Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

Railway Travel-Time Optimization System

Nzechukwu C. Otuneme[1], Monday O. Eze[2], Olubukola D. Adekola[3],
Adewale O. Adebayo[4], Samson O. Ogunlere[5]
[1,2,3,4,5]Computer Science, Babcock University,
Ilisan Remo, Ogun State,
Nigeria

***Abstract*—Effective transportation systems are an integral part of people's life and this means a great deal of economic advantages. The increase in the need for the movement of passengers and (especially) heavy goods from one location to the other introduced an alternative means of transportation to road - the railway system. The existing travel path implementation considers the shortest routes in terms of distance, which does not necessarily mean the shortest travel time due to some other factors not considered in classical Dijkstra Algorithm for transportation network. This research, therefore, focused on enhancing and adapting Dijkstra algorithm in order to incorporate factors (in addition to distance) such as number of stoppages and type of rail track, which directly impact or affect travel time. A travel time algorithm and railway scheduling system was implemented. This is to support passengers travel decision and to monitor and control the usage of the railway. The study concluded that for optimization of railway routes, the travel time should be considered and not just the distance between locations and that the railway system be effectively integrated with other transport systems in order to achieve seamless transport system at large.**

Keywords—Dijkstra algorithm; Optimization; shortest path; Stoppages; Travel-time

I. INTRODUCTION

The optimization of the railway system is essential to reduce loss of time and money [1], to improve on the economy of a nation and also to enhance the quality of life for its citizenry. The quest for safe movement of passengers and goods from one location to the other brought about the development of an alternative means of transportation to road; which is the railway. Rail transport is the use of wheeled vehicles running on rails (track) to convey humans (passengers) and goods from one location to the other. Rail tracks are made of steels installed on ties on which the rolling materials are filled as they move. Rail transportation has less frictional movement than the car on road and can also carry more load and passengers.

The invention of rail dates back to the 6th century BC in the Greek empire when horses and carriages were used for transportation on tracks [2]. The rail transportation became more popular in the 19th century in the British Empire by the development of steam engine as a source of locomotive power and later the diesel engines and electrically powered engines were developed [2]. Nigerian railway system was first developed by the British government from Lagos Colony to Ibadan in March 1896, the rail started operation in the year 1901 [3]. More development came in 1911 by the extension of the railway to Minna where it met the Northern railway developed by the Northern government [4]. Railway system in Nigeria still operates at a skeletal level. At the moment, it has not optimized its routes to reduce travel time and organize schedules for trains. Railway services in Nigeria are controlled by the Nigerian Railway Corporation. The railway system in Nigeria is made up of 3,505 kilometers of 3 feet 6 inches (1,067 mm) gauge lines and 479 kilometers of standard gauge lines. The Nigerian railway system reached its maximum patronage in the carriage of passengers and load in the year 1984 with approximately 15,500,000 passengers and 1,098,000 tonnes of Freight traffic in 1974 [5]. The railway system in Nigeria recorded its minimum passenger patronage in the year 2003 with less than 1,600,000 passengers and its minimum Freight traffic of less than 100,000 tonnes of Freight traffic in year 2000 [5]. The drastic decline experienced in Nigerian railway system was due to neglect and mismanagement. In the year 2013 there was only one working rail route from Lagos to Kano [6], a journey of about 31 hours with the speed of 45 km/h.

Restoration and rehabilitation came to the railway system in Nigeria from the year 2009 with the Nigerian government awarding a contract of the eastern line from Port Harcourt to Maiduguri at the cost of US$427 million to Lingo Nigeria, Eser West Africa, and the China Gezhouba Group. Subsequently, so many other projects aimed at restoring the railway system in Nigeria. The construction of new rail tracks led to a new challenge of optimizing the system in order for efficiency and reduce the travel time. To determine the shortest route between two points the travel distance between all possible routes to the destination from the source is measured, and the shortest route is determined after the summation of all possible paths to the destination from the source. In this study, to determine the shortest route between two locations (stations) Dijkstra's Algorithm is used; which also optimizes the routes. Traditionally, Dijkstra's Algorithm is used for deriving the shortest path to all nodes from a single source in a network and it is correct only when none of the weights is negative [7]. In practical application, the shortest path does not always result to the shortest travel time due to some other factors not considered in classical Dijkstra's Algorithm for transportation network [8]. The number of stop overs, number of interchanges, train contents (passengers only, freights only or mixed), traffic, topography and type of rail track can affect the travel time on railway system [9] and if not put into consideration it could amount to the shortest path not being the shortest travel time [10]. In reality, for most trips, travel time is the highly esteemed need of railroad users and not necessarily the shortest pathway. Thus, the need for this current research.

