**Improvement of Dijkstra’s Algorithm and Its Application in Route Planning**

Faisal Amin Abir  
American International University-Bangladesh,   
**faisalamin50106@gmal.com**

In the realm of optimizing route planning for road networks, much attention has been given to the study of Dijkstra's algorithm. While this algorithm offers significant advantages, it is not without its limitations when attempting to find the best possible path between two points. These drawbacks primarily revolve around challenges concerning how data is stored and the extent of the search conducted by the algorithm. To elaborate, the issues with the storage structure refer to how the algorithm stores and manages the information required to navigate the road network efficiently. Additionally, the searching area pertains to the breadth or range within which the algorithm explores potential paths, sometimes resulting in suboptimal solutions. In response to these identified shortcomings, this research aims to improve the algorithm in two main aspects: first, by refining how data is stored and organized, and secondly, by constraining the algorithm's search scope to enhance its accuracy and efficiency. The effectiveness of these enhancements is demonstrated through a comprehensive analysis of experimental results, showcasing the positive impact of these modifications on the algorithm's performance in finding optimal routes within road networks.

**CCS CONCEPTS •** Insert your first CCS term here • Insert your second CCS term here • Insert your third CCS term here

**Additional Keywords and Phrases:** Dijkstra’s algorithm; road network; route planning; storage structure; restricted searching area

**ACM Reference Format:**

First Author’s Name, Initials, and Last Name, Second Author’s Name, Initials, and Last Name, and Third Author’s Name, Initials, and Last Name. 2018. The Title of the Paper: ACM Conference Proceedings Manuscript Submission Template: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock ’18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1. **Introduction**

Based on a topological structure of the road network, Path planning is a process to search the optimal route for a vehicle before or during traveling to get its destination and a specific application of the shortest path problem in vehicle navigation system [1]. Dijkstra’s algorithm is the most classical and mature algorithm for searching a shortest path in the graph, however, this algorithm has the highly time complexity and takes up a larger storage space. In order to overcome the shortcomings of Dijkstra’s algorithm, a considerable number of researches have been done by many experts and scholars, and their research results have respective traits and merits, but the theoretical basis of related algorithm is Dijkstra’s algorithm [2~7]. Combining actual traffic network’s distribution characteristic, this paper has improved the classical Dijkstra’s algorithm to reduce the complexity of time and space and decrease the searching scale of the algorithm, and improve this algorithm's running efficiency.

**Literature Review**

Road network route planning is a critical component in transportation systems, aiming to optimize traffic flow and travel efficiency. Over the years, various algorithms have been employed for this purpose, with Dijkstra's algorithm emerging as a prominent method due to its capability to find the shortest path in weighted graphs. However, while widely utilized, Dijkstra's algorithm presents certain limitations that have motivated researchers to explore enhancements, seeking to improve its efficacy in road network route planning.

**A. Improvement of Storage Structure**

Figures are adopted to explain an improvement principle of the storage structure. Shown in Figure 1, if we adopt the adjacency matrix to store data of classical Dijkstra’s algorithm, and then Figure 2 is used to represent the adjacency matrix structure of the topological relations among each point. A 1010 × matrix is used to store the topological relations between points and points, the same number of rows and columns. The numeral values of the matrix ([xi, xj]) are corresponding to the values between points i and j, with the value is zero when start point and end point are the same point, and the value is ∞ when there is no direct path between two points. Then this matrix contains a large number of zero and ∞, which increases the number of invalid cycles and takes the massive space in storage. Therefore, it is unscientific from the perspective of space and time.

**B. Restricted Searching Area**

Because the classical Dijkstra’s algorithm is a comprehensive search process, there must be redundant. According to characteristics of Dijkstra’s algorithm and the spatial distribution feature of the real road network, the algorithm restricts the searching area reasonably. Because the straight line between two points is the shortest length, the direction from start point to destination point is generally strike of the shortest path when we plan the route of the real road network. Namely the ultimately actual shortest path between two points is generally on both sides of the connection line, and usually in the vicinity. If there is only one edge between two points, the edge itself is the 1903shortest path. However, sometimes maybe there exists reserve path of short distance in the two point’s vicinity. Namely, in order to enter the right traveled lane ，the vehicle travels the route.

1. **More about the submission template**

Thissubmission version of your paper should not have headers or footers, these will be added when your manuscript is processed after acceptance. It should remain in a one-column format—please do not alter any of the styles or margins.

