Babylonian Journal of Machine Learning
Vol.2023, **pp.** 7–16
DOI: https://doi.org/10.58496/BJML/2023/002; ISSN: xxxxx
https://mesopotamian.press/journals/index.php/BJML

Research Article

Improved GIS-T model for finding the shortest paths in graphs

Raed Abdulkareem Hasan [1,*], Mustafa Mahmood Akawee [2], Tole Sutikno [3]

[1] *Department of Electric, Faculty of Al-Hawija Technical Institute, Northern Technical University, Mosul, Iraq*.

[2] *Department of Arabic Languge, Imam Aladham University College, Baghdad, Iraq.*

[3] *Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.*

ARTICLE INFO

Article History

Received 10 Nov 2022
Accepted 24 Jan 2023
Published 21 Feb 2023

Keywords

Combinatorial optimization

Computational geometry

Dijkstra's algorithm

Geographic information

System-transportation

ABSTRACT

A system of models and methods is proposed for the stated problem, which is the development of software for searching the shortest paths in graphs. These models and methods are based on a comparative analysis of various types of geographic information system-transportation (GIS-T) and optimization methods that pertain to searching for the shortest paths in a graph. The goal of this research is to find a solution to the problem. The fundamental ideas behind GIS-T and computational geometry (CG), in addition to the approach of combinatorial optimization, were utilized in this endeavor. Compilation of the newly developed software takes place in the Java Eclipse development environment. Examples of graphic interfaces and potential solutions are used to illustrate its capabilities in relation to the quest for the shortest routes possible within the confines of a particular geographic area. The compiled geographic information system (GIS) tool known as "route" determines the routes that are the shortest distance between existing settlements as well as those that have been defined by the user. In addition to this, it enables the problem to be solved in circumstances in which certain roads become inoperable or certain settlements may be reached via the specified field roads.

1. INTRODUCTION

2. GEOGRAPHIC INFORMATION SYSTEM

2.1 The Relations between Geographic Information Systems and Computational Geometry

While CG may wish to regard GIS as a practical example of CG, this is not historically correct. For the most part, the GIS and CG communities have evolved their understanding of geometric computing separately [21]. GIS progress has surpassed that of CG in certain critical areas ("topological structure" for the storage of planar sub-divisions, "triangular irregular networks (TINs)"-triangulated terrains, and interpolation features of the Voronoi and Delaunay"). Computer cartography became popular in the 1960's. Many fundamental principles (map layers, topological structure, and TINs) date back to the Department of Canadian Government's Land Inventory or the Harvard Lab for spatial analysis and computer graphics. The most popular commercial GIS system, Environmental Systems Research Institute (ESRI)'s arc/info, began with Harvard Lab Technology.

In recent years, the relationship between CG and GIS has developed. As a result, companies like ESRI and Caliper Corporation have been employing members of the CG community and incorporating research findings, such as the nearest-neighbor properties of Delaunay triangulations and the utilization of Voronoi diagrams of line segments to aid in "buffering" (offsetting) operations. However, the acceptance of research findings by other domains, such as GIS, has been hampered by delays in publication and a lack of printed materials in CG, as complained by many related authorities.

*Corresponding author. Email: it@gmail.com

2.2 Geometric Problems in Geographic Information Systems

When looking at geometric difficulties in GIS, two aspects should be considered: first, while scholars in CG may see themselves as problem solvers, the concepts of shape, space, and geometric computing that emerge can be more valuable than the solution of any given problem. Secondly, the "geometric" component of a GIS is merely one among many. In [22]-[25] presents four perspectives on GIS in a survey for cartographers, geometers should consider these perspectives when judging the relevance of problems inspired by GIS: i) automated mapping: enabling the creation of standard maps, ii) map analysis: provides cheaper overlay and measurement tools compared to the conventional methods, iii) inventory: provides geographic access capabilities to the existing corporate and governmental databases, and iv) spatial analysis and spatial decision support: facilitating new ways of using old data via provision of analysis and query tools for the users. Hence, this work will rely on the use of the spatial map analysis.

2.3 Geographic Information Systems-Transportation Data Models

GIS-T are integrated technologies, software, people, data, firms, and institutional frameworks for gathering, storing, analyzing, and disseminating specific earth-related information [25]-[27]. Geographic regions and transportation systems that impact or are affected by these systems are the specific forms of information. Among the important and rapidly developing uses of GIS is transportation [28], [29].

It may seem easy to design data models for GIS-T applications at first glance. But unlike "many other GIS apps, GIS- T focuses on a single topic of study: the transportation network (TPN) in a given study area." On the other hand, it is a complex task to digitally represent these networks. TPN data is complicated because it is frequently multimodal, spans multiple domains, and has various logical interpretations based on the user. The requirement to reference events, such as accidents or pavement quality, in the network arises frequently. But based on the map scale of interest, a network might have different representations. The necessity to depict the links between network and non-network data also frequently arises. Hence, advanced GIS-T applications necessitate the capacity to track objects or conditions over time and handle navigation issues [30].