II. OBJECTIVE OF THE STUDY

The general objective of this study is to develop a framework to optimize travel time for railroad transport using Nigerian railway system as a case study. The specific objectives are to:

i. design an enhanced railway optimization algorithm to find shortest path incorporating travel time factors

ii. implement the developed algorithm in (i) and

iii. implement a railway scheduling system to monitor and control the usage of the railway.

III. METHODOLOGY

This study adopted Dijkstra algorithm to determine the shortest path between two distances. It further extended the algorithm to factor in number of stoppages and type of rail constraints in order to exhibit a more crucial requirement as travel time. The implement of the designed travel time algorithm was carried out using software development tools namely: Java programming language, XML, Java Script, MySQL database and SQL. Scope of the Study

IV. SCOPE OF THE STUDY

This research focused on optimizing railway travel time using the Nigerian railway system as a typical case study. Rail transport is relatively slower than road, therefore, this is a user-centered approach to solving the problem of traveling by railway medium.

V. SIGNIFICANCE OF THE STUDY

Optimization is important in the operation of railway as passengers could spend almost twice the expected time due to un-optimized system because shortest path does not necessary connote shortest travel time when crucial factors such as stoppages and type of rail track are considered. This is rather a user-centered concept- which should be the hallmark of any user-intensive system.

VI. OVERVIEW OF SEARCH ALGORITHM

A search algorithm is the step-by-step procedure used to locate specific data among a collection of data. It is considered a fundamental procedure in computing. In Computer Science, when searching for data, the difference between a fast application and a slower one often lies in the use of the proper search algorithm. In search algorithms there is always a tradeoff between space, preprocessing time and query time [11].

A. Shortest Path Algorithm

Let G = (V, E) be a weighted digraph, with weight function w: E → R mapping edges to real-valued weights. If e = (u, v), we write w (u, v) for w (e).
The length of a path p = (v_0, v_1, … v_k) is the sum of the weights of its constituent edges:

$$Length(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i) \qquad (1)$$

The distance from u to v, denoted ∂ (u, v), is the length of the minimum length path if there is a path from u to v; and is ∞ otherwise.
Figure 1 below shows that length(< a, b, c, e >) = 6
i.e. the distance from a to e is 6.

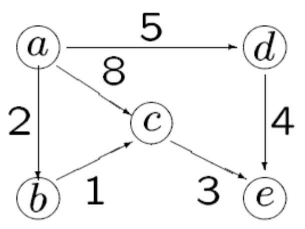

Figure 1: Shortest Path Algorithm [12]

Any sub-path of a shortest path must also be a shortest path, the existence of a shortest path tree in which distance from source to vertex v is length of shortest path from source to vertex in original tree.

B. Optimization of Algorithm

Optimization is the selection of a best element (with regard to some criteria) from some set of available alternatives [13]. It includes maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. Optimization is important in the operation of railway as passengers could spend almost twice the expected time due to un-optimized system [14].

C. Overview of Dijkstra's Algorithm

Dijkstra Algorithm also known as greedy search algorithm was developed by Edsger W. Dijkstra [15]. It is designed to find solution to two fundamental graph theoretic problems: the minimum weight spanning tree problem and the shortest path problem. Today, Dijkstra's Algorithm used in solving shortest path problem is one of the most efficient algorithms in computer science (CS) and a very popular algorithm in operations research (OR) [16]. Dijkstra's Algorithm is a non-polynomial algorithm with the worst case of mn. A non-polynomial algorithm is one which requires in the worst case a number of steps not less than some exponential expression [17] like Lnm, n!, 100n, where n and m refer to the row and column dimensions of the problem and L to the number of bits needed to store the input data.

1) Sequential Operation of Dijkstra Algorithm: Let the start node for the search be called the initial node. Let the distance of node Y be the distance from the initial node to Y. Dijkstra's Algorithm will assign some initial distance values and will try to improve them step by step.

a) Assign to every node a tentative distance value: set it to zero for the initial node and to infinity for all other nodes.

b) Set the initial node as current. Mark all other nodes unvisited. Create a set of all the unvisited nodes called the unvisited set.

c) For the current node, consider all of its unvisited neighbors and calculate their tentative distances. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B (through A) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.

d) Having considered all the neighbours of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.

e) If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.

f) Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step iii.

