ISSN: 2141 – 3290
www.wojast.com

MODIFIED DIJKSTRA ALGORITHM FOR DETERMINING MULTIPLE SOURCE SHORTEST PATH OF HOSPITAL LOCATION IN RIVERS STATE (NIGERIA)

[1]OKENGWU U.A. AND [2]NWACHUKWU E.O.
[1,2]Department of Computer Science,
University of Port-Harcourt, Port-Harcourt. Nigeria.
[1]ugodepaker@yahoo.com and [2]enoch.nwachukwu@uniport.edu.ng
+234(0)08037239319[1] and +234(0)8033099611[2]

ABSTRACT

Determining the shortest path to a particular hospital in the case of emergency could determine patient survival or not. The approach currently on ground in River State, Nigeria is subjective (depending on the skill, experience and exposure of the personnel prescribing the distance and not based on an objective approach) at best. This paper has adapted Dijsktra algorithm based on its drawback to propose a modified Dijkstra algorithm capable of implementing multiple source shortest path distance optimization to varied hospital within River State. Matrix Laboratory (MatLab) and Hypertext preprocessor (PHP) were the simulation tools and language of implementation respectively. The results of our simulation were satisfactory having been able to implement Multiple Source Shortest Path across several hospitals spread across River State, Nigeria.

INTRODUCTION

River State, Nigeria is a commercial, multi-culture and age range city, with population density on the steady increase resulting in increased road creation, industries, residential estates as well creating chaotic and hectic traffic jam (CAPD, 2014). The health of every individual is of paramount importance. It is also important that securities of lives are guaranteed through quick response to health issues when they occur. The conventional approach adopted for distance optimization to a given health or medical center and hospital from a given community within Rivers State, Nigeria has been subjective at best; depending on the will, emotional state and experiences of an individual expert prescribing such information utilizing his knowledge (Osami, 2013). The loss of life, or even irreplaceable damages incurred by a patient due to poor distance prescription (delay time) to a given hospital has indeed been enormous within Rivers State. This paper proposes a Modified Dijskstra Algorithm for determining multiple source shortest path of hospital location in Rivers State (Nigeria).

Pióro *et al*. (2002), handled flow allocation problems in IP networks using Open Shortest Path First (OSPF) routing. They proposed methods for finding settlements of OSPF link weight system realizing the assumed demand pattern for the given network resources. They discovered that such settlements can result in a significantly better network performance compared with the simplified weight setting heuristics typically used nowadays. Although the configuration of the link weight system is primarily done in the network planning phase, additional re-optimizations are feasible in order to cope with major changes in traffic conditions and with major resources' failures and rearrangements. They formulated a relevant OSPF routing optimization problem, proves it's NP-completeness. Two basic approaches were considered (the direct approach and the two-phase approach) and the resulting optimization algorithms are presented.

John and Patrick (2004) applied the meta-heuristic method of Ant Colony Optimization (ACO) to solve set of Vehicle Routing Problems (VRP). Their procedure simulates the decision-making processes of ant colonies as they forage for food and is similar to other adaptive learning and artificial intelligence techniques. Modifications were made to the ACO algorithm used to solve the traditional traveling salesman problem in order to allow the search of the multiple routes of the VRP.

Xing (2011), handled the Capacitated Arc Routing Problem (CARP) which is representative of numerous practical applications, they considered an extended version of this problem that entails both total service time and fixed investment costs. We subsequently propose a Hybrid Ant Colony Optimization (ACO) algorithm (HACOA) to solve instances of the extended CARP. Their approach was characterized by the exploitation of heuristic information, adaptive parameters, and local optimization techniques: Two kinds of heuristic information, arc cluster information and arc priority information are obtained continuously from the solutions sampled to guide the subsequent optimization process. The adaptive parameters ease the burden of choosing initial values and facilitate improved and more robust results. Finally, local optimization, based on the two-opt heuristic, is employed to improve the overall performance of the proposed algorithm. The resulting HACOA is tested on four sets of benchmark problems containing a total of 87 instances with up to 140 nodes and 380 arcs. In order to evaluate the effectiveness of their proposed method, some existing capacitated arc routing heuristics are extended to cope with the extended version of this problem; the experimental results indicate that the proposed ACO method outperforms these heuristics.

Mathematically, optimization is the search for combination of parameters commonly referred to as decision or design variables ($X=[X_1, X_2, X_3....X_n]$) which minimize or maximize some ordinal quantity (C) (typically a scalar called a score) assigned by an objectives function or cost function (F), under a set of constraints ($g= [g_1, g2, g3...gn]$). For example, a general minimization case would be as follows: $f(x^1) \leq f(x)$, for all x_i belonging to x (Jason, 2011). The following are ingredient under optimization:

Decision Variables: The formulation of an optimization problem begins with identifying the underlying design variables, which are primarily varied during the optimization process (Kennerley, *et al* 2012). Other design parameters usually remain fixed or vary in relation to the design variables (Michael *et al*, 1999 and Malcolm *et al*, 2006). The first thumb rule of the formulation of an optimization problem is to choose as few design variables as possible (Kennerley *et al*, 2012). The outcome of that optimization procedure may indicate whether to include more design variables in a revised formulation or to replace some previously considered design variables with new design variables (Kennerley *et al*, 2012).

