GIS Oriented Platform For Solving Real World Logistic Vehicle Routing Problem

Md. Shahid Uz Zaman, Yen-Wei Chen and Hayao Miyagi
Graduate School of Engineering and Science
University of the Ryukyus
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
Email: zaman@sys.ie.u-ryukyu.ac.jp

Abstract: Logistics optimization problems related with vehicle routing such as warehouse locating, track scheduling, customer order delivery, wastage pickup etc. are very interesting and important issues to date. Many Vehicle Routing and Scheduling Systems (VRSS) have been developed/proposed to optimize the logistics problems. But majority of them are dedicated to a particular problem and are unable to handle the real world spatial data directly. The system developed for one problem may not be suitable for others due to inter-problem constraint variations. The constraints may include geographical, environmental and road traffic nature of the working region along with other constraints related with the problem. So the developer always needs to modify the original routing algorithm in order to fulfill the purpose. In our study, we propose a general-purpose platform by combining GIS road map and Database Management System (DBMS), so that VRSS can interact with real world spatial data directly to solve different kinds of vehicle routing problems. Using the features of our developed system, the developer can frequently modify the existing algorithm or create a new one to serve the purpose.

1. Introduction

Vehicle Routing and Scheduling Problem (VRSP) such as warehouse locating, track scheduling, customer order delivery, waste pick-up etc. are important logistics problems in transportation. There are many Vehicle Routing and Scheduling System (VRSS)/algorithms which have been developed to solve Vehicle Routing Problem (VRP) optimally [7,9,10,12,13,14,16,17,18]. Research work is still going on to develop faster and reliable algorithms capable of tackling the growing size and the complexity of real life applications. A VRSS has no value until it can be applied in reality to solve some real life VRP. Unfortunately, very few VRSS have been implemented really to solve real world VRP problems. The main cause behind this is the inability of VRSS to interact with real world spatial data. The quality of modern day VRSS is determined not only by the effectiveness of the algorithm, but by also how efficiently it can handle the real world data. In past few years, a tremendous growth of Geographical Information System (GIS) technology has been achieved mainly due to advancement of computer hardware such as CPU, memory, graphics and so on, which makes tasks very simple when interacting with spatially referenced data. The technological advancement enhances computer's computing power as well as space to store huge amount of information. This idea led to the use of Database Management System (DBMS) when it is required to build an information-related system. The main advantages of DBMS over traditional systems are the data organization and its maintenance. The huge amount of data contained by GIS can be stored and manipulated efficiently using DBMS technique. Taking these advantages the authors [1,2,10,17,18] tried to build some support systems for solving VRP using GIS technique

The intent of this paper is to build a platform combining GIS and DBMS together so that VRSS can derive spatial data directly from real world. In the proposed platform, VRSS is the external unit so that different kinds of VRSS can be linked with the proposed platform. The platform can be viewed as a getaway between VRSS and real world. Similar to our project, authors of [11] tried to make a computer-based system to link VRP without using GIS.

The paper has been organized in the following way: we begin with model of proposed platform in section 2. We then discuss the architecture of the platform and its different units along with structure of road map in the succeeding 4 sections. Section 7 and 8 deal with a method of node weight calculation and interaction with VRSS. The paper concludes with a real world VRS demonstration by solving Travelling Salesman Problem (TSP) pictorially.

2. Model of Proposed Platform

The proposed platform model has five unit components shown in Fig.1. The units are: the DBMS unit, the GIS unit, the VRS unit, the Control unit, and the User Interface unit.

2.1 The DBMS unit

The proposed DBMS plays an important roll in the platform. The spatial along with other logistic data are stored in the DBMS, and is known as Spatial DBMS [5]. The structure of the DBMS is relational type, that is, flexible to use in both GIS related or non-related applications. The ultimate goal is to construct a single database that will be used by all units of the platform without the need for formal data exchange mechanisms [3]. Almost all GIS software contain an internal relational data model to support small data sets as well as the facilities to support external commercial databases to handle large data sets.

