المجلّة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

عدد خاص
بالمؤتمر الليبي الدولي للعلوم التطبيقية
و الهندسية
27-28- سبتمبر 2022

LICASE
المؤتمر الليبي الدولي للعلوم التطبيقية و الهندسية

Shortest Route on the Map Using estimating costs for travel trips (A*), Dijkstra, and Floyd algorithm)

Mohamed Seidi Ahmed Hmadi

Computer Science Department
Faculty of Science Azzaytuna University,Tarhuna -Libya
Hmadi2595@yahoo.com

Abdulqader Ab Sinusi

Computer Science Department
Higher Institute of Science and Technology , Soq AlkamisMsehel -Libya,
Saih80@yahoo.com

Mohamed Maatouk Koum

Mobile Computing Department
Faculty of Information Technology Tripoli University-Libya
M.koum@uot.edu.ly

الملخـــــص

الهدف الرئيسي من هذا النظام هو التحديث المستمر للعثور على أقصر طريق على الخريطة من حيث التكلفة ، حيث تم بناء النظام لحساب المسافة التقريبية بين نقطتين أو أكثر وتم تحسينها بشكل أساسي بتطبيق ثلاث خوارزميات وهي خوارزمية (A *) و خوارزمية Dijkstra وخوارزمية Floyd ، وهى مختلفة ودمجها لعمليات الاختيار الافضل .تم تطوير النظام باعتماده على احتياجات المستخدمين ،وله درجة اعتمادية وموثوقية لمساعدة المستخدمين للحصول على رؤية واضحة لمسار السفر على طوال الرحلة من حيث المسار الكامل والأقصر للطريق مع نقاط العبور لتلبية رغبة المستخدمين ، وأيضًا حساب تكلفة السفر حسب المسافة المقطوعة والمصاريف و أفضل الخيارات على المسار. حيث يقوم المستخدم بالتسجيل في النظام ومن ثم يتم الدخول إلى النظام بعد عملية التحقق ، ويقوم باختيارت المناسبة عللا الخريطة لتطبيق النظام هذه الخيارات على الخوارزميات للحصول على المسار الملائم .

Abstract

The main objective of this system is to promote the assistant of finding the shortest route on the map and travel costs. This system was built to calculate an approximate distance between two or more points. Mainly It enhanced by three algorithms, namely, the (A*) algorithm, the Dijkstra algorithm, and the Floyd algorithm, these algorithms are different and we developed them by adoption according to the users' needs. The reliability of the system is to help users to have a clear view of the travel path along the journey. The assistant system can find the shortest route on the map, the full and shortest path of the road with crossing points to meet the users desire, and also calculate the cost of travel according to the distance traveled and expenses.

1. Introduction

To find shortest route on the map is a research project developed to help people to calculate travel costs allocations and know the best choices in the path, while target user by giving the correct permissions to this user after entering the system with the verification process.

Keywords: Shortest route, Costs Allocations, Dynamic network, Route system

2. Problem statement

One of the most important problems in the systems of finding the shortest route on the map is knowing the changes in the roads in the map, there are many constructions on the roads, which may occur almost every day in different places, and give the length of the road that is under construction or maintenance. Another case is communications within the system, our system will provide registration for the flights stipulated in this system, and this rises side by side to the need for users to communicate with each application. Communication is a must for planning a trip consisting of small or large numbers of travelers.

3. Objectives

The system was built to order to provide a service to the users efficiently with different types of advantages stipulated in this system, the main objectives of this work is to find and display the route between two or more cities with travel costs and another goal is to provide the trip system to the users as it has been mentioned as follows: 1) Find the shortest path and travel cost between selected cities with different algorithms and show the differences between the algorithms used.

2)Combine Dijkstra's algorithm with A* algorithms to find the exact path with crossing points between departure and arrival.

4.Research Implements

The system has an effective method in the speed of processing the results to find road paths and calculate costs, as the added advantages of this system are:

1) Provides detailed information on the route, Access to the system via the Internet (24) hours.

2) Ease of updating to modern and new information on the road throughout Libya, and between cities as well as countries.

3) Administrators are able to manage the users' account.

4) Supports three different algorithms used to find the path of the road.