The rich properties and features of TPNs in the real world and the data models employed to represent them in a computer are vastly different. To begin with, many GIS software systems identify only simple geometric pre-defined elements, i.e. points, lines, and polygons. Data such as complex pathways, origin/destination flows, and temporal variations are difficult to handle in most GIS applications [31]. Many existing GIS include restricted and clunky representations of transportation elements like underpasses and overpasses, as well as links like intermodal transfers between highways and commuter rail [32]. Because transportation features are separated into distinct GIS layers, such data mostly lacks topological information between entities [33], [34].

3 THE MULTIFACETED NATURE OF TRANSPORTATION DATA

Physical representations of transportation entities are clear, but they can also have logical links with other transportation elements. Secondly, entities exist in the actual world as well as in databases or virtual environments. The logical and physical realms frequently have one-to-many relationships, which complicates the design of such databases. The real/physical mode refers to the transportation facilities in the real world that have been built and used, e.g. physical facilities like intersections, highways, and interchanges. Actual, logical, or legally defined transportation elements like federal and state roads also exist in the real world. Physical and logical entities frequently have one-to-many relationships that can exist in both directions [9]. For instance, the same physical highway may be shared by two state roads. A state route, on the other hand, can and frequently does cross many actual streets in a city.

Data structures, like nodes, connections, polygons, and networks, relate to logical entities. The geometry and attribute data linked with the transportation element relate to virtual and physical entities. This latter data is frequently represented graphically by the GIS. When presenting the network at a certain scale, a one-to-many relationship may occur if two or more network links are associated with the same graphical line "e.g. showing a 2-way street that is represented logically by two directed arcs as one cartographic line at small map scales. In addition, a single link might be represented by numerous cartographic lines, e.g. showing modal-specific flow in a network link."

3.1 Transportation Networks in a Geographic Information Systems: "The Arc-Node Model"

A network is a form of graph, which is a mathematical structure that depicts patterns and connections between objects. A network, rather than representing relationships, depicts movements or interactions between point locations. Arcs are the flow conduits between nodes, while nodes are the actual points of flow initiation, termination, or relay. Arcs, which can be either physical conduits like a road segment or logical relationships like airline service between two cities, connect nodes. There are two types of arcs: directed and undirected. The node ordering for a directed arc indicates the direction of flow. A network can contain weights associated with each arc, which is a significant difference between a graph and a network. Each arc is associated with a weight that indicates the penalty that one unit of flow incurs when passing through it [8]. We only deal with directed networks, i.e. a network made up of directed arcs in the basic "node-arc" form of a TPN because transportation systems often have significant directional flow features, e.g. one-way streets and variations in directional travel times based on the time of day.

3.2 Street Transportation Network

Nodes relate to street crossings in a street TPN, whereas arcs relate to roadway lengths between crossings. When portraying limited-access highways, nodes indicate interchanges and arcs symbolize highway segments. A two-way street or segments of a parallel limited access highway with opposite directional flow are represented by two directed arcs angled in opposite directions. The unit flow cost of traversing the arc is represented by a generalized cost function. These normally include two primary components: any out-of-pocket charges such as tolls and the necessary travel time. The latter is frequently associated with current flow. Though nodes represent the street network intersections, the arc-node model may depict these at various resolution levels. Figures 1(a) and (b) show two approaches for representing a street network as a network. The intersection is consolidated into a single node in Figure 1(a). Though efficient, this strategy is oversimplified and fails to account for a crucial intersection property: "the variable turn impedances associated with distinct directions of flow across the intersection [9]. A left turn, for instance, may take longer than a right turn or straight through the junction. Additionally, there may be turn restrictions (e.g. "no left turn")."

We can use the extended depiction in Figure 1(b) to represent these aspects. The intersection is expanded to four nodes with interconnecting arcs reflecting direction-specific travel in this manner. Even though this approach can capture the requisite intersection and turn features, the significant rise in the number of arcs and nodes in the network poses a concern. This not only significantly increases the demands for a network's data storage, but it can also reduce network analytical performance [9]. For instance, the time it takes for shortest path routines to run increases with the number of network nodes.

A transportation system is generally divided into discrete, modal-specific sub-structures using the node-arc format. For instance, an urban system of transportation may be divided into separate networks for public and private transportation. Transfer arcs-which stand for modal transfers-connect the various sub-networks. Even though the old node-arc layout is still extensively used, new models of TPN treat various modes in a more complex manner that avoids artificial separation.