2.2 The GIS unit

The GIS unit is responsible for processing and displaying spatial information such as user input and required output. This unit provides GIS based components like Feature, Layer, Label etc. available to the developers. In our project, we used MapInfo's MapX4.5 software as a GIS tool. Microsoft Visual Basic 6.0 (VB6) use MapX as one of its tools to build the GIS related applications.

2.3 The VRS unit

This unit actually is the solution/algorithm of a VRP. It receives data from control unit as input. Then after processing, sends the processed data back to control unit for next action.

2.4 The Control unit

The Control unit plays the central role of the proposed model. Every action originates and proceeds through this unit. The control unit can be sub-divided into the following four sub-units:

Table Maintenance unit: This unit is responsible for creating/editing a new/existing table. It also keeps track of necessary index files for faster record tracking.

Optimum Route Searching unit: The optimum Route searching unit calculates the optimum route by using the view of Dijkstra's method. According to output, it updates the 'Status' fields of Link table (discussed in sec. 6.2) for those records involved in optimum route. A better algorithm for optimum route searching between two nodes can also be used as an external unit.

Optimum Route Display unit: According to the 'Status' field of the 'Link' table updated by Optimum Route Searching unit, Optimum Route Display unit processes the necessary data for displaying the result. The output route map is displayed as separate GIS map layer.

Linking with VRS unit: discussed in sec. 8.1

2.5 The User Interface unit

The user can interact with GIS, DBMS or other modules through this unit. Different modules can be accessed by simply clicking the buttons or menu items. Interaction with GIS road map is also possible by clicking the mouse on the node point. There are several user-friendly interfaces that have been developed to accept and process user queries.

All the modules, interfaces, menus, forms etc. are designed with VB6 environment under windows 2000. Program module can interact with DBMS by using VB6's 'Data Access Object' or simply 'DAO'. VB6 is responsible for all required calculation, data management and data transfer from module/unit to module/unit.

3. Some Key Definitions

First of all, we need to define some key terms that will be used frequently in this paper. These are:

Node: A spatial point in the earth has a specific latitude and longitude.

Arc: Route between two consecutive nodes.

Cross-point: Common node at which two or more arcs meet.

Link: Route between two cross-points. A link is consists of one or more arcs.

'S' Node: Denotes the starting node of a Link.

'E' Node: Denotes the ending node of a Link.

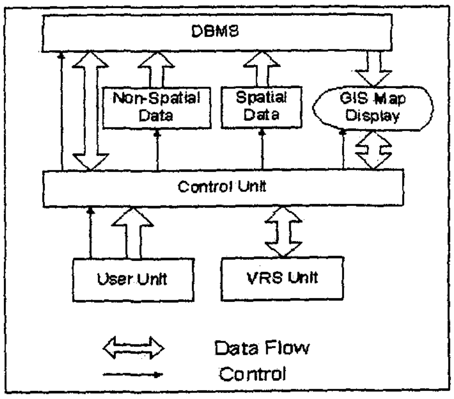

Fig.1 Proposed Model of SDSS

4. The Entity-Relation (ER) diagram

Fig.2 shows the ER diagram of the proposed DBMS consisting of six entities namely: Node, Link, Obstacle, Aysm_Link, Depot and Client. In Node to Link relation, each record in Link entity (table) involves with more than one record in Node entity. On the other hand each record in Node entity involves many records in Link entity, which suggests m:n relation. Link entity has another two relations 1:1 and 1:n with Asym_Link and Obstacle entities respectively. Each record in Depot and Client entity takes part in the relation Node_Depot (1:1) and Node_Client (1:n) relation but the reverse is not true. Because each record is responsible for creating the relation with Node entity makes Link, Depot and Client entities strong. Similarly Asym_link and Obstacle are strong entities compared to Link entity. Finally Aysm_link and Obstacle entities, on their own are weak entities.