5.The methods designed to find the shortest route on the map

Probably the most subjective part of a cost estimate occurs when the control system is to be installed on an existing facility [2]. A route finder is to get the itinerary from (A) to (H) and to consider all the various factors that may affect the journey, and from knowing all the shortcomings, accordingly, a good route plan consists of breaking down the journey into phases, and calculating the duration of each The stage necessary for coverage, all these elements are collected to obtain the total travel time, planning the routes also to calculate the most cost-effective route, including choosing two nodes, the best path to reduce the traveled distance and time.

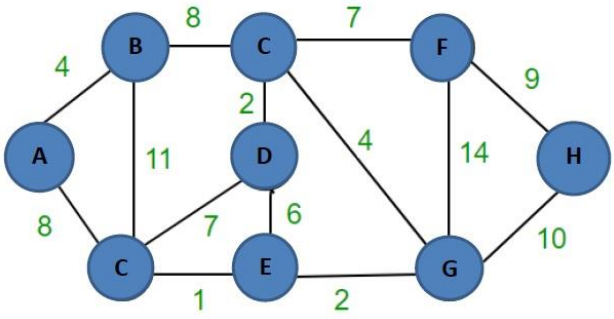

Figure (1) finding the shortest path on the planned route(map)

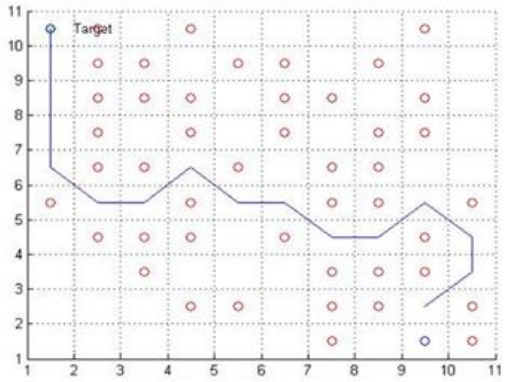

Figure (2) Selecting Initial position for the shortest path on the planned route(map)

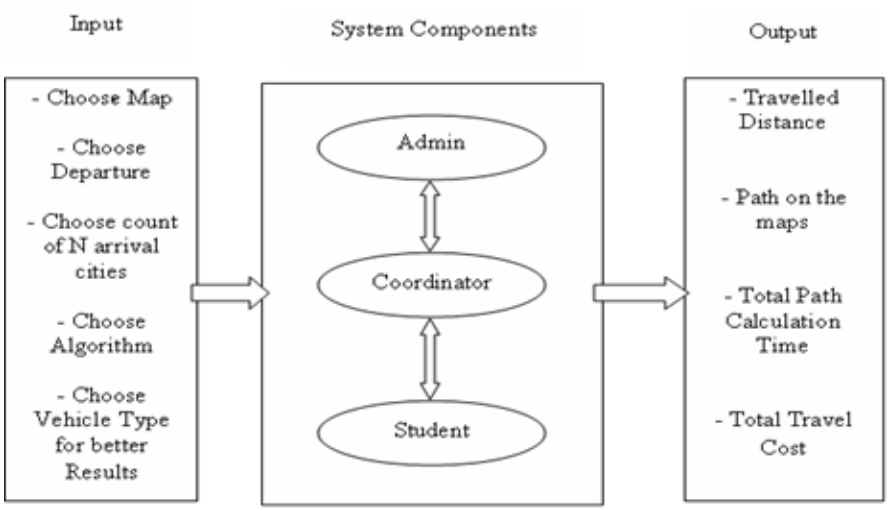

Figure (3) The input and output requirements of an assistant system

As shown in Figures (1,2,3) for the system to function correctly, there are some points that the user should specify (selected). The main point is to specify the main city from which it originated and the city of arrival from where the programs connect to the server and download the required information. Here, users need to choose a type The car before choosing the algorithm and before the start of operations. The output is represented by four main points, the distance of the main path, with the distance of the road as well and the calculation of the cost. All this will be done after calculating the time. innovative information and communication technologies (ICT) offer new potential for effective and efficient design of the calculation process. The availability of data along the entire product lifecycle, especially during the design phase **[1].**