2) Description of Dijkstra: Suppose you would like to find the shortest path between two intersections on a city map: a starting point and a destination. Dijkstra's Algorithm initially marks the distance (from the starting point) to every other intersection on the map with infinity. This is done not to imply there is an infinite distance, but to note that those intersections have not yet been visited; some variants of this method simply leave the intersections' distances unlabeled. Now, at each iteration, select the current intersection. For the first iteration, the current intersection would be the starting point, and the distance to it (the intersection's label) would be zero. For subsequent iterations (after the first), the current intersection will be the closest unvisited intersection to the starting point (which is easy to find). From the current intersection, update the distance to every unvisited intersection that is directly connected to it. This is done by determining the sum of the distance between an unvisited intersection and the value of the current intersection, and relabeling the unvisited intersection with this value (the sum), if it is less than its current value. In effect, the intersection is relabeled if the path to it through the current intersection is shorter than the previously known paths.
To facilitate shortest path identification, in pencil, mark the road with an arrow pointing to the relabeled intersection (if labeled or relabeled), and erase all others pointing to it. After the distances to each neighboring intersection is updated, the current intersection is marked as visited, and the unvisited intersection with lowest distance is selected (from the starting point) – or the lowest label—as the current intersection. Nodes marked as visited are labeled with the shortest path from the starting point to it and will not be revisited. This process of updating the neighboring intersections with the shortest distances is continued, then the current intersection is marked as visited and the closest unvisited intersection is moved onto until the destination is marked as visited. Once the destination is marked as visited (as is the case with any visited intersection), the shortest path to it is determined, from the starting point, and can be traced way back, following the arrows in reverse; in the algorithm's implementations, this is usually done (after the algorithm has reached the destination node) by following the nodes' parents from the destination node up to the starting node; that's why track of each node's parent is also kept. This algorithm makes no attempt to direct "exploration" towards the destination as one might expect. Rather, the sole consideration in determining the next "current" intersection is its distance from the starting point. This algorithm therefore expands outward from the starting point, interactively considering every node that is closer in terms of shortest path distance until it reaches the destination. When understood in this way, it is clear how the algorithm necessarily finds the shortest path. However, it may also reveal one of the algorithm's weaknesses: its relative slowness in some topologies.
The estimate d[v] of the length ∂(s, v) of the shortest path for each vertex v.
d[v] ≥ ∂(s, v) equals to the length of a known path. (d[v]= ∞ if there is no path)
Initially d[s] = 0 and all the other d[v] values are set to ∞. The algorithm will then process the vertices sequentially. The processed vertex's estimate will be validated as being real shortest distance, i.e. d[v] = ∂(s, v). Here "processing a vertex u" means finding new paths and updating d[v] for all v € Adj[u], that is, adjacent to u, if necessary. The process by which an estimate is updated is called relaxation. When all vertices have been processed d[v] = ∂(s, v) for all v.

3) ***Dijkstra Algorithm's Proof of Correctness:*** Place Proof of correctness is done by the use of induction on the number of visited nodes.
Invariant hypothesis: For each visited node v, dist[v] is the shortest distance from source to v; and for each unvisited node u, dist[u] is the shortest distance via visited nodes only from source to u (if such a path exists, otherwise infinity; note that it should not be assumed that dist[u] is the actual shortest distance for unvisited nodes).
The base case is when there is just one visited node, namely the initial node source, and the hypothesis is trivial.

Assume the hypothesis for n-1 visited nodes. Now, choose an edge vu where u has the least dist[u] of any unvisited node and the edge vu is such that dist[u] = dist[v] + length[v,u]. dist[u] must be the shortest distance from source to u because if there were a shorter path, and if **w** was the first unvisited node on that path then by hypothesis dist [w] < dist [u] creating a contradiction. Similarly, if there was a shorter path to u without using unvisited nodes then dist [u] would have been less than dist [v] + length [v,u]. After processing u it will still be true that for each unvisited node w, dist[w] is the shortest distance from source to w using visited nodes only, since if there were a shorter path which doesn't visit u, it would have been found it previously, and if there is a shorter path using u, it would be updated when processing u.

VII. OVERVIEW OF SEARCH

The following is a review of closely related works that presented efforts in route optimization and associated algorithms:

A. Railway Route Optimization Using Dijkstra Method

The development of smart cities to improve the quality of life with a greener and more reliable model, providing users with more regular and reliable information of the railway system, provide information about all trains and in cases of no direct train provides information that will lead the user to his/her destination. The system optimizes the railway using Dijkstra's Algorithm, giving user a more optimized route from one point to another [18]. Single-source shortest-path problems were considered.