5. Structure of the Road Map

Generally, the GIS road map consists of huge number of a nodes and links. Therefore the road map can be considered a directed graph and represented by the standard notation D={V, E}, where 'V' represents the nodes and 'E' represents the links [11]. All links are directed, so for an undirected link to be included in the digraph, two directed links are added in opposite directions. Each link has a weight, or "distance" value associated with it. In section 8 we discuss a method of link weight calculation. A link can have one or more arcs. The number of arcs as well as nodes increases with the curvature of a link. A link meets other links through its 'S'/'E' node. This suggests that a link can associate with at most two cross-points. A simple node-link structure of road map is shown in Fig.3. The large filled circled nodes (1,3,4,5) represent cross-points. According to Fig.3 link 1-3 consists of two arcs (1-2 and 2-3) and three nodes namely 1,2 and 3, of which nodes 1 and 3 are cross-points. The road map considered for this paper is the GIS road map of Okinawa Island, Japan. It consists of 3,42,534 nodes, 77333 links and 54116 unique cross-points.

The labeling process of nodes and links is very straightforward whereas for cross-points it is a little bit tricky. An algorithm has been developed to label the cross-points.

6. Design of Proposed Database

According to the guideline given in [3,6,8], we developed a database named as 'Road' to serve the purpose of vehicle routing. The 'Road' database is a Microsoft Jet Engine database and allows Microsoft Visual Basic 6.0 (VB6) to access and manipulate it via 'DAO'. Relational nature enables 'Road' database to establish relation among the tables. According to the entities in Fig. 2, there are six tables namely Node, Link, Obstacle, Aysm_Link, Depot and Client. These have been created to store and manipulate both spatial and non-spatial data effectively and efficiently.. The structure of each table are given as follows:

Table 1. 'Node' table to strore node information

Field	Purpose
NodeID	Unique number for each node
Lat	Latitude of the node
Lon	Longitude of the node
CPNO	Cross-point number of the node

Table 2. 'Link' table to store link information

Field	Purpose
LinkID	Unique number for each link
CPNOS	Cross-point number of 'S' node
CPNOE	Cross-point number of 'E' node
Width	Width of the link
Gradient	Gradient of the link
Weight	Weight of the link
Status	Contain the current status of the street (link)

Table 3. 'Obstacle' table to store obstacles

Field	Purpose
LinkID	Unique number for each link
Type	Type of the obstacle

Table 4. 'Asym_link table' to store Asymmetric links

Field	Purpose
LinkID	Unique number for each link
Width	Width of the link
Gradient	Gradient of the link
Weight	Weight of the link
Status	Current Status of the link

Table 5. 'Depot' table to store depot information

Field	Purpose
DepotID	Unique number for each Depot
Name	Name of the Depot
Address	Address of the Depot
Manager	Manager of the Depot
NodeID	Spatial location of the Depot

Table 6. 'Client' table to store each client's information

Field	Purpose
ClientID	Unique number for each client
Name	Name of the client
Address	Address of the client
NodeID	Spatial location of the client

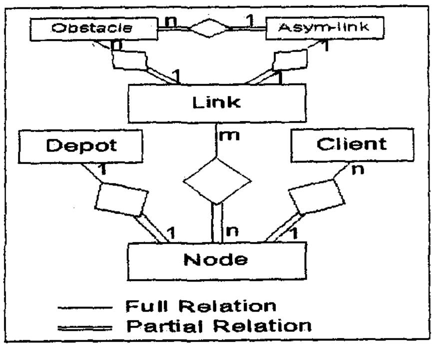

Fig 2: Entity-Relation Diagram

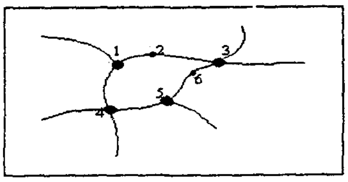

Fig3: Simple Road Structure

The tables discussed above are common to almost all VRP. Also, in order to fulfill the requirements, new tables can be created such as client order table and payment table, company employee table, vehicle table etc.