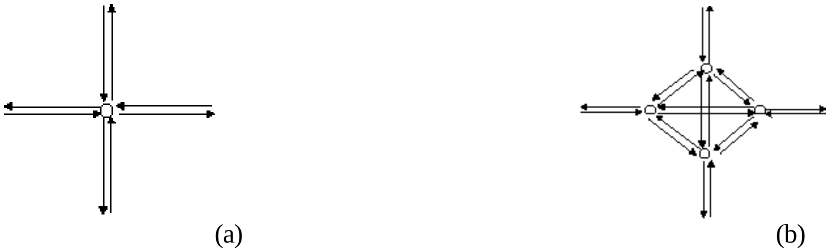


Fig. 1. Intersection; (a) single node representation and (b) expanded representation

4. METHOD

4.1 Relational Databases

The relational model is the most frequent logical data architecture used to support the node-arc representation. Turn tables and reference address tables are other examples of auxiliary relationships. As shown in Figure 1(b), turn tables are relations for storing data on enlarged intersection representations. Each direction of travel across an intersection is represented by a tuple in the turn table. The travel cost associated with that travel direction is kept in a separate field or perhaps a pointer to a flow cost function.

"A turn restriction can be indicated with a reserved character like a negative integer. The turn table technique, like the extended intersection representation in the formal arc-node model, is effective but inefficient. For each intersection in the street network, the turn table requires the addition of 12 tuples to the database. For a detailed metropolitan street network, the sum can be rather considerable [7]." We frequently want to incorporate information about network address locations. This is important for network-wide address matching, i.e. geo-referencing entities (such as homes and businesses) based on their street address. We can also create a reference address relationship to keep track of address information.

The "from address" and "to address" parameters contain address ranges for the specified arc and other related information like the area of application of the address range and whether the address numbers are always even or odd on each side. Often, the street name that corresponds to the arc must be divided into the following fields: i) a street prefix, e.g. "north", ii) a street name, e.g. "Palestine", iii) a street type, e.g. "avenue", and iv) a street suffix, e.g. "east".

4.2 Challenges With The Arc-Node Model

A GIS frequently requires planar embedding of the node-arc network to enhance database integrity. This necessitates the presence of a node at each arc crossing. Planar embedding ensures that the generated spatial data layer's topology connectivity is appropriate, such as by encircling all polygons with arcs. The criterion that all arc intersections correlate with network nodes in a planar network, on the other hand, does not reflect the real features of a TPN [7], [8]. A restricted-access highway, for instance, may cross over or under a surface street. By constructing a node at that overpass or underpass, automotive traffic will be able to exit or enter the highway. This might cause issues with network routing because the connection may or may not exist in reality.

The issue of planar networks can be partly resolved by relaxing the enforcement of planar topological consistency. When creating a spatial data layer, some GIS software allows for varied levels of enforced topological integrity, such as not "building" the polygons implied by the topology of the node-arc-area. However, this can cause issues with data integrity. Another option is to utilize the enlarged intersection representation to limit turns at overpasses and underpasses, albeit this is more of a "work around" than a true expansion beyond the planar network. As a result of this issue, some new GIS-T data models go much beyond the old arc-node model.

Another flaw in the classic network model is the presumption of the homogenous nature of arcs, in the sense that arc attributes do not vary between end nodes [7]. This does not apply for many transportation applications, like pavement management, where quality can vary significantly within a single street segment. Traffic flow modeling is a less obvious example. The arc-node model implies that the levels of network flow and the related features are homogenous within the arc, which is a less evident example. It's worth noting that this enforces a set degree of spatial resolution, which means we can't use data on lane configurations or street design. This data is frequently used in advanced applications, such as intelligent transport system (ITS).

The difficulty of supporting one-to-many relations within transportation entities is the fourth problem. The links between real or physical transportation entities and real or logical transportation entities are frequently ambiguous, as highlighted by [5]. These relationships cannot be accommodated by the standard planar network model and its related counterpart.

4.3 Basic Models And Methods Of Combinatorial Optimization

Combinatorial optimization is a problem in theoretical computer science and applied mathematics that involves finding an optimal object from a finite set of things. Exhaustive search is not possible in many of these situations, it works on optimization problems where the set of plausible options is discrete or may be minimized to discrete with the goal of discovering the optimal solution. The traveling salesman problem and the minimum spanning tree problem are two common combinatorial optimization problems. Combinatorial optimization is an aspect of optimization that is related to algorithm theory, operations research, and computational complexity theory. It is useful in many important fields like AI, ML, mathematics, and software engineering [1].

Despite the close link between these topics in literature, some scholars believe that discrete optimization is comprised of integer programming in combination with combinatorial optimization, which focuses on optimization problems that deal with graphs, matroids, and related structures. It frequently entails choosing the most efficient method to allocate resources used to find solutions to mathematical problems [2]. There is a substantial amount of work on polynomial- time methods for particular types of discrete optimization, much of it unified through linear programming theory. Under this paradigm, some of the problems that fit in are shortest paths and shortest path trees, flows and circulations, spanning trees, matching, and matroid problems.