5.1 Algorithm (A*)

The following Pseudocode describes the algorithm that has been included to give a clear understanding of how the algorithm works

function A(start,goal)*

closedset = the empty set // The set of nodes is already evaluated.

opensset = set containing the initial node // The set of tentative nodes to be evaluated.

g_score[start] = 0 // Distance from start along optimal path.

h_score[start] = heuristic_estimate_of_distance(start, goal)

f_score[start] = h_score[start] // Estimated total distance from start to goal through y.

while openset is not empty

x = the node in openset having the lowest f_score[] value

if x = goal

return reconstruct_path(came_from[goal])

remove x from openset

add x to closedset

foreach y in neighbor_nodes(x)

if y in closedset

continue

tentative_g_score = g_score[x] + dist_between(x,y)

if y not in openset

add y to o

penset

tentative_is_better = true

elseif tentative_g_score < g_score[y]

tentative_is_better = true

else

tentative_is_better = false

if tentative_is_better = true

came_from[y] = x

g_score[y] = tentative_g_score

h_score[y] = heuristic_estimate_of_distance(y, goal)

f_score[y] = g_score[y] + h_score[y] return failure

function reconstruct_path(current_node)

if came_from[current_node] is set

p = reconstruct_path(came_from[current_node])

return (p + current_node) else

return current_node

5.2 Dijkstra algorithms

In the Pseudocode describes the algorithm, the code: u := vertex in Q with smallest dist[], searches for the vertex of u in the set of vertex Q which contains least dist_between(u, v) and calculates the length of the neighbor nodes between u and v is the path length from root node to neighbor node v that passes through u . If this path is the current shortest path registered to v, this current path is replaced with the alternate path.

function Dijkstra(Graph, source):

for each vertex v in Graph: // Initializations

dist[v] = infinity // Unknown distance function from source to v

previous[v] = undefined // Previous node in optimal path from source

dist[source] = 0 // Distance from source to source

Q = the set of all nodes in Graph

// All nodes in the graph are unoptimized - thus are in Q

while Q is not empty: // The main loop

u = vertex in Q with smallest dist[]

if dist[u] = infinity:

break // all remaining vertices are inaccessible from source

remove u from Q

for each neighbor v of u: // where v has not yet been removed from Q.

alt = dist[u] + dist_between(u, v)

if alt < dist[v]: // Relax (u,v,a)

dist[v] = alt

previous[v] = u

return dist[]

If we are only interested in the shortest path between the source of the vertices and the target, then we can end the search in line 11 if u = target, now we can read the shortest path from the source to the target by iteration:

1) S = Empty sequence *2) u = Target* *3) While previous[u] is defined*

4) Insert u at the beginning of S *5) u = Previous[u]*

Now the S sequence is in the list of vertices which constitutes one of the shortest paths from source to target or an empty sequence if no path exists, and a more general problem would be to find all the shortest paths between source and target (there may be many different ones for the same Length) . A more general problem is that you find all the shortest paths between the source and the target (there may be many different types of the same length), so instead of storing only one node at a time. We store all the condition nodes to achieve a time shortening, for example, if Both r and the source connect to the target are on different paths of the shortest path to the target (because the edge cost is the same in both cases, then we add r , and the source [for the previous target]).

5.3 Floyd's algorithm

The edgeCost(i,j) task which returns the edge cost from i to j also assumes that n is the number of vertices and edge(ii) = 0,finally a 2-dimensional array, at each step in In the algorithm, the path [i] [j] is the shortest path from i to j using the mean vertices (1..k-1). Each path [i] [j] is initialized to edge (i, j) or infinity if there is no edge between i and j .

```
int path[][];
procedure FloydWarshall ( )
for k  = 1 to n
for i  = 1 to n
for j  = 1 to n
path[i][j] = min ( path[i][j], path[i][k]+path[k][j] );
```

5.4 Speed Comparative of Dijkstra’s, Floyd’s, and (A*) algorithm

In the figure (4) The algorithms used in the system were evaluated for the purpose of measuring the speed. It was found from the mark level of Floyd's algorithm that it is the fastest algorithm compared to others, while Dijkstra's algorithm is actually supposed to be much faster than A*, but in this work, we made a modified which combined both the A* algorithm, that is between the departure point and the arrival point, where crossing points are calculated by Dijkstra's algorithm, and finding the exact path between each point is calculated by A* algorithm.