7. The Link Weight Calculation

The authors of the paper [4] showed some guidelines for calculating the link weight. The weight of a link primarily depends on the length, gradient and speed associated with it. The gradient produces impedance of the link, whereas the no. of lanes determine the speed along it. Apart from these, a new factor (not considered in [4]) called artificial obstacle is also considered here for weight calculation. Considering all the factors, suppose a link contains the following data:

Length = 'L' m, Impedance = 'Z' sec/100m

Speed = 'V' m/sec, Obstacles = X_i sec (all obstacles)

Now the weight of the link can be calculated using the following formula:

$$\text{Link\_wt(in second)} = L/V + L*Z + \sum X_i$$

In the succeeding sections we discuss the procedure for calculating the minimum weighted route between two nodes. When the length of a link is considered only to calculate the weight then we call the route as "Shortest path route", otherwise when gradient, speed and artificial obstacles along with the length are considered then the route is termed as "Optimum path route". Both options are available in the proposed model.

8. Interaction with VRSS

One of the main features of the proposed platform model is to interact with external VRSS. The VRSS works on several nodes (depots and customers) in earth. To optimize VRP, VRSS need shortest/optimum path between each pair of nodes known as cost matrix. The cost matrix will be written in a text file for the input of VRSS. The proposed system can provide shortest/optimum path using Dijkstra's method with the help of table for faster calculation.

8.1 Input and Output for VRSS

A VRSS containing 'n' nodes (cities) needs a cost matrix of order nxn to initialize the calculation. The platform model will provide the cost matrix by writing it to an external text file named 'Route.dat'. The VRSS reads the matrix from 'Route.dat' file for processing. After processing, VRSS produces the solution of VRP containing a list of nodes arranged sequentially to indicate route direction written in a separate text file named 'VRS_Route.dat'. Finally, SDSS visualize the optimum routes between the nodes found in 'VRS_Route.dat' file sequentially.

9. Demonstration

A VRSS solving generalized Travelling Salesman Problem (TSP) is implemented here as a demonstration. Hybrid-GA technique has been used to solve the TSP problem. Fig.4 and Fig.5 show the 'Route.dat' and 'VRS_Route.dat' text file respectively. The solution of the TSP containing 7 nodes is shown in Fig.6.

10. Conclusions

This paper has tried to provide an opportunity to link VRSS with real world spatial data. A platform model has been designed an implemented in conjunction with GIS and DBMS, which is capable of handling huge amounts of spatial and non-spatial data to solve complex VRPs. The development platform was Microsoft VB6 under the environment of Windows 2000. The developed system is suitable for PC and capable of drawing GIS map very quickly and efficiently. The system is tested with GIS road map of Okinawa Island, Japan.

Acknowledgement

The authors thank the Tropical Technological Center (TTC), Okinawa, Japan for providing GIS map road Okinawa Island associated with the project. Also the authors want to thank Mr. N. Azuma and Y. Matsuda for their support on this work.