Algorithm Used: Dijkstra's Algorithm which is applicable to non-negative weight graphs that are both directed and undirected, finds the shortest route to the graph's vertices in the distance order from a given point starting from the shortest distance.
Limitation: The algorithm needs more optimization by including other factors that affect the travel time in order to produce a system that is efficient.

B. A Comparative Study of Algorithms for Shortest Route Problems and Other Extensions

Optimization of Algorithms using the shortest route model by implementation of Dijkstra and Floyd's algorithms with some extensions of Floyd's algorithm. A linear programing problem was formulated and a 0-1 integer linear programing problem was derived. In solving this the shortest path from one location to another was derived and the shortest possible route from the source to a sink node is also derived [19].

Algorithms Used:
i. Dijkstra Algorithm: for finding the shortest path that exists between two nodes and every other node.
ii. Floyd's Algorithm

The Floyd–Warshall algorithm is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles). A single execution of the algorithm will find the lengths (summed weights) of the shortest paths between all pairs of vertices. Although it does not return details of the paths themselves, it is possible to reconstruct the paths with simple modifications to the algorithm [20].

Floyd's Algorithm is used to determine the shortest route between all pair of nodes in the network, this makes the algorithm more general than Dijkstra Algorithm. It represents the network in a square metrics with n number of nodes, n number of rows and columns.

Limitation:
When the graph is undirected the number of decision variables becomes too large and the simplex method will take so many iterations to get the optimum path thereby making it cost ineffective.

C. A Computational Study of Routing Algorithms for Realistic Transportation System

This focuses on the analysis Shortest path using Transportation Analysis and Simulation System (TRANSIMS). It Studies the heuristic and accurate way associated with data structure effect on the computational performance of software developed for realistic transportation system. It shows that modified Dijkstra algorithm is efficient for various transportation planning applications [8]. Experiments are carried out in finding the shortest part between a pair of nodes as against finding the shortest path tress. Apart from the shortest path the travelers starting time affects the system. The study deals with up to 80,000 nodes and 120, 000 edges which results to a larger storage space.

Algorithm Evaluated:
i. Dijkstra algorithm with binary heap
ii. A* algorithm

A* (pronounced as "A star") is an algorithm that is used in finding paths and traversing of graphs, the process of plotting an efficiently directed path between multiple points, called nodes. It enjoys widespread use due to its performance and accuracy. However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance [21] although other work has found A* to be superior to other approaches [22].

Peter Hart, Nils Nilsson and Bertram Raphael of Stanford Research Institute (now SRI International) first described the algorithm in 1968 [23]. It is an extension of Edsger Dijkstra's 1959 algorithm. A* achieves better performance by using heuristics to guide its search.

Result Obtained:
Simple implementation of binary heap Dijkstra algorithm is a good choice for finding optimal route in a real road transportation network. The worst case complexity of one to one shortest path complexity is the same as one to all shortest path algorithms.

Limitation:
It is a simulation of a road network with no real life implementation and did not consider the factor of a rail transportation system.

D. Decision Support System for Real World High Speed Rail (HSR) Panning

The construction of railway system is faced with a lot of decisions to make concerning the infrastructure macro-location, the type of traffic supported by the rail, the speed limit, the elevation and geology, population density and climate. A research by [9] reported a number of difficult situation experienced in an effort to build an efficient

Portuguese railway system. Considering the cost of construction of a rail system, it is important to design an optimized, user-friendly system with an optimized geometric layout.

Problems Solved:

i. Natural barriers to infrastructure.
ii. Effect of layout safety requirement.
iii. Infrastructure cost

Based on the conceptual and operational frameworks developed, the capabilities of the approach are illustrated for the specific case of the Lisbon-Oporto HSR planning problem aimed at linking Lisbon, Coimbra, and Porto, Portugal, with a passenger-dedicated double-track HSR line.

Algorithm Used:

Simulated Annealing Algorithm (SAA) was used. The SAA traces its origins to the annealing process of materials to low-energy states and is credited to [24], who applied the Metropolis concepts to solve the traveling salesman problem. This research also gave an overview of the algorithm and comprehensive discussions. The algorithm starts with an initial system configuration, and neighboring configurations are tested and accepted as current configurations if they improve the value of the objective function. Worsening system configurations are also accepted as current configurations with a probability based on the Metropolis criteria, allowing the SAA to escape from local optima. The SAA is as a stochastic technique that applies a probabilistic mechanism for accepting worse solutions. However, even if the algorithm is disassociated from the physical meaning, the terminology borrowed from the annealing physical process is used. The probability of accepting worsening configurations decreases as the algorithm progresses (cooling) at descending values of a control parameter (temperature).