Objective functions: The next task in the formulation procedure is to find the objective function in terms of the design variables and other problem parameters. The common engineering objectives involve minimization of overall cost of manufacturing or minimization of overall weight of a component or maximization of total life of a product or others, (Leader, 2004).

Constraints: This represents some functional relationships among certain resource limitations, design variables and parameters satisfying certain physical phenomenon (Wenyu and Ya-Xiang, 2010). Constraints may have exact mathematical expressions or not. For example, maximum stress is a constraint of a structure (Dechter, 2003). If a structure has regular shape they have an exact mathematical relation of maximum stress with dimensions. But incase irregular shape, finite element simulation software may be necessary to compute the maximum stress (Leader, 2004)

METHODOLOGY

Dijsktra Algorithm is one of the predominantly shortest routing solutions to shortest path problem which is showed in Table 1.

1. **function** Dijkstra (*Graph*, *source*):
2. **for each** vertex *v* in *Graph*: *// Initializations*
3. dist[*v*]:= infinity; *// Unknown distance function from*

```
                                              // source to v
previous[v]:= undefined;               // Previous node in optimal path
end for                                    // 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 distance in dist [] ; // Source node in first case
 remove u from Q ;
 if dist[u] = infinity:
break ;                                   // all remaining vertices are
end if                                     // inaccessible from source

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 ;
decrease-key v in Q;                    // Reorder v in the Queue
end if
end for
end while
return dist;
end function
```

Table 1: Dijsktra Algorithm (Melissa, 2012)

Dijsktra Algorithm propagates one major demerit which fosters this research paper. Dijkstra Algorithm supports the Single Source Shortest Path; which imply that for any routing problem we must have one source vertex and a single destination or multiple destination path.

The Proposed Modified Dijkstra Algorithm for Determining Multiple Source Shortest Path of Hospital Location

The proposed Algorithm extends the scope and focus of the current algorithm, there providing a framework in which multiple source vertexes are possible. This Algorithm will be inappropriate in solving multiple source vertex problems which is indeed the bedrock of most real world problem, but still date few of these algorithm has be proposed for handling the aforementioned problem. The algorithm is depicted on Table 2.

Function Dijkstra // ***Graph, Source-Vertices***

For each Vertex (V) in the Graph (G) ∉ Source -Vertices (SV); **// V: = Hospitals in River State, SV: = Communities, Towns or Junctions all in River State**

Dist [V] := ∞; ***// Unknown distance for all non- source vertices in the graph***

Previous [V] : = Undefined;

End for

Dist [SV^1] := 0 ***// No previous nodes with optimal path form SV^1***

M^1: = First Queue holding the set of all nodes (V) in the Graph (G) ∉ Source-Vertices 1 (SV^1)

While M^1 ≠ **empty:**

Y^1:= Vertices in M^1 with Shortest distance in Dist [V];

Remove Y^1 from $M^{1;}$
If Dist[Y^1]: = ∞ // ***Unknown distance from neighboring nodes of Y^1 in the Graph;***
Break;
End if;
For each neighboring V of Y^1 with Dist [$V^1, V^2, V^3 \ldots V^n$]
K^1:=neighboring Vertices (V) of Y^1
K^1:= Dist [SV^1, Y^1] + Dist [Y^1, K^1]
If K^1[Dist]< Dist[$V^1, V^2, V^3 \ldots V^n$]
Remove K1 from M and add to SV^1, Y^1
Decrease-Key Y^1 in M^1
While M^1 = **empty**
Endif
End For
End while
While $M^2 \neq$ **empty:**
Y^2:= Vertices in M^1 with Shortest distance in Dist [V];

Remove Y^2 from $M^{2;}$
If Dist[Y^2]: = ∞ // ***Unknown distance from neighboring nodes of Y^2 in the Graph;***
Break;
End if;
For each neighboring V of Y^2 with Dist [$V^1, V^2, V^3 \ldots V^n$]
K^2:=neighboring Vertices (V) of Y^2
K^2:= Dist [SV^2, Y^2] + Dist [Y^2, K^2]
If K^2 [Dist]< Dist[$V^1, V^2, V^3 \ldots V^n$]
Remove K^2 from M and add to SV^2, Y^2
Decrease-Key Y^2 in M^2
While M^2 = **empty**
Endif
End For
End While
End

Table 2: The Proposed Modified Dijsktra Algorithm

PREVIOUSLY

The shortest path spans A-D-G-K-R-N-Q-T-Z, which is the sum total of these entire path. There we have 22km as the shortest path from A – Z for all vertices reachable from A.