The following topics are included in the current research literature for non-deterministic polynomial (NP)-complete discrete optimization problems: i) polynomial-time exactly-solvable special cases of the problem at hand, ii) algorithms that optimally perform on "random" cases, e.g. for traveling salesman problem (TSP), iii) approximation algorithms that run in polynomial time and find a near-optimal solution, and iv) solving real-world problems that arise in practice but do not often show the worst-case behavior found in NP-complete problems, e.g. TSP instances with numerous nodes. Because combinatorial optimization issues may be considered a search for the optimal item among a set of discrete items, any search technique or metaheuristic can theoretically be used to solve them. Generic search algorithms, however, do not ensure that they will identify the best solution or that they will run quickly in polynomial time. This is expected because some discrete optimization problems, such as the TSP are NP-complete unless P=NP. Specific problems included: i) vehicle routing problem, ii) TSP, iii) minimum spanning tree problem,iv) linear programming, v) integer programming, and vi) eight queens puzzle (a constraint satisfaction problem). When using traditional combinatorial optimization techniques to solve this problem, the target function is commonly expressed as the number of unsatisfied constraints (for example, the number of attacks) rather than as a single Boolean that indicates whether the problem is solved or not (the Knapsack problem) [2].

4.4 Combinatorial Models Of Routes

Road systems are best represented in science and mathematics using graphs and trees. They also show how to use the basic combinatorial optimization instrument to reduce the length of a road or another road characteristic. As a result, recall the key characteristics of these items that are used to build the GIS "route" optimization model [10]. A graph G=(N, A) is comprised of a set of N nodes and A is the arcs which is a set of ordered pairs of nodes. |N|=n and |A|=m.

A graph is said to be undirected if there is an arc (v, u) ∈A for each arc (u, v) ∈A. An undirected graph is said to be connected if a path exists between each pair of arcs. A graph is weighted if each arc in the graph is associated with a weight or number. A multipath graph is a graph with more than one arc between node pairs.

The sub-graph T=(N, A’) is a spanning tree of connected graph G, where A’⊂ A and T is a tree. A graph with n number of nodes can have m ranging between 0 and n2. A dense graph is a graph of n nodes with O(n2) arcs and sparse if the graph has O(n) arcs. Like trees, gra1, if there is a directed arc from u to v,

uv =α 0, otherwise (1)

A weighted graph G=(N, A, d) can be represented as (2):

a duv, if there is a directed arc from u to v,

uv = α ∞, otherwise (2)

A is symmetric if the graph is undirected, i.e. $auv = avu$. This graph representation demands O(n2) space and not useful if G is sparse.

Node-arc incidence matrix B. This is an n× m matrix such that (3):

1, if there is a directed arc a into u;

bua = α −1, if there is a directed arc a out of u;

0, otherwise (3)

List of arcs. The graph representation is done as a list of triples as in (4):

$$G = \{(u, v, duv\ )\}, \forall\ (u, v) \in A \qquad (4)$$

Undirected graphs are expected to have each arc (u, v, duv) appearing twice, one in both directions (forward and reverse). This isn't essential if we start the list with a "directed/undirected" indication variable. This representation requires only O(m) space, making it the most space efficient representations (the majority of operations are quite difficult), it's useful for graph creation and external storage.

$$G = ((A, B, 15); (A, C, 17); (B, D, 53); (C, D, 20); (C, E, 21); (C, F, 19); (D, F, 81) \qquad (5)$$

The representation of the graph in Figure 2:

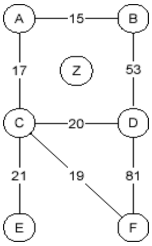


Fig. 2. An example of undirected graph

Adjacency list. G[1... n] is an array that stores pointers to lists of neighboring nodes, for example G[i] links to the list of nodes close to i. The position of an adjacent node in array D[1... n] is represented by an element on node[i]. Undirected arcs are depicted by two arcs that are directed in opposite directions. This is an efficient and useful method that takes up O(n+2m) space. Trees appear in applications naturally or are created to organize data for quick access. Now, we'll look at the following types of trees.

Spanning trees. Consider G=(N, A) as a graph and T=(N', A') as a tree, i.e. a sub-graph of G and T is a spanning tree of G if N'=N, meaning that it is a tree with the entirety of nodes of G. Spanning trees frequently appear in applications, they are considered the most viable and optimal solutions to most problems of network flow. Minimum spanning trees and shortest path spanning trees. The term "tree traversal" refers to the process of 'visiting' each node in a specific order. It's debatable whether the order matters. Each of the visited nodes undergoes some processing, which is dependent on the application. This application-specific processing is handled by the PreVisit, Visit, and PostVisit procedures. The general operations on trees include traversing all the nodes in a tree, finding, adding, and deleting nodes (or arcs).