VIII. SYSTEM DESIGN AND FACTORS CONSIDERED

The purpose of the system design is to produce specification that would yield an accurate implementation of the system. The system design was decomposed into modules to provide a software structure that implements the functions defined in the analysis.

Factors considered during the design of this system are as follows:

i. Designing a system that would make it possible to obtain the shortest possible route from one location to the other in the Railway system under consideration
ii. Designing a system that has a capacity to stores rail track information.
iii. Improving work efficiency and quality assurance.
iv. Providing a reliable, accessible and efficient Railway optimization system.

A. *Case Study of Nigerian Railway Route with Extensions and Development*

Figure 2 is the diagram of the existing Nigerian railway route with proposed extensions and development which includes the narrow gauges, standard gauge and high spend gauge.

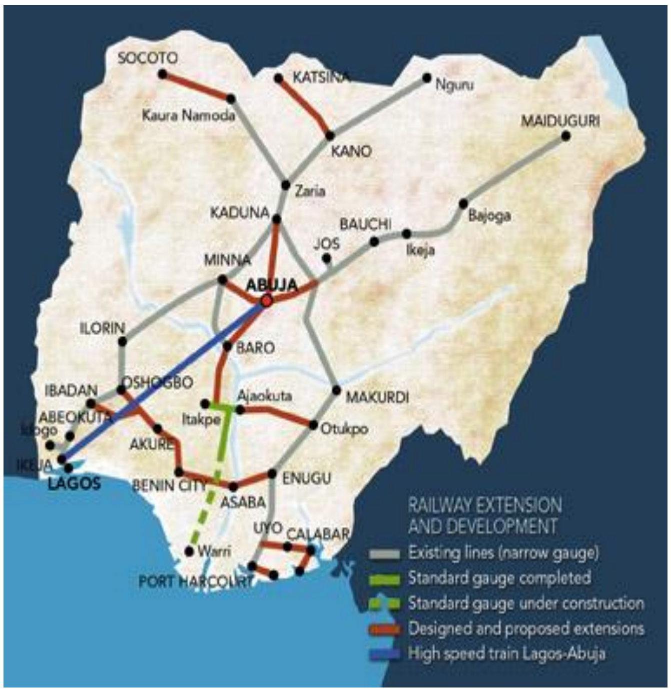

Figure 2: Nigerian Railway map with extension and development [25]

B. *Mathematical Modeling of the Adapted Algorithm*

Let G = (V, E) be a weighted digraph, with weight function w: E → R mapping edges to real-valued weights. If e = (u, v), we write w (u, v) for w (e).

t = expected time on a track
s = expected speed on track
d = delay estimate = summation of travel time factors
w = weight of a path
w =q
Travel time is the summation of time on each path + delay

$$t = s*l + d \qquad (2)$$

$$t = \sum (\text{speed}/ \text{distance})$$

$$q = \sum_{i>0}(t_i) + \sum \ (d_i) \qquad (3)$$

The time of a path p = (v0, v1, … vk) is the sum of the weights of its constituent edges, which is the expected time on the path:

$$Time(T) = \sum_{i=1}^{k} w(v_{i-1,} v_i) \qquad (4)$$

C. *Use Case Diagrams Describing Major Functions of User and the System*

Major use cases used here are to capture the business requirements for the system and to illustrate the interaction between the system and its environment. The major actors here include Manager, Operator and Passenger. Figures 3, 4, and 5 represent the use case diagrams for the proposed system.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 11, November-2018