*If a paper is accepted for publication*, authors will be instructed on the next steps. Authors must then follow the submission instructions found on their respective publication’s web page. Once your submission is received, your paper will be processed to produce the formatted Word, PDF, and HTML5 output formats, which will be provided to you for review, revision/resubmission (if applicable), and approval.

1. **Inserting CCS concepts**

The new template enables you to import required indexing concepts for your article from the ACM Computing Classification System (CCS) using an indexing support tool found in the ACM Digital Library (DL). The tool generates formatted text after you have selected your terms. To insert CCS terms into your document, copy and paste the formatted text from the CCS tool using the “https://dl.acm.org/ccs/ccs.cfm” link into the “CCS CONCEPTS” section.

An additional step is necessary to ensure that the proper CCS terms are added to the Digital Library citation page: from the “view CCS TeX Code” listing, click on “Show the XML Only.” Highlight and copy the XML code from the window. You must insert the XML code into your Word document’s properties: from your Word document, click on “**File**”, then click on the “**Info**” tab on the left-hand side panel, then click “**Properties**” and select “**Show All Properties.**” Click within the “Comments” metadata field and paste the XML data.

1. **Inserting Content Elements**

The next subsections provide instructions on how to insert figures, tables, and equations in your document.

1. **Tables**

Tables are “float elements” which should be inserted after their first text reference and have specific styles for identification.  Do not use images to present tables, or they will be inaccessible to readers using assistive technologies.

Authors can insert tables by using the MS Word option (INSERT ->Table) and providing the required row and column size. Every table must have a caption (title) above it, which must have the **“Table Caption**” style applied. Please note that tables **should not** be supplied as image files, but if they are images, they must have the “Image” style applied. As an example, Table 1 shows all the styles available in this template, to be applied to the respective element of your text.

Table 1: Styles available in the Word template

| Style Tag | Definition | Style Tag | Definition |
| --- | --- | --- | --- |
| Title document | main title of article | List Paragraph | list items |
| Subtitle | subtitle of article | Statements | math statements |
| Authors | author name | Extract | block quotations |
| Affiliation | author affiliation information | Algorithm Caption | caption for algorithm |
| AUTH Notes | footnote to author(s) | AckHead | heading for acknowledgements |
| Abstract | abstract text | AckPara | acknowledgements text |
| CCSHead | heading for CSS Concepts | GrantSponsor | sponsor of grant |
| CCSDescription | CSS terms | GrantNumber | number for the grant |
| KeyWordHead | heading for keywords | ReferenceHead | heading for references |
| Keywords | keywords text | Bib\_entry | references |
| ORCID | author's ORCHID # | AppendixH1 | appendix heading level 1 |
| Head1 | heading level 1 | AppendixH2 | appendix heading level 2 |
| Head2 | heading level 2 | AppendixH3 | appendix heading level 3 |
| Head3 | heading level 3 | TableCaption | title of table |
| PostHeadPara | first paragraph after a heading | TableHead  TableFootnote | column head of table  footnote to table |
| Para | Subsequent paragraphs of general text | Image | figures |
| ParaContinue  Display Formula | flush left text after display items like math equations, lists etc.  numbered math equation | DOI | Digital object identifier |
| DisplayFormulaUnnum | unnumbered equations | Label | labela |
| ComputerCode | Display Computer codes | In-text code | intext computer code |
| Short Title | Short title of article | History | Dates of article |

a This is example of table footnote.

Tables can be very difficult for people using screen reader technology to understand unless they include markup that explicitly defines the relationships between all the parts (i.e.: headers and data cells). *A key to making data tables accessible to screen reader users is to clearly identify column and row headers.* In Word, authors should identify which row or rows contain column headers. Below are the steps to do this:

1. Select that table’s row, then right-click the row and select “Table Properties”;
2. In the *Table Properties* window, click the *Row* tab and select the box that says “Repeat as header row at the top of each page.”

Or

Apply the “table head” style by highlighting the respective row and applying the “**TableHead**” style found in the “Body Element” section of the ACM Master Article Template.

1. **Figures**

Figures are “float elements” which should be inserted after their first text reference, and have specific styles for identification. Insert a figure and apply the “**Image**” paragraph style to it. For the figure caption, apply the style “**Figure Caption.**”

To accommodate readers with color vision differences, figures should still be usable when printed in grayscale. Refer to elements of the figure with non-color terms, for example “indicated as squares” instead of “indicated in blue”. Use different patterns in bar charts, different line patterns in graphs, and different shapes in plots to distinguish groups of elements and reinforce color differences.