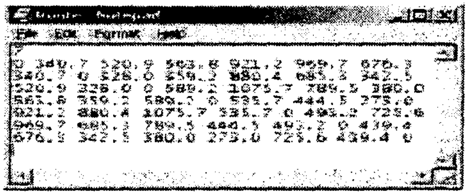

Fig4: Route File

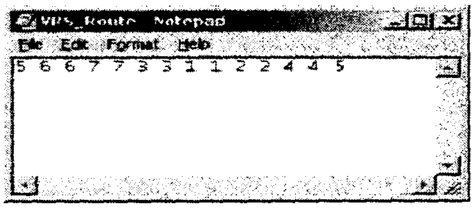

Fig5: VRS_Route File

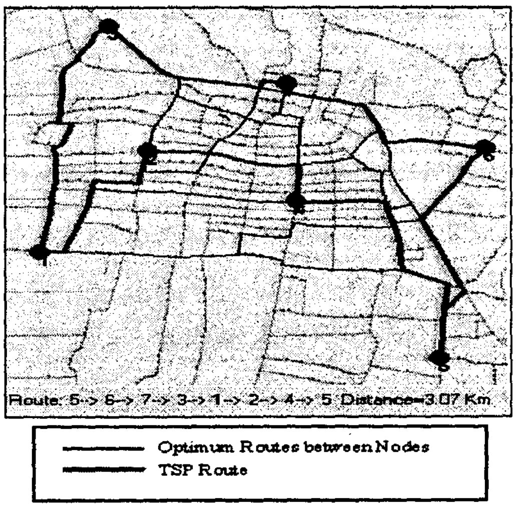

Fig6: Solution of TSP and Route Display

References

[1] C.D. Tarantilis, C.T. Kiranoudis, "Using a spatial decision support system for solving the vechicle routing problem", *Information & Management*, vol. 39, pp. 359-375, 2002

[2] P. Keenan, "Using a GIS as a DSS generator", *Working Paper MIS 95-9*, Graduate School of Business, University College of Dublin, 1997

[3] J. Allison (Al) Butler and Kenneth J. Dueker, "Implementing the Enterprise GIS in Transportation Database Design", *URISA Journal*, vol. 13, No. 1, pp. 17-28, 2001

[4] Peter W. Eklund, Steve Kirkby, Simon Pollitt, "A Dynamic Multi-source Dijkstra's Algorithm for Vehicle Routing", *Web Document*, http://citeseer.nj.nec.com/klund96dynamic.html

[5] Edward P.F. Chan, Janathan M.T. Wong, Pamela Y. Abbott, Sovannary Tan, "QL/L: A Spatial Database Management System", *Web Document* http://citeseer.nj.nec.com/

[6] Harvey J. Miller and Shih-Lung Shaw, "GIS-T MODELS", *Web Document* http://www.gisvisionmag.com/

[7] Sung Hun Song, Kwan Suk Lee, Gi Sub Kim, "A practical approach to solving a newpaper logistics problem using a digital map", *Computers & industrial engineering*, Article in press, CAIE1636, 2002

[8] Katy Appleton, Andrew Lovett, Gilla Sünnenberg, Trudie Dockerty, "Rural landscape visualisation from GIS databases: a Comparison of approaches, options and problems", *Computers, Environment and Urban Systems*, vol. 26, pp. 141-162, 2002

[9] Alexander Poot and Goos Kant, "A savings based method for real-life vehicle routing problems", *Econometric Institute Report EI 9938/A*, August 1, 1999

[10] C. Basnet, L. Foulds, M. Igbaria, "FleetManager: a micro-computer-based decision support system for vehicle routing", *Decision Support Systems*, vol. 16, pp. 195–207, 1996

[11] L.D. Bodin, D.J. Salamone, "The development of a micro-computer based system for vehicle routing and its use for solving spatial and temporal problems", *Mathematical and Computer Modeling*, vol. 11, pp. 558-562, 1988

[12] N. Christofides, "The vehicle routing problem", *RAIRO*, vol. 10, pp. 55–70, 1976

[13] G. Clarke, J.W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, vol. 12, pp. 568–589, 1964

[14] B.A. Foster, D.M. Ryan, "An integer programming approach to the vehicle scheduling problem", *Operational Research Quarterly*, vol. 27, pp. 367–384, 1976

[15] P. Keenan, "Spatial decision support systems for vehicle routing", *Decision Support Systems*, vol. 22, pp. 65–71, 1998

[16] D. Weigel, B. Cao, "Applying GIS and OR techniques to solve Sears technician-dispatching and home-delivery problems", Interfaces, vol. 29 (1), pp. 112–130, 1999

[17] Dang Vu Tung, Anulark Pinnoi, "Case Study: Vehicle routing scheduling for waste collection in Hanoi", *European Journal of Operational Research*, vol. 125, pp. 449-468, 2000

[18] G. Derekenaris, J. Garofalakis, C. Makris, J. Prentzas, S. Sioutas, A. Tsakalidis, "Integrating GIS, GPS and GSM technologies for the effective management of ambulances", *Computers, Environment and Urban Systems*, vol. 25, pp. 267-278, 2001