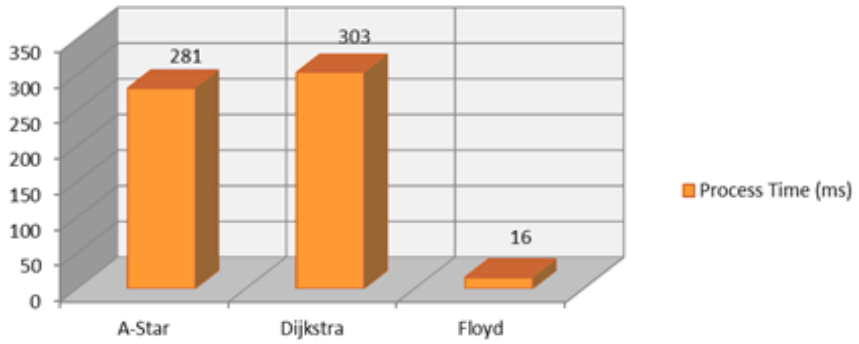

Figure (4) performance in calculation of time in Dijkstra's , Floyd's , and (A*) algorithms

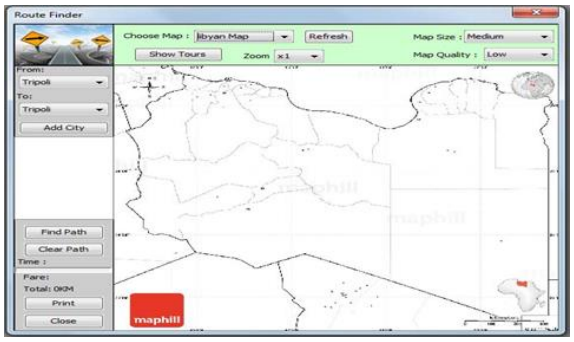

Figure (5) The main screen for calculating travel costs

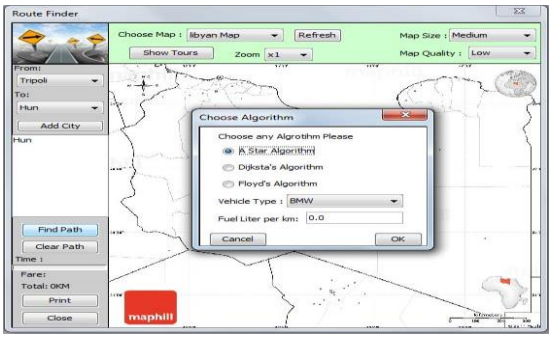

Figure (6) finding the path and choosing the type of vehicle

6.Route finder screen

Show trips in bellow Figure (5) provides full information for all trips prepared by the system administrators and the possibility of communicating with them easily to explain the details of the trip and the route.provide full information for all trips prepared by the system administrators, and the possibility of communicating with them easily to explain the details of the trip and the route. the main screen for calculating the journey and travel costs This is the most important window, this is where you get all the route results and costs and it gives a result of that at the top, there is a map selection box, all the maps are called from the database, and maps can be added and deleted by the administrator, and at the top of the window too Map size and quality allow you to increase the dimensions of the map and the smoothness of the image with low-quality image technology to make it readable. In the above figure (6) determining path and choosing the type of vehicle, where the result and the cost of travel will be based on the users' choice of the algorithm and the type of vehicle. On the left side, you can see the city selection boxes with the “Add City” button, the cities will be added to the list, and when you click on

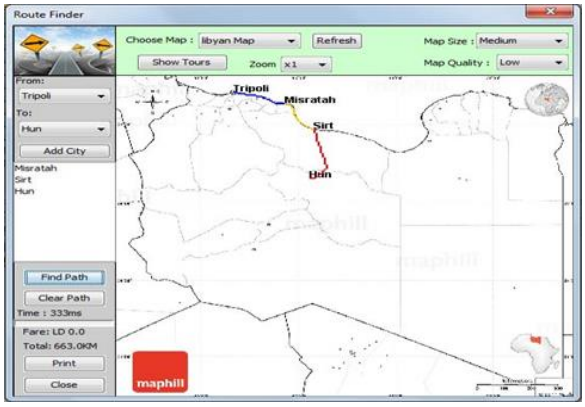

Figure (7) The output of calculating the time and path using the algorithm (A*)