1. *Half Width Figures.*

Figure 1 is an example of a figure and caption spanning the half-page width (one column in a two column format) with the styles applied.  If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.



Figure 1: 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB)

1. *Full Width Figures.*

Figure 2 is an example of a figure and caption spanning the full-page width with the styles applied. If your figure contains third-party material, you must clearly identify it as such, as shown in the examples.

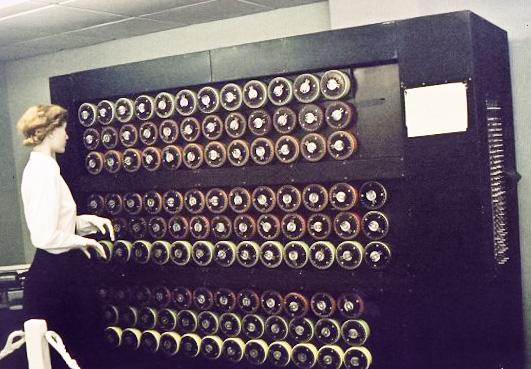


Figure 2: Mockup of a bombe machine at Bletchley Part. Photograph by Sarah Hartwell. [Public domain], via Wikimedia Commons. (https://commons.wikimedia.org/wiki/File:TuringBombeBletchleyPark.jpg)

1. *Multi-part figure.*

Authors can also insert a multi-part figure above a single caption. Every inserted figure must have the “Image” style applied. Below are instructions regarding how to insert a multi-part figure in your paper.

* If the author wants to insert two multi-part images, they must draw a one row and one column table and insert the images one-by-one in the cells.
* If the author wants to insert three multi-part images, they must draw a one-row and three-column table and insert the images one by one in all three cells.
* If the author wants to insert four multi-part images, they must draw a two-row and two-column table and insert the images one-by-one in all four cells. (see the following example):

| Figure 2: The layout of multipart images should be as per the above example within the table in image 1. | Figure 2: The layout of multipart images should be as per the above example within the table in image 2. |
| --- | --- |
| Figure 2: The layout of multipart images should be as per the above example within the table in image 3. | Figure 2: The layout of multipart images should be as per the above example within the table in image 4. |

Figure 3: The layout of multipart images should be as per the above example within the table. All images must have the “Image” style applied.

1. *Figure Descriptions.*

Every figure should have a figure description unless it is purely decorative. These descriptions convey what’s in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than xxx characters long.  Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper. For figures that convey important and complex new information, a short plain text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph.  For additional information regarding how best to write figure descriptions and why doing this is so important, please see https://www.acm.org/accessibility.

The instructions below describe the required steps authors need to follow in order to insert descriptive text for figures (alt-txt value) in **MS Word 2019 on Windows or Word 2016 and later on Mac**:

1. Insert a picture in the document.
2. Right-click the image and select “Edit Alt Text”.
3. In the “alt text” section, provide your text description of the image.

Below are the steps to insert figure descriptions in **MS Word 2013 and 2016**:

1. Insert a picture in the document.
2. Right click on the inserted picture and select the **Format Picture** option.
3. In the settings at the right side of the window, click on the “Layout & Properties” icon (3rd option).
4. Expand **Alt Txt** option.
5. In the “Title” and “Description” text boxes, type the text you want to represent the figure, and then click “Close.”

Below are steps to insert the alt-txt value in **MS Word 2010/2011 for Windows\***:

1. Insert a picture in the document.
2. Right click on the inserted picture and select the **Format Picture** option.
3. Select the **Alt Txt** option from the left-side panel options.
4. In the “Title” and “Description” text boxes, type the text you want to represent the picture, and then click “Close.”  
   \* The Mac 2011 version 14.0.0 and later allows the option for inserting “alt-text.” In the MAC version of Word 2016, right-click on the image and select “Edit Alt Text” from the pop-up menu and then enter the description for the alt text.
5. **Quotations and Extracts**

There are styles for block quotations, which should be used for quotes that are separated from in-line text.  Below is an example.

“Microsoft tried to revive the idea of an assistant with Clippy, who began popping up in Microsoft Office in 1997.  Its creator, Kevan Atteberry, was actually contracted by Microsoft to design Clippy, which, funnily enough, he did on a Mac …  Sure, people could disable Clippy, but the fact he was on by default angered people.” [10]

1. **Equations**

There are two types of math equations: the *numbered display math equation* and the *un-numbered display math equation*. Below are examples of both.