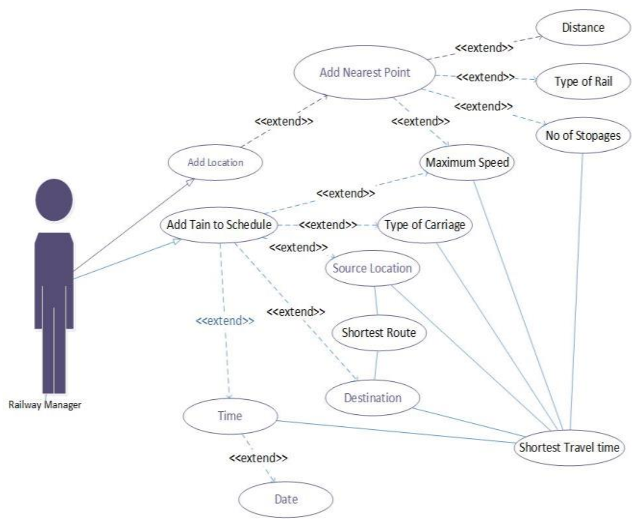

Figure 3: Railway Manager Use Case

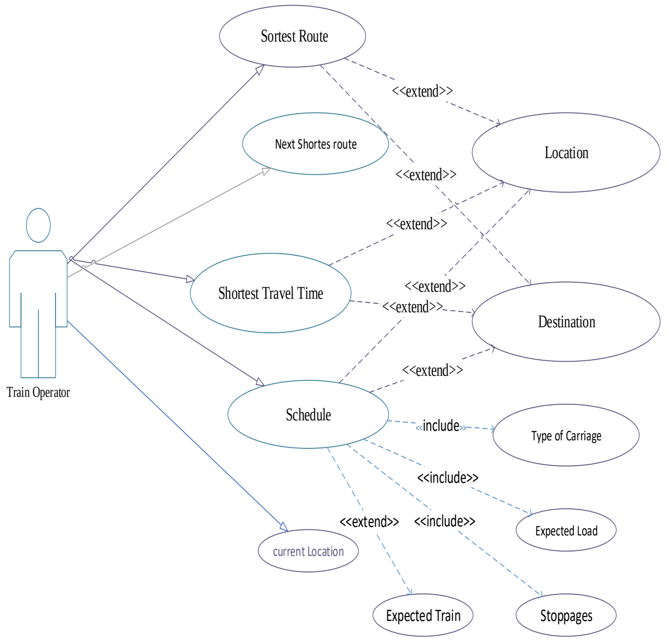

Figure 4: Train Operator Use Case

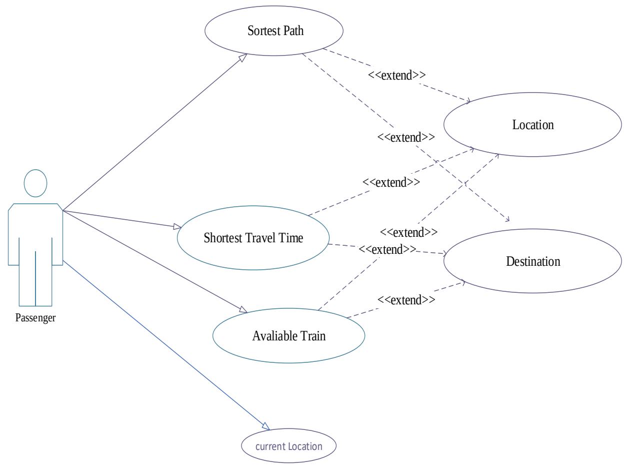

Figure 5: Passenger Use Case

D. *Travel Time Data Flow Diagram (DFD)*

This involves the illustration the flow of data within the system. Figure 6 shows the interactions that takes place between the specified functions such as data exchange, input and output information.

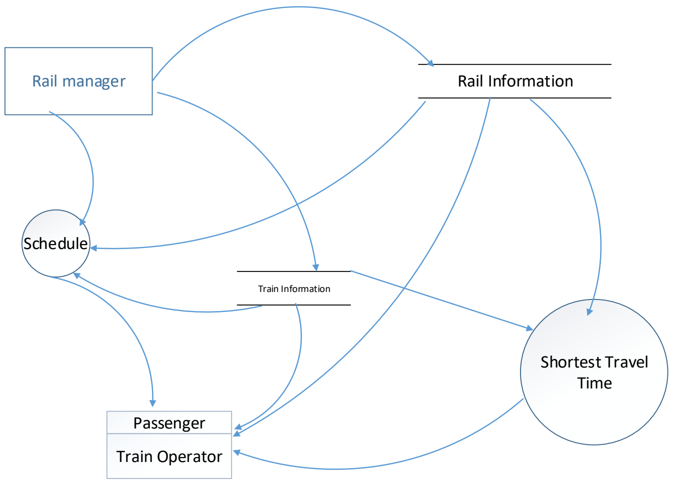

Figure 6: Data flow diagram for the travel time system

E. *Summary Flow Chart of the System Design*

Figure 7 is a chart representing the summary design of the railway travel time optimization system which extended Dijkstra's shortest path algorithm to consider, in addition the shortest distance between two locations, other factors that constitute waste of time on railway. By optimizing Dijkstra's Algorithm with new parameters such as the number of stop overs, number of interchanges, train content (passengers only, freights only, mixed) and type of rail track, the shortest travel time will be derived.