5. RESULTS AND DISCUSSION

5.1 Dijkstra’s Algorithm For Shortest Path

Consider the most fundamental form of method for locating the shortest paths in a network. The best algorithm of this type is mostly selected using the basic rule as in (6):

Select $u \in S$ with smallest $D[u]$ (6)

Dijkstra’s shortest path algorithm is the most famous of this type [11]:

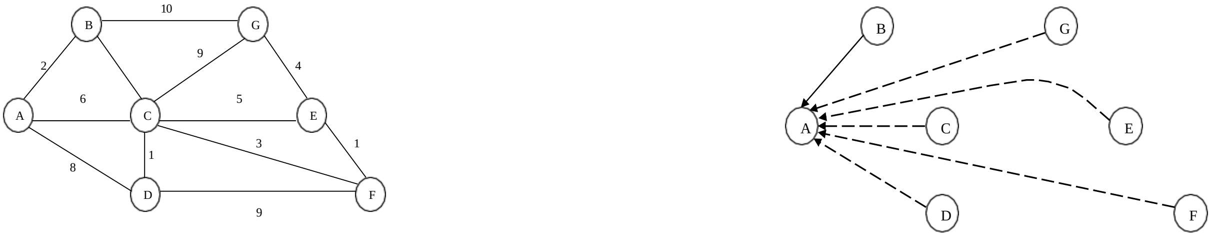
Theorem 1. "If G has positive arc distances, then when any u is selected from S its distance D[u] is optimal and it is never *added to S again."*

The original form of Dijkstra’s shortest path algorithm is as:

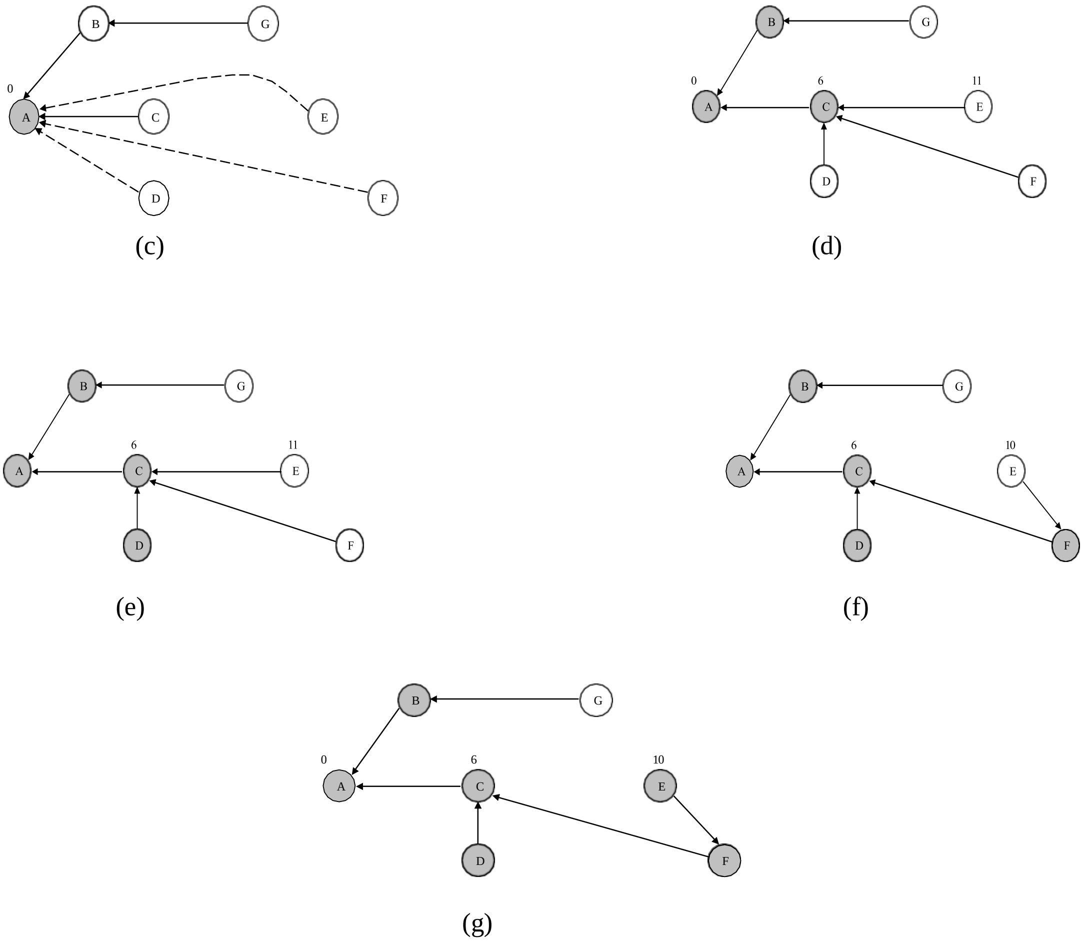
{S is represented as an unordered list}

```
    Initialize $(r, G, p, D)$
    $S: = N; D[r]:= 0;$
    $p[r]:=$ null WHILE S
    is not empty DO
            $u$ := DeleteMin (S)
            for each $v \in$ Adj $(u)$ DO
              if    $D[v] > D[u] + d_{uv}$ then
                    $D[v] := D[u] + d_{uv}$
                        $p[v] := u$
ENDIF
ENDFOR
 ENDWHILE
ENDAlg Dijkstra"
```