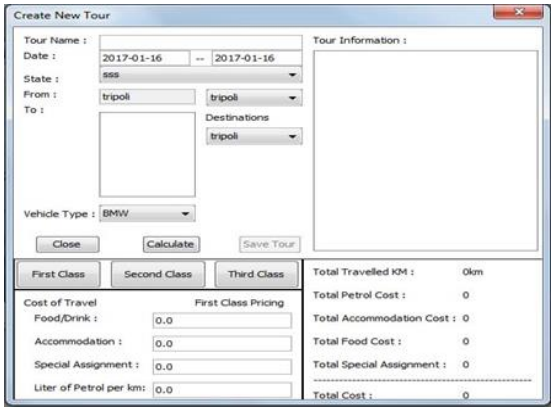

Figure (8) Creation of tour trips

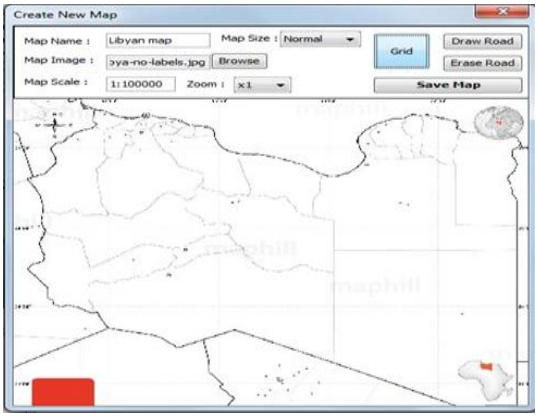

Figure (9) Add a map with route identification feature

"Find Route" you will have such a screen and this system provides three different algorithms that you will see in the following screenshots, and there List the vehicle that can be added or deleted by the administrator.

7.Experimental and results

We got that A* algorithm is finding the real path on the ground, and this is the difference between it and other algorithms used in this system. In fact, in this system, it has been combined with the A* algorithm to find the exact path with the crossing points, and that's why Dijkstra's algorithm takes more time to work as shown in the following:

1) (A*) is faster than Dijkstra, the only difference is that (A*) tries to come up with the best path using inference function and this gives priority to nodes that are supposed to be better than others while Dijkstra explores all possible paths.

2) (A*) It's supposed to be faster than Dijkstra even if it requires more memory and processes in the node because it detects very few nodes good gains anyway.

3) We can say in short (A*) that inference is used to improve the search. This means that the inference function is able to determine the cost to some extent from the node to the target.

4) (A*) In the normal case it finds the path if it exists and in the ideal case there is the shortest path.

5) Dijkstra is the opposite of (A*), ie it gives you all the paths, Dijkstra discovers all possible paths, for example (there are a huge number of paths, then the graph simply explores the least important paths). Our suggested designed system includes calculating the cost of travel and finding the route in one system, so if the user desires both, he needs to find different systems, for this reason the calculation of the cost of travel and finding the route has been combined in our system. In this system, you need to choose your route, it calculates the length of the route drawn on the map, and then shows you the cost of fuel consumption that will be needed during the trip. In the beginning, usability is meant as the ability of devices or systems to be used easily in order to achieve a specific goal, as various studies indicate over the years all over the world to evaluate the usability and application of the research path, and the goal of usability in all research and studies aims to meet the results or The objectives of using the system are at stake for the intended user. Therefore, the focus in this research is to use usability to include all usage evaluations of the system. We complete the tasks within a limited time to see the efficiency, usability and ease of the system and the following are the questions we addressed to the participants as shown in table (1).

Table. (1)

No.	Estimated Time (mm:ss)	Tasks Description
1	01:30	Register with the System(at the first time)
2	00:45	Login in the system using the name you registered with.
3	1:00	Change your password.
4	2:30	Search for route between cities in any map.
5	1:30	Printout the report regarding methods.
6	2:45	Register for any tour you wish to join.
7	3:00	Send an email to the Program Administrator regarding the tour you have just registered.
8	00:45	Log out of the system securely.