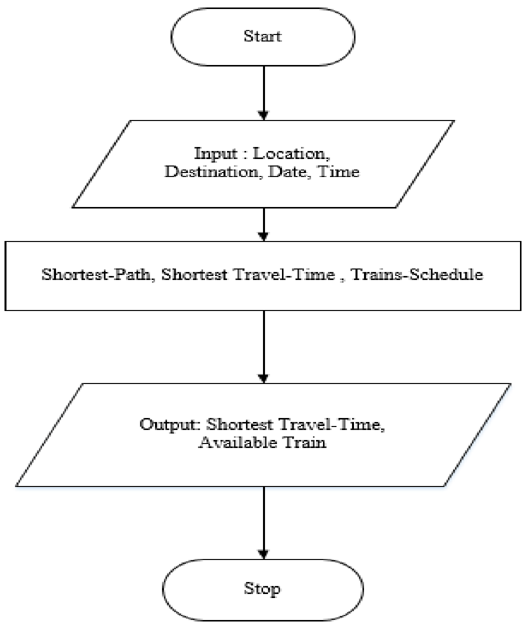
Start
Input : Location, Destination, Date, Time
Shortest-Path, Shortest Travel-Time , Trains-Schedule
Output: Shortest Travel-Time, Available Train
Stop

Figure 7: Summary design of the railway travel time optimization system

F. Implementation and Outcome

Figure 8 shows landing page of the developed system, this displays the map of the railway routes in Nigeria indicating different stations.

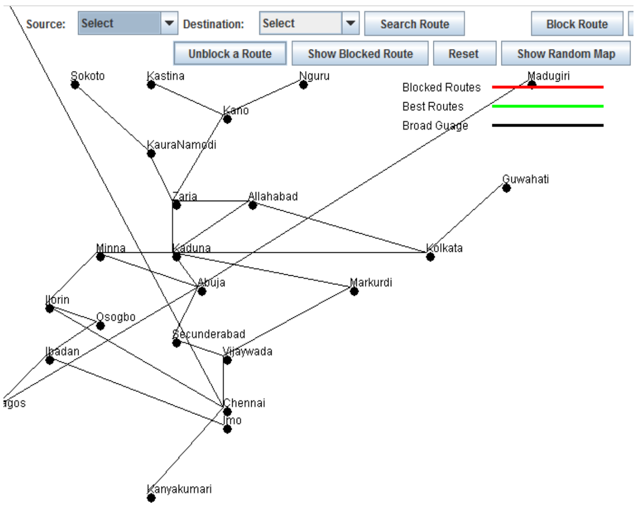

Figure 8: Implementation of the improved algorithm

Figure 9 shows the randomization of the graph but maintaining all the connections marked green is the selected route indicating the route shortest travel time.

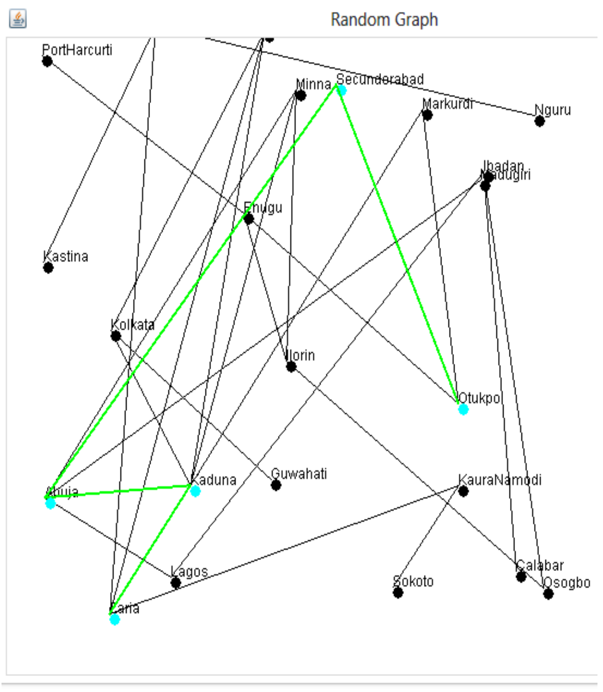

Figure 9: Part with Shortest travel time from Otukpo to Zaria (rather than the shortest path)

IX. CONCLUSION

Findings reveal that not all shortest travel routes mean shortest travel time. This has also contributed waste of time, money and leading passengers and railway operators to making irrational decisions which is contributory to the decline of patronage in the Nigerian railway industry and has set the Nigerian railway system below global competitive standard. Hence, the need to consider factors that could be incorporated into the shortest path which could yield the shortest travel time. This research rated shortest travel time a better consideration for railway operators and users than shortest travel route. Also, the automation and optimization of the Nigerian railway system would undoubtedly replace the drudgery of manual system and improve the output of the transportation industry. This would also reduce the time passengers spend on travel and also help in avoiding deadlocks on the railway tracks. Furthermore, it reduces stress on railway operators assuring that they are making the best decisions at every point in time. This could be adopted as a working tool to improve the railway system in many parts of the globe. The developed system is easy and convenient to use, assisting passengers in having better travel experience thereby improving the quality of life of people.