Being that they are yet to receive their final (optimal) D[.] value, the nodes in S are referred to as temporary at any step of this algorithm. The permanent nodes are those that are not in S and have their optimal value of D[.]. For any node u, the sequence of D[u] values decline monotonously from ∞ to D*[u], the ideal value. The complexity of this algorithm is *$O(n^2)$*. Figures 3(a)-(g) an example of Dijkstra’s shortest path algorithm, (a) A, B, 2, (b) A, C, 6, (c) A, C, 6, (d) A, D, 13, (e) A, E, 17, (f) A, F, 28, and (g) A, F, 15 gives an example of application of this algorithm



(a) (b)



(c)

(d)

(e)

(f)

(g)

Fig. 3. An example of Dijkstra’s shortest path algorithm, (a) A, B, 2, (b) A, C, 6, (c) A, C, 6, (d) A, D, 13, (e) A, E, 17, (f) A, F, 28, and (g) A, F, 15

5.2 Geographic Information Systems “Route” For searching The Shortest Path Between The Settlements in Iraq

GIS “route” operates in an interactive mode and provides entering, processing, and displaying of results with the accuracy of minutes of arc. In order to facilitate perception and to increase the quality of analysis, all basic input and output data are displayed in graphical and numerical form. Basic form of GIS “route”. For starting of GIS “route”, run the Eclipse file Iraq|Test.java. After running Test.java of GIS “route”, the basic form “GIS for search of optimal routes in Iraq” being given in Figure 4 becomes active. At this moment, the initial set of Iraq’s settlements is displayed on the map as well as the road network

(in green) that connects them. For the selected two cities, “starting city” and “destination (end) city”, the shortest path is automatically displayed as a red line on the map of Iraq as shown in Figure 5.

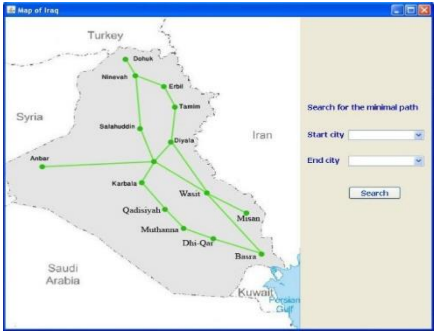
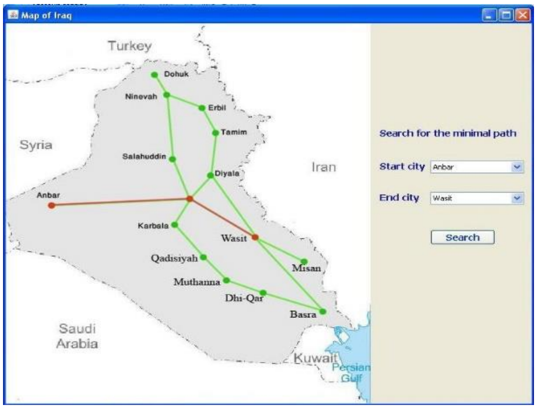


Fig. 4. Basic form of GIS “route”



Search for the minimal path

Start city: Anbar

End city: Wasit

Fig. 5. Finding of the shortest path between two cities (Anbar and Wasit)

6. CONCLUSION

On the basis of an analysis of the literature pertaining to GIS and GIS-T models as well as combinatorial optimization methods pertaining to searching the shortest paths in a graph, a system of models and methods is proposed for solving the stated problem: the development of software for searching the shortest paths in graphs. In particular, a comparative analysis of different types of GIS-T was undertaken. As well, a series of optimization methods are analyzed that allow us to search for optimal routes on the graphs with various constraints and time efficiency. For solving this problem, the basic concepts of GIS-T and CG were used, as well as the combinatorial optimization approach based on Dijkstra's algorithm. As a result, software for searching the shortest paths in graphs was developed, which allowed us to find the shortest route for the given geographical area. It is compiled in the Java Eclipse programming environment and illustrated with the user directory, together with the interface forms and examples of solutions.

The compiled model and algorithms are used for the development of software for a GIS "route" that allows to find the shortest routes for both existing settlements and interactively defined ones. Besides, it permits solving the problem in conditions when some roads become inoperative or some settlements might be reached via the specified field roads. Meanwhile, these are the proposed structure and technology for obtaining the inter-city distances from the map, which allows using the proposed software for new regions without the use of special GIS tools. Besides, the experiment with the developed software for variable geographical regions has certified its computational efficiency. This means that the task for the master's work is performed.

