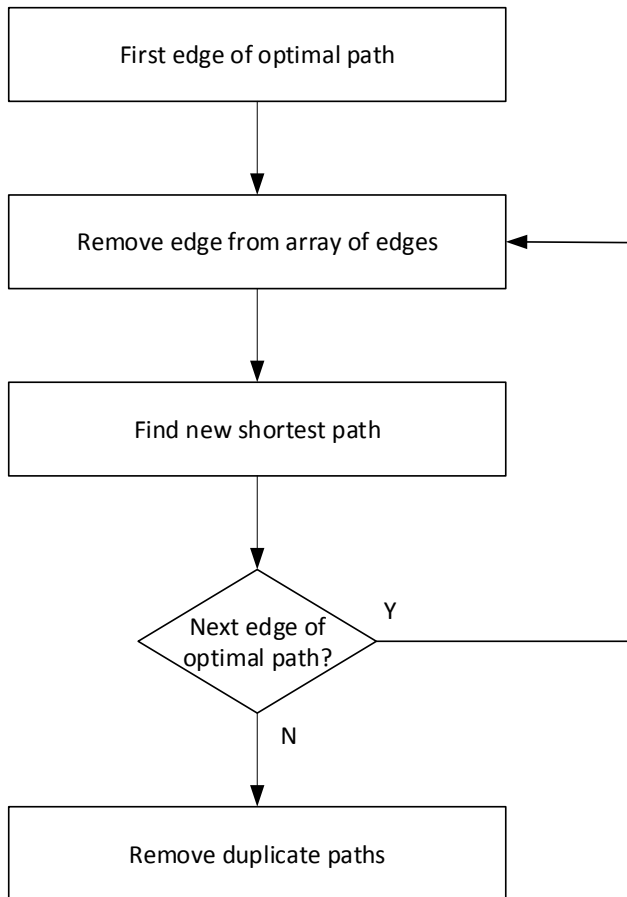


Fig 2: Diagram of searching new paths

The proposed scheme is shown in figure 2. Provides an overview of the optimal path, the removal of all further edges.

Resulting array can contain many of the same tracks. It is therefore removed from the same tracks. Of course, such removal is not a problem.

### 3. NUMERICAL EXAMPLE

The figure 1 shows an example of the unreal maritime restricted area. It consists of eight obstacles (in the form of islands), 21 turning points and 29 edges. Each edge is described by value of the distance between two vertices.

All connections are shown in the Table 1. Number of edge corresponds to the edge shown in Figure 1. Each connection is between initial node and final node. The distance between nodes describes a dimensionless measurement of the distance between the nodes.

Table 1: Table of all edges

| Number of edges | Initial vertex | Final vertex | Distance between the vertices |
|-----------------|----------------|--------------|-------------------------------|
| 0               | 0              | 1            | 295                           |
| 1               | 0              | 4            | 257                           |
| 2               | 4              | 5            | 415                           |
| 3               | 5              | 6            | 215                           |
| 4               | 4              | 9            | 207                           |

|    |    |    |     |
|----|----|----|-----|
| 5  | 9  | 10 | 98  |
| 6  | 10 | 7  | 73  |
| 7  | 7  | 6  | 183 |
| 8  | 7  | 8  | 160 |
| 9  | 10 | 8  | 187 |
| 10 | 8  | 16 | 203 |
| 11 | 16 | 17 | 181 |
| 12 | 17 | 18 | 139 |
| 13 | 18 | 3  | 130 |
| 14 | 17 | 15 | 107 |
| 15 | 15 | 19 | 148 |
| 16 | 19 | 20 | 183 |
| 17 | 9  | 11 | 168 |
| 18 | 9  | 12 | 137 |
| 19 | 1  | 11 | 210 |
| 20 | 11 | 12 | 102 |
| 21 | 12 | 13 | 212 |
| 22 | 1  | 14 | 176 |
| 23 | 14 | 13 | 225 |
| 24 | 13 | 19 | 216 |
| 25 | 13 | 15 | 293 |
| 26 | 14 | 2  | 452 |
| 27 | 2  | 20 | 343 |
| 28 | 20 | 3  | 362 |

Result of the algorithm is a path with a length equal 1365. The algorithm indicated that the shortest distance between the vertex labelled 0 and 3 leading by edges: 0, 22, 23, 25, 14, 12, 13. The shortest path is presented in Table 2 and in Figure 3.