REFERENCES

[1] R. J. Shi, B. H. Mao, Y. Ding, Y. Bai, and Y. Chen, “Timetable Optimization of Rail Transit Loop Line with Transfer Coordination”, *Discrete Dynamics in Nature and Society,* (2016).

[2] M. Lewis, *Railways in the Greek and Roman world.* Paper presented at the Early Railways. A selection of papers from the first International Early Railways Conference, (2001).

[3] J. M. Carland, *The Colonial Office and Nigeria, 1898-1914*: Hoover press, (1985).

[4] Y. B. Usman, *Northern Nigeria: a century of transformation, 1903-2003*: Arewa House, Ahmadu Bello University, (2005).

[5] O. Komolafe, Overseeing the resurrection of Nigerian railways. *News Agency of Nigeria,* (2015).

[6] G. Chuku, *Igbo women and economic transformation in southeastern Nigeria, 1900-1960*: Routledge, (2015).

[7] D. B. Johnson, A note on Dijkstra's shortest path algorithm. *Journal of the ACM (JACM),* vol. 20, no. 3, (1973), pp.385-388.

[8] R. Jacob, M. Marathe, and K. A. Nagel, “A computational study of routing algorithms for realistic transportation networks”, *Journal of Experimental Algorithmics (JEA),* vol. 4, no. 6, (1999).

[9] A. L. Costa, M. D. C. Cunha, P. A. Coelho, and H. H. Einstein, Decision Support Systems for Real-World High-Speed Rail Planning. *Journal of Transportation Engineering,* vol. 142, no. 5, (2016).

[10] L. Rothkrantz, *Dynamic routing using maximal road capacity.* Paper presented at the Proceedings of the 16th International Conference on Computer Systems and Technologies, (2015).

[11] C. Sommer, “Shortest-path queries in static networks”, *ACM Computing Surveys (CSUR),* vol. 46, no. 4, (2014).

[12] Mordecai, “Dijkstra’s Shortest Path Algorithm” DPV 4.4, CLRS 24.3, Revised, October 23, 2014.

[13] A. Holder, Mathematical Programming Glossary. INFORMS Computing Society, http. *glossary. computing. society. informs. o rg, 10,* (2006).

[14] L. M. Rios, and N. V. Sahinidis, “Derivative-free optimization: a review of algorithms and comparison of software implementations”, *Journal of Global Optimization,* vol. 56, no.3, (2013), pp. 1247-1293.

[15] E. W. Dijkstra, “A note on two problems in connexion with graphs”, *Numerische mathematik,* vol. 1, no. 1, (1959), pp. 269-271.

[16] S. Irnich, and G. Desaulniers, “Shortest path problems with resource constraints. In *Column generation*, . Springer, Boston, MA, (2005), pp. 33-65.

[17] G. B. Dantzig, “Linear programming”, *Operations Research,* vol. 50, no. 1, (2002), pp. 42-47.

[18] P. Pandey, and S. Dixit, “Railway Route Optimization System Using Dijkstra Method” *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 2, no. 3), (2014), pp. 435-440.

[19] S. Jahan, and M. S. Hasan, “A Comparative study on Algorithms for Shortest-Route Problem and Some Extensions”, *International Journal of Basic and Applied Sciences IJBAS-IJENS*, vol. 11, (2011), pp. 167-177.

[20] R. C. Backhouse, J. Van Den Eijnde, and A. Van Gasteren, "Calculating path algorithms", *Science of Computer Programming,* vol. 22(1-2), (1994), pp. 3-19.

[21] D. Delling, P. Sanders, D. Schultes, and D. Wagner, Engineering route planning algorithms *Algorithmics of large and complex networks*, Springer, (2009), pp. 117-139.

[22] W. Zeng, and R. Church, Finding shortest paths on real road networks: the case for A. *International journal of geographical information science,* vol. 23, no. 4, (2009), pp. 531-543.

[23] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", *IEEE transactions on Systems Science and Cybernetics,* vol. 4, no. 2, (1968), pp. 100-107.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing. *science, 220*(4598), (1983), pp. 671-680.

[25] T. A. Ayoola, "Establishment of the Nigerian Railway Corporation", *Journal of Retracing Africa,* vol. 3, no. 1, (2016), pp. 21-42.