1. ***DisplayFormula.***

*The* ***DisplayFormula*** *style is applied in the numbered math equation. A numbered display equation always has an equation number (label) on the right.*

x=-bb2-4ac2a (1)

1. *DisplayFormula.Unnum****.***

*The* ***DisplayFormulaUnnum*** *style is applied only in unnumbered equations. An unnumbered display equation never contains an equation number Bertot and Grimes (2012) on the right—this element distinguishes it from the numbered equation.*

x=-bb2-4ac2a

*Please note: the subsequent text after the* ***DisplayFormula*** *(numbered equation) or* ***DisplayFormulaUnnum*** *(unnumbered equation) must have the paragraph style* ***ParaContinue*** *applied.*

1. **Math statements**

Math statements should have the “Statement” style applied.

***Theorem/Proof/Lemma.*** *Math statements should have the “****Statement****” style applied. This paragraph is an example of the “****Statement****” style.*

1. **Algorithms**

Algorithms use the styles “AlgorithmCaption” and “Algorithm”.

ALGORITHM 1: Iterative Algorithm

current\_position  center

current\_direction up

current\_position is inside circle

while current\_position is inside circle, do

neighborhood all grid hexes within two hexes from current\_position

for each hex in neighborhood, do

for each neuron in hex do

      convert neuron\_orientation to vector

      scale vector by neuron\_excitation

            vector\_sum vector\_sum + vector

             end

      end

     normalize vector\_sum

end

1. **COMPUTER CODE**

Display Computer codes can be inserted using “ComputerCode” style.

CHAT Start

SAY Welcome to my world

WAIT 1.2

SAY Thanks for Visiting

ASK Do you want to play a game?

OPT Sure

OPT No Thanks

Similary, this is an example of intext code text.

Similary, this is an example of intext code text.

1. **Citing Related Work**

This section cites a variety of journal [5, 15], conference [1, 6, 8, 12, 13], and magazine [3] articles to illustrate how they appear in the references section. It also cites books [9, 10], a technical report [7], a PhD dissertation [4], an online reference [14], a software artifact [11], and a dataset [2].

As you build your article, you should note where you will be placing citations. If you are using numbered citations and references, the reference number - "...as shown in [5] ..." is sufficient. If you are using the "author year" style, a reasonable placeholder is the primary author's last name and the year of publication - "...as shown in [Harel 1978] ..." - we will be updating this placeholder later in the process with the citation label as generated by the Word macros in the "master template.

**ACKNOWLEDGMENTS**

Acknowledgments are placed before the references. Add information about grants, awards, or other types of funding that you have received to support your research. Author can capture the **grant sponsor information**, by selecting the grant sponsor text and apply style ‘GrantSponsor’. After this, select grant no and apply ‘GrantNumber’ from style panel. Example of Grant sponsor: Competitive Research Programme and example of Grant no: CRP 10-2012-03.

1. **HISTORY DATES**

In case of submissions being prepared for Journals or PACMs, please add history dates after References as (*please note revised date is optional*):

Received November 2019; revised August 2020; accepted December 2020

**REFERENCES**

[1] Zhao Yilin. Vehicle Location and Navigation System.Beijing: Publishing House of Electronics Industry, 1999.

[2] Lu Feng. Shortest path algorithm: Taxonomy and Advance in Research. Acta Geodaetica et Car tographica Sinica, 2001, 30(3): 269~275.

[3] Fu Mengyin, Li Ji, Deng Zhihong. A Hierarchical Route Planning Algorithm with Restricted Search Area. Journal of Computer-Aided Design & Computer Graphics, 2005, 17(8): 1773~1777.

[4] Su Haibin, Zhang Jitao. A New Algorithm of Hierarchical Route Planning with Restricted Search Area. Journal of Henan University (Natural Science). 2008, 8(1):81~84.

[5] Yan Hanbing, Liu Yingchun. A New Algorithm for Finding Shortcut in a City s Road Net Based on GIS Technology. Chinses Journal of Computers, 2000, 23(2):210~215.

[6] Li Qingquan, Zheng Nianbo, Xu Jinghai. A Hierarchical Route Planning Algorithm Based on Multi-level Topological Structure of Road Network. Journal of Image and Graphics, 2007, 12 (7): 1280~1285.

[7] Wang Yawen, Wang Xili, Cao Han. Multi-scale optimal route planning algorithm within restricted searching area. Application Research of Computers, 2007, 24 (12):66~68.

[8] Yan Weimin, Wu Weimin. Data Structure.Beijing: Tsinghua University Press,2002.