Conflicts Of Interest

The author's paper declares that there are no relationships or affiliations that could create conflicts of interest.

Funding

No institutional affiliations or sponsorship agreements are mentioned in the paper, indicating a lack of external funding.

Acknowledgment

The author extends gratitude to the institution for fostering a collaborative atmosphere that enhanced the quality of this research.

References

[1] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity. Mineola, New York: Dover Publication, 1998.

[2] E. M. Reingold, J. Nievergelt, and N. Deo, Combinatorial Algorithms: Theory and Practice. New Jersey, USA: Prentice Hal, 1977.

[3] F. P. Preparata and M. I. Shamos, Computational Geometry. Berlin, Heidelberg: Springer Science & Business Media, 1985.

[4] H. J. Miller and S. -L. Shaw, Geographic information systems for transportation: principles and applications. New York: Oxford University Press, 2001.

[5] D. Fletcher, “Modeling GIS transportation networks,” in Proceedings of the 25th Annual Meeting of the Urban and Regional Information Systems Association, 1987, pp. 84–92.

[6] L. Wiggins, K. Deuker, J. Ferreira, C. Merry, Z. Peng, and B. Spear, “Application Challenges for Geographic Information Science: Implications for Research, Education and Policy for Transportation Planning and Management,” Journal of the Urban and Regional Information Systems Association, vol. 12, no. 2, pp. 51–59, 2000.

[7] M. F. Goodchild, “Geographic information systems and disaggregate transportation modeling,” Geographical Systems, vol. 5, no. 1–2, pp. 19–44, 1998.

[8] B. D. Spear and T. R. Lakshmanan, “The role of GIS in transportation planning and analysis,” Geographical Systems, vol. 5, no. 1–2, pp. 45–58, 1998.

[9] S. L. Shaw, “GIS for urban travel demand analysis: Requirements and alternatives,” Computers, Environment and Urban Systems, vol. 17, no. 1, pp. 15–29, 1993, doi: 10.1016/0198-9715(93)90003-N.

[10] J. Sutton, “Data attribution and network representation issues in GIS and transportation,” Transportation Planning and Technology, vol. 21, no. 1–2, pp. 25–41, 1998, doi: 10.1080/03081069708717600.

[11] M. Alajanbi, M. A. Ismail, R. A. Hasan, and J. Sulaiman, "Intrusion Detection: A Review," Mesopotamian Journal of CyberSecurity, vol. 2021, pp. 1-4, 2021, doi: 10.58496/MJCS/2021/001.

[12] D. Nagaraju and V. Saritha, “An evolutionary multi-objective approach for resource scheduling in mobile cloud computing,” International Journal of Intelligent Engineering and Systems, vol. 10, no. 1, pp. 12–21, 2017, doi: 10.22266/ijies2017.0228.02.

[13] H. D. K. Al-janabi, H. D. K. Al-janabi and R. A. Hasan Al-Bukamrh, "Impact of Light Pulses Generator in Communication System Application by Utilizing Gaussian Optical Pulse," 2019 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 2019, pp. 459-464, doi: 10.1109/CSCS.2019.00084.

[14] C. S. Pawar and R. B. Wagh, “Priority based dynamic resource allocation in Cloud computing with modified waiting queue,” in 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, pp. 311–316, doi: 10.1109/ISSP.2013.6526925.

[15] H. Peng, W. S. Wen, M. L. Tseng, and L. L. Li, “Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment,” Applied Soft Computing Journal, vol. 80, pp. 534–545, 2019, doi: 10.1016/j.asoc.2019.04.027.

[16] P. S. Pillai and S. Rao, “Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory,”

[17] IEEE Systems Journal, vol. 10, no. 2, pp. 637–648, 2016, doi: 10.1109/JSYST.2014.2314861.

[18] V. Ramharuk, “Survivable cloud multi-robotics framework for heterogeneous environments,” M.S. thesis, School of Computing, University of South Africa, Tshwane, South Africa, 2015.

[19] M. J. Singh, “A Survey on QoS Based Task Scheduling Approach in Grid Computing,” International Journal of Engineering Trends and Technology, vol. 8, no. 7, pp. 359–366, 2014, doi: 10.14445/22315381/ijett-v8p265.

[20] C. Tang, S. Xiao, X. Wei, M. Hao, and W. Chen, “Energy Efficient and Deadline Satisfied Task Scheduling in Mobile Cloud Computing,” in 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), 2018, pp. 198–205, doi: 10.1109/BigComp.2018.00037.