Table 2: Description of shortest path

| Number of edges | Initial vertex | Final vertex | Distance between the vertices |
|-----------------|----------------|--------------|-------------------------------|
| 0               | 0              | 1            | 295                           |
| 22              | 1              | 14           | 176                           |
| 23              | 14             | 13           | 225                           |
| 25              | 13             | 15           | 293                           |
| 14              | 17             | 15           | 107                           |
| 12              | 17             | 18           | 139                           |
| 13              | 18             | 3            | 130                           |
| <b>Total</b>    |                |              | <b>1365</b>                   |

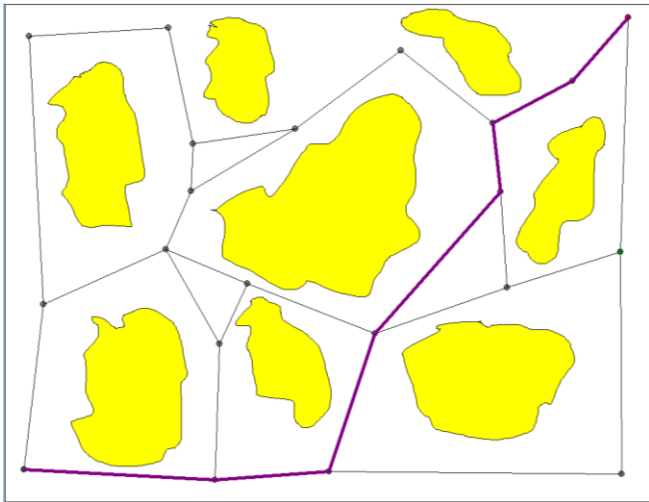


Fig 3: Shortest path

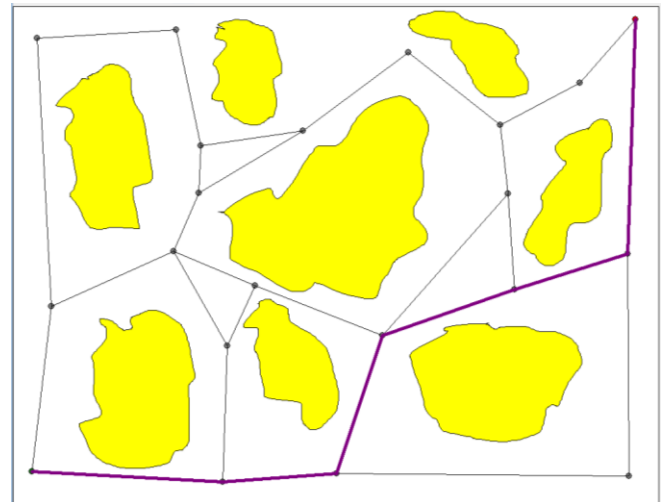


Fig 5: Third path

An easy way to find another quick way from the source to the target is to modify the optimal path, found a particular method. The article proposes to remove one edge. The cycle was repeated for each edge of the optimal path. In this way calculated two other paths 1,4,5,9,10,11,12,13 and 0,22,23,24,16,28:. Detailed data are shown in the Figures and tables.