CURRENTLY (Multiple Source and Destination Vertices Application)

(Source)B = E-H-L-M-P=T (Destination) 9
(Source)C = E-H-L-M-P=T (Destination) 11
(Source)D = G-K-R = (Destination) 8
(Source) E =H-L-M-P-T = (Destination) 9
(Source) F =G-K-R-T = (Destination) 12
(Source) G =K-R-T = (Destination) 9
(Source) I =M-P-S= (Destination) 6
(Source) J =N-R-T = (Destination) 9
(Source) K = R-T= (Destination) =7
.. . . . (Destination) S, Z, T

Implementation and Simulation

The implementation of the modified algorithm was handled utilizing Matrix Laboratory (MATLAB) which serves as our simulation tool in testing the algorithm interactively. It also provided an interactive environment for algorithm development, data visualization, data analysis, and numerical approach which was relevant to our numerical data set which was more appropriate than with spread sheets or traditional programming languages, such as C/C++ or Java. After pruning the dataset utilizing MATLAB, the algorithm was fully implemented utilizing Hypertext Preprocessor (PHP), which served as the language of implementation.

Simulation Results (*Application of the Modified Dijsktra Algorithm)*
Figure 1 shows the application of Dijkstra algorithm and the Modified approach adopted. The full application of the modified application fosters Multiple source path as opposed the single source path generated by Dijsktra Algorithm.

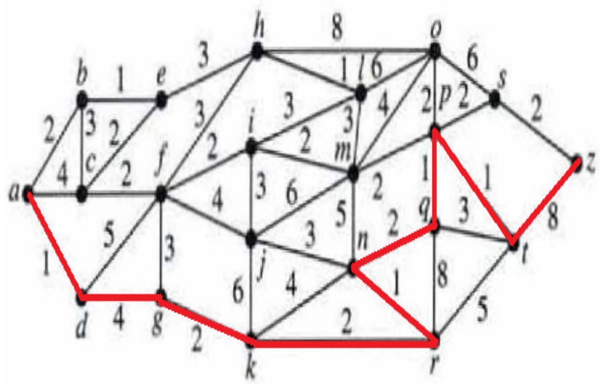

Figure 1: Modified Dijkstra Algorithm

DISCUSSION

The implemented algorithm provides an interactive base in determining multiple shortest hospital paths from various points (town, communities and junction) objectively as opposed to the subjective approach on ground previously. The result was satisfactory having been able to implement multiple source shortest paths as opposed to the single source shortest path sponsored by dijkstra algorithm.

CONCLUSION

This research has demonstrated the practical application of modified Dijsktra algorithm in determining multiple source shortest hospital paths in various hospitals within River State. This paper has adapted Dijsktra algorithm based on its drawback to propose a modified Dijkstra algorithm capable of implementing multiple source shortest path distance optimization to varied hospital within River State.

REFERENCES

CADP: Commercial Agriculture Development Project (2014), *Agricultural Production and commercialization,* http://www.cadpnigeria.org/crossriver/crscado.html, May 14, 2014

Dechter R. (2003), *Constraint Processing*, Morgan Kaufmann. ISBN 1-55860-890-7.

Jason B. (2011), *Clever Algorithms: Nature Inspired Programming Recipes,* http; cleverAlgorithms.com

John E. B. and Patrick R. M. (2004), *Ant colony optimization techniques for the vehicle routing problem*, Elsevier Advanced Engineering Informatics 18 (2004) 41–48.

Kennerley S.W., Dahmubed A.F., Lara A.H. and Wallis J.D. (2012), *Neurons in the frontal lobe encode the value of multiple decision variables,* http://www.ncbi.nlm.nih.gov/ pubmed/18752411.

Leader, J. J. (2004), *Numerical Analysis and Scientific Computation*. Addison Wesley. ISBN 0-201-73499-0.

Malcolm B., Ryan T., and Jeffrey W. (2006*), Predicting Returns with Managerial Decision, Variables: Is There a Small-Sample Bias?* The Journal of Finance, Vol. Lxi, No. 4

Melissa Y. *(2012), Shortest part Algorithm,* retrieved from math.mit.edu/~roth, May 2013

Michael L. P. & Paul W. G. (1999), *Neural correlates of decision variables in parietal cortex,* http://www.nature.com/nature/journal/v400/n6741/abs/400233a0.html, Nature 400, 233-238 (15 July 1999) | doi:10.1038/22268; Received 30 April 1999; Accepted 25 May 1999

Osami I. (2013), Implementing Vocational and Technical Education Programmes in South-South Nigeria: A Case of Rivers State´, International Journal of Scientific Research in Education, June 2013, 6(2), Pp. 128-148.

Pióro M., Szentesi A., Harmatos J., Jüttner A. Gajowniczek P. and Kozdrowski S. (2002), *On open shortest path first related network optimization problems*, Elsevier Performance Evaluation 48 (2002) 201–223.

Xing L.A. (2011), *A Hybrid Ant Colony Optimization Algorithm for the Extended Capacitated Arc Routing Problem*, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on (41) 4.

Wenyu S. and Ya-Xiang Y. (2010), *Optimization Theory and Methods: Nonlinear Programming*, Springer, ISBN 978-1441937650. p. 541