[21] R. A. Hasan, R. A. I. Alhayali, M. A. Mohammed, and T. Sutikno, "An improved fish swarm algorithm to assign tasks and cut down on latency in cloud computing," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 20, no. 5, pp. 1103-1108, 2022, doi: 10.12928/TELKOMNIKA.v20i5.22645.

[22] O. A. Hammood et al., “An effective transmit packet coding with trust-based relay nodes in VANETs,” Bulletin of Electrical Engineering and Informatics, vol. 9, no. 2, pp. 685–697, 2020, doi: 10.11591/eei.v9i2.1653.

[23] R. A. Hasan, M. N. Mohammed, M. A. B. Ameedeen, and E. T. Khalaf, "Dynamic load balancing model based on server status (DLBS) for green computing," Advanced Science Letters, vol. 24, no. 10, pp. 7777-7782, 2018, doi: 10.1166/asl.2018.13016.

[24] M. Aljanabi, S. N. Abd-Alwahab, R. R. Saedudin, H. R. Ebraheem, R. Hadi, and M. A. Ismail, "Cloud Computing Issues, Challenges, and Needs: A Survey," JOIV: International Journal on Informatics Visualization, vol. 5, no. 3,pp. 298-305, 2021, doi: 10.30630/joiv.5.3.671.

[25] R. A Hasan, S. N. Suhel, and M. A. Ahmed, “Correlation with the fundamental PSO and PSO modifications to be hybrid swarm optimization,” Iraqi Journal for Computer Science and Mathematics, vol. 2, no. 2, pp. 25–32, 2021, doi: 10.52866/ijcsm.2021.02.02.004.

[26] H. S. Salih, M. Yaseen, and M. Aljanabi, "Implementing an Automated Inventory Management System for Small and Medium-sized Enterprises," Iraqi Journal For Computer Science and Mathematics, vol. 4, no. 2, pp. 238-244, 2023, doi: 10.52866/ijcsm.2023.02.02.021.

[27] A. H. Ali, M. Aljanabi, and M. A. Ahmed, "Fuzzy generalized Hebbian algorithm for large-scale intrusion detection system," International Journal of Integrated Engineering, vol. 12, no. 1, pp. 81-90, doi: 10.30880/ijie.2020.12.01.008.

[28] R. A. Hasan, H. W. Abdulwahid, and A. S. Abdalzahra, “Using Ideal Time Horizon for Energy Cost Determination,” Iraqi Journal for Computer Science and Mathematics, pp. 9–13, 2021, doi: 10.52866/ijcsm.2021.02.01.002.

[29] W. A. Hammood, R. A. Arshah, S. M. Asmara, and O. A. Hammood, “User Authentication Model based on Mobile Phone IMEI Number: A Proposed Method Application for Online Banking System,” in 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), 2021, pp. 411–416, doi: 10.1109/ICSECS52883.2021.00081.

[30] W. A. Hammood, S. M. Asmara, R. A. Arshah, O. A. Hammood, H. A. Halbusi, and M. A. A. -Sharafi, "Factors influencing the success of information systems in flood early warning and response systems context," Telkomnika (Telecommunication Computing Electronics and Control), vol. 18, no. 6, pp. 2956–2961, 2020, doi: 10.12928/TELKOMNIKA.v18i6.14666.

[31] A. A. Ayoob, G. Su, D. Wang, M. N. Mohammed, and O. A. Hammood, "Hybrid LTE-VANETs based optimal radio access selection," Lecture Notes on Data Engineering and Communications Technologies, vol. 5, pp. 189–200, 2018, doi: 10.1007/978-3-319-59427-9_21.

[32] M. Yaseen, H. S. Salih, M. Aljanabi, A. H. Ali, and S. A. Abed, "Improving Process Efficiency in Iraqi universities: a proposed management information system," Iraqi Journal For Computer Science and Mathematics, vol. 4, no. 1, pp. 211-219, 2023, doi: 10.52866/ijcsm.2023.01.01.0020.

[33] S. Atheel Sabih, F. Y. Omar, A. Mohammad, Z. Abbood, and S. M. Mahdi, "SEEK Mobility Adaptive Protocol Destination Seeker Media Access Control Protocol for Mobile WSNs," Iraqi Journal For Computer Science and Mathematics, vol. 4, no. 1, pp. 130-145, 01/07 2023.

[34] R. Daoud, Z. H. Ali, Z. M. Abdullah, and A. H. Ahmed, "Neural Network Based Assessment the performance of the triangular in-tegrated collector," NTU Journal of Renewable Energy, vol. 1, no. 1, pp. 43–49, 2021, doi: 10.56286/ntujre.v1i1.6.

[35] A. H. Ahmed, R. Daoud, and Z. H. Ali, "Theoretical and experimental study of the effect of concentration ratio on CTPTC performance," NTU Journal of Renewable Energy, vol. 1, no. 1, pp. 30–37, 2021, doi: 10.56286/ntujre.v1i1.5.