8. Conclusions

Through the results of the questionnaire that we distributed, our system provides many features and gives more flexibility to users, and the study also showed that the results of the system outputs are more reliable than some other systems. Daily itineraries, which allows each person on the tour to easily view the daily itineraries (which works both on or offline), set lists for each venue are easy to access and view. Signalization and schedule systems.

9.Future Work

This system has been tested to ensure that all the input and output units are working together properly while the system test was conducted successfully with feedback received. It is good

according to the usability test. The usability test was done at this stage of testing the system and evaluating the participants through the questionnaire. The results of the analyzes that were conducted to develop the system.

The suggested area to be searches next regarding this system as follows

1) The assistant of finding the shortest route on the map must be tested on the web, and it is desirable to use it commercially on the web server.
2) The system must be upgraded in the future according to the needs of the users, and therefore the system will suit the level of everyone's satisfaction.
3) Advertising on the system using the trial version on a specific site and promoting the process of buying

References

[1] "Design model for the cost calculation of product-service systems in single and small series production", Günther Schuh, Christoph Kelzenberg And Jan Wiese. January 2019. Procedia CIRP 84:296-301 , DOI:10.1016/j.procir.2019.04.216.
[2] "Cost Estimation Concepts and Methodology", John L. Sorrels , and Thomas G. Walton, November 2017.
[3] http://www.willyork.com/route-scheduling-software/use-transportation- and-logistics-software-to-improve-efficiency/. "Use Transportation And Logistics Software To Improve Efficiency". October 15th, 2010 .
[4] Bennett, S., McRobb, S., & Farmer, R. (2006). "Object-oriented systems analysis and design using UML", London: McGraw-Hill, Volume 3, ISBN: 978-0-07711-000-0, pp. 371-375.
[5] Peters K. (2008). Advanced ActionScript 3.0 Animation, Friends of ED, ISBN: 978-1-43021-608-7, pp. 153-155.
[6] Puthuparampil, M. (2007), Algorithms - Dijkstra's Algorithm, pp. 3-5.
[7] Reilly, R. T. (2004). Handbook of Professional Tour Management. Volume 2, ISBN: 08-273-3525-3, pp, 158.
[8] Jacobson, I., Christerson, M., Johnsson, P., & Overgaard, G. (2004). Object-oriented Software Engineering: A use case driven approach. Harlow, England: Addison-Wesely, The University of Michigan. ISBN: 978-0-20154-435-0, p. 307-321, 456.
[9] Flinsenberg, I. C. M. (2004): Route planning algorithms for car navigation. PhD Thesis, Technische Universiteit Eindhoven. ISBN 903-8-60902-7, p. 1, 2-8
[10] Leffingwell, D. & Widrig, D. (2003). Managing software requirements: "A use case approach", Published by Addison-Wesley, ISBN: 978-0-32112-247-6, pp. 15-17.
[11] Russell, S. J. & Norvig, P. (2003). Artificial Intelligence: A Modern Approach. Upper Saddle River, N.J.: Prentice Hall. ISBN 0-13-790395-2, p. 97–104.
[12] Satoh, K. (2003). Proceedings of the Eastern Asia Society for Transportation Studies, Volume 5, pp. 1249 - 1264.
[13] Kessner, M. (December ,2000). On the reliability of usability testing, Carleton University Masters Thesis, Ottawa, Ontario
[14] Fielder, A. L. (1991) "Tour Management", The Gordaka Co. UK., pp. 72-76.

[15] Dechter, R. & Pearl, J. (1985). "Generalized best-first search strategies and the optimality of A*". Journal of the ACM 32 (3). doi:10.1145/3828.3830, p. 505-536.
[16] Klee, V. (1973). A Comment on Floyd's Algorithm for Shortest Chains, Washington University Seattle Dept. of Mathematics. Defense Technical Information Center, p. 1-9
[17] Hart, P. E., Nilsson, N. J. & Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2). doi:10.1109/TSSC.1968.300136, p. 100-107.
[18] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs." Numerische Mathematik 1, pp. 269-271.