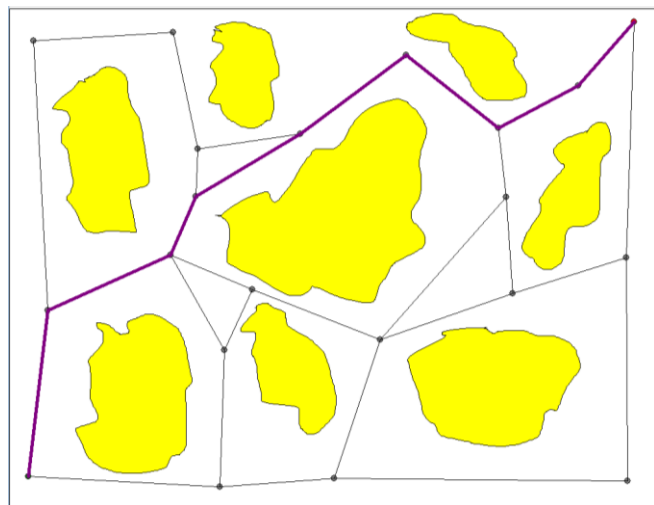


Fig 4: Second path

Table 3: Description of second path

| Number of edges | Initial vertex | Final vertex | Distance between the vertices |
|-----------------|----------------|--------------|-------------------------------|
| 1               | 0              | 4            | 257                           |
| 4               | 4              | 9            | 207                           |
| 5               | 9              | 10           | 98                            |
| 9               | 10             | 8            | 187                           |
| 10              | 8              | 16           | 203                           |
| 11              | 16             | 17           | 181                           |
| 12              | 17             | 18           | 139                           |
| 13              | 18             | 3            | 130                           |
| <b>Total</b>    |                |              | <b>1402</b>                   |

Table 4: Description of third path

| Number of edges | Initial vertex | Final vertex | Distance between the vertices |
|-----------------|----------------|--------------|-------------------------------|
| 0               | 0              | 1            | 295                           |
| 22              | 1              | 14           | 176                           |
| 23              | 14             | 13           | 225                           |
| 24              | 13             | 19           | 216                           |
| 16              | 19             | 20           | 183                           |
| 28              | 20             | 3            | 362                           |
| <b>Total</b>    |                |              | <b>1457</b>                   |

For the purposes of demonstration and calculations developed computer application [6] which graphically created schema presented in the article. For middle-class computer all the calculations were done in less than 150ms. Such a short calculation time can be a prerequisite for further research into the search for alternative paths.

The results require further calculations. According to the author to be in the discussion to consider not only the dimensionless values describing the edges, but also values such as time, cost, security etc. It is necessary to use the mechanism of multiple-choice decision or data fusion [4,5].

#### 4. CONCLUSIONS

The Dijkstra algorithm is well known. It was first published half a century ago. To this day, finding connections between vertices is used. But not always the shortest path is the best. It is to consider various criteria. This paper is an introduction to further research.

It has been shown that with the use of today's computers to find paths can be easy and quick.

#### REFERENCES

[1]. Bagheri H., Ghassemi H., Dehghanian A., 2014: Optimizing the Seakeeping Performance of Ship Hull Forms Using Genetic Algorithm. TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 8, No. 1, pp. 49-57

- [2]. Dijkstra, E.W., 1959: A note on two problems in connexion with graphs. *Numerische Mathematik*. 1, 269–271.
- [3]. Fagerholt, K., Heimdal, S., Loktu, A., 2000. Shortest path in the presence of obstacles: an application to ocean shipping. *Journal of the Operational Research Society* 51, 683–688.
- [4]. Gopika, N.A., Deeo, S. 2013. A survey on optimal route queries for road networks, *International Journal of Research in Engineering and Technology*, 02, 12, 447-450
- [5]. Neumann T., 2008: Multisensor Data Fusion in the Decision Process on the Bridge of the Vessel. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 2, No. 1, pp. 85-89
- [6]. Neumann T., 2011: A Simulation Environment for Modelling and Analysis of the Distribution of Shore Observatory Stations - Preliminary Results. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 5, No. 4, pp. 555-560
- [7]. Romeroa, G., Duran, G., Marengo, J. Weintraub, A. 2013. An approach for efficient ship routing. *International Transactions in Operational Research* 00, 1-28